



HAL
open science

Modélisation et interprétation d'images à l'aide de graphes

Romain Lerallut

► **To cite this version:**

Romain Lerallut. Modélisation et interprétation d'images à l'aide de graphes. Mathématiques [math]. École Nationale Supérieure des Mines de Paris, 2006. Français. NNT : 2006ENMP1385 . pastel-00003298

HAL Id: pastel-00003298

<https://pastel.hal.science/pastel-00003298>

Submitted on 5 Aug 2010

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 obtenir le grade de
Docteur de l'École des Mines de Paris

Spécialité «Morphologie Mathématique»

Présentée et soutenue publiquement
par

Romain LERALLUT

le 13 Septembre 2006

Modélisation et Interprétation d'Images à l'Aide de Graphes

Directeur de thèse : Fernand MEYER

Jury :

Dominique Jeulin	<i>Président</i>
Renaud Keriven	<i>Rapporteur</i>
Marc Van Droogenbroeck	<i>Rapporteur</i>
Monique Thonnat	<i>Examineur</i>
Lionel Martin	<i>Examineur</i>
Fernand Meyer	<i>Directeur de thèse</i>

*A mes parents
A ma famille
A mes amis*

Twenty years from now you will be more disappointed by the things that you didn't do than by the ones you did do. So throw off the bowlines. Sail away from the safe harbor. Catch the trade winds in your sails. Explore. Dream. Discover.

Mark Twain

Remerciements

Les enfants seuls savent ce qu'ils cherchent

Antoine de Saint-Exupéry, *Le Petit Prince*

Je tiens à remercier en premier Fernand Meyer, non seulement de m'avoir accueilli au CMM et dirigé mes travaux, mais surtout de m'avoir laissé trouver ma propre voie. Cette grande liberté a parfois été un peu effrayante et dure à porter mais je n'aurais pas autant apprécié ma thèse sans elle.

Merci également à Dominique Jeulin d'avoir présidé le jury de ma soutenance, à Renaud Kériveren et Marc Van Droogenbroeck d'avoir accepté d'être les rapporteurs de mon travail, ainsi qu'à Monique Thonnat et Lionel Martin d'être venus représenter les partenaires du projet «PACA». Merci à Michel Bilodeau de m'avoir encadré dans ce projet et de m'avoir fait confiance quand je promettais la Lune !

Le CMM ne serait pas le CMM sans ses indispensables anges gardiens, Catherine et Laura, ni sans son équipe de chercheurs. Parmi eux, je remercie tout particulièrement Michel, Etienne et Beatriz non seulement pour tout ce qu'ils m'ont appris mais aussi pour toutes les pauses-café, déjeuners-débats explosifs et soirées endiablées que nous avons passés ensemble.

Je remercie spécialement Raffi Enficiaud, l'autre «développeur fou», frère d'armes et compagnon de guérilla dans la conception et le développement de Morphée. Nous avons tant appris l'un de l'autre ! De nos différences, nous avons su tirer une grande force. J'aurai également toujours une pensée émue pour ceux qui ont partagé mon bureau : Mathilde, Gabriel, Timothée et Thibault. Je serai le premier à reconnaître que la tâche n'a pas toujours été aisée, mais mon séjour au CMM aurait été amoindri de leur absence. Mathilde me permettra de reprendre sa phrase : «Vous êtes un peu de ma famille maintenant».

Une graine ne peut pousser correctement que si elle est plantée dans une terre fertile, et régulièrement arrosée. Rien de tel pour cela que le gang des thésards de Fontainebleau. Ceux du CMM bien sûr (Costin, Jaromir, Max, Thomas R, Thomas W) et ceux des centres voisins (Aurélie, Caroline, l'autre Caroline, Julie, Rosalie, Sunsea, David, François, Hugo, Marc, Papy Hugues, Pedro). Merci pour les soirées, les week-ends, l'escalade, les fans en délire à nos concerts de «rock», les champignons, les cœurs s'épanchant sur le zinc du Martin's, les vacances, les débats sur la Vie, l'Univers, et le Cinéma. . .

Enfin, last but not least, je tiens à remercier ma famille et mes parents pour leur attention constante mais discrète, leur intérêt pour mes travaux et leur enthousiasme m'incitant à toujours chercher de nouveaux défis et de nouveaux horizons.

«Les enfants seuls savent ce qu'ils cherchent», merci à tous ceux qui m'ont permis de rester enfant encore quelques temps.

Résumé

L'analyse et la comparaison intelligentes d'images sont parmi les sujets suscitant le plus d'intérêt dans les milieux académiques autant qu'industriels. Décrire et comparer automatiquement les images est en effet un enjeu critique pour le plein développement de la «société de l'information».

Les moteurs de recherche fonctionnant sur le texte ont prouvé leur utilité de façon éclatante mais à l'heure actuelle il n'existe aucun système équivalent fonctionnant uniquement sur les images. Une explication possible est que nous ne disposons pas de langage permettant de décrire les images et que les comparaisons pertinentes sont ainsi beaucoup plus difficiles que dans le cas du texte. Cependant, le cas du texte nous montre qu'il n'est pas nécessaire que les machines comprennent ce qu'elles analysent pour renvoyer des résultats pertinents. Des méthodes simples d'analyse syntaxique associées à des règles de composition suffisent à piloter des moteurs de recherche d'une grande efficacité.

Pour permettre à des machines de simuler l'interprétation des images, il faudrait donc créer des descripteurs faisant office de mots et des règles pour les regrouper, ce qui permettrait de comparer des scènes comme on compare des phrases. On dispose d'ores et déjà de nombreuses méthodes pour détecter automatiquement de petits objets et des régions dans des images, par leur couleur commune, leur mouvement identique, etc. Poursuivant l'analogie, on pourrait comparer ces petits objets à des syllabes. La difficulté consiste à les grouper en mots, puis en phrases et comparer celles-ci, tout en étant robuste face aux perturbations.

Pour ce faire, nous utilisons des graphes pour stocker ces objets et leurs relations. Ces relations peuvent être de voisinage ou d'inclusion, ce qui conduit les graphes à être respectivement des graphes plans ou des arbres. Nous verrons ainsi plusieurs méthodes permettant de construire l'un ou l'autre type de représentation, ainsi que leurs avantages et inconvénients.

Dans une première étape, nous avons utilisé les algorithmes d'appariement de graphes développés par Cristina Gomila à la fin de sa thèse au CMM (1998-2001). Profitant du projet européen MASCOT étudiant l'utilisation de «métadonnées» pour faciliter le codage vidéo, nous avons étudié en détail les forces et faiblesses de cette approche. Nous avons d'abord testé le remplacement de l'algorithme au cœur de l'appariement de graphes. Nous avons obtenu une légère amélioration de la stabilité et également de meilleurs temps de calcul. Puis nous avons cherché à améliorer notre robustesse face aux variations de segmentation en utilisant une projection dans le domaine spectral. Malgré de bons résultats sur des images simples, nos essais sur des images plus difficiles n'ont pas été couronnés de succès. Pour pallier cette fragilité dès que les graphes ne sont plus similaires, nous avons préféré revenir à notre matériau source, les images.

La seconde étape de ce travail a porté sur le développement de techniques basées sur l'image pour réduire la sensibilité de nos algorithmes de segmentation au bruit et aux petites variations. Pour ce faire, nous avons développé une classe d'opérateurs de filtrage adaptatifs, les «amibes morphologiques», extrêmement efficaces pour réduire le bruit dans les images. Par ailleurs, nous avons également développé un opérateur de gradient couleur robuste permettant de mieux détecter les contours

dans les images bruitées. Ces deux opérateurs ont amélioré de façon parfois impressionnante la stabilité de nos modélisations, puis de nos graphes et donc des résultats globaux.

L'étape suivante dans ce travail a porté sur le développement de modélisations d'objets indépendamment du reste de l'image. La motivation derrière cette approche est de considérer que, dans certains scénarios, le contenu de l'image, hors de certains objets bien définis, n'est pas informatif. Il faut donc analyser directement et de la façon la plus précise possible les objets eux-mêmes. Nous avons dans un premier temps supposé que les segmentations des objets étaient connues, afin de nous concentrer sur le calcul d'une signature robuste de chaque objet. Pour l'obtenir, nous avons modifié un algorithme de ligne de partage des eaux pour effectuer une resegmentation «top-down» d'un espace d'échelle morphologique basé sur des nivellements. Ceci a donné lieu à une nouvelle modélisation robuste utilisant des arbres de régions imbriquées. Nous avons également développé une distance entre ces arbres et nous l'avons testée sur une base d'images classique dans le domaine de l'indexation.

La dernière étape est centrée sur l'aspect applicatif. En premier lieu en comparant les différentes approches présentées dans ce travail, notamment aux niveaux de leur robustesse et de leur vitesse d'exécution.

Enfin, nous avons cherché la meilleure combinaison de techniques pour concevoir une application de vidéosurveillance. En particulier, nous avons développé des techniques rapides et robustes de segmentation dans le cadre du projet PS26-27 «Environnement Intelligent» en collaboration avec ST Microelectronics et le groupe ORION de l'INRIA. Ce projet visait à construire un démonstrateur de technologies de vidéosurveillance appliquées à la détection d'accidents dans les cadres domestique et hospitalier. Notre part du travail consistait à la mise au point d'algorithmes de détection de silhouettes en mouvement dans des séquences vidéo.

Ainsi, en couplant ces techniques à nos descripteurs d'objets par arbres, nous avons pu définir des signatures robustes de personnes, qui pourront être utilisées avec un grande efficacité dans des systèmes automatisés de vidéosurveillance.

Mots-clefs : segmentation, espaces d'échelle morphologiques, graphes, arbres, réduction de bruit, amibes morphologiques

Abstract

Intelligent analysis and comparison of images is one of the most dynamic topic of research in both academia and industry. Describing and comparing images automatically is a critical issue for the full development of the «information society»

Search engines working on textual data have dramatically proved their value. However, there is currently no similar system for image-only data. One possible explanation is that we do not really have a language made for describing images, thus meaningful comparisons are much more difficult than in the case of text. Nevertheless, textual search engines have shown that it is not necessary that the machines understand what they analyse to return good results. Simple syntactic analysis methods, coupled with composition rules are enough to drive extremely efficient systems.

To enable machines to simulate the interpretation of images, it would be necessary to create descriptors playing the role of words in text and composition rules making it possible to compare images like search engines compare sentences. We already have at our disposal numerous methods to detected automatically simple objects or regions in images, by their common color, their identical motion, etc. Furthering the analogy, these objects could be seen as syllables. The difficulty lies in grouping them to form words, then sentences, and compare them while being robust to perturbations.

To achieve this, we use graphs to store these objects and their relationships. These can be either of a neighboring nature, or inclusion, which leads the graphs to be either planar graphs or trees. We will see several methods to construct either type as well as their pros and cons.

In a first step, we have used the graph-matching algorithms developed by Cristina Gomila at the end of her PhD thesis at the CMM (1998-2001). While working with the european project MASCOT studying the use of «metadata» to enhance video coding, we have studied in detail the algorithm and spotted its strengths and weaknesses. We first tested replacing the core of the matching algorithm by a better one. This resulted in slight improvements in both quality and computation time. Then we tried to reduce our sensitivity to variations in the segmentation process by using a spectral graph-matching algorithm. Despite good results on simple images, our tests on harder images have not succeeded. To improve our robustness with respect to the stability of the graphs, we then preferred working on the source material : images.

The second step of this work was the development of image-base techniques to reduce the sensitivity of our segmentation algorithms to noise and small variations. First, we developed a class of adaptive filtering operators, the «morphological amoebas», which proved extremely effective in reducing noise in image. Second, we created a robust color gradient operator that can detect contour lines in noisy images. These two operators have improved sometimes spectacularly the stability of our segmentations, hence that of our graphs and in the end the quality of the results.

The next step in this work has been the modeling of objects independently from the rest of the image. This approach was motivated by realizing that in some scenarii the content of the image out-

side some well-defined objects is not informative. We must thus analyse directly and as precisely as possible the objects themselves. We first supposed that the segmentation of the outline of the objects was a solved problem, and concentrated on creating a robust signature for each object. To get it, we modified a watershed algorithm in order to perform a top-down resegmentation of a morphological scalespace based on levelings. We used this resegmentation to build a robust tree of embedded regions, and we defined a distance between those trees. We tested the whole process on a commonly used database by the indexation community.

The last step was centered around applications. First by comparing the various approaches presented in this document, concentrating in particular on the speed versus robustness compromise.

Then we search for the best combination of techniques to build a videosurveillance application. In particular, we developed fast and robust segmentation techniques for the project PS26-27 «Intelligent Environment» in partnership with ST Microelectronics and the ORION group of the INRIA. This aim of this project is to build a technology demonstrator for videosurveillance applied to the detection of accidents in hospitals or at home. Our part of the work was the detection of the outline of people in video sequences.

Finally, by coupling these detectors to our tree-based objects descriptors, we were able to define robust signatures for people that could be use with great profit by automatic videosurveillance systems.

Keywords : segmentation, morphological scalespace, graphs, trees, noise reduction, morphological amoebas

Table des matières

1	Introduction	1
1.1	Motivation : structurer l'information dans les images	1
1.2	Modélisation par graphes	2
1.3	Plan de la thèse	2
I	Première approche	5
2	Problématique initiale	9
2.1	La modélisation par graphes	9
2.2	Le problème de la segmentation	13
2.3	L'algorithme d'appariement de Cristina Gomila	16
2.4	Le projet MASCOT	19
2.5	Premières conclusions et perspectives	26
3	Amélioration de l'appariement de graphes	27
3.1	Introduction	27
3.2	Approches «one-to-one»	28
3.3	Approche «many-to-many»	34
3.4	Synthèse et conclusion	39
II	Filtrages	47
4	Filtrage d'images par amibes morphologiques	51
4.1	Introduction	51
4.2	Problématique originelle	52
4.3	Principe	55
4.4	Construction	57
4.5	Propriétés	60
4.6	Amibes et morphologie mathématique	61
4.7	Les amibes en pratique	66
4.8	Résultats	67
4.9	Conclusions et perspectives	77

5	Gradients	81
5.1	Introduction	81
5.2	Réflexions sur le gradient et la segmentation	82
5.3	Le problème de la couleur	83
5.4	Gradients morphologiques généralisés	84
5.5	Résultats	87
5.6	Conclusion	96
III	Modélisation par arbres	101
6	Inondations contrôlées	105
6.1	Introduction	105
6.2	Segmentation par LPE contrôlée	106
6.3	Espaces d'échelle morphologiques	109
6.4	Segmentation «top-down»	110
6.5	Le gradient «pyramide»	119
6.6	Conclusion	122
7	Appariement d'arbres	127
7.1	Distance d'arbres	127
7.2	Recherche d'images semblables dans une base	129
7.3	Robustesse face aux perturbations	134
7.4	Conclusion	137
IV	Applications	141
8	Résultats	145
8.1	La segmentation	145
8.2	Appariement de graphes	160
9	Application à la vidéosurveillance	167
9.1	La problématique de la vidéosurveillance	167
9.2	Calcul de la silhouette	171
9.3	Création et suivi d'une signature	180
9.4	Conclusion	191
V	Conclusions et Annexes	193
10	Conclusion	195
10.1	Apports de cette thèse	195
10.2	Perspectives	197

A Amibes morphologiques : implémentation avec Morphée	199
A.1 Introduction	199
A.2 Spécificités	199
A.3 Adaptabilité	200
A.4 Voisinages imbriqués	200
A.5 Conclusion	203
Bibliographie	205

Chapitre 1

Introduction

千里之行，始于足下

*Un chemin de mille lis
commence par un seul pas.*

Proverbe chinois

Dans ce chapitre nous présenterons le cadre général de ce travail, en le rattachant notamment aux travaux antérieurs réalisés au CMM. Nous introduirons brièvement la problématique avant de décrire la structure de ce document.

1.1 Motivation : structurer l'information dans les images

La quantité d'information à laquelle nous avons accès croît de façon parallèle à l'augmentation des moyens de création et de stockage. Cependant, elle croît également plus vite que notre capacité à l'indexer. Ceux qui étaient déjà connectés aux débuts d'Internet se souviennent des quantités épuisantes de «signets» qu'il fallait conserver pour naviguer efficacement et retrouver les sites intéressants. De nos jours, cette tâche est largement facilitée par l'efficacité des moteurs de recherche. Le succès de Google, pour ne citer que lui, tient à deux choses : d'une part sa capacité à trouver l'information pertinente, d'autre part à la trouver rapidement. Pour répondre à ce besoin, il faut structurer l'information, et pour le faire vite, il faut le faire de la façon la plus synthétique possible.

À l'heure actuelle, il n'existe pas de moteur de recherche fonctionnant directement sur les images de façon aussi universelle que sur le texte. La raison à cela est que l'information stockée dans les images n'est pas aussi «facile» à analyser que celle contenue dans le texte. Les mécanismes d'analyse qui nous permettent de comprendre ce que nous voyons interviennent de façon principalement inconsciente et sont encore très mal connus. Nous connaissons les plus simples et les plus primitifs, mais nous sommes encore loin de pouvoir répliquer dans des machines notre capacité à utiliser nos connaissances pour interpréter une scène.

Pour Kant, «toute langue est désignation de la pensée et inversement, le meilleur mode de désignation de la pensée passe par le langage» (*Didactique anthropologique*), or nous ne disposons pour décrire les images que du langage naturel. Cependant, décrire une image complexe en langage naturel de façon utilisable reviendrait à concevoir une réelle Intelligence Artificielle ; ce n'est donc pas dans cet axe que nous devons orienter nos recherches. D'un autre côté, les moteurs de recherche textuels ne

comprennent pas les documents qu'ils analysent et se contentent simplement de compter les mots et d'utiliser des règles simples de composition. Par analogie, nous devons donc nous donner un langage descriptif des éléments constitutifs d'une image, ainsi que les règles d'associations pour former des «phrases», pour pouvoir comparer, trier et analyser automatiquement des images.

L'objectif principal de cette thèse est de développer des méthodes permettant de modéliser puis de comparer l'information présente dans des images. Sans aller jusqu'au niveau sémantique, nous chercherons à définir une représentation intermédiaire permettant d'accomplir de façon automatique certaines tâches d'interprétation. Les deux domaines d'applications visés sont l'indexation par le contenu (CBIR chez les anglo-saxons, pour «Content-Based Image Retrieval») et le suivi intelligent d'objets dans des séquences vidéos. Ces deux sujets apparemment éloignés partagent en effet le même besoin d'outils efficaces de structuration et de comparaison de l'information.

1.2 Modélisation par graphes

Les graphes sont des objets mathématiques permettant de modéliser des objets et leurs relations. À ce titre, ils sont très bien adaptés à la représentation des données présentes dans une image. On pourra ainsi modéliser les objets présents dans une scène, leurs positions relatives, les sous-objets qui les composent, etc.

Cette modélisation nous semble se situer à un niveau intermédiaire intéressant, s'approchant du niveau sémantique par les relations entre objets tout en restant proche de l'image et raisonnablement facile à calculer. Par ailleurs, elle nous permettrait d'utiliser l'immense corpus mathématique formé par la théorie des graphes et ses concepts affiliés. En effet, depuis le milieu du XX^{ème} siècle et en particulier l'essor de la recherche opérationnelle, les graphes ont quitté le domaine des curiosités mathématiques pour devenir des outils incontournables de modélisation et de prévision.

Nous verrons au cours de cette thèse plusieurs représentations d'images et d'objets sous formes de graphes, leurs utilisations, leurs avantages et inconvénients. En particulier, nous tâcherons de comprendre comment et pourquoi choisir une modélisation plutôt qu'une autre en tenant compte des spécificités des domaines d'applications et des éventuelles contraintes industrielles ou de performances.

1.3 Plan de la thèse

Dans ce travail, nous avons procédé selon trois axes de façon globalement chronologique et suivant l'évolution de notre compréhension du problème. L'agencement de ce manuscrit reflète cette organisation.

Dans une première partie nous présenterons l'approche proposée par Cristina Gomila. Nous proposerons des extensions et nous analyserons les limites de cette approche. En particulier, nous verrons que le maillon faible de cette chaîne de traitement réside dans l'instabilité de la segmentation des images. Ces problèmes sont amplifiés dans le cas de la vidéosurveillance utilisant des caméras bon marché. Nous détaillerons ensuite les filtres que nous avons conçus pour améliorer les images, donc leurs segmentations, et ainsi augmenter la portée des premières méthodes. Cependant, certains problèmes inhérents à ce type d'approche ne pourront être résolus qu'avec une approche considérant les objets individuellement plutôt qu'une modélisation complète de la scène. La création et la comparaison de ces modèles sera l'objet de la troisième partie. Enfin, nous consacrerons une dernière partie d'une part à des benchmarks entre les différentes méthodes que nous proposons et d'autre part à la réalisation d'une chaîne complète de traitement dans le cadre d'une problématique réelle d'indexation pour la vidéosurveillance.

1.3.1 Première partie : extension des techniques initiales

Cette thèse s'appuie initialement sur les travaux réalisés par Cristina Gomila au cours de sa thèse soutenue au CMM en 2001. Plus spécifiquement, sur les techniques de construction et d'appariement de graphes qu'elle a développées vers la fin de sa thèse.

Dans cette première partie, nous analyserons puis nous étendrons les algorithmes de Cristina en les appliquant à des domaines légèrement différents. Nous présenterons certains des essais réalisés dans le cadre du projet européen MASCOT de développement de technologies pour le codage de flux vidéos. Par la suite, nous proposerons des modifications plus ou moins importantes de cette approche, via d'autres algorithmes de comparaison de graphes. Nous étudierons également l'élargissement du champ d'application de ces méthodes en les testant sur des images plus difficiles mais surtout plus proches des problématiques industrielles réelles. Nous tâcherons d'identifier les problèmes liés à la modélisation proprement dite et ceux liés à son utilisation. Nous verrons ainsi qu'une majorité des problèmes est liée à la qualité de la modélisation, en particulier aux segmentations utilisées pour construire les graphes.

1.3.2 Seconde partie : filtrages

Ayant constaté certaines limitations dans nos premières approches, en particulier dans le cas des images difficiles, nous nous recentrerons ici sur l'image en proposant deux filtrages permettant de retrouver de bonnes modélisations malgré des conditions peu favorables.

Le premier filtrage utilise des «amibes morphologiques» qui sont des éléments structurants pouvant adapter leur forme au contenu de l'image sur laquelle on les applique. Ces amibes nous ont permis de construire des filtres extrêmement efficaces pour débruiter des images, en particulier dans un but de segmentation.

Le second développement exposé dans cette partie sera une étude sur la notion de gradient couleur. Nous proposerons plusieurs améliorations au gradient morphologique généralisé à la couleur, dans le but de le rendre plus résistant au bruit.

Ces deux approches sont indépendantes et complémentaires, on peut les utiliser séparément ou successivement. La seule contrainte est la puissance de calcul dont on dispose. L'une ou l'autre nous ont permis de modéliser par des graphes des images qui nous causaient des difficultés précédemment. Nous avons pu ainsi étendre la validité des travaux de la première partie à des images plus difficiles.

1.3.3 Troisième partie : modélisation orientée-objet

Ayant reconnu dans la première partie les limites de la modélisation de scènes, nous étudierons dans cette partie une technique de modélisation d'objets par des arbres, en appliquant une variante de la Ligne de Partage des Eaux (LPE) sur une représentation de l'image par espaces d'échelle morphologiques. Cette représentation sera une arborescence des régions et sous-régions de l'objet considéré.

Nous présenterons dans un second temps une mesure de similarité entre ces arbres, et nous étudierons la robustesse de l'ensemble face au bruit, aux rotations et aux changements d'échelle. Cette approche, couplée à un détecteur d'objets, représente un excellent candidat pour l'indexation en particulier grâce à ses propriétés d'invariance et de robustesse.

1.3.4 Quatrième partie : résultats et applications

Dans une première étape, nous comparerons les diverses approches que nous avons développées au cours de ce travail, en nous intéressant aux différents compromis entre qualité des résultats et

vitesse de calcul.

Nous construirons enfin une chaîne complète de traitement d'images de type vidéosurveillance utilisant plusieurs des briques développées dans les parties précédentes. On détaillera notamment la détection et le suivi de silhouettes, qui ont fait l'objet de développements spécifiques dans le cadre du projet PS26-27 «Environnement Intelligent». L'objectif de ce projet était d'analyser la posture d'une personne, dans le but de détecter les situations anormales (chutes, crises d'épilepsie, etc), en particulier dans le milieu hospitalier. Ce projet parrainé par ST-MicroElectronics a été réalisé en collaboration avec le groupe ORION de l'INRIA, notre rôle consistant dans un premier temps à fournir les segmentations les plus précises possibles des personnes, le plus rapidement possible. Nous verrons en particulier comment nous avons pu gérer ces impératifs de vitesse d'exécution sans pour autant sacrifier la qualité des résultats. Dans un second temps, nous présenterons l'utilisation des outils de modélisation développés dans la partie précédente pour l'aide à l'interprétation et à la structuration des séquences vidéo.

Première partie

Première approche

2	Problématique initiale	9
2.1	La modélisation par graphes	9
2.2	Le problème de la segmentation	13
2.3	L'algorithme d'appariement de Cristina Gomila	16
2.4	Le projet MASCOT	19
2.5	Premières conclusions et perspectives	26
3	Amélioration de l'appariement de graphes	27
3.1	Introduction	27
3.2	Approches «one-to-one»	28
3.3	Approche «many-to-many»	34
3.4	Synthèse et conclusion	39

Première Approche : extension des techniques initiales

Le début de ce travail s'inscrit dans le prolongement de la thèse de Cristina Gomila, soutenue au CMM en 2001 [Gomila, 2001], en particulier dans les derniers développements sur le suivi d'objets à l'aide de segmentations et de graphes. Le principe est d'encoder la structure de l'image dans un graphe représentant les régions principales de l'image et leurs relations de voisinage. Pour comparer deux images, on compare leurs graphes. En particulier, on peut effectuer un appariement qui vise à associer chaque nœud du premier graphe à un nœud du second.

Cette approche, développée initialement pour le suivi du locuteur dans des séquences de vidéoconférences a été élargie pour fournir un cadre général permettant de suivre les objets dans une séquence vidéo moins strictement formatée.

Dans le chapitre 2 nous décrivons cet algorithme, ainsi que nos tentatives d'utiliser ces méthodes de suivi pour améliorer la qualité de la compression d'un codeur vidéo dans le cadre du projet MAS-COT. Nos essais de réutilisation et d'extensions de cette approche nous ont permis d'identifier ses points faibles que nous chercherons à renforcer dans les chapitres suivants.

Le chapitre 3 sera consacré au remplacement du moteur d'appariement pour améliorer sa robustesse face aux erreurs de segmentation. Nous verrons deux approches : l'une basée sur une méthode d'optimisation et la seconde sur du «clustering» dans le domaine spectral.

Chapitre 2

Problématique initiale

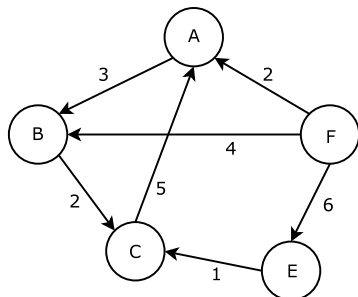
*If you think it's simple, then you have
misunderstood the problem.*

Bjarne Stroustrup

Dans ce chapitre nous présenterons d'abord les notions générales de la modélisation par graphes avant de décrire l'algorithme développé par Cristina Gomila. Nous décrirons ensuite nos apports dans le cadre de cette approche pour finir par les limitations que nous avons rencontrées.

2.1 La modélisation par graphes

2.1.1 Historique et définitions

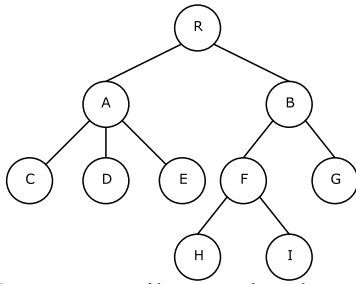


Un graphe est un outil mathématique permettant de représenter de manière synthétique des objets (les nœuds) et leurs relations (les arêtes). D'après Gondran et Minoux [Gondran and Minoux, 1985], la théorie des graphes aurait été fondée en 1736 par Euler, qui a posé le premier problème de cheminement dans son *Solutio problematis ad geometriam situs pertinentis*. Des noms illustres y sont attachés, tels Cauchy (1813) et Poincaré (1900).

On définit en général un graphe G comme la donnée de deux ensembles N et A . Les éléments de N sont appelés les *nœuds* (ou parfois les sommets) du graphe. On notera n le cardinal de l'ensemble N et on numérotera les nœuds $i = 1 \dots n$. Les éléments de A sont des couples (i, j) de nœuds de N que l'on appelle les *arêtes* du graphe.

On peut attacher aux nœuds et/ou aux arêtes des données représentant ce que l'on cherche à modéliser : distance entre deux points, capacité d'un tube, couleurs moyennes, etc. On parle alors de graphes «valués». Dans l'exemple ci-dessus, les nœuds sont les points A,B,C,E et F, et les arêtes sont représentées avec le poids qui leur est attaché.

Les graphes peuvent être *orientés* ou non. Dans un graphe orienté les arêtes ne peuvent être parcourues que dans un seul sens, ce qui est important dans les problèmes de cheminement. Par la suite sauf mention explicite, nous n'utiliserons que des graphes non-orientés.



Les *arbres* sont un cas particulier de graphe. Dans notre cas, il s'agira d'arbres ayant une racine (R) sans ordre sur les branches. Les arbres sont des graphes dans lesquels de nombreux problèmes ont une complexité théorique moindre. Ils sont particulièrement appropriés pour représenter des structures hiérarchiques (généalogie, taxinomie, etc).

Dans ce travail, nous chercherons d'abord à modéliser des images en construisant des graphes synthétisant les informations présentes dans les images, par exemple les couleurs moyennes des régions, la force des contours, etc. Dans un second temps, nous utiliserons ces modélisations pour comparer les images entre elles, trouver des objets communs, et plus généralement décrire la structure des images.

2.1.2 Modélisation de scènes

Dans le domaine du traitement d'images en général et de la reconnaissance des formes en particulier, un objet physique peut être considéré principalement de deux façons :

1. comme un objet existant dans un espace à trois dimensions
2. comme la projection bidimensionnelle de cet objet dans le plan image

Cette distinction traduit deux approches radicalement opposées de l'analyse de formes : la première tente de reconstruire la scène dans sa totalité, en particulier avec l'information de profondeur, alors que la seconde se base principalement sur l'étude des images. En général la première approche n'est possible qu'avec de l'information extérieure, comme un modèle 3D prédéfini de la scène, alors que la seconde n'est pas efficace dans les scènes les plus complexes.

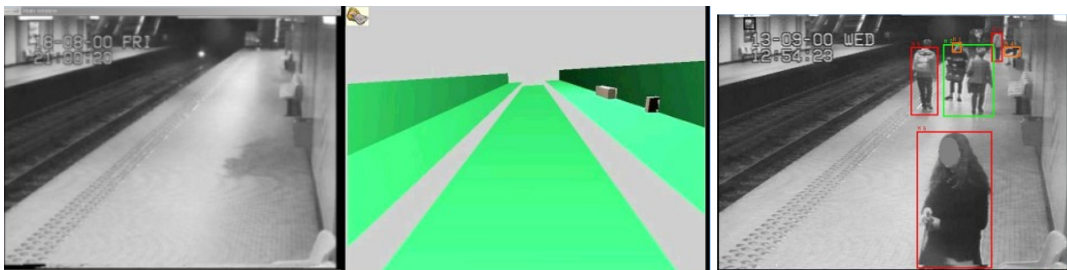


FIG. 2.1: Modélisation 3D d'une scène pour permettre la détection des individus et de leur positions. Gauche : une des vues utilisées pour la reconstruction 3D. Centre : le modèle 3D reconstruit par stéréo. Droite : détection des objets présents. (source : INRIA/Orion).

Comme la création d'un modèle 3D de scène est une opération coûteuse, effectuée manuellement, et ce pour toute nouvelle implantation, il est souvent préférable d'améliorer l'approche 2D, malgré ses limitations. Notre travail consistera donc à modéliser la scène vue par une seule caméra, en détectant en particulier les objets en mouvement. Pour ce faire, nous allons utiliser des graphes pour décrire ces objets et leurs relations.

2.1.3 Construction de graphes par segmentations

Étant donné une image composée de pixels disposés sur une trame, le graphe le plus immédiat est celui dans lequel chaque pixel est un nœud et dont les arêtes relient les pixels voisins entre eux. Ce type de graphe n'est qu'un changement de représentation des données de l'image et ne cherche pas à

synthétiser l'information présente dans l'image. Par ailleurs, ces modèles non simplifiés sont souvent si gros que des algorithmes classiques de théorie des graphes ont un coût prohibitif. Pour améliorer la pertinence de cette approche, on va donc effectuer en premier lieu une segmentation de l'image en zones d'intérêt pour regrouper les pixels possédant des caractéristiques similaires.

L'objectif est de séparer l'image en un nombre de zones restreint mais néanmoins suffisant pour que chaque zone soit bien représentative des pixels qui la composent. Une fois cette opération effectuée, l'idée est que chaque objet de l'image sera représenté par seulement quelques régions. On associe ensuite à l'image segmentée un graphe traduisant les relations (arêtes) entre ces régions (nœuds). La difficulté réside dans la détermination des régions décrivant «au mieux» les objets présents dans l'image, car il est difficile de construire des critères objectifs et mesurables de notions fortement attachées à la *signification* de l'image.

L'opération de segmentation se déroule généralement en deux passes. La première passe est la création de petites régions de couleurs très uniformes, permettant de passer d'un million de pixels (par exemple) à quelques milliers de régions. Sauf cas pathologiques, cette étape fonctionne bien et nécessite rarement de précautions particulières. Cependant, la segmentation ainsi obtenue n'est pas suffisamment descriptive pour pouvoir être directement utilisée par des algorithmes de haut niveau. En effet, il ne s'agit que d'une simplification de l'image, qui ne présente pas réellement de caractère sémantique¹.

La seconde passe, beaucoup plus critique, vise à simuler l'approche sémantique à l'aide de règles simples. Par exemple, on cherchera à regrouper les régions de couleur voisines ou de mouvement similaire. On cherchera à éliminer les petites régions, en particulier si elles sont peu contrastées, etc. Cette étape est fragile et «non-continue» car devant effectuer des choix qui pourront s'avérer différents entre deux images pourtant semblables. C'est cette fragilité qui complique énormément notre travail, car elle impose une grande incertitude sur la validité de la modélisation par graphes. Nous devons donc être prêts à ce que les différences entre graphes soient dûes soit à des images réellement dissemblables, soit à des choix de modélisation divergents.

En morphologie mathématique, les méthodes habituelles pour effectuer cette fusion sont de type «bottom-up» : elles partent d'une sur-segmentation obtenue par une LPE et fusionnent les régions jusqu'à obtenir le nombre voulu de régions. Cette approche est très rapide mais également instable, car les informations stockées dans les petites régions ne sont plus nécessairement représentatives quand on considère des agrégats plus gros. Nous proposerons dans le chapitre 6 une approche inverse, «top-down» basée sur des resegmentations successives. On verra que cette approche est très robuste mais également très coûteuse en temps. Nous finirons donc ce chapitre en proposant une approche hybride pour définir un gradient possédant certaines des propriétés de l'approche top-down, et que l'on segmentera par une approche bottom-up.

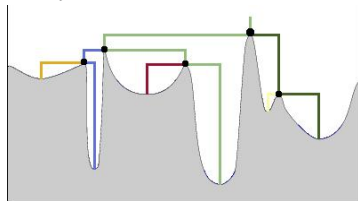
Pour ce qui est des graphes, nous verrons dans cette thèse deux approches pour la représentation des relations entre régions. La première utilise les relations d'inclusion entre régions et sous-régions, et la seconde des relations de voisinage entre régions du même niveau. Nous verrons par la suite trois méthodes pour gérer les incertitudes de modélisation : d'abord l'édition de partitions, dans ce chapitre, puis l'appariement «many-to-many» dans le chapitre 3 et enfin les distances d'arbres robustes dans le chapitre 7.

¹ Si l'on considère que privilégier les régions de couleur uniforme n'est pas *déjà* imposer un sens à l'image.

2.1.4 Graphes de hiérarchie

Construction

Ces graphes traduisent les relations d'inclusion entre les diverses composantes d'un objet. Ils sont pertinents principalement à l'intérieur d'un objet donné. L'outil traditionnel de la morphologie mathématique pour déterminer ce type de graphe est la segmentation hiérarchique basée sur la ligne de partage des eaux introduite dans [Najman and Schmitt, 1994, Vachier, 1995], développée dans [Meyer, 2001]. et initialement inspirée des travaux de Grimaud [Grimaud, 1990].



La ligne de partage des eaux peut être modifiée pour construire un arbre stockant les fusions des bassins versants. Les régions sont ensuite fusionnées en suivant les branches de l'arbre. L'ordre des fusions peut être modifié pour privilégier certains bassins.

Cette approche «bottom-up» est extrêmement rapide, puisqu'elle s'effectue en une seule passe de ligne de partage des eaux. Cependant, on verra plus bas qu'elle est aussi instable. Nous décrivons dans le chapitre 6 un procédé nouveau pour déterminer des hiérarchies plus robustes, à l'aide d'espaces d'échelles morphologiques.

Caractéristiques

Ces graphes de hiérarchies sont en particulier des arbres. Ce point est important car de nombreux algorithmes coûteux sur des graphes (NP-complets, etc) deviennent plus abordables (polynomiaux P) quand on les applique à des arbres.

Les relations entre les différents niveaux de la hiérarchie sont des relations d'inclusion. Ceci revient à décrire un objet en donnant sa décomposition en sous-objets, récursivement.

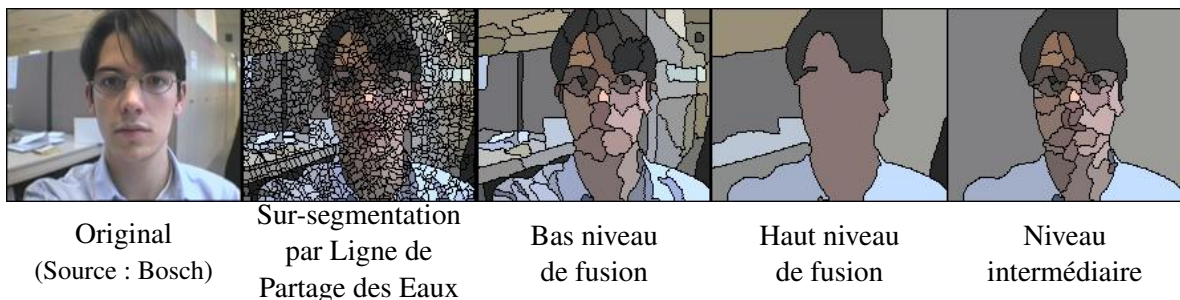


FIG. 2.2: Fusion hiérarchique de régions après une ligne de partage des eaux.

2.1.5 Graphes de voisinage

Construction

Ces graphes encodent les relations entre des objets du même niveau, comme par exemple les différents objets (au sens visuel du terme) d'une image (figure 2.3). On les construit généralement en effectuant une segmentation de l'images en régions suivie par une modélisation par graphe de ces régions et de leurs relations de voisinage.

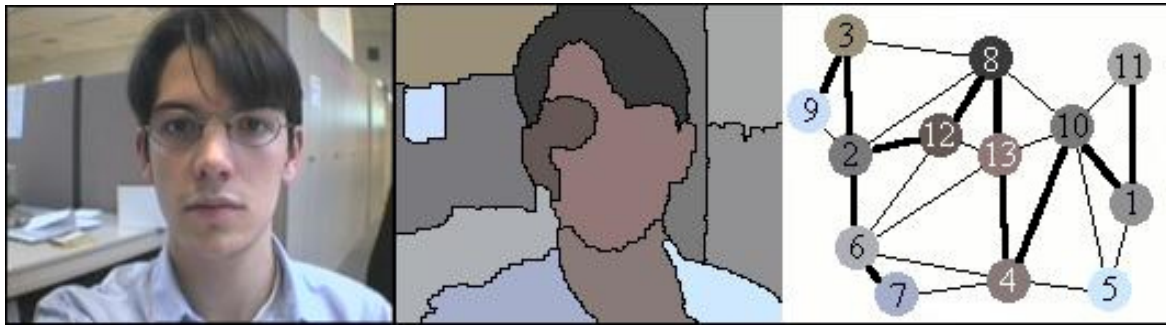


FIG. 2.3: Graphe de voisinage d'une image segmentée en treize régions.

Caractéristiques

Ces graphes ne sont en général pas des arbres, donc les algorithmes simples deviennent rapidement trop lourds à utiliser. Cependant, ces graphes ont idéalement des tailles modérées (quelques dizaines de nœuds) et donc certains algorithmes d'une complexité raisonnable restent utilisables. Il convient néanmoins de faire attention car en réduisant le nombre de régions, on accroît la probabilité de fusionner des régions sans réelle correspondance ce qui fait perdre de la pertinence au modèle.

Par ailleurs, ce sont des graphes auxquels il est facile d'attacher des mesures complexes et de les comparer car tous les nœuds représentent des objets de même niveau, au contraire des arbres précédemment dans lesquels les feuilles et les nœuds proches de la racine n'avaient pas la même signification.

2.2 Le problème de la segmentation

Dans cette section, nous présentons le problème de la segmentation en un nombre limité de régions, ainsi qu'une première solution au problème des segmentations divergentes d'images semblables : l'édition de partitions.

2.2.1 Position du problème

Notre objectif est de comparer des images à travers leurs graphes respectifs. Ces graphes sont des descriptions des résultats des segmentations indépendantes de chaque image. Le problème est que la segmentation est en général un problème mal posé, qui dépend fortement des données initiales. Par exemple, si on choisit de simplifier une image en une «mosaïque», combien de régions (voir figure 2.4) va-t-on considérer ?

Laurent Guigues propose dans sa thèse ([Guigues, 2003], chapitre 7) une méthode intéressante pour choisir un niveau de détail optimal étant donné un compromis entre nombre de régions et homogénéité. Cependant, on n'obtient pas nécessairement le niveau le plus pertinent sémantiquement. Il n'existe en effet pas de méthode automatique réellement optimale, car le résultat souhaité² est fortement dépendant du contenu de l'image. De plus, même en supposant résolu le problème du nombre de régions, on est confronté à l'instabilité de la segmentation appliquée indépendamment à deux images (figure 2.5).

Nous verrons dans cette thèse trois approches pour résoudre ce problème. La première approche consiste à partir d'une segmentation en un nombre faible de régions et d'ajuster les segmentations

²*Souhaité* : on notera la forte connotation subjective de ce terme.

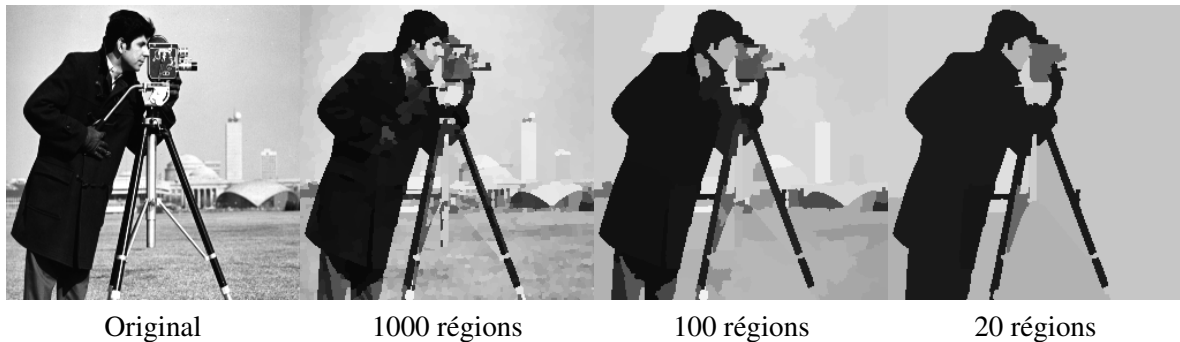


FIG. 2.4: Segmentation hiérarchique : quel est le meilleur nombre de régions ?



FIG. 2.5: Segmentation hiérarchique en 10 régions de deux images d'une même séquence. Dans un cas (gauche) le visage a fusionné avec le fond, dans l'autre cas (droite), c'est le casque qui a fusionné avec le fond.

pour les rendre comparables avant de construire les graphes les représentant. La seconde méthode sera exposée dans la seconde partie du chapitre 3 et utilisera une méthode d'appariement «many-to-many» pour appairer des sur-segmentations. Enfin, la troisième méthode, développée dans le chapitre 7, sera de modéliser les images d'une façon robuste, de stocker ces modèles dans des arbres, et enfin de comparer ces arbres en s'adaptant aux écarts de segmentation.

2.2.2 La segmentation hiérarchique

La segmentation hiérarchique ([Meyer, 2001]) est un outil permettant d'améliorer la segmentation par ligne de partage des eaux de façon rapide et flexible. De par sa rapidité, c'est notre outil de choix pour la modélisation d'images en un nombre restreint de régions. Nous allons dans ce qui suit analyser les causes de son instabilité et rappeler la solution palliative proposée par Cristina Gomila.

Causes de l'instabilité

L'instabilité de la segmentation hiérarchique est due à la conjonction de trois facteurs :

1. **la ligne de partage des eaux n'est pas un outil stable** : La ligne de partage des eaux est une méthode à la fois rapide et efficace pour extraire des contours dans une image. En particulier, ce n'est pas un opérateur local, et une modification de la valeur d'un pixel peut avoir des répercussions dans toute l'image. On se référera à [Beucher, 2004] pour une discussion sur la LPE et ses implémentations les moins biaisées.
2. **la fusion des régions n'est pas un processus stable** : La fusion des régions est effectuée en effectuant une mesure sur toutes les régions (surface, hauteur du col le plus bas, etc), et triant ces

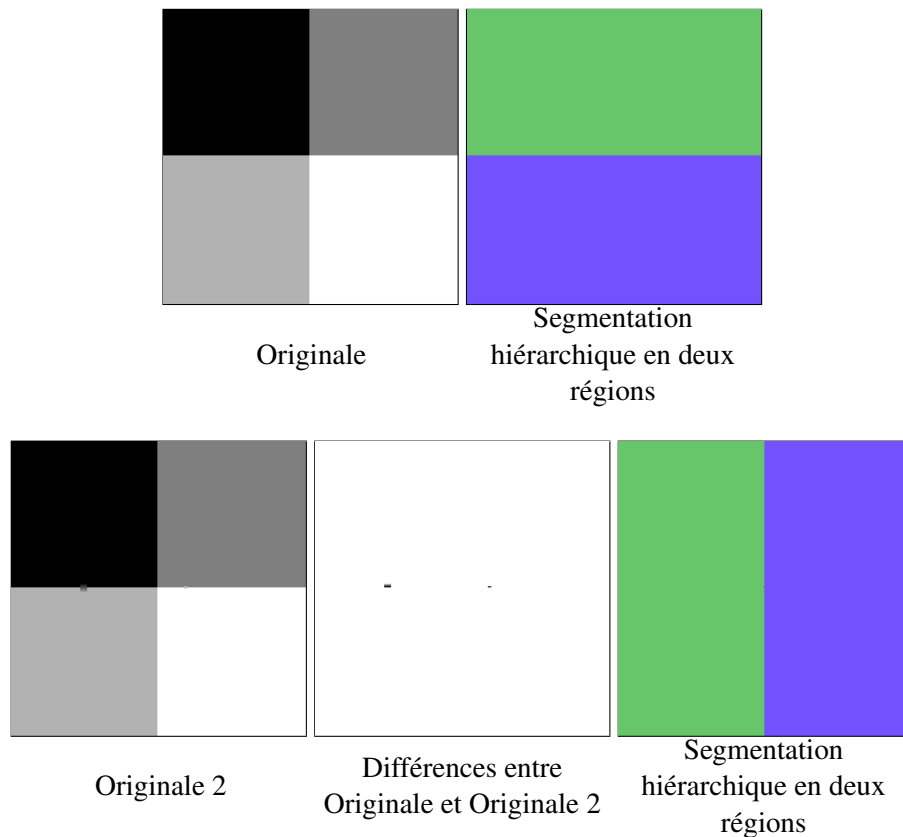


FIG. 2.6: Instabilité naturelle de la segmentation hiérarchique : une infime variation dans l'image segmentée entraîne une modification importante de la segmentation.

mesures par ordre de priorité. Une petite variation dans une des mesures peut modifier l'ordre des fusions et donc l'allure de la segmentation finale.

3. **les ambiguïtés sont résolues de façon purement arbitraire** : Dans les cas d'équivalence du cas précédent, les ambiguïtés sont généralement résolues de façon arbitraire ("premier arrivé, premier servi").

La figure 2.6 illustre le second point, qui est le plus critique.

Édition de partitions

Les deux segmentations de la figure 2.6 ne sont pas appariables en l'état. Les régions que les deux segmentations ont produites ne représentent pas les mêmes réalités.

En descendant au niveau le plus bas, au niveau du pixel, on retrouve l'information initiale qui permettrait un bon appariement. De façon plus réaliste, on va chercher dans les deux hiérarchies le niveau le plus haut dans lequel l'appariement est envisageable.

Cette approche permet donc de retrouver des segmentations comparables et ainsi de rattraper les problèmes causés par une fusion trop enthousiaste. Cependant, on verra plus bas les limites de cette méthode, y compris dans des séquences simples.

À présent que l'on dispose d'une méthode permettant de rapprocher deux segmentations, on va pouvoir les modéliser par des graphes et les comparer.

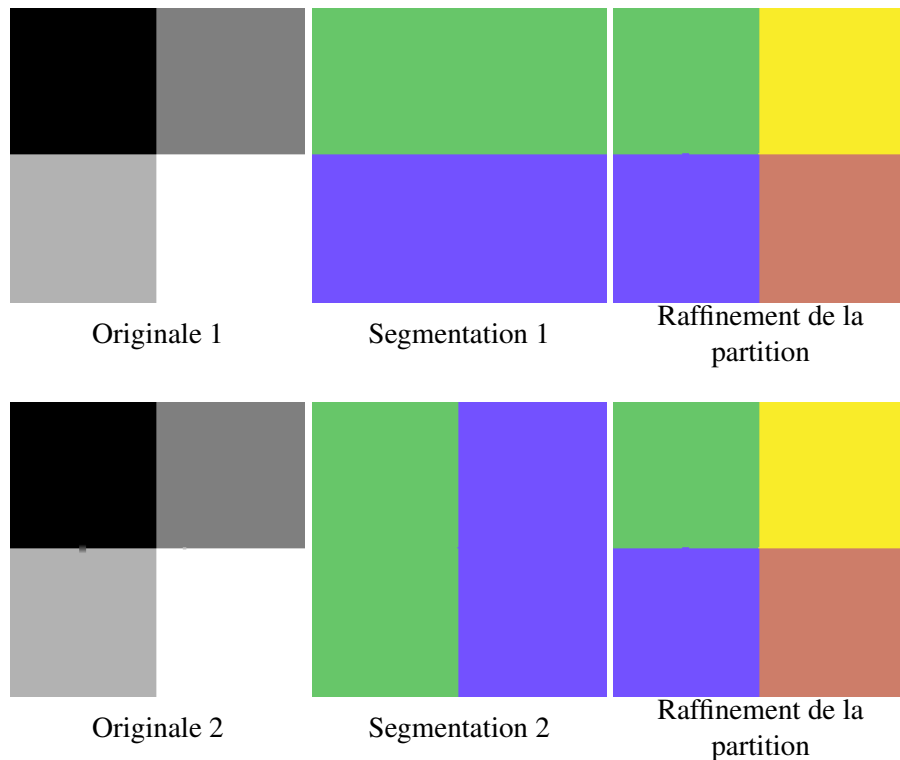


FIG. 2.7: Édition de chaque partition : on «défusionne» les régions problématiques jusqu'à obtenir un résultat comparable

2.3 L'algorithme d'appariement de Cristina Gomila

2.3.1 Présentation de l'algorithme

Cette méthode présentée dans [Gomila, 2001] est basée sur un design de traitement en continu d'un flux vidéo. Pour garder un système simple, les images sont traitées individuellement, et l'objet suivi est propagé successivement sur chaque trame.

Algorithme 1 Suivi d'objets par appariements de graphes

initialisation de l'objet à suivre (manuelle ou automatique)

pour tout image I_{n+1} dans la séquence **faire**

segmentation hiérarchique de I_{n+1} en N régions

édition des segmentations de I_n et I_{n+1}

calcul des graphes de voisinage pour chacune des segmentations

mise en correspondance des régions

propagation des régions marquées «objet» et «fond»

régions disparues stockées en mémoire

régions apparues comparées à celles en mémoire, éventuellement rajoutées à l'objet

fin pour

L'idée forte de cette approche est de ne faire aucune supposition sur les images, et en particulier de ne pas utiliser la cohérence temporelle pour faciliter le travail. Les images sont donc segmentées séparément et sans croisement d'information. Ceci permet en théorie d'étendre l'approche à des couples

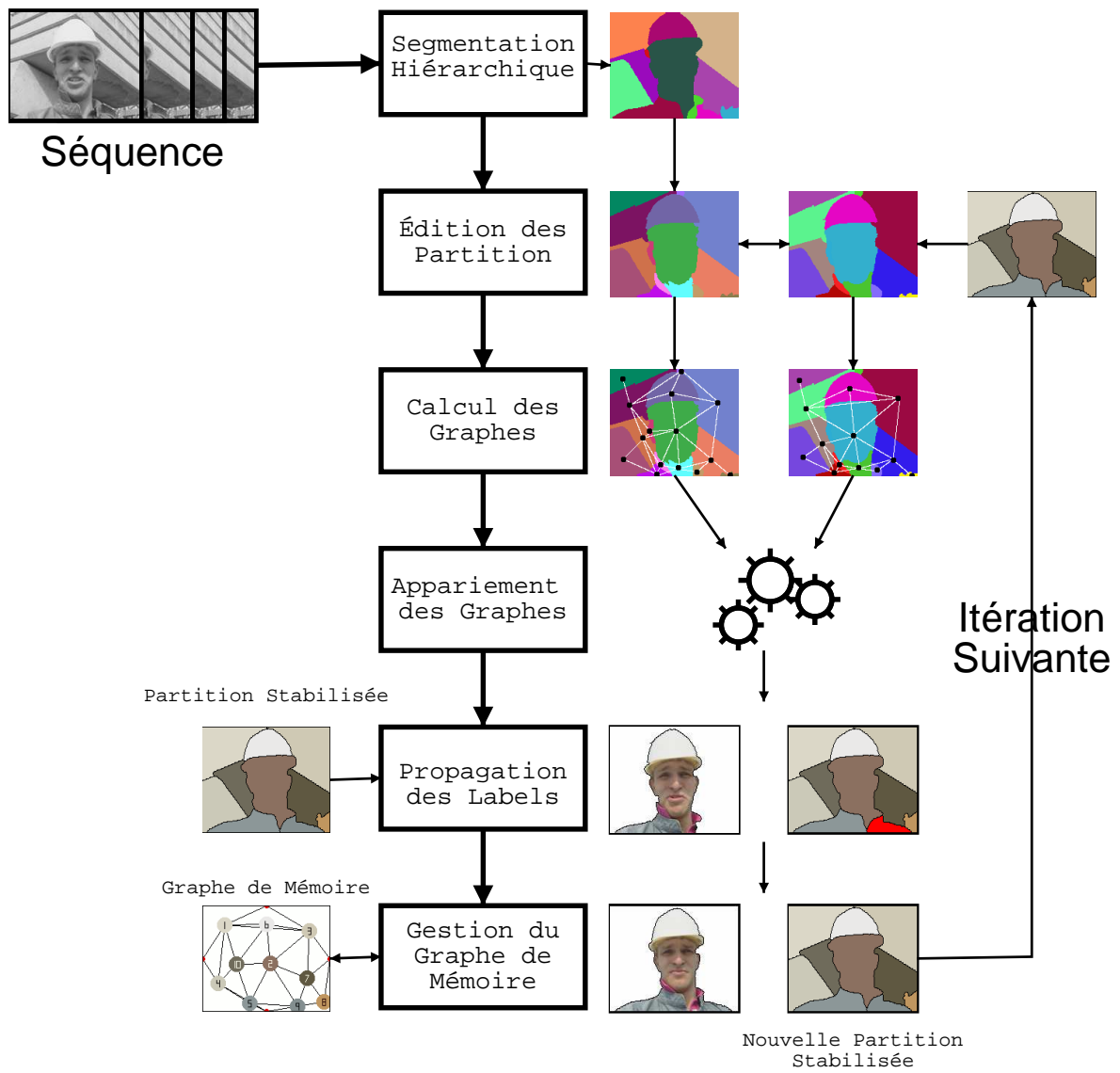


FIG. 2.8: Vue synthétique de l'algorithme de suivi d'objet

d'images ne se suivant pas dans une séquence vidéo, voire à des images non apparentées.

2.3.2 Construction des graphes

La première étape est une segmentation hiérarchique en un faible nombre de régions, suivie d'une édition des partitions raffinant localement chaque segmentation.

Une fois que l'on a obtenu des partitions se ressemblant, on construit sur chacune le graphe de voisinage correspondant : un nœud par région, une arête entre les régions voisines. On value ensuite les nœuds et les arêtes par des mesures de couleur moyenne et de position. On cherche ensuite à mettre en correspondance les nœuds des graphes.

Les nœuds contiennent de l'information sur la région qu'ils représentent : couleur moyenne, taille, position du barycentre. Les arêtes contiennent des informations sur la frontière entre les régions qu'elles relient : différence de couleur moyenne, longueur du contour, etc.

2.3.3 Appariement des graphes

Algorithme

Nos premiers essais ont repris l'algorithme de Cristina Gomila de «relaxation labeling». Par la suite, nous lui avons préféré un algorithme plus performant de «graduated assignment» (voir chapitre 3).

L'algorithme initial de Cristina est une variante sur une relaxation qui inclut une première passe de mise en correspondance directe des nœuds suffisamment proches. Les fondements formels de la relaxation probabiliste utilisée ici ont été présentés par Rosenfeld dans [Kitchen and Rosenfeld, 1979] et élargis par Hummel et Zucker dans [Hummel and Zucker, 1981].

Algorithme 2 Algorithme d'appariement par relaxation de Cristina Gomila

ENTRÉES: Graphes G_1 et G_2

SORTIES: Un appariement «un-à-un» des nœuds de G_1 à ceux de G_2

Similarité des nœuds :

pour tout i, j nœuds de $G_1 \times G_2$ **faire**

Calcul de $P(j \rightarrow i)$ *Probabilité d'apparier les nœuds i et j*

fin pour

Étape de seuillage *Accélère le processus en fixant les probabilités des nœuds déjà très proches*

Similarité des voisinages :

pour tout i, j nœuds de $G_1 \times G_2$ **faire**

Calcul de $P(V_j \rightarrow V_i)$ *Probabilité associée à la conformité des voisinages*

fin pour

Processus récursif de mise à jour :

pour tout i, j nœuds de $G_1 \times G_2$ **faire**

$P^{(n+1)}(j \rightarrow i) = f(P^n(j \rightarrow i), P(V_j \rightarrow V_i))$ *Selon le choix de la fonction f , la compatibilité des voisinages empêchera ou favorisera l'appariement de nœuds plus ou moins directement semblables.*

fin pour

Le seuillage fonctionne en regardant pour tous les nœuds i_0 et j_0 si $\max_j P(i_0, j)$ est atteint pour $j = j_0$ et inversement $\max_i P(i, j_0)$ est atteint pour $i = i_0$, et que la valeur est supérieure à un

seuil. Moins formellement, cela signifie que i_0 préfère être avec j_0 et réciproquement, et que leur compatibilité est forte.

Quand ce cas arrive, on force $P(i_0, j_0)$ à 1 et les autres probabilités $P(i_0, \cdot)$ et $P(\cdot, j_0)$ sont mises à zéro. L'effet bénéfique est d'accélérer la convergence de la relaxation. En revanche, cela suppose qu'il n'y aurait pas eu de renversement de situation. Selon les images et la valeur du seuil, nous avons constaté les deux comportements.

Distances

Les distances utilisées mêlent l'information couleur et la géométrie des objets. Cristina en combine trois :

- Une distance couleur dans YUV (luminance et chrominance traitées séparément)
- Une comparaison de la forme des régions (aire de l'intersection sur l'aire de l'union)
- Une distance entre les barycentres des régions (rapportée aux dimensions de l'image)

2.3.4 Analyse des résultats

Dans l'optique de réutiliser les algorithmes de Cristina Gomila, nous avons cherché à mieux comprendre leur comportement dans le cadre de la comparaison d'images, en particulier dans des images légèrement plus complexes que celles qu'elle avait utilisées dans sa thèse. La figure 2.9 présente une analyse des résultats donnés par le système complet sur une paire d'images typiques des flux vidéos télévisés.

On voit que la grande majorité des régions est correctement mise en correspondance (en vert). En revanche, les zones situées autour du bandeau incrusté posent plusieurs problèmes. Tout d'abord, pour certaines régions (en bleu), apparemment simples à apparier, l'algorithme n'a pas trouvé de solution. L'algorithme, perturbé par le bandeau n'apparaissant que dans une image, n'a pas réussi à obtenir un score suffisant pour considérer un appariement ferme. Par ailleurs, d'autres zones (en rouge) ont été assignées aux mauvaises régions : on a apparié la zone droite du bandeau (en rouge) à l'épaule (en rouge) au lieu de la région correspondante dans la veste (en bleu). Les régions jaunes sont celles pour lesquelles l'appariement est correct, mais où les contours correspondent mal.

Enfin, il est intéressant de voir que l'étape d'édition de partitions a détecté dans la première image une zone à gauche correspondant apparemment au bandeau (là où il est affiché «Odile» et «Présidente»), alors qu'à première vue il n'y a aucun contour dans le fond. En effet, le fond est en réalité légèrement texturé et n'est pas segmenté en une seule région. L'édition de partition fait donc apparaître de façon complètement artificielle une région qui est appariée avec succès à un objet complètement nouveau.

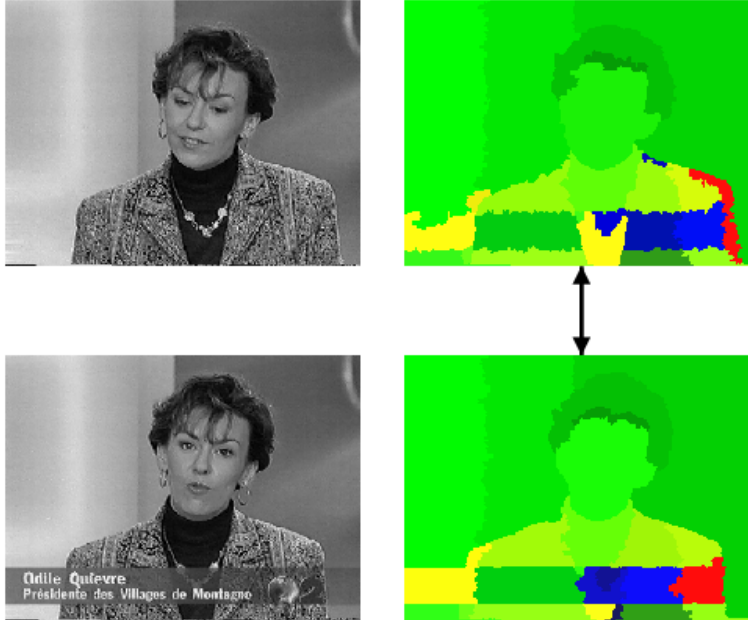
Cette première analyse nous montre que nous devons être prudent dans l'extension de cet algorithme à d'autres séquences ou paires d'images.

2.4 Le projet MASCOT

2.4.1 Motivation du projet

Le projet MASCOT (Metadata for Advanced Scalable Video Coding Tools) était un projet européen (IST-2000-26467) visant à développer l'utilisation de «métadonnées» pour améliorer la compression vidéo. Ces métadonnées pouvaient couvrir un champ extrêmement large dont seule une partie a été standardisée par MPEG-7.

Images Originales

Segmentation
et Appariement

Vert: appariement réussi et jugé correct
Jaune: appariement réussi malgré une segmentation douteuse
Bleu: appariement échoué
Rouge: appariement réussi mais jugé incorrect

FIG. 2.9: Analyse du résultat d'un appariement entre deux images d'une séquence vidéo

Depuis 1998, le standard MPEG-4 [Pictures and Audio, 2002] permet d'encoder des flux vidéo de façon extrêmement puissante. Les concepts de plan objets, en particulier, permettent en théorie d'obtenir des qualités de compression inégalées en permettant de traiter séparément les objets qui se recouvrent et donc de limiter la perte de qualité sur les frontières des champs de mouvement.

Malheureusement, ces technologies n'étaient utilisées que dans les démonstrations et peinaient à gagner l'acceptation de la communauté de compression vidéo. La raison principale à ceci était que la majeure partie des images compressées n'étaient pas accompagnées de métadonnées, les objets n'étaient pas répartis sur plusieurs plans, seule l'information brute était disponible. Il fallait donc d'une part développer les outils qui permettent d'analyser l'image et de créer des métadonnées et d'autre part déterminer les scénarios qui justifiaient l'utilisation de ces métadonnées en termes de gains soit en fonctionnalités, soit en performances.

Le rôle du CMM était d'apporter ses compétences en termes d'analyse d'image et de segmentation pour déterminer des métadonnées associées aux objets détectés dans l'image et suivis au cours du temps via nos mécanismes d'appariements de graphes. Dans ce qui suit nous verrons deux approches

utilisant de la segmentation et des graphes pour tenter d'améliorer les taux de compression.

2.4.2 Suivi du locuteur, recollage de plans

Motivation

Dans la plupart des schémas de compression vidéo, il existe deux types d'images³ : celles qui sont transmises intégralement, et celles qui sont interpolées. Les images intégrales sont coûteuses à transmettre, alors que dans le cas des images interpolées, on ne transmet que l'image des différences qui se compresse bien. Dans une séquence relativement peu agitée, on trouve ainsi une majorité d'images interpolées entre deux images transmises intégralement.

Certaines scènes sont construites en entremêlant deux plans relativement fixes, par exemple une interview télévisée : l'invité et le présentateur alternent dans le champ de la caméra. On paie donc régulièrement le prix d'une image entière. Dans ce cas, il serait intéressant de compresser séparément les deux vues afin de bénéficier au maximum de la continuité des images. Charge au décompresseur de réorganiser les images au moment de l'affichage.

Pour ce faire, il faut principalement repérer les changements de plans. Plusieurs méthodes sont possibles, allant de très simple à plus complexe. Une première approche serait de comparer directement les images et de décider qu'il y a eu une transition lorsque la distance globale dépasse un seuil prédéterminé. Cependant, cette méthode est risquée car il est possible que la vue change sans pour autant qu'une majorité des pixels soit affectés. Par exemple, sur un plateau de tournage il est fréquent que le fond soit le même derrière chaque locuteur.

Méthode

Pour avoir une granularité plus fine dans la détection, nous avons employé notre système de suivi d'objet, mais en l'utilisant pour comparer des images plutôt que pour suivre des objets (voir figure 2.11). La modification est simple, puisqu'elle consiste à demander à l'algorithme de mise en correspondance une mesure de la qualité de l'appariement. Cette mesure peut être adaptée pour ne pas tenir compte du fond, et permet en général de mieux gérer les déplacements des personnes, les mouvements de bras, etc.

Résultats et perspectives

La figure 2.12 présente les résultats de la comparaison d'images. On voit que les bonnes associations ont toujours un meilleur score que les associations mixtes (présentateur/invitée ou invitée/plan large, etc), et ce malgré les modifications importantes de l'image (mouvement des bras, bandeau informatif, etc).

Ce système fonctionne donc bien pour retrouver des scènes similaires dans un flux vidéo. Par ailleurs, on a vu dans la figure 2.9 que l'algorithme était capable d'identifier la plupart des éléments présents dans les deux images. On verra dans le chapitre suivant une modification de l'algorithme qui améliore ses performances et sa capacité à correctement appairer les régions même en présence de bruit.

En ayant une vision plus large, cet algorithme pourrait servir également de base à un outil permettant la structuration puis l'indexation semi-automatique de séquences vidéo, dans la lignée des

³Les experts savent qu'on considère en général *trois* types d'image : les images «intra» I, les images «prédites» P utilisant le champ de mouvement, et les images interpolées B. Pour la clarté du propos, on ne fera pas de distinction entre les trames P et B.



FIG. 2.10: Séquence typique d'un entretien télévisé. On voit trois types de plans : le plan large, le présentateur et l'invitée. On cherchera à regrouper les plans de la même personne en les séparant de ceux de l'autre personne et des plans larges.

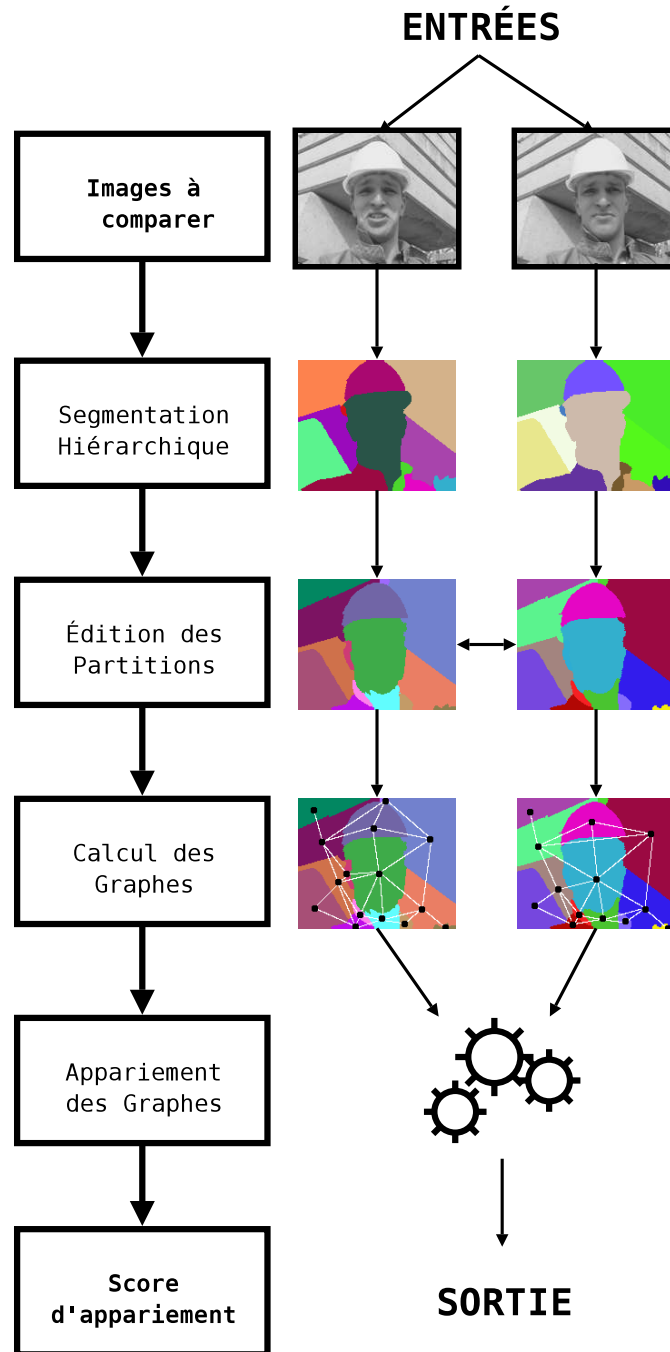


FIG. 2.11: Algorithme simplifié pour la comparaison d'images

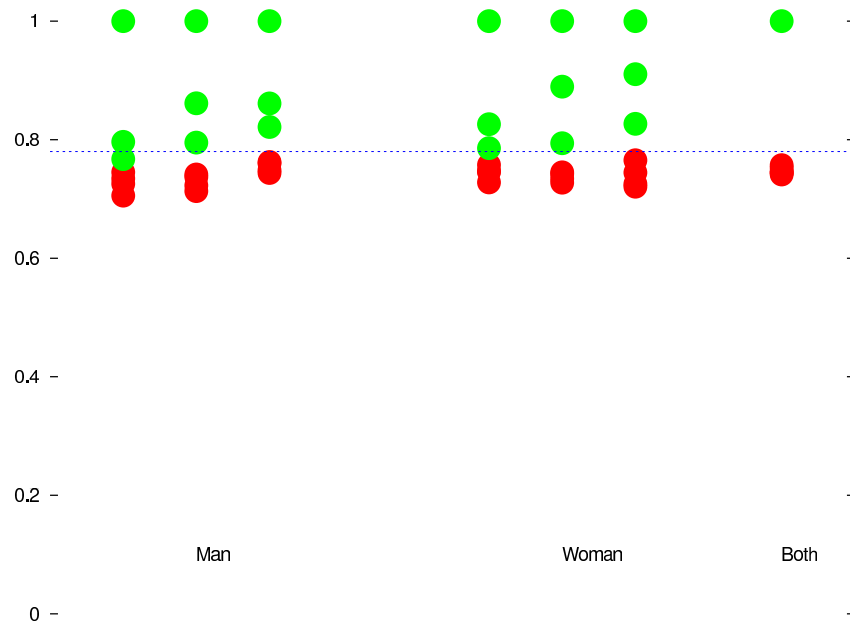


FIG. 2.12: Association de plans : dans chaque cas, les points verts sont les plans de la bonne personne, les points rouges sont les autres plans (plan large ou autre personne).

travaux de Claire-Hélène Demarty [Demarty, 2000]. En particulier, la segmentation offre de grandes possibilités pour la gestion des modifications dans l'image. Doit-on s'arrêter sur l'incrustation d'une vidéo dans un coin de l'image ? Ou un bandeau d'information le long du bas de l'écran ? Une grande partie de l'analyse de l'image pourrait être factorisée dans une étape de segmentation et de création d'un graphe suffisamment informatif mais qui serait infiniment plus léger que l'image d'origine.

2.4.3 Pré-interpolation

Outre les images transmises intégralement (trames I), une grande partie du coût de compression (en termes de bande passante) est le calcul et la compression du champ de mouvement entre deux images (trames P). Il existe globalement trois types de mouvements :

1. traveling latéral de la caméra
2. caméra fixe, déplacement d'un objet
3. traveling avant ou arrière de la caméra (zoom)

Dans le premier cas, on est en présence d'un champ de déplacement globalement uniforme. Dans le second cas, le champ est uniforme par objet, ou tout au moins raisonnablement uniforme par composante de l'objet en mouvement. Le troisième cas n'est pas modélisable globalement pas un modèle de déplacement uniforme, mais la plupart des objets de la périphérie auront un mouvement pseudo-uniforme.

Ceci nous incite à utiliser notre connaissance de la structure de l'image (par segmentation) pour essayer d'améliorer les méthodes de calcul de champ de mouvement en calculant un modèle de mouvement pour chaque région. Ce modèle pouvant être uniforme, linéaire, ou tout autre méthode, par exemple les modélisations par krigeage inverse de Decencièrè, de Fouquet et Meyer [Decencièrè et al., 1999]

reprises par Lothar Bergen dans sa thèse [Bergen, 1999]. En pratique, nous avons utilisé un champ de mouvement uniforme par région, car dans nos essais les champs plus complexes étaient quasi-uniformes et n'apportaient ainsi que de la complexité aux résultats.

Le codeur de référence disposait déjà de son mécanisme calculant les champs de mouvement et celui-ci n'étant pas facilement modifiable, nous avons donc dû dans une première étape compenser en mouvement nos images suivant le champ de mouvement grossier estimé grâce aux graphes, avant de les fournir au codeur.

Malheureusement la figure 2.13 montre que cette approche se traduisait par des pertes en qualité de compression. Pour expliquer ce phénomène, il faut bien comprendre qu'en compression vidéo, on cherche à minimiser le coût de transmission d'un champ de mouvement et d'une erreur commise. Il se peut que le meilleur résultat soit obtenu pour un champ faux mais facile à coder et une erreur raisonnable. En cherchant à optimiser le champ de mouvement séparément de l'erreur, nous nous sommes écartés de l'optimalité pour les contraintes données.

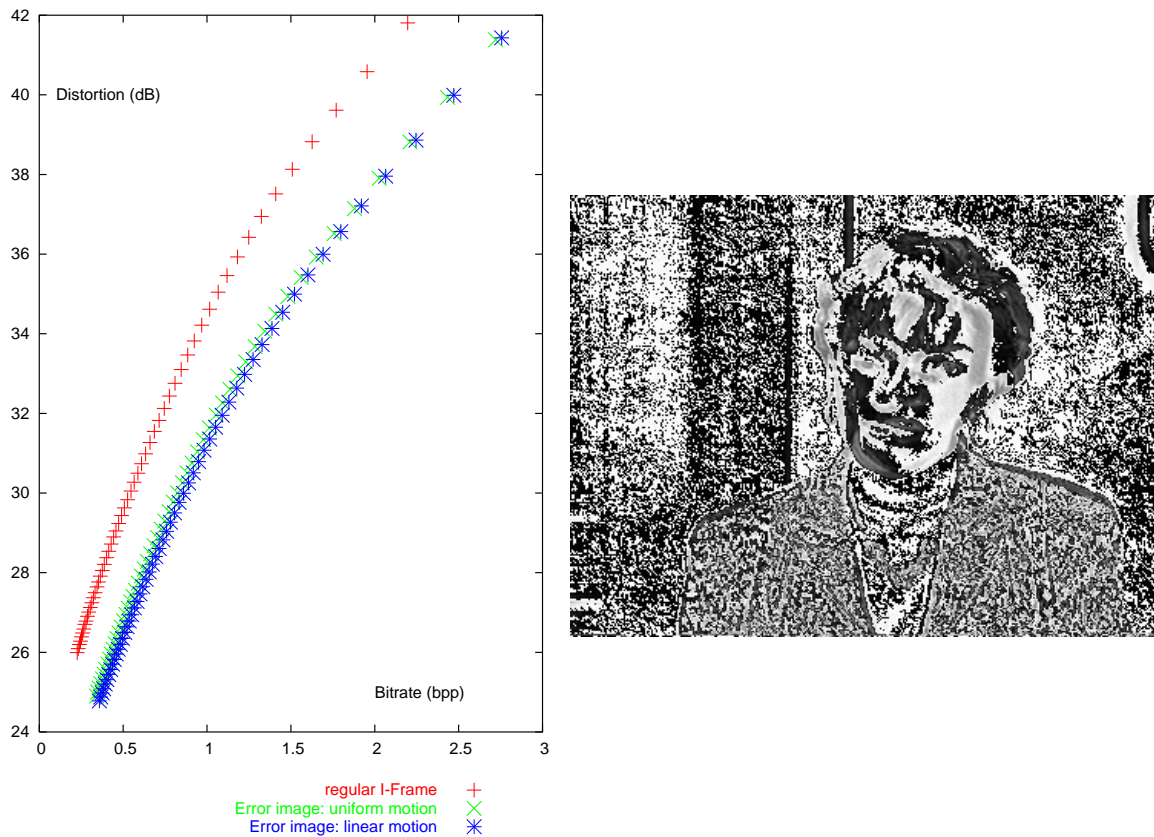


FIG. 2.13: Mesures débit/distorsion à la sortie du codeur vidéo (gauche). Les résultats sont nettement plus mauvais avec une étape de pré-interpolation. Ceci s'explique par le fort coût de compression de l'image des différences (droite).

Cette approche a donc été écartée. Il est possible qu'un couplage plus proche avec le block matching aurait donné de meilleurs résultats, malheureusement la structure du codeur ne permettait pas une telle liberté. Les codeurs vidéos utilisés réellement étant actuellement monoplans, il semble ainsi qu'ils soient difficilement compatibles avec une approche visant à modéliser l'image et son mouvement en plan objets distincts. Cette approche a donc été mise en sommeil en attendant une évolution des méthodes de codage des flux vidéos permettant de tirer pleinement parti de nos outils d'analyse.

2.5 Premières conclusions et perspectives

La modélisation par graphe présente des perspectives intéressantes en même temps que des défis à relever. Le grand attrait de cette approche réside dans sa capacité à utiliser des techniques de segmentation pour synthétiser l'information présente dans une image, ce qui permet de traiter de façon plus simple cette information.

Cette simplicité peut ensuite se décliner selon deux axes : soit un traitement automatique plus rapide, soit une intervention manuelle simplifiée. Le projet VOGUE [Marcotegui et al., 1999] par exemple montre l'utilisation de la segmentation hiérarchiques pour assister la segmentation supervisée.

Ces outils de manipulation de graphes étant encore neufs, nous allons devoir d'une part les mettre à l'épreuve sur des problèmes plus complexes et d'autre part déterminer les zones pouvant être améliorées de façon incrémentale. Les points de faiblesse principaux que nous avons identifiés sont le mécanisme d'appariement d'une part et l'édition de partitions d'autre part.

Nous verrons dans le chapitre qui suit les extensions que nous avons développées et nous présenterons également les limites que nous avons rencontrées dans l'application de cette approche à des problèmes plus complexes.

Chapitre 3

Amélioration de l'appariement de graphes

*Notre raison est toujours déçue par
l'inconstance des apparences.*

Blaise Pascal, *Pensées*

Ce chapitre sera consacré aux améliorations des techniques d'appariement décrites dans le chapitre précédent. Nous y exposerons d'abord l'algorithme de «graduated assignment» qui se substituera à la relaxation probabiliste au cœur du moteur d'appariement. Puis nous présenterons une approche différente utilisant une projection du graphe dans l'espace spectral. Nous terminerons en analysant les difficultés que nous avons rencontrées dans le cadre de cette approche, et nous proposerons deux nouveaux axes de recherche.

3.1 Introduction

Partant des outils de construction de graphes développés par C. Gomila, nous allons chercher des algorithmes d'appariement de graphes plus performants.

Le problème de l'appariement de graphes issus de segmentations est soit raisonnablement simple, soit extraordinairement complexe, selon la difficulté des images et la qualité de la modélisation. En effet, il est souvent difficile de déterminer si une erreur est due à une faute de l'algorithme de comparaison, ou à une modélisation peu fidèle. De plus, il est important de noter que notre problème est plus complexe que le calcul d'*isomorphismes* entre les graphes. Dans notre cas, un objet peut se voir séparé en deux objets, soit à cause d'une erreur de segmentation, soit parce que l'image a réellement changé. On peut également voir des éléments de l'image apparaître ou disparaître selon l'évolution de la séquence vidéo.

Dans une certaine mesure, on pourrait se contenter de chercher des sous-graphes isomorphes maximaux, mais alors il faudrait avoir une confiance absolue dans la modélisation : si un objet est séparé en deux régions dans l'un des graphes, il ne sera pas apparié. C'est donc une approche «floue» qui sied le mieux à nos besoins. Par certains aspects, cela facilite les choses car on a une plus grande

liberté dans la recherche des solutions, mais en contrepartie il manque parfois un cadre plus rigide sur lequel s'appuyer.

Dans ce qui suit, nous étudierons l'état de l'art avant de choisir et d'approfondir deux méthodes d'appariement. La première, «*graduated assignment*», se substitue simplement à la méthode de relaxation présentée dans le chapitre précédent, et la seconde utilise une décomposition spectrale pour comparer les graphes dans un sous-espace de vecteurs propres.

3.2 Approches «one-to-one»

3.2.1 État de l'art

La comparaison d'objets par l'analyse de leurs graphes est un sujet d'intérêt depuis quelques années dans la communauté de la théorie des graphes. Diverses approches ont été testées, se focalisant principalement sur l'aspect appariement du problème. On peut dégager globalement deux écoles : celle de l'appariement «un-à-un» («one-to-one») et celle plus récente de l'appariement «beaucoup-à-beaucoup» («many-to-many»). Les méthodes de la première sont en général plus simples mais nécessitent des structures très proches. Celles de la seconde sont plus complexes mais permettent d'apparier des graphes moins semblables.

Les approches entrant dans la première catégorie peuvent elles-mêmes être classées grossièrement en deux ensembles : les approches combinatoires d'une part, et les approches numériques d'autre part. Les approches combinatoires utilisent des techniques classiques de théorie des graphes, comme la recherche de couplages par chemins augmentants, les distances d'édition ou les cliques. Les approches numériques visent à minimiser le coût d'une fonctionnelle dépendant de l'appariement en plongeant la recherche des solutions dans un espace continu. Nous détaillerons ci-dessous quelques méthodes combinatoires avant de décrire plus en détail une méthode numérique plus compatible avec notre approche.

Le couplage consiste à modéliser l'appariement entre les nœuds comme un graphe biparti pondéré par les distances entre couples de nœuds et à chercher le couplage de coût minimum via des algorithmes classiques de théorie des graphes (algorithme «hongrois» [Kuhn, 1955], ou plus récemment [Goldberg and Kennedy, 1995]). L'approche par relaxation dont nous disposons est une variante de ce problème. Shokoufandeh [Shokoufandeh et al., 1999] propose une approche en espace d'échelle utilisant les ondelettes pour calculer une représentation arborescente de chaque objet. Celles-ci seront ensuite comparées en cherchant le couplage de coût minimal. Cette représentation est une des sources d'inspiration de la représentation arborescente décrite dans le chapitre 6.

Une autre représentation est celle des «*shock graphs*», qui sont des sortes de squelettes. Kimia et Sebastian [Kimia et al., 2001] l'utilisent pour modéliser des objets et les comparer par distance d'édition. La distance d'édition est la distance calculée en comptant le nombre d'opérations nécessaires pour transformer un objet en un autre à partir de trois opérations : insertion, suppression et substitution. C'est une distance très souvent utilisée pour comparer des chaînes de caractères. Or, un arbre peut être codé sous forme d'une chaîne en utilisant un parcours eulérien ([Gondran and Minoux, 1985], chapitre 8). Cependant, ce problème devient plus complexe lorsqu'il n'y a pas d'ordre naturel des branches. Par ailleurs, trouver et comparer des chemins dans des graphes qui ne sont pas des arbres (voire simplement dans des arbres sans racine) est un problème nettement plus complexe.

Enfin, Pelillo propose une approche par cliques¹ dans [Pelillo et al., 1999, Pelillo, 2002] qui est basée sur la constatation que la recherche d'un isomorphisme maximal entre deux graphes revient à

¹Une clique est un ensemble de nœuds d'un graphes tous reliés deux-à-deux par des arêtes.

chercher une clique maximale dans le graphe produit. La faiblesse de cette approche est qu'il est plus difficile d'inclure une notion de distance entre les nœuds. Toute la richesse de l'information doit donc être contenue dans la topologie du graphe représentant l'objet.

La plupart de ces approches supposent le problème de la segmentation résolu et concentrent leur robustesse sur la gestion des occultations. Malheureusement, cette confiance aveugle n'est pas toujours méritée car la segmentation est un processus instable et mal posé. Combien de régions choisir ? Comment les choisir ? Comment rattraper un mauvais choix dans l'une des segmentations ? Il faut donc écarter tous les algorithmes cherchant des isomorphismes trop stricts et se concentrer sur ceux capables d'absorber des erreurs situées en amont.

3.2.2 Approche numérique : graduated assignment

Un certain nombre des approches mentionnées ci-dessus nécessitent une modélisation différente de celle dont nous partons : shock graphs, arbres dans un espace d'échelle, etc.

Dans un premier temps, nous avons préféré une amélioration simple dans laquelle il suffise de remplacer le cœur du moteur de l'appariement, sans modifier le reste de l'algorithme (segmentation, édition des partitions, etc). L'objectif est donc de trouver un algorithme plus robuste face au bruit que la relaxation probabiliste mais qui fonctionnerait sur le même type de graphe. Nous avons trouvé un bon candidat avec l'algorithme de «graduated assignment» de Gold et Rangarajan [Gold and Rangarajan, 1996]. Cet algorithme plus performant nous a permis d'obtenir de meilleurs résultats (à la fois plus rapides et plus robustes) pour un faible coût d'intégration.

Description

Étant donné deux graphes G^1 et G^2 , cette approche vise à minimiser une fonctionnelle codant les similarités entre nœuds et entre arêtes en fonction d'une matrice M codant l'appariement entre les nœuds de G^1 et ceux de G^2 . Cette matrice est définie par $M_{ij} = 1$ ssi le i -ème nœud de G^1 est apparié au j -ème nœud j de G^2 , et zéro partout ailleurs.

On stockera les similarités $s_{\text{nœuds}}$ entre les nœuds N_i^1 du graphe G^1 et les nœuds N_j^2 du graphe G^2 dans une matrice C^N définie par :

$$C_{ij}^N = s_{\text{nœuds}}(N_i^1, N_j^2)$$

Pour les similarités $s_{\text{arêtes}}$ entre arêtes, on utilisera une matrice de dimension 4 notée C^A et définie par :

$$C_{ijkl}^A = \begin{cases} 0 & \text{si } A_{i,j}^1 \text{ n'est pas une arête de } G^1 \text{ ou} \\ & A_{k,l}^2 \text{ n'est pas une arête de } G^2 \\ s_{\text{arêtes}}(A_{i,j}^1, A_{k,l}^2) & \text{sinon} \end{cases}$$

La fonctionnelle est en général du type :

$$E(M) = -\frac{1}{2} \underbrace{\sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N M_{ik} M_{jl} C_{ijkl}^A}_{\text{terme mesurant la compatibilité des voisinages}} + \alpha \underbrace{\sum_{i=1}^N \sum_{j=1}^N M_{ij} C_{ij}^N}_{\text{terme mesurant la compatibilité des nœuds}} \quad (3.1)$$

Où α est un paramètre déterminant l'attachement soit aux nœuds, soit aux arêtes.

Résolution

M est une matrice de permutation. Une caractérisation de ces matrices est qu'il y ait exactement un seul 1 par ligne et par colonne et des zéros partout ailleurs. Ceci peut s'écrire de façon plus numérique comme :

$$\begin{cases} \forall i \in [1; n], \sum_{j=0}^n M_{i,j} = 1 \\ \forall j \in [1; n], \sum_{i=0}^n M_{i,j} = 1 \\ \forall i, j \in [1; n]^2, M_{i,j} \in \{0, 1\} \end{cases}$$

On peut relâcher la dernière contrainte et prendre les valeurs de M dans \mathbb{N} en constatant que les sommes forcent les valeurs (entières et positives) à être en réalité dans $\{0, 1\}$:

$$\begin{cases} \forall i \in [1; n], \sum_{j=0}^n M_{i,j} = 1 \\ \forall j \in [1; n], \sum_{i=0}^n M_{i,j} = 1 \\ \forall i, j \in [1; n]^2, M_{i,j} \in \mathbb{N} \quad \leftarrow \text{contrainte relâchée} \end{cases}$$

La méthode de résolution proposée plonge la matrice M dans un espace plus large, continu, et fait progressivement converger M vers une matrice aux valeurs binaires 0 ou 1. L'espace continu choisi est celui des matrices doublement stochastiques, qui représente *grossièrement* l'extension des matrices de permutations aux valeurs réelles :

$$\begin{cases} \forall i \in [1; n], \sum_{j=0}^n M_{i,j} = 1 \\ \forall j \in [1; n], \sum_{i=0}^n M_{i,j} = 1 \\ \forall i, j \in [1; n]^2, M_{i,j} \in \mathbb{R}^+ \quad \leftarrow \text{espace continu} \end{cases}$$

Maintenant que les valeurs de M sont dans un espace continu, on va pouvoir utiliser des méthodes classiques d'optimisation numérique pour faire converger M vers le résultat minimisant le coût .

Le principe de l'algorithme de Gold et Rangarajan est de constater qu'en faisant un développement limité au premier ordre de E , on se ramenait à un problème de couplage. Ainsi, étant donné une solution admissible M_0 , on peut trouver un ensemble de constantes Q telles que minimiser E au voisinage de M_0 revienne à maximiser :

$$\sum_{i=1}^n \sum_{j=1}^n M_{i,j} Q_{i,j}$$

Qui est l'expression numérique d'un problème de couplage.

On va donc ainsi en partant d'une solution admissible approcher la solution exacte en résolvant des problèmes de couplages et en réinjectant le nouveau M à la place de M_0 dans la formulation de Taylor.

Au cours des itérations, on va chercher à maintenir l'invariant que M est une matrice doublement stochastique. De plus, on va faire converger cette matrice vers une matrice doublement stochastique aux valeurs entières, donc une matrice de permutation.

La convergence vers une matrice doublement stochastique résolvant le problème de couplage s'effectue en itérant des normalisations de type «softmax» [Bridle, 1990]. Étant donné un ensemble de valeurs $\{X_i\} \in \mathbb{R}^n$, cette méthode permet de trouver la valeur maximale en utilisant des variables m_i telles que $\sum_i m_i = 1$, en faisant converger les m_i vers 1 si X_i est la valeur maximale et 0 sinon. La méthode softmax itère avec un β croissant :

$$m_i = e^{\beta X_i}, \forall i \text{ suivi de :}$$

$$m_i = \frac{m_i}{\sum_j m_j}, \forall i$$

L'exponentielle garantit des valeurs positives et à la limite quand β grandit le m_i correspondant au maximum tend vers 1 et les autres vers 0 (à l'exception des cas d'égalité).

En itérant cette normalisation sur les lignes et les colonnes d'une matrice M , il a été montré [Sinkhorn, 1964] que cette matrice convergeait vers une matrice stochastique. Cette méthode est appelée «softassign» car elle résout presque² un problème de couplage («assignment problem» en anglais).

Algorithme 3 Algorithme d'appariement de Gold et Rangarajan

ENTRÉES: Fonctionnelle des coûts d'appariements $E(M)$ et valeurs initiales β_0 et M_0

SORTIES: Une matrice de permutation M

Initialisation de $\beta = \beta_0$ et $M = M_0$

tantque $\beta \leq \beta_{final}$ **faire**

répéter

Développement de Taylor :

$$Q_{i,j} = -\frac{\partial E}{\partial M_{i,j}}$$

Résolution du couplage par softassign :

$$M_{i,j} = \exp(\beta Q_{i,j})$$

répéter

Normalisation sur les rangées :

$$M_{i,j} = \frac{M_{i,j}}{\sum_k M_{k,j}}$$

Normalisation sur les colonnes :

$$M_{i,j} = \frac{M_{i,j}}{\sum_k M_{i,k}}$$

jusqu'à convergence de M

M est maintenant une matrice doublement stochastique

jusqu'à convergence de M

 Incrémenter β

M se rapproche d'une matrice de permutation

fin tantque

Post-processing de M

Une étape de post-processing est nécessaire dans le cas où on ne converge pas tout à fait vers une matrice de permutation. Elle consiste à mettre à la main le maximum de chaque colonne (ou ligne, c'est équivalent) à 1 et les autres à 0.

La formulation complète de l'algorithme inclut des paramètres de lissage et des variables de souplesse («slack variables») permettant de s'affranchir d'un certain niveau de bruit. Par souci de clarté nous n'en parlons pas ici, mais le lecteur intéressé se référera à l'article original [Gold and Rangarajan, 1996].

Critique

Cette approche est à la fois plus robuste et plus rapide que la relaxation probabiliste, ce qui nous a permis de l'utiliser sur des images posant auparavant quelques difficultés. L'article original montre des

²Presque, car on n'a pas encore convergé vers des valeurs entières

comparaisons entre les deux méthodes et assure la supériorité de leur méthode en termes de robustesse aussi bien que de rapidité. Le bruit utilisé est, hélas comme souvent dans les articles théoriques, choisi de façon aléatoire (probabilité d'ajout ou de suppression de nœud ou d'arête, etc). Heureusement, nos tests ont confirmé l'accroissement de la robustesse dans des cas pathologiques plus réalistes.

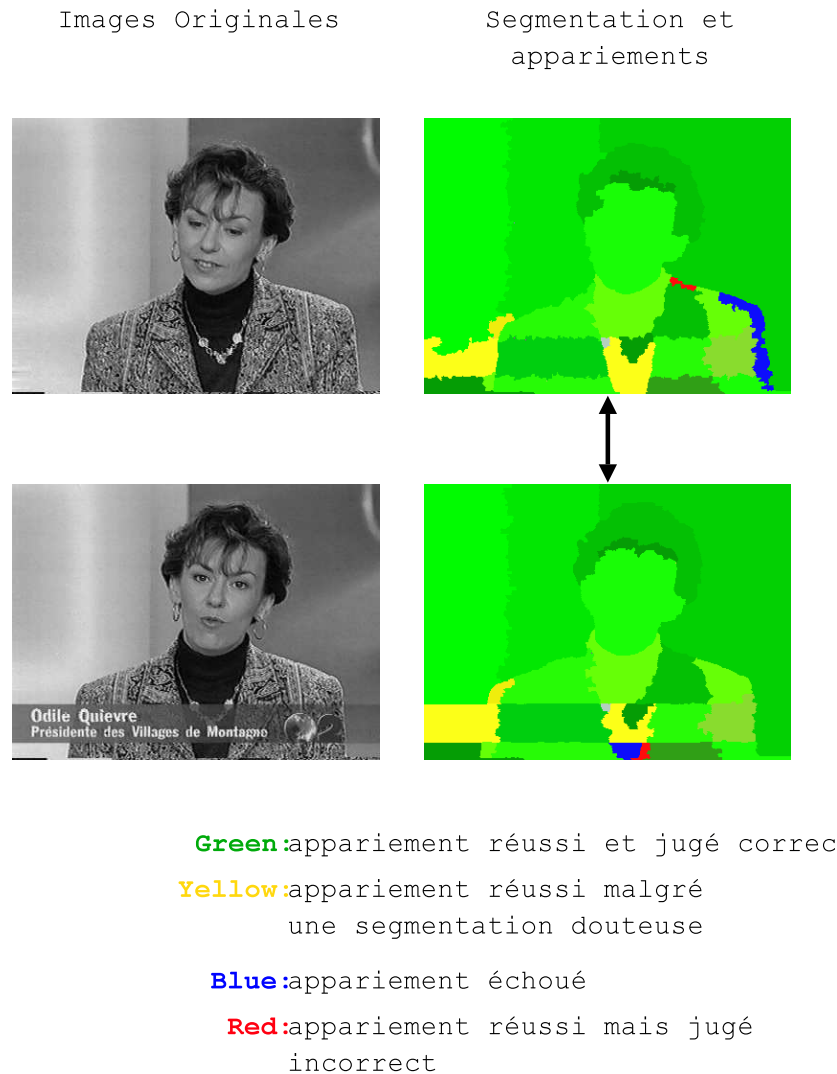


FIG. 3.1: Analyse du résultat d'un appariement entre deux images d'une séquence vidéo, à l'aide de l'algorithme «graduated assignment» pour la mise en correspondance des graphes.

Les résultats présentés dans la figure 3.1 sont à comparer aux précédents de la figure 2.9 (p. 20). On voit que mis à part deux petites zones mal détectées (en rouge), l'algorithme s'est bien comporté. En particulier, il a bien remis en correspondance les zones du bandeau incrusté qui avaient posé des problèmes à la version précédente. De plus, la région du collier a elle aussi été mieux appariée. Les zones problématiques dues aux aléas de la segmentation n'ont bien sûr pas été corrigées, mais dans l'ensemble les résultats paraissent satisfaisants dans ce cas d'utilisation.

Cependant, après avoir renforcé le maillon le plus faible de la chaîne, de nouvelles limitations sont apparues dès que nous l'avons employée sur des cas plus complexes. En effet, lorsque les images sont trop distantes, il est difficile de rapprocher les segmentations suffisamment pour que des algorithmes

d'appariement «un-à-un» puissent fonctionner. Inversement, lorsque les images sont trop bruitées, l'édition de partition va toujours trouver un contour permettant de rapprocher les segmentations. Ainsi, la figure 3.2 présente le cas de deux images d'une même scène. Dans la seconde image, un personnage apparaît. L'édition de partition réussit à trouver des contours dans l'image vide correspondant à ceux de la personne. L'algorithme d'appariement utilisera les bons résultats des régions présentes dans les deux images pour propager un appariement incorrect dans les régions ne se correspondant pas. De fait, toutes les régions de l'image seront appariées malgré la présence d'un intrus. Nous ne pouvons donc pas utiliser cette méthode pour l'interprétation de scènes si nous ne pouvons pas différencier un intrus du reste de l'image !

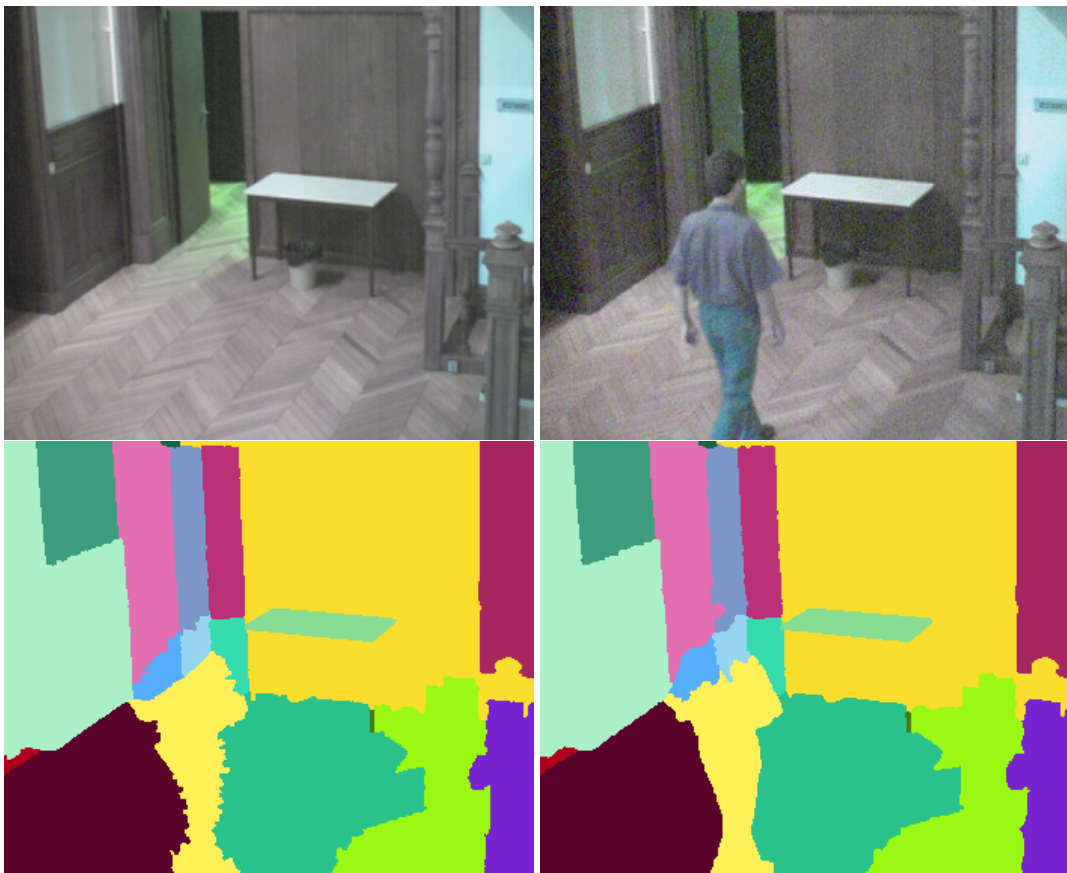


FIG. 3.2: Appariement de régions. On affecte une couleur identique aux couples de régions appariées. On voit ainsi que les régions de l'intrus sont incorrectement appariées à des régions de l'image vierge. L'image est suffisamment bruitée pour que l'édition de partition trouve des contours similaires, et les couleurs sont trop peu marquées pour être discriminantes.

Nous atteignons ainsi les limites de cette approche. Les difficultés qui persistent sont à présent dues à la segmentation, et en particulier à la difficulté d'obtenir une segmentation en un nombre raisonnable de régions, via la segmentation hiérarchique. Une conception différente de l'appariement va être nécessaire pour pouvoir traiter des images plus difficiles.



FIG. 3.3: Extraits pathologiques du suivi de l'individu dans la séquence MPEG «foreman». Malgré nos améliorations de l'algorithme d'appariement, le problème de la qualité de la segmentation hiérarchique en un faible nombre de régions reste déterminant.

3.3 Approche «many-to-many»

Une première modification fondamentale consiste à employer une approche «many-to-many». En l'occurrence, il s'agit d'admettre qu'il est vain de supposer que la segmentation ou la modélisation peuvent engendrer des représentations suffisamment proches pour qu'une correspondance exacte soit possible. En revanche, il est beaucoup plus naturel de chercher des correspondances entre des *groupes* de régions. La figure 3.4 illustre cette approche : au lieu de chercher à appairer les régions individuellement, on va chercher à appairer des groupes de régions.

Cette approche nous permettrait de retirer le mécanisme d'édition de partitions en l'insérant de facto dans le cœur de l'algorithme.

3.3.1 État de l'art

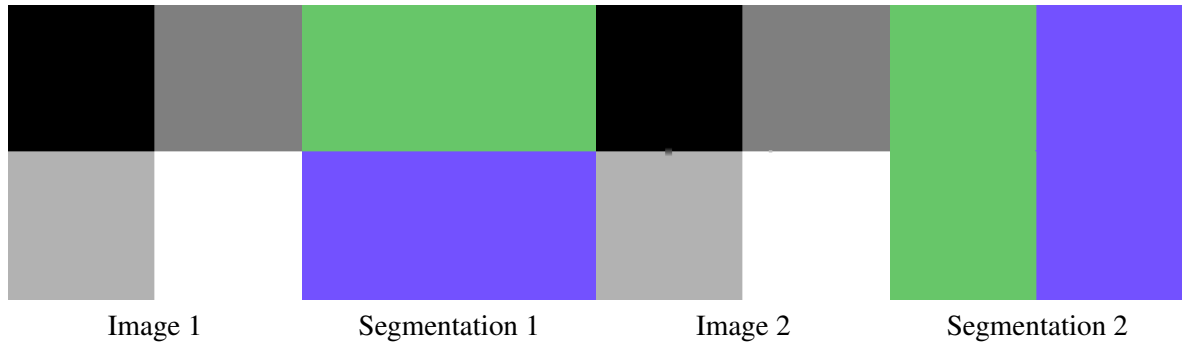
Le problème de l'appariement «many-to-many» est sa grande complexité algorithmique (au plus NP pour les isomorphismes de graphes, NP-complet pour les isomorphismes de sous-graphes³). Les solutions combinatoires et en général les approches exactes sont donc à écarter. Les approches que l'on trouve dans la littérature ne sont en général pas aussi bien ancrées mathématiquement que dans le cas «one-to-one». Il est donc difficile d'évaluer a priori quelle méthode sera la plus appropriée. Le choix est d'ailleurs compliqué par le fait que chaque approche nécessite sa propre représentation : l'une fonctionnera sur des graphes, l'autre sur des arbres.

Une approche qui semble intéresser la communauté de la théorie des graphes est le «low-distortion tree embedding». On citera par exemple [Demirci et al., 2003] comme représentant cette mouvance. Il s'agit de construire à partir d'un graphe un arbre représentant au mieux les distances entre les nœuds. On utilise ensuite des outils de comparaison d'arbres (projection spectrale, «Earth-Mover Distance», etc) avant de revenir sur les graphes d'origine. Cette notion d'«embedding» nous intéresse peu car nous disposons toujours de l'image qui contient une information beaucoup plus riche. Nous pouvons donc nous baser sur celle-ci pour construire un arbre. En revanche, les techniques de comparaison d'arbres nous seront utiles plus loin. Un autre souci de nombre de ces approches est qu'elles ne répondent pas réellement au problème du bruit. Par exemple Keselman [Keselman et al., 2003], qui utilise une approche similaire à Demirci, teste sa robustesse au bruit en retirant des composantes connexes d'une taille comprise entre 5% et 25% du nombre total de nœuds. Ceci correspond à tester la robustesse face à une occlusion partielle mais les problèmes causés par des divergences de segmentations sont différents. Les figures 3.15 p. 45 et 3.14 p. 45 montrent un cas trivial où 33% des arêtes sont problématiques.

Une approche différente nous a paru plus intéressante, notamment par sa capacité à, apparemment, gérer le problème des objets différemment segmentés.

³Ce problème nomme d'ailleurs une classe de complexité à part : GI pour «graph isomorphism».

Problème des instabilités :



Solution par appariement “many-to-many” :

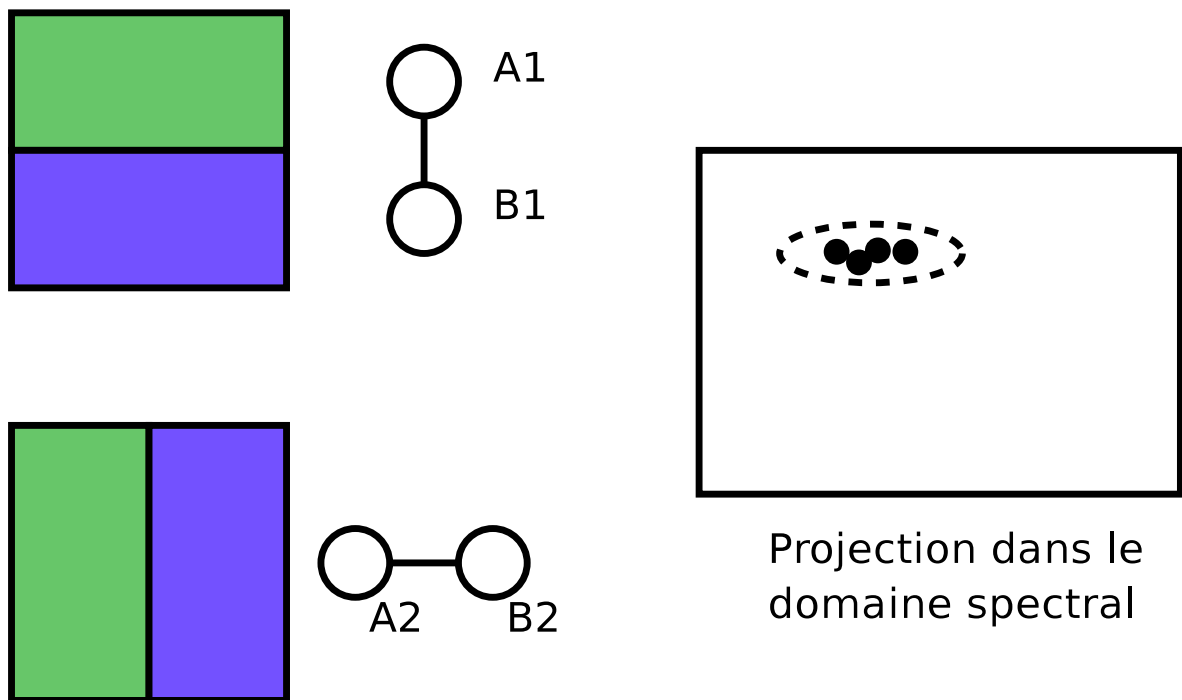


FIG. 3.4: Solution du problème de l’instabilité par l’appariement “many-to-many” dans le domaine spectral. Le principe est de constater que les quatre régions sont semblables structurellement (par rapport aux régions avoisinantes) et visuellement (puisque la segmentation oscille entre les deux formes) et donc peuvent être rapprochées par la projection dans un espace adéquat.

3.3.2 Approche spectrale

Cette approche héritée de l'analyse de données vise à projeter un graphe dans un espace rapprochant les points ayant une structure topologique similaire. Un des points-clés de la décomposition spectrale est que les valeurs propres sont invariantes par toute permutation P : la matrice PAP^{-1} a les mêmes valeurs propres que la matrice A . Dans le domaine de la théorie des graphes, cela signifie que le spectre d'un graphe, calculé à partir de sa matrice d'adjacence est indépendant de l'ordre dans lequel on numérote ses nœuds.

Nous nous sommes basés sur l'approche développée par Kosinov et Caelli dans [Kosinov and Caelli, 2002, Caelli and Kosinov, 2004] qui consiste à projeter indépendamment chaque graphe dans un sous-espace propre de dimension réduite (2 ou 3), à identifier les deux espaces et à regrouper les nœuds par un algorithme de «clustering».

3.3.3 L'approche Kosinov

Étant donnés deux graphes X et Y , on va calculer, pour chaque graphe, les vecteurs propres de sa matrice d'adjacence. Comme X et Y peuvent avoir un nombre de nœuds différents, on ne peut pas comparer directement les espaces propres. On va donc procéder en deux étapes pour retrouver des choses comparables.

La première idée est inspirée de l'analyse en composantes principales (ACP). On va ici projeter les nœuds de chaque graphe sur le sous-espace propre associés aux n valeurs propres les plus grandes, avec un n bien adapté aux données (en général 2 ou 3, mais il y a des méthodes plus précises pour trouver le meilleur n).

Les projetés des nœuds sont des vecteurs de même dimension mais dont les valeurs ne sont pas comparables. La seconde idée est d'accepter une perte d'information en normalisant les vecteurs, et en constatant que l'agencement angulaire des vecteurs reste discriminant (Voir figure 3.5).

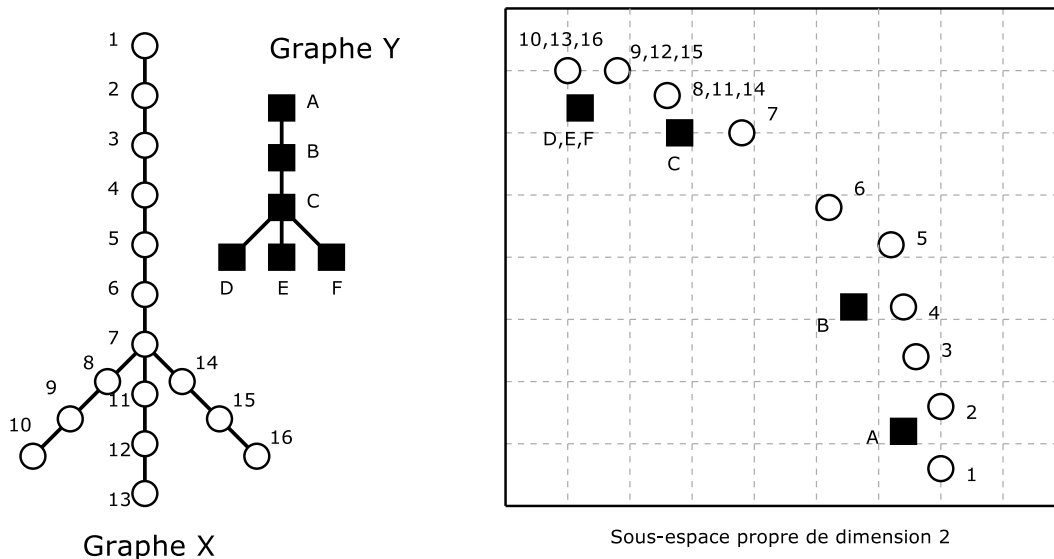


FIG. 3.5: Projection des nœuds de deux graphes dans leurs sous-espaces propres de dimension 2, et normalisation. Les points des zones topologiquement identiques sont superposés (C,D,E, etc.).

Une fois les points projetés et normalisés, on peut aligner les deux sous-espaces propres et tenter

de regrouper les nœuds proches dans les deux graphes. L'idée étant que les nœuds formant deux composantes topologiques se correspondant auront été projetés dans des mêmes zones.

Un algorithme de clustering du type «plus proches voisins» peut ensuite être appliqué à l'union des deux ensembles de nœuds.

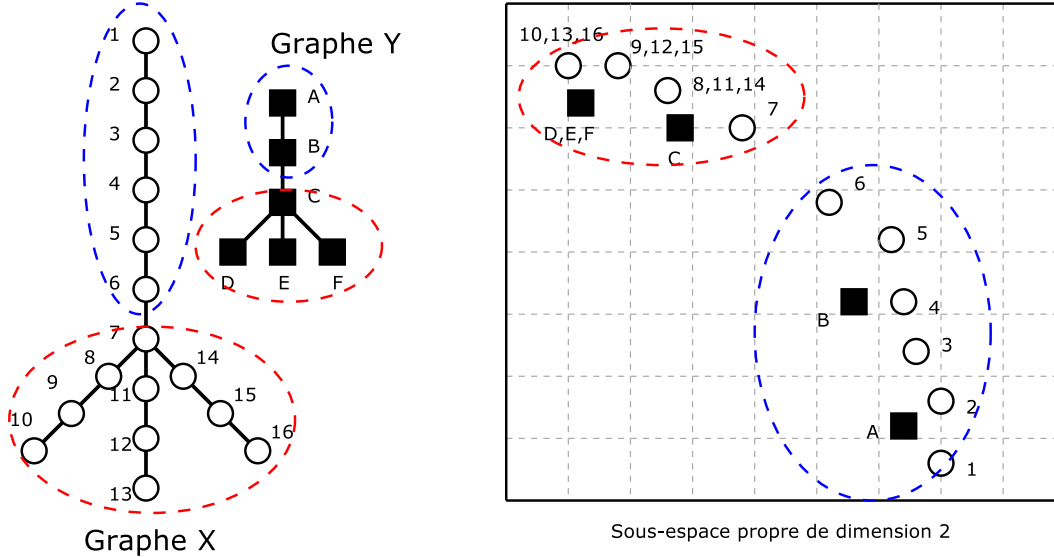


FIG. 3.6: Appariement «many-to-many» en regroupant les points projetés en deux nuages dans le sous-espace spectral.

3.3.4 Nos extensions

La méthode de Kosinov fonctionne sur des graphes non valués, en comparant exclusivement leur topologie. Nous avons cherché à l'étendre au cas où les nœuds des graphes étaient valués et où nous disposions d'une distance entre nœuds.

Pour ce faire, nous avons choisi de procéder en deux étapes : la première étape utilise l'algorithme de Kosinov de façon standard. On obtient ainsi une projection dans l'espace spectral encodant la topologie des graphes. Dans une seconde passe, on va utiliser à la fois les informations de distances couleur entre les nœuds ainsi que les distances dans le sous-espace spectral pour construire une matrice de distances que l'on projettera de la même façon dans un sous-espace propre.

La figure 3.7 montre les graphes utilisés et leur projection purement topologique dans un sous-espace propre de dimension 2, ainsi que le résultat d'un clustering dans cet espace propre.

Dans une seconde étape, on value le graphe en affectant des couleurs à chaque nœud, et on calcule une matrice de distances couleurs entre tous les nœuds des deux graphes :

$$M_{ij}^{couleur} = dist_{couleur}(N_i, N_j), \forall N_i, N_j \in G_1 \cup G_2$$

On calcule également la matrice des distances euclidiennes des projections dans le sous-espace propre calculé précédemment (basé uniquement sur la topologie) :

$$M_{ij}^{topo} = dist_{geom} N_i, N_j \forall N_i, N_j \in G_1 \cup G_2$$

On normalise ces matrices (en divisant par leur max) et on calcule une somme pondérée :

$$M^{final} = \alpha \frac{M^{couleur}}{\max(M^{couleur})} + (1 - \alpha) \frac{M^{topo}}{\max(M^{topo})}$$

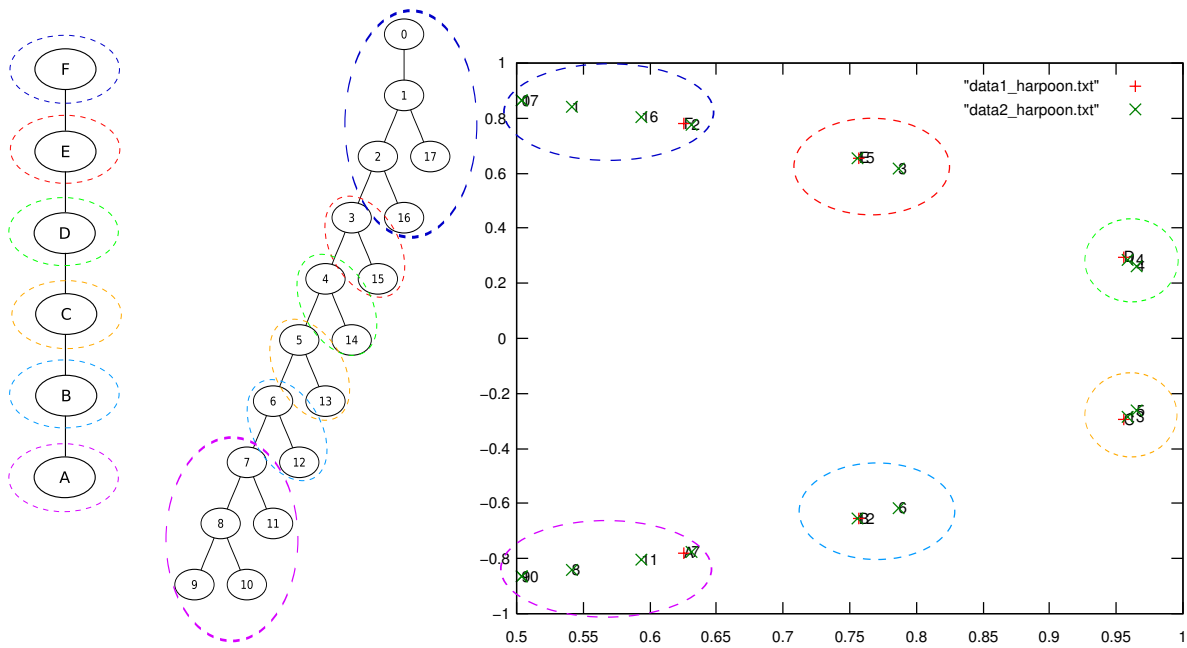


FIG. 3.7: Projection dans des sous-espaces propres de dimension 2 de deux graphes. Seule la topologie des graphes est utilisée.

De façon standard, on modifie ensuite cette matrice pour écarter les valeurs très distantes :

$$M_{ij}^{final} = \exp\left(-\frac{(M_{ij}^{final})^2}{2\sigma^2}\right)$$

On calcule ensuite un sous-espace propre de cette matrice dans lequel on projette tous les nœuds. La figure 3.8 montre que l'on a bien réussi à injecter l'information de couleur dans l'appariement.

Nous avons essayé de fusionner les deux approches en intégrant directement les distances couleur dans le clustering topologique. Nous avons ainsi construit des matrices de distance plutôt que des matrices d'adjacence que nous avons projeté de façon indépendante. Le problème que nous avons rencontré dans cette approche était que les clusters des deux graphes ne se correspondaient pas : l'hypothèse de Kosinov que les vecteurs propres sont superposables n'est apparemment plus valide dans ce cas.

Ceci nous a conduit à cette approche en deux étapes : une gérant la topologie, la suivante intégrant la couleur. L'exemple ci-dessus montre un bon comportement de l'algorithme face aux problèmes de resegmentation excessive (chaque nœud du premier graphe étant resegmenté en de multiples nœuds dans le second graphe) tout en tenant bien compte de l'information chromatique.

Nous allons à présent pouvoir utiliser cette approche sur des graphes issus de nos segmentations hiérarchiques.

3.3.5 Résultats sur images

Le premier essai que nous avons mené utilisait deux images de la base COIL. Ces deux images d'une tirelire ont été prises avec un écart angulaire de 5 degrés. Ceci correspond au cas fréquent en analyse de vidéos où la caméra tourne autour d'un objet (soit réellement, soit de façon apparente lors du déplacement de l'objet).

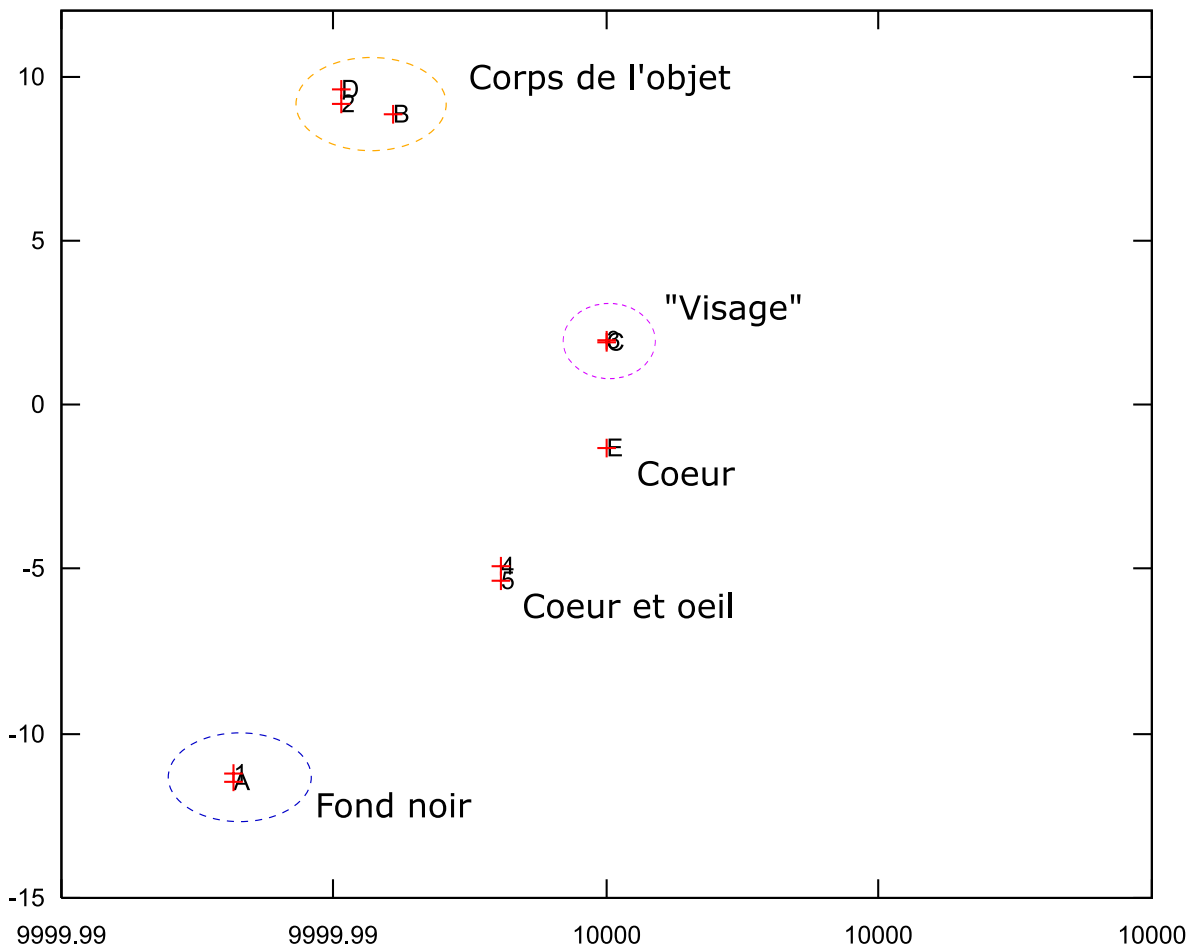
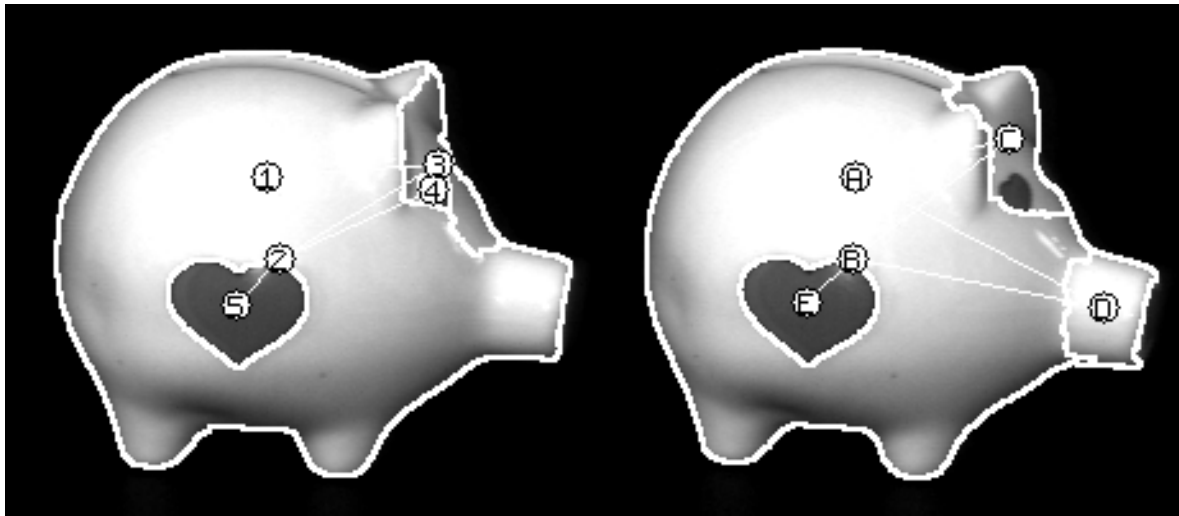


FIG. 3.9: Appariement des segmentations dans le domaine spectral. Le corps de l'objet a été correctement apparié (régions 2 d'une part et B et D d'autre part), mais les cœurs (et l'oeil) posent problème.

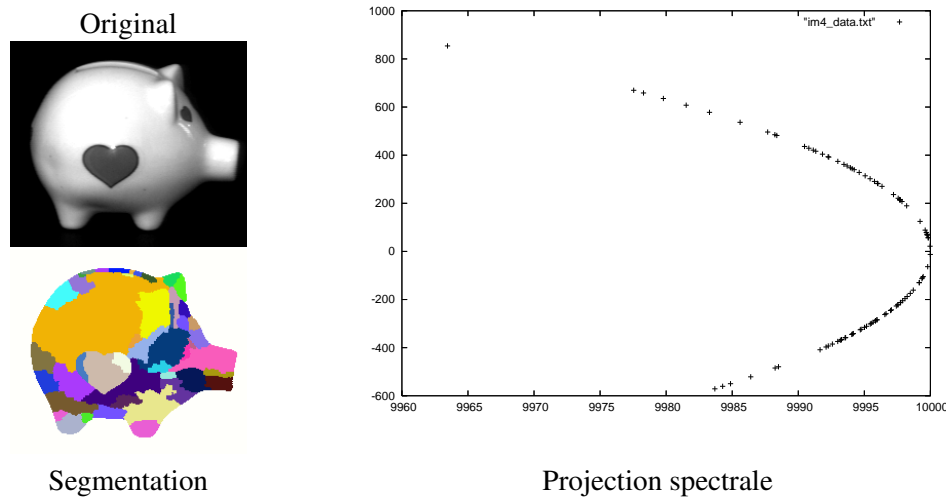


FIG. 3.10: Projection d'une sursegmentation dans le domaine spectral. Malgré l'utilisation des niveaux de gris, aucune structure ne ressort. Ce résultat n'est pas utilisable.

à celui-ci, nous avons obtenu un léger gain dans la précision de nos appariements ainsi qu'une diminution des temps de calcul. Cependant, l'approche «one-to-one» nécessite par nature des segmentations semblables, et ce quelle que soit la robustesse de l'algorithme d'appariement. Or, la méthode dont nous disposons pour ajuster des segmentations (l'édition de partitions) n'est valable que pour des images très voisines (faible mouvement global, peu d'occultations, etc).

Par la suite, nous avons cherché à relâcher cette contrainte de proximité des images en utilisant un algorithme «many-to-many». Nous avons donc développé une méthode se basant sur les travaux de Kosinov et Caelli sur la comparaison de graphes dans le domaine spectral. Malgré des résultats initiaux encourageants, cette méthode ne s'est pas révélée adéquate pour résoudre notre problème.

3.4.2 Pertinence des développements théoriques

Nous avons relevé un point important dans notre étude de la littérature sur l'appariement de graphes : dans presque tous les cas, la segmentation des objets est supposée donnée et correcte. En général, on y considère que l'occultation d'un objet (donc la suppression d'un groupe de nœuds) est le principal problème à résoudre. Ceci n'est en général vrai qu'avec des images faciles à segmenter ou avec des segmentations effectuées manuellement. Par ailleurs, la robustesse face au bruit est testée en modifiant aléatoirement les graphes. Dans la réalité, une erreur de modélisation causera des modifications de la topologie fortement localisées autour de l'erreur. Cependant, dans les simulations, les choix des nœuds ou des arêtes à ajouter ou retirer suivent des lois de probabilités qui ne tiennent pas compte de cette dépendance.

3.4.3 Images réelles et images de laboratoire

Du côté du traitement d'image, le problème est similaire : une grande part de la recherche académique actuelle en segmentation et en particulier en segmentation d'images couleur est testée sur des bases de données relativement «faciles» (voir figure 3.11). La base COIL de l'université de Columbia

[Murase et al., 1996] présente des objets colorés sur un fond noir, évitant ainsi l'écueil de la segmentation. La base de Berkeley [Malik et al., 2001], plus complexe, présente des images de grande qualité, en particulier au niveau de la richesse de l'information chromatique : les images sont peu bruitées, et les couleurs sont belles car toute la gamme de saturation est utilisée.

Nombre de ces algorithmes ou de ces avancées technologiques deviennent malheureusement caducs lorsqu'on leur présente des images «réelles» plus difficiles (cf fig 3.12). Les images rencontrées dans les projets industriels applicatifs accumulent plusieurs effets néfastes. D'une part, le niveau de bruit dans les images est souvent important. La taille (via la résolution) de ces images étant souvent limitée, l'impact d'un pixel bruité sera plus grand que sur une image de grande taille. Par ailleurs, les capteurs CCD ont souvent un comportement assez pathologique vis-à-vis de la couleur. Les deux images de la figure 3.12 ont été prises à quelques minutes d'intervalle. On voit que la teinte globale des deux images a changé, et que la compensation automatique de gain a fait virer la teinte de la lumière dans l'image de gauche au verdâtre (alors que c'est une lampe à incandescence donnant la lumière jaune usuelle).

Pris indépendamment, ces problèmes ont souvent fait l'objet de recherches de la part des chercheurs. Dans les cas qui nous occupent, en revanche, les difficultés tendent à s'accumuler. La figure 3.13 montre que le postulat selon lequel la segmentation hiérarchique permet de bien extraire les objets au moins jusqu'au niveau moyen de la partition n'est plus valide.

Ces images issues de systèmes «réels» représentent une problématique industrielle tout autant qu'une gageure scientifique, et nécessitent une révision de nos méthodes de traitement.



FIG. 3.11: Images "de laboratoire" de bonne qualité, actuellement utilisé comme références pour les algorithmes de segmentation.

3.4.4 Conclusion : vers une redéfinition de l'approche par graphes

Cette approche a été initialement développée pour des séquences dans lesquelles le mouvement est faible, et la segmentation peu difficile. Dans le cas d'une approche plus générale, avec des séquences plus complexes, outre le problème de la segmentation mentionné plus haut, on se retrouve face à plusieurs problèmes :



FIG. 3.12: Images acquises par un imageur ST dans des conditions non-idéales. Le bruit devient sensible lorsque la lumière baisse, de même que les dérives chromatiques dues à la compensation automatique. Les décrochages sont causés par des problèmes de synchronisation entre l'imageur et le serveur de traitement (Source : CMM).

1. à supposer qu'on obtienne des segmentations décentes, il est difficile de les ajuster pour les rendre comparables dès lors qu'on a un mouvement non négligeable.
2. pour les objets en mouvement, les relations de voisinage avec le fond ou les autres objets ne sont plus valides.

Pour le premier point, on a vu que l'édition des partitions pouvait avoir des comportements curieux même dans des images faciles. Par ailleurs, gérer cette difficulté au niveau de l'appariement des graphes s'est avéré beaucoup moins efficace que nous ne le pensions à l'origine.

En pratique, il s'avère difficile de rattraper une segmentation incorrecte, que ce soit via édition de partition, appariement «many-to-many», ou autre. Il faut donc trouver une manière de stabiliser la segmentation pour limiter la probabilité que des images comparables aient des segmentations trop distantes.

Le second point, la stabilité des relations de voisinage, apparaît même avec des images faciles et des segmentations parfaites. La figure 3.14 illustre le problème de la conservation des relations de voisinage au cours d'une séquence avec un objet en mouvement. Sur un autre aspect, la figure 3.15 montre comment de faibles mouvements d'un objet peuvent modifier de façon conséquente la topologie du graphe. Il faut donc au minimum traiter différemment les relations de voisinage entre les régions d'un même objet de celles entre des objets différents lorsque ceux-ci sont en mouvement relatif. On en vient ainsi à devoir traiter indépendamment les objets, comme cela est suggéré dans le cadre des plans-objets de MPEG4 ([Pictures and Audio, 2002]).

Nos travaux futurs vont donc se concentrer sur deux axes : d'une part l'amélioration des images, afin que les segmentations fluctuent moins et d'autre part le développement d'approches plus centrées sur les objets : d'abord les segmenter correctement et ensuite les modéliser chacun indépendamment.

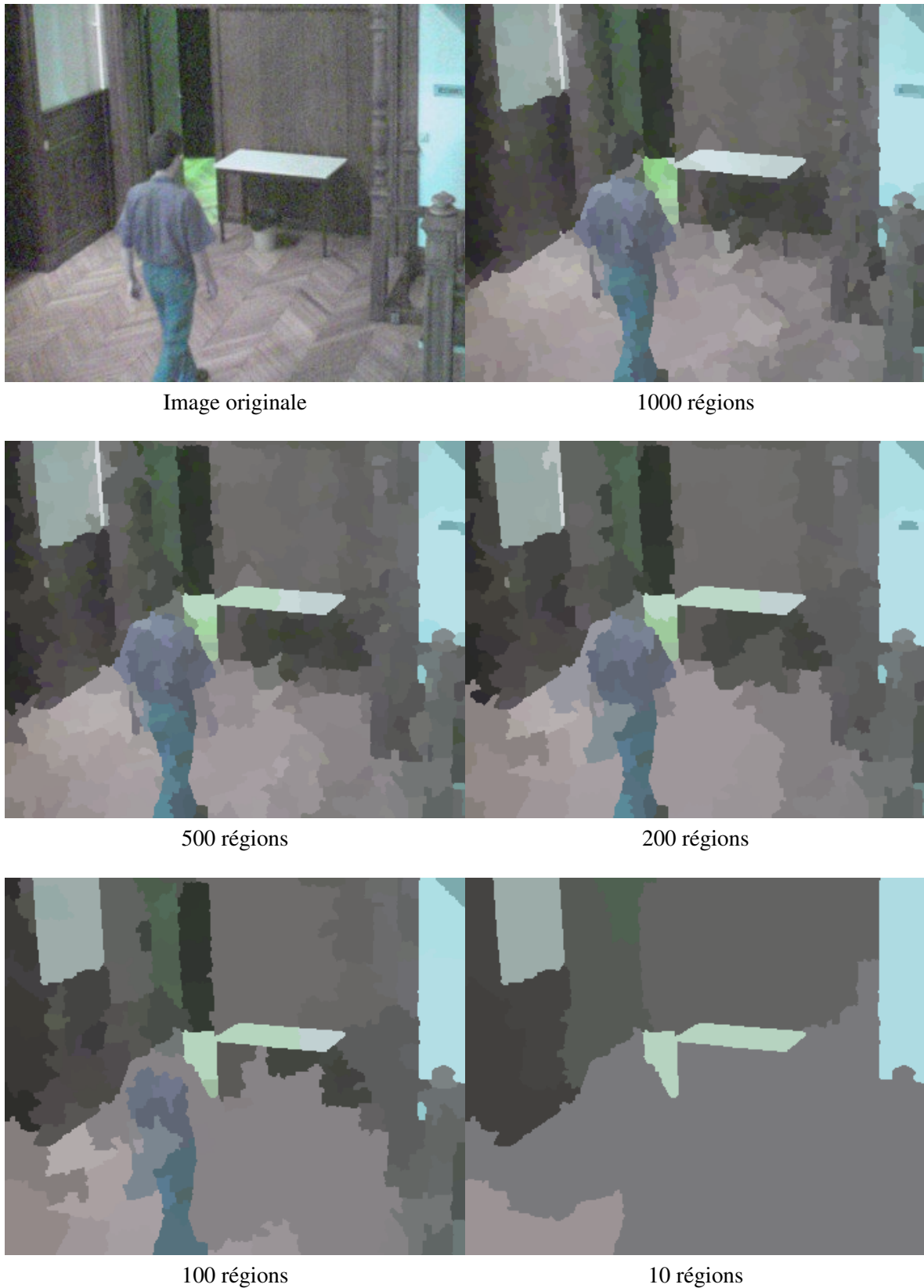


FIG. 3.13: Segmentation hiérarchique d'une image «difficile». Le personnage n'est plus correctement détecté dès la fusion en 200 régions. La segmentation qu'un humain jugerait «correcte» utilise en réalité un contenu sémantique (personne, objets, fond) que nous devons considérer pour définir des algorithmes corrects.

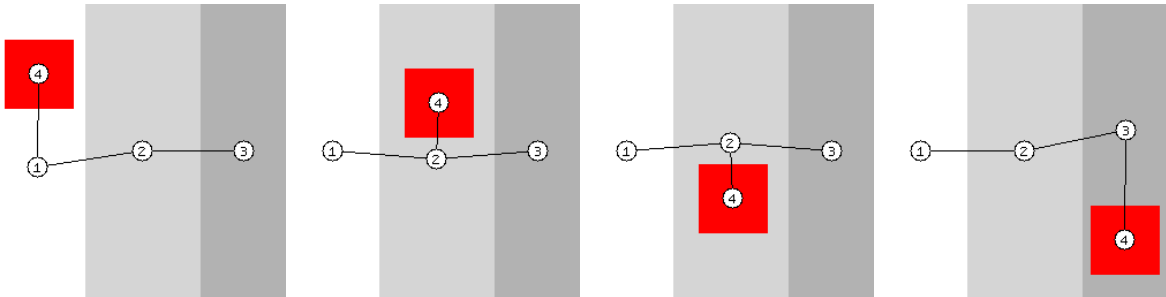


FIG. 3.14: Suivi d'un objet en mouvement : sur trois arêtes, deux seulement peuvent être mises en correspondance, l'arête attachée au mobile étant toujours en situation de parasite. Peu d'algorithmes de la littérature sont capables de supporter un rapport de bruit de 33%.

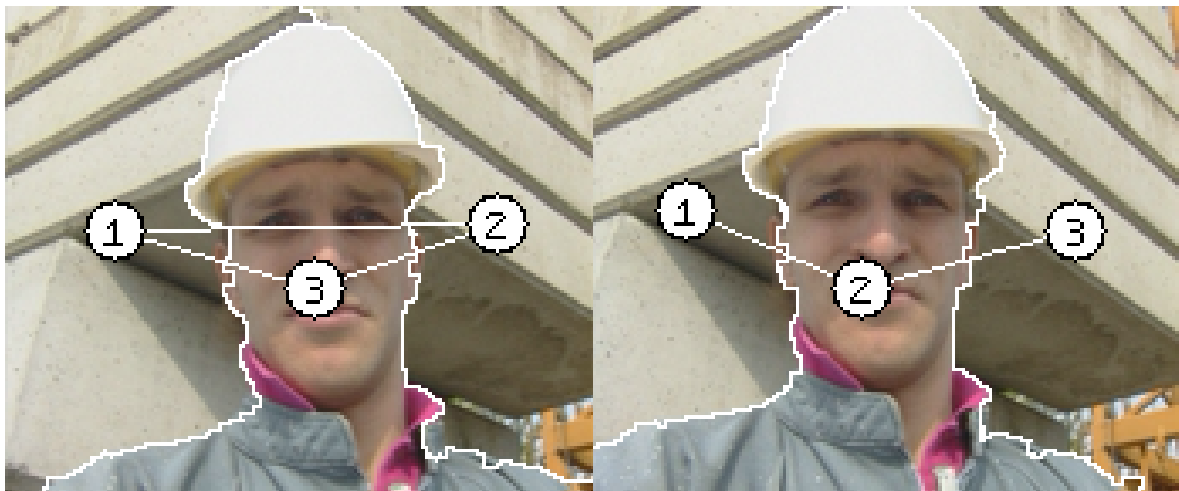


FIG. 3.15: Suivi d'un objet : même avec un mouvement faible, la restriction imposée par le champ de la caméra peut fortement influencer sur la topologie du graphe. Selon la position du casque, les zones de fond sur les deux côtés sont connectées ou non. Encore une fois, 33% des arêtes sont problématiques.

Deuxième partie

Filtrages

4	Filtrage d'images par amibes morphologiques	51
4.1	Introduction	51
4.2	Problématique originelle	52
4.3	Principe	55
4.4	Construction	57
4.5	Propriétés	60
4.6	Amibes et morphologie mathématique	61
4.7	Les amibes en pratique	66
4.8	Résultats	67
4.9	Conclusions et perspectives	77
5	Gradients	81
5.1	Introduction	81
5.2	Réflexions sur le gradient et la segmentation	82
5.3	Le problème de la couleur	83
5.4	Gradients morphologiques généralisés	84
5.5	Résultats	87
5.6	Conclusion	96

Filtrages

La première partie de ce travail visait à étudier jusqu’où nous pouvions étendre la validité du travail de Cristina Gomila et à identifier les points de faiblesses sur lesquels nous pouvions concentrer nos efforts.

Nous avons ainsi étendu cette approche initialement développée pour comparer les images successives dans une séquence vidéo d’un type précis (vidéo-conférence) à des séquences plus générales, puis à des images de plus en plus distantes. Nous avons constaté, au moins de façon empirique, une rapide diminution de la qualité dès lors que les images cessaient d’être facilement appariables.

Nous avons cherché dans le chapitre suivant à utiliser des méthodes d’appariement de graphe plus performantes. Les résultats ont été sensiblement améliorés, mais pas au point de pouvoir correctement gérer des segmentations trop différentes.

Il existe deux cas dans lesquels deux images peuvent avoir des segmentations différentes : soit ces images sont proches mais les segmentations indépendantes ont divergé, soit les images sont naturellement éloignées. Dans le premier cas, on peut chercher à segmenter de façon moins indépendante. Dans le second cas, on peut considérer le cas où certains objets d’une image se retrouvent dans l’autre. La partie qui va suivre vise à améliorer le premier cas en introduisant des filtrages et des améliorations dans nos algorithmes de gradient. Le second cas sera le sujet de la partie suivante.

Premier axe d’amélioration : le filtrage

L’idée sous-jacente à cet axe de recherche est de considérer que deux images voisines d’une même scène ou d’un même objet ne sont que des représentations légèrement différentes d’une même réalité physique. Ainsi, même si le bruit présent dans deux images similaires conduit à faire diverger leurs segmentations respectives, il n’en reste pas moins que ces images ont essentiellement le même sujet.

Lorsqu’un être humain examine deux images similaires, il est parfaitement capable de les appairer, même en présence d’un fort niveau de bruit. On peut citer par exemple les images d’échographie comme cas typique d’images difficiles à segmenter automatiquement et qui pourtant sont analysées avec une grande précision par des médecins. La raison de cette efficacité de l’homme face à la machine réside dans sa capacité à intégrer de l’information sur de larges zones, aussi bien que sur des détails saillants. C’est cette constatation qui a donné naissance à la théorie des espaces d’échelle ([Witkin, 1983], [Koenderink, 1984] et [Lindeberg, 1994]). Sans aller jusqu’au niveau le plus haut de l’échelle, on peut chercher à étudier l’image dans un voisinage plus large que ce qui est fait habituellement.

En pratique, nous proposons deux approches indépendantes utilisant cette idée de voisinage étendu. La première consiste à filtrer directement l’image en réduisant le bruit et lissant les zones plates tout en respectant les contours. Ce sont les filtrages à base d’«amibes morphologiques» (chapitre 4). La seconde approche consiste à revisiter la notion de gradient et à définir des gradients plus robustes face

à un fort niveau de bruit (chapitre 5).

Ces deux approches peuvent être utilisées indépendamment l'une de l'autre, ou être combinées. On verra qu'elles permettent de renforcer les contours et de retrouver l'information présente globalement dans les objets de l'image tout en restant relativement robustes face au bruit.

Chapitre 4

Filtrage d'images par amibes morphologiques

*Il n'y a guère que les amibes qui se divisent
pour se multiplier.*

Hervé le Tellier

Ce chapitre présente nos travaux sur les «amibes morphologiques» qui sont des éléments structurants se déformant pour s'adapter aux contenus des images. Nous présenterons dans ce chapitre leur construction ainsi que leurs propriétés ensemblistes et morphologiques avant de les utiliser pour définir des opérateurs de réduction de bruit extrêmement efficaces.

4.1 Introduction



Amoeba proteus
(Photographie : S. Durr)

En biologie, une amibe est un type de protozoaire qui se déplace à l'aide de pseudopodes. C'est un des plus célèbres organismes unicellulaires. On les trouve partout dans le monde dans les eaux troubles, douces ou salées, ainsi que dans le sol ou comme parasites. Les amibes commencent maintenant une nouvelle vie dans le domaine du traitement d'images.

Ce chapitre présente une récente addition à la boîte à outil du traiteur d'images : les filtrages basés sur les amibes morphologiques. Il s'agit de filtres¹ dont le noyau est déformable et adapte sa forme aux valeurs de l'image sur laquelle on l'applique.

¹Le terme est à prendre dans son acceptation la plus large. On précisera «*filtre morphologique*» pour désigner ces filtres aux propriétés bien précises (croissance et idempotence).

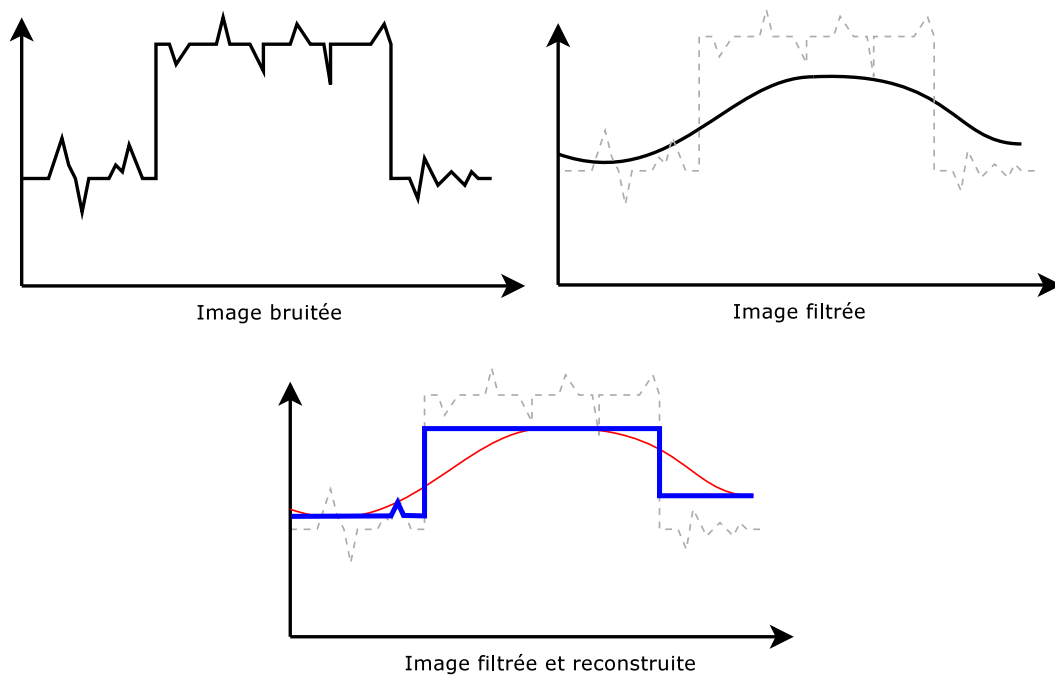


FIG. 4.1: Quand tout se passe bien, le filtre débruiteur retire les signaux parasites et la reconstruction retrouve les contours en reconstruisant peu de bruit.

L'amibe morphologique est une extension pratique et naturelle de la notion d'élément structurant dans le domaine du filtrage anisotrope.

Le champ d'application originel des amibes a été le traitement d'images médicales issues de scanner 3D ([Boehm, 2004]), mais il a rapidement été étendu au traitement de nombreux autres types d'images. Les amibes morphologiques sont aujourd'hui un outil efficace pour débruiter des images avec un minimum de distorsion et des utilisations dans d'autres domaines que l'amélioration d'images sont à l'étude.

4.2 Problématique originelle

4.2.1 Le bruit connexe

C'est au cours des travaux réalisés par Mathilde Boehm [Boehm, 2004] qu'est apparu un problème majeur dans l'approche morphologique standard de réduction du bruit. Cette approche consiste dans un premier temps à appliquer un filtrage qui a pour but de supprimer le bruit (convolution par une gaussienne, filtre médian, filtres alternés séquentiels, etc.) suivi par une reconstruction qui tente de replacer ceux des contours qui n'ont pas complètement disparu (voir figure 4.1). Dans une optique de segmentation, il est en effet important que les contours ne soient pas globalement déplacés.

Le problème principal de cette méthode est illustré par la figure 4.2. On voit bien que même si le bruit est parfaitement effacé dans les zones plates, il est reconstruit au voisinage des zones de gradient.

Dans le domaine des images 2D, il est rare que ce problème s'avère un réel handicap. Il en va tout autrement dans le domaine des images 3D.

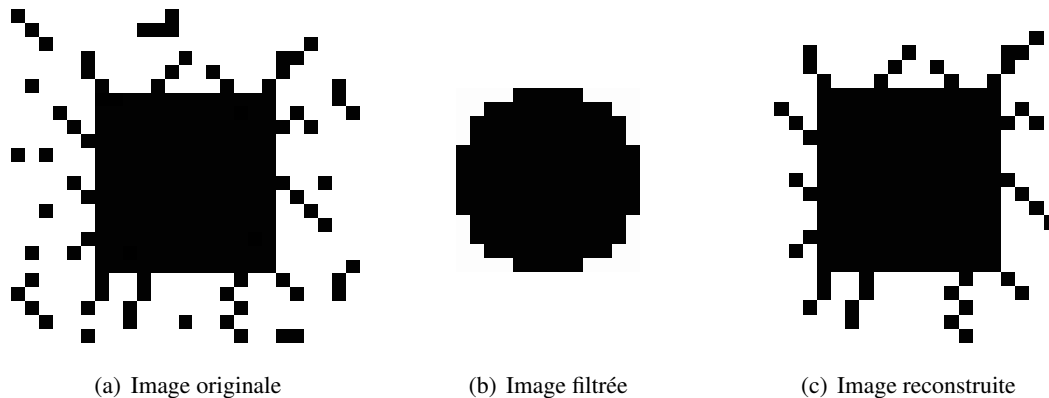


FIG. 4.2: Le filtre débruiteur élimine tout le bruit de l'image, mais perd beaucoup d'information. La reconstruction permet de retrouver la position des contours mais reconstruit également tout le bruit connexe à l'objet.

4.2.2 Le cas 3D

La visualisation de volumes 3D est un problème vaste qui dépend fortement du domaine d'application. Deux méthodes principales existent pour visualiser ces volumes : d'une part celles basées sur un algorithme de lancer de rayons, et d'autre part celles qui calculent des surfaces à partir du volume (isosurfaces ou autres) et affichent ensuite ces surfaces, en général après une triangulation.

Ce travail concerne principalement les algorithmes basés sur du lancer de rayons, mais il est probable que les méthodes de calcul de surfaces bénéficieraient également des techniques de réduction du bruit décrites dans ce chapitre.

Dans les grandes lignes, la plupart des algorithmes de visualisation de volumes 3D par lancer de rayons utilisent de façon similaire l'information de gradient (via le calcul d'une normale locale) à diverses étapes du rendu, en particulier pour calculer la contribution de chaque point. La qualité finale dépend donc beaucoup plus fortement de celle du gradient que dans le cas de l'affichage d'images 2D.

Pour pouvoir effectuer des mesures réellement significatives, Mathilde Boehm a conçu un volume de synthèse présentant des structures similaires à celles présentes dans des images de scanner de la zone cardiaque. Ce volume parfaitement maîtrisé lui a permis de tester le comportement de divers algorithmes face au bruit et à des textures prises dans des volumes réels et plaquées sur le volume de synthèse.

La figure 4.4 montre le rendu du volume de synthèse dans sa forme idéale et dans sa forme bruitée. On voit en particulier que la sphère externe est devenue opaque, alors qu'elle est quasiment transparente dans l'image idéale.

La figure 4.5 illustre le problème des reconstructions : le bruit reconstruit aux frontières des objets bloque pratiquement tout effet de transparence.

Il apparaît donc essentiel de trouver d'autres techniques permettant de retirer le bruit à l'intérieur des objets, tout en préservant la position des contours (pour se passer de l'étape de reconstruction) sans être excessivement perturbé par le bruit connexe aux contours. C'est dans le cadre de cette problématique que nous avons défini les amibes morphologiques et leurs utilisations dans divers types de filtres.

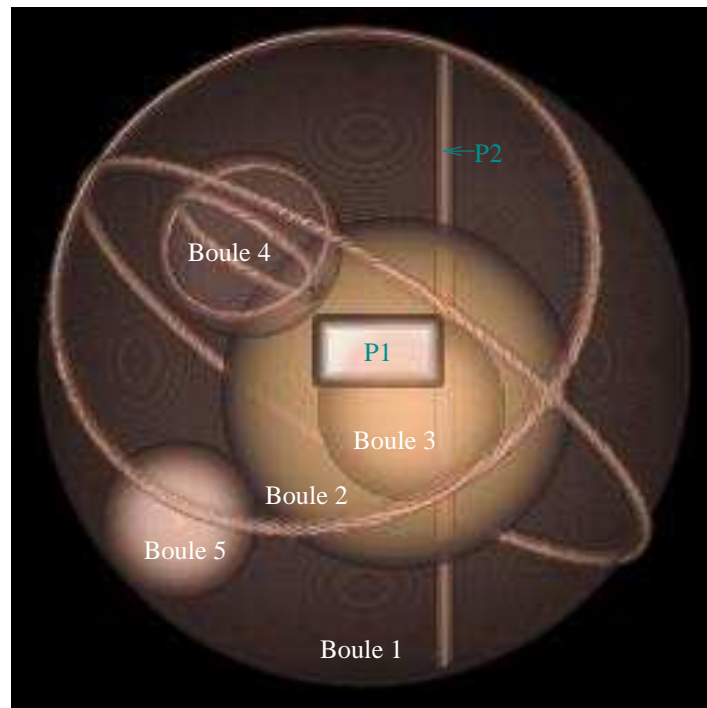
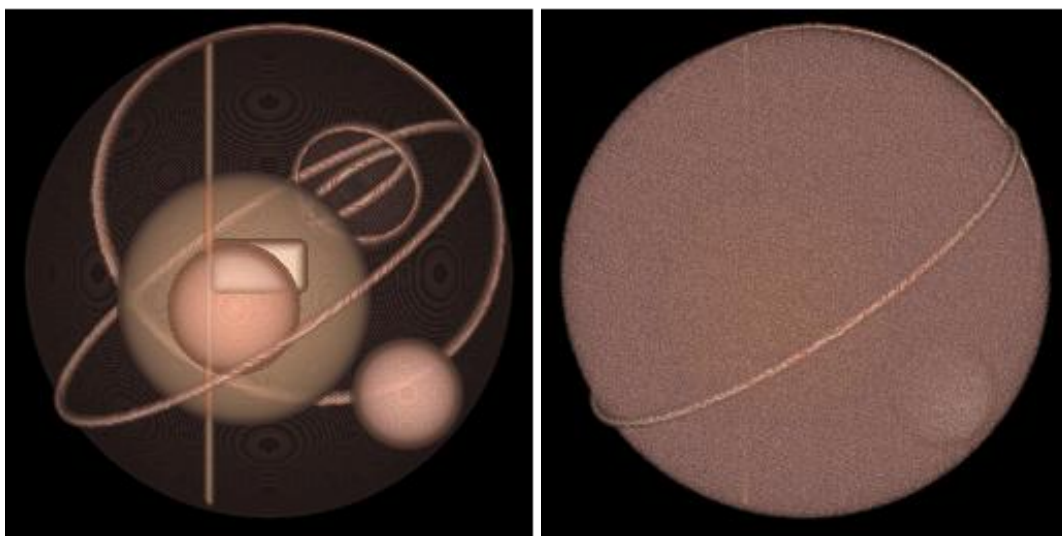


FIG. 4.3: Le volume de synthèse de Mathilde Boehm et ses diverses composantes.



(a) Image idéale

(b) Image bruitée

FIG. 4.4: Incidence du bruit sur la transparence dans le rendu 3D.

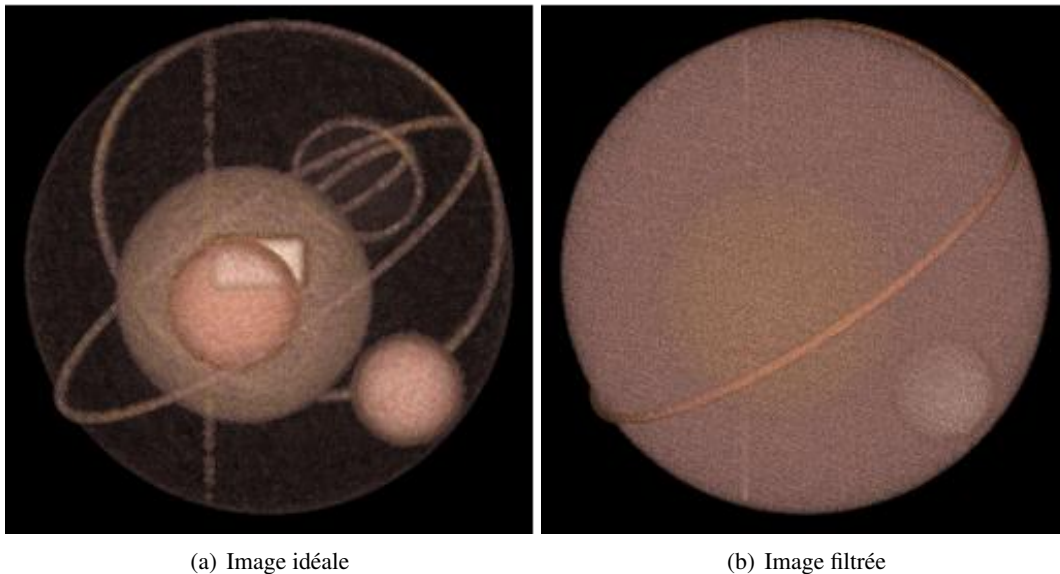


FIG. 4.5: Application d'un filtre alterné séquentiel (FAS) de taille 2 suivi d'une reconstruction. Le FAS retire une bonne partie du bruit mais diffuse excessivement les contours. Problème : le bruit est presque totalement reconstruit au niveau des contours et la transparence n'est que très partiellement recouverte.

4.3 Principe

Les amibes se placent dans le cadre des opérateurs effectuant une mesure sur un certain voisinage dans une image et affectant le résultat de cette mesure au centre du voisinage dans l'image de sortie. Parmi ceux-ci on trouve les opérateurs morphologiques de base (érosion, dilatation) mais aussi les filtres par convolution (du moins ceux à réponse impulsionnelle finie) approximés dans un petit noyau le plus souvent de faible taille (3x3, 5x5, etc).

4.3.1 L'approche traditionnelle

Les opérateurs «classiques», qu'ils soient morphologiques ou par convolution, opèrent sur une fenêtre glissante de taille et de forme fixées et invariantes par translation. Comme la forme de cette fenêtre ne s'adapte pas au contenu de l'image (voir figure 4.6), les résultats sont mauvais à proximité des frontières ou dans les zones où les structures sont petites par rapport à la taille de la fenêtre. On voit par exemple que la diffusion par une gaussienne va atténuer les contours quand son noyau approchera des frontières d'une zone plate, ou qu'une fermeture va relier deux objets pourtant distincts.

Une première amélioration est l'utilisation d'un masque qui permet de restreindre le voisinage à la partie révélée par le masque. On définit ainsi en morphologie mathématique les opérateurs géodésiques basés sur l'érosion et la dilatation (figure 4.7).

4.3.2 Les traitements anisotropes et adaptatifs

Pour pallier les défauts des approches classiques, plusieurs solutions ont été proposées, la plupart rompant complètement avec les méthodes décrites plus haut.

Les travaux de Perona et Malik [Perona and Malik, 1990] sont à la base de la plupart des filtres par diffusion anisotrope qui inhibent la diffusion à travers les forts gradients. Weickert a exposé dans

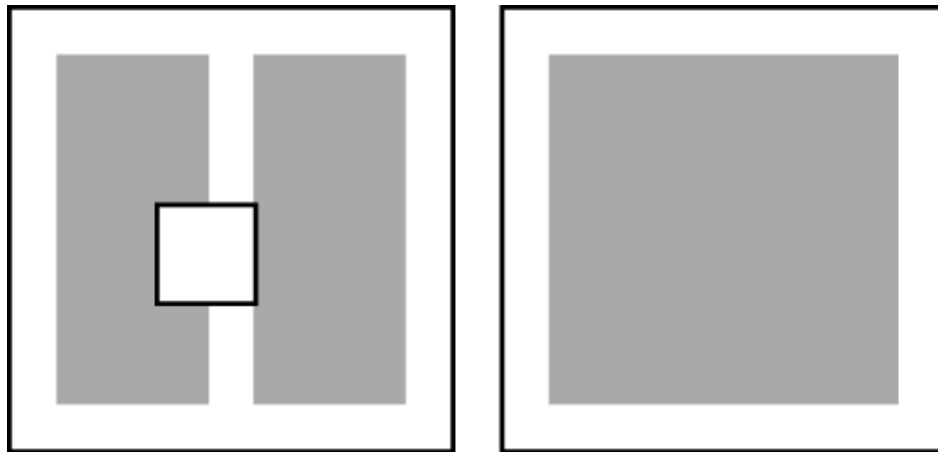


FIG. 4.6: Fermeture par un élément structurant classique. Deux objets distincts sont fusionnés en un seul car l'élément structurant n'adapte pas sa forme et recouvre le canal

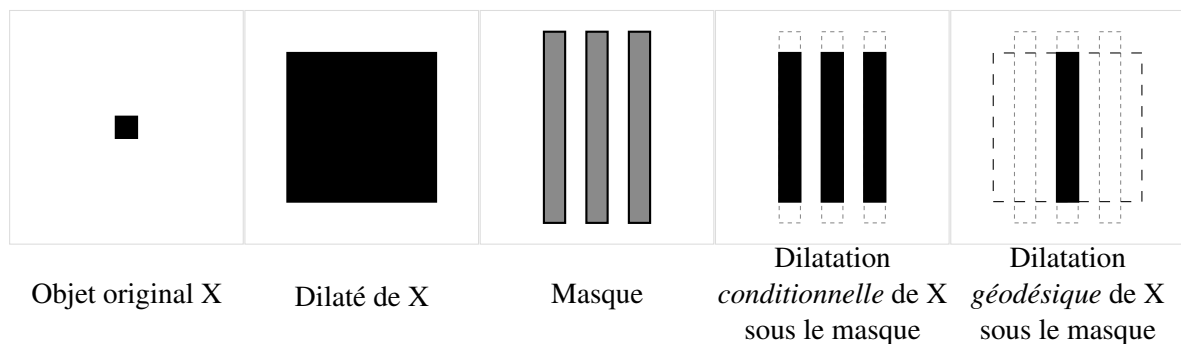


FIG. 4.7: Dilatation géodésique. Forcer une dilatation à respecter un masque permet de lui faire respecter la géométrie des objets définis par ce masque. Le problème est qu'en faisant d'abord une dilatation, puis une comparaison au masque (dilatation conditionnelle), un point peut «déborder» d'une structure vers une autre. La dilatation géodésique impose un critère supplémentaire de connexité.

[Weickert, 1996] des variantes sur cette approche et l'a étendu aux images 3D [Weickert, 1999] et couleurs [Weickert, 1997]. Cependant, ces formulations en équation aux dérivées partielles (EDP) sont assez abstraites et leurs paramètres sont difficiles à relier aux réalités physiques des images sur lesquelles on les applique.

Tomasi et Manduchi ont décrit dans [Tomasi and Manduchi, 1998] un opérateur de filtrage «bilatéral» ne sélectionnant dans une fenêtre que les pixels ayant une valeur proche de celle du centre. Par opposition aux formulations différentielles en EDP, les paramètres de cet opérateur (taille de la fenêtre et écart maximal autorisé avec la valeur du centre) sont des quantités que l'on peut en partie déduire de mesures dans l'image. Cependant, cette méthode ne garantit pas que l'élément structurant sera connexe, et peut causer des soucis lorsque l'on filtre des structures fines, comme dans le contre-exemple de la dilatation géodésique (figure 4.7, «dilatation conditionnelle»).

Danny Barash propose dans [Barash, 2000] une méthode pour rapprocher le filtrage bilatéral de Tomasi des diffusions anisotropes de Weickert et Perona-Malik. Cependant, son approche se base elle-aussi sur des EDP et s'avère donc difficile à paramétrer.

Les travaux de Braga-Neto [Braga-Neto, 1996] reprennent cette notion d'écart maximal autorisé avec le centre en y ajoutant celle de chemin : les points sélectionnés doivent être connexes avec le

centre.

Dans cette chronologie, nos travaux sur les amibes morphologiques ont commencé en 2003 et ont été publiés à partir de 2005 dans [Lerallut et al., 2005a, Lerallut et al., 2005b], et à paraître dans [Lerallut et al., 2006].

4.3.3 L'approche par amibes

Dans ces approches, certains points ont particulièrement attiré notre attention :

- la connexité des éléments structurants déformables de Braga-Neto
- le rôle du gradient dans les diffusions anisotropes
- les paramètres orthogonaux du filtrage bilatéral : écart maximal et taille de la fenêtre.

Nous avons donc exprimé ces caractéristiques dans un cadre général permettant de développer des éléments structurants connexes pouvant se déformer et ainsi ne franchissant pas (ou peu) les zones de fort contour (voir figure 4.8). Ces éléments structurants ont été baptisés «amibes morphologiques».

L'intérêt de notre méthode, comparée à celles basées sur des équations aux dérivées partielles (EDP), est qu'elle dépend d'un faible nombre de paramètres et que ceux-ci peuvent être facilement reliés à des grandeurs mesurables dans l'image.

Par ailleurs, notre approche est dans la continuité des opérateurs définis sur des noyaux à forme fixe : les opérateurs «traditionnels» de traitement d'images comme la moyenne ou la médiane peuvent être utilisés sur des amibes, de même que des opérateurs de rang ainsi que de nombreuses autres mesures statistiques. De plus, sous réserve de quelques vérifications théoriques que nous détaillerons plus loin, les amibes peuvent être utilisées pour définir des filtres morphologiques et plus généralement apporter aux opérations classiques de morphologie mathématique les avantages qui font le succès des approches anisotropes.

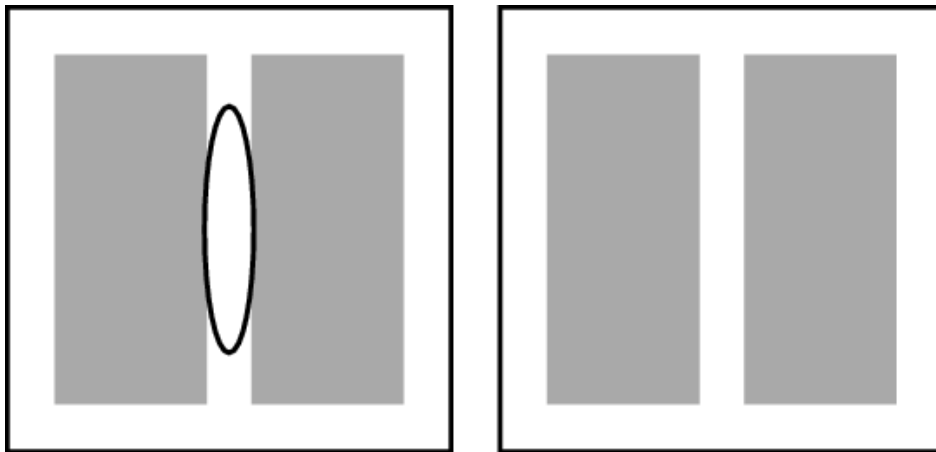


FIG. 4.8: Fermeture par une amibe. L'amibe ne traverse pas les contours forts et préserve donc même les canaux les plus fins.

4.4 Construction

4.4.1 Distance d'amibe

Le noyau d'un filtre par convolution, ou d'un élément structurant morphologique de base est souvent défini formellement sur une boule de rayon r relativement à la norme correspondante à la

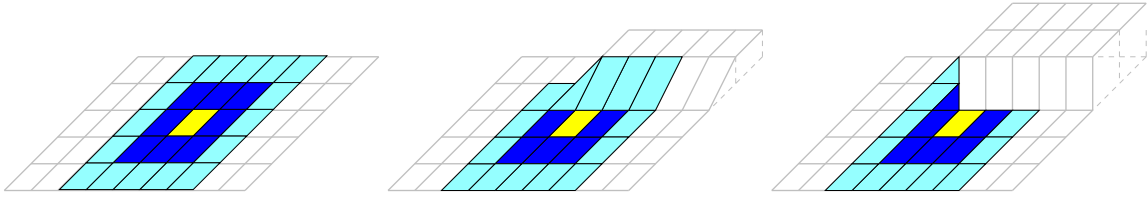


FIG. 4.9: Comportement souhaité d'une amibe sur différents types de relief. Dans les zones plates, l'amibe s'étend au maximum alors que des gradients modérés freinent sa propagation de et que des gradients forts l'empêchent complètement.

connexité choisie. Par exemple, les éléments structurants classiques 8-connexes $\begin{matrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{matrix}$ et 4-connexes $\begin{matrix} \bullet & & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & & \bullet \end{matrix}$ correspondent à la boule de rayon 1 pour les normes L^∞ et L^1 respectivement.

Nous conserverons cette définition pour les amibes, en remplaçant la norme par une qui prenne en compte le gradient de l'image. L'amibe est donc simplement la boule de rayon r associée à une «distance d'amibe».

Définition 1. On appellera «écart», noté $e(p_1, p_2)$ une semi-distance définie dans un espace en niveaux de gris, un espace couleur, ou plus généralement tout espace dans lequel une image peut avoir ses valeurs (notées ici p_i). Dans le cas des niveaux de gris, on prendra en par exemple la valeur absolue de la différence.

Il faut que l'écart soit au minimum une semi-distance :

Propriété 1. L'écart doit vérifier :

- $e(p, p) = 0$
- $e(p, q) \geq 0$
- $e(p, q) = e(q, p)$
- $e(p, q) \leq e(p, r) + e(r, q)$

La propriété manquant à e pour en faire une distance est $e(p, q) = 0 \Rightarrow p = q$. Ceci nous permet de considérer deux teintes voisines comme égales vis-à-vis de l'amibe et donc d'absorber de petites variations de couleur.

Définition 2. Pour tout λ réel positif et pour une image donnée Im , on appellera «distance de voisins» de paramètre λ , notée $v_\lambda(x, y)$ la distance définie par :

$$v_\lambda(x, y) = 1 + \lambda \cdot e(Im(x), Im(y))$$

où x et y sont deux points voisins de valeurs dans l'image respectivement $Im(x)$ et $Im(y)$. Par convention, on pose $v_\lambda(x, x) = 0$.

Pour simplifier la notation, on ne précise pas l'image Im dans les paramètres de v_λ .

Propriété 2. La distance de voisins est une autre semi-distance (cette fois on retire l'inégalité triangulaire) :

- $v_\lambda(x, x) = 0$
- $v_\lambda(x, y) \geq 0$
- $v_\lambda(x, y) = v_\lambda(y, x)$
- $v_\lambda(x, y) = 0 \Rightarrow x = y$

La dernière propriété vient par construction : $v_\lambda(x, y) \geq 1$ sauf dans le cas particulier $x = y$ où elle est définie comme nulle.

L'inégalité triangulaire de v_λ serait vérifiée uniquement pour tous les triplets de points (x, y, z) voisins deux à deux, et n'aurait donc pas d'utilisation pratique. Cependant, on verra plus bas que cette propriété n'est pas essentielle pour construire la distance d'amibe.

Étant donnée cette distance définie entre tous les voisins, on peut maintenant définir la longueur d'un chemin reliant x et y comme étant la somme des distances de voisins le long du chemin :

Définition 3. Soit λ un réel positif et soit $\sigma = (x = x_0, x_1, \dots, x_n = y)$ un chemin entre les points x et y . Alors, on définit la longueur $L_\lambda(\sigma)$ du chemin σ par :

$$L_\lambda(\sigma) = \sum_{i=0}^{n-1} v_\lambda(x_i, x_{i+1})$$

On pose par convention que pour tout x la longueur du chemin (x, x) est nulle.

Il ne reste plus qu'à définir la distance d'amibe entre x et y comme la longueur du plus court chemin reliant x à y au sens de L_λ .

Définition 4. Soit λ un réel positif, on définit la «distance d'amibe» de paramètre λ comme :

$$d_\lambda(x, y) = \min_{\sigma} L_\lambda(\sigma)$$

Cette distance est définie de façon récursive, d'abord entre les voisins immédiats (via v_λ) et ensuite de proche en proche selon un chemin de coût minimal.

Vérifions que d_λ vérifie les propriétés d'une distance :

Propriété 3. La distance d_λ définie ci-dessus est une distance :

- $d_\lambda(x, y) \geq 0$
- $d_\lambda(x, x) = 0$
- $d_\lambda(x, y) = d_\lambda(y, x)$
- $d_\lambda(x, y) \leq d_\lambda(x, z) + d_\lambda(z, y)$
- $v_\lambda(x, y) = 0 \Rightarrow x = y$

Démonstration. La première propriété découle du fait que v_λ est positif, donc L_λ aussi.

La seconde découle de la construction de la distance : le chemin $\sigma = (x, x)$ a une longueur nulle (et les longueurs sont positives).

La symétrie découle de la symétrie de v_λ dans le cas où x et y sont voisins, et de la réversibilité du plus court chemin de x à y , qui est aussi le plus court chemin de y à x .

Enfin, $d_\lambda(x, y) \leq d_\lambda(x, z) + d_\lambda(z, y)$ car (x, \dots, z, \dots, y) est un chemin menant de x à y , de longueur $d_\lambda(x, z) + d_\lambda(z, y)$. C'est forcément le cas car sinon cela contredirait $d_\lambda(x, z)$ (resp. $d_\lambda(y, z)$) longueur du plus court chemin de x à z (resp. de y à z). Or $d_\lambda(x, y)$ est la longueur du plus court chemin de x à y , d'où le résultat. \square

Finalement, on peut définir l'amibe² par :

Définition 5. Soit λ un réel positif. Pour tout point x on définit l'amibe de paramètre λ et de rayon r comme la boule fermée de rayon r associée à la distance d_λ :

$$Amibe_\lambda(x, r) = \{y / d_\lambda(x, y) \leq r\}$$

²Le choix de l'écart (distance entre les valeurs des pixels) est aussi un paramètre de l'amibe, mais on le change moins souvent, donc on ne l'inclut pas dans la notation.

4.5 Propriétés

De par leur construction, les amibes possèdent de nombreuses propriétés intéressantes. On se limitera ici aux amibes définies plus haut, mais il est évident qu'il est possible de reprendre l'approche décrite ici avec d'autres paramétrages ou d'autres types de distance.

4.5.1 Propriétés ensemblistes

Propriété 4. *Réflexivité* : $x \in \text{Amibe}(x)$

Démonstration. Par construction : $d_\lambda(x, x) = 0$. □

Propriété 5. *Symétrie* : $x \in \text{Amibe}_\lambda(y) \Leftrightarrow y \in \text{Amibe}_\lambda(x)$

Démonstration. La preuve est triviale en se basant sur la symétrie de la distance d_λ et en inversant le chemin de x à y . □

Nous verrons plus bas que grâce à cette dernière propriété, les érosions et dilatations construites sur les amibes sont des opérateurs duaux et adjoints. Cette distance offre également une intéressante propriété d'inclusion sur les λ :

Propriété 6. À rayon r fixé, la famille des amibes Amibe_λ relative à la famille de distances d_λ est décroissante au sens de l'inclusion :

$$\begin{aligned} 0 \leq \lambda_1 \leq \lambda_2 &\Rightarrow \forall(x, y), d_{\lambda_1}(x, y) \leq d_{\lambda_2}(x, y) \\ &\Rightarrow \forall r \in \mathbf{R}^+, \text{Amibe}_{\lambda_1, r} \supset \text{Amibe}_{\lambda_2, r} \end{aligned}$$

Démonstration. Soient x et y deux points voisins, comme $\lambda_1 \leq \lambda_2$ et $e(x, y) \geq 0$, alors

$$v_{\lambda_1}(x, y) \leq v_{\lambda_2}(x, y)$$

Soient x et y deux points quelconques. Pour tout chemin σ entre x et y , on a :

$$L_{\lambda_1}(\sigma) \leq L_{\lambda_2}(\sigma)$$

Si σ_2 est le plus court chemin au sens de d_{λ_2} , la propriété devient en particulier $L_{\lambda_1}(\sigma_2) \leq L_{\lambda_2}(\sigma_2) = d_{\lambda_2}(x, y)$. Avec σ_1 le plus court chemin au sens de d_{λ_1} on a par définition $L_{\lambda_1}(\sigma_1) \leq L_{\lambda_1}(\sigma_2)$ et donc finalement :

$$d_{\lambda_1}(x, y) \leq d_{\lambda_2}(x, y)$$

□

Dans le cas des éléments structurants classiques, on construit une famille de filtres en augmentant le rayon de l'élément structurant. Dans le cas des amibes, ceci est faisable, mais la propriété ci-dessus ouvre la possibilité supplémentaire de définir des famille de filtres ayant le même rayon r mais des λ de plus en plus petits. Les formes de ces amibes seraient assez différentes de celles obtenues en fixant λ et en augmentant le rayon.

Par ailleurs, le cas $\lambda = 0$ est particulièrement intéressant, car la distance d'amibe se réduit à la distance L_1 (en 4-connexité) ou L_∞ (en 8-connexité), etc. L'amibe ainsi définie sera exactement la boule associée à L_1 (un losange), à L_∞ (un carré), etc. L'application de la propriété précédente au cas $\lambda = 0$ nous donne la propriété suivante.

Propriété 7. *L'amibe de rayon r est incluse dans la boule de rayon r relative à la norme correspondant à la connexité choisie pour le cheminement.*

Ceci prouve en particulier que l'extension maximale de l'amibe est bornée et permet de majorer le volume de l'amibe, ce qui sera utile plus bas dans l'étude de la complexité théorique des amibes. De plus, avoir une borne sur la taille de l'amibe permet des implémentations plus performantes.

4.6 Amibes et morphologie mathématique

De par leur capacité à changer de forme, les amibes nécessitent de prendre des précautions avant de les utiliser pour définir des opérateurs morphologiques. Du point de vue formel, l'essentiel du travail de théorie a été défriché par Serra dans [Serra, 1988b] avec son étude des fonctions structurantes, ainsi que par Heijmans dans [Heijmans, 1994].

4.6.1 Rappel de définitions

Définition 6. *Les transformations δ et ϵ sont duales si pour tout X et son complémentaire \bar{X}*

$$\delta(X) = \overline{\epsilon(\bar{X})}$$

Définition 7. *Les transformations δ et ϵ sont adjointes si pour tout X et Y*

$$\delta(X) \subset Y \Leftrightarrow X \subset \epsilon(Y)$$

4.6.2 L'adjonction

Historiquement [Matheron, 1967], les premières définitions de la dilatation et de l'érosion par une boule B étaient les suivantes :

$$\begin{aligned} \delta(X) &= \{x/B_x \cap X \neq \emptyset\} \\ \epsilon(X) &= \{x/B_x \subset X\} \end{aligned}$$

Ce qui donne dans le cas des amibes :

$$\begin{aligned} \delta(X) &= \{x/Amibe(x) \cap X \neq \emptyset\} \\ \epsilon(X) &= \{x/Amibe(x) \subset X\} \end{aligned}$$

Ces deux opérateurs sont duaux : $\delta(\bar{X}) = \overline{\epsilon(X)}$. Informatiquement, ils correspondent en niveaux de gris à la recherche en chaque point du minimum (resp. le maximum) sur le voisinage de ce point et à affecter cette valeur dans l'image de sortie en ce point.

Le problème est apparu lorsqu'on a cherché à les composer pour former des ouvertures et des fermetures. Ces opérations ne sont pas adjointes pour des éléments structurants non symétriques et les «ouvertures» (resp. fermetures) construites en les combinant ne sont pas idempotentes.

Pour contourner cette difficulté et conserver cette formulation duale, on a décidé d'appliquer l'une des opérations sur le *transposé* de l'élément structurant. En tout point x on définit le transposé \check{B}_x de la boule B_x centrée en x par :

$$y \in B_x \Leftrightarrow x \in \check{B}_y$$

Pour faire une ouverture, on applique ainsi à l'image l'érosion par B suivie de la dilatation non pas par B mais par \check{B} . Pour des éléments structurants invariants par translation, il suffit de prendre pour

Algorithme 4 Érosion par une boule B (niveaux de gris)

pour tout pixel x dans l'image originale **faire**
 calculer le minimum m des pixels dans B_x
 valeur(x)= m
fin pour

Algorithme 5 Dilatation duale par une boule B (niveaux de gris)

pour tout pixel x dans l'image originale **faire**
 calculer le maximum M des pixels dans B_x
 valeur(x)= M
fin pour

\tilde{B} le symétrique par rapport au centre. Cette méthode est simple et efficace, et dans le cas d'éléments symétriques, il n'y a rien à faire.

Les algorithmes 4 et 5 présentent l'érosion et la dilatation duales. L'algorithme 6 présente l'utilisation de la dilatation duale avec la boule transposée pour construire la dilatation adjointe.

Une autre formulation possible de la dilatation est de la définir comme étant l'union des boules dont les centres sont dans X :

$$\delta_I(X) = \bigcup_{x \in X} B_x$$

Cette formulation est directement l'adjointe de l'érosion précédente et ne nécessite pas de passer par une transposition. L'algorithme 7 décrit l'implémentation de cette dilatation.

Cette seconde formulation a l'avantage d'être plus proche à relier à l'approche géométrique quand on calcule une ouverture (figure 4.10). De façon formelle, l'ouvert d'un ensemble X par une boule B est l'union des boules B incluses dans X . À rayon donné, une boule est intégralement décrite par son centre, il suffit donc de trouver les centres des boules incluses dans X . Cette opération est l'érosion. Une fois ces centres déterminés, il suffit de reconstruire les boules à partir de leurs centres. Il s'agit exactement de la dilatation décrite plus haut.

Le lecteur intéressé se référera à [Serra, 1988b] pour une discussion plus complète sur les diverses formes d'adjonctions et de transpositions de fonctions structurantes.

4.6.3 Amibes, dualité, adjonction et formalisme

Les fondements théoriques des opérateurs morphologiques de base ont été étudiés notamment par Heijmans qui montre dans [Heijmans, 1994] que si deux opérateurs sont adjoints, alors l'un définit

Algorithme 6 Dilatation adjointe par une boule B (niveaux de gris)

\tilde{B} =transposé(B)
pour tout pixel x dans l'image originale **faire**
 calculer le maximum M des pixels dans \tilde{B}_x
 valeur(x)= M
fin pour

Algorithme 7 Dilatation adjointe par une boule B (niveaux de gris)

pour tout pixel x dans l'image originale **faire**
 pour tout pixel y dans B_x **faire**
 valeur(y)= $\max(\text{valeur}(y), \text{valeur}(x))$
fin pour
fin pour

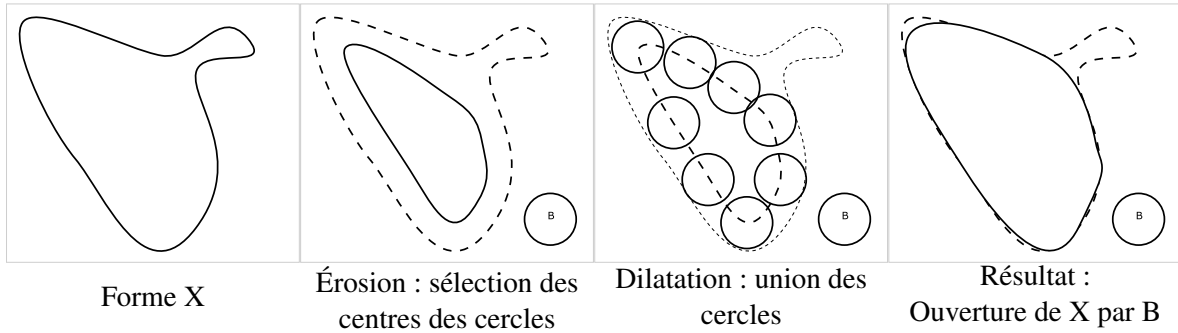


FIG. 4.10: L'ouverture d'une forme X par une boule B est l'union des boules contenues dans cette forme. On peut décomposer cette opération en une étape d'érosion qui sélectionne les centres et une dilatation qui reconstruit les boules à partir des centres.

une érosion et l'autre une dilatation. On peut construire à partir de là des ouvertures, des fermetures, des combinaisons de ces opérations, etc.

On définit deux opérateurs δ et ϵ ci-dessous, et on va montrer qu'ils sont adjoints et que donc ils forment un couple érosion/dilatation :

$$\delta(X) = \bigcup_{x \in X} \text{Amibe}(x)$$

$$\epsilon(X) = \{x / \text{Amibe}(x) \subset X\}$$

Propriété 8. Adjonction : ϵ et δ sont adjoints :

$$\delta(X) \subset Y \Leftrightarrow X \subset \epsilon(Y)$$

Démonstration. Supposons $\delta(X) \subset Y$:

$$\begin{aligned} x \in X &\Rightarrow \text{Amibe}(x) \subset \delta(X) \\ &\Rightarrow \text{Amibe}(x) \subset Y \\ &\Rightarrow x \in \epsilon(Y) \end{aligned}$$

Donc $\delta(X) \subset Y \Rightarrow X \subset \epsilon(Y)$

A présent supposons $X \subset \epsilon(y)$:

$$\begin{aligned} X \subset \epsilon(y) &\Rightarrow \forall x \in X, \text{Amibe}(x) \subset Y \\ &\Rightarrow \bigcup_{x \in X} \text{Amibe}(x) \subset Y \\ &\Rightarrow \delta(X) \subset Y \end{aligned}$$

Donc $X \subset \epsilon(Y) \Rightarrow \delta(X) \subset Y$, et on a bien l'équivalence. □

Par ailleurs, la propriété de symétrie des amibes entraîne que ces deux opérateurs (qui forment donc maintenant une dilatation et une érosion) sont non seulement adjoints, mais également duaux :

Propriété 9. *Dualité* : ϵ et δ sont duaux : $\epsilon(X) = \overline{\delta(\overline{X})}$

Démonstration.

$$\begin{aligned}
 \epsilon(X) &= \{x/Amibe(x) \subset X\} \\
 \epsilon(\overline{X}) &= \{x/Amibe(x) \subset \overline{X}\} \\
 \overline{\epsilon(\overline{X})} &= \{x/Amibe(x) \cap X = \emptyset\} \\
 \overline{\epsilon(\overline{X})} &= \{x/\exists y \in Amibe(x) \cap X \neq \emptyset\} \\
 &= \{x/\exists y \in Amibe(x), y \in X\} \text{ et par symétrie de l'amibe :} \\
 &= \{x/\exists y/x \in Amibe(y), y \in X\} \\
 &= \bigcup_{y \in X} Amibe(y) = \delta(X)
 \end{aligned}$$

□

Cette propriété nous montre que dans le cas des amibes (comme pour tout élément structurant étant son propre transposé) les deux algorithmes de dilatation décrits plus haut sont équivalents. On peut donc utiliser aussi bien l'un que l'autre, selon les propriétés que l'on recherche.

L'extension du domaine binaire au domaine en niveaux de gris se fait par la méthode traditionnelle des seuils explicitée dans [Serra, 1988a].

Un point important est à noter : telle qu'on l'a exprimée jusqu'à présent la forme de l'amibe au point x n'est a priori pas la même avant et après une opération sur l'image. Or les propriétés énoncées ci-dessus supposent que $Amibe(x)$ définit le même ensemble à chaque moment. Ainsi, il faut choisir une méthode pour garantir que l'amibe décrira les mêmes régions lors de la dilatation que lors de l'érosion.

Une première possibilité est de calculer la forme des amibes dans une première passe et d'appliquer les transformations successives sur le même jeu d'amibes. Nous avons préféré une autre approche utilisant une image secondaire, qui nous offre par la même occasion plus de flexibilité dans la robustesse face au bruit. Cette approche utilisant une «image pilote» est décrite plus loin.

4.6.4 Algorithmes adjoints

Les algorithmes utilisés pour l'érosion et la dilatation sont similaires à ceux faisant intervenir des éléments structurants classiques, à l'exception de l'étape supplémentaire de calcul de la forme de l'amibe.

Algorithme 8 Érosion (niveaux de gris)

pour tout pixel x dans l'image originale **faire**
 calculer la forme de l'amibe centrée sur x
 calculer le minimum m des pixels dans l'amibe
 valeur(x)= m
fin pour

On a vu plus haut que les deux version de la dilatation étaient équivalentes. Le choix de l'une ou de l'autre dépend donc plus des spécificités de l'application. D'un côté, il est plus commode de ne

Algorithme 9 Dilatation adjointe (niveaux de gris)

```

pour tout pixel  $x$  dans l'image originale faire
  calculer la forme de l'amibe centrée sur  $x$ 
  pour tout pixel  $y$  dans l'amibe faire
    valeur( $y$ )=max(valeur( $y$ ),valeur( $x$ ))
  fin pour
fin pour

```

pas se préoccuper de la notion de transposition et de réutiliser intégralement l'algorithme 9 pour tout élément structurant qu'il soit invariant par translation ou non, symétrique ou non, etc. D'un autre côté, cette version a l'inconvénient d'aller écrire dans la mémoire à plusieurs endroits, alors que la version «chercher le maximum» n'écrit qu'au centre. Cette dernière est donc probablement mieux adaptée à des implémentations en parallèle ou directement matérielles.

4.6.5 Décomposition

Avec des éléments structurant classiques la dilatation et l'érosion se prêtent bien à une approche en plusieurs passes.

Par exemple, il est classique d'utiliser n dilatations avec un élément de rayon 1 plutôt qu'une seule dilatation avec un élément de rayon n car la complexité de l'un croît linéairement en $O(n)$, alors que l'autre croît en $O(n^d)$ où d est la dimension de l'image (2 ou 3 en général). Des approches plus complexes permettent par ailleurs de diminuer encore les coûts des opérations (voir [Pecht, 1985] pour une complexité en $O(\log(n))$, [van Herk, 1992] pour du $O(1)$ sur des segments et plus récemment [Van Droogenbroeck and Buckley, 2005]). Pouvoir décomposer des amibes en éléments simples et rapides permettrait ainsi des implémentations efficaces et rapides.

Avec des amibes, si $r = r_1 + r_2$, obtient-on le même résultat en faisant un filtrage de taille r_1 puis un filtrage de taille r_2 sur le résultat qu'en faisant directement un filtrage de taille $r_1 + r_2$?. Malheureusement, dans le domaine discret ce n'est pas le cas car on est obligé de s'arrêter sur des pixels aux positions entières.

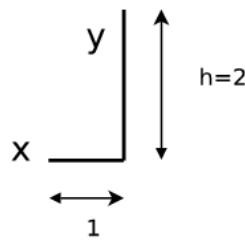


FIG. 4.11: Situation non décomposable pour certaines amibes

Dans la figure 4.11, en prenant $\lambda = 1$, on a :

$$d_\lambda(x, y) = \underbrace{1}_{\text{dist.geom.}} + \underbrace{1}_\lambda \cdot \underbrace{2}_{\text{dist.couleur}} = 3$$

donc $Amibe(x, 3) = \{x, y\}$

Peut-on utiliser deux amibes, une de taille 1 suivie d'une de taille 2 et obtenir le même résultat ?

Posons $A_1 = Amibe(x, 1) = \{x\}$, alors pour le filtrage par la seconde amibe on prendrait

$$\bigcup_{y \in A_1} Amibe(y, 2) = Amibe(x, 2) = \{x\} \neq \{x, y\}$$

Dans notre exemple, les amibes décomposées ne voient pas les mêmes points que l'amibe «complète», et donc les résultats ne seront pas les mêmes.

On doit se contenter d'une inclusion :

Propriété 10. *Pour tout λ , on a :*

$$Amibe_\lambda(x, r_1 + r_2) \supset \bigcup_{y \in Amibe_\lambda(x, r_1)} Amibe_\lambda(y, r_2)$$

Démonstration. C'est une application de l'inégalité triangulaire. Pour tout $y \in Amibe(x, r_1)$ on a $d_\lambda(x, y) \leq r_1$. Ensuite, pour tout $z \in Amibe(y, r_2)$, on a $d_\lambda(y, z) \leq r_2$. Donc

$$d_\lambda(x, z) \leq d_\lambda(x, y) + d_\lambda(y, z) \leq r_1 + r_2$$

Donc $z \in Amibe(x, r_1 + r_2)$. □

4.7 Les amibes en pratique

Les sections précédentes ont détaillé la construction des amibes et certaines de leurs propriétés. Cette partie s'intéressera de plus près à la mise pratique des amibes (certains détails plus techniques sont exposés dans l'annexe A).

4.7.1 L'image pilote

Le principe de l'image pilote est de séparer l'image sur laquelle la forme de l'amibe est calculée de celle sur laquelle les valeurs sont échantillonnées et fournies à l'opérateur de filtrage (moyenne, médiane, min, max, ...). Ceci permet de pallier deux problèmes majeurs :

Conserver adjonction et idempotence

Avec les formulations simples données plus haut, il est probable que la forme d'une amibe en un point donnée change après l'application d'un opérateur. Ceci fait perdre les propriétés d'adjonction et d'idempotence.

En conservant la même image pilote au cours des diverses opérations on garantit que la forme des amibes sera la même et donc que les propriétés théoriques démontrées plus haut restent vraies. En pratique, il suffit de conserver la même image pilote entre une passe d'érosion et la passe de dilatation correspondante (et réciproquement).

Limiter l'impact du bruit

L'image pilote la plus simple est une copie de l'image initiale. Mais nous avons constaté que le bruit dans l'image pouvait souvent déformer les amibes.

Ce problème a déjà été rencontré dans les approches analytiques de la diffusion anisotropes et la littérature propose diverses approches pour le résoudre, par exemple dans [Catté, 1992] qui propose d'inclure un facteur de lissage dans la formulation même du gradient.

Cependant, cette approche directe n'est pas une option viable pour les algorithmes morphologiques car nous avons besoin de conserver l'image pilote de toute façon.

Nous choisissons donc pour l'image pilote le résultat d'un fort filtrage de l'image originale, avec comme objectif de retirer un maximum de bruit, tout en essayant de préserver au mieux les contours les plus marquants. Un filtrage par une gaussienne de belle taille fonctionne relativement bien, mais nous verrons plus bas qu'une itération d'amibes fonctionne encore mieux.

4.7.2 Complexité

La complexité théorique d'un filtre simple basé sur des amibes (érosion, dilatation, moyenne, médiane, etc) peut être approximée asymptotiquement par

$$T(n, k, op) = O \left[n * \left(op(k^d) + amibe(k, d) \right) \right]$$

où n est le nombre de pixels de l'image, d la dimension de celle-ci (souvent 2, parfois 3), k est le rayon maximal de l'amibe (voir 7), $op(k^d)$ est le coût de l'opérateur et $amibe(k, d)$ est le coût de calcul de la forme de l'amibe pour un pixel donné.

La forme de l'amibe est obtenue par un algorithme classique de croissance de région, implémenté à l'aide d'une file d'attente hiérarchique. Selon le type de file utilisée, la complexité de cette opération est légèrement supérieure à $O(k^d)$ (donc linéaire en le nombre de points de l'amibe). De nombreux travaux (voir en particulier [Cherkassky and A. V. Goldberg, 1996] et [Brodnik et al., 2001]) visent à se rapprocher d'un coût réellement linéaire, mais dépassent le cadre de notre propos car ils peuvent aller jusqu'à nécessiter une architecture matérielle dédiée.

Ainsi, pour les érosions, dilatations, moyennes, etc, nous avons une complexité à peine supérieure à $O(n * k^d)$, qui est la complexité *théorique* d'un filtre sur un noyau de forme fixe. Bien sûr, la plupart des filtres classiques sont soit séparables, soit décomposables, et en tout cas facilement optimisables et bien plus rapides que l'expression de leur complexité ne le laisserait sous-entendre.

En pratique, nous avons vérifié qu'un filtrage par amibe avait effectivement un temps de calcul maîtrisé, bien que nettement plus long que son équivalent sur un noyau de forme fixe. À titre indicatif, les tests ont porté sur des image 3D de taille 512x512x100, avec des amibes d'une taille allant jusqu'à 21x21x21.

4.8 Résultats

Nous avons implémenté trois des opérateurs de débruitage les plus utilisés : le médian, la moyenne et les filtres alternés séquentiels morphologiques. Pour illustrer les résultats, nous allons comparer ces trois opérations entre elles et également à leurs équivalents sur des éléments structurant classiques à forme fixe.

4.8.1 Le Fivre

La figure 4.12 montre les résultats sur une image modérément contrastée et souffrant d'un bruit proche du poivre-et-sel.

Le filtre médian préserve bien les contours mais tend à aplatir les beaucoup les régions. La moyenne conserve moins bien la dynamique de certains gradients mais donne un résultat plus agréable à l'oeil. Dans les deux cas, le résultat est bien supérieur à une médiane ou à une moyenne construites avec des éléments structurants «rigides».

Le filtrage alterné séquentiel (FAS) a tendance à créer de grandes zones plates, comme la médiane, mais présente une réponse beaucoup moins nuancée. Certains détails comme l'oreille droite sont éliminés car ils correspondent à un niveau de gris intermédiaire.

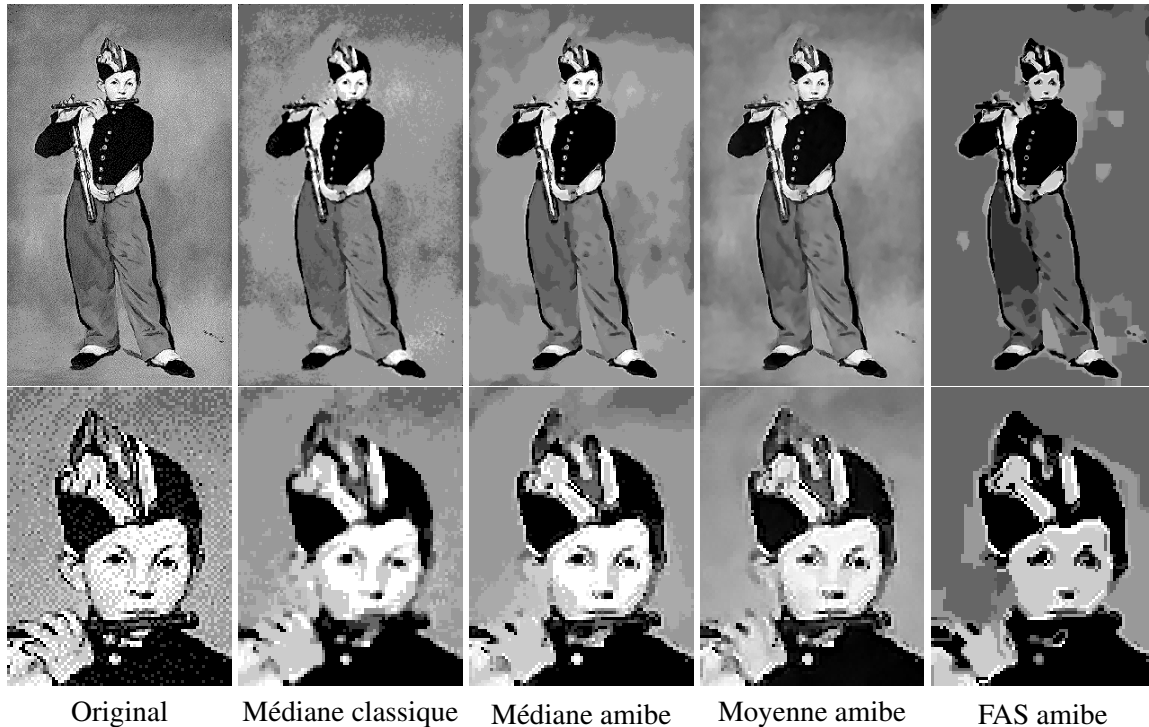


FIG. 4.12: Résultats d'un filtrage médian classique, et de deux filtrages basés sur des amibes : une médiane et une moyenne (*Le Joueur de Fife*, E. Manet, 1866).

4.8.2 Vaisseaux réiniens

Les images de la figure 4.13 comparent les résultats sur une image présentant clairement deux populations de points : les vaisseaux très sombres, et le fond de l'oeil plus clairs. Le bruit présent est plus gaussien que dans la série précédente. L'objectif du filtrage est de réduire le bruit de fond tout en préservant les vaisseaux.

On voit que les FAS construits sur des amibes préservent très bien les vaisseaux sanguins tout en lissant fortement le fond de la rétine. La moyenne ou les médianes (petite amibe et grande amibe) ont tendance à épaissir les vaisseaux.

Dans les deux cas, les opérateurs définis sur des amibes éliminent bien le bruit tout en préservant presque parfaitement les contours des vaisseaux, au contraire des filtres classiques.

Il est intéressant de constater que sur les deux images (*Fife* et *Rétine*), les meilleurs résultats ne sont pas obtenus par les mêmes filtres. Ainsi, on préférera une moyenne ou une médiane pour filtrer une image sans dominante particulière, alors qu'un filtre alterné séquentiel sera plus efficace pour séparer les composantes claires et sombres dans une autre image.

4.8.3 Considération sur les filtres alternés séquentiels

Une caractéristique majeure des filtres alternés séquentiels est qu'ils privilégient les valeurs extrêmes. On a vu dans le cas du *Fife* que cela pouvait nuire aux résultats. En revanche, c'est une qualité

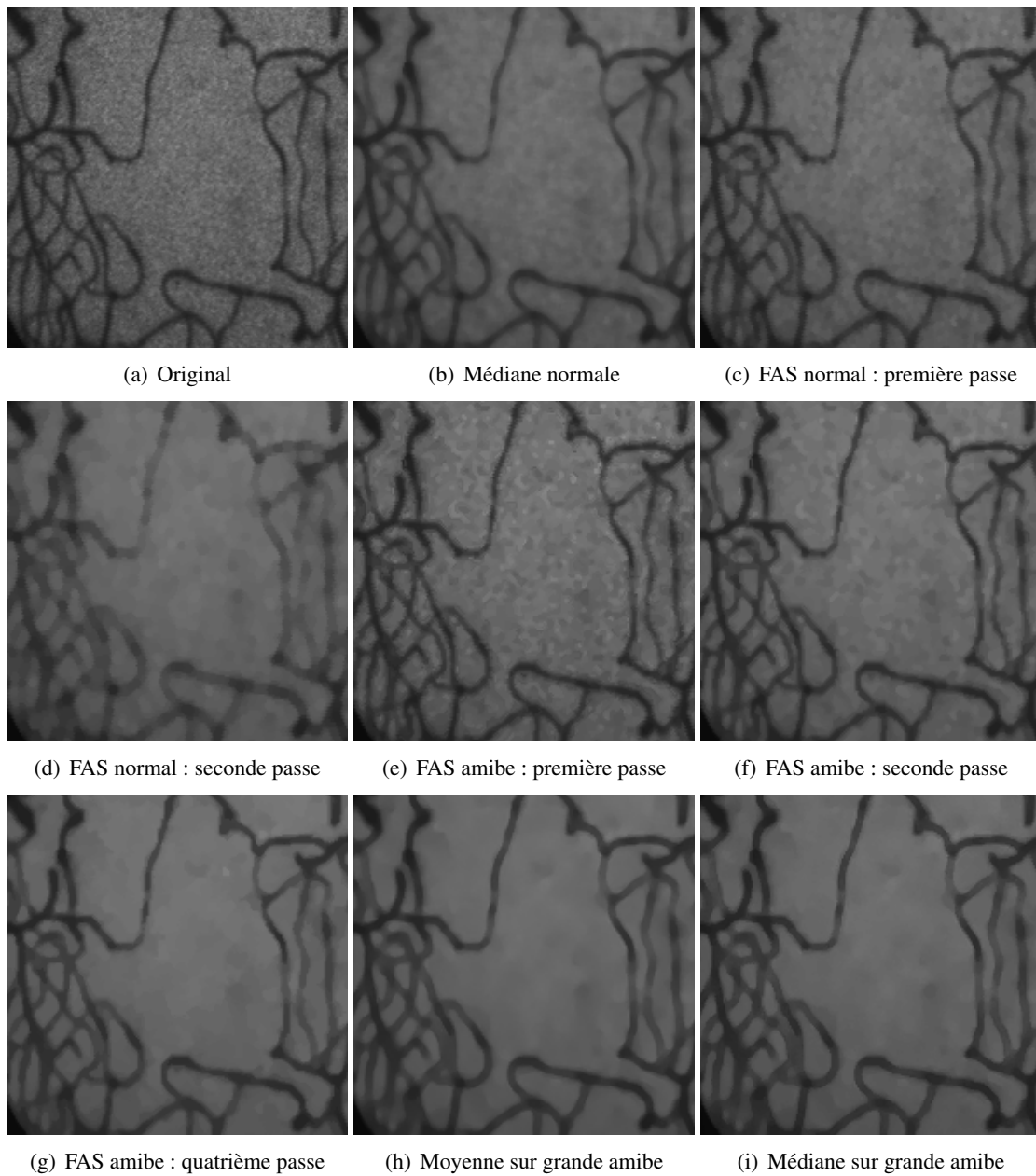


FIG. 4.13: Médianes, moyennes et filtres alternés séquentiels (FAS) sur des noyaux classiques et sur des amibes (Source : CMM).

intéressante dans le cas d'images aux valeurs bien séparés comme les vaisseaux rétinien.

Une amélioration possible pour mitiger ce comportement extrême consisterait à utiliser des filtres de rang d'ordre inférieur à la place de l'érosion et de la dilatation. En effet, on peut considérer que même si l'amibe se trouve essentiellement dans un objet, il est possible qu'elle ait assez d'énergie pour faire des incursions dans un objet voisin. Dans ce cas, l'érosion ou la dilatation liront les valeurs de cet objet parasite et les diffuseront dans l'objet principal. En revanche, en utilisant par exemple un filtre de rang prenant le quantile à 75% à la place d'une dilatation classique (qui prend le maximum) on permettrait ainsi à l'amibe d'empiéter sur un objet plus lumineux jusqu'à 25% de sa surface avant que cela n'influe excessivement sur le résultat.

4.8.4 Extension à la couleur

Moyenne et médiane pour les images couleur

Dans le cas des images couleur, la moyenne est remplacée par la moyenne selon chaque composante dans l'espace RGB. Pour la «médiane», on choisit le point le plus proche du barycentre.

Le choix des espaces couleurs et celui des distances a un léger effet sur la qualité du résultat. Cependant, l'effet le plus remarquable est le choix de l'image «pilote». Dans le cas présent, on a utilisé un filtre gaussien séparément sur chaque composante R,G et B et recombina les trois canaux. Nous verrons plus bas comment améliorer encore les résultats en utilisant une meilleure image pilote.

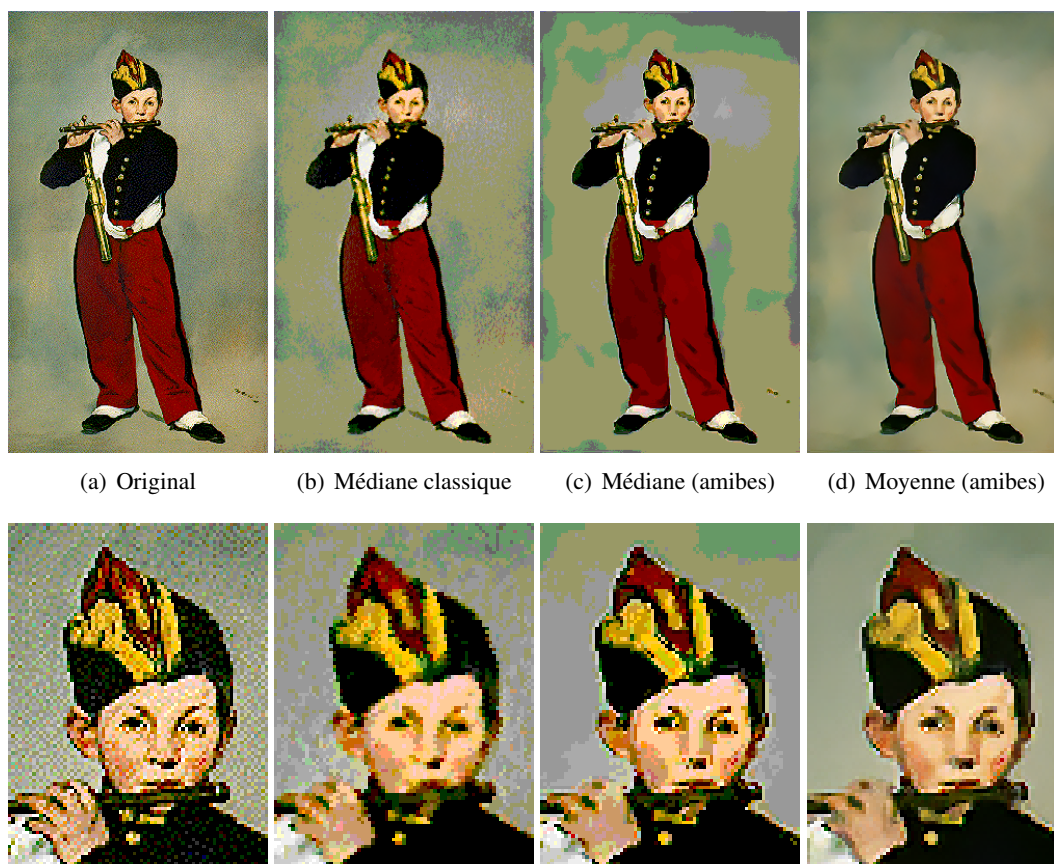


FIG. 4.14: Résultats d'un filtrage médian classique, et de deux filtrages basés sur des amibes : une médiane et une moyenne.

4.8.5 Images 3D

Comme souvent en morphologie mathématique, le formalisme employé est indépendant de la dimension des objets sur lesquels on l'applique, donc il n'y a aucun travail supplémentaire à faire pour utiliser les amibes sur des images 3D (voir figures 4.15 et 4.16).

Les résultats présentés ici ont été obtenus sur le volume de synthèse présenté plus haut, et montrent l'efficacité des amibes dans ce type de filtrage. Bien que cela soit difficile à voir, les contours sont bien mieux préservés que dans les images de la figure 4.5. Il est à noter que le filtre conserve non seulement la position mais également la valeur des contours, ce qui est critique dans une application médicale, car dans ce type d'images, la valeur des voxels code le type de tissu. Un décalage de valeur peut donc engendrer une mauvaise interprétation clinique de l'image.

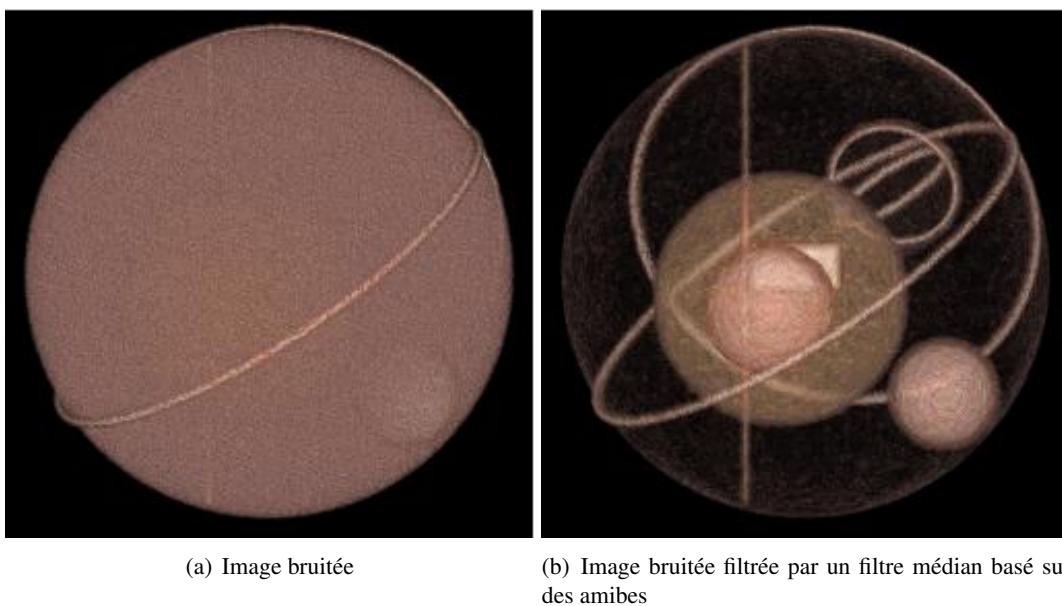
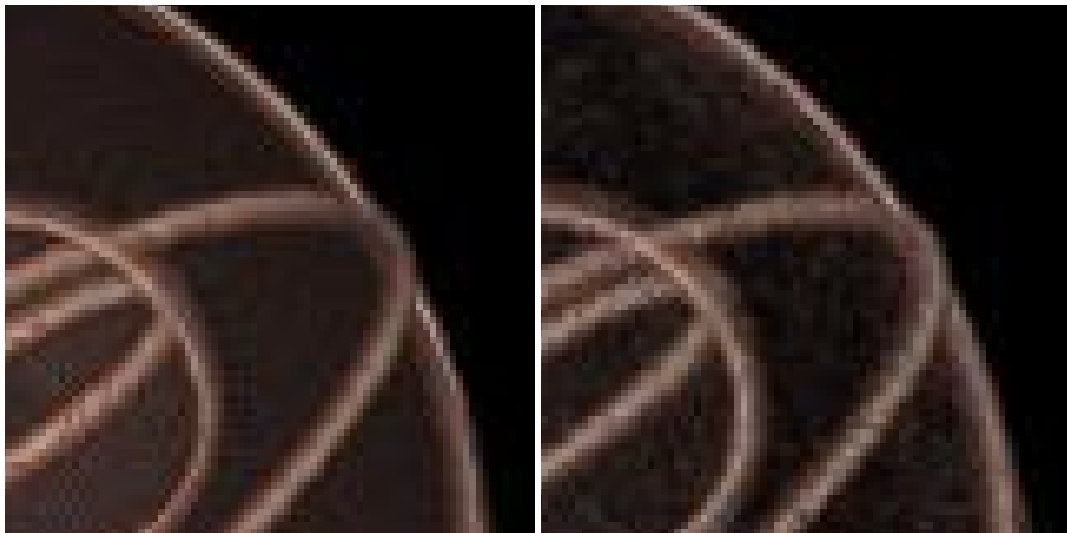


FIG. 4.15: Le filtre médian calculé sur des amibes recouvre l'essentiel de l'information, y compris les détails fins des contours. On a également un effet de transparence presque parfait, grâce au filtrage efficace des gradients.

Il existe de nombreuses méthodes pour quantifier des résultats, en particulier quand on dispose de l'image originale. Cependant, peu de ces mesures sont réellement informatives. Par exemple, le PSNR (Peak Signal-to-Noise Ratio) utilisé fréquemment effectue une intégration des différences sur toute l'image. Il est donc possible (et fréquent !) qu'un couple d'images identiques presque partout mais différant dans certaines zones critiques soit préféré à un couple d'images globalement moins similaires mais aussi moins éloignées dans les zones critiques.

Dans le cas qui nous occupe, les zones critiques sont les zones de contour : les méthodes que nous comparons ont toutes de bons résultats à l'intérieur des objets, mais diffèrent grandement dans leur traitement des frontières des objets. Ces caractéristiques uniques nécessitent une des outils d'analyse spécifiques.

Une mesure exprimant précisément ce besoin a été développée par Mathilde Boehm [Boehm, 2004]. Cette mesure utilise un afficheur de volumes 3D par lancer de rayons (ray-caster) modifié pour renvoyer la contribution de chaque voxel au pixel final. Quand un voxel de l'image filtrée contribue de façon identique à celui de l'image idéale, la mesure vaut 1. Inversement, plus la différence est grande entre les deux contributions, plus la mesure sera proche de zéro. Cette mesure est intégrée sur l'image



(a) Image originale (zoomée)

(b) Image filtrée par un médian calculé sur des amibes (zoomée)

FIG. 4.16: Le filtre médian sur amibes retrouve bien les détails des contours, avec un bon effet de transparence.

entière, ainsi que sur les surfaces des objets présents dans l'image.

La figure 4.17 montre une comparaison entre un filtre médian défini sur une amibe et un filtre alterné séquentiel traditionnel. Les chiffres montrent deux éléments importants : le premier est que les résultats de la mesure intégrée sur l'image entière (colonne «Global») sont proches pour les deux filtres. Ceci confirme l'impression générale dégagée par les images qui semblent très semblables, du moins à première vue.

On voit par la suite que les résultats sont très proches sur les structures de taille moyenne (sphères B2, B3, B5 et P1) mais qu'en revanche, les amibes sont nettement plus efficaces sur les structures filaires (le parallélogramme vertical très allongé P2 et les multiples anneaux). Par ailleurs, il y a avec les FAS un décalage visible de la valeur de l'image, en particulier sur les anneaux externes, ce qui pourrait causer des erreurs de diagnostic.

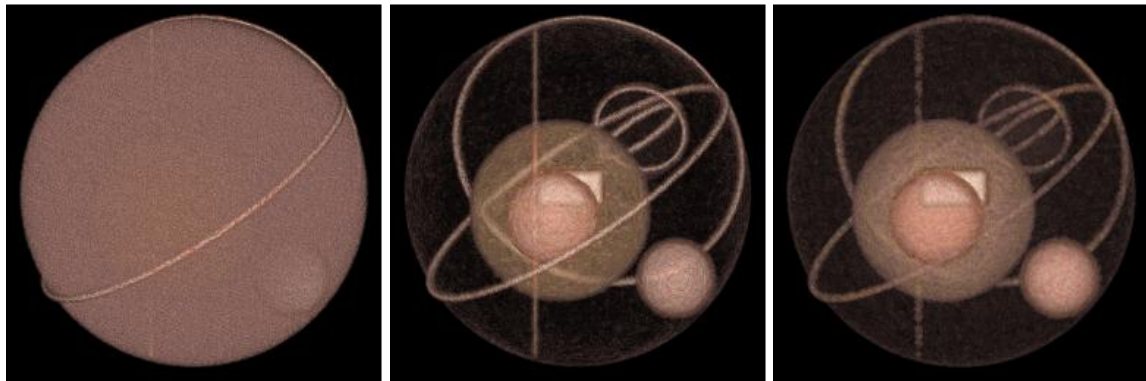
4.8.6 Itérations

La qualité du filtrage dépend en grosse partie de l'image pilotant la forme des éléments. Les exemples précédents ont utilisé l'image originale filtrée par une gaussienne, mais ceci ne donne pas toujours de bons résultats (voir également [Catté, 1992]).

Il est en effet fréquent qu'un petit détail de l'image soit excessivement lissé dans l'image pilote et donc disparaisse dans l'image filtrée. D'un autre côté, des pixels de bruit pourront être intégralement conservés si l'image pilote de les lisse pas. Une solution directe est en quelque sorte d'itérer le processus en se servant du résultat précédent comme image *pilote*.

La première itération suit le schéma décrit plus haut en utilisant une gaussienne comme pilote, avec un filtrage passe-bas sur amibes assez agressif. Le résultat est une image très lisse dans les zones plates mais qui préserve raisonnablement bien les contours.

Chaque étape successive utilise le résultat de l'image précédente comme pilote, mais en échantillonnant les valeurs sur l'image d'origine, avec des paramètres de moins en moins agressifs.

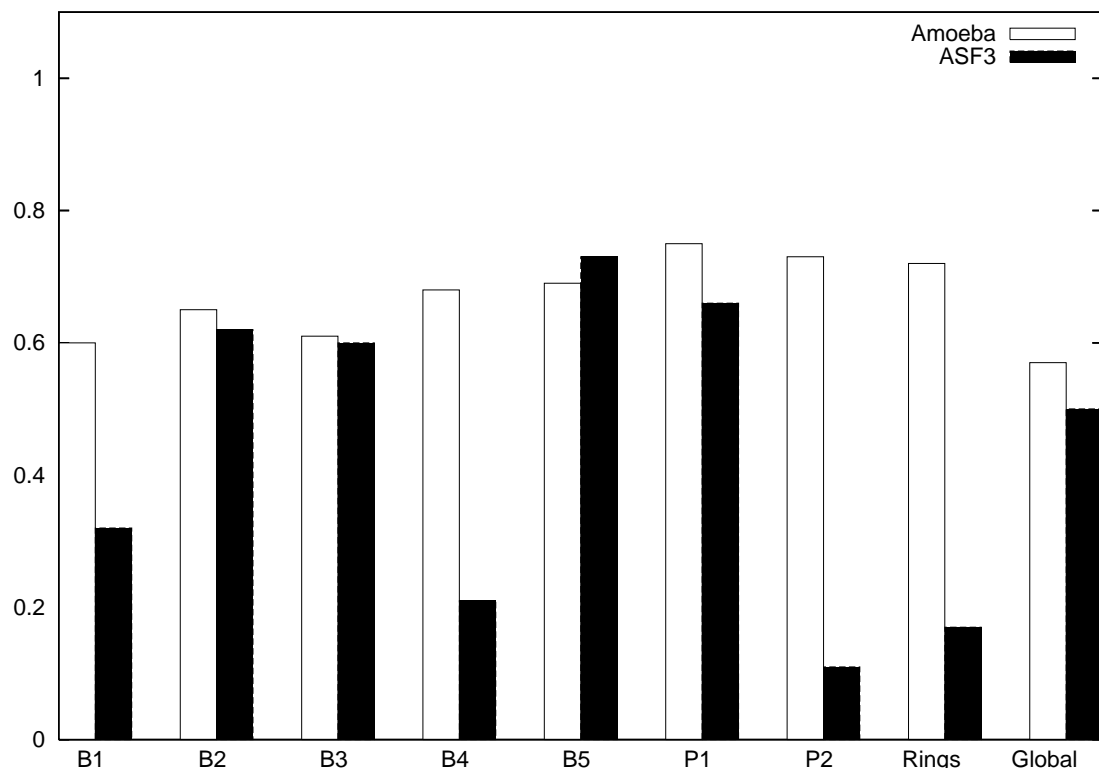


(a) Image bruitée

(b) Médian sur amibes

(c) FAS taille 3

Texture 50



(d) Mesure des résultats

FIG. 4.17: Comparaison des résultats entre un filtre médian défini sur des amibes et un filtre alterné séquentiel (FAS) de taille 3. La mesure est effectuée sur chacune des composantes de l'image (sphères B1-B5, parallélépipèdes P1 et P2, anneaux R, et moyenne totale).

En pratique, nous avons constaté que deux itérations suffisent : la première passe de lissage suivie d'une seconde passe de filtrage normal. On peut voir sur la figure 4.19 que le bruit a été presque totalement éliminé mais que les détails les plus fins ont été préservés. Il est cependant probable que d'autres types d'images et de bruit nécessitent plusieurs itérations.

Les figures 4.18 et 4.19 présentent les deux étapes de ce processus. On notera tout particulièrement dans la figure 4.19 la façon dont la main gauche du joueur de fibre est préservée dans l'approche itérative, alors qu'elle est fortement dégradée dans l'approche directe. On se référera à la figure 4.18 pour vérifier que la gaussienne lie presque complètement la main au fond³.

Cette méthode est également utile pour les images couleur. En effet, l'image pilote obtenue par l'amibe offre un meilleur couplage des informations couleur que la simple fusion des résultats d'un filtrage gaussien indépendant sur chaque canal.

4.8.7 Comparaison avec la diffusion anisotrope

Nous comparerons ici les réponses entre un filtre «moyenne» basé sur les amibes à une diffusion anisotrope de type Weickert, qui est l'une des plus communément utilisée. Notre méthode d'évaluation consistera à bruite de façon croissante une image par un bruit gaussien, à appliquer les deux filtres et à mesurer l'écart entre l'image d'origine et l'image débruitée. Nous utiliserons le PSNR et l'erreur quadratique moyenne comme mesure de l'écart et nous comparerons le tout au temps d'exécution.

Rappel : La diffusion de Weickert

Weickert fait une analogie entre la diffusion et le processus physique d'équilibre de différences de concentrations avec conservation de masse, à l'aide de la loi de Fick (déplacement aléatoire d'un ensemble de particules des régions de haute concentration vers des régions de plus basse concentration). L'équation de diffusion s'écrit alors :

$$\begin{aligned} \frac{j}{\delta t} &= -w_m(\|\nabla f\|)\vec{\nabla} f \\ w_m(\|\nabla f\|) &= \begin{cases} 1 & \text{si } \|\nabla f\| = 0 \\ 1 - \exp\left(\frac{-C_m}{\left(\frac{\|\nabla f\|^2}{\lambda}\right)^m}\right) & \text{si } \|\nabla f\| > 0 \end{cases} \end{aligned} \quad (4.1)$$

où w_m est un paramètre nommé *tenseur de diffusion*, j correspond au flux de matière, et m et C_m sont reliés par la relation : $1 = \exp(-C_m)(1 + 2mC_m)$ (NB : si $m = 0$ on obtient alors une diffusion linéaire classique) (Source : M. Boehm)

Paramètres

Nous avons cherché sur l'image bruitée à $\sigma = 20$ les meilleurs paramètres pour chaque filtre en terme de qualité de résultat :

- Pour la diffusion de Weickert, nous ferons 50 itérations, et nous prendrons $m = 4$ (donc $c_m = 3.31$) avec un seuil $\lambda = 6$.

³Parmi les autres paramètres agissant sur ce résultat, on notera que la moyenne itérée est basée sur une distance dans l'espace LAB, alors que la gaussienne est calculée indépendamment sur les trois composantes dans RGB



FIG. 4.18: Utilisation d'une moyenne sur amibes pour obtenir une image grossièrement débruitée sans introduire un flou excessif. Noter la différence au niveau de la main et des sourcils.



FIG. 4.19: Comparaison entre la moyenne sur amibes pilotées par la gaussienne, et la moyenne sur amibes pilotées par l'itération précédente de la moyenne sur amibes. La main est très peu attaquée dans la dernière image, et les sourcils ne commencent pas à fusionner avec les yeux.

- Pour les amibes, nous commencerons par une gaussienne de taille 3 pour l'image pilote initiale. Ensuite, nous lisserons fortement le résultat par une amibe de taille 12, avec un paramètre λ de 2 et un rayon maximal de 3. Ceci nous définira une image pilote pour une seconde amibe de taille 8, de paramètre λ égal à 2 et de rayon maximal 2. Nous calculerons sur cette amibe la moyenne de l'image bruitée.

Résultats

La figure 4.20 présente les résultats pour des valeurs croissantes du bruit. On voit que les amibes préservent globalement bien la structure de l'image et ne lisent pas de façon excessive lorsque le bruit est faible. Par ailleurs, on voit de bonnes propriétés de préservation de l'image lorsque le bruit augmente.

La figure 4.21 illustre numériquement le bon comportement des amibes malgré l'augmentation du bruit.

En terme de temps de calcul, les amibes prennent de 32s à 42s et la diffusion anisotrope entre 58 et 61s. En diminuant le nombre d'itérations, on peut ramener celle-ci à une vitesse plus compétitive (25s pour 20 itérations), mais la qualité du résultat se dégrade. Au niveau des amibes, on peut arriver à des résultats corrects en utilisant directement une gaussienne (environ 10s) plutôt qu'en travaillant en deux passes. Les images résultantes sont alors plus bruitées.

Un point important à constater est que les images résultant du filtrage par amibes sont toujours d'une qualité relativement uniforme. En particulier, on ne voit pas un lissage excessif lorsque les paramètres ont été choisis pour un niveau de bruit supérieur.

4.9 Conclusions et perspectives

Les amibes sont un nouveau type d'élément structurant qui peut être utilisé de façon transparente dans de nombreux algorithmes morphologiques. En profitant du traitement de l'information locale fourni par les amibes, on peut construire des filtres plus robustes sur des images bruitées et qui se comportent en général de façon plus respectueuses de l'image que ceux construits sur des noyaux classiques.

De plus, les amibes morphologiques (tout comme leurs homonymes biologiques) sont extrêmement adaptables et peuvent être utilisées pour traiter des images couleur aussi bien qu'en niveaux de gris et de n'importe quelle dimension (2D, 3D, nD, ...). Selon l'application, on choisira soit un filtre visant à améliorer l'aspect visuel des images (moyenne), ou au contraire un filtre qui rend l'image plus facile à traiter par la suite (médian, FAS).

Il est parfaitement possible d'utiliser des amibes pour construire des opérateurs géodésiques, des filtres par reconstruction ou encore des inondations qui bénéficieraient de la capacité des amibes d'adapter leur forme au contenu des images. Cependant le comportement des amibes est beaucoup plus difficile à prendre en compte quand on les intègre dans des algorithmes complexes, ce qui peut causer des résultats corrects mais inattendus. Par exemple, les amibes ont souvent un rayon supérieur à 1, donc on prendra garde à ne pas identifier dilatation conditionnelle et dilatation géodésique, comme cela peut être fait avec des éléments structurants unitaires.

Comme toujours quand on adapte à la couleur une technique développée pour les images niveaux de gris, on garde une grande marge de manœuvre dans le choix des espaces et des distances couleurs. Le choix final étant en général dicté par l'application. Les résultats présentés ici ne sont que de simples extensions des algorithmes scalaires à l'espace RGB muni d'une distance euclidienne. L'utilisation

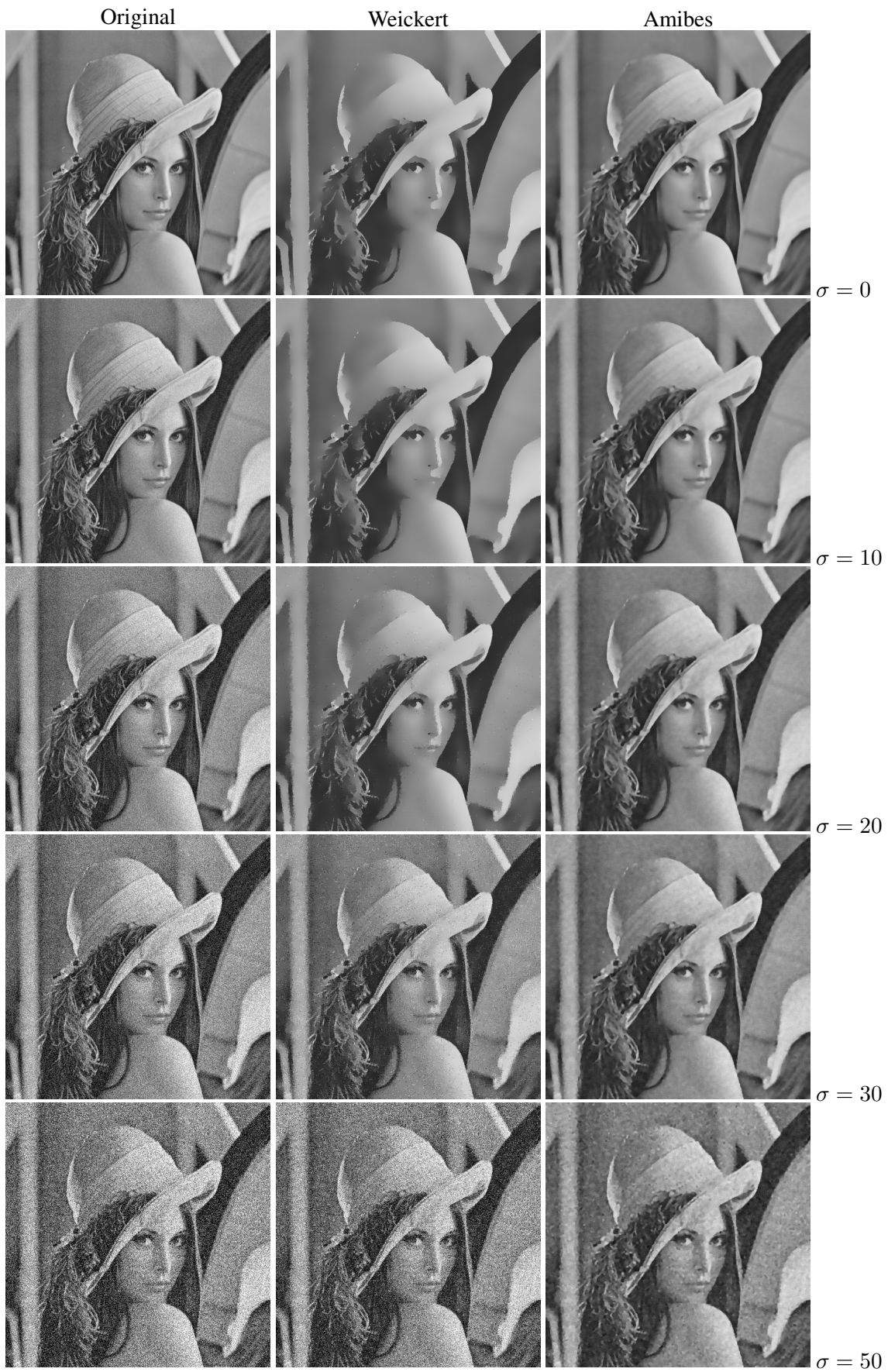


FIG. 4.20: Comparaison des résultats du filtrage par des amibes (droite) et par une diffusion anisotrope de Weickert (centre) quand on augmente le niveau du bruit gaussien.

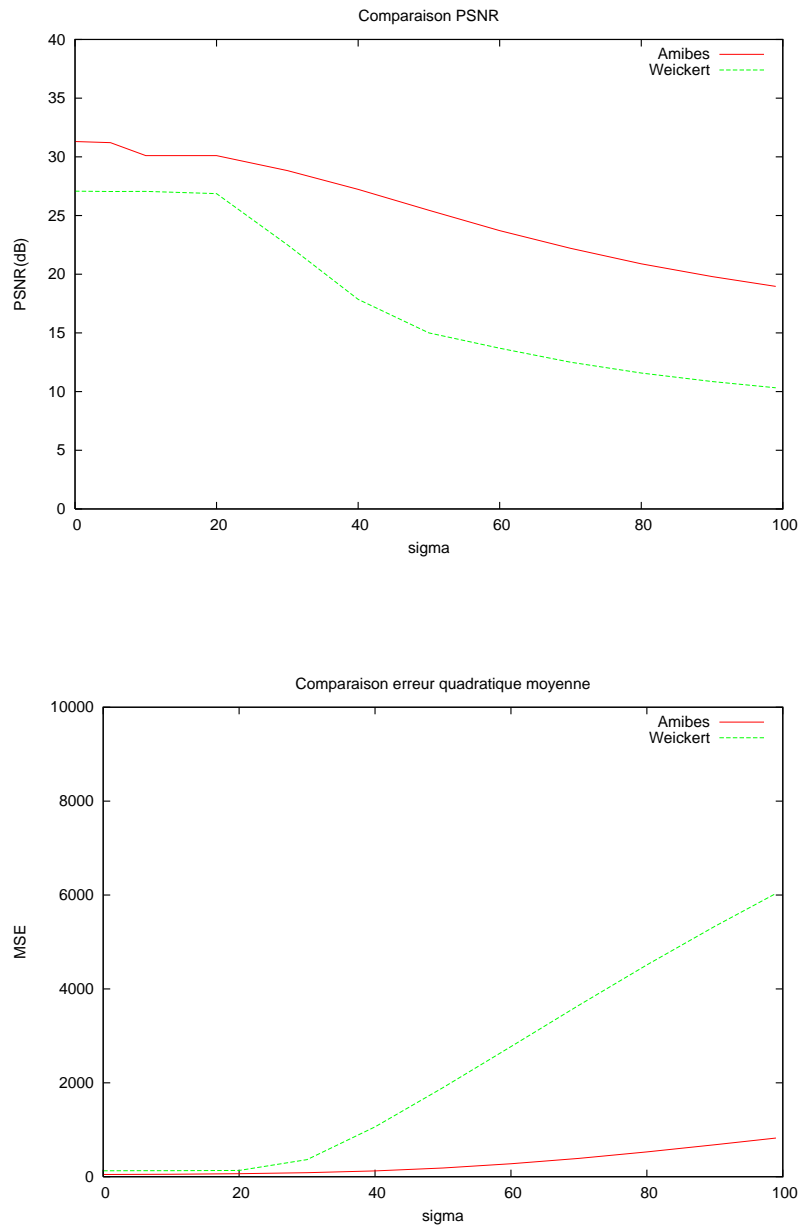


FIG. 4.21: Comparaison numérique des résultats entre les amibes et la diffusion de Weickert. Les mesures sont effectuées par rapport à l'image originale non bruitée. Pour le PSNR (haut), plus hautes sont les valeurs, meilleur est résultat, pour l'erreur moyenne (bas) c'est l'inverse.

d'espaces et de distances plus «perceptuels» (HLS ou LAB) nous permet d'envisager des résultats encore meilleurs.

Enfin, il faut noter que ce cadre de développement est suffisamment général pour être utilisé avec d'autres types de distance. Celle présentée ici réalise un couplage entre la géométrie et les niveaux de gris, mais d'autres mécanismes similaires peuvent être exprimés en termes d'amibes. Ainsi, les approches présentées dans [Braga-Neto, 1996] et plus récemment étendues dans [Debayle and Pinoli, 2005] et qui offrent des résultats intéressants peuvent être exprimées sous forme d'amibes munies d'une distance appropriée. Par ailleurs, le cas classique des éléments structurants de forme fixe n'est après tout qu'un cas particulier d'amibes. Ces approches diverses montrent autant la viabilité que la vitalité des amibes morphologiques ainsi que les vastes champs d'application qui leurs sont ouverts.

Chapitre 5

Gradients

*Car la vie se nourrit de différences ;
l'uniformité mène à la mort.*

Albert Jacquard

Nous présenterons ici les résultats de nos recherches sur la définition d'un opérateur de gradient pour les images couleurs, en nous focalisant sur sa résistance au bruit. Nous utiliserons une approche se basant sur l'étude des différences couleurs entre tous les couples de points voisins. Nous comparerons deux variantes de cette étude aux gradients plus classiques, en particulier le gradient de Beucher et le gradient morphologique généralisé.

5.1 Introduction

Dans sa définition la plus large, le gradient est un outil permettant de détecter les zones de changement dans une image. Morphologiquement parlant, c'est un résidu. Il est donc indépendant du niveau absolu des valeurs dans l'images, ainsi que des dérivées à grand échelle. En traitement du signal, on dirait que c'est un filtre passe-haut qui coupe donc les basses fréquences dans l'image.

Serge Beucher définit dans [Beucher, 1991] le «paradigme de la segmentation morphologique» comme l'utilisation de la Ligne de Partage des Eaux (LPE) [Beucher and Lantuéjoul, 1979] pour inonder une image gradient à partir d'un jeu de marqueurs.

Mais au-delà de la morphologie mathématique, de nombreux outils de segmentation utilisent une estimation du gradient local. En particulier les approches globales minimisant une énergie comme les «snakes» [Kass et al., 1988] et les «ballons» [Cohen, 1991], les «level sets» [Osher and Sethian, 1988] ainsi que les approches basées sur les coupes dans les graphes [Boykov and Kolmogorov, 2001]. On peut également rappeler les algorithmes de lancer de rayon du chapitre précédent qui nécessitaient une information de gradient pour calculer le rendu d'un volume 3D.

Toutes ces applications bénéficieraient d'un estimateur robuste et efficace du gradient. Cet estimateur robuste devra résoudre deux problèmes critiques : d'une part, les gradients classiques sont en général très sensibles au bruit. D'autre part, ils utilisent en général mal l'information chromatique. Dans ce chapitre, nous proposerons une extension robuste au gradient morphologique de Beucher [Beucher, 1990] pour les images couleur bruitées.



FIG. 5.1: Impact du bruit sur le gradient (le gradient utilisé est le gradient morphologique, mais le résultat est valable pour la plupart des opérateurs de ce type).

5.2 Réflexions sur le gradient et la segmentation

5.2.1 Impact du bruit

Le gradient est un filtre passe-haut assez simple, et il réagit beaucoup au bruit (figure 5.1). Ceci est vrai pour la plupart des opérateurs traditionnels de gradient, qu'ils soient morphologiques ou basés sur des noyaux de convolutions.

Dès les premiers travaux sur les gradients, des opérateurs comme ceux de Prewitt, Sobel ou Kirsch ont inclus une notion de filtrage local dans une direction orthogonale à la mesure du gradient. En continuant dans cette ligne, une approche efficace nécessiterait donc une mesure des différences robuste face au bruit.

5.2.2 En procédant par conditions nécessaires

Dualité du problème

L'utilisation du gradient vise à répondre à la question «où sont les contours de l'image?». Bien que parfaitement valide, cette question n'est pas celle qui nous intéresse.

Dans l'approche que nous avons de la segmentation, nous cherchons à répondre à la question duale : «où sont les *objets* de l'image?». Cette question est duale dans la mesure où l'on considère que les contours sont les duaux des objets. Comme on le sait cette affirmation n'est valable que dans les cas idéaux et peu bruités, et pour les contours fermés.

Dans cette approche duale, nous cherchons prioritairement des *régions*. À cet effet, nous utilisons des algorithmes de segmentation pour lesquels le calcul des contours n'est qu'une étape préalable à l'agrégation de pixels en régions.

La notion de contour

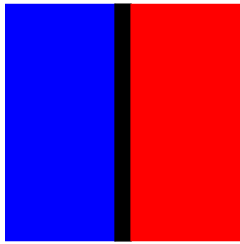
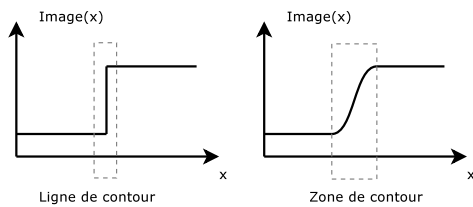


FIG. 5.2: Une frontière est une séparation entre deux régions.

Une région est un ensemble connexe de pixels partageant certaines propriétés. Cette définition très vague permet de multiples variations : connexité par arc ou connexité forte ? Types de propriétés, etc.

Dans la plupart des cas pratiques, on considère en premier lieu la couleur. Une région est un groupe de pixels ayant la même couleur, ou dans laquelle la variation de couleur est localement faible.



Une zone de contour (car on ne peut pas réellement parler de *ligne* de contour) est donc une zone dont les points sont proches de deux régions distinctes. Pour déterminer si un point est dans une zone de contour, il suffit donc d'examiner son voisinage et de déterminer s'il est proche de plus d'une région.

En niveaux de gris, la mesure naturelle de distance est la valeur absolue des différences. Dans le cas des espaces couleur, les choses sont plus complexes.

5.3 Le problème de la couleur

L'utilisation de l'information chromatique est une tâche complexe, aussi bien du point de vue théorique que du point de vue de la mise en œuvre pratique.

La plupart des travaux basés sur une approche «traitement du signal» ne sont pas adaptés au traitement des images couleurs. L'approche retenue est souvent de traiter séparément chaque composante (R,G et B, ou Y,U et V, etc) et de combiner le résultat. Ceci est sans doute raisonnable en première approximation mais n'est pas complètement satisfaisant, et ne rend pas compte des nombreux travaux sur la perception de la couleur et des différences entre couleurs.

5.3.1 Sensibilité aux variations

Le problème de l'approche séparant les canaux est que la différence entre deux couleurs est parfois plus importante quand on la prend en compte directement que ne l'est la somme entre la différence de luminance et celle de chrominance.

Par ailleurs, il ne faut pas oublier que notre cadre applicatif est l'analyse d'images «réelles» c'est à dire des images pour lesquelles l'être humain est la référence ultime. À ce titre, de nombreuses études ont été réalisées visant à quantifier les différences perçues entre diverses couleurs par des humains. Les plus anciennes sont celles menées par la CIE (Commission Internationale de l'Éclairage) qui a établi dès 1931 un «observateur colorimétrique de référence». Les travaux de MacAdam tout au long de la seconde moitié du 20ème siècle ont porté sur la quantification des zones dans lesquelles un observateur humain ne perçoit pas de différence entre des couleurs (ellipses de MacAdam).

5.3.2 Espaces perceptuels

La méthode la plus souvent retenue pour tenir compte de cette subjectivité humaine comporte deux phases. La première consiste à changer l'espace des couleurs de l'image. D'un espace initialement RGB ou YUV (pour la vidéo) on va passer dans un espace «perceptuel» (LAB, par exemple, défini en 1976 par la CIE) qui facilitera la seconde étape. Celle-ci est la détermination et l'utilisation d'une distance associée à chaque espace perceptuel qui transcrit au mieux les différences subjectives entre les couleurs.

Par exemple, LAB est conçu pour être utilisé avec une distance euclidienne (norme L_2 des 3 composantes). Pour HLS, on dispose de plusieurs approches. La thèse de Jesus Angulo [Angulo, 2003] comporte des analyses détaillées des espaces couleurs et de leurs distances associées et propose certaines améliorations. Par ailleurs, Sharma et Trussell [Sharma and Trussell, 1997] présentent un récapitulatif assez complet des technologies liées à la couleur aussi bien dans le domaine du traitement d'image que dans les systèmes d'affichage ou d'acquisition.

5.4 Gradients morphologiques généralisés

Le gradient morphologique (ou gradient de Beucher) compare un objet à son érodé (ou son dilaté, ou plus généralement les deux entre eux). Pour pouvoir utiliser ces opérateurs classiques, il faut pouvoir se placer dans un treillis.

Les treillis sur les ensembles binaires peuvent être étendus à des ensembles de scalaires ordonnés [Serra, 1988a] par l'opération traditionnelle «d'empilage», ou «méthode des seuils». Cependant, ce n'est pas possible dès lors que l'on opère sur des images dont les pixels ne disposent pas d'un ordre total, en particulier des images couleur.

Plusieurs solutions sont possibles pour pallier ce manque d'ordre. En particulier l'utilisation d'un ordre lexicographique [d'Ornellas, 2000] (avec un biais arbitraire), ou un ordre calculé sur une projection des vecteurs couleur dans un espace 1-D [Gomila, 2001], etc.

On peut également définir un «gradient morphologique généralisé» dont les propriétés sur la restriction à l'ensemble des images en niveaux de gris coïncident avec le gradient morphologique épais. En effet, le gradient morphologique épais revient à calculer la distance maximale entre les voisins du point¹ :

$$\begin{aligned} \text{Gradient}(x) &= \delta_B(x) - \epsilon_B(x) \\ &= \max\{y, y \in B_x\} - \min\{y, y \in B_x\} \\ &= \max\{|y - z|, (y, z) \in B_x^2\} \end{aligned}$$

Le problème de cette approche, comme celui de tant d'autres détecteurs de contours, est qu'elle répond fortement à un unique point distinct de tous les autres (valeur aberrante). D'autres approches sont possibles pour mitiger l'impact d'un bruit faiblement structuré (gaussien ou poivre-et-sel). Au lieu de chercher le maximum des différences, on va chercher d'abord à écarter les valeurs aberrantes et ensuite trouver une valeur de gradient aussi proche que possible de la valeur réelle non bruitée.

¹Le lecteur aura noté que l'on a soigneusement escamoté tous les problèmes d'adjonction, et que la formule n'est valable que pour un élément structurant symétrique et invariant par translation.

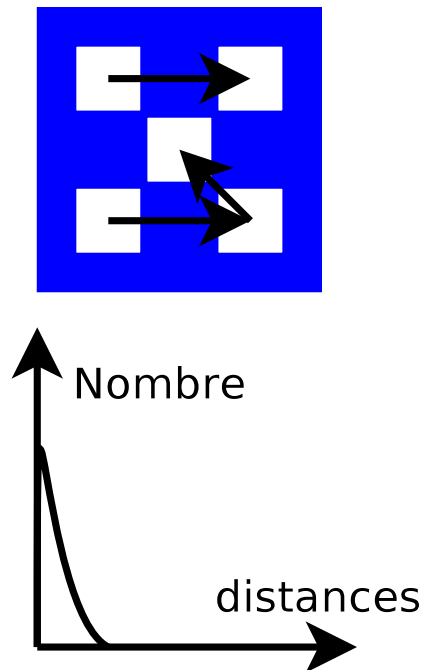


FIG. 5.3: Histogramme des distances entre tous les couples de points d'un voisinage centré au coeur d'un objet.

5.4.1 L'état de l'art : l'approche de Evans

L'approche de Evans dans [Evans, 2004] consiste à écarter les n paires de points les plus distants. Celle-ci permet d'améliorer la robustesse et pour les images scalaires, correspond *grossièrement* à remplacer l'érosion et la dilatation par des filtres de rang moins actifs. L'effet obtenu est efficace, mais une étude plus approfondie nous a permis d'améliorer les résultats.

5.4.2 Amélioration

Pour bien comprendre le comportement d'un opérateur de type gradient morphologique généralisé, il faut étudier la population sur laquelle il travaille. Dans notre cas, il s'agit de l'histogramme des distance entre tous les couples de points.

Dans le cas où on est au milieu d'un objet supposé de couleur raisonnablement uniforme (voir figure 5.3), la plupart des distances sont faibles, car entre points de même couleur. Ainsi avec une densité de pixels bruités de 0.2, on a 64% de toutes les distances qui sont entre points réellement appartenant à l'objet. Dans l'histogramme des distances, on aura donc un mode principal centré autour de 0 qui comprendra les distances intrinsèques à l'objet, et le reste réparti selon le type de bruit.

Lorsqu'on s'approche d'un autre objet, la population sur laquelle on échantillonne change, car le nombre de pixels du nouvel objet va augmenter proportionnellement. L'histogramme des distances aura donc trois zones (voir figure 5.4) : un mode centré sur 0, correspondant aux couples intra-régions, un mode centré sur la distance moyenne entre les deux régions, et un nuage de points de bruits, dont la répartition dépend du type de bruit (dans le cas d'un bruit gaussien, on voit principalement un élargissement des modes).

Pour déterminer une valeur «de gradient» acceptable, il suffit donc de trouver l'abscisse du mode de droite (voir figures 5.5 et 5.6). La méthode que nous utilisons est celle du «mean shift» déve-

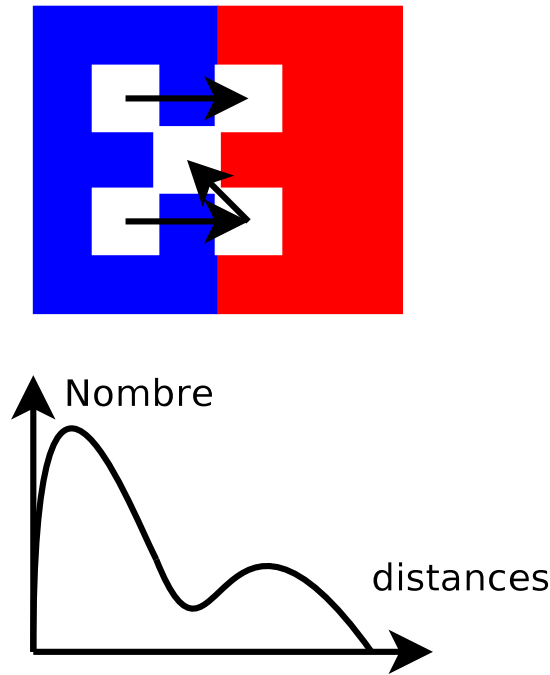


FIG. 5.4: Histogramme des distances entre tous les couples de points d'un voisinage proche d'une frontière.

loppé initialement par Fukunaga et Hostetler [Fukunaga and Hostetler, 1975] et popularisée par Cheng [Cheng, 1995] et Comaniciu [Comaniciu and Meer, 1997]. Cette méthode statistique de recherche de mode nous permet de converger itérativement de façon très robuste vers le mode de droite de l'histogramme en prenant un point de départ situé vers la droite de l'histogramme (le quantile à 0,9). Ce gradient «meilleur-mode» offre des résultats excellents, mais au prix d'un calcul coûteux de la position du mode correspondant au gradient. Les résultats sont présentés dans la figure 5.7.

5.4.3 Mesure de l'écart-type

Dans beaucoup de cas, on peut faire des hypothèses assez peu contraignantes sur la variabilité de la couleur à l'intérieur d'une région. En particulier, on peut supposer que celle-ci est relativement constante. Ceci est compatible avec l'approche morphologique de la segmentation qui se base sur la détermination de régions. Les zones fortement texturées posent toujours des problèmes que l'on ne peut en général résoudre qu'à l'aide d'outils spécifiques ou en faisant des suppositions supplémentaires (analyse du mouvement, modèle de texture, couleur, etc). De plus, on peut supposer que le bruit est du type gaussien, ce qui est généralement avéré dans nos applications.

Dans les illustrations précédentes, on a vu que l'on pouvait utiliser l'histogramme des couples de distance pour calculer un gradient tout à fait efficace. La mesure de l'écart-type global de l'histogramme nous donne une mesure intéressante sur la «largeur» de cet histogramme. Quand cet histogramme est mono-mode, l'écart-type est de l'ordre de grandeur de celui du bruit. Quand l'histogramme est bi-mode, donc que l'on est à la frontière entre deux régions, l'écart-type est fonction de la différence de couleur entre les régions ainsi que du bruit. Globalement, quand le bruit est faible, l'écart-type est proportionnel à la distance couleur entre les deux régions. Nous verrons que ceci reste vrai même quand le bruit augmente de façon significative.

La figure 5.8 montre les résultats d'un gradient de ce type appliqué à une image bruitée. On voit

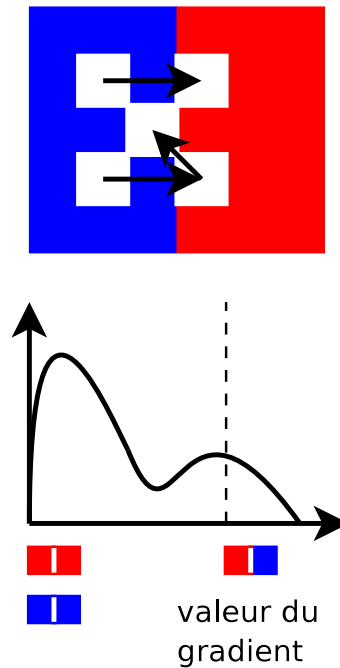


FIG. 5.5: Histogramme des distances et calcul du gradient. Le premier mode correspond aux couples intra-objets, le second aux couples inter-objets. L'abscisse de ce mode nous donne une bonne valeur de gradient.

ainsi qu'en pratique les résultats sont bons et peu influencés par le bruit.

5.4.4 Gradients par amibes morphologiques

Il découle des résultats de la section précédente qu'un bon gradient est un gradient capable d'utiliser de l'information issue d'une large zone pour générer une valeur locale.

Les approches précédentes utilisent des noyaux de forme fixe, et l'évolution naturelle est donc d'utiliser des amibes morphologiques pour améliorer encore l'échantillonnage.

Une amibe morphologique étant conçue pour ne pas déborder des frontières d'un objet, il va donc falloir trouver un mécanisme pour échantillonner correctement à travers les lignes de contours. La méthode la plus immédiate consiste à utiliser plusieurs amibes réparties autour du point central (voir figure 5.9). Cette «nuée d'amibes» aura donc tendance à rester dans l'objet jusqu'au moment où elle commencera à enjambrer le contour. À ce moment, les amibes se répartiront en deux groupes : celles qui sont restées dans le premier objet, et celles qui sont allées se déployer dans le second objet (voir figure 5.10).

5.5 Résultats

5.5.1 Comparaison directe entre gradients

La figure 5.13 présente de façon synthétique le comportement des différents gradients en fonction du bruit gaussien ajouté à l'image «Lena».

Le gradient de Beucher de l'image originale (sur le canal de luminance) a été choisi comme référence (voir figure 5.12) selon des critères subjectifs. On compare à celui-ci les résultats des différents

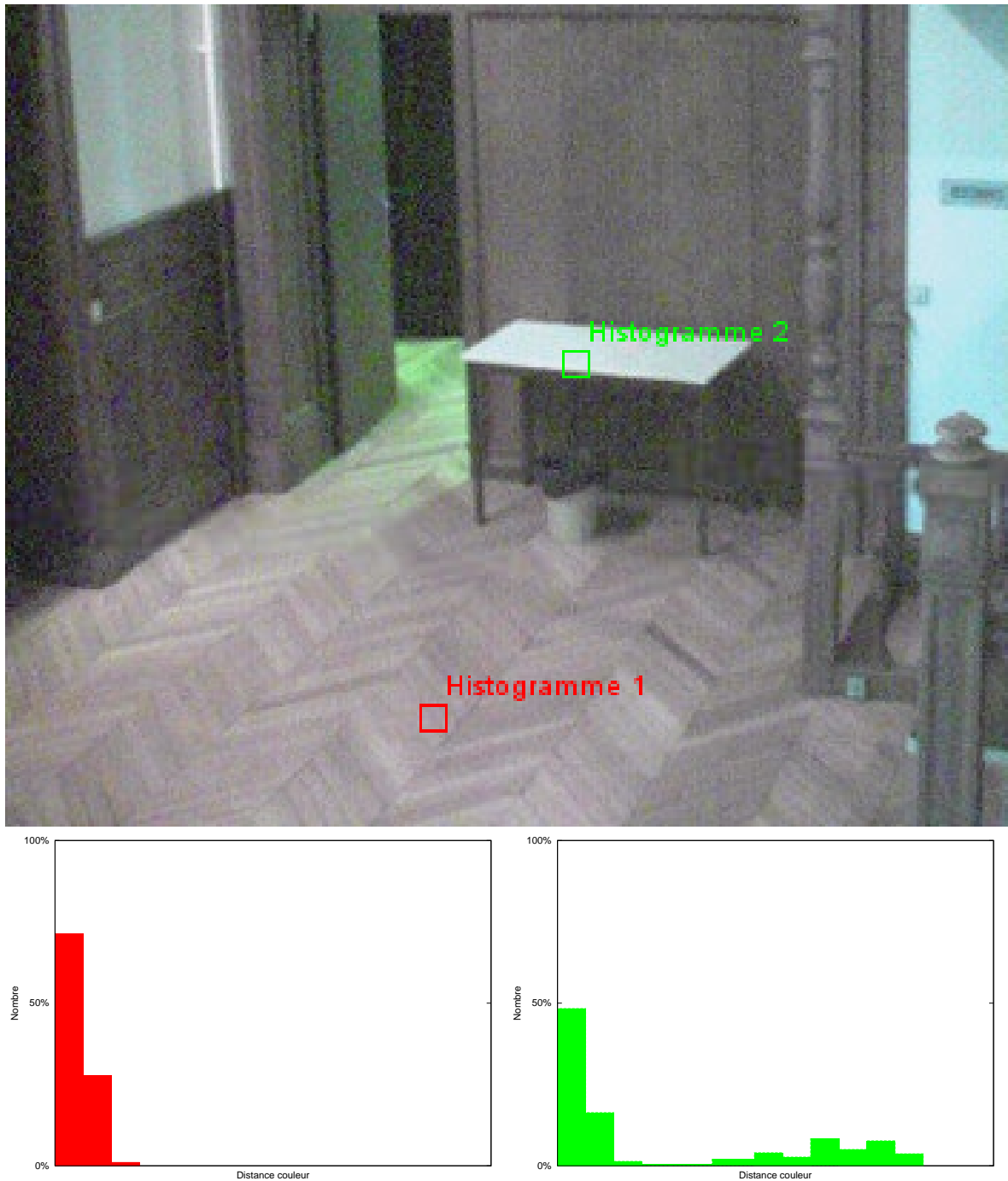


FIG. 5.6: Histogrammes de distances entre tous les couples de points dans le voisinage. On voit que la forme de l'histogramme dépend de la position dans l'image. Un seul mode dans une zone relativement plate (zone 1, à gauche) et deux modes (voire plus) près d'une zone de contour (zone 2, à droite).

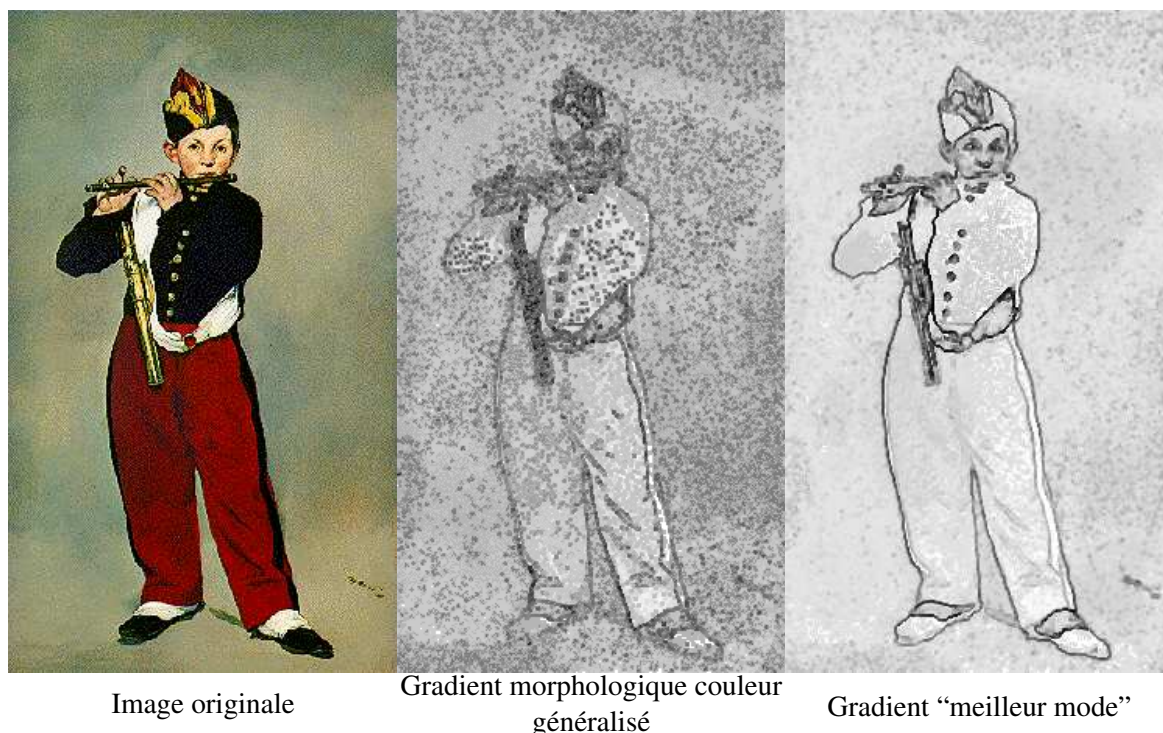


FIG. 5.7: Résultat du gradient couleur "meilleur mode" sur l'image du joueur de fifre. Le bruit n'a pratiquement aucune influence, alors qu'il est très présent dans l'image témoin.

gradients à chaque niveau de bruit au moyen d'une mesure de $PSNR^2$ (Peak Signal-to-Noise Ratio).

Le gradient calculé sur la «nuée d'amibes» est une variante du gradient «écart-type» qui offre les meilleurs résultats sur une trame fixe.

Les algorithmes comparés sont :

- le gradient morphologique de Beucher appliquée au seul canal de la luminance
- le gradient «meilleur mode» sur l'histogramme des distances couleur
- le gradient «écart-type» sur l'histogramme des distances couleur
- le gradient «écart-type» précédent appliqué sur un échantillon calculé à l'aide d'amibes morphologiques.

Et ce sur une fenêtre 3x3 et une fenêtre 5x5. Les autres gradients couleurs ont été écartés car leurs résultats sont nettement inférieurs dès que le bruit augmente. Afin de rendre les choses comparables, les images ont été post-traitées par un étirement d'histogramme jusqu'à [0; 255]. Ceci est un point de détail mineur, car dans la plupart de nos applications, c'est l'ordre relatif des valeurs du gradient qui importe et non leurs valeurs dans l'absolu.

Ces résultats sont intéressants à de nombreux titres.

Tout d'abord, on voit que dans les gradients à noyau fixe («écart-type», «meilleur mode» et Beucher) la taille de la fenêtre a une grande importance. Elle a même un impact plus important que le choix de la mesure effectuée. L'exception étant le gradient de Beucher qui est naturellement très sensible au bruit. Un gradient de type Evans, dans lequel on retire un nombre prédéfini de valeurs a un résultat proche du «meilleur-mode», dans les deux tailles de fenêtre.

Le résultat le plus étonnant est celui obtenu par les amibes morphologiques. En effet, le résultat

²Malgré ses nombreux défaut et sa faible pertinence, le PSNR la mesure universellement adoptée pour la comparaison d'images.



Image non bruitée

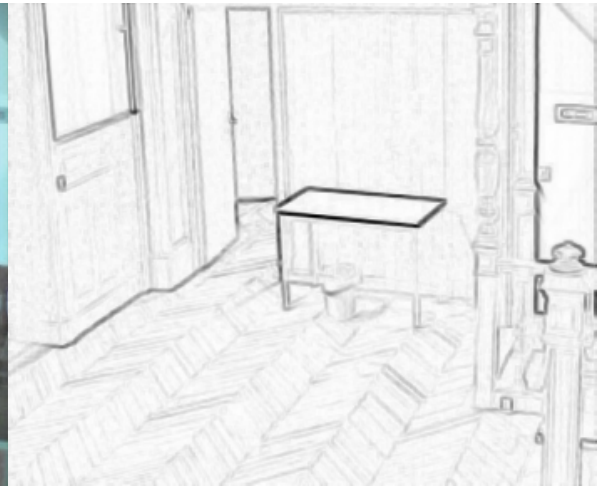


Image "gradient" par écart-type



Image originale (bruitée)

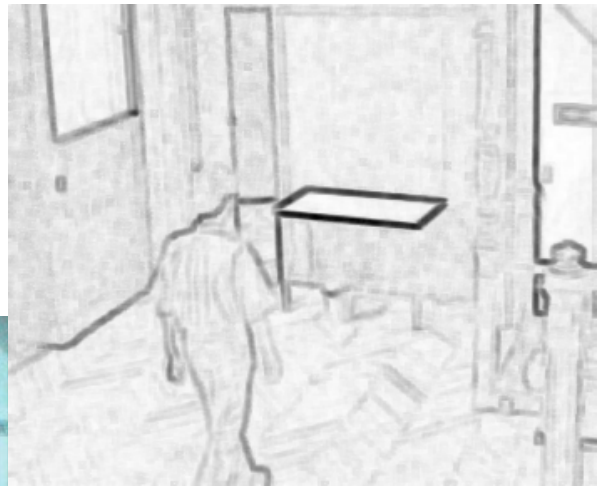
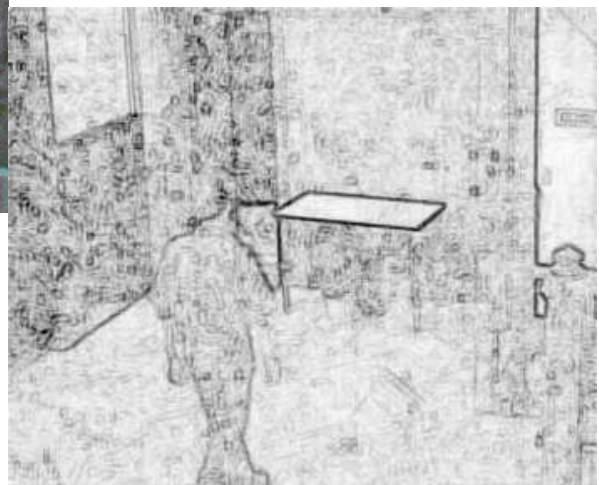


Image "gradient" par écart-type



Comparatif : image gradient morphologique couleur généralisé

FIG. 5.8: Image "gradient" par calcul de l'écart-type des distances sur une image non bruitée, puis sur une image fortement bruitée. On voit clairement que le gradient morphologique généralisé a une réponse au bruit extrêmement forte, alors que le gradient «écart-type» est capable de retrouver les contours pertinents malgré une augmentation du bruit de fond.

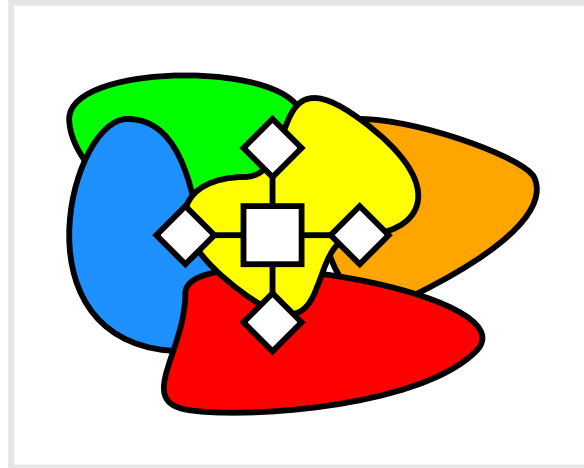


FIG. 5.9: Nuée d'amibes sur une base de 4-connexité : on place une amibe sur chaque voisin 4-connexe du point central, et on échantillonne sur l'union des amibes.

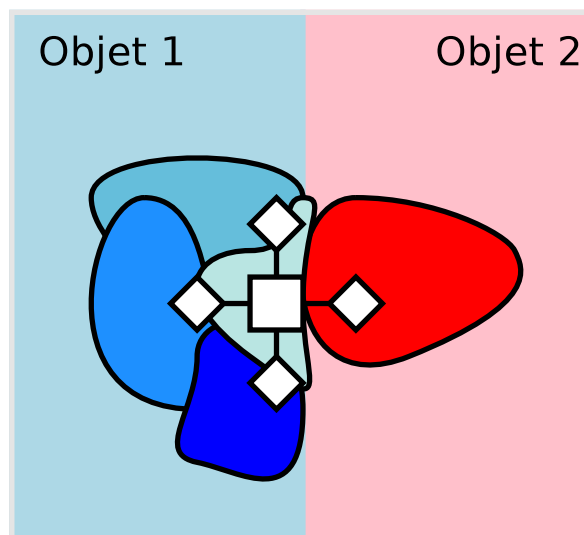


FIG. 5.10: Nuée d'amibe au voisinage d'une frontière : l'amibe de droite se développe exclusivement dans le nouvel objet, alors que les autres amibes ne franchissent pas la frontière.

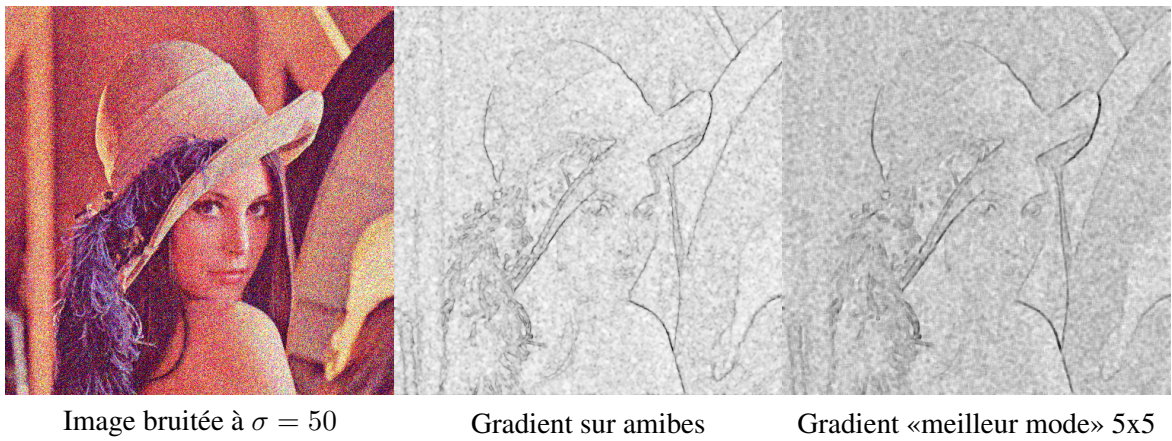
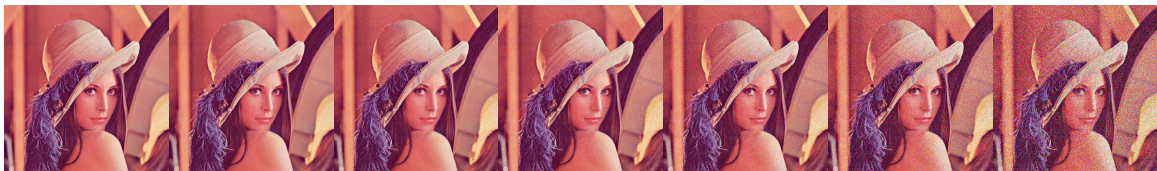
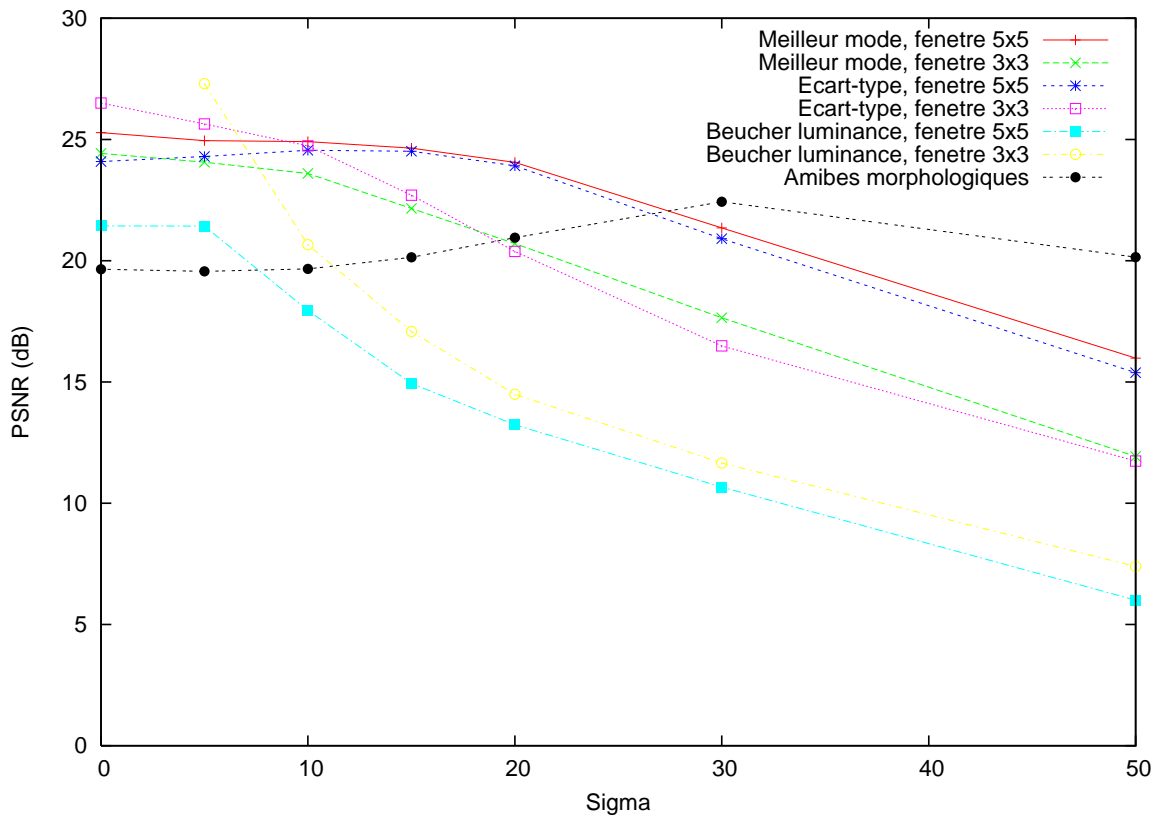


FIG. 5.11: Comportement du gradient d'amibes lorsque le niveau de bruit augmente.



FIG. 5.12: Choix d'un gradient de référence : le gradient de Beucher sur la luminance offre le résultat le plus plaisant (test réalisé sur un panel de 3 personnes).



Original $\sigma = 5$ $\sigma = 10$ $\sigma = 15$ $\sigma = 20$ $\sigma = 30$ $\sigma = 50$

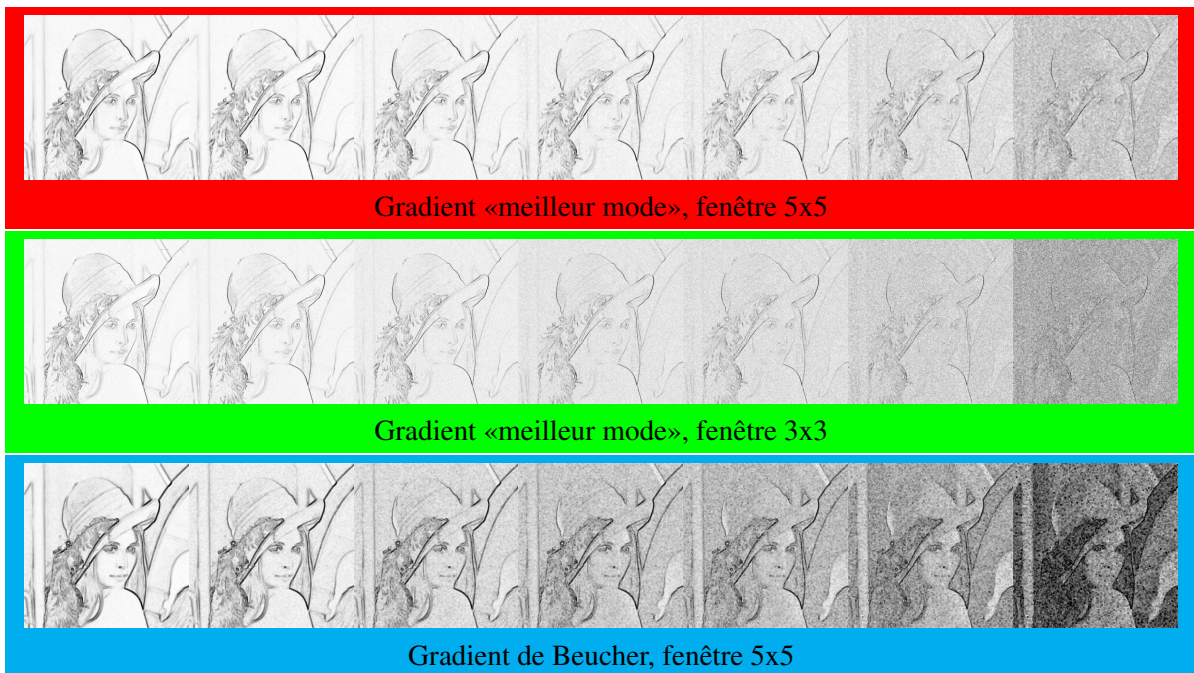


FIG. 5.13: Comparaison des résultats entre gradients par rapport au gradient de Beucher de l'image non bruitée servant de référence.

s'améliore lorsque le bruit augmente. Ce phénomène incongru est expliqué par le fait que les paramètres des amibes ont été optimisés pour une forte quantité de bruit. Les amibes gardant la même énergie dans les images peu bruitées, elles auront tendance à s'étendre excessivement dans les images plus lisses. Paradoxalement, on retrouve dans les images très bruitées des résultats proches des meilleurs résultats obtenus dans les images peu bruitées. Pour minimiser ce problème, on peut commencer par filtrer l'image par une amibe, afin de retirer le bruit tout en préservant les contours. On peut également chercher à apprendre les paramètres en fonction du bruit. Cette approche-ci est particulièrement valable dans le cas où on traite de nombreuses images aux caractéristiques similaires.

5.5.2 Comparaison des segmentations résultantes

Au début de ce chapitre, nous avons rappelé qu'une des utilisations principales des gradients en traitement d'images était la segmentation. Le résultat de la ligne de partage des eaux dépend de deux éléments : le gradient et les marqueurs initiaux. Afin de tester nos gradients, nous allons donc fixer des marqueurs et comparer les segmentations lorsque le niveau de bruit augmente.

Pour comparer uniquement l'effet du gradient, nous avons marqué deux régions dans «Lena» et nous allons comparer pour chaque gradient le résultat de l'inondation suivant les marqueurs spécifiés.

Choix d'une mesure

Par souci de cohérence avec notre choix d'un gradient de référence, on pose que le résultat de référence est celui obtenu avec le gradient de Beucher de la luminance sur une fenêtre 3x3.



Le choix d'une mesure comparant les résultats est plus complexe. Pour tenir compte des différentes façons dont deux contours peuvent être dissemblables, nous utiliserons une mesure basée sur des fonctions distances évaluées dans la zone entre les deux contours inspirée de la méthode développée dans [Villegas et al., 1999]. On affectera à chaque point situé entre les deux contours le maximum des distances entre ce point et chaque contour. Dans un premier temps nous ferons la somme de ces valeurs, et dans un second temps nous examinerons les histogrammes des distances pour déterminer si les contours diffèrent par quelques grandes zones, ou bien par de nombreuses petites variations.

La figure 5.14 illustre la méthode et la figure 5.15 présente les courbes des mesures agrégées sur toute les zones de différences en fonction de la quantité de bruit ajouté.

Résultats

En première analyse, un point marquant de ce diagramme est la grande variabilité des résultats. En effet, le gradient de Beucher en fenêtre 3x3 présente un écart spectaculaire pour $\sigma = 20$, et curieusement, tous les résultats convergent vers un bon score pour $\sigma = 30$. Sans doute aurait-il fallu faire plusieurs réalisations pour une même valeur de σ et garder la moyenne des mesures. D'un autre côté, dans la réalité on est obligé d'utiliser les images dont on dispose, et ceci sert de belle illustration à l'instabilité de la ligne de partage des eaux évoquée précédemment.

Parmi les différents gradients, on remarque que le gradient «écart-type» sur une fenêtre 5x5 a de façon constante les meilleurs résultats. Cependant, le même gradient sur une fenêtre 3x3 a des résultats

très mauvais pour le niveau de bruit énorme de $\sigma = 99$. Ceci s'explique principalement par le fait que l'échantillonnage dans le cas 3x3 est trop faible pour avoir un effet de moyenne.

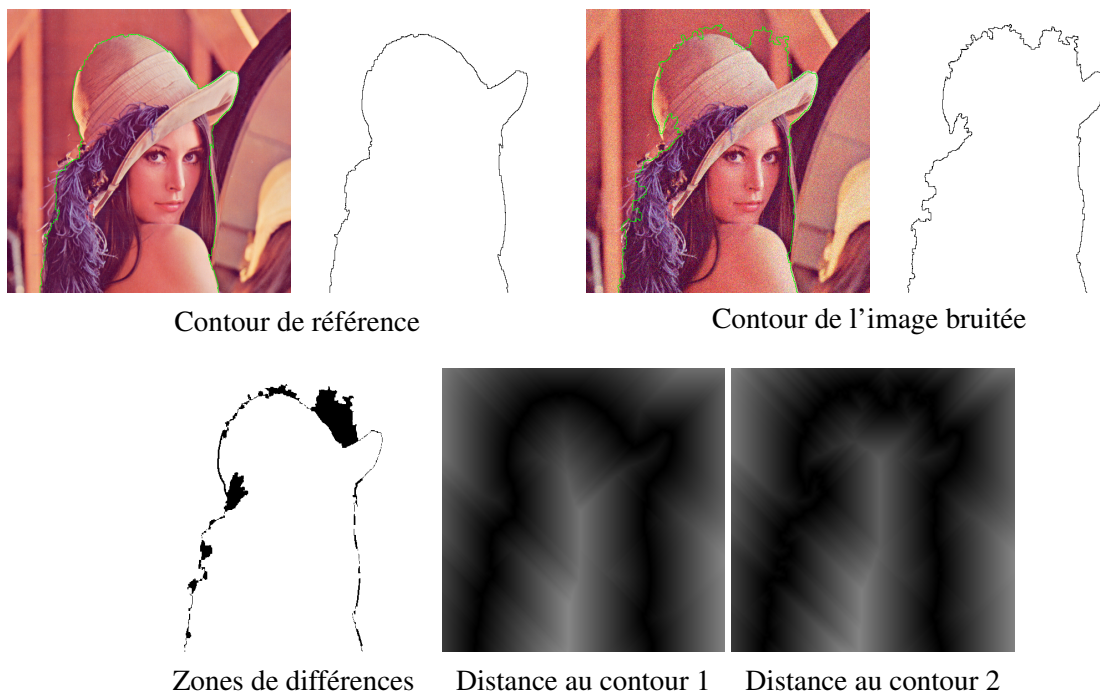


FIG. 5.14: Principe de la mesure entre deux contours : pour chaque pixel dans les zones de différences on prend le maximum des deux distances. On analyse ensuite la somme des pixels de l'image résultante, ou son histogramme.

La figure 5.16 présente un histogramme de la répartition des valeurs des distances. Dans le premier histogramme, pris pour $\sigma = 20$ on peut voir trois phénomènes : le premier est que le gradient «écart-type» sur une fenêtre 5x5 a de très bons résultats. En particulier, il n'y a aucun point ou presque dans les grandes distances. Ceci signifie que le contour est toujours très proche du contour de référence. Par opposition, on voit dans l'histogramme très allongé du gradient de Beucher 3x3 que celui-ci est souvent très éloigné du contour de référence. On est sans doute en présence d'une «fuite» de la LPE. Un autre point intéressant est la comparaison entre le gradient de Beucher 5x5 (en bleu) et le gradient écart-type 3x3 (en vert). Dans le graphique précédent, les valeurs intégrées pour ces deux gradients donnent un résultat quasi-identique. En revanche, on voit dans cet histogramme que le gradient écart-type a plus de valeurs faibles (à gauche de l'histogramme) que celui de Beucher qui le rattrape à une distance de 20 pixels et pousse jusqu'à 50 pixels d'écart. Ceci signifie que le contour du gradient «écart-type» n'est pas aussi précis que celui de Beucher mais que ce dernier présente au moins une «fuite» qui alourdit son score.

Dans le cas $\sigma = 99$, on voit que les décalages des modes des histogrammes expliquent nettement les résultats du graphique 5.15. En particulier, on voit que le gradient «écart-type» 3x3 souffre visiblement d'une fuite d'extension maximale d'environ 130 pixels, ce qui explique un résultat très mauvais.

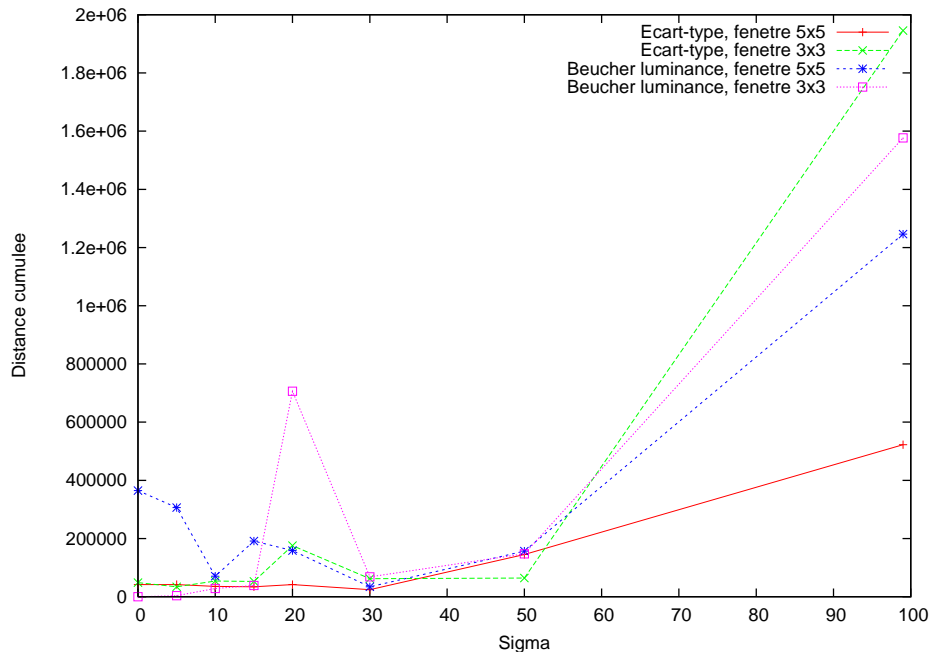


FIG. 5.15: Evolution des distances au contour de référence quand le bruit augmente.

5.6 Conclusion

Ce chapitre a présenté de nouvelles approches du gradient basées sur l'analyse plus pertinente des voisinages. Nos approches sont fondées sur l'analyse de l'histogramme des distances entre couples de points dans un voisinage 3x3, 5x5 ou autre. Nous cherchons ainsi à déterminer quelle valeur représente le mieux les différences entre les régions.

L'intérêt de la recherche d'un meilleur gradient est double : d'une part, c'est une composante essentielle de la segmentation morphologique, mais bien au-delà c'est également un élément important dans de nombreux algorithmes non morphologiques de détection de contours et également dans des algorithmes moins axés sur la segmentation, comme la visualisation 3D. Les améliorations que nous apportons ici peuvent ainsi être réutilisées dans de nombreuses applications.

Dans le cas spécifique de la segmentation morphologique, un avantage énorme de l'utilisation d'opérateurs de gradients robustes est qu'il est possible de continuer à utiliser des algorithmes de segmentation simples sur des images plus complexes ou plus bruitées.

Les chapitres suivants seront consacrés aux améliorations des techniques de segmentation et de modélisation, notamment pour les cas où même les meilleurs gradients ne peuvent pas garantir une bonne segmentation.

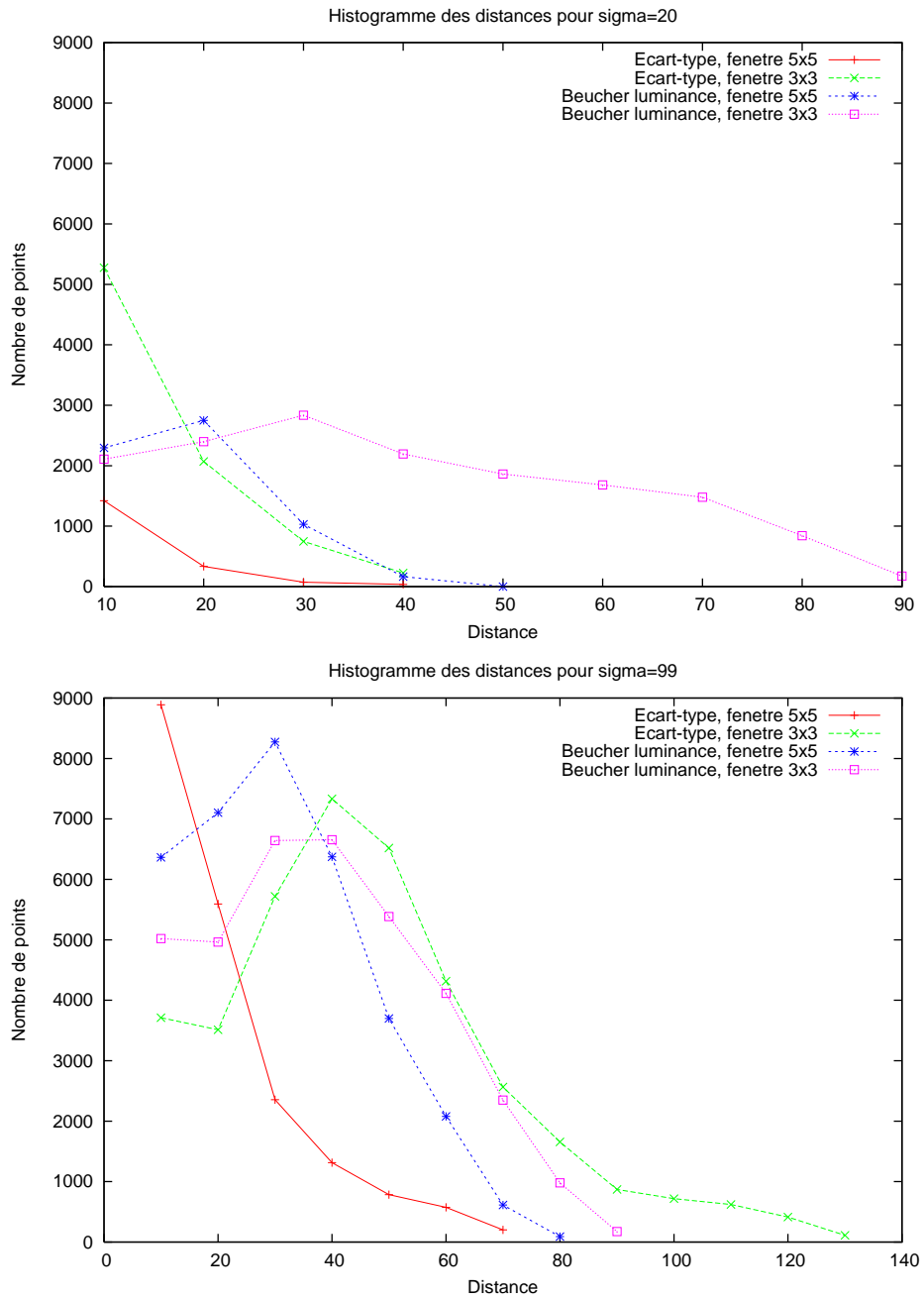


FIG. 5.16: Histogrammes des distances au contour de référence pour $\sigma = 20$ et $\sigma = 99$.



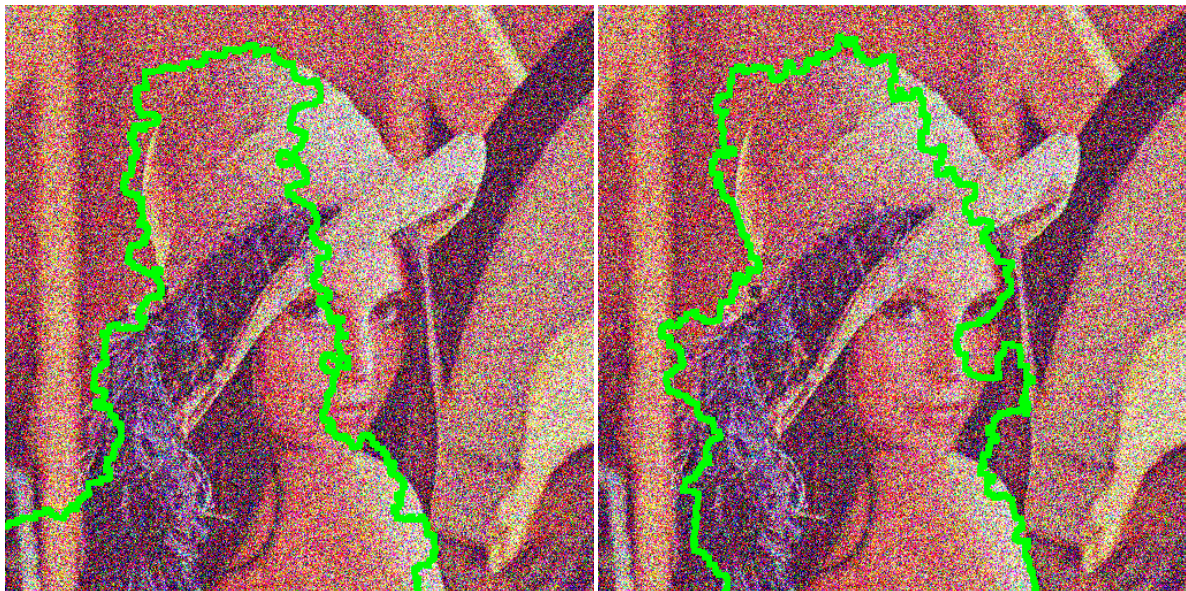
Contour de référence

$\sigma = 20$, gradient Beucher 3x3. La fuite le long du chapeau est la cause d'un très mauvais score.



$\sigma = 20$, comparaison entre le gradient «écart-type» 3x3 (gauche) et le gradient de Beucher 5x5 (droite). À droite le contour est plus proche dans les plumes et le contour inférieur du chapeau, mais le gros écart en haut à droite le pénalise beaucoup par rapport à celui de gauche qui est plus uniformément proche, malgré quelques écarts.

FIG. 5.17: Analyse des résultats pour $\sigma = 20$



Gradient «écart-type», fenêtre 3x3

Gradient «écart-type», fenêtre 5x5

FIG. 5.18: Résultats difficiles pour $\sigma = 99$ (note : pour la rendre plus visible, nous avons épaissi la ligne de contour)

Troisième partie

Modélisation par arbres

6	Inondations contrôlées	105
6.1	Introduction	105
6.2	Segmentation par LPE contrôlée	106
6.3	Espaces d'échelle morphologiques	109
6.4	Segmentation «top-down»	110
6.5	Le gradient «pyramide»	119
6.6	Conclusion	122
7	Appariement d'arbres	127
7.1	Distance d'arbres	127
7.2	Recherche d'images semblables dans une base	129
7.3	Robustesse face aux perturbations	134
7.4	Conclusion	137

Segmentation et modélisation

Les chapitres précédents visaient à améliorer l'image ou son gradient. Les amibes morphologiques que nous avons développées se sont avérées être un outil très efficace de réduction du bruit. Parallèlement, nous avons développé des opérateurs de gradient, fonctionnant notamment sur des images couleur, robustes face au bruit.

Ces opérations permettent de limiter, voire d'éviter les divergences entre les segmentations d'images qui se *ressemblent* mais malgré ces améliorations, une nouvelle approche de la modélisation est nécessaire pour gérer le cas des images plus dissemblables.

Second axe d'amélioration : vers une approche orientée-objet

Selon les résultats que nous avons obtenus dans nos premiers essais d'extension du travail de Cristina Gomila, il semblerait que l'approche visant à modéliser l'ensemble de l'image par un graphe ne fonctionne que dans les cas simples. Si les extensions et améliorations que nous avons apportées à cette approche n'ont pas réussi à améliorer sa robustesse de manière très significative sur des images moins faciles, c'est sans doute qu'il est temps de définir un nouveau cadre plus approprié à l'évolution de nos besoins.

L'analyse développée à la fin du chapitre 3 sur la stabilité des arêtes au cours du temps, ainsi que nos essais d'amélioration de la méthode de Cristina, nous conduisent à chercher des méthodes de modélisation qui traitent séparément les différents objets. Le principe serait une approche en deux passes : d'abord repérer les objets intéressants à modéliser, et ensuite décrire ces objets à l'aide d'outils puissants basés sur les graphes.

D'une part ceci nous permet d'espérer une certaine stabilité à l'intérieur des frontières de l'objet, rendant la modélisation plus stable et donc les différences plus faibles entre deux graphes issus de deux images. D'autre part, cette approche entre mieux dans le cadre des applications de type vidéo-surveillance.

Le chapitre 6 sera consacré à l'utilisation d'espaces d'échelles morphologiques alliés à une variante de la LPE pour obtenir des pyramides de segmentation qui encodent de façon robuste la structure des objets. Ces pyramides seront représentées à l'aide d'un arbre dont la racine sera le niveau le plus grossier de l'espace d'échelle et les feuilles seront les régions de l'image proprement dite. Au passage, nous utiliserons ces espaces d'échelle pour définir un gradient combinant l'information locale de couleur avec une information plus globale ordonnant la taille des objets.

Le chapitre 7 sera consacré à la conception d'une mesure de dissemblance entre les arbres ainsi créés, ainsi qu'à des tests sur une base d'images standard du monde de l'indexation.

Chapitre 6

Inondations contrôlées et espaces d'échelle morphologiques

Les eaux grossirent de plus en plus, et toutes les hautes montagnes qui sont sous le ciel entier furent couvertes.

Génèse, 7 :19

Ce chapitre présente une variante de l'algorithme de Ligne de Partage des Eaux (LPE) permettant de tenir compte d'une segmentation préalable de l'image. Nous verrons comment combiner cette méthode aux espaces d'échelle morphologiques pour obtenir une représentation arborescente puissante et robuste des objets d'une image, en supposant connue leur silhouette.

Le chapitre 9 illustrera dans une approche plus applicative le problème de la détection de la silhouette. Le chapitre 7 développera l'utilisation de ces «inondations contrôlées» pour la comparaison d'objets modélisés par cette approche.

6.1 Introduction

Nous avons vu au début du chapitre 2 que l'approche classique de la segmentation comprenait deux étapes : l'une très proche de l'image visant à passer du pixel à la région élémentaire et l'autre fusionnant ces régions selon divers critères jusqu'au niveau de l'objet ou du sous-objet. Par ailleurs, nous avons vu à la fin de ce chapitre que cette approche «bottom-up» butait sur deux problèmes : d'une part les images difficiles, dans lesquelles l'information locale servant à fusionner les régions n'était pas fiable, et d'autre part les problèmes liés à la modélisation «monoplanaire» de séquences en mouvement. Le premier problème, le plus fondamental, est lié au fait qu'une région élémentaire a une définition *locale* et donc ne peut pas connaître son rôle dans l'image. Il faut pour cela une information plus étendue que celle disponible au niveau régional.

Ce chapitre présente une approche inverse, «top-down», du problème. Plutôt que de partir des régions les plus petites pour les regrouper, nous allons au contraire chercher directement les grandes régions de l'image et les subdiviser. Pour ce faire, nous allons d'abord avoir besoin d'une méthode

nous permettant d'extraire les zones significatives de l'image. Puis nous aurons besoin d'un mécanisme pour raffiner notre connaissance de ces zones.

La méthode que nous utiliserons se basera sur les espaces d'échelle, qui visent précisément à modéliser une image à travers plusieurs niveaux de taille. Cependant, comme notre objectif est la segmentation, nous aurons besoin d'une méthode plus précise au niveau des contours que les espaces d'échelle traditionnels. Nous utiliserons donc des espaces d'échelle morphologiques, qui présentent de bonnes propriétés de simplification par la taille tout en préservant relativement bien les contours.

Au cœur de ce système se trouvera la Ligne de Partage des Eaux (LPE) contrôlée, dont le rôle est de propager l'information de position des objets du niveau le plus grossier vers le niveau le plus fin. Ainsi, à chaque niveau, les résultats du niveau plus grossier serviront de référence à un processus d'inondation qui raffine la segmentation précédente. La référence initiale pouvant être soit une LPE simple du niveau le plus grossier, soit être issue d'un processus externe, dessinée à la main, etc.

Nous commencerons par décrire ci-dessous cette nouvelle LPE, puis nous rappellerons brièvement la notion d'espace d'échelle morphologique avant d'ajouter quelques raffinements au système. Nous terminerons par une excursion sur une variante utilisant les espaces d'échelle morphologiques pour construire un gradient robuste, que nous comparerons à ceux du chapitre 5.

6.2 Segmentation par LPE contrôlée

6.2.1 Description

Cette approche modifie le «paradigme de la segmentation morphologique» (gradients, marqueurs, LPE) en lui ajoutant une composante supplémentaire : la partition de référence. Les utilisations possibles sont multiples et nous ne prétendons pas en donner une liste exhaustive. Initialement, on suppose que l'on dispose d'une partition de l'image en régions, appelée *partition de référence*. Cette partition peut être par exemple obtenue par un autre système de détection, tracée à la main, ou obtenue par une méthode de segmentation différente de la LPE. Le principe de cette méthode consiste à contraindre chaque lac formé par l'inondation à rester dans la partition contenant son marqueur initial. Le résultat est donc une segmentation plus fine respectant la partition de référence. Dans le cadre de notre problème, la segmentation et la représentation d'objets, nous supposons en particulier que nous disposons d'une silhouette globale de l'objet. Cette silhouette formera la base de la partition de référence initiale. Ceci nous permet de nous concentrer dans cette étape sur la modélisation de l'objet.

Une implémentation simplifiée¹ de l'algorithme standard de la LPE par file d'attente hiérarchique [Meyer, 1991] est cité ci-dessous comme référence (voir algorithme 10, page 107).

L'algorithme de resegmentation est décrit ci-dessous, les modifications par rapport à l'algorithme original étant notées en **gras souligné**. Comme on le voit, la modification est minime. En particulier elle n'ajoute pas de complexité algorithmique.

Il est essentiel de comprendre que le résultat est en général *différent* de celui obtenu en faisant l'intersection de la segmentation de contrôle avec le résultat d'une LPE simple. L'explication étant simplement que l'inondation n'est pas un processus local et qu'elle dépend en général de la topographie de l'image entière.

¹Pour une implémentation plus complète et non biaisée, on se référera à [Beucher, 2004]

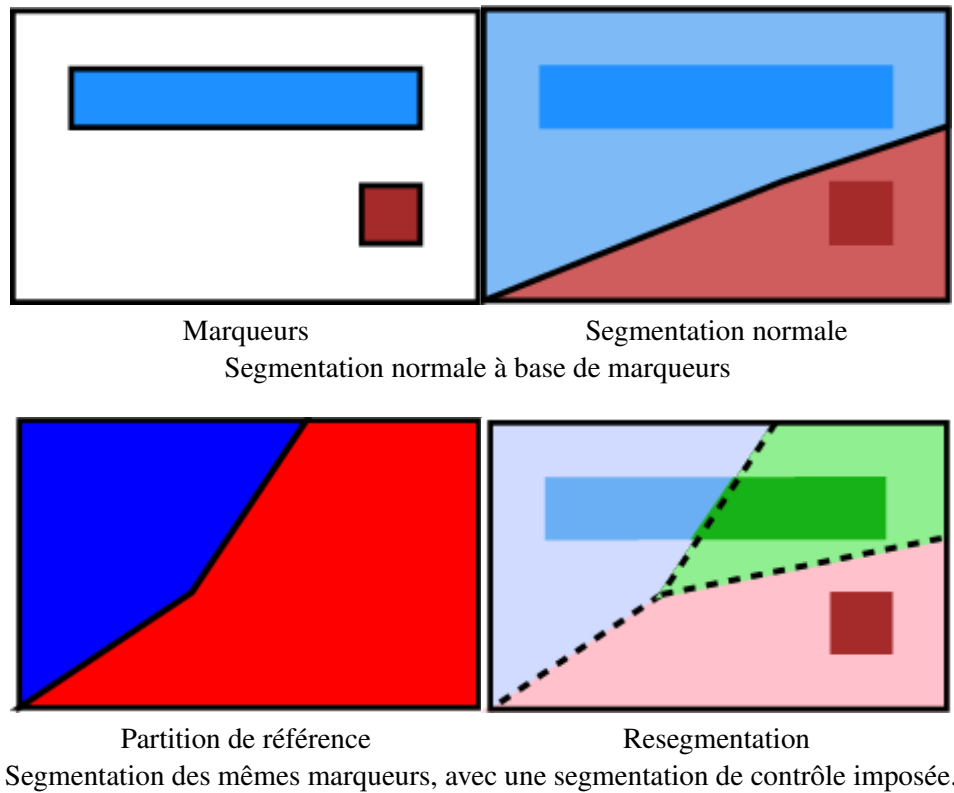


FIG. 6.1: Resegmentation d'une image suivant des marqueurs donnés

Algorithme 10 LPE standard par file d'attente hiérarchique (FAH)

Mettre les minima dans une FAH
tantque FAH non vide **faire**
 p = sortir le pixel le plus bas
 pour tout q dans le voisinage de p **faire**
 si q n'est pas labellisé **alors**
 q prend le label de p
 mettre q dans la FAH
 finsi
 fin pour
fin tantque

Algorithme 11 LPE contrôlée

```

Mettre les minima dans une FAH
tantque FAH non vide faire
   $p$  = sortir le pixel le plus bas
  pour tout  $q$  dans le voisinage de  $p$  faire
    si  $p$  et  $q$  sont dans la même région de référence alors
      si  $q$  n'est pas labellisé alors
         $q$  prend le label de  $p$ 
        mettre  $p$  dans la FAH
      fin si
    fin pour
  fin tantque

```

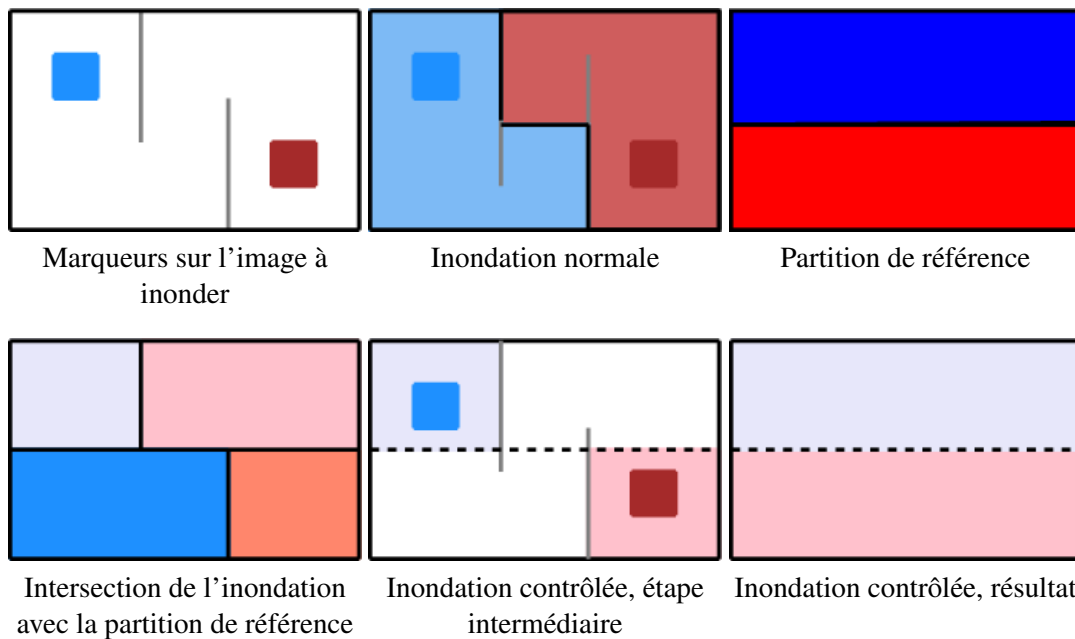


FIG. 6.2: Le résultat de l'inondation contrôlée est en général différent de l'intersection de la partition de référence avec le résultat de l'inondation normale. Ainsi, comme il n'y a qu'un seul marqueur par région de la partition de référence, chaque marqueur inondera la totalité de la région.

6.2.2 Regard furtif sur les marqueurs

L'algorithme énoncé précédemment maintient un invariant : une région ne croît jamais au-delà des limites de la région la contenant dans la partition de référence. Cependant, rien ne garantit que les marqueurs utilisés comme sources de l'inondation soient inclus chacun dans une seule région. Ceci peut causer les problèmes illustrés dans la figure 6.3.

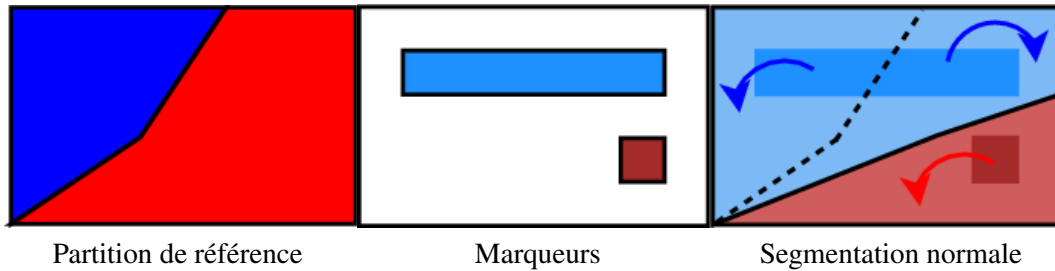


FIG. 6.3: Un marqueur couvrant initialement plus d'une région peut croître en une région qui ne respecte pas la segmentation de contrôle. Si on ne sépare pas le marqueur bleu en deux, on pourrait retrouver exactement le résultat de l'inondation non contrôlée.

La solution est de faire l'intersection entre les marqueurs et la segmentation de contrôle, ce qui divisera les marqueurs couvrant plus d'une région (voir fig. 6.4), et d'inonder à partir de ces nouveaux marqueurs.

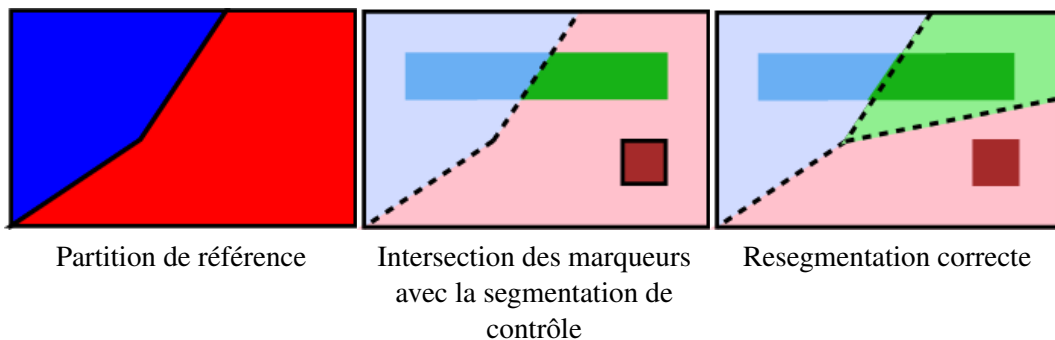


FIG. 6.4: Le marqueur problématique a été divisé et chaque région croît indépendamment. Le résultat respecte la segmentation de contrôle.

6.3 Espaces d'échelle morphologiques

6.3.1 Introduction

Les espaces d'échelle («*scalespace*» en anglais) sont une invention relativement récente en traitement d'images qui vise à résoudre le problème de la localité de l'information en déterminant à quelle échelle une information est la plus pertinente.

Cette approche, initialement développée par Witkin [Witkin, 1983], Koenderink [Koenderink, 1984] et Lindeberg [Lindeberg, 1994], est rapidement devenue un élément standard dans le domaine du traitement d'image. Le principe de cette méthode est de générer à partir d'une image $f(x, y)$ une famille L définie par :

$$L(x, y, t) = G(x, y, t) * f(x, y)$$

Où G est la gaussienne de variance t et $*$ désigne le produit de convolution. D'un autre point de vue, ceci revient à définir L comme la solution de l'équation de la chaleur :

$$\partial_t L = \frac{1}{2} \nabla^2 L$$

avec la condition initiale $L(x, y, 0) = f(x, y)$.

Ces formulations découlent naturellement des propriétés désirées des espaces d'échelle :

- invariance par translation
- invariance par rotation
- invariance aux changements d'illuminations : la transformation doit commuter avec une anamorphose croissante des niveaux de luminance

Ainsi que des propriétés liées à l'évolution de l'information le long de l'échelle :

- La transformation doit être une simplification de l'image, elle ne doit donc pas être réversible.
- Causalité : le résultat à une échelle donnée ne doit dépendre que des échelles plus fines
- La transformation ne doit pas créer de nouvelles structures aux échelles plus grossières.

Dans l'approche par EDP, celle décrite plus haut serait la seule possible. Mais un des inconvénients de cette formulation analytique est que, au niveaux les plus hauts, les objets ne sont plus reconnaissables. Pour pallier ce problème, plusieurs approches ont été proposées, dont celles basées sur les diffusions anisotropes mentionnées dans le chapitre 4 [Perona and Malik, 1990, Weickert, 1996].

6.3.2 Espaces d'échelle et nivellements

En prenant du recul, on se rend compte que les caractéristiques théoriques (linéarité, formulation par EDP) des espaces d'échelles analytiques ne correspondent pas nécessairement à des besoins en termes d'analyse d'images, et qu'il existe des alternatives vérifiant aussi bien les propriétés énoncées ci-dessus.

Meyer et Maragos proposent ainsi dans [Meyer and Maragos, 1999] l'ajout une propriété intéressante pour la segmentation : que les contours restent nets et ne soient pas déplacés. Pour ce faire, la solution proposée est d'effectuer à chaque échelle un lissage de type gaussien (ou autre) suivi d'un nivellement. Le lissage simplifie l'image en éliminant les structures fines, et le nivellement retrouve les contours. La propriété des nivellements de préserver les zones plates garantit que les images résultantes auront les propriétés attendues.

On substitue ainsi aux propriétés analytiques associées à la formulation standard des propriétés morphologiques sur la préservation ou la disparition de contours lorsqu'on remonte l'échelle vers les niveaux plus grossiers. Ces propriétés font des espaces d'échelle morphologiques parfois appelés «pyramides de nivellements» des outils idéaux pour la segmentation. On verra dans la section suivante comment utiliser les inondations contrôlées sur les espaces d'échelle morphologiques pour obtenir une représentation arborescente robuste des objets. Nous comparerons cette approche «top-down» avec l'approche classique de fusion de régions «bottom-up» [Meyer, 1994].

6.4 Segmentation «top-down»

6.4.1 Construction du dendrogramme

Grâce aux contours bien définis de l'espace d'échelle morphologique et à la LPE contrôlée, il est maintenant possible d'effectuer une segmentation «top-down» de la pyramide complète. Chaque

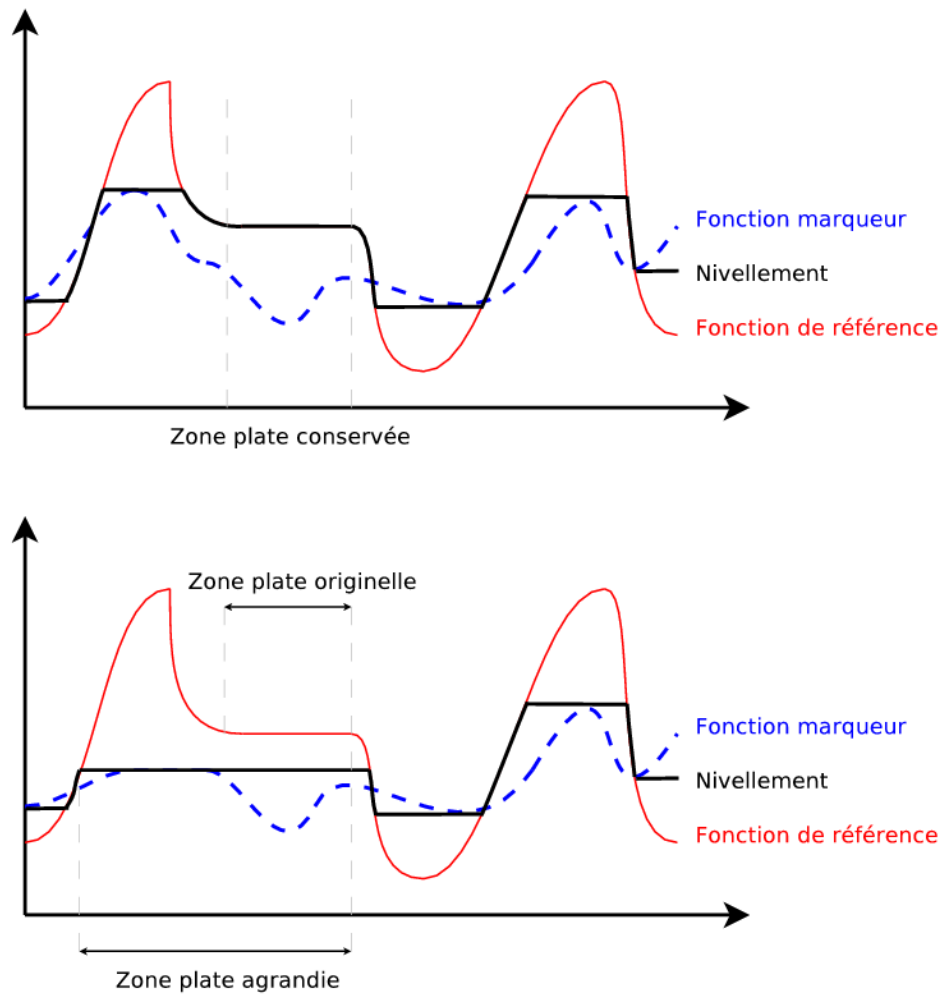


FIG. 6.5: Un nivellement peut conserver (haut) ou agrandir (bas) les zones plates d'une image, mais jamais les supprimer, quel que soit le marqueur qu'on utilise.

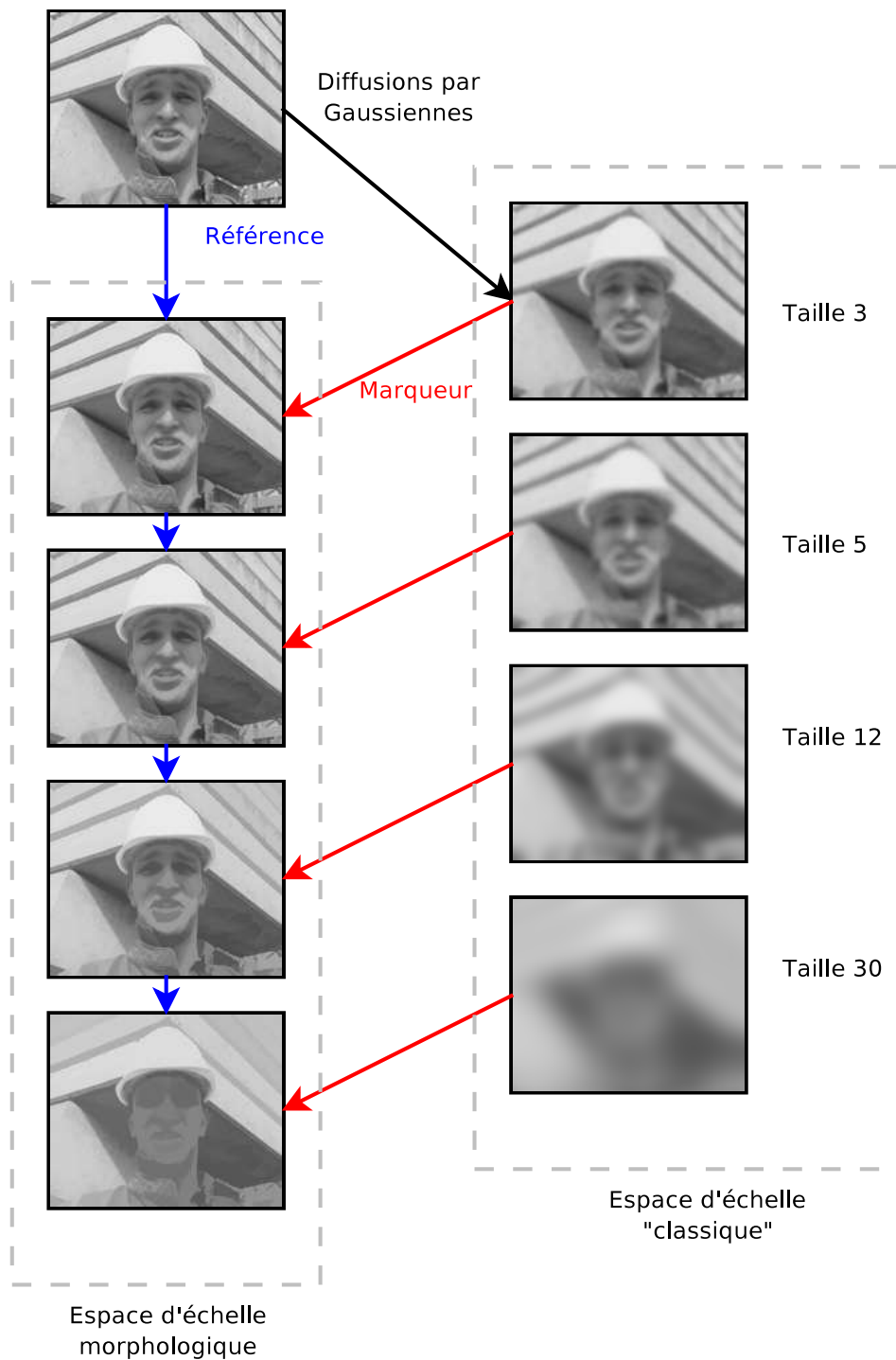


FIG. 6.6: Vue complète de l'algorithme d'espace d'échelle morphologique. Les images de l'espace d'échelle classique servent de marqueurs aux nivellements qui ne reconstruisent donc que des éléments de plus en plus gros. L'image obtenue au niveau précédent sert de marqueur au nivellement courant, ce qui garantit la croissance des zones plates.

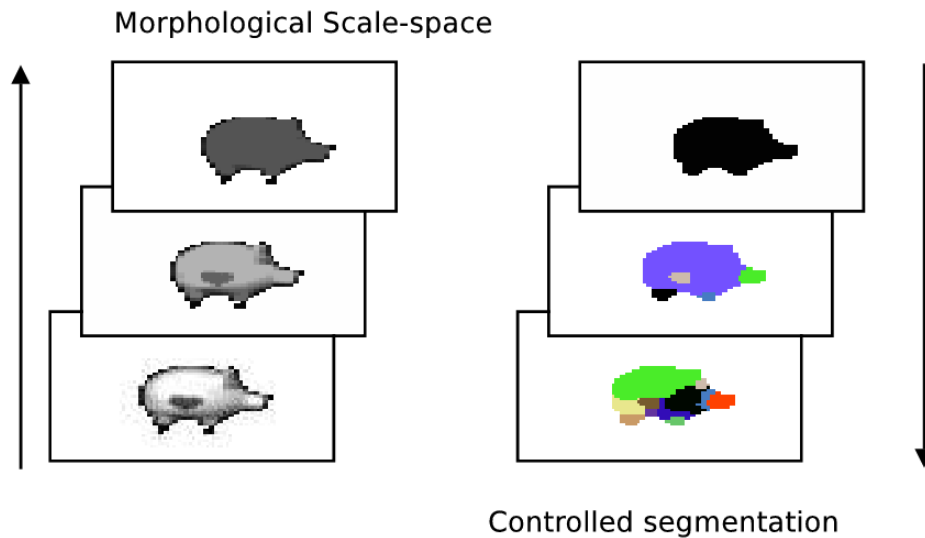


FIG. 6.7: Chaque segmentation est une segmentation de contrôle pour celle au niveau plus fin

segmentation agissant comme une segmentation de contrôle pour celle à l'échelle plus fine (voir figure 6.7).

La structure arborescente formée par la pyramide des (re)segmentations a une structure de dendrogramme : à chaque niveau, pour chaque région resegmentée, les régions correspondantes au niveau plus fin forment une partition de la région au niveau supérieur.

Le point critique de cette approche est que chaque contour visible à un niveau grossier de la pyramide correspond à un contour au niveau le plus bas, c'est-à-dire l'image brute. Ainsi, en détectant les contours aux niveaux supérieurs, on peut d'ores et déjà sélectionner les contours représentant des objets de la façon la plus durable.

6.4.2 Segmentation «top-down» hiérarchique

Avec l'algorithme précédent utilisé sans précaution, une image réelle sera largement sur-segmentée, et ce dès les plus hauts niveaux de la pyramide.

Plusieurs solutions sont possible pour pallier ce problème. Dans le cas d'une LPE classique, on utilise fréquemment la segmentation hiérarchique pour fusionner astucieusement les régions les moins significatives. Cette opération est effectuée traditionnellement en une passe d'inondation dans laquelle on construit un graphe, suivie d'une opération sur le graphe et enfin une passe d'affichage des composantes connexes du graphe ([Meyer, 1994]).

Il est facile de combiner les deux variantes de la LPE : la resegmentation d'une part et la segmentation hiérarchique d'autre part. En effet, on peut construire le graphe de façon identique en ajoutant la contrainte de resegmentation de façon indépendante (voir algorithme 12).

En partant de ce graphe, il est trivial de calculer la forêt couvrante de poids minimal et de là implémenter les schémas traditionnels de segmentation morphologique hiérarchique. L'intérêt extrême de cette approche est que comme les composantes connexes du graphe initial respectent la segmentation de contrôle et que les opérations de fusion consistent exclusivement à retirer des arêtes, alors toutes les hiérarchies issues de ce graphe respecteront la segmentation de contrôle.

Le choix du critère de fusion est ouvert. Lorsqu'on construit l'espace d'échelle avec des gaussiennes (suivies d'un nivellement), il y a un lien entre le niveau dans la pyramide et la taille ainsi que

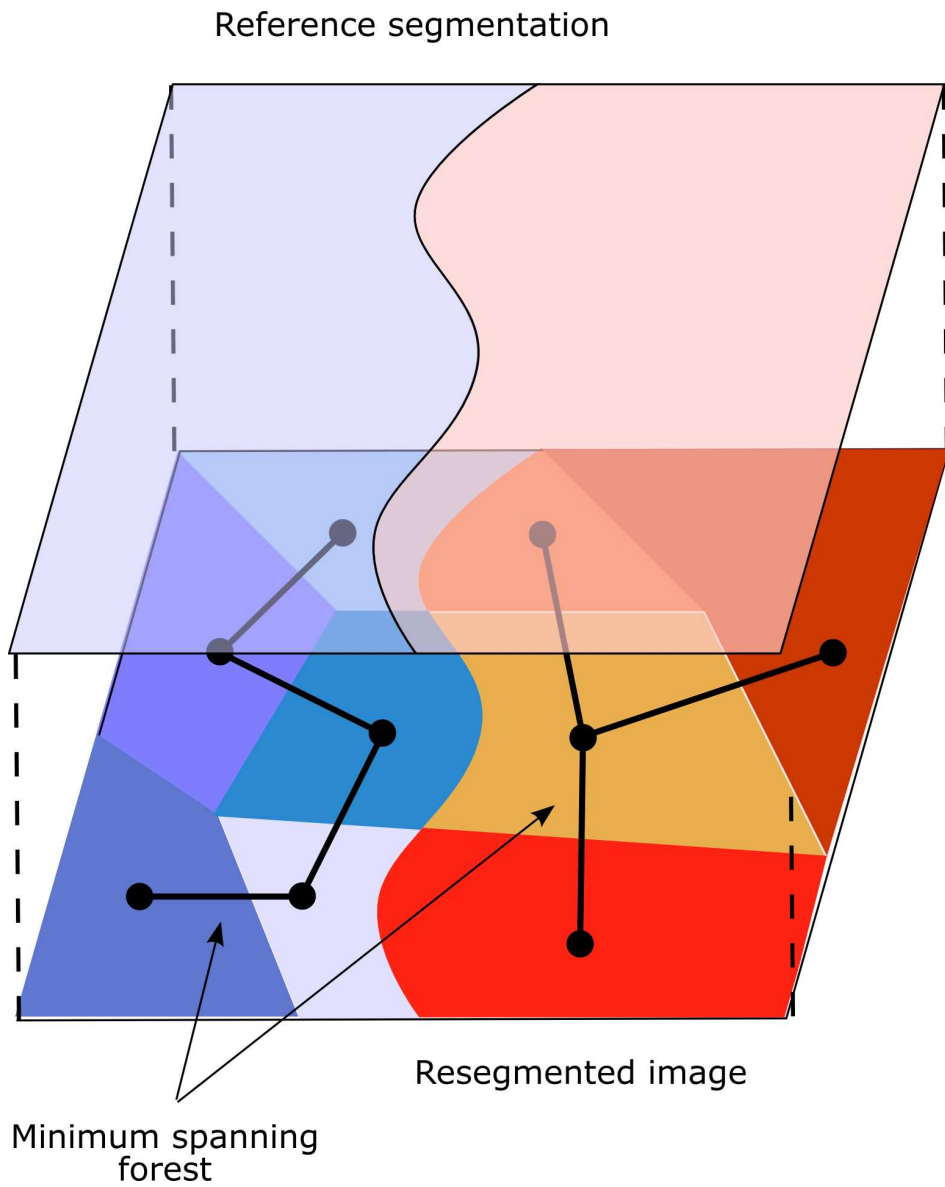


FIG. 6.8: Forêt de voisinage d'une image sous une segmentation de référence

Algorithme 12 LPE hiérarchique contrôlée

```

Mettre les minima dans une FAH
Préparer le graphe, un nœud par minimum
tantque FAH non vide faire
   $p$  = sortir le pixel le plus bas
  pour tout  $q$  dans le voisinage de  $p$  faire
    si  $p$  et  $q$  sont dans la même région de référence alors
      si  $q$  n'est pas labellisé alors
         $q$  prend le label de  $p$ 
        mettre  $q$  dans la FAH
      sinon
        Ajouter une arête entre le label de  $p$  et celui de  $q$ 
      finsi
    finsi
  fin pour
fin tantque

```

la dynamique des structures préservées. Ainsi, on pourra piloter un critère surfacique ou volumique en fonction du niveau de la pyramide.

La figure 6.9 illustre le problème : la petite région rouge est segmentée par la ligne de partage des eaux. Le problème est qu'au niveau courant de la pyramide, on sait que les régions significatives ont une taille supérieure. On court donc le risque de conserver une région éventuellement parasite non seulement au niveau courant, mais aussi à tous les niveaux inférieurs, comme l'impose les propriétés des resegmentations.

Fusionner les régions jusqu'à ce que leur taille dépasse le seuil critique permet facilement de pallier ce problème.

6.4.3 Comparaison avec la segmentation hiérarchique «classique»

Comparons les résultats donnés par la segmentation «top-down» à ceux obtenus par une segmentation morphologique «classique» utilisant une hiérarchie basée sur un critère volumique² ([Meyer, 2001])

²Ce critère donnant les meilleurs résultats dans nos tests comparant les critères usuels : volume, surface et dynamique.

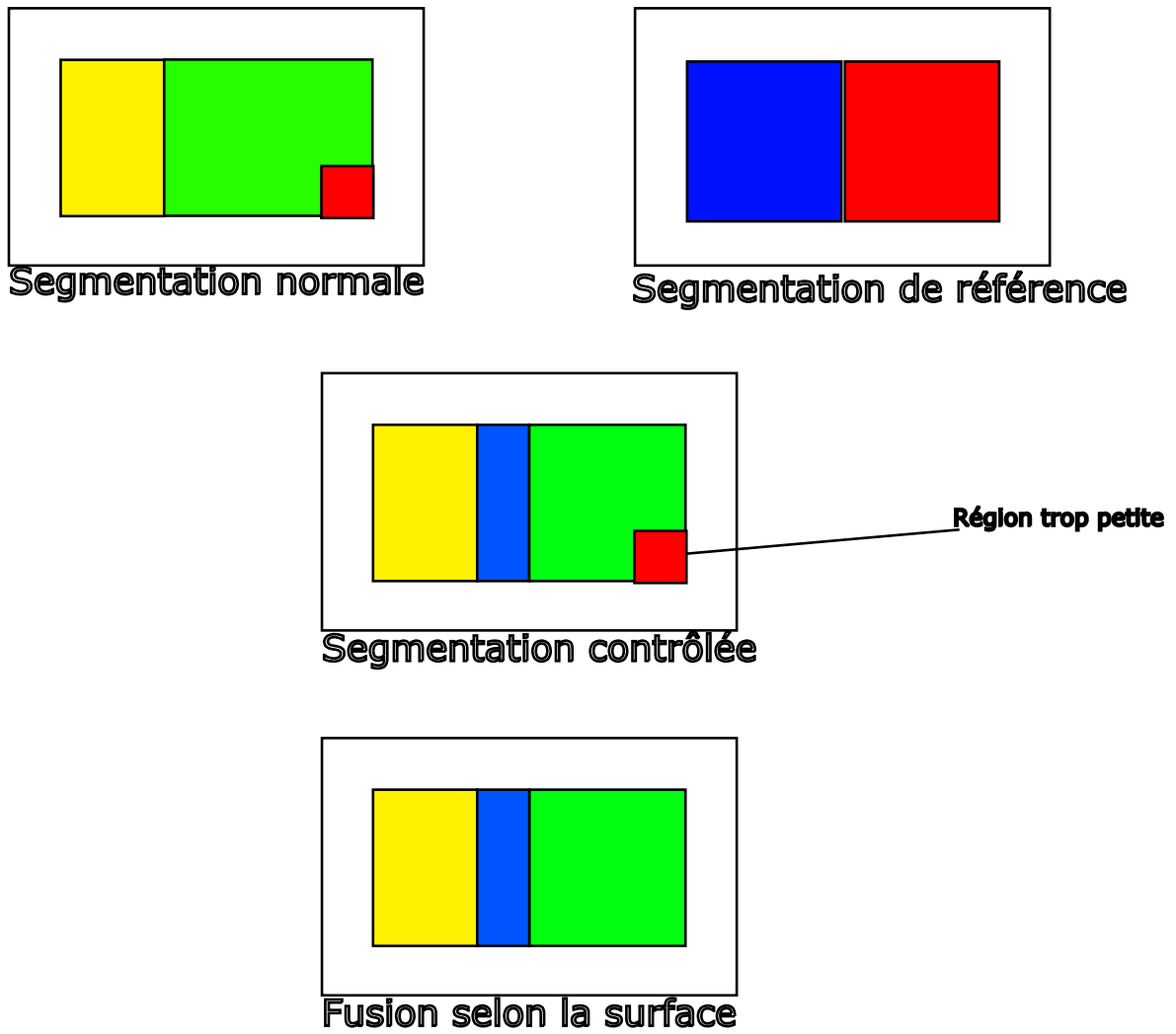


FIG. 6.9: La segmentation hiérarchique surfacique permet de supprimer des régions dont la taille est trop faible.



Nous comparerons les deux algorithmes sur l'image du tigre ci-contre (Source : Berkeley), qui s'avère difficile pour la segmentation hiérarchique à cause de la texture tigrée : les valeurs faibles du gradient localisées au niveau des rayures causent rapidement une corruption du contour. En effet les rayures formeront de petites zones de dynamique et de surface faibles qui fusionneront rapidement avec le fond.

On verra dans cet exemple simple que l'approche top-down permet de constituer une hiérarchie de région possédant des propriétés intéressantes. En particulier, on pourra remarquer une bonne capacité à extraire les structures importantes d'un objet, ainsi qu'une résistance prolongée aux petites structures moyennement contrastées. L'utilisation d'une pyramide de nivellements comme espace d'échelle permet en effet de sélectionner les contours selon à la fois leur force, mais également selon la taille des zones plates qu'ils délimitent.

Les résultats de la figure 6.10 sont remarquables car ils illustrent parfaitement la séparation entre les deux approches. Avec la segmentation «bottom-up» traditionnelle, on voit dès le niveau à 49 régions que les rayures commencent à se fondre avec le fond. La tête est déjà mal segmentée au niveau 97, et la forme générale du tigre est perdue à partir de 20 régions.

À l'inverse, la segmentation «top-down» conserve en permanence la forme du tigre. En effet, les rayures de l'animal ayant été lissées aux niveaux les plus hauts, le contour du dos est parfaitement reconstruit. Une fois ce contour imposé, il n'est plus possible que le fond envahisse l'objet, malgré un faible gradient au niveau des rayures.

Malgré tout, il reste clair que cette approche ne saurait remplacer complètement la segmentation hiérarchique traditionnelle, en particulier car celle-ci est d'une extrême rapidité. En revanche, on serait tenté d'utiliser cette nouvelle approche de façon très localisée pour modéliser des objets complexes, ou dans les cas où le facteur temps n'est pas critique.

6.4.4 Représentation arborescente

Le dendrogramme formé par la resegmentation suit une structure d'arbre avec comme racine la (ou les) régions au niveau le plus grossier. Les feuilles de cet arbres sont les régions de la segmentation la plus fine.

Plus en détail, on va garder dans la pyramide de resegmentations les régions du niveau le plus fin, et on ajoutera ensuite un nœud à chaque fois que deux régions fusionneront (donc à chaque disparition de région).

La figure 6.11 présente les niveaux supérieurs de cet arbre superposé aux régions de la pyramide de segmentations.

Les nœuds de l'arbre sont typiquement valués par la couleur moyenne de la région qu'ils représentent, la position de son barycentre, ainsi que sa surface et sa position dans la pyramide. Le chapitre suivant comparera des arbres de ce type dans une optique d'indexation par le contenu.

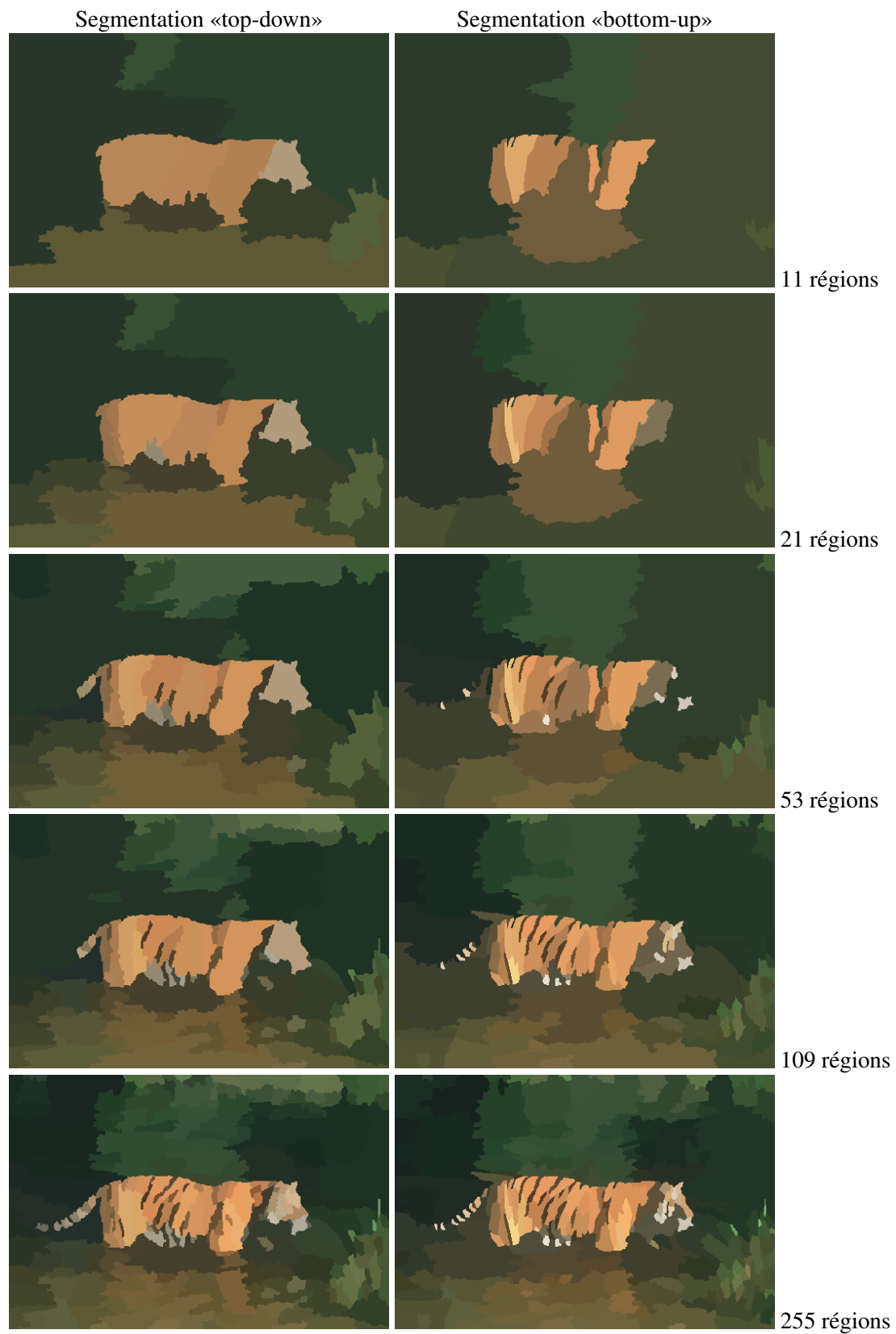


FIG. 6.10: Comparaison entre différents niveaux de la segmentation «top-down» à gauche et la segmentation hiérarchique «bottom-up» correspondante (en terme de nombre de régions) à droite. On voit nettement que l'allure générale du tigre est toujours bien détectée, ce qui n'est pas le cas avec l'approche «bottom-up».

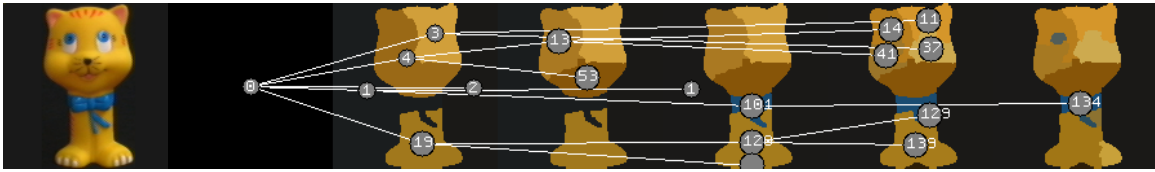


FIG. 6.11: Représentation arborescente du dendrogramme formé par les régions d'une pyramide de resegmentations.

6.5 Le gradient «pyramide»

Un des défauts de l'approche par resegmentation est son coût. En effet, il faut d'une part calculer l'espace d'échelle morphologique, et d'autre part calculer de nombreuses inondations par LPE. Or, le «paradigme de la segmentation morphologique» nous montre qu'une grande partie de l'information de l'image est synthétisée dans l'image gradient. Dans cette partie nous allons ainsi chercher à construire un gradient utilisant les propriétés intéressantes de l'espace d'échelle morphologique, mais sans payer le prix d'un grand nombre d'inondations. Ce gradient sera ensuite segmenté par une méthode «bottom-up» classique et rapide.

6.5.1 Calcul du gradient

La propriété la plus importante des espaces d'échelle morphologiques est qu'ils ne déplacent pas les contours. Ainsi, les contours d'un objet resteront à la même position tant que l'objet restera visible dans la pyramide. Nous allons donc pouvoir mesurer la force de chaque contour en regardant jusqu'à quel point il survivra dans la pyramide. En combinant ceci à une mesure directe de la valeur du contour, on obtient ainsi une mesure décrivant la force du contour dans l'image ainsi que sa «survivabilité».

En pratique, à chaque niveau i de la pyramide, on calcule un gradient morphologique simple, et on accumule ces gradients :

$$gradient_{pyramide}(Image) = \sum_i f(i) gradient_{basique}(Pyramide_i(Image))$$

Les $f(i)$ sont des poids associés à la hauteur dans la pyramide, afin de renforcer le rôle des contours les plus marqués. En effet, les nivellements ont la propriété de conserver la *position* des contours, mais réduisent malgré tout leur force. Nous avons utilisé $f(i) = i$ et normalisé la somme finale pour obtenir les résultats de la figure 6.12. Le gradient de base utilisé est le gradient «écart-type» décrit dans le chapitre 5.

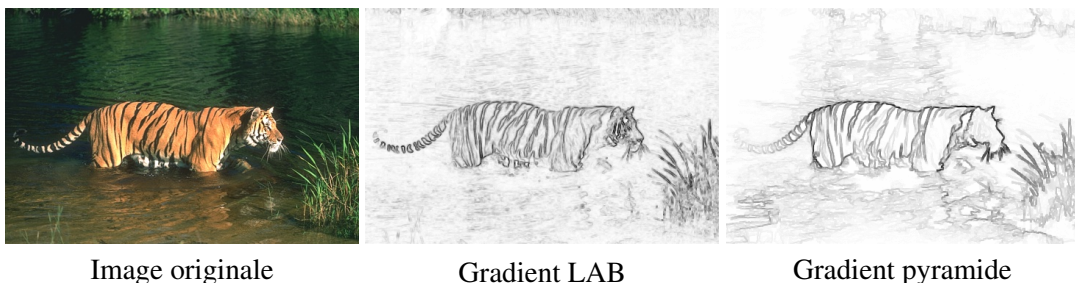


Image originale

Gradient LAB

Gradient pyramide

FIG. 6.12: Gradient pyramide comparé à un gradient LAB de référence. Les détails sont moins marqués, mais les contours des grandes structures sont bien renforcés.

6.5.2 Comparaison numérique

Pour comparer de façon objective notre gradient pyramide à ceux vus précédemment, nous allons employer le même protocole de test que celui du chapitre 5, basé sur l'image Lena.

La figure 6.13 compare l'évolution du gradient pyramide en fonction du bruit à celle du gradient «écart-type». On voit que la mesure du PSNR n'est pas favorable au gradient pyramide, ce qui confirme l'impression visuelle qui se dégage des images gradient.

En revanche, les choses changent dès lors que l'on calcule la ligne de partage des eaux de ce gradient à partir des marqueurs définis précédemment. On voit en effet sur la figure 6.14 que la segmentation basée sur le gradient pyramide diverge beaucoup moins que les autres en présence d'un fort niveau de bruit. Ceci est confirmé visuellement par les images de la figure 6.15.

De plus, dans le cas extrêmement pathologique de la figure 6.16, on voit que le gradient pyramide est peu affecté par la grande quantité de bruit. L'augmentation de la distance au contour de référence par rapport aux résultats précédents est directement liée au déplacement de la zone droite du contour. En effet, la ligne de partage des eaux de l'image très bruitée s'est attachée à la ligne du visage plutôt qu'à la chevelure. Ceci s'explique par le fait que celle-ci présente une zone de contour extrêmement étroite, rapidement corrompue par le bruit. A l'inverse, la ligne du visage représente un contour entre deux zones de grande taille : le visage et les cheveux. Ces zones seront préservées jusqu'aux niveaux élevés de la pyramide de nivellements et renforceront donc le gradient pyramide à cet endroit. Inversement, la mèche formant le contour de la chevelure, déjà affaiblie par le bruit, sera lissée et disparaîtra plus rapidement.

Ce comportement du gradient pyramide est extrêmement intéressant, car il permet de combiner une notion de taille avec une notion de contraste et de sélectionner en conséquence les objets les plus proéminents par la taille aussi bien que par la dynamique.

On notera au passage la preuve de la faiblesse des mesures intermédiaires : les résultats de la LPE sont meilleurs avec le gradient pyramide, alors que son score de PSNR par rapport au gradient de référence est plus faible. Ceci est une illustration du caractère non monotone du watershed, en tout cas relativement à la mesure du PSNR : ce n'est pas parce qu'une image gradient sera proche en terme de PSNR que le résultat de la LPE sera proche lui aussi.

Il faut donc, quand cela est possible, mesurer et optimiser le résultat définitif plutôt que les étapes intermédiaires qui ne tiennent pas compte de tous les aspects non linéaires ou non monotones.

6.5.3 Segmentation hiérarchique

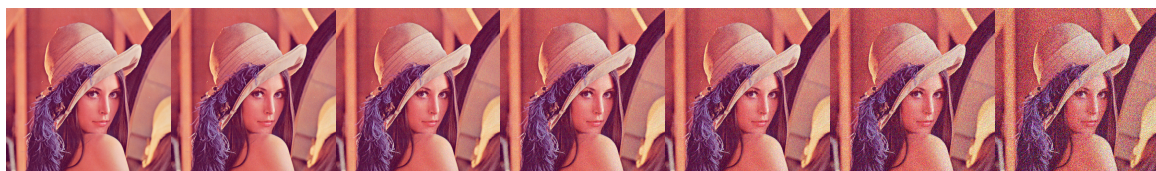
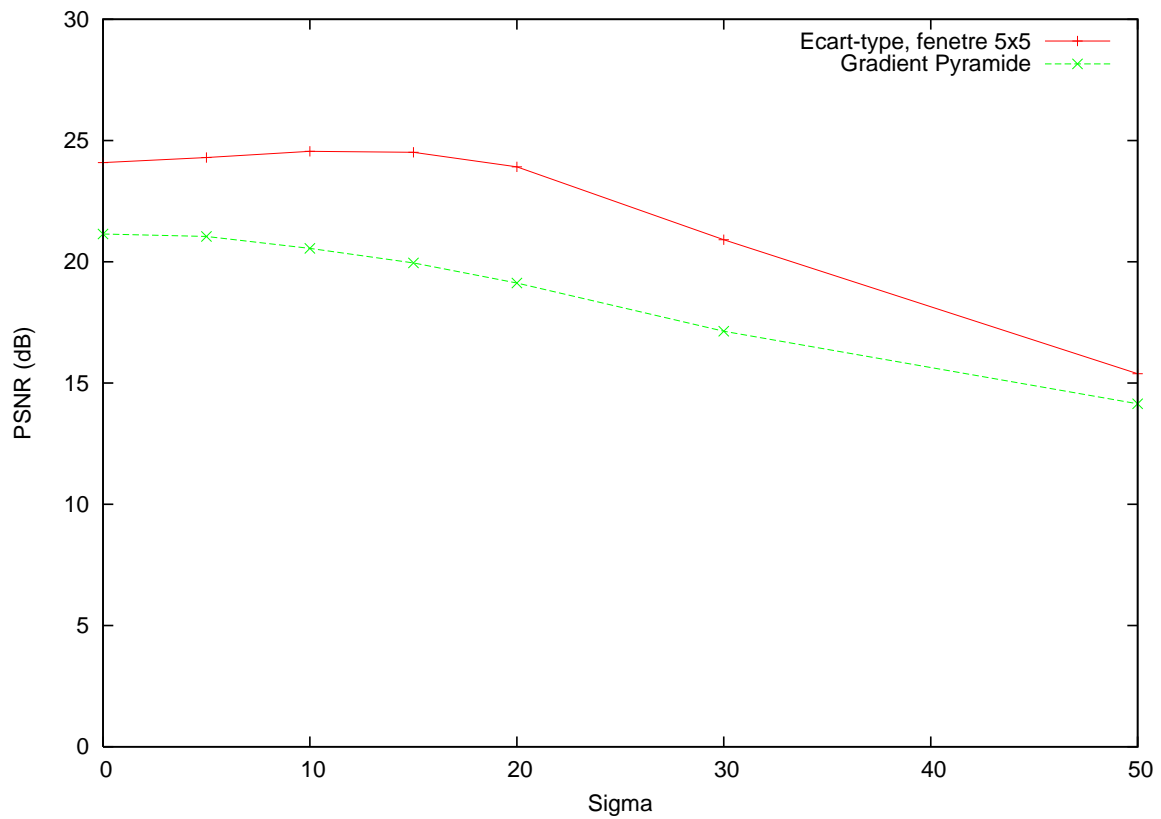
La figure 6.17 présente les résultats d'une segmentation hiérarchique avec critère volumique d'une image de la base de Berkeley avec un gradient classique et avec le gradient pyramide.

On voit qu'au niveau le plus fin (1000 régions), le gradient classique détaille mieux les petites structures, notamment autour de la tête du tigre.

Au niveau suivant (100 régions), les faiblesses locales du gradient classique apparaissent et on voit que certaines zones du tigre commencent à se fondre dans le fond (queue, partie inférieure de la tête, etc).

Il n'y a que peu de modifications entre 100 et 50 régions du côté du gradient pyramide, les fusions s'opèrent principalement dans les petites zones de détails (herbes sur le côté droit, queue, tête et rayures).

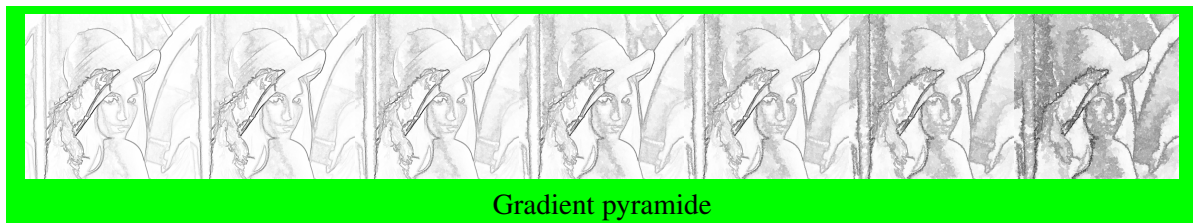
Au dernier niveau, la tête du tigre n'est plus visible avec l'algorithme classique, alors que la silhouette est globalement préservée avec le gradient pyramide.



Original $\sigma = 5$ $\sigma = 10$ $\sigma = 15$ $\sigma = 20$ $\sigma = 30$ $\sigma = 50$



Gradient «écart-type», fenêtre 5x5



Gradient pyramide

FIG. 6.13: Comparaison du gradient pyramide et du gradient «écart-type», relativement au gradient de Beucher de l'image non bruitée servant de référence.

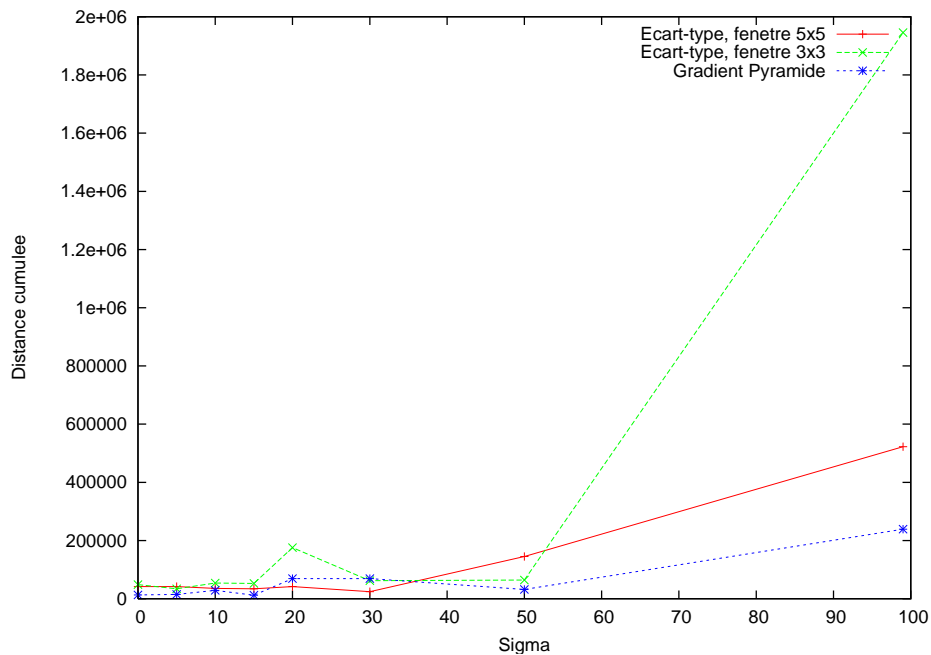


FIG. 6.14: Évolution des distances au contour de référence quand le bruit augmente.

Enfin, un aspect intéressant est celui présenté dans la figure 6.18 qui confronte les deux méthodes utilisant les espaces d'échelle morphologiques : l'une utilisant les resegmentations «top-down» et l'autre basée sur le gradient pyramide. On voit principalement deux choses : en premier lieu, les segmentations en un faible nombre de régions sont extrêmement semblables et renvoient correctement l'objet principal. D'un autre côté, la segmentation en un grand nombre de régions (255) est plus fine avec la méthode «top-down» qu'avec le gradient pyramide. Ceci s'explique par le fait qu'on recherche explicitement dans ce dernier à réduire les contours associés aux petits objets, éventuellement de façon excessive. Il serait intéressant de tester l'impact de plusieurs jeux de pondérations sur ces résultats, déjà largement satisfaisants.

6.6 Conclusion

Nous avons présenté ici une variante inédite de l'algorithme de LPE qui est complémentaire avec de nombreux algorithmes d'inondation et qui peut leur être associée sans nécessiter de changements importants.

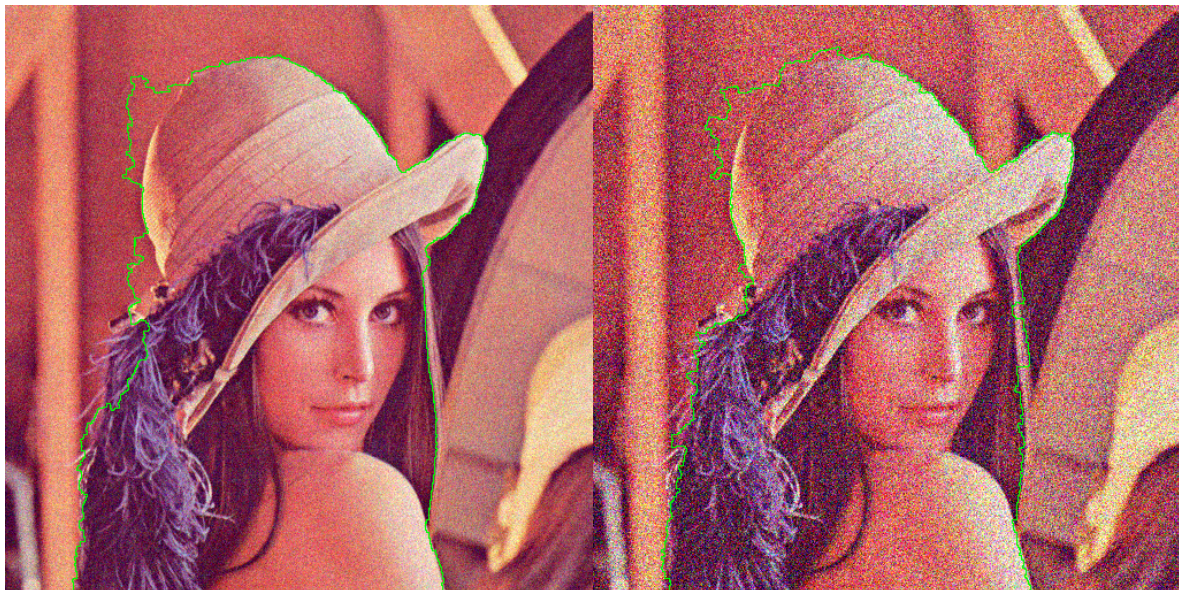
La complexité de l'algorithme de LPE contrôlée est identique à celle de la LPE normale, et celle de l'espace d'échelle est comparable à d'autres approches du même ordre. En effet, la complexité d'un nivellement est en $O(n)$, soit l'équivalent d'une étape de filtrage par une gaussienne, une ouverture, un filtre médian, etc. De plus, la LPE contrôlée étant compatible avec les schémas classiques de «nettoyage» de la LPE (fusions hiérarchiques, etc), on peut obtenir des résultats particulièrement robustes et faiblement paramétriques, en particulier en utilisant la hauteur dans la pyramide pour en déduire la dimension caractéristique des objets à chaque niveau et s'en servir pour piloter une fusion hiérarchique à chaque niveau.

Par ailleurs, cette approche se prête globalement bien à l'injection de connaissance extérieure, en imposant une segmentation initiale (au niveau le plus grossier). Ceci permet par exemple d'utiliser une



Contour de référence

Gradient pyramide (image non bruitée)

 $\sigma = 20$, gradient pyramide $\sigma = 50$, gradient pyramide**FIG. 6.15:** Comportement du gradient pyramide lorsque le bruit augmente.



Gradient «écart-type», fenêtre 5x5

Gradient pyramide

FIG. 6.16: Résultats difficiles pour $\sigma = 99$, mais meilleurs avec le gradient pyramide (note : pour la rendre plus visible, nous avons épaissi la ligne de contour).

information issue du mouvement ou d'une comparaison avec l'image de référence pour imposer un contour bien défini. L'algorithme de resegmentation d'espace d'échelle étant alors utilisé pour obtenir une signature de l'objet.

Enfin, avec le gradient pyramide nous avons présenté une troisième voie entre l'approche «bottom-up» standard et notre approche «top-down» qui préserve certaines des bonnes propriétés de cette dernière, tout en étant beaucoup plus rapide. Il sera intéressant de comparer les résultats de ce gradient en utilisant d'autres lissages (par la dynamique, par la taille) ou d'autres types nivellements.

Le chapitre suivant s'inscrit dans la continuation de celui-ci en utilisant la technique de segmentation décrite ici pour modéliser et comparer des objets. On y présentera notamment une distance entre les arbres et des applications dans le domaine de l'indexation par le contenu.



FIG. 6.17: Comparaisons entre la segmentation hiérarchique du gradient pyramide (à droite) et celle d'un gradient LAB classique (à gauche).

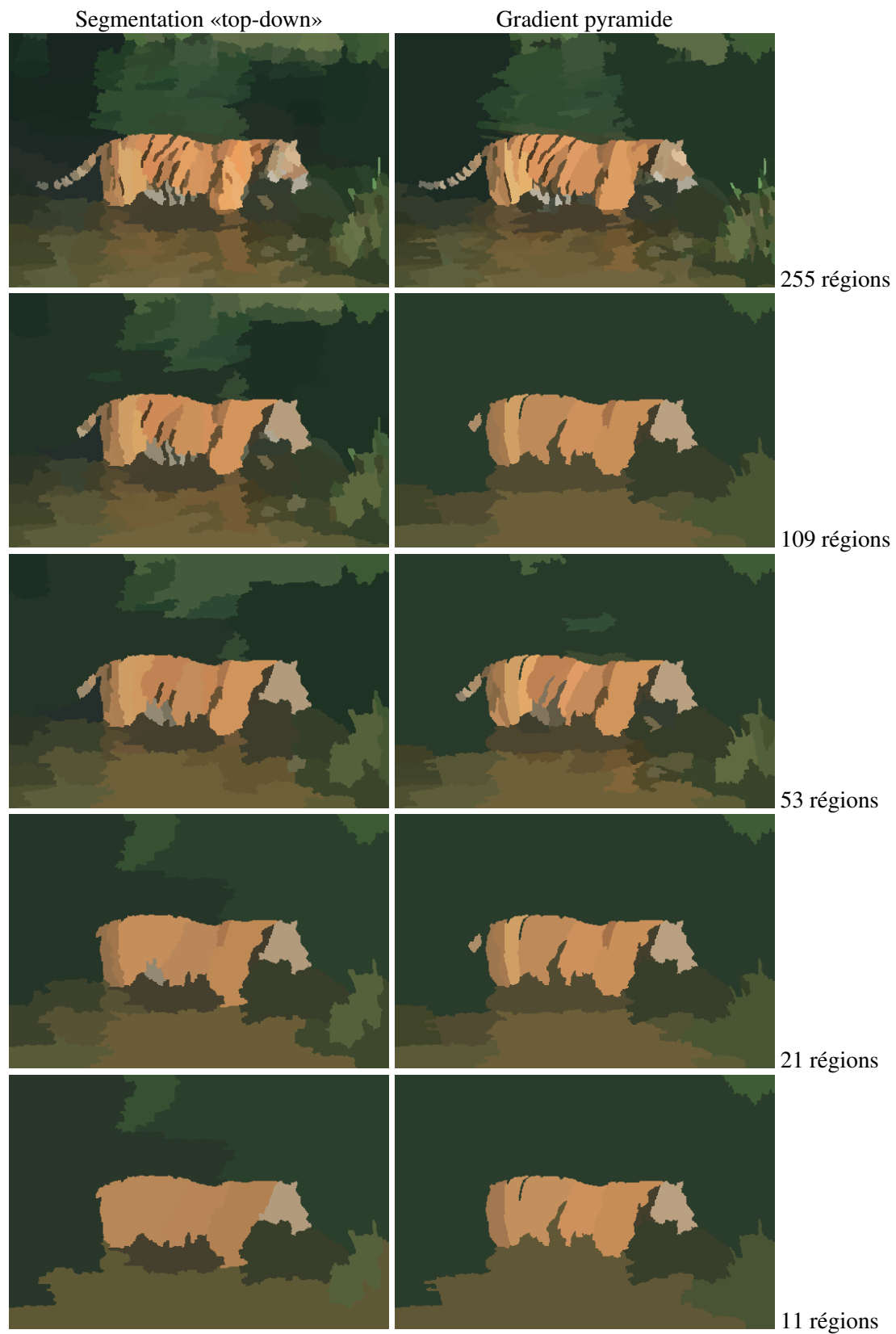


FIG. 6.18: Comparaisons entre la segmentation «top-down» (à gauche) et la segmentation hiérarchique du gradient pyramide (à droite).

Chapitre 7

Appariement d'arbres

Back to the trees !

Roy Lewis,
Pourquoi j'ai mangé mon père

Le chapitre précédent était consacré à la construction d'un arbre encodant de la façon la plus robuste possible la structure interne d'un objet et sa décomposition en sous-objets encastrés. Nous décrivons dans ce chapitre une méthode pour établir une distance entre deux arbres et nous comparerons cette approche à une approche basée sur des histogrammes dans le cadre de la recherche basée sur le contenu (CBIR).

Dans ce chapitre, nous testerons nos algorithmes sur une base d'images standard du domaine de l'indexation mais nous les comparerons à d'autres approches sur des images plus difficiles au chapitre suivant.

7.1 Distance d'arbres

7.1.1 État de l'art

La distance d'arbres doit être capable d'intégrer deux éléments distincts : d'une part une distance entre régions, basée sur les couleurs, la position, etc, et d'autre part profiter de la structure arborescente. Par ailleurs elle doit être relativement rapide à calculer.

Parmi les méthodes existant dans la littérature, trois catégories ont retenu notre attention. Nous avons déjà examiné dans le chapitre 3 certaines des méthodes dans nos recherches sur les algorithmes d'appariement de graphes. Les premières méthodes sont les méthodes combinatoires, par exemple [Shokoufandeh et al., 1999]. Ces méthodes sont en général d'une complexité excessive pour être utilisées en dehors de conditions extrêmement favorables (peu d'erreurs de segmentation, etc). Les secondes sont les méthodes spectrales, qui encodent chaque sous-arbre dans un vecteur de valeurs propres (voir par exemple [Siddiqi et al., 1999]). Encore une fois, l'intérêt de cette approche est son invariance à l'ordre de numérotation des sous-arbres. En revanche, si elle fonctionne correctement pour des isomorphismes, les résultats se dégradent quand on cherche des solutions approchées.

Les approches de la troisième classe sont les méthodes à noyaux. Ces méthodes décrites de façon extensive dans [Shawe-Taylor and Cristianini, 2004] sont connues pour être très rapides et efficaces

dans les cas où peut les exprimer de façon récursive. La récursivité est difficile à exprimer dans des graphes en général, mais facile dans le cas des arbres. Des noyaux pour les arbres ordonnés sont proposés dans le livre de Shawe-Taylor et Cristianini, et des noyaux pour arbres non ordonnés peuvent être décrits dans [Collins and Duffy, 2001] et [Micheli et al., 2005]

7.1.2 Construction

Le premier noyau que l'on peut construire sur des arbres est celui de Collins-Duffy, réutilisé par Micheli et qui compare des arbres en comptant le nombre de sous-arbres communs. On représente tout arbre t à m sous-arbres par le vecteur $[h_1(t), h_2(t), \dots, h_m(t)]$ où $h_i(t)$ est le nombre d'occurrences du sous-arbre indexé par i dans l'arbre t . Alors on peut définir le noyau suivant :

$$K(t_1, t_2) = \phi(t_1) \cdot \phi(t_2) = \sum_{i=1}^m h_i(t_1) h_i(t_2)$$

L'approche naïve nécessite d'énumérer les m sous-arbres et de calculer h_i pour tous les i de 1 à m . La complexité serait alors exponentielle. Fort heureusement, on peut calculer ce noyau de façon récursive sur la profondeur des sous-arbres. La complexité est alors en $O(n_1 n_2)$, le produit de la taille des arbres.

Par analogie, notre premier noyau sera défini ainsi :

$$K_1(T_1, T_2) = K_{direct}(T_1, T_2) + \alpha \sum_{\substack{t_1 \in Ch(T_1) \\ t_2 \in Ch(T_2)}} K(t_1, t_2)$$

où $K_{direct}(T_1, T_2)$ est la similarité entre les régions (critères géométriques, couleur, etc) et α est un paramètre réglant l'importance relative donnée au sous-arbre par rapport à la comparaison directe entre régions. On note également $Ch(N)$ les nœuds enfants directs du nœud N .

Un point important à noter est que la contribution des sous-arbres à la profondeur d est donc multipliée par α^d , ce qui est similaire à la pondération proposée par Micheli. Ceci signifie que pour $\alpha \in [0; 1[$ l'importance des nœuds du bas de l'arbre est faible.

L'inconvénient de cette approche est qu'elle combine les similarités (ou le manque de similarités) entre tous les couples de sous-arbres. Une amélioration serait donc de prendre non pas la somme des similarités $K(t_1, t_2)$ entre tous les couples (t_1, t_2) possibles, mais de prendre la somme des plus grandes similarités pour chaque t_1 :

$$K_2(T_1, T_2) = K_{direct}(T_1, T_2) + \alpha \sum_{t_1 \in Ch(T_1)} [\max_{t_2 \in Ch(T_2)} K(t_1, t_2)]$$

L'inconvénient est que ce noyau n'en est plus un, car le «max» n'est pas linéaire. Par ailleurs, la formulation n'est plus symétrique. Pour retrouver cette propriété, on peut prendre la somme : $K(T_1, T_2) = K_2(T_1, T_2) + K_2(T_2, T_1)$. Toutefois, dans la mesure où nous n'utilisons pas les propriétés des noyaux liées à l'apprentissage, l'aspect non linéaire n'est pas un réel problème et nous utiliserons ce «noyau» pour estimer des distances entre des arbres.

L'implémentation de cette mesure se fait par récurrence sur la hauteur des sous-arbres : pour deux sous-arbres terminaux, alors calculer K_2 revient à calculer la similarité entre régions. Pour la récurrence, il suffit de connaître les valeurs de K_2 pour tous les sous-arbres de hauteur d pour construire celles pour les sous-arbres de hauteur $d + 1$.

Un problème de cette approche est qu'elle ne considère pas, ou mal, les variations de profondeur : un motif de l'arbre T_1 devra être présent dans l'arbre T_2 à la même profondeur. Ceci est une limitation qui semble répandue dans la plupart des noyaux que nous avons examinés. Pour contourner ce problème, nous proposons de comparer les sous-arbres non seulement entre eux, mais aussi avec leurs parents¹.

$$K_3(T_1, T_2) = K_{direct}(T_1, T_2) + \alpha \sum_{t_1 \in Ch(T_1) \cup \{T_1\}} [\max_{t_2 \in Ch(T_2) \cup \{T_2\}} K(t_1, t_2)]$$

Il est peut-être possible d'approximer le faux noyau K_2 par de véritables noyaux linéaires du type de K_1 en utilisant une convergence du type «softmax», comme cela a été fait pour les problèmes de couplage du chapitre 3. Cependant, l'intérêt de présenter les noyaux comme des produits scalaires n'apparaît que dans une seconde étape, les problèmes d'apprentissage. Dans la mesure où ceci n'est pas encore notre objectif, et dans l'attente d'une solution académiquement optimale, nous testerons notre approche en pratique plus efficace pour notre problème actuel.

7.2 Recherche d'images semblables dans une base

7.2.1 Détails techniques

Nous nous plaçons dans le contexte de la recherche d'image basée sur le contenu (CBIR), c'est à dire le problème d'extraire d'une base d'images celles qui se rapprochent le plus de l'image requête. La base choisie est la base COIL de l'université de Columbia, une des références les plus utilisées dans le domaine (figure 7.1).

La modélisation sera effectuée par l'approche décrite plus haut à base de resegmentation d'une pyramide de nivellements. La comparaison des arbres sera effectuée par l'algorithme décrit dans la section précédente.

Le *modus operandi* est le suivant : pour chaque image de la base, on va la comparer à toutes les autres images de la base, ainsi qu'à une image du même objet, mais sous un angle différent. Les résultats seront affichés triés par ordre de pertinence.

7.2.2 Résultats

La figure 7.2 présente le résultat de la recherche de quelques objets dans la base COIL.

Comme on le voit, les résultats sont bons dans l'ensemble. La seconde vue de l'objet est toujours citée en premier, ce qui prouve une certaine robustesse face aux petites perturbations. Ceci est à contraster avec les problèmes relevés dans le chapitre 3 avec l'utilisation de l'algorithme spectral.

En particulier, on voit dans le cas des objets rouges (la tomate en première position et la boîte en dernière position) que notre méthode a bien différencié les objets plutôt uniformément rouges (avec la tomate) de ceux rouges et blancs. De la même façon, le texte rouge sur la boîte jaune de «Dristan Cold» a favorisé la voiture jaune avec son intérieur rose. Cette voiture qui a correctement été séparée de la voiture jaune clair à flammes rouges.

Ces résultats montrent une bonne utilisation de la structure des objets pour les comparaisons. Cependant, on peut d'ores et déjà repérer quelques pistes d'amélioration. Parmi les caractéristiques que l'on peut relever :

¹Par souci de lisibilité, on n'explicite pas dans la formule suivante que le couple $(t_1, t_2) = (T_1, T_2)$ est exclu de la recherche : il faut qu'il y ait au moins un descendant dans le couple, sinon on boucle indéfiniment.



FIG. 7.1: Les vues de face des objets de la bibliothèque COIL. Chaque objet est ensuite photographié selon chaque angle (72 vues à 5° d'écart).

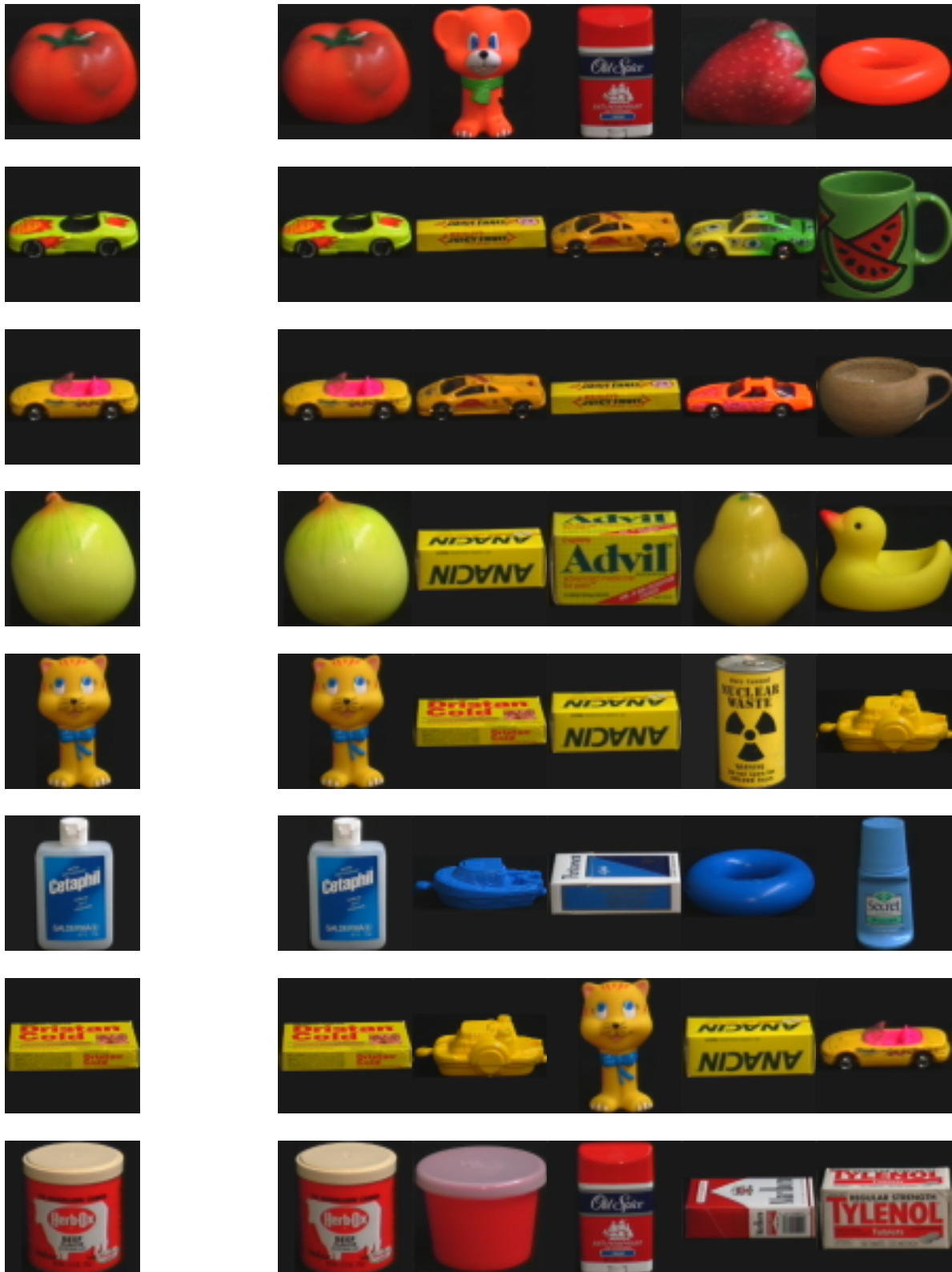


FIG. 7.2: Recherche d'images dans une base de données : chaque image de la colonne de gauche est comparée à toutes les images de la base COIL. Les cinq meilleures correspondances sont affichées dans la colonne de droite par ordre de pertinence. L'algorithme utilisé est celui de la pyramide de nivellements et de la comparaison des arbres de hiérarchies

- **Pas de correspondance de taille** : les derniers résultats de l'appariement des voitures (les tasses) sont portés par les appariements des petites structures (les pastèques dans le premier cas, avec les flammes sur la voiture). Ajouter une comparaison des tailles respectives des régions pénaliserait ces objets, au profit d'objets plus proches en taille (le bateau jaune, etc). Inversement, nombre de problèmes nécessitent d'être invariant par la taille, donc il faudrait trouver un juste milieu, comme prendre la fraction surfacique par rapport au nœud parent.
- **Pas de correspondance de position** : ni de position relative. Par exemple dans le dernier cas le couvercle de la boîte est apparié à la partie blanche du paquet de cigarettes, alors que l'un est vers le haut de l'objet et le second est vers la droite. Encore une fois, cela peut être une fonctionnalité importante. Il paraît plus complexe de l'incorporer sans perdre l'invariance par rotation. De plus, certaines de nos applications nécessitent une invariance par symétrie (vidéo-surveillance).

La section suivante sera consacrée à la comparaison avec une approche plus traditionnelle basée sur des histogrammes. Ce contrepoint illustrera l'importance de l'approche par régions ainsi que celle du traitement structuré, par rapport à une approche qui ne considère pas les corrélations entre les valeurs de pixels voisins.

7.2.3 Comparaison avec d'autres algorithmes

Une étape intéressante est de comparer cet algorithme à d'autres approches cherchant à répondre au même problème. Par exemple, la comparaison d'histogrammes. Il s'agit d'une approche simple, rapide et qui présente beaucoup des bonnes propriétés énoncées plus haut.

L'algorithme employé consiste à calculer un histogramme couleur sur toute l'image pour l'image requête et pour chaque image de la base, et à les comparer. L'échantillonnage le plus utilisé dans la littérature est de diviser l'espace RGB (en général) en 16 cellules par canal. On calcule ensuite une distance entre les histogrammes soit purement intra-cellules, soit mixte. Angulo et Serra présentent dans [Angulo and Serra, 2002] un état de l'art des méthodes de comparaison les plus utilisées.

Comme on le voit dans les figures 7.2 et 7.3, les résultats ne sont pas identiques.

L'explication principale est que l'approche par histogramme est purement globale, alors que celle basée sur les arbres a un aspect local plus prononcé. En effet, l'échantillonnage effectué sur les histogrammes couvre indifféremment toute l'image, alors que les segmentations effectuées aux différents niveaux de la pyramide engendrent une forte localité de l'information.

Par exemple, dans le cas du premier objet, la tomate, on voit que l'approche par graphe va préférer un objet globalement rouge plus clair (l'ourson) alors que l'approche par histogramme va préférer un objet possédant des zones du même rouge, mais entremêlées de blanc (la boîte de Tylenol).

Il en va de même pour la bouteille de shampoing bleu et blanche. Avec notre algorithme, les différents bleus sont fusionnés dans les niveaux élevés de la pyramide et sont par la suite correctement associés aux autres objets bleus. En particulier, le second meilleur appariement ne contient pas de blanc. À l'inverse, dans l'approche par histogrammes, les bleus sont dispersés dans plusieurs classes et ce sont plutôt les pixels blancs qui jouent dans les comparaisons.

Cette comparaison illustre principalement l'importance de considérer la couleur globale d'une région, et non pas seulement la valeur des pixels qui la composent. Notre approche est également relativement robuste face au bruit et au tramage.



FIG. 7.3: Recherche d'images dans une base de données : algorithme traditionnel basé sur la comparaison d'histogrammes couleur.

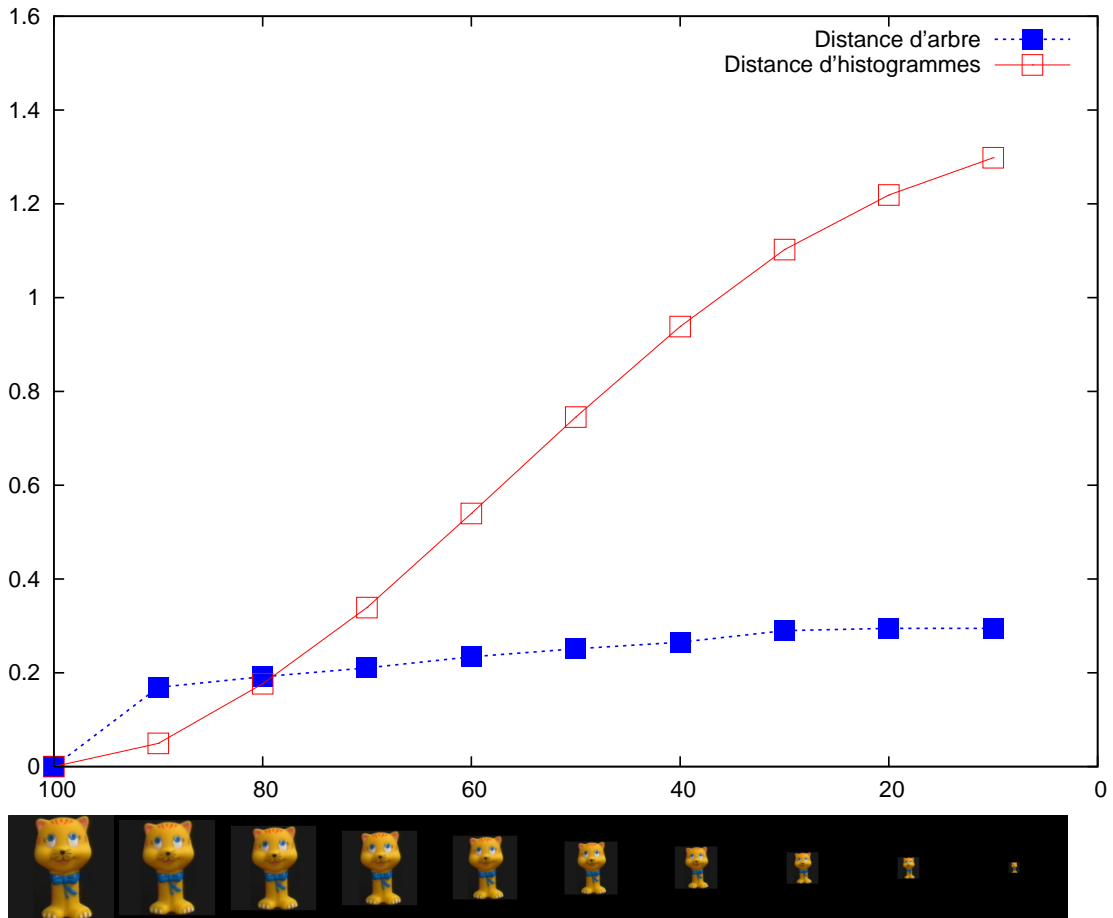


FIG. 7.4: Distances entre l'image à l'échelle normale et l'image réduite, calculée pour les arbres et les histogrammes.

7.3 Robustesse face aux perturbations

Une modélisation n'a que peu de valeur si elle n'est pas robuste. Les sections qui suivent vont tenter d'estimer les propriétés d'invariance de notre approche face aux rotations et aux changements d'échelle, puis ensuite nous testerons la robustesse face au bruit.

Les résultats précédents nous conduisent à nous interroger sur la variabilité de la modélisation quand on observe l'objet sous tous ses angles. Va-t-on pouvoir construire une signature de cet objet qui reste efficace face aux rotations ? Et aux changements d'échelle ? Pour ce faire, nous allons effectuer une comparaison entre une vue de référence et plusieurs autres vues sous différents angles et facteurs d'échelle.

7.3.1 Changements d'échelle

La figure 7.4 présente l'évolution des distances entre un objet à l'échelle 100% et cet objet vu à différents facteurs de réduction. On voit que la distance se stabilise rapidement autour de 0.2 et reste constante jusqu'à un niveau très bas. Ceci est dû au fait que malgré la faible information disponible, l'appariement retrouve dans les niveaux les plus grossiers de la pyramide de nivellement un objet semblable au résultat de la contraction. (voir figure 7.5).

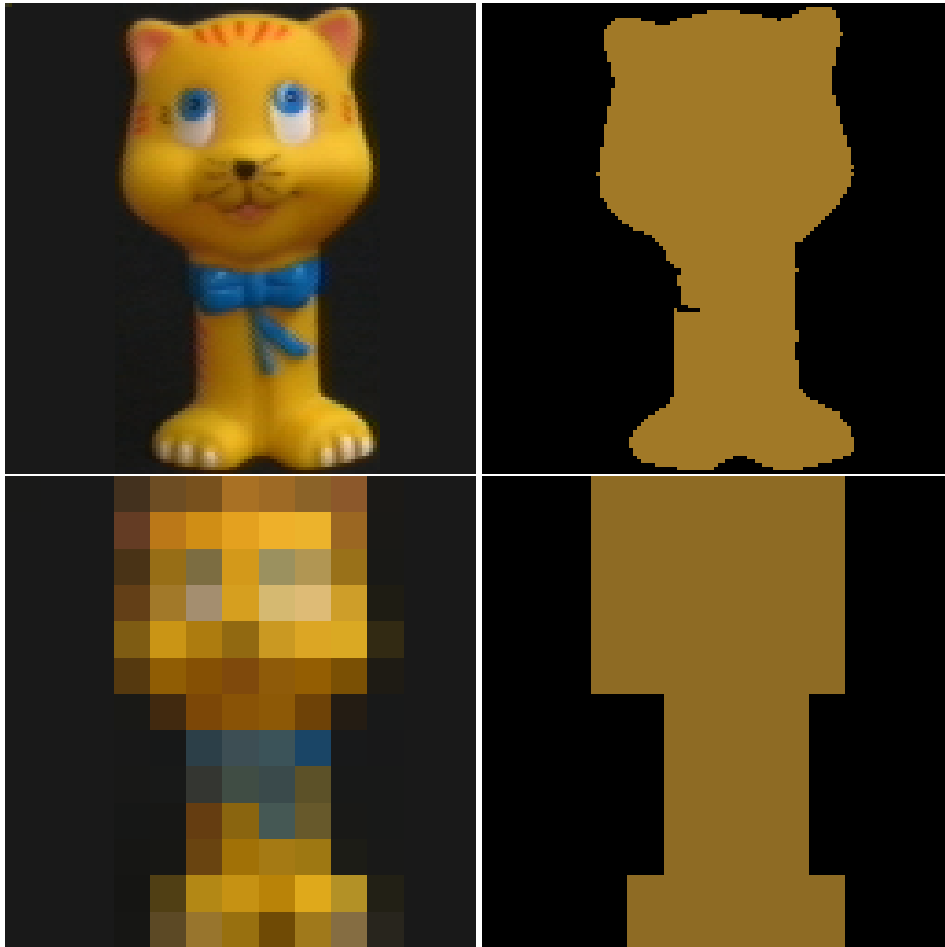


FIG. 7.5: Comparaison entre l'image du haut de la pyramide pour l'objet à l'échelle 100% et l'objet à l'échelle 10%. Il paraît donc évident que les nœuds racine des arbres de hiérarchies se ressembleront, si l'on exclut une comparaison des aires.

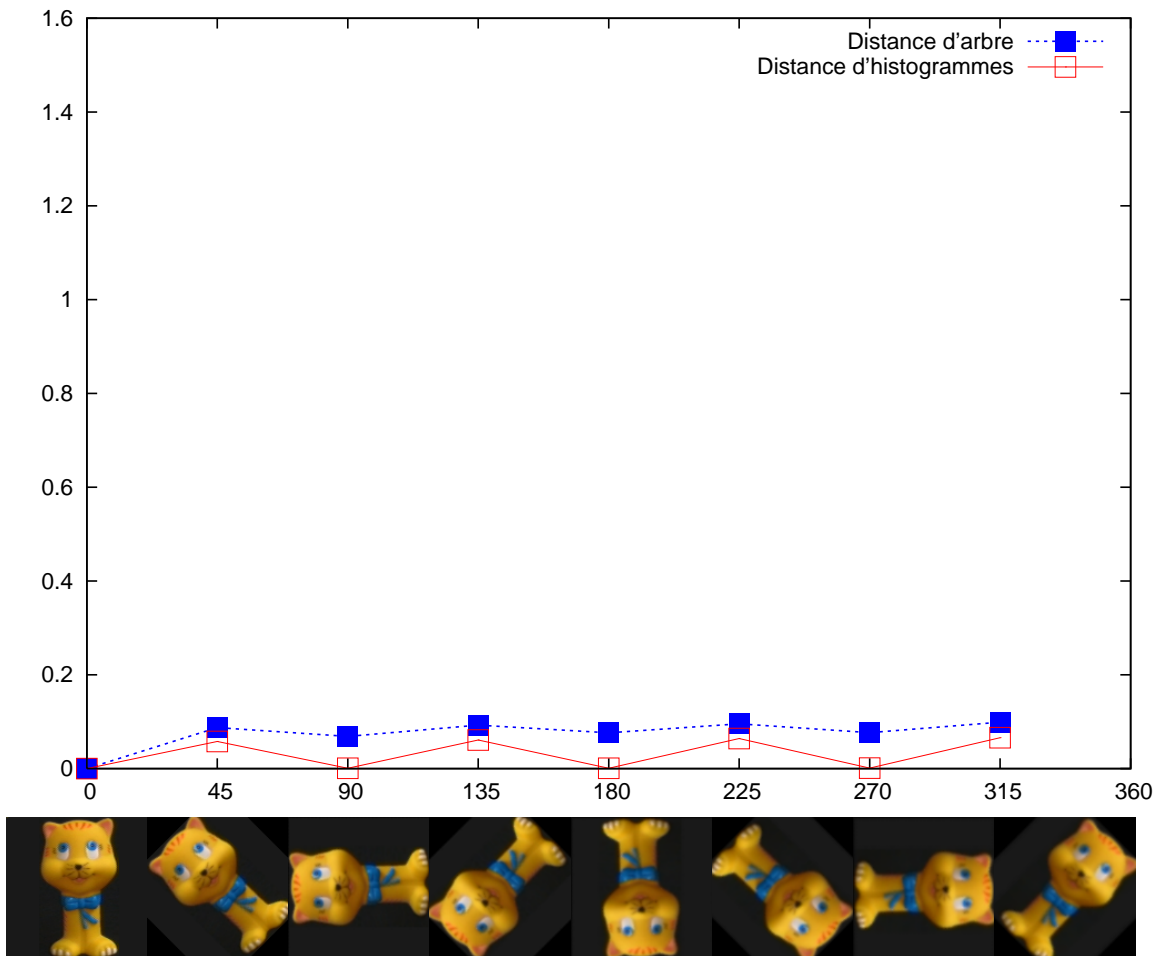


FIG. 7.6: Distances entre l'image à l'échelle normale et l'image en rotation, calculée pour les arbres et les histogrammes.

7.3.2 Rotation dans le plan XY

Comme on le voit dans la figure 7.6, la distance évolue peu entre l'image de référence et les images ayant subi une rotation. Ceci s'explique bien par le fait que la segmentation est relativement isotrope² et donc ne dépend pas de l'orientation de l'objet.

On remarque par ailleurs un bon résultat de l'approche par histogramme : la distance est en effet nulle sur les angles droits (90, 180, 270 degrés) et faible sur les angles obliques (45, 135, 225 et 315 degrés). La distance n'est pas parfaitement nulle dans ce second cas à cause de l'interpolation effectuée lors de la rotation.

7.3.3 Rotation selon l'axe Z

C'est la rotation qui nous intéresse le plus. Dans de nombreuses applications, il est en effet rare de voir des objets effectuer des sauts périlleux. Par ailleurs, c'est également le cas le plus difficile,

²Ceci serait vrai dans un domaine continu, mais c'est en théorie faux dans le domaine discret. Les directions privilégiées des éléments structurants, les effets de bord, les changements dans l'image dûs aux rotations et les (légers) biais des algorithmes contribuent à la déstabilisation. Cependant, en pratique, l'isotropie est assez proche d'être vraie.

puisque'il n'y a pas la même information entre deux images : certaines faces de l'objet sont occultées, il se peut que l'objet n'ait pas le même aspect selon un autre angle, etc.

Du point de vue applicatif, c'est cette stabilité (parallèlement à la robustesse au bruit) qui est la plus importante. En effet, dans le cadre du suivi d'un objet entre plusieurs vues, on ne peut pas garantir que l'objet apparaîtra sous le même angle.

La figure 7.7 présente les distances entre la vue de face et les autres vues pour deux objets aux caractéristiques différentes. Comme on pouvait s'y attendre, pour ce qui est du chat, la distance reste globalement constante. Inversement, pour la bouteille, on voit que la distance entre les deux côtés de la bouteille est assez importante. Ceci s'explique bien sûr par le fait que la bouteille a une face très bleue et que les autres faces sont plutôt blanches. Les creux autour des angles 125 et 235 correspondent aux angles où le texte de la face arrière de la bouteille est visible mais forme une région bleue qui est appariée à la celle de la face avant.

Globalement, on voit que les résultats sont bons dans les cas raisonnables. Le problème des objets non autosimilaires ne peut pas être réglé uniquement par du traitement d'images et nécessitera un autre axe de résolution.

7.3.4 Robustesse face au bruit

Comme on l'a vu dans les premiers chapitres, la gestion du bruit est un élément crucial dans la capacité d'un algorithme à être utilisé dans des cas réels. Les approches par espace d'échelle traditionnelles sont relativement robustes face au bruit. Nous allons voir que la nôtre l'est tout autant.

La figure 7.8 montre l'évolution de la distance entre un objet et le même objet après ajout de bruit. On voit que dans les deux cas, l'évolution est nette mais progressive, et que jusqu'à un niveau de bruit élevé ($\sigma = 80$) les distances intra-objets sont inférieures à la distance inter-objet (calculée sur les objets non-bruités).

7.4 Conclusion

Nous avons proposé dans ce chapitre une méthode de comparaison des arbres calculés dans le chapitre précédent. Nous avons testé cette méthode sur une base d'images classique dans le domaine de l'indexation. A titre illustratif, nous l'avons comparée à une approche basée sur des prémices entièrement différentes : la comparaison d'histogrammes couleur.

Nous avons vu que notre modélisation, couplée à notre distance, était robuste face aux rotations dans le plan image ainsi qu'aux changements d'échelle. Par ailleurs, lorsque les objets sont eux-mêmes autosimilaires, la modélisation est assez constante lors des rotations des objets autour de leur axe vertical.

Au niveau de la complexité, le coût majeur de cette méthode est le calcul des gaussiennes de grande taille dans l'étape de modélisation. Le coût des nivellements est faible devant celui d'une gaussienne, et la distance d'arbres est quasi-instantanée en comparaison. Cependant, il existe des méthodes très rapides pour implémenter (ou approximer) des filtrages par des gaussiennes, par exemple récursivement [Deriche, 1987]. On peut aussi disposer de moyens matériels pour effectuer ces calculs. Ces options complexes ne sont pas disponibles pour les autres parties de l'algorithme (nivellements, distances) donc ce sont sur ceux-ci que nous avons porté nos efforts.

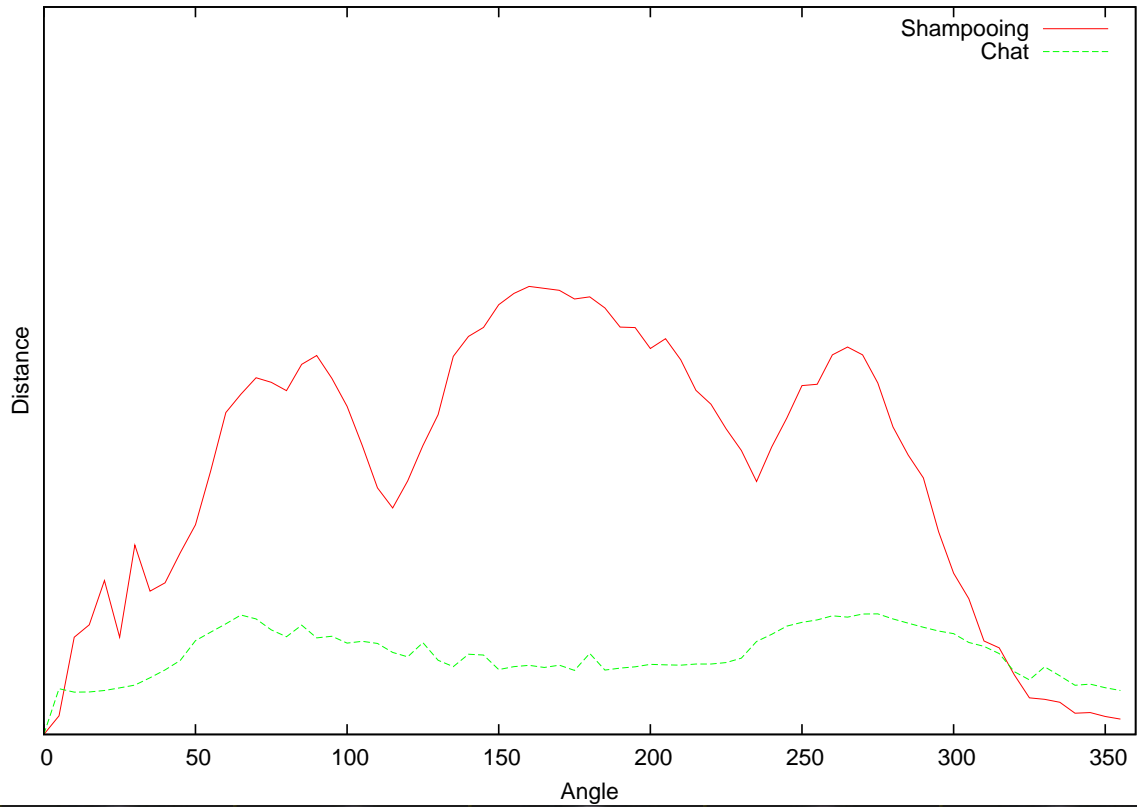


FIG. 7.7: Comparaison des distances entre la vue de face et les vues de l'objet en rotation autour de son axe Z. On voit que le chat est assez autosimilaire, mais que la bouteille évolue beaucoup.

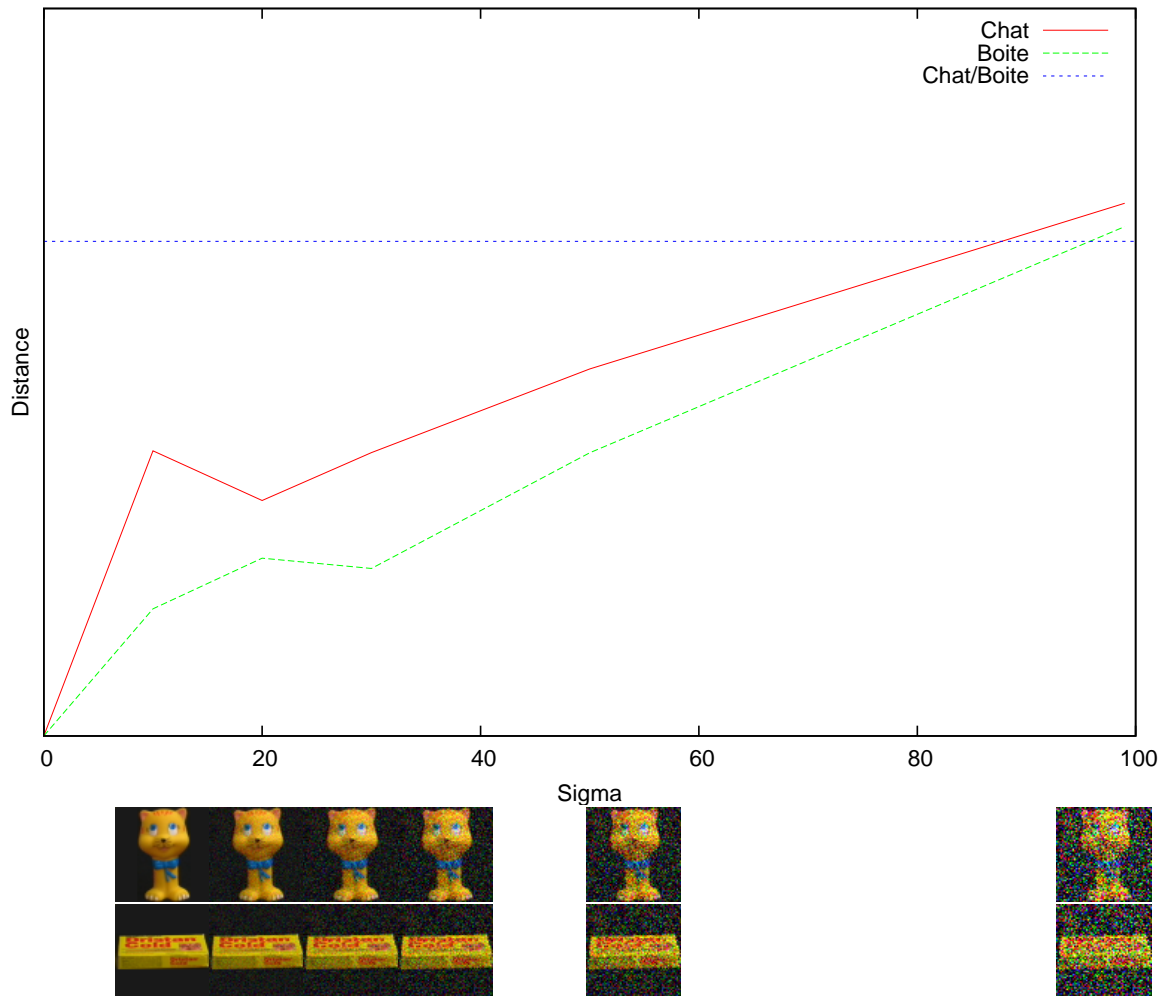


FIG. 7.8: Évolution de la distance entre une image et cette même image bruitée par un bruit gaussien de variance croissante σ pour deux objets. Comparaison avec la distance entre les deux objets (Chat/Boite). On voit que les deux objets ont la même évolution, mais que pour $\sigma = 100$ les distances entre l'objet initial et l'objet bruité sont du même ordre qu'entre les deux objets.

Quatrième partie

Applications

8	Résultats	145
8.1	La segmentation	145
8.2	Appariement de graphes	160
9	Application à la vidéosurveillance	167
9.1	La problématique de la vidéosurveillance	167
9.2	Calcul de la silhouette	171
9.3	Création et suivi d'une signature	180
9.4	Conclusion	191

Résultats et Applications

Synthèse des développements

Le problème auquel nous avons tenté de répondre était de développer ou d'améliorer des méthodes de modélisation d'images basées sur les graphes. Dans cette optique, nous avons avancé selon deux axes principaux : le premier était l'amélioration du matériau de base, c'est-à-dire l'image. Le second était le développement d'outils de modélisation plus appropriés à nos problèmes, suivis de méthodes plus complexes de comparaisons de graphes.

Le filtrage par amibes morphologique s'est révélé un outil particulièrement efficace pour la réduction du bruit. En particulier, c'est un outil qui offre une grande richesse de paramétrage, par exemple pour les images couleurs, ou pour utiliser d'autres types d'amibes que celles proposées ici. Pourtant, les amibes simples que nous avons décrites ici sont relativement simples d'emploi et ne comportent que deux paramètres. Leur inconvénient majeur est le temps de calcul relativement important d'un filtrage par amibe.

Les gradients couleurs améliorés se sont révélés être des outils efficaces pour la détection de contours dans des images bruitées. On verra dans le chapitre suivant qu'ils présentent notamment un bon compromis entre qualité de détection et vitesse de calcul.

La modélisation par arbres, basée sur les espaces d'échelle morphologiques est un outil très efficace pour décrire des objets pour peu que ceux-ci soient relativement riches et comportent des sous-objets, et que la segmentation réussisse à les isoler. Moyennant ces conditions, on a vu qu'ils offraient une description souple et efficace, dont on peut moduler à loisir la précision. En particulier les techniques de comparaison et de recherche que nous avons développées semblent prometteuses.

Résultats

Pour résumer et synthétiser ces développements, nous allons effectuer des comparaisons entre les diverses techniques de modélisation ou de filtrage décrites dans ce document. L'objectif de ce chapitre sera de décrire les comportements des différentes technologies que nous avons développées, afin de permettre à l'utilisateur de choisir l'outil le plus approprié à son problème.

La question principale qui retiendra notre attention sera la description du compromis offert entre la qualité du résultat définitif et le temps mis à l'obtenir.

Application à la vidéosurveillance

Enfin, nous terminerons par l'étude des composants nécessaires pour construire une application de vidéosurveillance utilisant les modélisations que nous avons développées dans cette thèse. En par-

ticulier, nous développerons les techniques de détection d'objets que nous avons considérées comme disponibles au début du chapitre 7.

Ces techniques ont été conçues dans le cadre d'un projet applicatif, mené en collaboration avec l'INRIA sous l'égide de la société ST Microelectronics. Nous décrirons les concessions que nous avons dû faire pour respecter les contraintes de performances imposées, ainsi que les méthodes que nous avons développées spécifiquement pour pallier les problèmes que nous avons rencontrés. Le point critique de ce projet étant la vitesse, nous expliquerons notamment les méthodes que nous avons employées pour hiérarchiser l'information et raffiner nos techniques d'analyse tout en réduisant le volume de données à traiter.

Disposant des silhouettes des individus, nous pourrions ainsi les modéliser par des arbres, et tester des outils de comparaison qui permettraient de suivre les individus de façon prolongée dans le temps, malgré des interruptions ou des changements d'angles de vue.

Chapitre 8

Résultats

*When you can measure what you are speaking
about and express it in numbers, you know
something about it ;
but when you cannot measure it, when you cannot
express it in numbers, your knowledge is of the
meager and unsatisfactory kind.*

Lord Kelvin

Dans ce chapitre, nous présenterons les résultats de nos améliorations sur le filtrage et la modélisation en les comparant à ceux que nous avons dans les premiers chapitres. Nous comparerons les temps de calcul, mais sans écarter d'approche pour des raisons de complexité ou de durée d'exécution. L'impératif est d'obtenir une solution qui fonctionne, l'optimisation pouvant venir dans un second temps.

Dans le cadre d'une chaîne de traitement, la seule mesure réellement pertinente est celle du résultat final. Cependant, ceci rend l'optimisation des composants individuels extrêmement difficile. On a vu ainsi à la fin du chapitre 7 le cas d'un gradient relativement médiocre visuellement mais donnant de bons résultats sur la segmentation.

Malgré cela, il existe des «charnières» dans la chaîne de traitement qui cloisonnent les composants de sorte qu'on puisse chercher à les optimiser séparément. Dans notre cas, la charnière principale se situe entre la segmentation et la modélisation par graphe.

En premier lieu, nous étudierons la segmentation, qui est l'étape fondamentale de toutes nos approches. En particulier, la segmentation hiérarchique «bottom-up» qui est la plus rapide, mais aussi la plus instable. Nous chercherons à l'optimiser en améliorant la qualité du gradient sur laquelle on l'applique. Ensuite, nous comparerons les diverses approches de modélisation par graphes.

8.1 La segmentation

Le point de départ de toutes nos modélisations, qu'elles soient basées sur des graphes ou des arbres est la segmentation de l'image initiale. On a vu dans les premiers chapitres les difficultés rencontrées par les approches simples dans les images complexes. Nous allons ici comparer les méthodes avancées

de filtrage en étudiant la qualité de la segmentation obtenue par le même algorithme de segmentation hiérarchique.

Afin de pousser nos algorithmes jusqu'à leurs limites, nous utiliserons une image particulièrement difficile, présentant des structures de bruit à plusieurs niveaux. Les échecs seront ainsi particulièrement visibles et il sera donc plus facile d'estimer les qualités et les défauts des différentes approches.

8.1.1 Choix d'une mesure

Étant donné un nombre de régions, on va considérer que l'objectif de la segmentation hiérarchique est de renvoyer une partition de l'image qui minimise la variance interne dans chaque région. Ainsi, les partitions associées à une faible distance seront celles dans lesquelles les régions seront très représentatives des pixels qui les composent. Dans un espace scalaire, la variance est la moyenne des carrés des distances à la moyenne. En couleur, on fera donc la moyenne des distances couleurs élevées au carré entre chaque point et la couleur moyenne de la région à laquelle il appartient. Pour ce faire, on passera par la création d'une image mosaïque dans laquelle chaque région de l'image segmentée est évaluée par sa couleur moyenne.

Par ailleurs, on postulera que le choix du nombre de région est un critère piloté par des besoins externes, comme par exemple les performances des algorithmes d'appariement, etc. On ne l'inclura donc pas dans la formulation de la mesure. Nous considérerons en effet que l'utilisateur cherchera la modélisation optimale en terme de vitesse et de distorsion étant donné la consigne du nombre de régions. Nous conseillerons au lecteur intéressé par une mesure du compromis entre nombre de régions et distorsion de se référer à [Guigues, 2003], en particulier le chapitre 8.

Pour la distance entre l'image d'origine f et l'image mosaïque f_{mos} , on s'inspirera de la mesure de variance normalisée de [Levine and Nazif, 1985] définie par :

$$M(Image : x \rightarrow f(x)) = \frac{1}{A\sigma_{max}^2} \sum_i A_i \sigma_i^2$$

où A_i est l'aire de la région \mathcal{R}_i , σ_{max} est l'écart maximal de couleur et σ_i est la variance interne de la région i .

La variance σ_i est définie sur une région \mathcal{R}_i d'aire A_i et de valeur moyenne \bar{f}_i par :

$$\sigma_i^2 = \frac{1}{A_i} \sum_{x \in \mathcal{R}_i} (f(x) - \bar{f}_i)^2$$

Dans notre cas, la valeur moyenne est donnée par notre image mosaïque f_{mos} , et nous prendrons une distance couleur plutôt qu'une différence absolue :

$$\tilde{\sigma}_i^2 = \frac{1}{A_i} \sum_{x \in \mathcal{R}_i} [d_{Lab}(f(x), f_{mos}(x))]^2$$

Enfin, pour permettre des comparaisons plus pertinentes, le facteur de normalisation d_{max} sera identique pour toutes les images comparées. Ceci permet de comparer les résultats entre eux de façon absolue, et non pas leur évolution relative. Dans le cas contraire, on aurait tendance à sous-estimer fortement les valeurs des images très simplifiées, et les résultats obtenus ne correspondraient pas aux impressions visuelles. En définitive, la formulation de notre mesure revient à un écart quadratique moyen :

$$d(f, f_{mos}) = \frac{1}{A} \sum_x [d_{Lab}(f(x), f_{mos}(x))]^2$$

8.1.2 Amélioration des gradients

Dans un premier temps, nous étudierons le gain réel obtenu avec un des gradients couleurs que nous avons décrits dans le chapitre 5. D'après les résultats que nous avons établis, le meilleur gradient couleur avancé que nous ayons est le gradient «écart-type». C'est donc celui-ci que nous comparerons au gradient de référence (le gradient morphologique généralisé).

Résultats

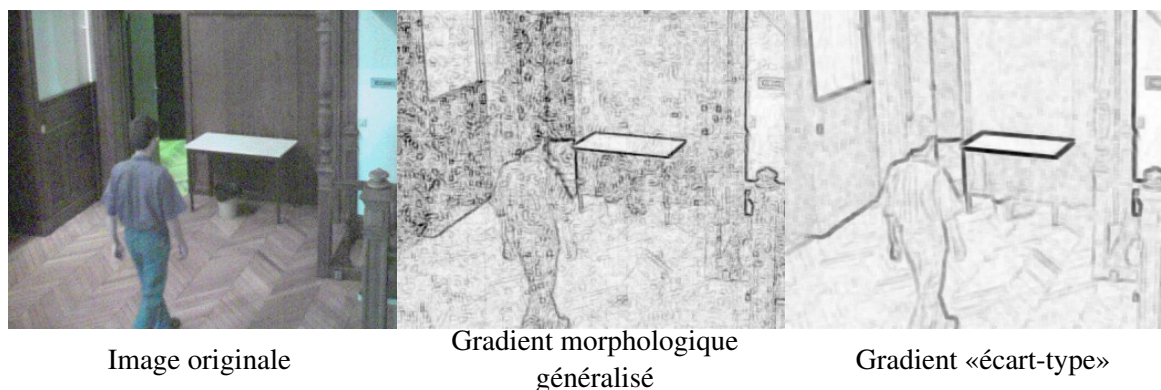


FIG. 8.1: Gradients couleur

La figure 8.1 présente côte-à-côte le gradient morphologique généralisé qui est l'extension à la couleur du gradient de Beucher (maximum des distances entre tous les couples de couleurs dans le voisinage) et le gradient «écart-type».

La figure 8.2 présente les résultats d'une segmentation hiérarchique basée sur chacun de ces gradients.

Dans l'image à 500 régions, on voit déjà que le gradient «écart-type» est plus fidèle aux contours de l'image. En particulier, on voit apparaître la structure en quinconce du plancher et les montants de la porte sont mieux préservés, ainsi que le pilier droit.

Dans l'image à 200 régions, le personnage est encore bien détecté à droite notamment au niveau de la tête et de la taille qui ont commencé à fusionner dans l'image de gauche.

Dans l'image à 100 régions, il est intéressant de noter qu'à droite ce sont principalement les contours intérieurs qui ont fusionné, en particulier dans la chemise.

Dans la dernière image, la silhouette du personnage a été préservée de façon imparfaite, alors qu'il a complètement disparu à gauche. Nous allons maintenant détailler la progression qui a conduit à la fusion excessive entre des régions du fond et des parties du personnage.

Fusions excessives

La figure 8.4 se focalise sur les dernières étapes de fusion dans la segmentation hiérarchique basée sur le gradient «écart-type». On voit ainsi que c'est d'abord le visage qui fusionne avec la zone verdâtre, suivi par l'épaule qui fusionne par le bras avec le plancher. Les dernières fusions regroupent correctement les régions survivant dans l'objet. Il est intéressant de voir qu'entre la segmentation à 15 et à 10 régions, on aurait tendance à séparer la zone du visage du reste de la personne, car la couleur verte est très différente de la couleur bleu sombre. Cependant, l'algorithme choisit (correctement !) de l'associer à l'objet. La raison à cela est simplement que la fusion suit l'arbre d'inondation, lequel reste

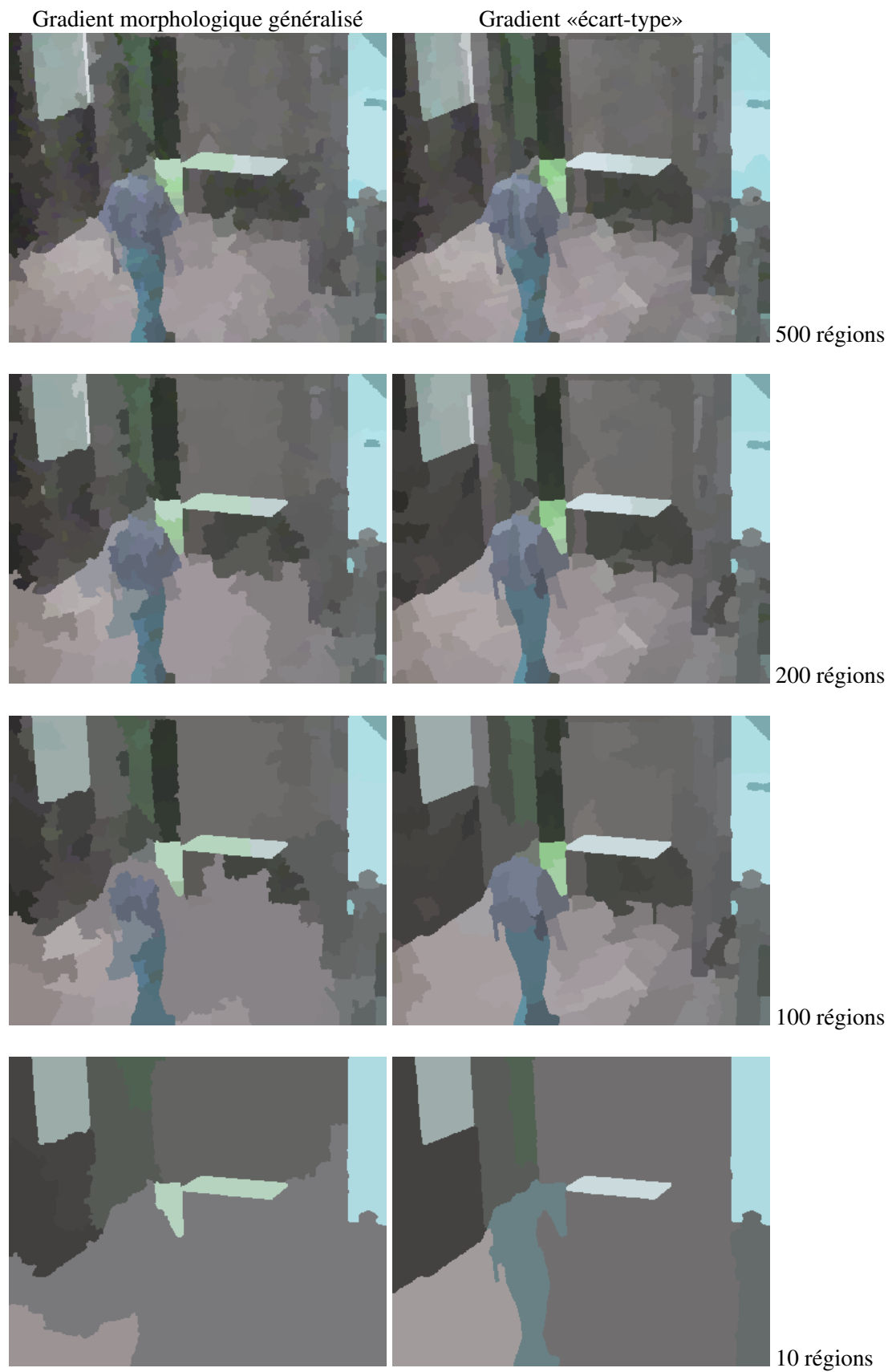


FIG. 8.2: Comparaison entre la segmentation hiérarchique basé sur le gradient morphologique généralisé et celle basée sur le gradient «écart-type» pour 500, 200, 100 et 10 régions.

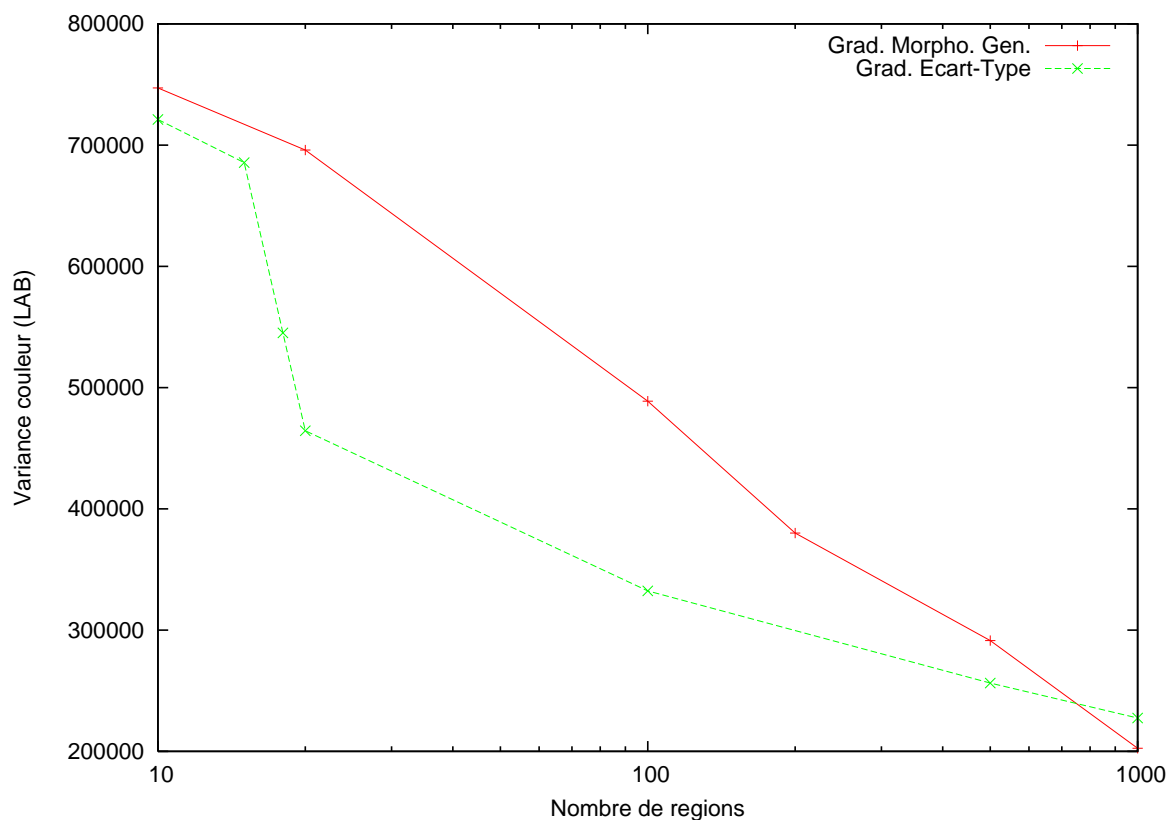


FIG. 8.3: Mesure des distorsions en fonction du nombre de régions entre les segmentations hiérarchiques basées sur le gradient morphologique généralisé et le gradient «écart-type».

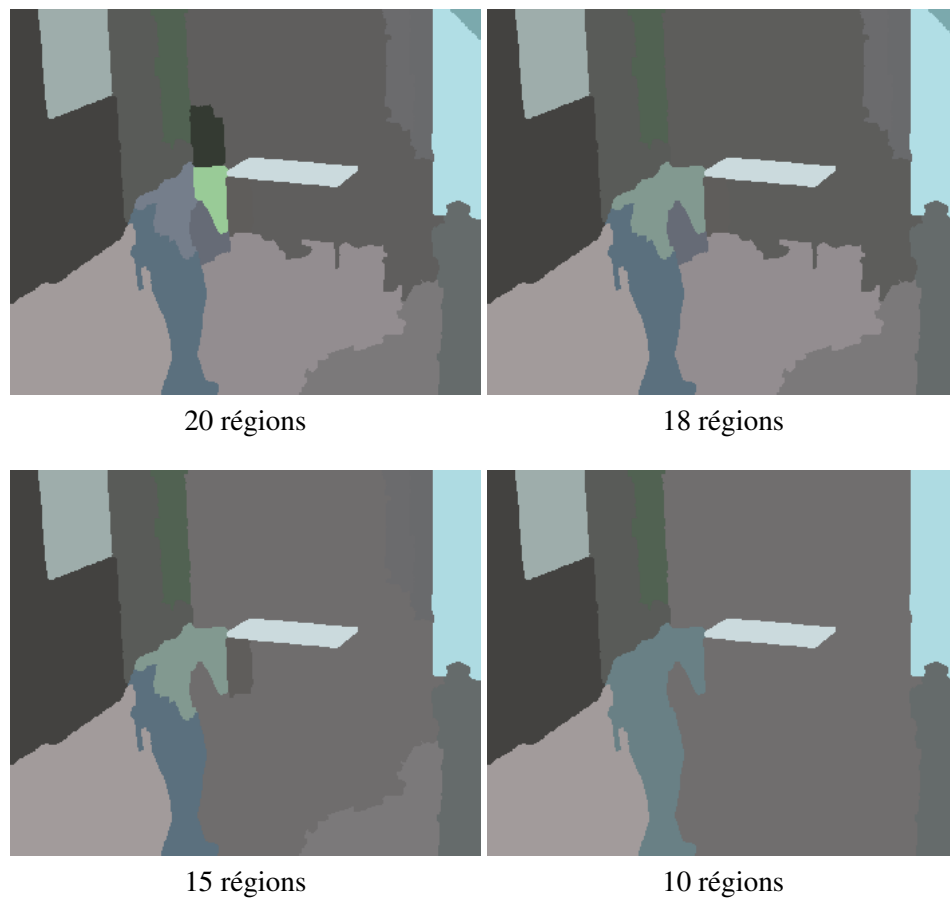


FIG. 8.4: Progression de la fusion des régions dans la hiérarchie basée sur le gradient «écart-type». Les points de défaillance du gradient sont le visage et le bras droit.

basé sur le gradient, alors que le choix de la couleur moyenne est effectué par une mesure globale sur la région. Ainsi, le contour entre la région verte et le corps de la personne est beaucoup moins fort (figure 8.5) qu'il n'y paraît au vu des images mosaïques.



FIG. 8.5: Zoom sur le visage et le bras : le gradient estime un contour relativement faible autour de l'épaule (ce qui explique la fusion du visage malgré le contour localement très fort) et du bras (expliquant ainsi l'autre fusion du dos avec le fond).

Conclusion

On peut conclure de cette expérience que le gradient couleur amélioré «écart-type» peut améliorer nettement le résultat d'une segmentation hiérarchique. L'expérience ci-dessus a montré que la silhouette du personnage était correctement préservée jusqu'à 20 régions alors qu'elle commençait à disparaître vers 100 régions dans le cas précédent. Le tableau suivant résume les temps de calculs rapportés à ceux du gradient de Beucher de l'image de la luminance :

Type de gradient	Beucher	Généralisé	«Écart-Type»
Temps de calcul	1	3.8	3.13

TAB. 8.1: Temps de calcul de différents gradients couleur, rapportés au gradient de Beucher du canal luminance.

Comme on le voit, le coût des gradients couleur est supérieur d'un facteur supérieur à 3 au gradient en niveaux de gris. Cependant un point intéressant est révélé par ces chiffres : le gradient «écart-type», est non seulement meilleur que le généralisé mais il est également plus rapide. Ceci s'explique par le fait que la recherche de la valeur maximale est moins bien optimisée par le compilateur que le calcul simple et brutal de l'écart-type, qui se prête mieux à la vectorisation ou au «pipe-lining».

8.1.3 Approches coûteuses

Utilisation du gradient pyramide

On a vu au chapitre 6 que le gradient pyramide était un outil intéressant pour permettre la segmentation d'objets ayant tendance à se fondre dans leur environnement. L'utilisation des espaces d'échelle morphologiques permettant de renforcer les contours localement faibles en lissant la forme générale de l'objet.

La figure 8.6 présente le gradient pyramide à côté du gradient écart-type. Deux points viennent immédiatement à l'attention : d'une part le gradient est plus fin et d'autre part, il est également moins bruyé. L'épaisseur s'explique par le fait que le gradient écart-type est calculé sur une fenêtre 5x5, alors que le gradient pyramide est une accumulation de gradients 3x3 (on notera au passage que les contours ne sont effectivement pas déplacés par les nivellements). De plus, les petites variations (plancher, murs) ne génèrent pas autant de bruit. Les résultats semblent donc très prometteurs.

La figure 8.7 compare les segmentations hiérarchiques du gradient pyramide et du gradient écart-type. On peut voir que les bras sont bien conservés, de même que l'allure générale du personnage. En revanche, un point apparaît qui n'était pas aussi frappant sur l'image du gradient : les contours sont légèrement déformés, comme on peut le voir sur les lattes de plancher. Ceci s'explique a posteriori par le fait que les contours reconstruits par les nivellements sont légèrement corrompus par le bruit, et donc suivent moins bien les lignes réelles des objets. On retrouve ici la problématique qui a conduit à la conception des amibes morphologiques. Paradoxalement, ce problème qui n'a apparemment pas gêné le gradient écart-type réapparaît suite au lissage effectué par l'espace d'échelle morphologique.

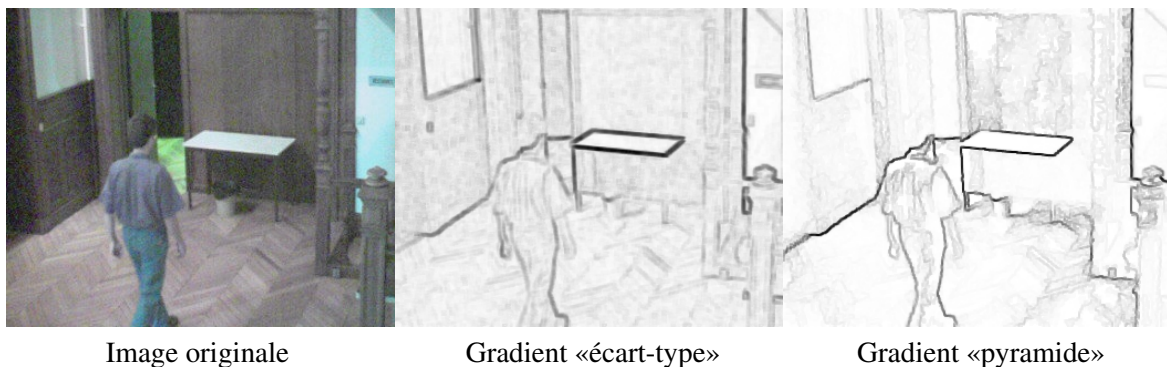


FIG. 8.6: Comparaison entre le gradient écart-type et le gradient pyramide.

Amélioration par les amibes

Une autre approche pour l'amélioration de la qualité de la segmentation consiste à filtrer l'image originale par une amibe morphologique. L'objectif de cette approche est d'augmenter la cohérence intra-régionale et par là-même renforcer la valeur du gradient.

La figure 8.9 montre le résultat du même algorithme de segmentation hiérarchique, en particulier le même algorithme de gradient couleur généralisé, sur l'image originale et l'image filtrée par une amibe morphologique. On a utilisé la moyenne calculée sur l'espace RGB et une amibe de taille relativement grande, pilotée par l'image originale filtrée par une petite gaussienne.

Les points forts de la comparaison précédente entre les gradients sont encore plus valides dans cette approche. Le point le plus notable est que la personne est encore presque parfaitement segmenté

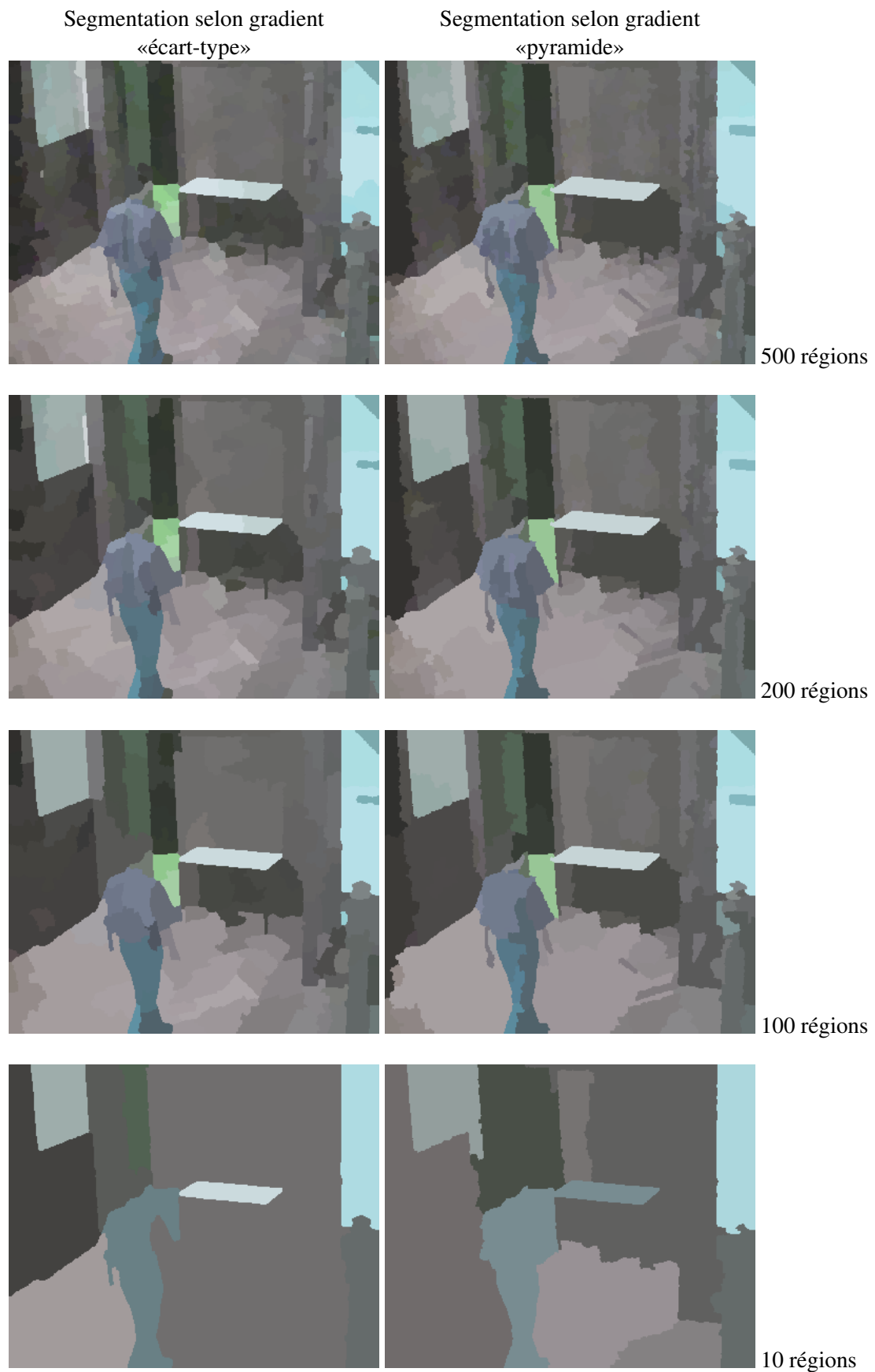


FIG. 8.7: Comparaison des segmentations : à gauche le gradient «écart-type», à droite le filtrage par amibe morphologique suivi par un gradient morphologique généralisé.

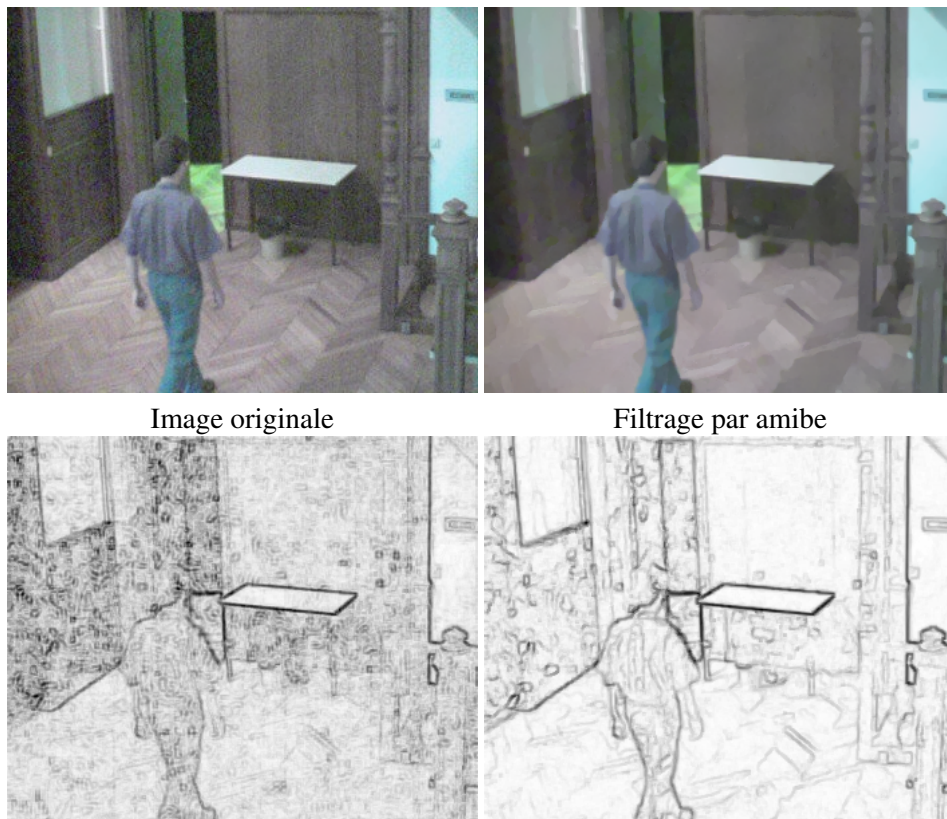


FIG. 8.8: Filtrage de l'image originale (gauche) par une amibe morphologique (droite), et gradients correspondants. Dans une optique de segmentation hiérarchique, on a cherché le meilleur compromis entre une image aussi lisse que possible, sans toutefois excessivement dégrader les contours. Les images de gradient ont été obtenues par le même algorithme de gradient morphologique généralisé.

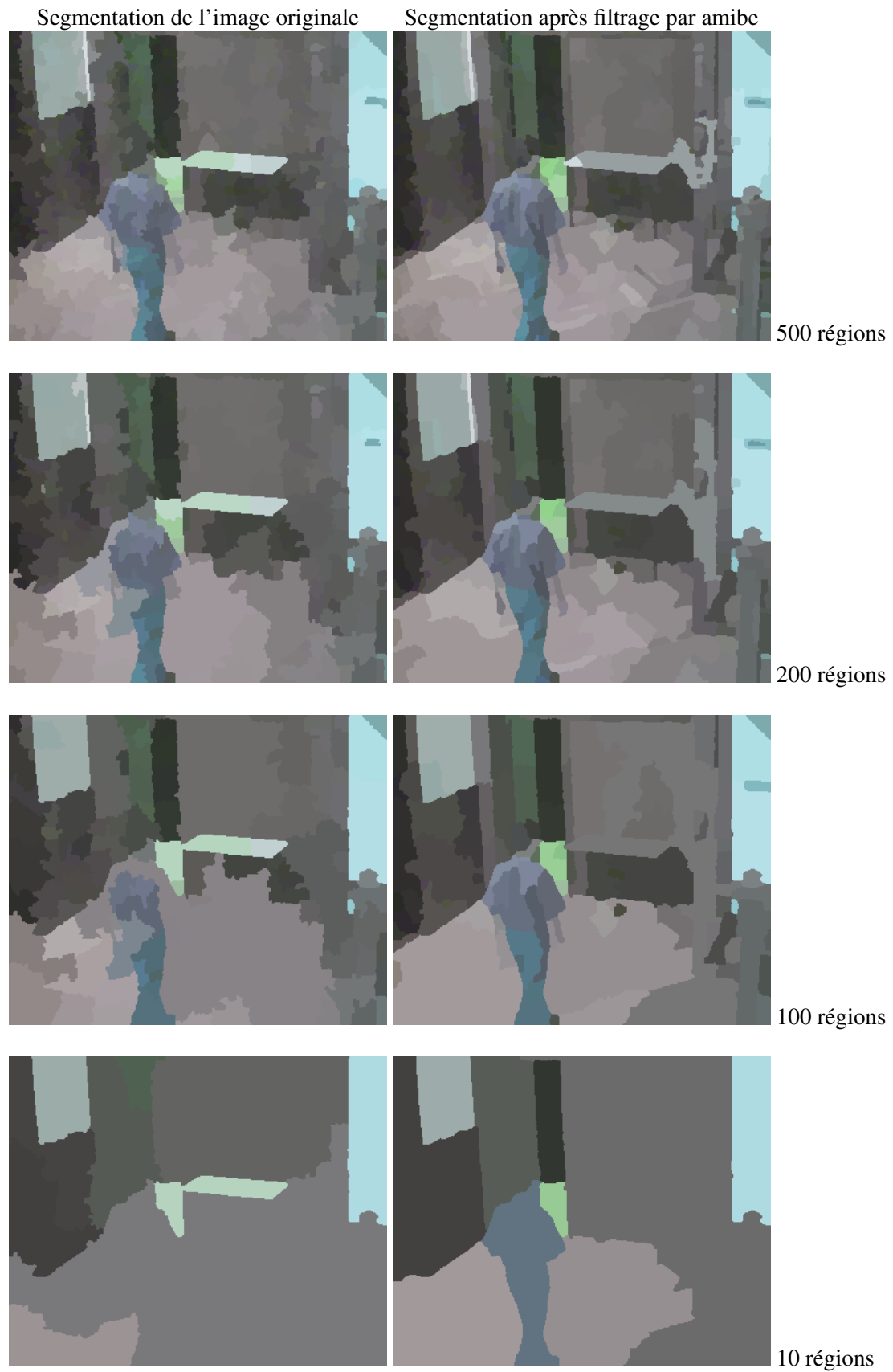


FIG. 8.9: Segmentation hiérarchique basé sur le gradient morphologique généralisé sur l'image originale et celle filtrée par une amibe morphologique pour 500, 200, 100 et 10 régions.

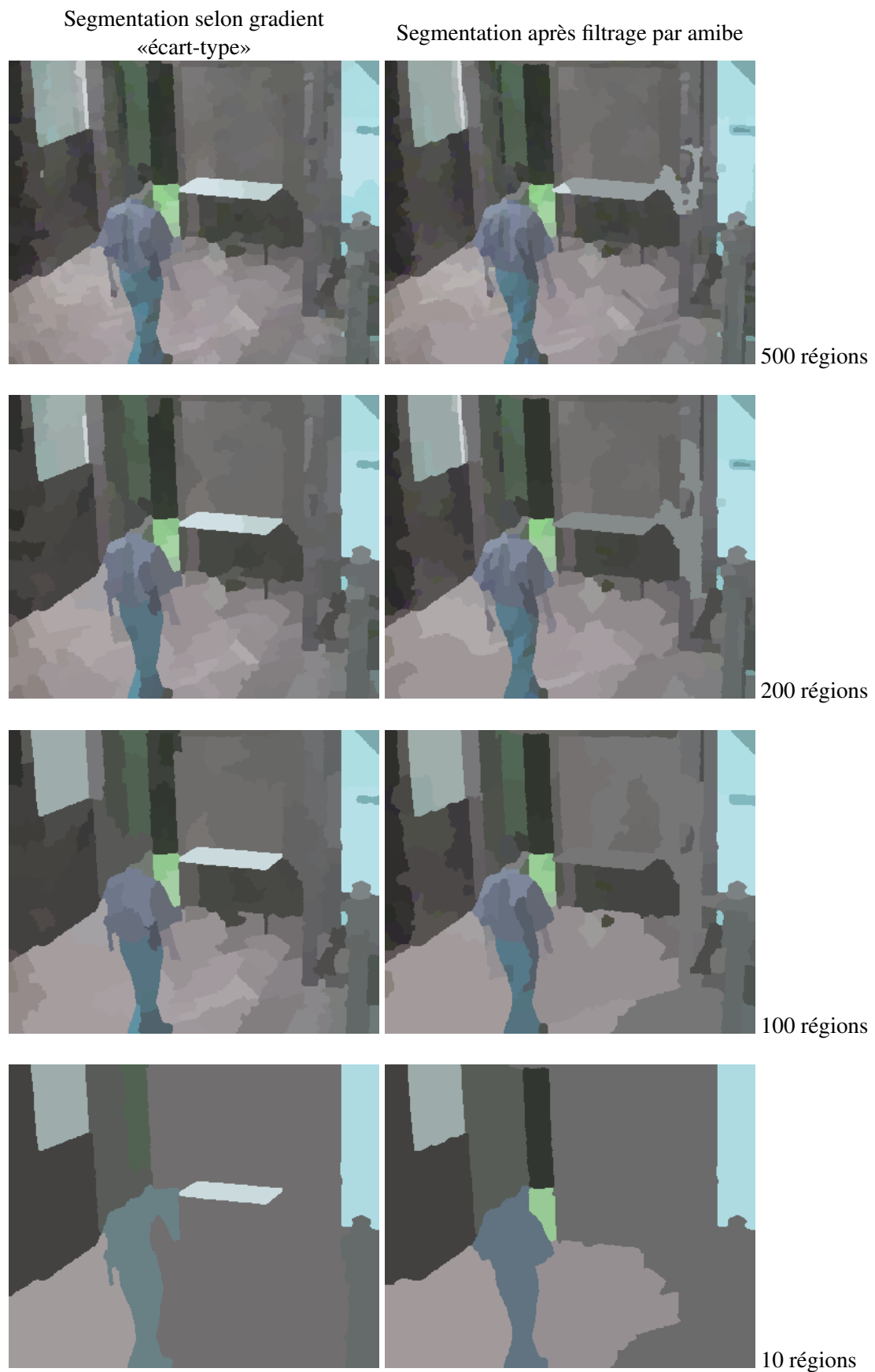


FIG. 8.10: Comparaison des segmentations : à gauche le gradient «écart-type», à droite le filtrage par amibe morphologique suivi par un gradient morphologique généralisé.

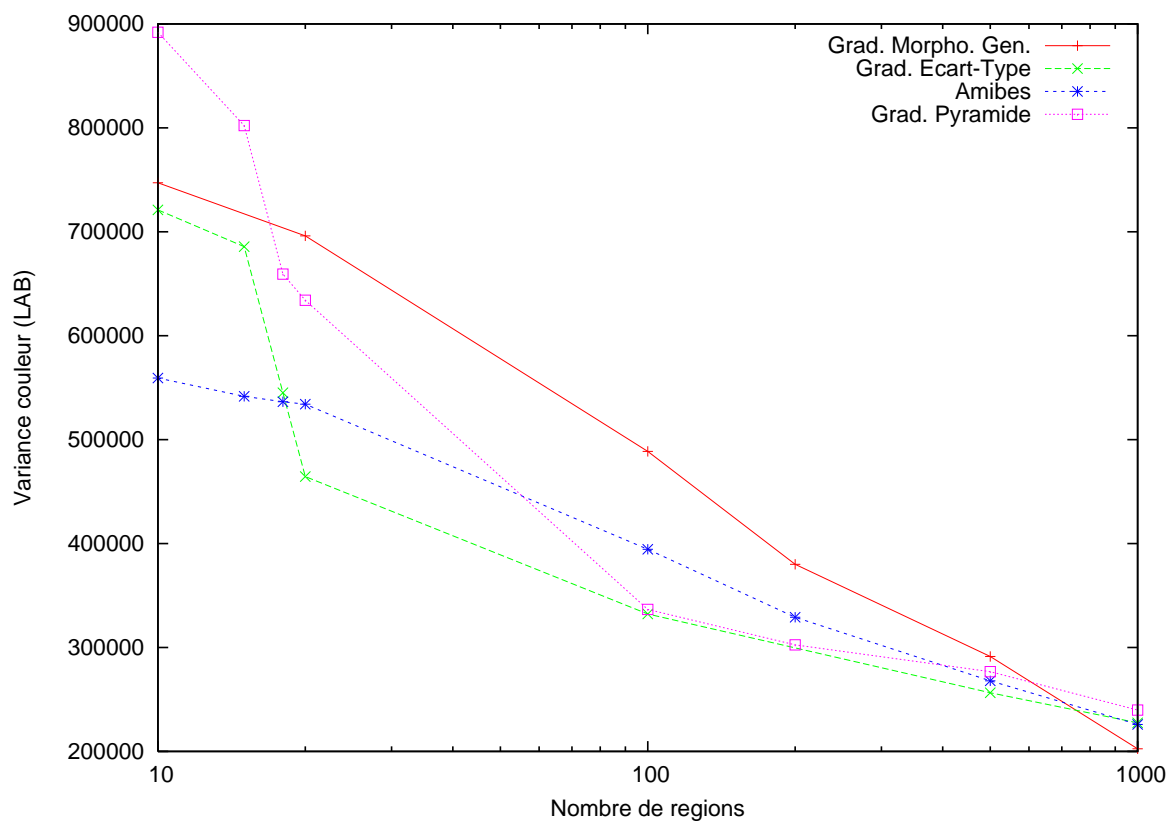


FIG. 8.11: Mesure des distorsions en fonction du nombre de régions entre les segmentations hiérarchiques basées sur les deux gradients couleurs précédents sur l'image filtrée par une amibe morphologique (suivie du gradient morphologique généralisé).

au niveau des 10 régions. Cette qualité de résultats appelle une comparaison directe entre les deux approches évoluées.

On voit dans les résultats présentés dans la figure 8.10.

A 500 régions, on voit nettement l'effet lisseur des amibes : les lattes du parquet sont moins bien segmentées, et il y a une zone très fautive autour du pilier à droite. D'un autre côté, la main droite est correctement segmentée et plusieurs petites zones fortement contrastées sont préservées (le texte dans le panneau sur le mur à droite).

A 200 régions, on voit que le lissage de l'amibe permet de fusionner de grandes régions dans le parquet et dans le fond, ce qui permet de conserver beaucoup de petites régions dans les zones très contrastées. La main droite est préservée, ainsi que de nombreux détails dans le dos.

A 100 régions, la segmentation basée sur le gradient «écart-type» commence à fusionner de façon excessive certaines régions, comme par exemple le visage et les cheveux. Du côté des amibes, on continue à fusionner les lattes du parquet et le fond, et certains détails commencent à être éliminés, comme par exemple la main droite qui finit absorbée par le plancher.

A 10 régions, on voit qu'il ne reste plus que le corps de la personne et les grands objets du fond, mais que le contour est mieux préservé du côté de l'amibe. Le fait que le visage et l'épaule ne soient pas absorbés comme dans le cas du gradient «écart-type» (ou pire encore, de l'approche directe !) s'explique en regardant de près l'allure du gradient calculé après filtrage de l'amibe : figure 8.12.

Dans cette comparaison, on voit que les gradients sont particulièrement forts dans les zones problématiques de l'épaule et le long de la manche. En particulier, on voit que l'amibe a bien su lisser les artefacts de compression DCT (voir le coin de la manche droite) qui perturbaient le gradient «écart-type».

Conclusion

Il apparaît nettement dans ces comparaisons que le type de filtrage opéré par les amibes morphologiques est particulièrement adapté à l'amélioration de contours corrompus par le bruit, et en particulier le bruit structuré comme les vaguelettes dues à une forte compression destructive. En revanche, le coût de filtrage pour cette image est de 142 fois le coût du gradient de Beucher (contre 3 fois pour le gradient «écart-type»). Il faut donc réserver cette approche aux cas particulièrement complexes, ou à ceux pour lesquels la rapidité n'est pas un critère primordial.

Le gradient pyramide coûte 207 fois plus cher que le gradient de Beucher, mais on peut envisager de réduire ce coût en limitant le nombre de niveaux sur lesquels on le calcule. On pourrait ainsi atteindre le niveau des amibes. Outre les résultats ci-dessus, les comparaisons du chapitre 6.1 ont montré l'intérêt de cette approche dans les cas où le contour est localement affaibli par une texture interne à l'objet. Les amibes ne sont pas un bon candidat pour filtrer ce type d'images car elles préserveront la texture (les rayures du tigre) et donc les faiblesses du contour de l'objet. Encore une fois, l'utilisation de ce gradient est réservée à une certaine classe de problèmes difficiles pour lesquels la contrainte de temps n'est pas un facteur essentiel.

8.1.4 Récapitulatif

Nous avons décrit plusieurs approches visant à améliorer les images et leurs gradients pour être utilisées dans une segmentation hiérarchique. Nous avons effectué des mesures de qualité des résultats de cette segmentation et nous avons également mesuré le temps d'exécution de chaque approche.

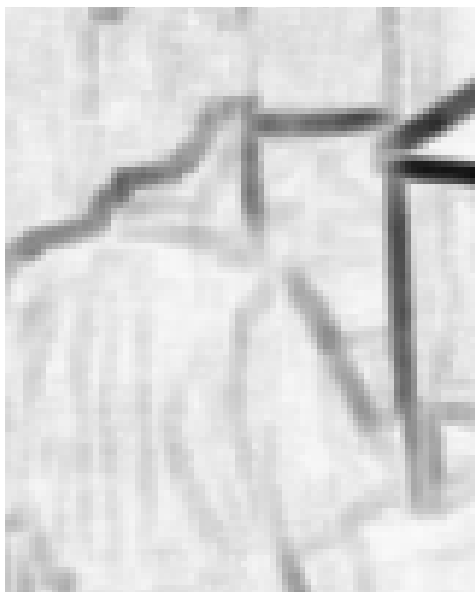
Dans la catégorie des approches rapides, le gradient écart-type a clairement montré sa supériorité sur les approches antérieures tant au niveau de la qualité de la segmentation qu'en vitesse d'exécution.



Image originale



Image filtrée par amibe



Gradient «écart-type»



Gradient morphologique généralisé

FIG. 8.12: Zoom sur le visage et le bras pour les deux images et leur gradient. Après filtrage par l'amibe, le gradient est beaucoup plus fort car les aplats de l'amibe ont lissé renforcé les lignes de contour.

Cette approche est à la fois meilleure et plus rapide.

Nous avons également étudié l'utilisation de techniques intensives de filtrage : les amibes morphologiques et le gradient pyramide. Ce dernier a produit un gradient visuellement intéressant et a été le seul capable de préserver longtemps certains détails. Cependant, les mesures de qualité n'ont pas rendu compte de ce fait et certains choix malheureux de la segmentation hiérarchique ont conduit à un très mauvais score. Cependant, nous pensons que ce gradient peut offrir de bons résultats lorsqu'on l'applique à une image appropriée. Les amibes quant à elles ont produit le meilleur résultat de segmentation en un nombre faible de régions. Elles sont les seules à avoir correctement préservé la silhouette de la personne au niveau des 10 régions (modulo les cheveux et les bras).

Ces résultats, complétés par ceux du chapitre 6 (le tigre), nous permettent ainsi de clore le panorama des gradients entamé dans le chapitre 5. En particulier, on voit que différents problèmes appellent différentes solutions, selon la priorité des contraintes de qualité des résultats ou de vitesse.

8.2 Appariement de graphes

8.2.1 Choix du problème

La difficulté dans l'évaluation du problème de l'appariement de graphes est dans la définition d'une mesure. En particulier une mesure automatique et pertinente des résultats entre deux approches d'appariement de graphes.

Les problèmes posés par les séquences de vidéosurveillance sont souvent plus liés à la détection et à la segmentation des personnes qu'à la modélisation par graphes. Souvent les différences entre les modèles sont trop faibles pour bien estimer la qualité du moteur d'appariement.

Nous avons donc décidé de baser nos mesures sur un autre problème : la recherche d'images dans une base de données. En particulier, la base COIL. Il y a trois raisons à ce choix :

- les objets sur leur fond noir sont quasiment déjà segmentés, ce qui retire un facteur important du problème.
- il y a une grande variabilité dans les objets présents, et l'analyse des réponses des algorithmes sera d'autant plus riche que l'on verra des comportements différents.
- on peut aisément inventer à partir de cette base des problèmes de classification dont les résultats seront faciles à mesurer objectivement.

8.2.2 Recherche d'images par l'exemple

La recherche par l'exemple est un des problèmes les plus courants en indexation : l'utilisateur fournit une image et on lui renvoie les images les plus ressemblantes de la base. Les applications peuvent aller du domaine biomédical au classement des photos de vacances.

Le problème que nous nous donnerons utilisera la base de données COIL. Étant donné un exemple d'objet, nous chercherons à retrouver les objets de la même classe. La classe la plus utilisée dans COIL étant la voiture miniature, c'est celle-ci que nous utiliserons pour nos exemples (voir figure 8.13). Plus précisément, nous choisirons le sous-ensemble des voitures à teinte globalement jaune-orangée.

Nous comparerons deux algorithmes sur ce problème, le premier est celui basé sur les arbres décrit dans le chapitre 7 et le second est celui utilisant la méthode de Cristina Gomila et le «*graduated assignment*» du chapitre 3.



FIG. 8.13: Les voitures de COIL

Mesure

Pour évaluer les résultats de chaque algorithme, nous chercherons pour chaque voiture les meilleurs appariements et nous comptabiliserons les mesure de précision et de rappel en fonction du nombre de résultats souhaités (de 1 à 5).

Pour mémoire, si \mathcal{B} est l'ensemble d'éléments corrects dans la base et \mathcal{R} est le résultat de l'algorithme de classification, alors :

Définition 8. Le *rappel* d'un résultat est le rapport du nombre d'éléments corrects trouvés sur le nombre total d'éléments corrects de la base de données, soit

$$r = \frac{|\mathcal{R} \cap \mathcal{B}|}{|\mathcal{B}|}$$

Définition 9. La *précision* d'un résultat est le rapport du nombre d'éléments corrects trouvés sur le nombre total d'éléments trouvés.

$$p = \frac{|\mathcal{R} \cap \mathcal{B}|}{|\mathcal{R}|}$$

Où $|X|$ est le cardinal de l'ensemble X . Intuitivement, la précision représente la pertinence du système et le rappel sa couverture.

Résultats

Les résultats renvoyés par les algorithmes sont classés par ordre de pertinence. Nous quantifierons cet ordre en donnant pour chaque algorithme les mesures de précision et rappel en fonction du nombre de résultats souhaités, de un à cinq (nombre de voitures dans la base).

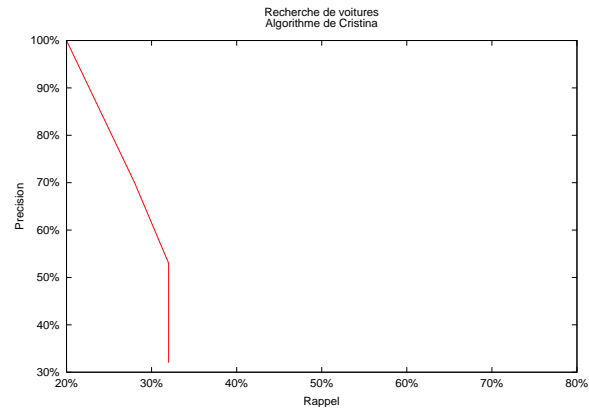
Les résultats sont présentés sur les figure 8.15 et 8.16, et les chiffres de précision-rappel sont donnés dans les tableaux 8.2 et 8.3 ci-dessous.

Pour l'algorithme de Cristina, malgré notre ajout du «graduated assignment», les résultats sont cohérents avec nos conclusions du chapitre 3. On voit assez clairement que l'algorithme ne peut pas généraliser à partir des modèles présentés. Les résultats manquent d'ailleurs de cohérence visuelle, bien que cela soit difficile à formuler de façon objective et mesurable.

L'algorithme d'appariement d'arbres offre de meilleurs résultats, notamment lorsqu'on demande une grande partie de la classe, voire toute la classe. Cette meilleure capacité à généraliser se traduit d'ailleurs par des résultats beaucoup plus cohérents visuellement.

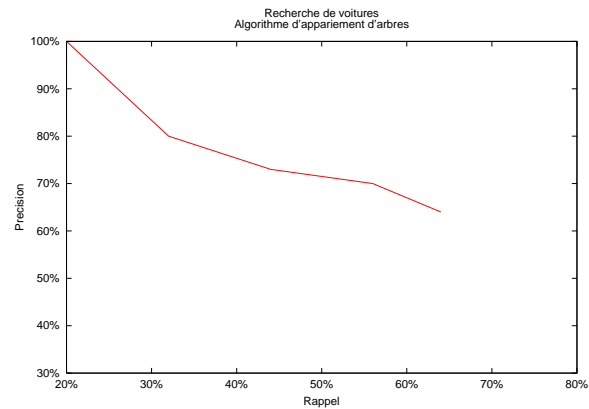
8.2.3 Temps de calcul

Un point crucial à considérer avant de comparer les temps de calcul est de voir que l'algorithme par arbres peut se décomposer en deux parties : d'une part la création des arbres, et d'autre part la



	1 objet	2 objets	3 objets	4 objets	toute la classe
Précision	100%	70%	53%	40%	32%
Rappel	20%	28%	32%	32%	32%

TAB. 8.2: Mesures précision-rappel pour la recherche des voitures dans la base COIL sur l'algorithme de Cristina.



	1 objet	2 objets	3 objets	4 objets	toute la classe
Précision	100%	80%	73%	70%	64%
Rappel	20%	32%	44%	56%	64%

TAB. 8.3: Mesures précision-rappel pour la recherche des voitures dans la base COIL sur l'algorithme d'appariement d'arbres.

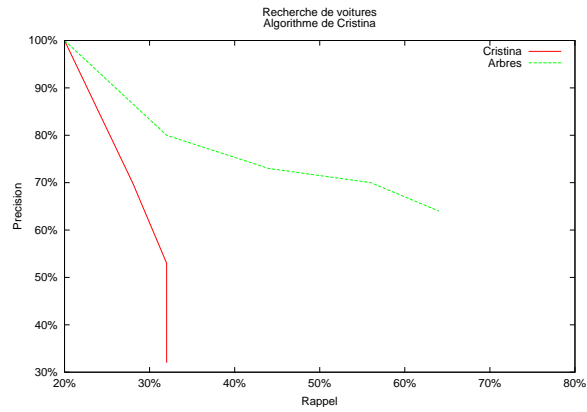


FIG. 8.14: Superposition des courbes précision-rappel des deux algorithmes. On voit clairement que l’algorithme de Cristina s’effondre rapidement. L’algorithme d’appariement d’arbres, en revanche offre de bons résultats même en ajoutant les quatrième et cinquième choix.

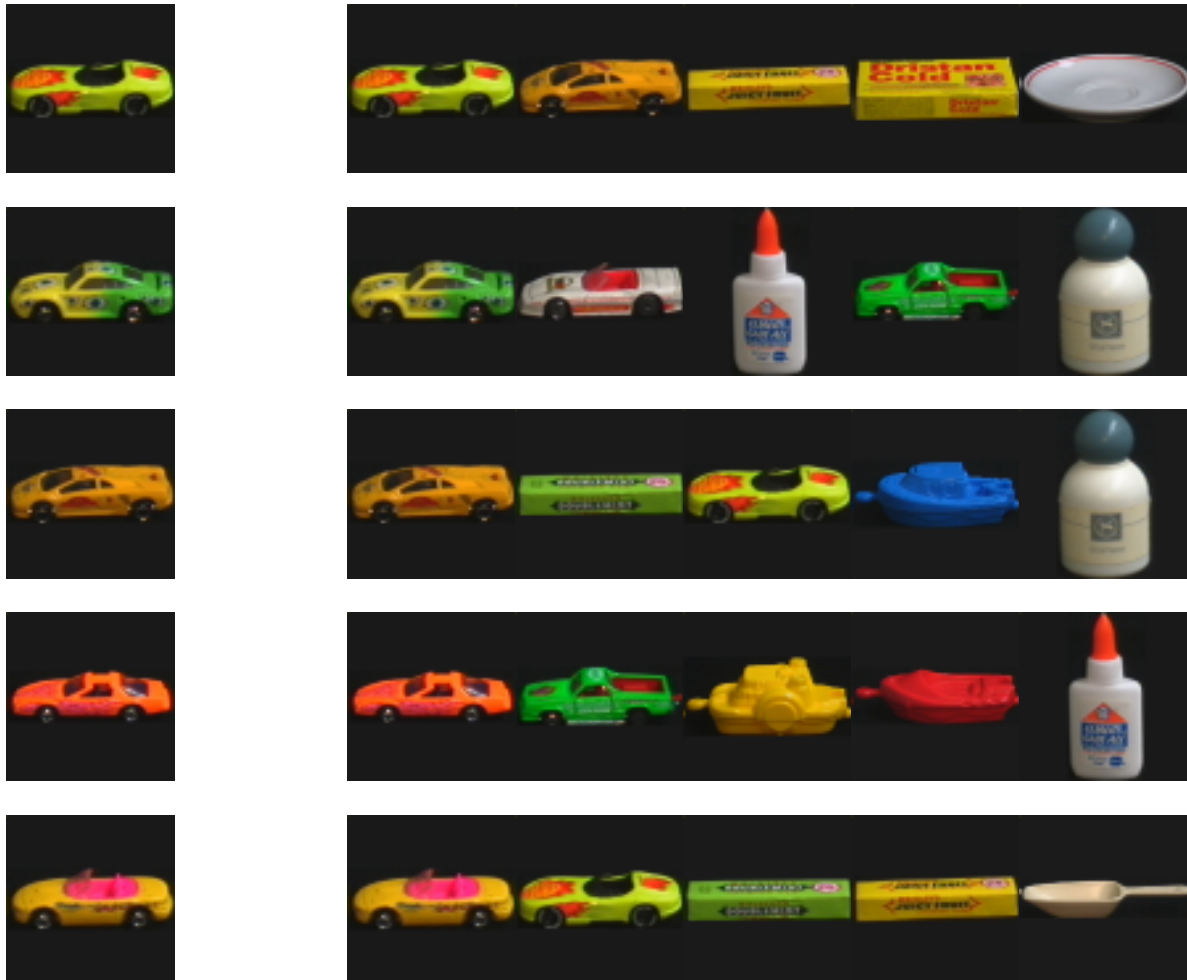


FIG. 8.15: Recherche des voitures dans la base COIL : algorithme de Cristina + «graduated assignment»

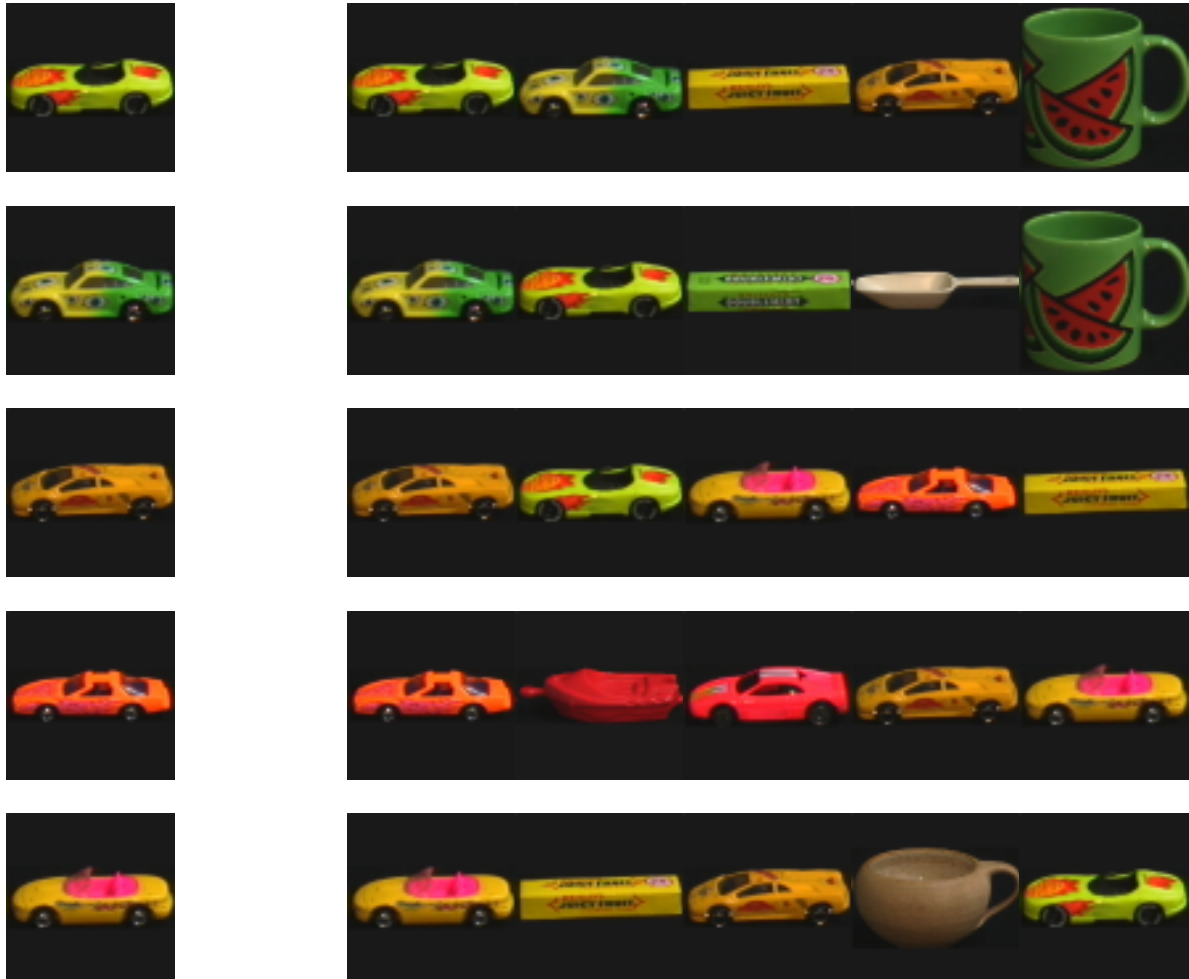


FIG. 8.16: Recherche des voitures dans la base COIL : algorithme d'appariement d'arbres

mesure des distances entre les arbres. Ceci est important car dans le cas de la recherche dans une base de données, les arbres peuvent être construits à l'avance. Ceci n'est pas le cas dans l'approche Cristina qui nécessite de réajuster les segmentations (donc les graphes en résultant) à chaque comparaison.

Le tableau suivant récapitule les temps de calcul pour comparer les images par les graphes (méthode de Cristina) et par les arbres, avec et sans précalcul de la base.

Type d'appariement :	Cristina	Arbres sans précalcul	Arbres avec précalcul
Temps de calcul (secondes) :	0.97	1.51	0.9

TAB. 8.4: Temps de calcul d'une comparaison entre deux images par les graphes (Cristina) ou par les arbres.

8.2.4 Conclusion

Les résultats offerts par la modélisation par arbres sont nettement meilleurs que ceux utilisant l'ancien algorithme. Bien que cela soit difficile à quantifier, les images obtenues présentent en effet une bien meilleure cohérence visuelle, les objets semblent plus «se ressembler».

Heureusement, nos mesures confirment effectivement cette impression car il est important de noter cet aspect subjectif de l'évaluation. En effet, malgré notre souhait de chercher des mesures objectives pour valider notre approche, nous devons admettre que le but de la recherche par l'exemple est de renvoyer à l'utilisateur les images que *lui-même* estime les plus proches. L'ultime arbitre des performances d'un tel algorithme est donc entièrement subjectif.

Dans le futur, nous chercherons à tester notre approche sur des problèmes de classification d'images disposant d'une vérité-terrain, afin de vérifier que notre modélisation et notre résolution représentent réellement une simulation valable des comportements humains de comparaison d'images.

Chapitre 9

Application à la vidéosurveillance

*La différence entre la théorie et la pratique,
c'est qu'en théorie, il n'y a pas de
différence entre la théorie et la pratique,
mais qu'en pratique, il y en a une.*

Dans ce chapitre nous présenterons l'ensemble des outils à notre disposition pour construire un prototype d'application de type vidéosurveillance. Celle-ci se décomposera en trois phases : la détection et le suivi de personnes dans une scène d'intérieur, l'extraction d'un signalement pour chacune de ces personnes et enfin la comparaison de ces signalements.

La première phase, la détection et le suivi de personnes, a été réalisée dans le cadre d'un projet en collaboration avec l'INRIA et ST Microelectronics. Nous avons implémenté une méthode de détection robuste et rapide de la silhouette d'individus dans des séquences vidéo, avec des contraintes d'exécution en temps réel.

Dans la phase suivante, nous utiliserons nos modélisations à base d'arbres (chapitre 6) pour la création de signatures robustes des individus détectés. Enfin, nous étudierons l'utilisation des comparaisons d'arbres du chapitre 7 pour le suivi de personnes dans le temps ou dans un réseau de caméras, ou pour tout autre tâche visant à structurer l'information et faciliter son interprétation.

9.1 La problématique de la vidéosurveillance

9.1.1 Introduction

La vidéosurveillance consiste en l'utilisation de caméras pour se substituer à la présence humaine dans un objectif de sûreté ou de sécurité. Cette surveillance peut être effectuée en temps réel, ce qui permet de déclencher des interventions de secours ou de maintien de l'ordre, ou a posteriori pour comprendre les causes d'un événement, par exemple en retraçant le parcours des personnes présentes. Ces deux tâches sont encore effectuées essentiellement de façon manuelle. Étudions d'abord les améliorations possibles dans le premier cas, celui de la surveillance en temps réel.

L'approche manuelle se contente de caméras simples envoyant l'intégralité de leurs données aux opérateurs distants installés dans un centre de surveillance. Le problème est qu'il est humainement difficile de surveiller simultanément un grand nombre de caméras, en particulier quand les événements à détecter ne durent que quelques secondes (figure 9.1).

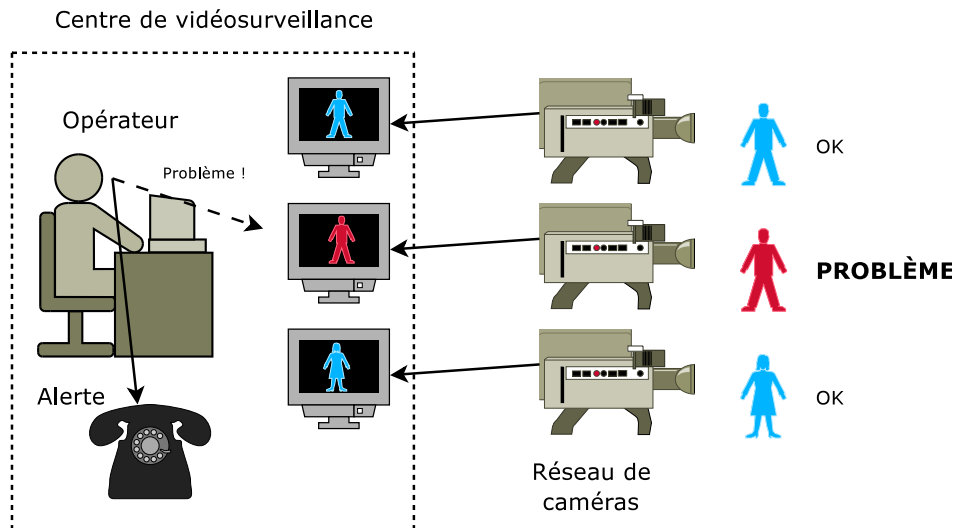


FIG. 9.1: Schéma traditionnel de vidéosurveillance : un ou plusieurs opérateurs devant de nombreux écrans. Les situations problématiques peuvent passer inaperçues.

Les recherches les plus audacieuses visent à automatiser entièrement le rôle des opérateurs. Des systèmes experts se chargeraient alors de détecter les scénarios anormaux et de déclencher les alertes appropriées. De façon plus évolutive, on peut chercher à développer des systèmes automatiques pour hiérarchiser l'information en ne présentant à l'opérateur que les images les plus pertinentes. Par exemple, en n'affichant que les images contenant des personnes (figure 9.2).

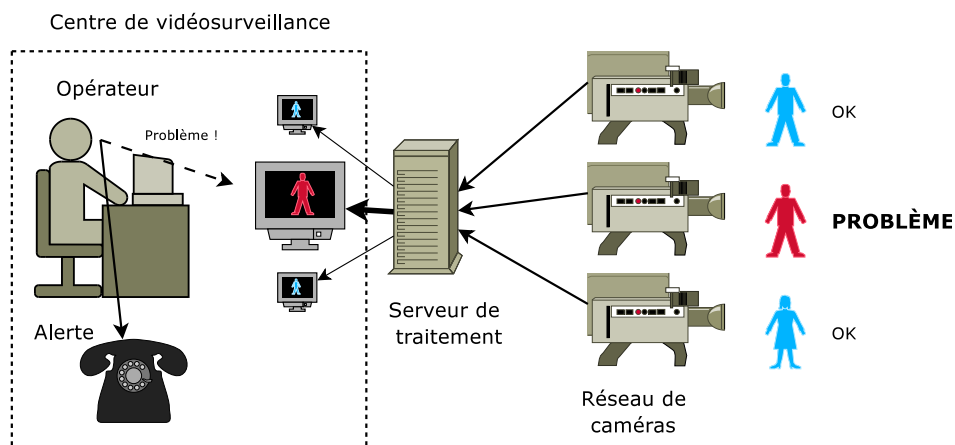


FIG. 9.2: Vidéosurveillance semi-automatique : un serveur de traitement filtre les images et affiche en priorité les images présentant un intérêt pour l'opérateur.

Le second problème, la compréhension a posteriori d'événements, nécessite d'être capable de suivre le parcours des individus. Dans le triste exemple des attentats de Londres de Juillet 2005, la police anglaise a ainsi du visionner 1800 heures de film pour retrouver des informations sur les



FIG. 9.3: Exemple d'image de vidéosurveillance : les responsables des attentats de Londres du 7 juillet 2005. Ces images de mauvaise qualité sont bien représentatives de la difficulté de l'analyse automatique (Source : London Metropolitan Police).

terroristes. On peut imaginer un système automatique facilitant la tâche qui d'une part suivrait les individus en remontant le temps, ou qui chercherait dans les vues antérieures les images présentant les mêmes individus.

Dans les deux cas, ces automatisations des tâches de surveillance nécessitent de développer des routines capables non seulement de détecter les individus présents dans une scène, mais également de les suivre, voire de les retrouver, de façon robuste, et peut-être de détecter des scénarios classiques : chute de personne sur les rails, agression, colis abandonné, etc.

Nous ne prétendons pas résoudre tous ces problèmes dans ce chapitre, mais peut-être contribuer à une meilleure compréhension des enjeux et difficultés qu'on y rencontre.

9.1.2 Les spécificités au niveau image

Les images de vidéosurveillance possèdent des caractéristiques uniques qui les rendent différentes des champs d'application traditionnels du traitement d'images (voir figure 9.3).

En premier lieu, il faut noter la faible qualité des images. En effet, l'intérêt d'un réseau de caméras dépend fortement de sa couverture et de l'absence d'angles morts. Il faut en général installer un grand nombre de caméras, qui ne pourront donc pas être des caméras de grande qualité. Par ailleurs, il faut également considérer dans le budget leur robustesse face aux agressions extérieures, que celles-ci soient environnementales ou d'origine humaine, et éventuellement prévoir d'en remplacer fréquemment. De plus, les conditions d'éclairage sont rarement optimales et peuvent même être assez difficiles, par exemple dans le cas des vues d'extérieur. Enfin, il ne faut pas négliger l'importance de la bande passante disponible sur le réseau reliant les caméras au centre de traitement. Avoir beaucoup de caméras signifie disposer d'un réseau à forte bande passante et/ou transmettre des images très

fortement compressées.

Au niveau du contenu des images, contrairement aux images de vidéoconférence ou de films, il n'y a pas de sujet principal présent sur l'image. Au contraire, chaque individu est filmé à grande distance. Les motifs précis du visage ne sont donc que rarement visibles. De plus, chaque caméra verra l'individu sous un angle différent, et les occultations, causées par d'autres personnes ou des objets de la scène, masqueront certaines zones de l'objet. Tout ceci nécessite le développement d'un signalement robuste de chaque personne.

9.1.3 Architecture distribuée

Parmi les problèmes mentionnés ci-dessus, un des plus importants est le nombre de caméras présentes dans le réseau. Installer plus de caméras implique l'utilisation d'un réseau de communication plus gros et de serveurs de traitement plus puissants. Par exemple, la Préfecture de Police de Paris prévoit pour fin 2007 de traiter dans son centre «TN Métro» les images de 6540 caméras couvrant le seul réseau de la RATP. Au total l'équipement des réseaux franciliens et des centres de traitement représente un budget de 13 millions d'euros [Huchon, 2005].

Une solution possible pour réduire ces coûts consisterait à hiérarchiser le traitement de l'information, en effectuant le prétraitement des images dans les caméras (ou dans des serveurs de traitement intermédiaires). La figure 9.4 montre comment des caméras «intelligentes» seraient capables d'estimer automatiquement l'intérêt des images qu'elles produisent. Ceci permet de ne pas surcharger le serveur de traitement, voire d'augmenter le nombre de caméras sans accroître la puissance de traitement du serveur, ni la bande passante du réseau, ou encore la capacité des systèmes de stockage.

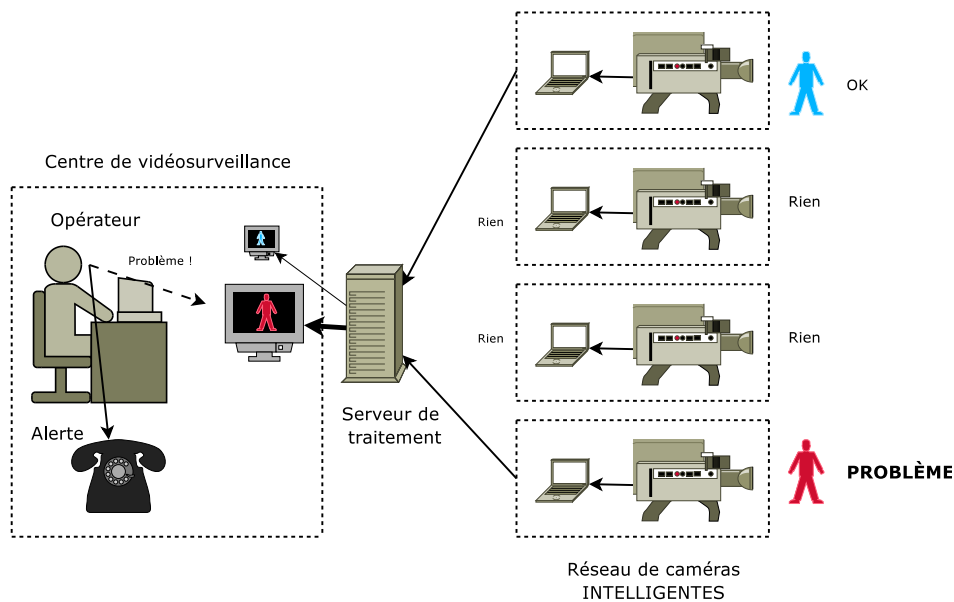


FIG. 9.4: Vidéosurveillance semi-automatique avec des traitements intermédiaires : chaque caméra (ou groupe de caméras) est doté d'une puissance de traitement permettant de décider de si l'image mérite d'être envoyée au serveur principal pour des traitements plus lourds.

Cette approche hiérarchique correspond à l'organisation de notre méthode : d'abord la détection des objets, puis leur modélisation et enfin leur comparaison. La première opération peut être effectuée au moins partiellement par les caméras «intelligentes». La modélisation nécessite une certaine puissance de calcul qui serait plutôt réservée à des serveurs de traitement. La comparaison de signalements

peut être relativement rapide, mais nécessite pour être distribuée de connaître la topologie du réseau, ce qui sort du cadre de notre travail. Nous la considérerons donc intégrée au serveur de traitement.

9.2 Calcul de la silhouette

9.2.1 Introduction

Cette première étape dans la chaîne complète de vidéosurveillance a été réalisée dans le cadre d'un projet en collaboration avec le groupe ORION de l'INRIA et parrainé par ST Microelectronics.

Le projet **PS26-27 Environnement Intelligent** vise à produire des démonstrateurs de technologies dans le domaine de la surveillance en environnement domestique ou hospitalier. Un environnement intelligent est un «*système informatique immergé dans un environnement capable de percevoir, de réagir en fonction des changements de l'environnement et doué de capacité de raisonnement. Il devance les besoins de l'utilisateur et lui permet d'interagir directement avec son environnement*».

La plate-forme de base est une ou plusieurs caméra(s) ST Microelectronics liée(s) à un serveur de traitement responsable de l'analyse des scènes. Une perspective est l'embarquement de tout ou partie des algorithmes de traitement dans des caméras «intelligentes» disposant d'une capacité de traitement local. Ceci permettrait de soulager le serveur principal qui serait libre d'effectuer des analyses de plus haut niveau (suivi multi-caméras, détections de scénarios, déclenchement d'alertes).

Le rôle du CMM dans le projet est de fournir des algorithmes performants de détection de la personne et de segmentation de la silhouette, qui seront couplés aux algorithmes de détection de postures du groupe ORION de l'INRIA. L'embarquement sur caméra n'est pas encore le point focal, mais les contraintes imposées sont un traitement en temps réel (ou quasi-réel) sur un ordinateur de type PC moderne. La méthode de détection des postures est décrite dans [Boulay et al., 2005] et [Boulay et al., 2006].

Imageurs bon marchés

La condition *sine qua non* du projet est de fonctionner sur des imageurs d'entrée de gamme, portés par le marché de la «photographie» sur les téléphones mobiles. L'inconvénient étant que les images obtenues ne sont pas de bonne qualité, en particulier quand les conditions de prise de vue se dégradent. On rappellera ici la figure 9.5 déjà présentée à la fin du chapitre 3 et présentant certaines pathologies liées aux conditions difficiles d'acquisition des images.

Solutions retenues

La méthode que nous avons retenue est basée sur une approche en plusieurs passes, dans laquelle on raffine progressivement l'information traitée. On commence donc d'abord par repérer grossièrement la personne par une différence d'images avec le fond. On approche ensuite la silhouette en écartant les ombres projetées, puis on s'accroche aux contours réels de l'image. Enfin, on envoie la silhouette soit au moteur de détection de posture, soit à l'algorithme de calcul de suivi d'objets (voir figure 9.6).

Nous verrons dans ce qui suit les différentes étapes du processus, ainsi que les divers modules optionnels que nous avons développés. Ces modules (détection des ombres, utilisation du mouvement) offrent un large panel d'options pour atteindre un compromis entre vitesse de traitement et faible taux d'erreurs.



FIG. 9.5: Images acquises par un imageur ST dans des conditions non-idéales. Le bruit devient sensible lorsque la lumière baisse, de même que les dérives chromatiques dues à la compensation automatique. Les décrochages sont causés par des problèmes de synchronisation entre l'imageur et le serveur de traitement.

9.2.2 Première passe

La première étape consiste à calculer pour chaque pixel la différence entre l'image courante et l'image du fond. En pratique, cette différence est calculée dans l'espace RGB. Les tests portant sur les différents espaces couleur ayant montré que les espaces couleur plus complexes n'étaient pas suffisamment avantageux pour justifier le coût d'une conversion (les caméras fournissant nativement du RGB). Cette approche a été préférée à d'autres techniques de modélisation du fond car elle nous permettait plus facilement de nous insérer dans la plate-forme développée par l'INRIA.

La différence permet en général de récupérer un «blob» qui indique la position de la personne ainsi qu'une idée approximative de ses contours. Comme on le voit dans la figure 9.7, le «blob» n'est pas nécessairement connexe ou uniforme. Par ailleurs on voit nettement une superposition des textures : le plancher apparaît nettement à travers la chemise. Ce blob n'est donc pas facilement utilisable en l'état, il va falloir déterminer ses contours avec plus de précision et de robustesse.

Algorithme 13 Détection des «blobs» à partir de l'image courante et de l'image du fond

ENTRÉES: Image d'entrée (couleur) *imCurrent* et image du fond (couleur) *imBG*

SORTIES: Image de sortie (binaire) des blobs et liste des boîtes englobantes des blobs à traiter

imDiff (binaire) ← différence dans RGB et seuillage entre *imCurrent* et *imBG*

bboxList ← liste des boîtes englobantes de chaque composante connexe de *imDiff*

augmentation de la taille de chaque boîte

fusion des boîtes se recouvrant

élimination des boîtes trop petites

return *bboxList* et *imDiff*

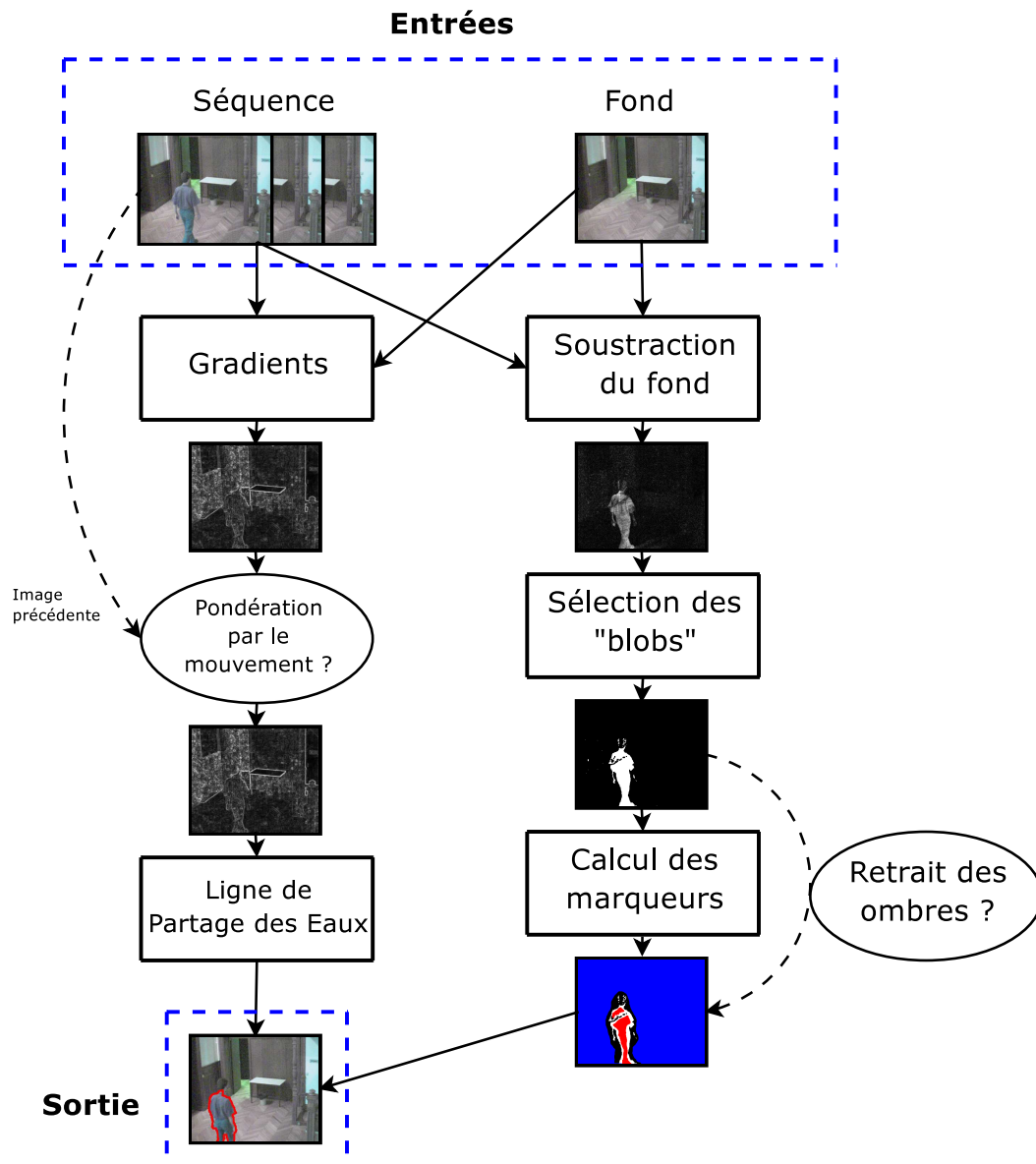


FIG. 9.6: Vue générale de l'algorithme de détection de silhouettes. La sortie est envoyée au moteur de détection de postures.



Fond

Image courante

Différence RGB

FIG. 9.7: Distance RGB entre l'image courante et l'image du fond

9.2.3 Détection d'ombres portées

Le problème des ombres portées est un problème très important en traitement de séquences vidéo. De nombreuses approches complexe d'analyse de scène visant à rattacher les ombres aux objets sont actuellement incompatibles avec les contraintes temps-réel. Ceci nous a poussé à utiliser une approche plus simple, basée sur la détection d'invariants chromatiques. Nous avons utilisé un sous-ensemble de la méthode décrite par Cavallaro dans [Cavallaro et al., 2004], en nous restreignant aux opérations sur les pixels. Les opérations plus complexes de la seconde partie du papier n'étant pas compatibles avec le reste de notre approche.

Cette méthode améliore la robustesse d'une approche relativement ancienne basée sur le fait qu'une ombre portée ne modifie par la chrominance de la région passant à l'ombre mais simplement qu'elle diminue l'intensité lumineuse renvoyée par cette surface. L'approche consiste donc à comparer la région supposée être à l'ombre avec celle de l'image de référence, et déterminer s'il s'agit d'un nouvel objet ou simplement d'une ombre portée.

L'avantage de cette méthode est sa relative simplicité, car elle ne nécessite que quelques comparaisons de valeurs de pixels. L'inconvénient est qu'elle ne fonctionne correctement que sur des zones non achromatiques, il est donc difficile de décider si un objet gris sombre apparaissant sur un fond gris clair est réellement un objet ou une ombre portée. Conscients de ces limitations, nous allons pouvoir réduire les faux positifs de la détection de silhouette avec un bon taux de réussite.

9.2.4 Segmentation complète

Comme on l'a rappelé au début du chapitre 5, la segmentation morphologique de la silhouette comporte trois étapes : d'une part le calcul d'une image gradient, ensuite (ou en parallèle) le calcul de marqueurs intérieurs et extérieurs, et enfin l'inondation du gradient depuis ces marqueurs.

L'image gradient

La différence d'images nous donne une idée générale de la position de l'objet, et de celle du fond. Pour déterminer au mieux la silhouette de la personne détectée, nous allons toutefois devoir utiliser plus d'information que celle présente dans l'image des différences. En effet, cette image présente une fusion des informations de l'image courante et de l'image du fond. Une détection des contours sur cette image risque donc souvent d'être insuffisante pour piloter une bonne segmentation.

Un gradient couleur de l'image courante retrouve généralement bien la forme des objets, mais présente lui aussi des zones de valeurs trop faibles, qui risque de générer des «fuites» (voir fig 9.8).

Pour contourner ce problème, la méthode que nous proposons consiste donc à fusionner les deux gradients pour que leurs forces et faiblesses se compensent. On va ainsi dilater le gradient dont l'information est la plus incertaine (celui des différences) et lui ajouter celui qui est le plus précis (le gradient de l'image courante). Le résultat de cette manipulation est un gradient plus robuste face aux fuites (voir fig 9.9). Nous verrons plus bas une méthode utilisant le mouvement pour stabiliser le gradient et améliorer encore ce mécanisme de prévention des erreurs de segmentation.

Les marqueurs et l'inondation

On calcule les marqueurs intérieurs et extérieurs à partir du "blob" de l'image de différences, en filtrant celui-ci, et en l'érodant dans un cas (marqueur intérieur) et en le dilatant dans l'autre cas (marqueur extérieur). Le rôle de l'inondation sera de retrouver le meilleur contour entre ces marqueurs.



FIG. 9.8: L'utilisation directe du gradient couleur peut causer des "fuites" lors de l'inondation. Ainsi le gradient très fort le long des tiroirs du bureau va fixer l'inondation et causer une erreur de segmentation sur la gauche de la silhouette.

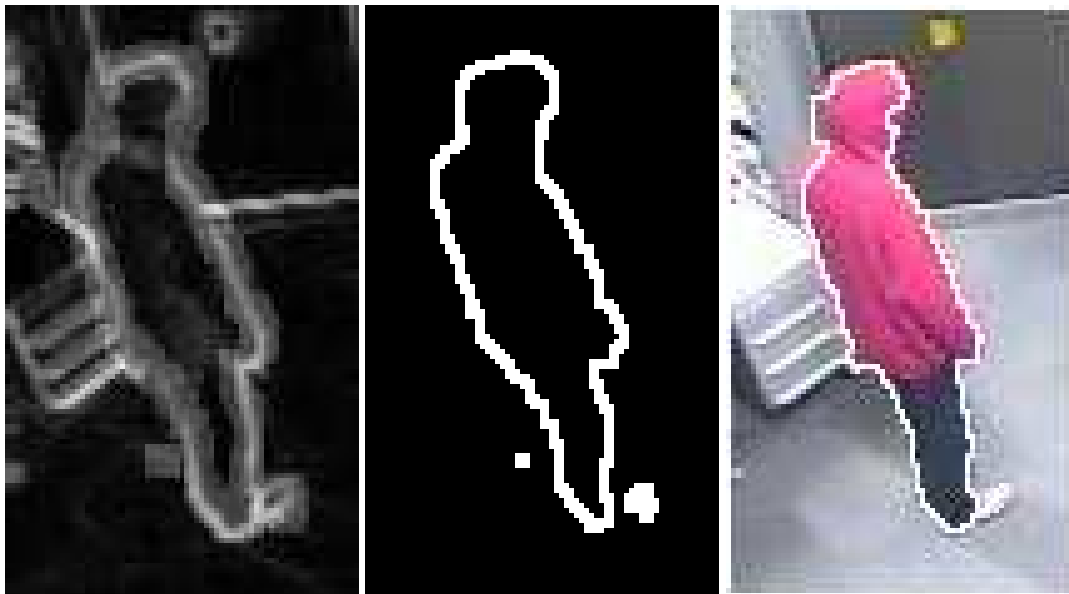


FIG. 9.9: Mélanger le gradient couleur direct (gauche) au gradient de l'image des différences après filtrage et détection du blob (centre) permet de récupérer suffisamment d'information sur la position du contour pour éviter les fuites.

Algorithme 14 Retrait des ombres et calcul des marqueurs.

ENTRÉES: Image d'entrée (couleur) $imCurrent$ et image du fond (couleur) $imBG$ et image des blobs (binaire) $imDiff$

SORTIES: Image de sortie des marqueurs intérieurs et extérieurs $imMark$

$imLabels \leftarrow \mathbf{Label}(imDiff)$ *Étiquetage des composantes connexes*

$imShadows$ (binaire) $\leftarrow \mathbf{Shadows}(imCurrent, imBG)$ *détection des ombres portées*

$imDiffNoShadow$ (binaire) $\leftarrow \mathbf{Inf}(imShadows, imDiff)$ *On retire les ombres du blob*

$imIntMark$ (binaire) $\leftarrow \mathbf{Érosion}(imDiffNoShadow)$ *Érosion du marqueur pour obtenir un marqueur intérieur. La taille de l'érosion dépend de la taille du blob, obtenue à partir de sa boîte englobante.*

$imIntMark = \mathbf{Multiplier}(imIntMark, imLabels)$ *Projeter $imLabels$ sur $imIntMark$ pour récupérer les étiquettes, ainsi un objet coupé en deux par l'érosion ne sera pas séparé*

$imExtMark$ (binaire) $\leftarrow \mathbf{Dilatation}(imDiff)$ *Dilatation du marqueur pour obtenir un marqueur extérieur. On prend l'image avec les ombres pour éviter de marquer dans l'extérieur des zones intérieures montrant un faux positif dans la détection des ombres*

Inverser $imExtMark$

$imExtMark \leftarrow \mathbf{Multiplier}(imExtMark, \max(imLabels) + 1)$

$imMark \leftarrow \mathbf{Sup}(imIntMark, imExtMark)$ *$imMark$ sera donc marqué 1,2,...,N pour les objets, N+1 pour le fond et 0 pour la zone d'incertitude.*

Les zones d'ombre sont éliminées du marqueur intérieur, pour éviter les faux-positifs, mais on les ajoute pas pour autant au marqueur extérieur, ce qui permet de limiter les faux-négatifs lorsqu'on détecte trop d'ombres. En effet, on a mentionné plus haut le problème de l'individu gris sombre sur un fond gris clair qui sera incorrectement classé comme ombre. Grâce à cette gestion prudente de l'information des ombres, le marqueur extérieur sera correct et le marqueur intérieur contiendra les régions qui auront échappé à la détection d'ombre incorrecte.

9.2.5 Boîtes englobantes

La chaîne de traitement est relativement simple, mais pas toujours assez pour fonctionner suffisamment rapidement, une étape d'optimisation a donc été nécessaire.

Une rapide étude du prototype et des séquences-type a montré que l'essentiel du temps était perdu à traiter des pixels situés très loin des zones d'intérêt. Nous avons donc procédé à une présélection des zones d'intérêt à partir de l'image des différences, le reste du traitement étant inchangé, si ce n'est qu'il est effectué sur des images beaucoup plus petites (fig 9.11).

Cette méthode simple de gestion de l'information nous a facilement permis d'atteindre les performances voulues, puisque l'ensemble du traitement (lecture/écriture des images non comprises) prend entre 20 et 40ms, soit un taux de 25 à 50 images par secondes, sur un Pentium®4 cadencé à 2.8GHz, pour des images 388x284 en couleur. Il reste encore une grande marge d'optimisation en utilisant des versions dédiées de nos algorithmes.

Par ailleurs, nous en avons profité pour obtenir des informations sur la taille des objets via le diamètre de leur boîte englobante que nous pouvons utiliser pour choisir les tailles d'érosion et de dilatation plutôt que de recourir à des chiffres arbitrairement choisis.



FIG. 9.10: Résultats : détection de la personne dans deux séries de séquences. La seconde série est plus difficile car l'éclairage est médiocre et l'image très bruitée.



FIG. 9.11: La détection rapide des zones d'intérêt permet d'accélérer le traitement par un facteur 100 sur certains jeux de données.

9.2.6 Stabilisation du contour

La chaîne présentée jusqu'ici fonctionne correctement et offre de bonnes performances, ce qui libère un maximum de temps pour le système de détection de postures.

Cependant, le problème des fuites présenté plus haut persiste même si les fusions de gradients permettent en général de les limiter. Pour une solution plus permanente, il faut d'abord comprendre les causes qui entraînent ces fuites.

Analyse des fuites

Une fuite apparaît lorsqu'il existe entre les marqueurs intérieurs et extérieurs un contour plus fort que celui de l'objet. Ce contour bloque l'inondation provenant des marqueurs extérieurs suffisamment longtemps pour que celle provenant des marqueurs intérieurs «déborde» du contour de l'objet dans la zone intermédiaire (voir figure 9.12) Une première solution est d'améliorer les marqueurs pour qu'ils encadrent mieux le bon contour. Dans notre situation, ceci ne s'est pas avéré possible.

Solution par le mouvement

Si on suppose que la silhouette précédente a été correctement détectée, on peut utiliser cette information pour diminuer la force du contour parasite (voir figure 9.13). On va ainsi privilégier les contours proches de l'objet en pondérant la force des contours par une fonction décroissante de leur distance au contour obtenu pour la trame précédente. Pour corriger les déplacements dûs au mouvements de l'objet, on effectue un recalage du contour en calculant le champ de mouvement entre la trame précédente et la trame courante.

Cette opération se passe donc en trois temps :

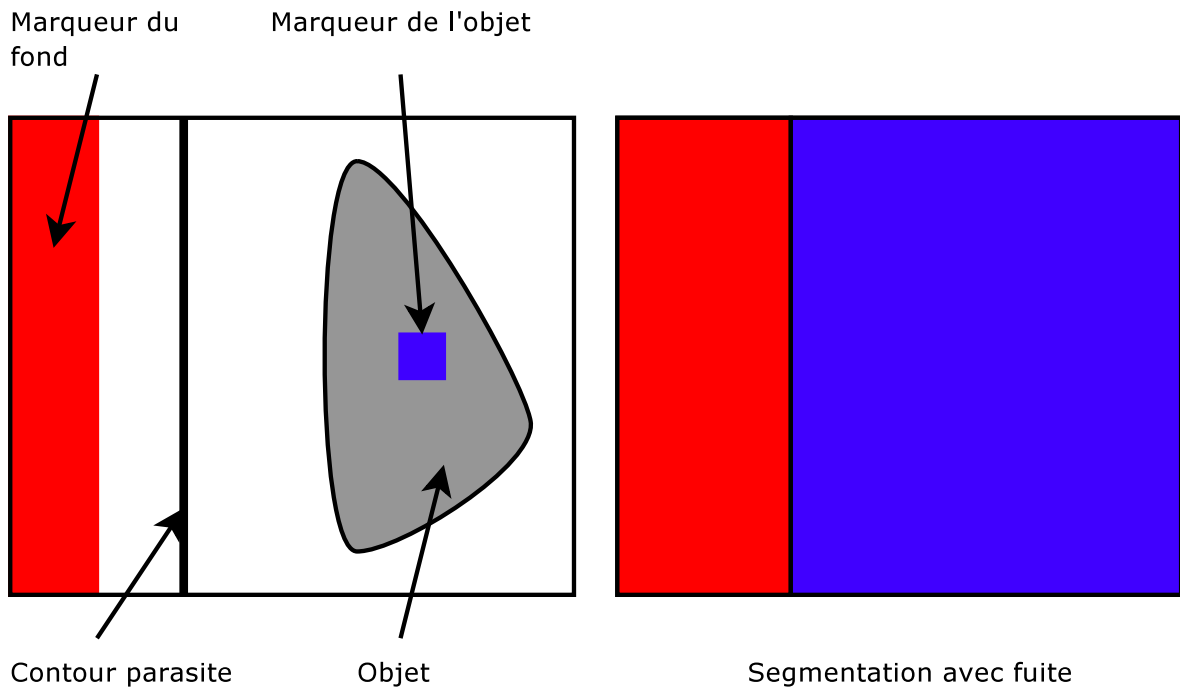


FIG. 9.12: La ligne de partage des eaux s'attache à un contour fort dans la zone intermédiaire plutôt qu'au contour de l'objet.

1. recalcr la silhouette obtenue au temps $t - 1$, selon le champ de mouvement calculé entre les trames aux temps $t - 1$ et t .
2. calculer une fonction distance autour du contour de la silhouette recalée.
3. pondérer le gradient en chaque point en fonction de sa distance au contour

La pondération actuellement utilisée est une fonction linéairement décroissante jusqu'à une distance donnée, et zéro au-delà de cette distance. La pente et la distance de seuil sont pour l'heure des constantes définies par des heuristiques, mais on peut envisager un mécanisme plus fin se basant par exemple sur la valeur moyenne des vecteurs mouvements ainsi que sur la taille de l'objet.

Résultats

La figure 9.14 présente l'amélioration obtenue en recalant le résultat précédent. On voit que l'ombre initialement rattachée à la silhouette a été correctement séparée. On voit aussi que l'on a propagé le «trou» dans le bras gauche déjà présent dans la trame précédente, heureusement celui-ci ne gêne pas la détection de posture.

Malgré ces détails légèrement pénalisants, l'objectif principal de stabilisation du contour est atteint.

9.3 Création et suivi d'une signature

9.3.1 Motivation

Dans le domaine de la détection de posture comme dans d'autres scénarios de vidéosurveillance, un problème fréquent est le suivi et l'identification des individus. Le problème se pose de façon par-

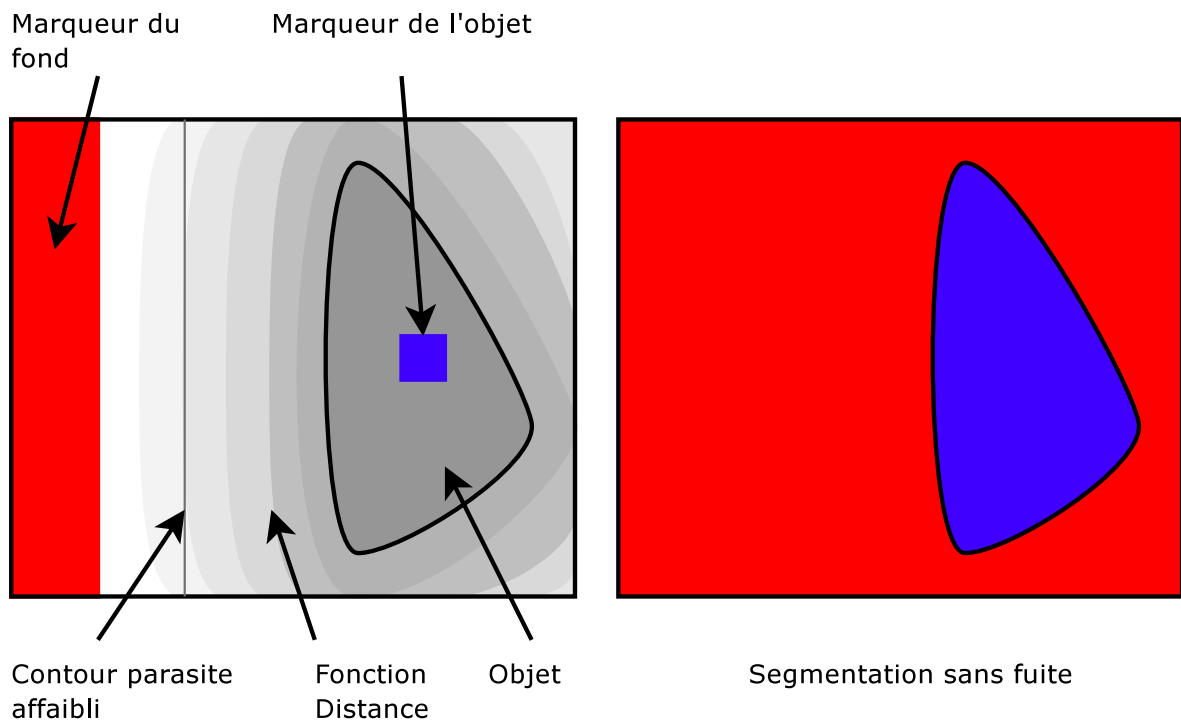


FIG. 9.13: On diminue les contour distants de l'objet, ce qui replace le contour réel de l'objet comme plus fort contour entre les deux marqueurs.

Algorithme 15 Algorithme complet de calcul du gradient

ENTRÉES: Image d'entrée (couleur) *imCurrent*, image précédente (couleur) *imPrevious*, contour précédent *imContourPrevious* et image des blobs (binaire) *imDiffCurrent*

SORTIES: Image de gradient robuste *imGrad*

imGradColor ← **Gradient**(*imCurrent*) *gradient couleur RGB*

imGradDiff ← **Gradient**(*imDiffCurrent*) *gradient morphologique simple*

imGrad = $\lambda \cdot \text{imGradColor} + (1 - \lambda) \cdot \text{imGradDiff}$ *Fusion des gradients*

imMotion ← **Motion**(*imPrevious*, *imCurrent*) *Champ de mouvement entre imPrevious et imCurrent*

imContourCompensated ← **MotionCompensate**(*imContourPrevious*, *imMotion*)
Contours précédents, recalés grâce au mouvement

imDistance ← **Distance**(*imContourCompensated*) *fonction distance au contour recalé*

imGrad = $\frac{\text{imGrad}}{1 + \frac{1}{d} \text{imDistance}}$ *Atténuation de imGrad en fonction de la distance au résultat précédent*



FIG. 9.14: Toutes les étapes d'amélioration du gradient avant l'inondation. L'utilisation du résultat précédent a permis d'éviter la fuite dans la zone d'ombre sous le bureau. En revanche, on a aussi propagé la mauvaise détection du bras gauche, perdu quelques trames auparavant.



Sans stabilisation

Avec stabilisation

FIG. 9.15: Comparaison des résultats avec et sans la stabilisation

ticulièrement ardue lorsqu'il n'y a pas de continuité entre les observations de la personne. Quand la personne suivie reste dans le champ de la caméra, le problème est relativement simple, à condition bien sûr de n'avoir que des occultations partielles et pas ou peu d'objets parasites (autres personnes, objets en mouvement, etc). Le problème se pose de façon forte lorsque la personne est passée hors-cadre et que le système détecte un nouvel arrivant. Cette nouvelle personne est-elle la première qui revient ou est-elle quelqu'un de complètement différent ?

Pour répondre à cette question, la solution que nous proposons est de calculer une signature qui représente aussi fidèlement que possible le sujet suivi et qui se prête bien à la comparaison.

La solution la plus classique consiste à comparer des histogrammes couleur. Cette solution parfaitement valide et efficace sera sans doute préférée par les systèmes ayant des contraintes de simplicité ou de rapidité. Le calcul de la silhouette permet notamment d'échantillonner uniquement sur l'objet, ce qui n'est pas le cas dans les nombreuses approches plus simplistes que l'on trouve dans la littérature. Par ailleurs, elle peut servir de filtre préalable à l'utilisation de techniques plus complexes.

Cependant, l'étude menée dans le chapitre 7 montre que les histogrammes ne sont pas les outils les plus discriminants et que les comparaisons d'arbres offrent des résultats plus riches. Par ailleurs, l'utilisation des arbres permet de comparer des objets de couleurs identiques mais variant dans l'agencement de ces couleurs. Cet atout est particulièrement intéressant dans le cadre de la vidéosurveillance dans lequel les images sont souvent de faible qualité, et où l'information chromatique est rarement déterminante. Quelle que soit la solution ultimement retenue, il est toujours intéressant de disposer d'un riche continuum de compromis entre qualité des résultats et rapidité.

9.3.2 Modélisation de l'individu

Le travail réalisé plus haut visait l'extraction précise et robuste de la silhouette d'une personne. Ceci nous place exactement dans le cadre des resegmentations décrites dans le chapitre 6. En effet, la détection de la silhouette fonctionnant bien, on peut calculer la segmentation «top-down» de la pyramide de nivellements à l'intérieur en imposant l'image silhouette/fond comme segmentation de référence initiale.

9.3.3 Classification

La figure 9.17 présente les images que nous allons utiliser. Il y a trois individus dans les mêmes locaux, le problème est la détection de l'individu et sa classification. Nous allons segmenter chaque personne en utilisant l'algorithme décrit plus haut, le modéliser par les méthodes «top-down» du chapitre 6 et comparer les arbres obtenus par l'algorithme du chapitre 7.

La figure 9.18 présente les résultats de l'algorithme de comparaison d'arbres présenté dans le chapitre 7 appliqué aux silhouettes segmentées par la méthode décrite ici. L'image de gauche est la requête et les images de droite sont les résultats triés par ordre de pertinence. On voit que les personnages sont correctement classifiés, même dans des postures différentes. En particulier, on voit qu'il ne serait pas possible dans cette application d'utiliser des critères de forme ou de position, car les régions se déforment et les personnes ne restent pas verticales. Seules les relations d'inclusions successives ou d'adjacence restent valides entre les différentes postures.

Les prochaines étapes de ce travail consisteront en l'analyse et la modélisation de cas plus difficiles. Dans les exemples illustrant ce chapitre, la difficulté réside dans la segmentation des silhouettes (à laquelle s'ajoute la contrainte de rapidité), mais la comparaison des modèles peut s'effectuer relativement facilement. En revanche, on a vu dans le chapitre 7 que nos modélisations par arbres offraient



FIG. 9.16: Modélisation de la silhouette par un arbre de resegmentations «top-down».

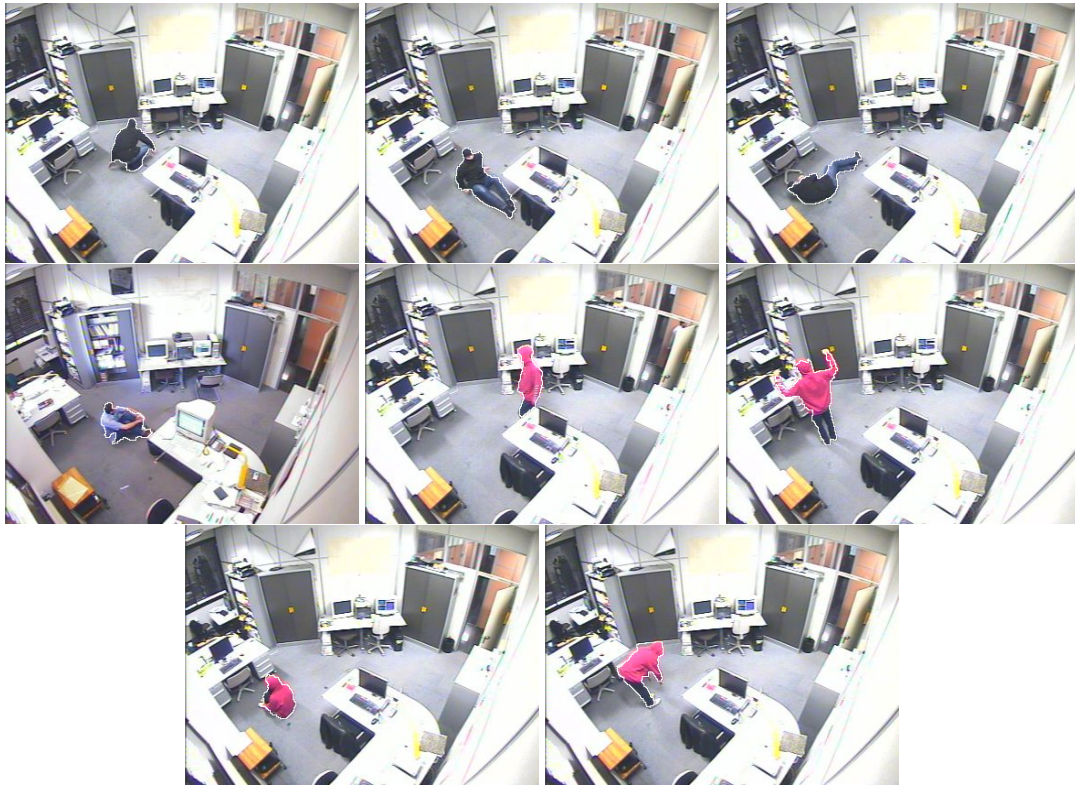


FIG. 9.17: Vues de trois personnes dans les mêmes locaux, et segmentation de la personne.

une réelle amélioration par rapport aux histogrammes couleur. On peut donc penser qu'ils offriront une meilleure finesse dans des scénarios plus complexes que ceux présentés ici.

Par exemple, il serait relativement facile et intéressant d'utiliser les résultats de l'algorithme de détection de postures pour tenter d'identifier les membres, le tronc, etc. Une possibilité serait d'utiliser cette information pour privilégier par exemple les similarités «bras-bras» et «jambe-jambe» et pénaliser les similarités «bras-jambe».

9.3.4 Suivi

Le problème du suivi de personnes est double : d'une part on peut chercher pour chaque trame de la séquence à repérer la personne, ou on peut chercher à la repérer uniquement dans les changements majeurs. Dans le premier cas, on utilisera un système simple et rapide. Dans le second cas, on pourra envisager l'utilisation d'un système plus complexe car devant être plus robuste. En effet, la variabilité de la signature entre deux vues distantes dans le temps peut être grande.

Notre test consistera à repérer laquelle des personnes A ou B est présente dans la séquence. En l'occurrence, il s'agit de la personne A (voir figure 9.19). Dans les deux cas, la détection de la silhouette sera un prérequis à la méthode de comparaison. Le premier essai utilisera un histogramme couleur du type de ceux décrits au chapitre 7 et le second utilisera nos modélisations par arbres.

La figure 9.20 présente les résultats de la comparaison des silhouettes de la séquence avec les signatures A et B, en utilisant les distances d'histogrammes. La figure 9.21 montre les mêmes comparaisons en utilisant les distances d'arbres. Curieusement, on voit que les résultats sont moins bien séparés par cette approche. Deux explications à ce phénomène : d'une part la couleur moyenne du

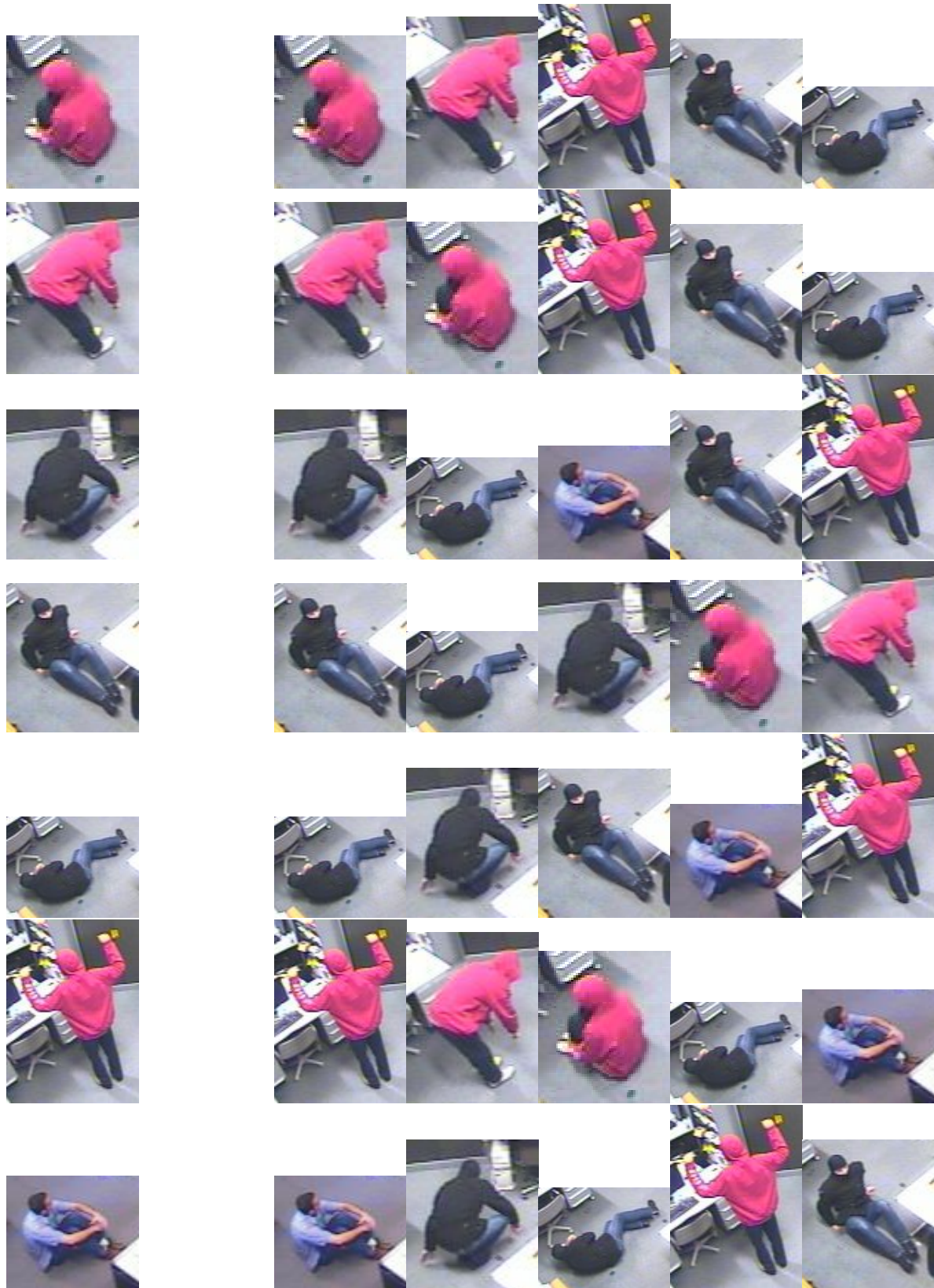


FIG. 9.18: Résultat de la classification : pour chaque exemple à gauche, on ordonne à droite les résultats par ordre de pertinence.

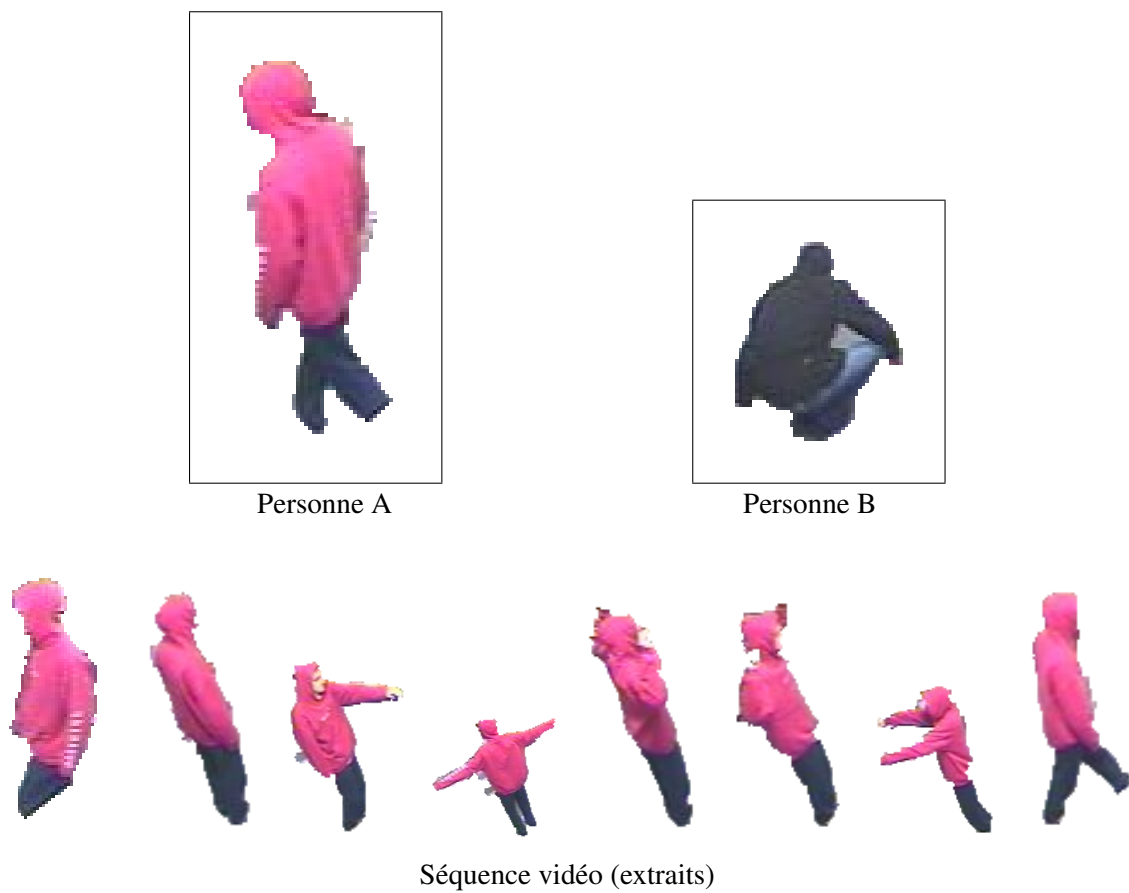


FIG. 9.19: Les deux candidats que l'on cherche à détecter, ainsi que quelques extraits de la séquence vidéo.

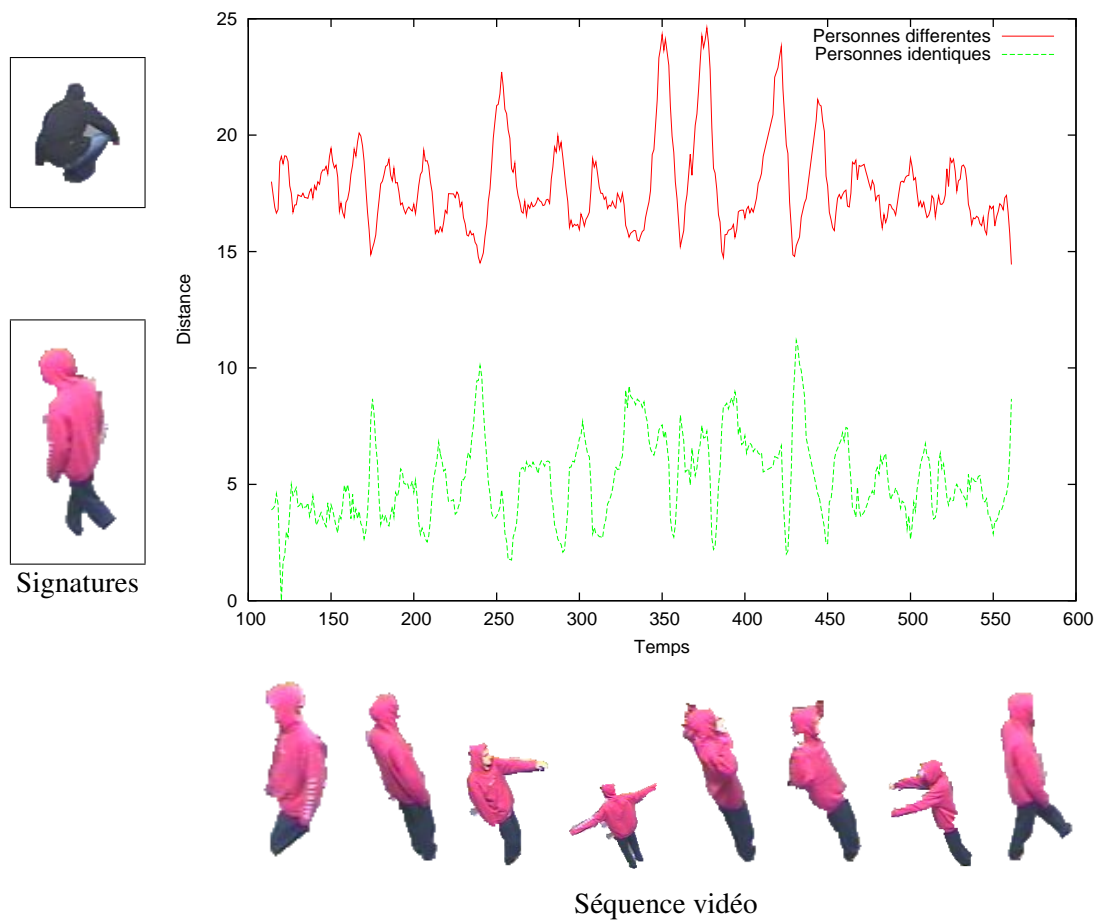


FIG. 9.20: Distance entre chaque candidat et les images de la séquence par comparaison des histogrammes couleur. On voit clairement que l'algorithme sépare bien la bonne personne de la mauvaise.

personnage rouge et noir est globalement assez sombre, et d'autre part les résultats sont portés par un bon appariement du pantalon noir avec la silhouette complètement noire.

On voit dans ce cas d'utilisation que notre choix d'un algorithme simple cherchant à appairer le plus possible s'avère moins bon que dans les essais de type indexation sur les images de COIL. Nos travaux futurs sur cette approche viseront à enrichir les descripteurs des régions afin de mieux utiliser l'information à notre disposition. Par exemple en tenant compte des surfaces relatives, ou en pondérant par une fonction de la variance interne de chaque région. Un descripteur que nous nous interdirons dans le cas PACA est la position relative des régions. En effet, la motivation du projet est de détecter les chutes, qui sont notamment caractérisées par un changement de position relative des éléments constitutifs de la personne, principalement selon l'axe vertical. Nous pouvons difficilement envisager un système qui cesserait de fonctionner dans les situations mêmes qui justifient son existence ! Ces améliorations, ainsi que des améliorations des performances, seront l'objet de travaux futurs dans le cadre de ce projet.

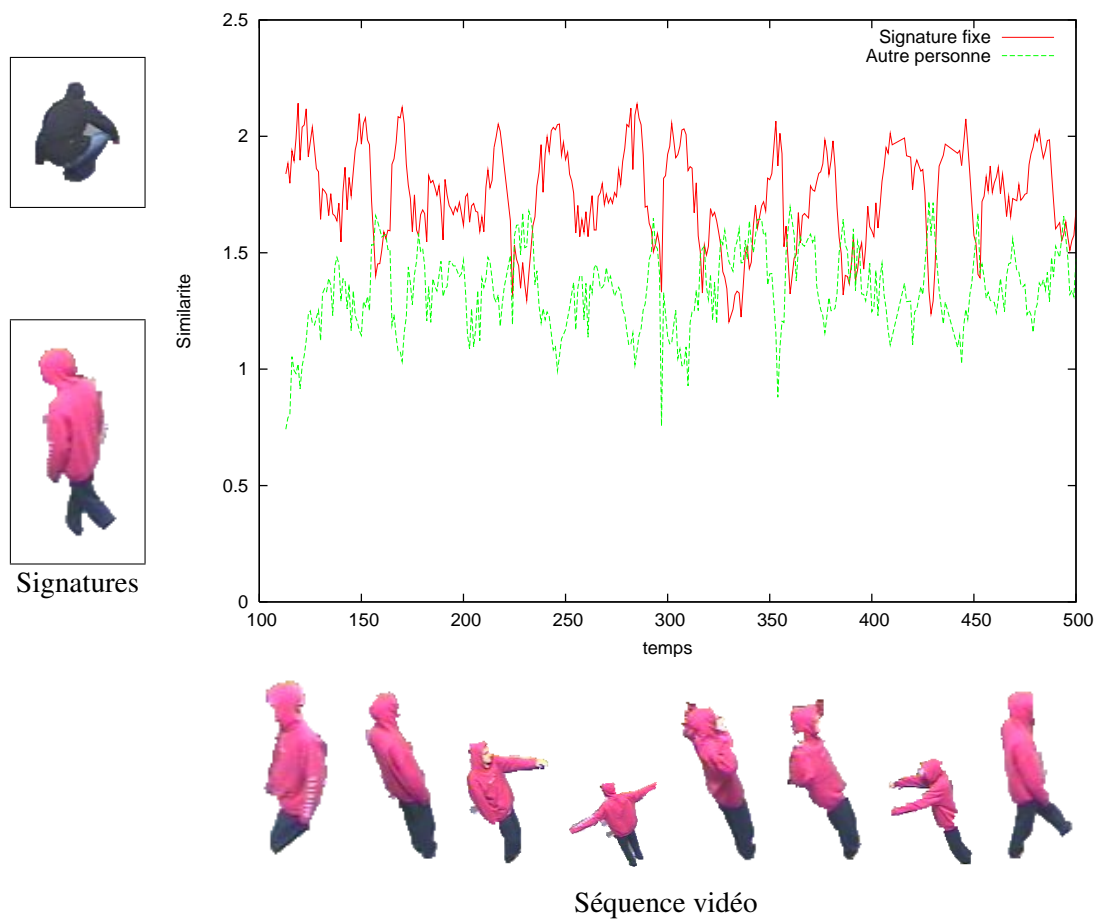


FIG. 9.21: Distance entre chaque candidat et les images de la séquences par comparaison d'arbres. On voit que les signatures sont globalement bien séparées, mais que certaines images causeront des erreurs de classification.

9.4 Conclusion

Nous avons présenté ici une chaîne de traitement permettant de détecter des silhouettes dans des séquences vidéo. Notre approche se base sur l'utilisation d'informations issues de plusieurs sources en la focalisant en deux endroits : d'une part le gradient et d'autre part les marqueurs. Cette fusion d'informations offre une certaine redondance au système et lui donne une bonne robustesse. L'utilisation de l'information temporelle pour ajuster le gradient s'est avérée extrêmement efficace.

Nous avons décomposé cette chaîne en plusieurs modules, dont certains sont optionnels, afin de fournir une grande souplesse dans le compromis entre temps de calcul et qualité du résultat. En effet, la segmentation obtenue n'est qu'une étape dans un système de traitement complet. Il est ainsi parfaitement possible de choisir une version rapide ayant un taux de reconnaissance plus faible qu'une version lente mais parfaite. Dans ce cas, ce sont les systèmes situés en aval de la segmentation qui devront absorber et corriger l'erreur.

Par la suite, anticipant les besoins du projet PS26-27, nous avons étudié l'utilisation de nos algorithmes de modélisation pour l'aide à la détection, l'indexation et la structuration des séquences. Le point critique étant la détection et la segmentation de la personne, nous avons obtenu des résultats nous encourageant à poursuivre dans cette voie. En effet, il nous semble que la capacité à identifier les personnes présentes dans différentes scènes sera essentielle au projet. Les études futures se concentreront sur des scénarios plus réalistes, dans lesquels nos modélisations complexes s'avèreront plus efficaces que les approches simples qui suffisent à l'heure actuelle.

Il est à noter que pour leur dernière génération d'imageurs, ST Microelectronics a résolu beaucoup des problèmes que nous avons relevés (bruit, dérive des couleurs, etc). Cette amélioration de la qualité des images, y compris en lumière faible, nous permet d'envisager en retour une simplification de la chaîne de traitement et donc des performances brutes encore meilleures. Cependant, il faut considérer qu'en pratique, dans de nombreux contextes d'utilisations en vidéosurveillance les images sont (ou deviennent) médiocres pour des raisons physiques difficilement contrôlables : poussières sur les objectifs, maintenance, vieillissement du matériel, compression des images, etc. Notre attitude prudente et critique vis-à-vis des images reste donc parfaitement valide.

Enfin, au niveau scientifique, il est important de reconnaître le stimulus intellectuel majeur qu'ont représenté ces images difficiles, y compris dans le développement d'algorithmes qui n'ont pas été utilisés directement dans le cadre de ce projet. En effet, beaucoup d'améliorations potentielles de nos méthodes de traitement ont été jaugées à l'aune des images «PACA», qui ont souvent forcé l'humilité.

Cinquième partie

Conclusions et Annexes

10 Conclusion	195
10.1 Apports de cette thèse	195
10.2 Perspectives	197
A Amibes morphologiques : implémentation avec Morphée	199
A.1 Introduction	199
A.2 Spécificités	199
A.3 Adaptabilité	200
A.4 Voisinages imbriqués	200
A.5 Conclusion	203
Bibliographie	205

Chapitre 10

Conclusion

Il ne faut s'occuper que des objets dont notre esprit paraît capable d'acquérir une connaissance certaine et indubitable.

René Descartes,
Règles pour la direction de l'esprit

10.1 Apports de cette thèse

Notre objectif principal consistait à étendre nos connaissances sur l'utilisation des modélisations par graphes d'images et de séquences vidéo, déterminer leurs forces et leurs faiblesses, identifier les domaines d'application privilégiés. Pour cela, nous avons balayé un large spectre de méthodes que nous rappellerons ici.

10.1.1 Première partie : extension des techniques initiales

Nous avons commencé par étudier le domaine de validité de l'approche conçue par Cristina Gomila et ses possibilités d'extension à d'autres types d'images que les flux vidéos. Nous avons proposé l'utilisation d'un nouvel algorithme d'appariement, le «graduated assignment» qui nous a donné des résultats légèrement meilleurs que la relaxation probabiliste précédemment utilisée. Cependant, nous sommes restés sensibles dans cette approche à la qualité des segmentations.

Pour pallier ce problème, nous avons expérimenté une technique d'appariement spectral «many-to-many» visant à réduire notre dépendance à des modélisations très voisines, c'est-à-dire à des segmentations de bonne qualité. Cette technique ne nous ayant pas convaincu, nous avons décidé d'une part de continuer à réduire les variations entre les segmentations d'images voisines, et d'autre part de concevoir une approche radicalement différente de la modélisation qui serait moins tributaire de processus instables.

10.1.2 Deuxième partie : filtrage

Le premier pan de cette nouvelle approche était focalisé sur le traitement des images, étape préalable à la modélisation. Nous y avons proposé deux types de filtrages ayant pour but final de séparer l'information pertinente du bruit et des perturbations, afin de nous fournir de meilleures segmentations.

Le premier développement, et le plus riche d'un point de vue théorique, a été les amibes morphologiques. Ce nouvel outil nous a permis de développer des filtres semblables aux diffusions anisotropes et d'une grande efficacité. Nous avons ainsi testé des filtres débruiteurs sur des images en niveaux de gris et en couleur, 2D et 3D, et proposé plusieurs variantes pour obtenir le meilleur lissage tout en respectant au mieux les contours de l'image. Les segmentations de ces images lissées étant plus robustes face au bruit, nous avons donc pu étendre de façon remarquable les algorithmes de la première partie à des images plus difficiles. Nos travaux sur les amibes morphologiques ont donné lieu à plusieurs publications : [Lerallut et al., 2005b, Lerallut et al., 2005a, Lerallut et al., 2006].

Bien que les résultats des amibes soient impressionnants, leur temps de calcul important nous a conduit à chercher une autre solution pour les applications à forte contrainte de temps. Nous nous sommes donc intéressés aux améliorations que nous pouvions apporter aux gradients, l'étape suivant le filtrage dans la chaîne de traitement. Nous avons étudié plusieurs approches et conçu un gradient rapide et bien adapté aux images très bruitées.

10.1.3 Troisième partie : modélisation orientée-objet

Les filtrages et les gradients ont permis des améliorations sensibles de nos résultats. Cependant, nous avons vu que la modélisation de scènes complètes était difficilement généralisable au-delà des domaines déjà explorés par Cristina Gomila, et ce quelque soit la qualité du filtrage ou même de la segmentation.

Le second pan de notre nouvelle approche a donc porté sur la conception de techniques de modélisation des objets eux-mêmes. En effet, les études de la première partie nous ont montré que dans les cas plus complexes que la vidéoconférence, les informations doivent être hiérarchisées entre celles concernant l'intérieur des objets et celles décrivant les relations externes entre les objets.

Dans une première étape nous avons proposé une modification de la Ligne de Partage des Eaux, en lui adjoignant une partition de référence. Nous l'avons utilisée pour segmenter des espaces d'échelle morphologiques et obtenir des représentations hiérarchiques robustes des objets détectés dans une image. Cette approche est par construction facile à intégrer dans un système de détection d'objets et peut profiter de cette détection pour construire une signature de chaque objet. Nous avons également modifié cette approche en espace d'échelles morphologiques pour construire un gradient aux propriétés intéressantes, notamment pour la segmentation d'objets texturés.

La seconde étape a consisté à comparer ces modèles. Nous avons proposé une méthode inspirée des méthodes à noyaux qui s'est avérée rapide et efficace. En l'ajoutant à notre modélisation, nous avons obtenu un ensemble d'une grande robustesse dans les tests que nous avons effectués.

10.1.4 Quatrième partie : résultats et applications

Dans cette dernière partie, nous nous sommes recentrés sur des problématiques réelles en cherchant les forces et les faiblesses des approches que nous avons développé dans les deux parties précédentes. Nous avons procédé en deux phases.

D'abord, nous avons cherché à synthétiser les différentes approches que nous avons développées au cours de cette thèse. Nous avons comparé les différents filtrages, en nous intéressant également aux temps de calculs. Nous avons illustré les différences entre nos diverses approches, et leur position dans le compromis entre qualité de résultats et vitesse d'exécution.

Ensuite, nous avons réalisé un projet applicatif se plaçant en complément des modélisations orientées-objets de la troisième partie, comprenant notamment le projet «PACA» PS26-27. Nous

avons développé un système de segmentation de silhouettes rapide et efficace, et utilisé dans le prototype commun INRIA/CMM. Nous avons utilisé des techniques de fusion de l'information issue de plusieurs représentations afin d'améliorer la robustesse. L'utilisation de l'information temporelle, en particulier, s'est avérée extrêmement efficace. Ce développement s'est inséré naturellement dans le cadre plus large d'une chaîne complète de traitement de séquences d'images de type vidéosurveillance.

Au cours de ces deux chapitres nous avons beaucoup appris sur les outils que nous avons développés. Confrontés à des images parfois difficiles, certains ont donné d'excellents résultats, d'autres ont été plus décevants. Dans tous les cas notre connaissance de ces algorithmes, de leurs qualités et de leurs limites, s'est grandement enrichie.

De façon plus globale, les imageurs d'entrée de gamme spécifiés par le projet PACA nous ont fourni un stimulus intellectuel constant visant à nous interroger sur la robustesse de nos modélisations face au bruit dans le cadre de problèmes industriels réels. Ils nous ont forcés à reconsidérer nos algorithmes et nos méthodes en écartant certaines approches théoriquement viables mais insuffisamment robustes. Cet aller-retour entre théorie et pratique, images de tests et images difficiles, s'est avéré extrêmement motivant dans nos travaux.

10.2 Perspectives

Moyen terme

Du point de vue technique, les deux axes de développements principaux que l'on peut envisager suite à ce travail sont les filtrages, en particulier les amibes morphologiques, et les modélisations orientées objets.

L'amélioration la plus importante que l'on pourrait apporter aux amibes morphologiques serait une accélération des performances. Bien que celles-ci soient bonnes en comparaison d'autres approches anisotropes, il existe probablement une formulation plus efficace. Peut-être via un parallélisme massif, ou des implémentations matérielles, ou encore par des approximations. De plus, l'essentiel des travaux réalisés jusqu'ici sur les amibes ont été focalisés sur la réduction du bruit. Des expériences ont été réalisées et sont à continuer pour construire des opérateurs géodésiques, des inondations, ainsi que sur des amibes aux comportements particuliers (d'une taille minimum fixée, ou au contraire moins flexibles et ne «passant pas» à travers certains passages étroits).

Sur un autre plan, la modélisation orientée-objet est probablement une manière très efficace d'encoder la structure d'un objet. En premier lieu parce que la détection de l'objet, c'est-à-dire la segmentation initiale, reste du ressort du traitement d'images, et donc reste proche de l'information originelle. Dans un second temps, les inondations contrôlées et les méthodes que nous avons utilisées pour en construire des arbres sont encore assez simples. Il y a probablement un vaste domaine à étudier, par exemple sur les critères de fusion à un niveau donné de la pyramide. De plus, des optimisations en vitesse du processus de calcul des espaces d'échelle seraient bénéfique non seulement en vue de cette modélisation, mais aussi en permettant d'accélérer le calcul du gradient pyramide.

Enfin, la distance sur les arbres que nous avons proposée n'est qu'un exemple parmi d'autres. Elle a l'avantage d'être simple, rapide et compréhensible, mais on peut parfaitement envisager des méthodes plus subtiles utilisant mieux l'information disponible, ou étant plus robustes face à des niveaux parasites dans la modélisation, etc. Dans une application moins contrainte que le projet «PACA», une amélioration intéressante serait d'intégrer une notion de voisinage horizontal entre les nœuds «frères», par exemple en incluant des relations floues comme «à gauche», «à droite», etc.

Long terme

Nous avons eu tout au long de ce travail une attitude très critique envers les différentes techniques de segmentation que nous avons utilisées et parfois conçues ou améliorées. Les résultats de nos techniques de modélisation dépendaient en effet fortement de la qualité de ces segmentations. Il reste certainement de nombreuses améliorations possibles pour améliorer nos outils de segmentation, les rendre plus robustes, plus fiables, etc. Mais il nous semble que malgré les meilleurs efforts passés ou encore à venir dans ce domaine, toute approche découplant modélisation et segmentation restera globalement infructueuse.

Les travaux futurs dans ce domaine de l'interprétation automatique devront très probablement effectuer des allers et retours entre segmentation et modélisation. On peut ainsi imaginer trois axes de recherche :

Le premier serait la modélisation/segmentation conjointe de deux images. Soit prises à des moments différents, soit selon des angles différents. L'idée serait de tester sur une image une hypothèse venant de l'autre image. *«Ce pied appartient-il au personnage ? Probablement, car dans l'autre image la chaussure ressemble bien à celle-ci.»*

La seconde approche serait de disposer d'un modèle fixé pour les objets et de chercher à calquer la réalité sur le modèle. Une personne serait décrite comme un pantin articulé, et on chercherait où sont placés les membres, la couleur des vêtements portés, etc. Ces travaux sont ébauchés dans le projet «PACA» avec la détection des postures.

Le troisième axe qui nous semble intéressant serait la détermination automatique de modèles à partir d'une base d'exemples. Le principe serait de disposer de modélisations d'objets plus ou moins ajustées à la main et d'en déduire un modèle permettant de décrire ou de classifier les nouveaux objets. Les applications les plus immédiates se trouveraient dans le domaine de l'indexation et de la recherche d'images par le contenu. Les groupes d'images similaires seraient ainsi représentés par des archétypes déterminés automatiquement à partir des images elles-mêmes.

Ces sujets sont largement ouverts, mais ils nous semblent très prometteurs car ils visent à améliorer le cœur du problème. Tous les solveurs sont équivalents quand la modélisation est bonne.

Annexe A

Amibes morphologiques : implémentation générique avec Morphée

*If it were done when 'tis done, then
'twere well it were done quickly*

William Shakespeare,
Macbeth I.7

A.1 Introduction

Dans le chapitre 4 nous avons défini les amibes morphologiques, un type d'éléments structurants adaptatifs. Nous avons également présenté les résultats obtenus pour divers types de filtres, sur des images couleur aussi bien que niveaux de gris.

Ce travail a été rendu possible par la bibliothèque de traitement d'images Morphée que j'ai développée en collaboration avec Raffi Enfciaud, également en thèse au CMM. Nous avons réalisé cette plateforme pour nous permettre de concevoir et de tester *rapidement* des algorithmes nouveaux en maximisant la réutilisation des composants. La combinaison des langages C++ et Python nous a permis de développer à plusieurs niveaux en offrant une fine granularité dans le compromis entre temps de développement et temps d'exécution. Morphée est depuis utilisé comme outil principal de recherche et de développement dans la majorité des projets actuels du CMM.

Morphée traite des images 2D, 3D ou n-D, en niveaux de gris, couleur ou multispectrales et de tailles variant de quelques octets à plusieurs giga-octets. Il a été porté et utilisé avec succès sur plusieurs plateformes, en particulier Microsoft Windows, Linux et Mac OSX, aussi bien sur des architectures 32 que 64 bits.

A.2 Spécificités

Sans entrer dans les détails techniques, ce qui fait la spécificité de Morphée par rapport à des outils plus conventionnels est son organisation en briques élémentaires correspondant par des interfaces standardisées.

Ainsi, un élément structurant aura une entrée unique qui est la position à laquelle on le centre dans l'image et une sortie unique qui est la liste des voisins du points sur lequel on l'a centré. Cette liste

est fixe (modulo les effets de bord ou les trames exotiques) pour les éléments structurants usuels, mais dans le cas des amibes, elle sera calculée en chaque point.

La figure A.1 présente le schéma d'un algorithme générique pour les opérations sur voisinage. L'algorithme examine le voisinage en chaque point, effectue une mesure sur ce voisinage et écrit cette valeur dans l'image de sortie. Avec l'opérateur «max», on obtient une dilatation, avec l'opérateur «min», une érosion.

Une fois ces opérateurs définis, ils fonctionneront avec tout élément structurant se conformant à l'interface standard. Pour implémenter les amibes, il a donc suffi de suivre cette interface pour que *tous les algorithmes* deviennent *instantanément* disponibles avec les amibes¹.

Initialement, nous hésitions entre deux approches pour les amibes. Morphée nous a permis de comparer les deux prototypes sur de nombreux algorithmes (érosion, dilatation, moyenne, médiane, filtres alternés séquentiels, nivellements, etc) après seulement une journée de développement et une de tests. Nous avons donc pu éliminer rapidement le moins efficace et nous concentrer sur l'amélioration du meilleur prototype.

A.3 Adaptabilité

La formulation de la «distance d'amibe» utilise un écart e entre les valeurs des pixels de l'image pilote. Cet écart est lui aussi un composant interchangeable. Par exemple, on utilisera la valeur absolue des différences en niveaux de gris, une distance euclidienne dans l'espace couleur LAB, etc. On peut même définir une distance par rapport au centre de l'amibe, ce qui permet de retrouver les schémas présentés par Braga-Neto ou Debayle.

La figure A.2 décrit la procédure *CalculVoisinage* pour les amibes ainsi que deux distances possibles. L'une est la distance d'amibe classique, l'autre est la distance de Braga-Neto. Dans les deux cas, la norme utilisée entre les pixels de l'image (notée simplement $\| \cdot \|$) est elle-aussi modulaire. On peut par exemple choisir une distance couleur LAB ou HLS, ajouter une anamorphose, etc.

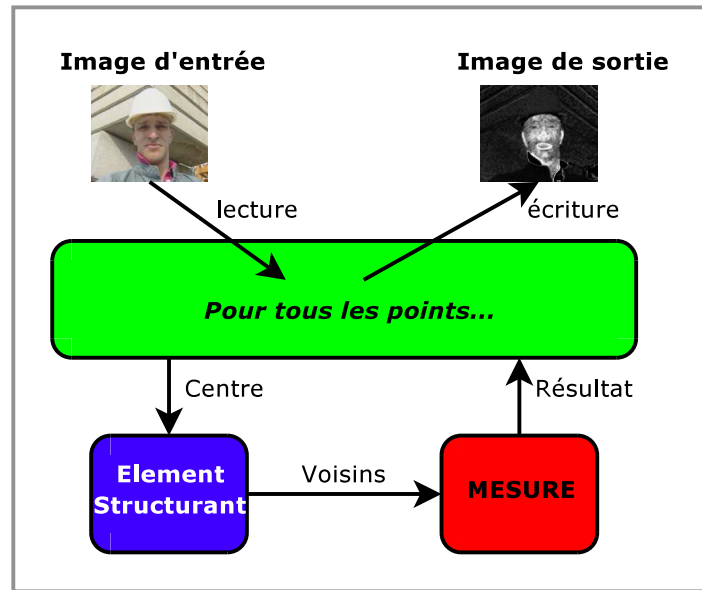
A.4 Voisinages imbriqués

Un des aspects les plus séduisants de cette formulation réside dans la possibilité de chaîner les éléments structurants. En effet, la formulation de l'amibe utilise une notion de voisinage (*pour tout voisin v_q de q , faire...*). Ce voisinage est en général défini par un élément structurant fixe classique, mais peut également être défini autrement.

Par exemple, par un élément structurant suivant le mouvement comme défini par Nicolas Laveau [Laveau, 2005]. Un élément structurant suivant le mouvement (ESM) permet de traiter un flux d'images 2D comme un volume 3D (2D+t en réalité) mais dont la connexité selon l'axe z suivrait le flot optique : le pixel de position (x, y, t) à l'instant t est voisin du pixel $(x + \Delta x, y + \Delta y, t + 1)$ à l'instant $t + 1$, pour un champ de mouvement $(\Delta x, \Delta y)$. Ceci permet de construire des opérations morphologiques 3D efficaces sans avoir besoin de recalculer les images avant de construire le volume 3D.

L'extension des amibes au domaine de la vidéo via la formulation en 2D+t est extrêmement simple avec Morphée, puisqu'il suffit de préciser qu'un ESM sera utilisé pour définir le voisinage de base.

¹Vrai pour le C++. L'utilisation via Python nécessite un peu de travail supplémentaire du type «copier-coller».

**Boucle principale**

ENTRÉES: image $imIn$, élément structurant SE , et mesure $OpérateurMesure$

SORTIES: image $imOut$

pour tout pixels p **faire**

voisins = **CalculVoisinage**($SE, imIn, p$)

$imOut(p)$ = **OpérateurMesure**($imIn, voisins$)

fin pour

CalculVoisinage : voisinage 4-connexe

ENTRÉES: image im , position p

SORTIES: voisins de p

deltas = $\{(-1, 0), (0, 0), (1, 0), (0, -1), (0, 1)\}$

voisins = \emptyset

pour tout δ dans deltas **faire**

si $p + \delta \in im$ **alors**

ajouter $p + \delta$ à voisins

finsi

fin pour

retourner voisins

Opérateur max

ENTRÉES: image im , ensemble de positions $voisins$

SORTIES: valeur maximale de im sur les positions définies dans $voisins$

valMax = $\max_{x \in voisins} (im(x))$

retourner valMax

FIG. A.1: Dilatation utilisant un algorithme générique pour des opérations sur voisinage (érosion, dilatation, etc) utilisé dans Morphée. L'élément structurant reçoit sa position et renvoie la liste des voisins. L'opérateur de mesure calcule la valeur de retour sur les valeurs de ces voisins. L'algorithme générique boucle sur toute l'image et orchestre les deux composants. En combinant les briques détaillées ci-dessus on obtient la dilatation par un élément structurant 4-connexe.

CalculVoisinage pour une amibe de taille *seuil***ENTRÉES:** image *imIn*, position *centre***SORTIES:** voisins de *p* au sens de l'amibeinitialiser FAH, *voisins* = {*centre*}**tantque** FAH non vide **faire** sortir *q* de la FAH, avec la distance *d_q* ajouter *q* à *voisins* **pour tout** *v_q* voisin de *q* **faire** **si** **Distance** (*im*, *d_q*, *centre*, *q*, *v_q*) ≤ *seuil* **alors** ajouter *v_q* à la FAH avec la priorité *Distance(im, d_q, centre, q, v_q)* **fin** **fin pour****fin tantque****return** *voisins***Distance d'amibe standard de paramètre λ****ENTRÉES:** image *im*, distance *d_q* et positions *centre*, *q*, *v_q***SORTIES:** distance d'amibe entre *centre* et *v_q***return** $d_q + (1 + \lambda \|im(q) - im(v_q)\|)$ **Distance d'amibe pour amibes de type Braga-Neto****ENTRÉES:** image *im*, distance *d_q* et positions *centre*, *q*, *v_q***SORTIES:** distance de pixels entre *centre* et *v_q***return** $\|im(centre) - im(v_q)\|$

FIG. A.2: Calcul de la forme de l'amibe en un point *p*. On utilise une file d'attente hiérarchique (FAH) pour effectuer une croissance de région autour du centre. La première distance proposée est la distance d'amibe décrite dans cette thèse. La seconde distance est celle utilisée par Braga-Neto. Il suffit de choisir la distance voulue pour tester l'une ou l'autre des amibes sur la totalité des algorithmes.

A.5 Conclusion

Morphée est un outil de développement orienté recherche dont l'objectif est de permettre aux chercheurs de répondre rapidement à la question «*Que se passerait-il si on faisait . . . ?*». L'implémentation et les tests des amibes en quelques jours a été le premier exemple de la validité de cette approche.

La difficulté principale à laquelle nous avons dû faire face est de trouver l'équilibre entre rapidité de prototypage, rapidité d'exécution et rapidité d'apprentissage. En effet, cet outil est prévu pour permettre la création rapide d'algorithmes nouveaux (prototypage), mais il est également utilisé dans des projets applicatifs et à visée industrielle (execution). Et c'est également un outil que les nouveaux thésards du CMM doivent pouvoir maîtriser rapidement (apprentissage).

On ne manquera pas de trouver un parallèle intéressant entre cette recherche d'équilibre dans un logiciel et les contraintes similaires (simplicité, robustesse, rapidité) que nous avons dû gérer dans nos approches sur les graphes.

Bibliographie

- [Angulo, 2003] Angulo, J. (2003). *Morphologie mathématique et indexation d'images couleur. Application à la microscopie en biomédecine*. Thèse de doctorat en morphologie mathématique, ENSMP.
- [Angulo and Serra, 2002] Angulo, J. and Serra, J. (2002). Morphological color size distributions for image classification and retrieval. In *Proceedings of ACIVS 2002 (Advanced Concepts for Intelligent Vision Systems)*, Ghent, Belgium.
- [Barash, 2000] Barash, D. (2000). Bilateral filtering and anisotropic diffusion. Technical report.
- [Bergen, 1999] Bergen, L. (1999). *Analyse de l' Ordre de Profondeur des Objets dans une Scène par l' Étude du Mouvement*. Thèse de doctorat en morphologie mathématique, ENSMP.
- [Beucher, 1990] Beucher, S. (1990). *Segmentation d' images et morphologie mathématique*. Thèse de doctorat en morphologie mathématique, ENSMP. 1822.
- [Beucher, 1991] Beucher, S. (1991). The watershed transformation applied to image segmentation. In *10th Pfeifferkorn Conf. on Signal and Image Processing in Microscopy and Microanalysis*, Cambridge, UK.
- [Beucher, 2004] Beucher, S. (2004). Algorithmes sans biais de ligne de partage des eaux. Technical report.
- [Beucher and Lantuéjoul, 1979] Beucher, S. and Lantuéjoul, C. (1979). Use of watersheds in contour detection. In *International Workshop on Image Processing : Real-time edge and motion/detection/estimation. - CCETT - INSA - IRISA,*, pages 2.1–2.12, Rennes. 609.
- [Boehm, 2004] Boehm, M. (2004). *Contribution à l'amélioration du rendu volumique de données médicales 3D*. PhD thesis, Centre de Morphologie Mathématique - École Nationale Supérieure des Mines de Paris.
- [Boulay et al., 2005] Boulay, B., Brémond, F., and Thonnat, M. (2005). Posture recognition with a 3d human model. In *IEE International Symposium on Imaging for Crime Detection and Prevention*.
- [Boulay et al., 2006] Boulay, B., Brémond, F., and Thonnat, M. (2006). Applying 3d human model in a posture recognition system. *Pattern Recognition Letter, Special Issue on vision for Crime Detection and Prevention*. In Press.
- [Boykov and Kolmogorov, 2001] Boykov, Y. and Kolmogorov, V. (2001). Computing geodesics and minimal surfaces via graph cuts. In *IEEE International Conference on Computer Vision (ICCV'01)*, volume 1, pages 26–33, Nice, France.
- [Braga-Neto, 1996] Braga-Neto, U. M. (1996). Alternating sequential filters by adaptive-neighborhood structuring functions. In *Proceedings of International Symposium on Mathematical Morphology ISMM'96*, pages 139–146.

- [Bridle, 1990] Bridle, J. S. (1990). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in Neural Information Processing Systems*, (2) :211–217.
- [Brodnik et al., 2001] Brodnik, A., Carlsson, S., Karlsson, J., and Munro, J. I. (2001). Worst case constant time priority queue. In *Symposium on Discrete Algorithms*, pages 523–528.
- [Caelli and Kosinov, 2004] Caelli, T. and Kosinov, S. (2004). An eigenspace projection clustering method for inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4).
- [Catté, 1992] Catté, F. i. e. a. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numerical Analysis*.
- [Cavallaro et al., 2004] Cavallaro, A., Salvador, E., and Ebrahimi, T. (2004). Detecting shadows in image sequences. In *Visual Media Production, 2004. (CVMP). 1st European Conference on*, pages 165–174.
- [Cheng, 1995] Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8) :790–799.
- [Cherkassky and A. V. Goldberg, 1996] Cherkassky, B. V. and A. V. Goldberg, A. V. (1996). Heap-on-top priority queues. Technical Report TR 96-042, NEC Research Institute, Princeton, NJ.
- [Cohen, 1991] Cohen, L. D. (1991). On active contour models and balloons. *Computer Vision, Graphics and Image Processing : Image Understanding*, 2(53) :211–218.
- [Collins and Duffy, 2001] Collins, M. and Duffy, N. (2001). Convolution kernels for natural language. In *Proc. of Neural Information Processing Systems*.
- [Comaniciu and Meer, 1997] Comaniciu, D. and Meer, P. (1997). Robust analysis of feature spaces : Color image segmentation. In *IEEE Conf. on Comp. Vis. and Pattern Recognition*, pages 750–755, Puerto Rico.
- [Debayle and Pinoli, 2005] Debayle, J. and Pinoli, J. C. (2005). Multiscale image filtering and segmentation by means of adaptive neighborhood mathematical morphology. In *Proceedings of IEEE International Conference on Image Processing*, Genoa, Italy.
- [Decenci re et al., 1999] Decenci re, E., de Fouquet, C., and Meyer, F. (1999). Applications of kriging to image sequence coding. *Signal Processing : Image Communication*, (13) :227–249.
- [Demarty, 2000] Demarty, C.-H. (2000). *Segmentation et structuration d’un document vid o pour la caract risation et l’indexation de son contenu s mantique : Application aux journaux t l vis s*. Th se de doctorat en morphologie math matique, ENSMP.
- [Demirci et al., 2003] Demirci, F., Shokoufandeh, A., Keselman, Y., Dickinson, S., and Bretzner, L. (2003). Many-to-many matching of scale-space feature hierarchies using metric embedding. In *Scale Space Methods in Computer Vision, 4th International Conference*, pages 17–32, Isle of Skye, UK.
- [Deriche, 1987] Deriche, R. (1987). Using canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2) :167–187.
- [d’Ornellas, 2000] d’Ornellas, M. C. (2000). A multi-scale gradient approach for color-based morphological segmentation. volume 3, pages 363–366.
- [Evans, 2004] Evans, A. N. (2004). Morphological gradient operators for colour images. In *Image Processing, 2004. ICIP ’04. 2004 International Conference on*, volume 5, pages 3089–3092.

- [Fukunaga and Hostetler, 1975] Fukunaga, K. and Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21 :32–40.
- [Gold and Rangarajan, 1996] Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4).
- [Goldberg and Kennedy, 1995] Goldberg, A. V. and Kennedy, R. (1995). An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming : Series A and B*, 71(2) :153–177.
- [Gomila, 2001] Gomila, C. (2001). *Mise en correspondance de partitions en vue du suivi d'objets*. Thèse de doctorat en morphologie mathématique, ENSMP.
- [Gondran and Minoux, 1985] Gondran, M. and Minoux, M. (1985). *Graphes et Algorithmes*. Collection de la Direction des Études et Recherches d'Électricité de France.
- [Grimaud, 1990] Grimaud, M. (1990). *géodésie numérique en morphologie mathématique, La*. Thèse de doctorat en morphologie mathématique, ENSMP.
- [Guigues, 2003] Guigues, L. (2003). *Modèles Multi-Echelles pour la Segmentation d'Images*. PhD thesis, Université de Cergy-Pontoise.
- [Heijmans, 1994] Heijmans, H. (1994). *Morphological Image Operators*, volume 24 of *Advances in Electronics and Electron Physics*. Academic Press, Boston.
- [Huchon, 2005] Huchon, J.-P. (2005). Rapport sécurité, lycées, poste de police, prévention, transports sixième rapport pour l'année 2005. Technical report, Commission Permanente du Conseil Régional d'Île de France.
- [Hummel and Zucker, 1981] Hummel, R. A. and Zucker, S. W. (1981). On the foundation of relaxation labeling processes. Technical report, Computer Vision and Graphics Laboratory, Department of Electrical Engineering, McGill University Montréal, Quebec, Canada.
- [Kass et al., 1988] Kass, M., Witkin, A., and D., T. (1988). Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331.
- [Keselman et al., 2003] Keselman, Y., Shokoufandeh, A., Demirci, M., and Dickinson, S. (2003). Many-to-many graph matching via metric embedding. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*, volume 1, pages 850–857.
- [Kimia et al., 2001] Kimia, T. S., Klein, P., and B. (2001). Recognition of shapes by editing shock graphs. In *IEEE International Conference on Computer Vision (ICCV'01)*, volume 1.
- [Kitchen and Rosenfeld, 1979] Kitchen, L. and Rosenfeld, A. (1979). Discrete relaxation for matching relational structures. *IEEE Transactions on System, Man and Cybernetics*, 9(12) :869–974.
- [Koenderink, 1984] Koenderink, J. (1984). The structure of images. *Biological Cybernetics*, (50) :363–370.
- [Kosinov and Caelli, 2002] Kosinov, S. and Caelli, T. (2002). Inexact multisubgraph matching using graph eigenspace and clustering models. In *SSPR/SPR*, pages 133–142.
- [Kuhn, 1955] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, (2) :83–97.
- [Laveau, 2005] Laveau, N. (2005). *Mouvement et vidéo : estimation, compression, et filtrage morphologique*. Thèse de doctorat en morphologie mathématique, ENSMP.

- [Lerallut et al., 2005a] Lerallut, R., Boehm, M., Decenci re, E., and Meyer, F. (2005a). Noise reduction in 3d images using morphological amoebas. In *Proceedings of IEEE International Conference on Image Processing (ICIP'2005)*, Genoa, Italy.
- [Lerallut et al., 2005b] Lerallut, R., Decenci re, E., and Meyer, F. (2005b). Image processing using morphological amoebas. In *Proceedings of the 5th International Symposium on Mathematical Morphology*. Kluwers Academic Publishers.
- [Lerallut et al., 2006] Lerallut, R., Decenci re, E., and Meyer, F. (2006). Image processing using morphological amoebas. *Image and Vision Computing*. in press.
- [Levine and Nazif, 1985] Levine, M. D. and Nazif, A. M. (1985). Dynamic measurement of computer generated image segmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2).
- [Lindeberg, 1994] Lindeberg, T., editor (1994). *Scale-space theory in computer vision*. Kluwer Academic Publ., Boston.
- [Malik et al., 2001] Malik, D. M., Fowlkes, C., Tal, D., and J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision (ICCV)*, volume 2, pages 416–423.
- [Marcotegui et al., 1999] Marcotegui, B., Correia, P., Marqu s, F., Mech, R., Rosa, R., Wollborn, M., and Zanoquera, F. (1999). Vogue : The momusys video object generator with user environment. In *WIAMIS'99 : Workshop on Image Analysis for Multimedia Interactive Systems*, pages 25–28, Berlin, Germany, June 1999.
- [Matheron, 1967] Matheron, G. (1967). * l ments pour une th orie des milieux poreux*. Masson et Cie, Paris.
- [Meyer, 1991] Meyer, F. (1991). Un algorithme optimal de ligne de partage des eaux. In *Actes 8 me Congr s AFCET Reconnaissance des Formes et Intelligence Artificielle*, pages 847–857, Lyon-Villeurbanne.
- [Meyer, 1994] Meyer, F. (1994). Minimum spanning forests for morphological segmentation. In Serra, J. and Soille, P., editors, *Proc. Mathematical Morphology and its Applications to Image Processing*, pages 77–84, Fontainebleau, Septembre 1994. Kluwer Ac. Publ., Nld. 4757.
- [Meyer, 1998] Meyer, F. (1998). The levelings. In *Mathematical Morphology and its Application to Image and Signal Processing, Proc. ISMM*. Kluwers Academic Publishers.
- [Meyer, 2001] Meyer, F. (2001). Hierarchies of partitions and morphological segmentation. In *IEEE Workshop on Scale-Space and Morphology in Computer Vision and ICCV 2001*, Vancouver, Canada.
- [Meyer and Maragos, 1999] Meyer, F. and Maragos, P. (1999). Morphological scale-space representation with levelings. In Mads, N., Johansen, P., Olsen, O. F., and Weickert, J., editors, *Scale-Space Theories in Computer Vision, Proc. Scale-Space'99*, volume 1682 of *Lecture Notes in Computer Science*, pages 187–198, Corfu, Greece, September 1999. Berlin : Springer.
- [Micheli et al., 2005] Micheli, A., Portera, F., and Sperduti, A. (2005). A preliminary empirical comparison of recursive neural networks and tree kernel methods on regression tasks for tree structured domains. *Neurocomputing*, 64 :73–92.
- [Murase et al., 1996] Murase, S. N., Nayar, S., and H. (1996). Columbia object image library : Coil. Technical Report CUCS-006-96, Department of Computer Science, Columbia University.

- [Najman and Schmitt, 1994] Najman, L. and Schmitt, M. (1994). A dynamic hierarchical segmentation algorithm. In *Mathematical Morphology and its applications to Signal Processing, ISMM'04*, pages 13–14.
- [Osher and Sethian, 1988] Osher, S. and Sethian, J. A. (1988). Front propagating with curvature dependent speed. *Journal of Computational Physics*, (79) :12–49.
- [Pecht, 1985] Pecht, J. (1985). Speeding-up successive minkowski operations with bit-plane computers. *Pattern Recognition Letters*, 3(2).
- [Pelillo, 2002] Pelillo, M. (2002). Matching free trees, maximal cliques and monotone game dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11) :1535–1541.
- [Pelillo et al., 1999] Pelillo, M., Siddiqi, K., and Zucker, S. (1999). Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11).
- [Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7).
- [Pictures and Audio, 2002] Pictures, I. J. C. o. M. and Audio (2002). N4668 : Overview of the mpeg-4 standard. <http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [Serra, 1988a] Serra, J. (1988a). *Image analysis and mathematical morphology - Volume 1*, chapter Morphology for Grey-tone Functions, pages 424–478. Academic Press.
- [Serra, 1988b] Serra, J. (1988b). *Image analysis and mathematical morphology - Volume 2 : Theoretical advances*, chapter Mathematical morphology for boolean lattices, pages 37–46. Academic Press.
- [Sharma and Trussell, 1997] Sharma, G. and Trussell, H. J. (1997). Digital color imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(7) :901–932.
- [Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge Texts in Applied Mathematics.
- [Shokoufandeh et al., 1999] Shokoufandeh, A., Marsic, I., and Dickinson, S. J. (1999). View-based object recognition using saliency maps. *Image and Vision Computing*, 17(5/6) :445–460.
- [Siddiqi et al., 1999] Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and W, Z. S. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, (30).
- [Sinkhorn, 1964] Sinkhorn, R. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statis.*, 35 :876–879.
- [Tomasi and Manduchi, 1998] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of IEEE International Conference on Computer Vision, Bombay, India*.
- [Vachier, 1995] Vachier, C. (1995). *Extraction de caractéristiques, segmentation d'images et morphologie mathématique*. Thèse de doctorat en morphologie mathématique, ENSMP.
- [Van Droogenbroeck and Buckley, 2005] Van Droogenbroeck, M. and Buckley, M. (2005). Morphological erosions and openings : fast algorithms based on anchors. *Journal of Mathematical Imaging and Vision, Special Issue on Mathematical Morphology after 40 Years*, 22 :121–142.
- [van Herk, 1992] van Herk, M. (1992). A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13(7) :517–521.

- [Villegas et al., 1999] Villegas, P., Marichal, X., and Salcedo, A. (1999). Objective evaluation of segmentation masks in video sequences. In *WIAMIS'99 : Workshop on Image Analysis for Multimedia Interactive Systems*, pages 85–88, Berlin, Germany, June 1999.
- [Weickert, 1996] Weickert, J. (1996). *Anisotropic diffusion in image processing*. PhD thesis, Universität Kaiserslautern.
- [Weickert, 1997] Weickert, J. (1997). Coherence-enhancing diffusion of colour images. In *Proceedings of the 7th National Symposium on Pattern Recognition and Image Analysis*, volume 1, pages 239–244, Barcelona, Spain.
- [Weickert, 1999] Weickert, J. (1999). Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2/3) :111–127.
- [Witkin, 1983] Witkin, A. P. (1983). Scale-space filtering. In *Proc. 8th Int. Joint Conf. Art. Intell.*, pages 1019–1022, Karlsruhe, Germany.