

Construction et classification d'objets à partir d'images de télédétection par une approche itérative guidée par des connaissances du domaine

THÈSE

présentée et soutenue publiquement le 5 Mai 2009

pour l'obtention du

Doctorat de l'Université de Strasbourg

Spécialité Informatique

par

Sébastien Derivaux

Composition du jury

<i>Rapporteur interne :</i>	Pierre Collet	Professeur des universités	Université de Strasbourg
<i>Rapporteurs externes :</i>	Florence Sèdes Monique Thonnat	Professeur des universités Directeur de recherches	Université de Toulouse III INRIA Sophia Antipolis
<i>Examineur :</i>	Bruno Crémilleux	Professeur des universités	Université de Caen
<i>Directeur :</i>	Jerzy Korczak	Professeur d'université	Université de Strasbourg
<i>Invités :</i>	Anne Puissant Cédric Wemmert Sébastien Lefèvre	Maître de conférences Maître de conférences Maître de conférences	Université de Strasbourg Université de Strasbourg Université de Strasbourg

Mis en page avec la classe thloria.

Table des matières

Remerciements

Chapitre 1

Introduction

1.1	Classification d'image et ses limites	1
1.2	Approche enrichie avec des connaissances	2
1.2.1	Segmentation	2
1.2.2	Classification	3
1.2.3	Détection	3
1.3	Domaines de recherche abordés	4
1.4	Structure du mémoire	5

Chapitre 2

Approche d'intégration de connaissances
--

2.1	Panorama de méthodes	7
2.1.1	Méthodes basées sur des exemples	7
2.1.2	Méthodes basées sur des connaissances du domaine	9
2.1.3	Méthodes hybrides	12
2.1.4	Réflexions sur ce panorama	14
2.2	Schéma général	14
2.2.1	Segmentation	14
2.2.2	Classification	16
2.2.3	Détection	17
2.3	Sources de connaissances	17
2.3.1	Connaissances du domaine	18
2.3.2	Les exemples d'objet comme source de connaissance	22
2.3.3	Déduction de connaissances	23
2.3.4	Induction de connaissances à partir des exemples	24
2.4	Conclusion	27

Chapitre 3**Segmentation supervisée par des exemples**

3.1	Segmentation et évaluation	29
3.1.1	Problématique	29
3.1.2	Critères d'évaluations	30
3.2	Apprentissage de paramètres de segmentation par algorithme évolutionnaire	37
3.2.1	Algorithmes évolutionnaires pour la segmentation	38
3.2.2	État de l'art et positionnement	38
3.2.3	Présentation de l'algorithme	40
3.2.4	Évaluation	41
3.3	Transformation d'espace pour la segmentation	45
3.3.1	État de l'art et positionnement	47
3.3.2	Présentation de l'algorithme	48
3.3.3	Évaluation	50
3.4	Approche hybride	53
3.4.1	Présentation de l'algorithme	53
3.4.2	Évaluation	55
3.5	Conclusion	57

Chapitre 4**Classification : Identification des régions**

4.1	Introduction	61
4.1.1	Problématique	61
4.1.2	Caractérisation des régions	62
4.1.3	Critères d'évaluation	64
4.2	Classification à l'aide d'exemples	65
4.2.1	Problématique du jeu d'exemples d'apprentissage	66
4.2.2	Résultats	66
4.2.3	Conclusion sur l'apprentissage à partir d'exemples	71
4.3	Classification à l'aide de connaissances du domaine	72
4.3.1	Méthodes de classification par connaissances du domaine	72
4.3.2	Algorithme d'identification par connaissances du domaine	73
4.3.3	Évaluation	76
4.4	Fusion de résultats de classifications	81
4.4.1	Comparaison des deux méthodes précédentes	83
4.4.2	Méthodes de fusion de résultats de classifications	84
4.4.3	Algorithme de fusion	84

4.4.4	Résultats	85
4.4.5	Conclusion	86
4.5	Remise en cause de l'étape de segmentation par la classification	88
4.5.1	Présentation de la méthode	88
4.5.2	Expérimentations	89
4.5.3	Conclusion sur la remise en cause de la segmentation	92
4.6	Conclusion sur la classification des régions	92

Chapitre 5

Détection : Extraction des objets d'une classe

5.1	Introduction	95
5.1.1	Interprétation par ensemble de détecteurs	95
5.1.2	Comparaison avec l'approche par segmentation suivie d'une classification	96
5.2	Extraction de détecteurs spécifiques à partir de la base de connaissances	97
5.2.1	Séparation des préoccupations	98
5.2.2	Structure générale des détecteurs	98
5.2.3	Description des filtres pouvant composer un détecteur	99
5.2.4	Unification des résultats de détecteurs	102
5.2.5	Évaluation	103
5.2.6	Conclusion	108
5.3	Collaboration de détecteurs et objets composites	109
5.3.1	Utilisation de détecteurs comme filtres pour d'autres détecteurs	109
5.3.2	Utilisation de détecteurs pour la détection d'objets composites	112
5.3.3	Collaboration plus complexe entre détecteurs	114
5.4	Conclusion	117

Chapitre 6

Application à l'interprétation d'images satellites

6.1	Présentation du jeu de données	119
6.1.1	Description de l'image	119
6.1.2	Description des connaissances du domaine	120
6.1.3	Exemples d'apprentissage	122
6.1.4	Exemples d'évaluation	123
6.2	Application des méthodes d'interprétation	123
6.2.1	Classification basée pixels	124
6.2.2	Segmentation suivie d'une classification basée objets	124
6.2.3	Approche par détecteurs	125

6.3	Comparatif des approches	126
6.4	Robustesse de l'apprentissage	128

Chapitre 7

Conclusion générale

7.1	Contributions	141
7.2	Perspectives	143
7.2.1	Perspectives à court terme	143
7.2.2	Perspectives à long terme	144
7.3	Réflexion sur les limites d'intégration de connaissances	145

Bibliographie	147
----------------------	------------

Annexe A

Méthodes usuelles de segmentation
--

Annexe B

Algorithme évolutionnaire

Annexe C

Méthodes usuelles de classification supervisée

Annexe D

Publications

D.1	Chapitre de livres	169
D.2	Conférences internationales avec comité de lecture	169
D.3	Conférences nationales avec comité de lecture	170

Remerciements

*On est parti dans la vie avec les conseils des parents. Ils
n'ont pas tenu devant l'existence.*

Céline – Guignol's band

Je tiens tout d'abord à remercier mes encadrants, Sébastien Lefèvre et Cédric Wemmert. Ils ont toujours été à même de me soutenir au jour le jour et de m'aider constamment par leurs idées. Sans eux, ma thèse ne ressemblerait aucunement à celle-ci.

Je tiens également à remercier mon directeur de thèse Jerzy Korczak qui a bien accepté de diriger ma thèse, pour ses remarques et pour ses conseils.

Je tiens à remercier Pierre Gańczarski, le mécène, pour avoir fait que je ne manque de rien durant ces longues années. J'espère qu'il comprendra pourquoi je n'ai pas "inversé la problématique".

Je tiens à remercier les membres du LIVE, son émissaire Anne Puissant, mais aussi Christiane Weber et l'énigmatique Aziz (l'homme aux images). Ils m'ont fait comprendre la difficulté du traitement et de l'obtention des images.

Je remercie les membres de mon jury pour le temps qu'ils ont bien voulu prendre à lire ma prose, ce qui n'était pas une mince affaire.

Je tiens à remercier tous les membres de mon équipe de recherche FDBT.

Je remercie mes camarades intermittents de la recherche, Arnaud Quirin, Nicolas Durand, Julien Lesbeurier, Alexandre Blansché, Aurélie Bertaux, Germain Forestier et Jonathan Weber (plus ou moins dans l'ordre d'ancienneté). On s'est beaucoup amusé entre les soirées ciné, jeux, putchs manqués, . . . Je souhaite beaucoup de plaisir à Jonathan comme nouvel admin système.

Je remercie les stagiaires pour leur présence continue. Ils étaient une valeur sûre en cas de besoin.

Je remercie ma cousine, Sarah Derivaux, pour l'intérêt qu'elle a porté à ma thèse et qui continue à me questionner.

Je remercie mes parents qui ont été d'accord avec mes choix de carrière.

Enfin, la dernière et la plus importante, je remercie Cécile Mirabet qui a partagé ma vie durant toutes ces années. Elle a non seulement su me supporter dans cette épreuve, mais aussi me donner la force de continuer dans les moments de doutes. Promis, j'aurai un vrai travail un jour.

Chapitre 1

Introduction

A computer lets you make more mistakes faster than any other invention, with the possible exceptions of handguns and Tequilla.

Mitch Ratcliffe

L'exploitation d'images par des méthodes automatiques est un problème de plus en plus important. On peut, par exemple, citer Google Maps qui contient des images satellites et aériennes de l'intégralité du globe. Pour les grandes villes, un pixel représente une surface au sol de moins d'un mètre carré. De nombreuses applications pourraient être construites, utilisant ces images (mise à jour de cadastres, analyse de l'occupation des sols, ...), mais elles nécessitent des informations de plus haut niveau (objets sémantiques et non pixels). En effet, il existe un fossé sémantique entre les données brutes, disponibles sous forme de pixels, et les données exploitables, qui doivent être exprimées sous forme d'occupation des sols.

Il s'agit d'affecter à chaque objet présent une étiquette représentant une classe d'objets (maison, route, ...). Dans le cas d'images satellites à très haute résolution spatiale, les objets sont représentés par des ensembles de pixels et non plus des pixels isolés. Les images à très haute résolution spatiale sont utiles pour disposer d'une information cohérente, homogène et mise à jour régulièrement. Affecter une étiquette à chacun des pixels indépendamment des autres est une tâche ardue du fait de la non séparation des classes au niveau d'un pixel. En effet, en ne considérant que les caractéristiques d'un pixel pris isolément, plusieurs classes sont plausibles. Les étiquettes de classes ne sont pas appliquées à des pixels isolés, mais des objets c'est-à-dire des composantes connexes de l'image composées de plusieurs pixels. Ces objets ne sont pas connus a priori, et doivent être construits et caractérisés. On utilise alors le terme de classification basée objets.

L'objectif de nos travaux est de traiter cette problématique en proposant une approche de construction et de classification d'objets qui soit guidée par des connaissances. Nous nous intéresserons principalement aux images de télédétection à très haute résolution car cette thèse s'inscrit dans le projet FoDoMuSt¹ qui traite de telles images.

1.1 Classification d'image et ses limites

Dans les cas où une classification basée pixel n'est pas possible à cause de la non séparation des classes au niveau pixel (cas des images satellites à très haute résolution spatiale), la classification

¹<http://fodomust.u-strasbg.fr>

basée objets est usuellement composée de deux étapes : la segmentation puis la classification [22, 7].

La segmentation consiste à regrouper des pixels pour construire des régions, ou objets sans étiquette. L'opération a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Nous verrons que les critères utilisés exploitent la similarité entre les pixels. Deux pixels voisins similaires seront donc regroupés dans la même région.

La classification utilise les objets créés par la segmentation afin de leur affecter une classe. Alors qu'il n'était pas possible de trouver la classe d'un pixel, cela devient possible avec des objets. En effet, un objet peut être décrit de façon plus détaillée qu'un pixel (par sa forme, sa composition, ...), donnant ainsi plus de possibilités pour séparer les classes et donc permettre une meilleure classification.

Cependant, cette approche présente plusieurs limites. La première est l'absence d'utilisation de connaissances dans l'étape de segmentation. De ce fait, l'utilisateur doit guider le processus de segmentation par un jeu d'essais-erreurs sur les paramètres du processus afin d'obtenir un résultat satisfaisant. La seconde limite est le faible couplage entre la segmentation et la classification. La segmentation est obtenue de manière indépendante du processus de classification et n'est jamais remise en cause.

1.2 Vers une approche de construction et classification d'objets enrichie par des connaissances

Les contributions de nos travaux, que nous détaillons plus loin, sont multiples mais guidées par un objectif unique : utiliser au mieux les connaissances disponibles pour améliorer le processus.

Deux types de connaissances sont utilisées dans nos travaux :

- Les connaissances de l'expert sont la première source utilisée. Cette connaissance est implicite et nécessite d'être formalisée, via des méthodes de représentation des connaissances, afin de la rendre opérationnelle.
- Les exemples fournis par l'expert sont aussi considérés comme une source importante de connaissance. Cette connaissance est néanmoins indirecte et nécessite d'être extraite par des méthodes d'extraction de connaissances. De plus, cette extraction peut-être contrôlée via des connaissances de l'expert sur les connaissances qui peuvent être extraites des exemples.

Ces différentes sources de connaissances sont donc hétérogènes et nécessitent des traitements afin d'être exploitées.

Les contributions de nos travaux sont classées selon trois tâches :

- la segmentation qui consiste à agréger des pixels connexes afin de construire des objets qui idéalement doivent correspondre aux objets d'intérêt de l'image ;
- la classification qui associe une étiquette de classe à chaque objet ;
- la détection enfin, qui cherche à construire les objets d'une classe d'intérêt dans l'image.

Pour chacune de ces tâches, nous proposons une ou plusieurs méthodes utilisant les connaissances disponibles et appliquons celles-ci dans le domaine de la télédétection.

1.2.1 Segmentation

Nous commençons par traiter du problème de réglage des paramètres. Le choix des paramètres de segmentation est un problème important [22], l'utilisateur devant en général procéder par essais-erreurs jusqu'à obtenir un résultat convenable. La méthode que nous proposons détermine automatiquement les paramètres à l'aide d'un algorithme évolutionnaire guidé par

des exemples d'apprentissage. Elle permet de simplifier la tâche de l'utilisateur et d'obtenir de meilleurs résultats.

Nous proposons ensuite de remettre en cause l'hypothèse souvent faite par les algorithmes de segmentation considérant qu'un objet sémantique donné est composé de pixels connexes similaires. Nous postulons que cette hypothèse n'est pas adaptée aux objets hétérogènes et complexes. Nous introduisons une connaissance exogène, sous forme d'une classification floue basée pixels afin de transformer l'espace des données. Nous utilisons ainsi une vision alternative et plus sémantique des données. Chaque pixel est représenté non plus par ses attributs visuels propres à l'image, mais par des appartenances à différentes classes sémantiques. Appliquer un algorithme de segmentation sur un tel espace permet d'obtenir des segmentations plus conformes aux attentes.

Enfin, nous proposons d'utiliser conjointement ces deux méthodes. Les résultats obtenus mettent en avant une forte synergie entre les méthodes. Nous développons ainsi une nouvelle méthode de segmentation ne nécessitant aucun paramètre car guidée par les exemples.

1.2.2 Classification

Le processus de classification basée objets s'applique sur le résultat du processus de segmentation. Nous traitons tout d'abord du problème de création d'exemples dans le cas d'une classification supervisée par des exemples. Nous constatons en effet que l'utilisation directe d'exemples fournis par l'expert donne des résultats de qualité moindre du fait de la non représentativité des objets de l'expert par rapport aux objets obtenus par segmentation automatique. Nous étudions ainsi plusieurs méthodes de projection des objets de l'expert dans la segmentation afin d'obtenir une meilleure classification.

Après avoir étudié l'utilisation d'exemples pour la classification, nous présentons une méthode de classification capable d'utiliser les connaissances du domaine. L'utilisation de ces connaissances permet notamment de réduire, voire d'annuler, le besoin d'exemples. L'algorithme fonctionne en transformant les connaissances du domaine en contraintes que doit vérifier une région pour appartenir à une classe. Nous constatons qu'il est nécessaire d'introduire dans notre algorithme un paramètre pour relâcher ces contraintes. À nouveau, cela est rendu nécessaire pour permettre d'appliquer les contraintes sur des régions issues d'une segmentation automatique.

Nous analysons ensuite les différents avantages et inconvénients propres à chacune de ces méthodes de classification de régions. Nous introduisons une approche utilisant les résultats de ces différentes classifications. Les résultats obtenus montrent que de nombreuses complémentarités peuvent ainsi être exploitées.

Nous terminons les travaux relatifs à la classification en proposant une approche pour résoudre le problème majeur de cette étape, qui est la qualité de la segmentation. Nous proposons une méthode basée sur un algorithme évolutionnaire permettant de trouver la meilleure segmentation à utiliser lors de la classification basée régions. Cette méthode permet de déterminer automatiquement les paramètres de la segmentation pour maximiser le résultat de la classification.

1.2.3 Détection

Après avoir présenté nos travaux sur la segmentation et la classification d'objets, nous nous intéressons à une approche originale et radicalement différente. L'approche par segmentation puis classification d'objets effectue une séparation séquentielle, i.e. un algorithme de segmentation est appliqué puis, ensuite, un algorithme de classification des objets produits. L'approche par détecteur, que nous proposons ici, effectue une séparation parallèle [6]. En effet, plutôt que de

faire une séparation entre la segmentation et la classification, nous séparons le traitement de chaque classe d'objets. Cette approche permet de définir des traitements spécifiques à chaque classe sans avoir à se conformer au schéma de segmentation puis classification.

La littérature contient une multitude de méthodes pour rechercher différentes classes d'objets (la route, les immeubles, ...). Ces méthodes sont très spécifiques et intègrent les connaissances implicites qu'ont leurs concepteurs sur les classes d'objets recherchées. Ces connaissances sont ensuite difficiles à extraire si l'on souhaite les modifier ou les améliorer.

Nous proposons une approche générique de construction de détecteurs pour chaque classe d'objets à partir d'une base de connaissances. Nous séparons ainsi clairement la partie acquisition des connaissances de la partie opérationnelle. L'objectif est de permettre à l'expert du domaine de pouvoir facilement exprimer ses connaissances sans être un expert en traitement d'image ou en extraction de connaissances.

Nous présentons deux applications de cette approche, plus précisément deux niveaux d'abstraction de détecteurs. Le premier niveau recherche des classes d'objets simples dans l'image (route, végétation, ...). Le second niveau recherche quant à lui des classes composites, par exemple la classe *zone pavillonnaire*. Pour ce faire, ce type de détecteurs se base sur les résultats d'autres détecteurs de plus bas niveau pour construire des objets de plus haut niveau.

1.3 Domaines de recherche abordés

L'ensemble des contributions de cette thèse font appel à différents domaines de recherches que nous allons présenter.

Fouille de données

La fouille de données [127] consiste à extraire des connaissances à partir de données, c'est-à-dire à découvrir des informations implicites, inconnues auparavant et potentiellement utiles, à partir d'un ensemble de données [39].

Nous nous intéressons ici à la classification supervisée. Cette tâche de la fouille de données consiste à extraire de données étiquetées un modèle permettant de prédire une étiquette pour de nouvelles données non étiquetées.

Gestion des connaissances

La gestion des connaissances, ou ingénierie des connaissances, traite de l'identification, de la création, de la formalisation, de l'organisation, du partage et de l'opérationnalisation des connaissances.

Dans cette thèse, nous nous intéressons particulièrement à l'opérationnalisation des connaissances [45], c'est-à-dire leur utilisation dans des systèmes opérationnels.

Optimisation par algorithmes évolutionnaires

Les algorithmes d'optimisation cherchent à déterminer le jeu de paramètres d'entrée d'une fonction donnant à cette fonction la valeur maximale ou minimale [34]. Les algorithmes évolutionnaires [50] sont une famille de tels algorithmes s'inspirant de la théorie de l'évolution. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.

Nous utilisons ces algorithmes comme outils pour résoudre des problèmes d'optimisation dans nos méthodes.

Morphologie mathématique

La morphologie mathématique [115] regroupe un ensemble de méthodes de traitement d'images (filtrage, segmentation, mesures, analyse de texture). Ces méthodes sont basées sur la théorie des ensembles, des treillis, de la topologie.

La morphologie mathématique est utilisée dans cette thèse pour résoudre des problèmes de traitement d'image, tant dans un but de segmentation d'image que de filtrage.

Combinaison de méthodes

Si la combinaison de méthodes n'est pas un domaine de recherche en tant que tel, il découle naturellement de la multitude de méthodes qui sont proposées dans les différents domaines de recherche (notamment dans le cadre de la fouille de données [30]). La combinaison de méthodes se base sur l'hypothèse que l'utilisation conjointe de plusieurs méthodes permet de résoudre plus efficacement un problème que l'utilisation de la meilleure méthode prise isolément.

Dans cette thèse, proposant plusieurs méthodes, nous nous attachons à les combiner pour obtenir de meilleurs résultats.

1.4 Structure du mémoire

Le plan de ce mémoire est le suivant. Nous commençons par détailler notre approche générale d'intégration de connaissances dans le processus de construction et de classification d'objets (chapitre 2). Pour ce faire, nous dressons un panorama de différentes approches que nous jugeons représentatives des recherches actuelles. À partir de ce panorama, nous présentons le schéma général sur lequel se base le reste de ce mémoire. Nous détaillons enfin les différentes connaissances utilisées dans nos contributions.

Nous nous intéressons ensuite à trois tâches dans lesquelles il est possible d'intégrer de la connaissance. La segmentation (chapitre 3) et la classification (chapitre 4) sont deux processus complémentaires qui séparent la construction de la classification des objets. La détection (chapitre 5) a pour objectif de construire des objets d'intérêt d'une classe précise à partir d'une image. Dans chacun de ces chapitres, après avoir détaillé la problématique et les mesures d'évaluation, nous proposerons des méthodes permettant l'intégration de connaissances.

Le chapitre 6 illustre nos propositions en les appliquant à un jeu de données plus important que lors de la présentation des méthodes. Ce jeu de données est composé d'une image satellite à très haute résolution de la région de Strasbourg, France qui couvre plus de 200km², ainsi qu'une seconde sur la ville de Marseille.

Enfin, nous concluons (chapitre 7) notre travail de thèse en rappelant nos contributions et en proposant des perspectives à nos travaux.

Chapitre 2

Vers une approche d'intégration de connaissances dans les processus de construction et de classification d'objets

In tomorrow's world, men will not need artificial instruments such as jets and space ships. In the world of tomorrow, the new man will 'think' the place he wants to go, then his mind will take him there.

Sun Ra, 1956

Dans ce chapitre, nous posons les bases nécessaires aux chapitres suivants. Plus précisément, nous commençons par présenter un aperçu d'approches existantes en intégration de connaissances dans les processus de construction et de classification d'objets. Nous décrivons ensuite l'approche générale utilisée dans cette thèse, approche qui sera détaillée dans les chapitres suivants. Enfin, nous présentons les sources de connaissances sur lesquelles nos méthodes s'appuient pour la construction et la classification d'objets à partir d'images.

2.1 Panorama sur les méthodes d'intégration de connaissances

Nous présentons dans cette section différents travaux qui ont été menés sur la construction et classification d'objets et l'intégration de connaissances dans ces processus. Nous pensons que ces travaux sont représentatifs de l'état actuel des recherches sur ce domaine.

Nous pouvons classer les différentes méthodes existantes en trois catégories selon le type de connaissances utilisées. Nous verrons d'abord les méthodes basées uniquement sur des exemples. Nous présenterons ensuite les méthodes utilisant des connaissances du domaine. Nous traiterons aussi des méthodes hybrides, utilisant à la fois des exemples et des connaissances des experts. Enfin, nous finirons cette section par une réflexion sur ces différentes méthodes.

2.1.1 Méthodes basées sur des exemples

La première catégorie de méthodes se base sur des exemples. Ces méthodes nécessitent donc que l'utilisateur étiquette la totalité ou une partie des pixels des images, afin que les méthodes puissent apprendre sur ces exemples.

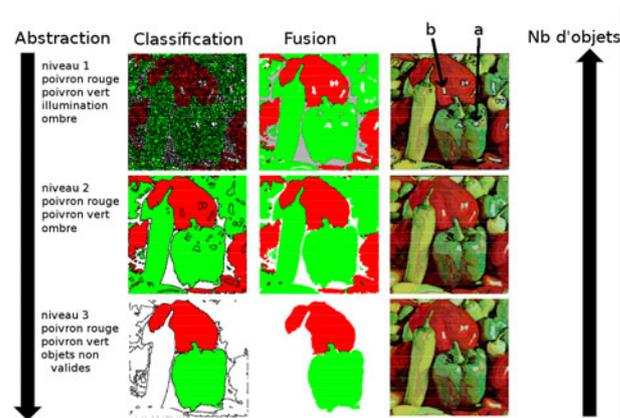


FIG. 2.1 – Résolution de problématique où l'utilisation d'une classification hiérarchique est pertinente [23].

Debeir [23] propose une approche de classification hiérarchique de métapixels. Dans son travail, une segmentation permet d'obtenir des métapixels (correspond à ce nous appelons régions dans cette thèse). Ces régions sont ensuite successivement étiquetées par plusieurs classifieurs. Chacun de ces classifieurs peut avoir des classes cibles, des exemples et des attributs différents. La structure des couches est définie par l'expert pour un problème donné.

Par exemple, prenons le cas de détection de poivrons dans l'image segmentée de la figure 2.1. Les régions pointées par *a* et *b* sont toutes les deux spectralement proches car étant des zones d'illumination. En utilisant des caractéristiques de ces régions, il est donc impossible de déterminer si elles représentent un bout de poivron vert ou bien un bout de poivron rouge. À la première étape de classification, la classe illumination est recherchée. À la seconde étape de classification, il est possible de reclasser ces régions en poivron vert ou rouge en utilisant l'étiquette de classe des régions environnantes.

Cette approche permet de résoudre les problèmes d'une segmentation en transférant à la classification, une partie de l'étape de segmentation. En effet, dans l'exemple présenté à la figure 2.1, la segmentation au début de la classification est très sur-segmentée, c'est-à-dire qu'à chaque poivron (les objets d'intérêt) correspond beaucoup de régions. La première et la seconde classification sont utilisées pour fusionner ces régions afin de faire correspondre, finalement, une seule région à chaque objet d'intérêt. Dans ce sens, seule la dernière classification correspond réellement à ce que nous entendons par ce terme. L'inconvénient de cette approche est qu'elle nécessite une importante ingénierie de la part de l'utilisateur pour définir les différentes couches de classification/fusion.

Sadgal et al. [111] proposent une méthode de segmentation multi-critères : l'image est segmentée selon la texture, selon la couleur et selon la forme. Les différentes segmentations résultantes sont ensuite fusionnées par intersection. Les régions ainsi obtenues sont ensuite comparées à des prototypes selon trois modalités (forme, couleur et texture) fournies par l'expert afin de leur affecter des étiquettes de classes possibles. L'intersection entre les classifications selon les trois modalités permet d'obtenir l'étiquette finale.

Cette approche utilise donc, tant pour la segmentation que la classification, différents algorithmes ou ensemble d'attributs. La limite de cette approche nous semble être la robustesse de la fusion des résultats : si un algorithme de segmentation ou de classification produit une erreur, cette erreur ne pourra être corrigée par les autres algorithmes.

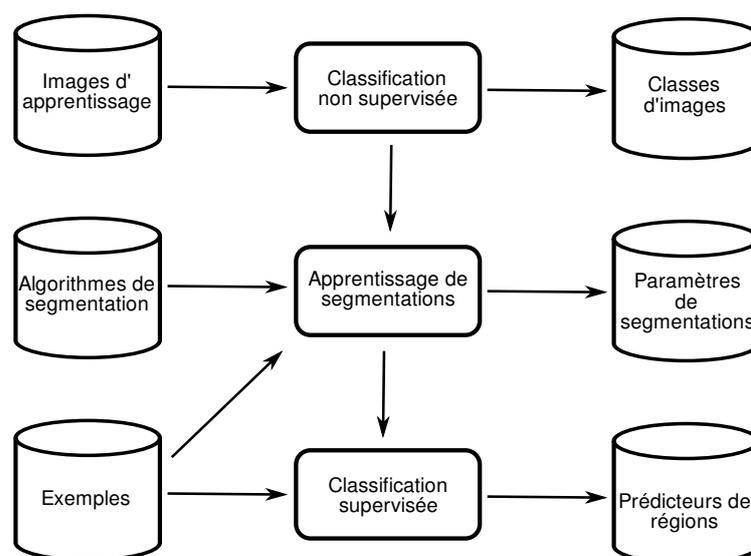


FIG. 2.2 – Plateforme de vision cognitive proposée par Martin [84].

Martin [84] propose une approche sémantique pour la segmentation d'image (voir figure 2.2). Une grande importance est donnée à l'étape de segmentation des images. Pour ce faire, une procédure d'optimisation de paramètres pour les algorithmes de segmentation ainsi qu'une procédure de sélection d'algorithmes de segmentation sont données. Trois apprentissages sont réalisés. Premièrement, un apprentissage non supervisé est réalisé pour trouver parmi l'espace des images à interpréter, des groupes d'images similaires. Deuxièmement, pour chacun de ces groupes, un apprentissage est effectué pour définir l'algorithme de segmentation ses paramètres qui doivent être utilisés. Troisièmement, des classifieurs de régions sont appris pour affecter à chaque région une étiquette sémantique.

Cette approche est dirigée par les données. En effet, pour résoudre la problématique de la segmentation, un apprentissage est effectué. Il consiste à apprendre, à partir de segmentations exemples, quel algorithme de segmentation et quels paramètres sont à utiliser. L'hypothèse est faite qu'il existe au moins un algorithme de segmentation qui, avec des paramètres adéquats, soit capable de segmenter tous les objets d'intérêt de l'image.

Ces trois approches pour l'interprétation d'images que nous venons de voir sont basées sur des exemples. Nous pouvons constater que dans chaque cas, un intérêt semble porté à la segmentation que ce soit en transférant la fusion de régions à la classification, en faisant de multiples segmentations ou en apprenant des algorithmes de segmentation spécialisés pour l'image à traiter. Nous allons à présent nous intéresser aux méthodes basées sur des connaissances de l'expert.

2.1.2 Méthodes basées sur des connaissances du domaine

L'utilisation des connaissances du domaine cherche à reproduire le raisonnement d'un expert du domaine. Cette simulation de raisonnement doit permettre de résoudre les problèmes avec la même efficacité qu'un expert.

Le système GeoAIDA proposé par Bückner et al [13] se base sur un réseau sémantique. Ce réseau sémantique est structuré par des relations de composition. Par exemple, une zone indus-

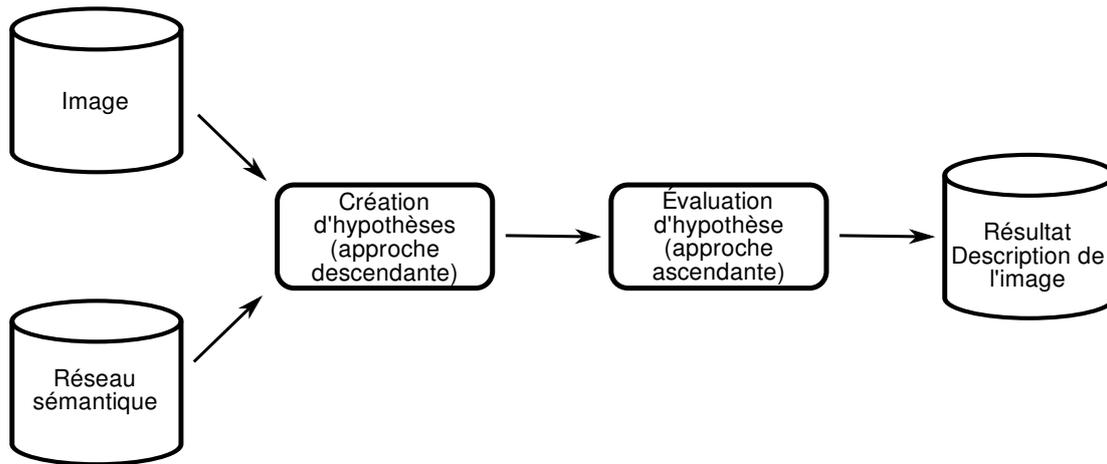


FIG. 2.3 – Le système d'interprétation sémantique d'images de télédétection GeoAIDA [13].

trielle est composée de bâtiments et de parkings. Le système fonctionne par création d'hypothèses qui sont ensuite évaluées par une mesure de qualité. La création des hypothèses se fait à l'aide d'opérateurs spécifiques à chaque classe qui subdivisent un objet de celle-ci en plusieurs objets de classes de plus bas niveau. Par exemple, à partir d'une zone habitée, il est possible de générer des maisons et des jardins. Une fois ces hypothèses générées, des opérateurs de généralisation sont utilisés afin d'évaluer ces différentes hypothèses. À la fin du processus, les hypothèses avec les meilleures qualités sont conservées. Le fonctionnement de cette méthode est illustré à la figure 2.3.

Cette approche repose sur l'hypothèse que la segmentation hiérarchique effectuée par le système de création d'hypothèse soit de bonne qualité. Si la segmentation segmente mal certains objets, ceux-ci pourront ne pas être détectés. Ceci risque d'invalider des hypothèses de haut niveau et donc de conduire à des erreurs de classification sur de larges parties de l'image suite à une erreur de segmentation localisée.

Zlatoff [135] présente une approche utilisant une ontologie du domaine. Une ontologie du domaine est un regroupement des connaissances de l'expert sur le problème. Une illustration est donnée à la figure 2.4. Tout d'abord, une segmentation est effectuée sur l'image afin de créer des régions homogènes. L'auteur indique ne pas chercher une méthode de segmentation optimale, mais une solution simple qui sépare bien les différentes instances de classes dans des régions distinctes. En effet, l'auteur considère que la sous-segmentation est à éviter, quitte à créer des zones sur-segmentées. La segmentation est issue des composantes connexes obtenues par l'algorithme K-Means [46], basé sur des indices de texture. Ces zones sursegmentées sont ensuite regroupées par une fusion de bas niveau. Cette fusion de bas niveau se base uniquement sur la connexité spatiale et la similarité des indices de texture des régions. Enfin, la dernière étape est celle de la fusion sémantique. Elle utilise des règles extraites de l'ontologie du domaine et se déroule en trois phases. Premièrement, la phase d'initiation détecte des instances qui sont connues pour être toujours présentes dans l'image. Deuxièmement, la phase de propagation utilise les règles de relations spatiales pour assigner les classes possibles à chaque instance. Troisièmement, la phase de relaxation élimine les classes impossibles de chaque instance.

Cette approche prend en compte la possibilité d'une sur-segmentation de l'image. Elle ne se base néanmoins que sur l'existence de relations spatiales, très réduites dans certains domaines d'application comme l'imagerie satellitaire.

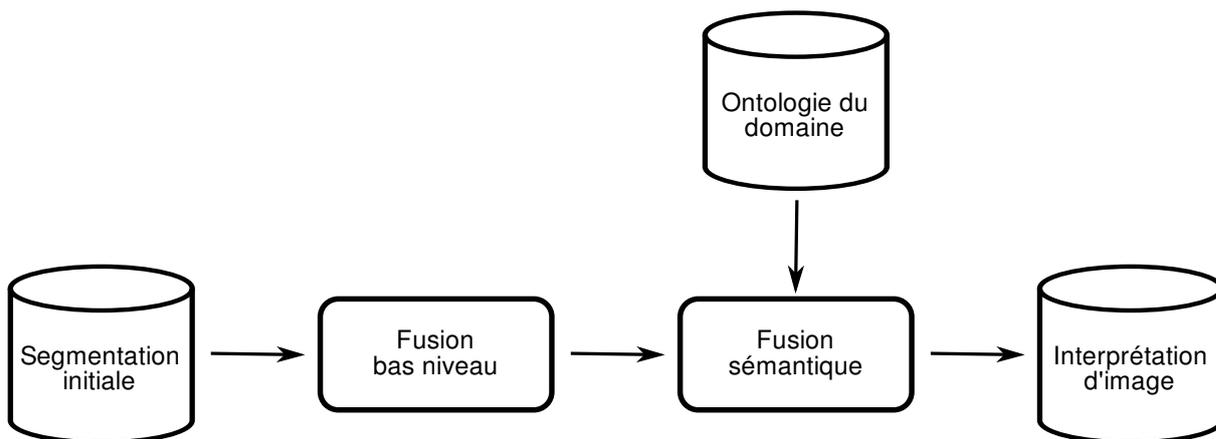


FIG. 2.4 – Moteur générique d’interprétation d’images proposé par Zlatoff [135].

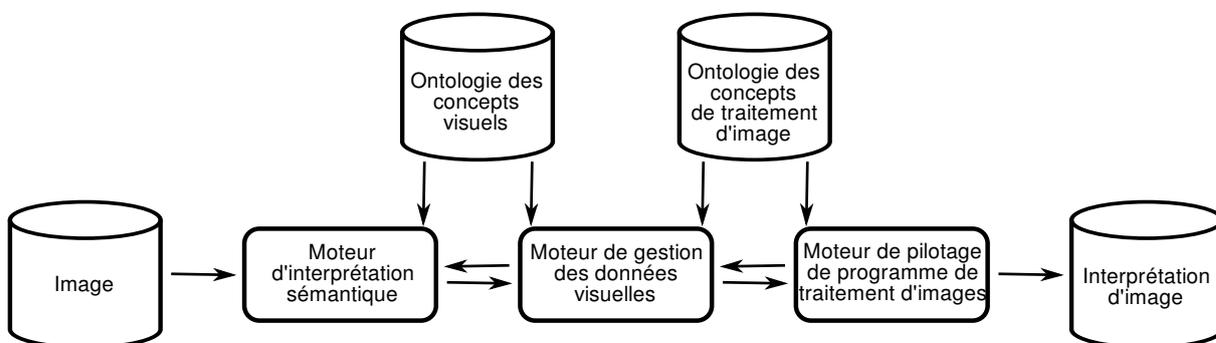


FIG. 2.5 – Plateforme de vision cognitive proposée par Hudelot [60].

Hudelot [60] propose une plateforme de vision cognitive (voir figure 2.5). Cette plateforme pose une séparation importante entre la partie sémantique, la partie visuelle et la partie de traitement d’images. Le fossé sémantique entre ces trois aspects est comblé par l’utilisation d’ontologies liant ces différents niveaux.

La partie sémantique décrit les classes du domaine fournies par l’expert, le contexte du domaine (informations non visuelles qui peuvent influencer l’interprétation), le contexte d’acquisition et les requêtes du domaine (problème initial d’interprétation). Le moteur d’interprétation sémantique utilise ces connaissances pour construire des hypothèses d’objets visuels qui sont envoyées à la partie gestion des données visuelles.

La partie gestion des données visuelles permet de faire la mise en correspondance entre les symboles abstraits utilisés par l’expert pour décrire les objets et les pixels (valeurs numériques) de l’image. Le lien entre ces symboles abstraits et pixels doit être défini par l’utilisateur à l’aide d’intervalles numériques.

Enfin, la partie de pilotage des programmes de traitement d’images s’occupe de construction des objets d’intérêt et de leurs descriptions numériques.

Renouf [107] propose un système à base d’ontologies pour permettre à un utilisateur, expert de son domaine mais non expert en traitement d’image, de formuler des processus de traitement d’images. Le schéma global est présenté à la figure 2.6. Une première ontologie fournie par l’utilisateur décrit le domaine. Celle-ci contient les informations sur le niveau physique, perceptif

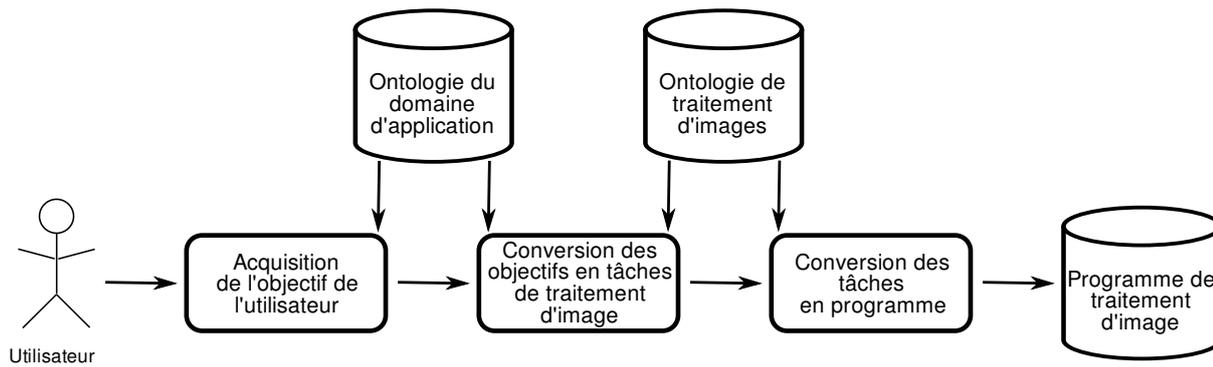


FIG. 2.6 – Vue systémique du système proposé par Renouf [107].

et sémantique de l'image. Le niveau physique décrit le système d'acquisition utilisé. On y retrouve par exemple les conditions d'éclairage, les résolutions spatiales et spectrales de l'image ainsi que des informations sur la modélisation du bruit. Le niveau perceptif traite des primitives visuelles sans traiter de la sémantique de celles-ci. Ces primitives sont définies par des descripteurs visuels, comme la luminance des régions, qui caractérisent la primitive ou le contraste de ses contours. Enfin, le niveau sémantique décrit les objets d'intérêt de l'image pour l'utilisateur. Il organise ces objets selon une structure hiérarchique. Chaque objet est défini par des primitives visuelles telles que définies dans le niveau perceptif. Sont aussi spécifiées les relations spatiales, topologiques et de composition des objets. Une seconde ontologie de traitement d'images permet de réduire le fossé sémantique entre les objectifs de l'utilisateur et les outils classiques de traitement d'images. A partir de l'ontologie du domaine et de la formulation de haut niveau de l'objectif de l'utilisateur, le système proposé permet de générer un programme de traitement d'images.

Cette approche est intéressante car elle ne nécessite pas de segmentation. Elle peut ainsi permettre à l'utilisateur de demander au système de créer un programme pour construire les objets d'une classe spécifique de l'image. Comme pour l'approche précédente, l'utilisateur doit lier les primitives visuelles aux pixels à l'aide d'intervalles sur la valeur de ceux-ci.

Ces travaux utilisant uniquement la connaissance du domaine ont des difficultés à combler le fossé existant entre les concepts visuels comme la couleur et leur représentation au niveau des pixels. Renouf [107] et Hudelot [60] demandent à l'utilisateur de fournir des intervalles sur les valeurs des pixels tandis que Zlatoff [135] se limite aux relations spatiales. Nous allons voir que l'utilisation d'exemples permet de résoudre ce problème.

2.1.3 Méthodes hybrides

La connaissance du domaine et les exemples peuvent être utilisés en même temps. C'est ce que nous allons voir avec les méthodes hybrides.

Maillot [82] propose une approche de vision cognitive qui se compose de deux phases (voir figure 2.7). Chaque classe d'objets du domaine est définie par des concepts visuels de haut niveau. La première phase est l'apprentissage de ces concepts visuels. Cette phase utilise des exemples d'images segmentées et étiquetées et un apprentissage supervisé pour faire correspondre des descripteurs de bas niveau de l'image aux concepts visuels. La seconde phase consiste en la classification des objets construits par la segmentation. Grâce à la phase d'apprentissage des concepts visuels, il est possible d'affecter à une image des concepts visuels de haut niveau. Chaque

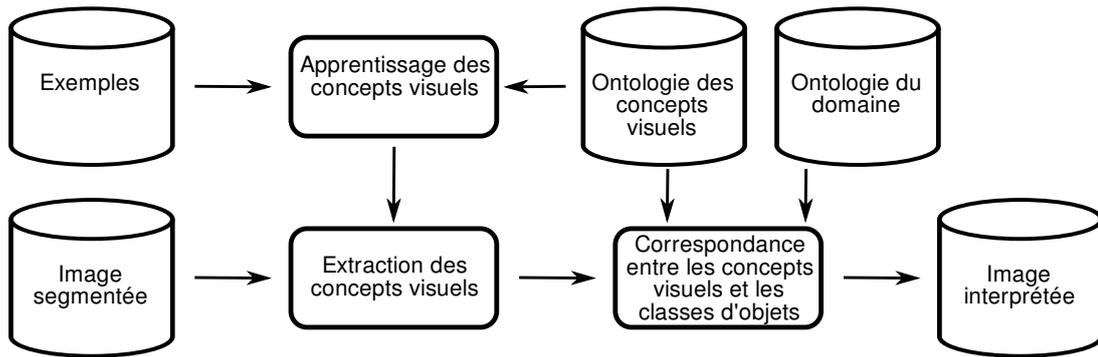


FIG. 2.7 – Approche de vision cognitive de Maillot [82].

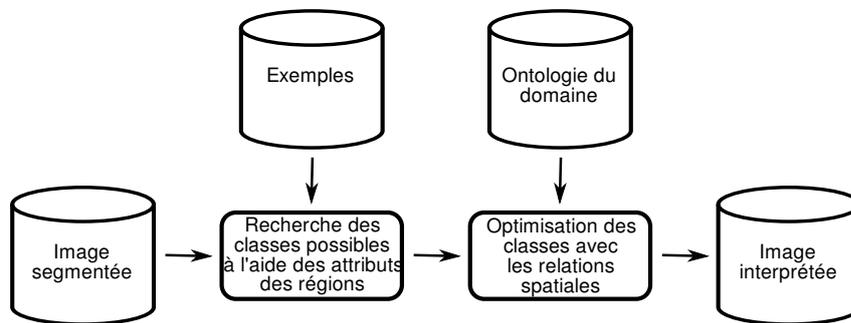


FIG. 2.8 – Système proposé par Papadopoulos et al. [99].

classe d'objet étant décrite par des concepts visuels, il suffit de trouver la correspondance entre les concepts visuels de l'image et la classe d'objet qu'ils décrivent via la génération d'hypothèses.

Cette approche cherche à réduire le fossé sémantique qu'il peut y avoir entre les connaissances du domaine et les valeurs numériques qui les représentent.

Papadopoulos et al. [99] proposent un système qui regroupe l'utilisation d'une ontologie, d'une méthode d'apprentissage et d'un algorithme évolutionnaire. Ce système est présenté à la figure 2.8. Une segmentation divise d'abord l'image en régions qui sont censées représenter des objets sémantiques (c'est-à-dire ayant une signification pour l'utilisateur). Ensuite, des caractéristiques sont extraites pour chaque région. A partir de ces caractéristiques, des hypothèses sont générées sur les classes possibles de chaque région à l'aide d'une méthode de fouille de données (ici des machines à vecteurs de support ou SVM). Chaque région est reliée à son entourage par des relations spatiales floues. L'objectif de l'algorithme évolutionnaire est de satisfaire au mieux :

- l'adéquation entre les caractéristiques de chaque région et la classe retenue ;
- le respect des relations spatiales entre les instances de classes contenues dans l'ontologie.

Dans cette approche, les exemples sont utilisés pour apprendre à fournir des hypothèses sur la classe des objets. Les connaissances du domaine (sur les relations spatiales) sont ensuite utilisées pour réduire les hypothèses.

Dans ces deux approches qui combinent connaissances du domaine et exemples, nous constatons que les exemples sont utilisés pour prendre en compte les attributs numériques difficilement exprimables par les utilisateurs en tant que connaissance du domaine.

2.1.4 Réflexions sur ce panorama

Nous avons passé en revue plusieurs systèmes d'interprétation d'images. Plusieurs éléments peuvent être extraits de ces présentations.

Tous d'abord, il semble y avoir un large consensus sur l'utilisation d'une étape préliminaire de segmentation de l'image qui n'utilise pas de connaissances. Cette étape de segmentation a cependant un objectif différent selon les approches. Pour Debeir [23], la segmentation est fortement sur-segmentée. C'est au niveau des différentes étapes de classification que des fusions seront faites. Zlatoff [135] introduit aussi le concept de fusion sémantique pour gérer les problèmes de sur-segmentation. Pour les autres auteurs, l'étape de segmentation est supposée faire correspondre à chaque instance sémantique une région dans l'image. Cette situation est impossible à obtenir dans le cas général. Afin de résoudre la problématique de la segmentation, Martin [84] propose comme piste d'introduire les connaissances pour améliorer l'étape de segmentation.

Un autre consensus semble émerger, celui de l'utilisation de méthodes de fouille de données conjointement à l'utilisation d'une base de connaissances. La réalité numérique est souvent difficilement formalisable par l'expert. L'utilisation de méthodes de fouille de données [99, 82] permet de créer des connaissances qui seraient difficilement formalisables autrement.

2.2 Schéma général

Inspirés par les approches vues dans la section précédente, nous nous proposons ici de définir le schéma général retenu pour le reste de cette thèse. Ce schéma général est présenté à la figure 2.9. Notre objectif n'est pas ici de proposer une nouvelle approche, mais de définir un cadre dans lequel nos contributions s'insèrent. Le système prend une image en entrée et fournit comme résultat un ensemble d'objets étiquetés d'une classe. Deux approches de construction et de classification de ces objets sont proposées.

La première approche sépare la partie construction de la partie classification. Tout d'abord, les objets sont construits à l'aide d'une segmentation. Ensuite, une étape de classification permet de leur affecter une classe d'objets. Cette approche est inspirée par le panorama dressé dans la section précédente qui montre qu'une séparation est usuellement faite entre la construction et la classification des objets.

La seconde approche, par détection, est plus proche des travaux de Renouf [107] qui propose un système permettant à un utilisateur de formuler des processus de traitement d'images. Nous proposons ici d'utiliser les connaissances sur les classes d'objets pour créer des processus de construction et de classification d'objets pour chacune de ces classes.

L'ensemble de ces processus est guidé par une base de connaissances contenant des connaissances du domaine et des exemples.

Nous donnons ici une courte description ainsi que nos propositions pour chaque étape du processus. La base de connaissance utilisée sera définie plus précisément à la section suivante.

2.2.1 Segmentation

La segmentation, qui est la première étape d'un processus d'interprétation basé objets, est traitée au chapitre 3. Cette étape a pour but de construire des objets sans étiquette en agrégeant des pixels connexes. Soit I l'image d'entrée et \mathcal{X} un ensemble de régions, c'est-à-dire des objets sans étiquette, le processus de segmentation est représenté par la fonction s :

$$s(I) \mapsto \mathcal{X} \tag{2.1}$$

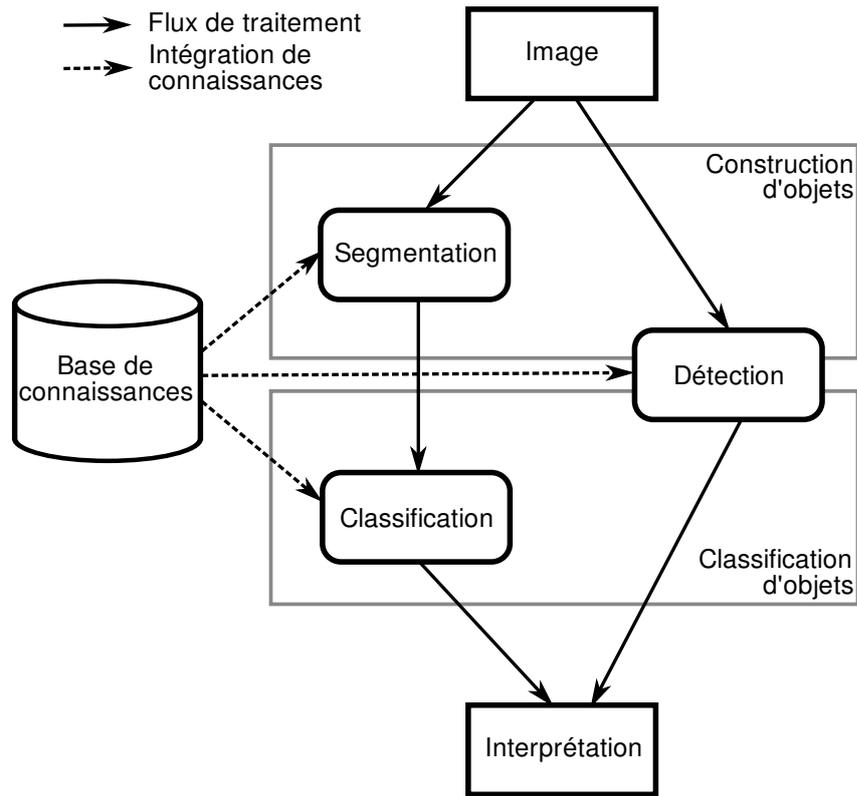


FIG. 2.9 – Schéma général de l'approche proposée dans cette thèse.

L'utilisation d'une étape de segmentation avant l'étape de classification a montré son intérêt, notamment dans le cas d'images de télédétection à très haute résolution spatiale [20, 49]. À partir d'une image d'entrée, l'étape consiste à extraire des agrégats de pixels spatialement connexes. Ces agrégats sont nommés régions, ce sont des objets sans étiquette. L'objectif implicite de la segmentation est de fournir des régions qui correspondent le plus fidèlement possible aux objets sémantiques (c'est-à-dire ayant une signification pour l'utilisateur) contenus dans l'image. Nos propositions visent à améliorer la qualité de la segmentation obtenue, qui est trop faible avec les algorithmes existants.

Comme nous l'avons vu, l'hypothèse qu'un objet sémantique donné est composé de pixels connexes similaires est souvent faite. Les algorithmes de segmentation ont souvent des paramètres donnant à l'utilisateur l'opportunité de définir sa notion de similarité. Par exemple, l'utilisateur peut définir un paramètre nommé *facteur d'échelle*. Plus ce paramètre est grand, plus les objets produits sont grands car des pixels moins similaires seront regroupés. De plus, d'autres paramètres sont proposés, comme un paramètre de compacité des régions ou de lissage des contours (paramètres usuels dans le domaine de la télédétection). Ces paramètres sont en général difficilement appréhendables pour l'utilisateur. La méthode habituelle consiste à procéder par essais/erreurs. Cette méthode est coûteuse en temps pour l'utilisateur et l'utilisateur se limitera à un nombre d'essais réduit. De ce fait, l'utilisateur se limitera à une segmentation non optimale, c'est-à-dire de qualité inférieure à ce que l'algorithme de segmentation utilisé aurait pu fournir.

Dans la section 3.2, nous étudions une approche par algorithme capable de déterminer automatiquement de bons paramètres pour une segmentation. La détermination se fait à l'aide

d'un algorithme évolutionnaire guidé par la base de connaissances, plus particulièrement par des exemples. Cette approche itérative permet d'optimiser les paramètres à utiliser. Cette technique permet de simplifier le travail de l'utilisateur. S'il dispose déjà d'exemples sur des images similaires, il n'aura qu'à lancer l'algorithme. Même s'il ne dispose pas d'exemple, il est plus facile de déterminer des exemples que des paramètres d'algorithme. De plus, ces exemples peuvent être réutilisés pour optimiser les paramètres d'autres algorithmes de segmentation dans le cas où l'utilisateur constate que les résultats obtenus avec l'algorithme ne sont pas probants.

Dans la section 3.3, nous remettons en cause l'hypothèse selon laquelle un objet sémantique donné est composé de pixels connexes similaires. En effet, comme nous le constaterons, cette hypothèse n'est pas adaptée aux objets hétérogènes. Pour résoudre ce problème, nous introduisons de la connaissances exogène, sous la forme d'une classification basée pixels pour transformer l'espace des données. Au lieu de se placer dans un espace propre à l'image, nous créons un nouvel espace plus sémantique. Ainsi, chaque pixel sera représenté non plus par des attributs visuels propres à l'image, mais par des appartenances à différentes classes. Nous montrons qu'une segmentation sur un tel espace permet d'obtenir des segmentations plus proches de l'objectif voulu par l'utilisateur.

Ces deux méthodes ont également l'avantage de pouvoir être utilisées conjointement. En effet, la première proposition traite de l'optimisation des paramètres. La seconde propose une étape de transformation des données avant d'appliquer un algorithme de segmentation pouvant nécessiter des paramètres. Ces paramètres peuvent donc être optimisés par la première proposition. Ainsi, la section 3.4 présente l'utilisation simultanée des deux propositions précédentes.

Une fois une segmentation obtenue, il est possible d'identifier la classe correspondante à chacun des objets ainsi construit. C'est l'objectif du processus de classification.

2.2.2 Classification

La classification identifie chaque objet d'une segmentation (ensemble d'objets \mathcal{X}) comme appartenant à une classe $C_i \in \mathcal{C}$, c'est-à-dire :

$$c(x \in \mathcal{X}) \mapsto C_x \in \mathcal{C} \quad (2.2)$$

C'est la problématique couverte par le chapitre 4. Il s'agit là d'un problème classique en fouille de données : affecter à un objet la classe à laquelle il correspond. Deux approches classiques sont possibles. Nous nous intéressons à utiliser des exemples et des connaissances du domaine afin de résoudre cette problématique.

La première, issue de la fouille de données, est l'utilisation d'exemples. Elle est étudiée à la section 4.2. Si l'on dispose d'un ensemble d'exemples dont la classe est connue, il est possible d'induire un modèle de prédiction. Ce modèle est capable d'affecter, à un objet de classe inconnue, une étiquette de classe. Le principal problème que nous développerons est l'inadéquation entre les exemples dont l'algorithme dispose et les objets à classer.

La seconde approche est l'utilisation de connaissances du domaine. Ces connaissances peuvent être formalisées par un expert sous forme, par exemple, d'un système expert ou d'une ontologie. On dispose ainsi d'un modèle de prédiction. Une telle approche est étudiée à la section 4.3.

De façon similaire à la partie traitant de la segmentation, il semble intéressant d'étudier la complémentarité et l'utilisation conjointe des deux sources de connaissances : les connaissances formalisées du domaine et les exemples. C'est l'objet de la section 4.4 qui constate que les deux types de modèles de prédictions peuvent avoir des caractéristiques différentes. De fait, une utilisation conjointe est proposée via la fusion de décisions.

Enfin, la section 4.5 propose une approche de remise en cause de l'étape de segmentation. En effet, en faisant l'hypothèse que la qualité d'une classification pour une segmentation donnée est calculable, il est possible de décider quelle segmentation donne le meilleur résultat pour la classification. De façon similaire à ce qui a été fait pour la segmentation (cf. section 3.2), l'utilisation de l'évaluation de l'étape de classification peut être utilisée pour trouver une meilleure segmentation. A nouveau, une approche évolutionnaire est utilisée afin de trouver les paramètres de segmentation qui optimisent l'évaluation de l'étape de classification.

2.2.3 Détection

La détection consiste à rechercher des objets d'intérêt dans une image. On appelle détecteur un algorithme capable, à partir d'une image, d'associer un ensemble d'objets de la classe qu'il recherche. Pour une classe $C_i \in \mathcal{C}$, un détecteur est donc une fonction telle que :

$$d_i(I) \mapsto \mathcal{X}_i \quad (2.3)$$

avec \mathcal{X}_i contenant l'ensemble des objets construits appartenant à la classe C_i .

La détection d'objets se place dans un autre paradigme que le processus de segmentation puis classification. En effet, dans le contexte de la détection, l'extraction de chaque classe d'objets est traitée à part. Cela permet d'appliquer une méthode spécifique pour chaque classe. *A contrario*, cela fait émerger le problème de fusion des différents objets extraits qui peuvent avoir une intersection non nulle ou dont l'union ne forme pas une partition de l'image. Nous étudierons le processus de détection au chapitre 5.

L'approche classique est de concevoir un détecteur pour chaque classe à l'aide des connaissances du domaine d'un expert. Néanmoins, en général, ces connaissances ne sont pas formalisées et sont directement transcrites en algorithmes. Cela empêche la réutilisation de ces connaissances et la maintenance ou la modification des algorithmes de détection. Dans la section 5.2, une méthodologie est proposée qui sépare clairement la formalisation des connaissances de l'algorithmique qui exploite ces connaissances pour établir un détecteur. Cette approche utilise le *design pattern* de séparation des responsabilités introduit par Dijkstra [31]. Cela permet d'une part à un expert du domaine non compétent en algorithmique de formaliser ses connaissances et d'autre part à un expert en algorithmique de créer des algorithmes utilisant ces connaissances en restant générique.

Dans la section 5.3, une méthode est également proposée pour détecter les objets composites. En effet, certains objets ne sont pas identifiables tels quels à certaines résolutions spatiales. Par exemple, un parc est composé de plans d'eau, de terrains de jeu et de végétation. Seules ces différentes composantes peuvent être détectées sur une image satellite. C'est ainsi, grâce à ses composantes, qu'il est possible d'identifier un parc. Il faut alors utiliser des connaissances spatiales. En effet, une zone de végétation dans une zone urbaine peut être un parc, mais la même zone au milieu de champs n'est vraisemblablement pas un parc. Nous étudierons comment détecter des objets composites grâce à l'utilisation conjointe de connaissances et de relations spatiales entre les objets et leur environnement.

2.3 Sources de connaissances

Chacun des processus décrits à la section précédente utilise la base de connaissances. Cet élément est donc la clef de voûte de l'approche proposée. Nous considérons la base de connaissances comme regroupant l'ensemble des connaissances disponibles, que ce soit des connaissances du

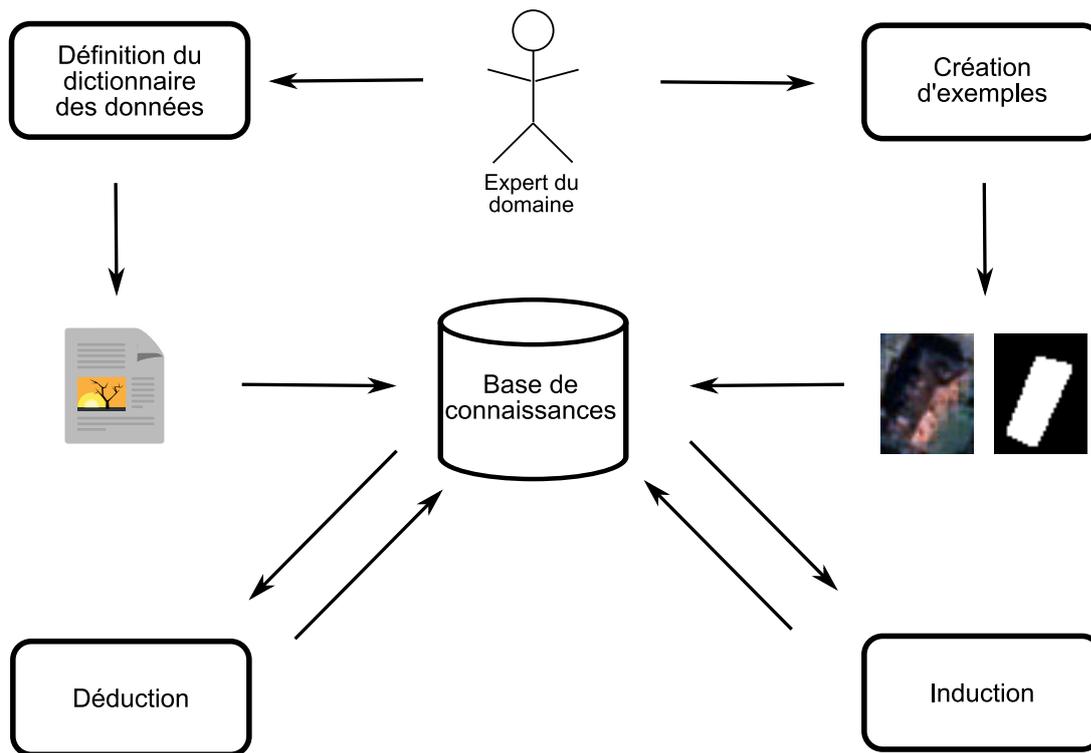


FIG. 2.10 – Création et enrichissement de la base de connaissances.

domaine ou bien des exemples. Nous allons étudier dans cette section les différentes sources de connaissances qui seront intégrées dans la base de connaissances.

La figure 2.10 illustre les sources de connaissances possibles et leur centralisation dans la base de connaissances.

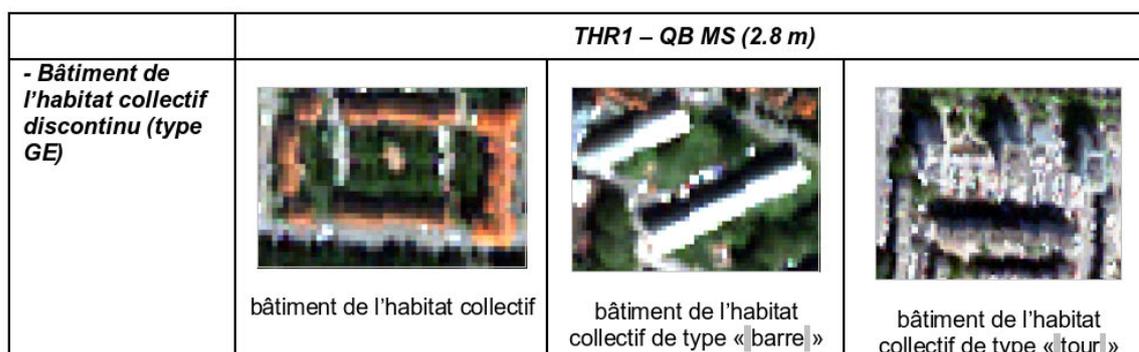
L'expert enrichit la base de connaissances par deux approches : la création d'un dictionnaire des données qui regroupe ses connaissances du domaine d'un côté et la création d'exemples de chaque classe de l'autre.

Une fois la base de connaissances initialement alimentée, elle est enrichie *via* deux mécanismes de raisonnement (le lecteur peut se référer à [58] pour plus de détails sur les mécanismes de raisonnement). Le premier est celui de la déduction à partir des connaissances du domaine. Il consiste, à partir des connaissances déjà présentes, à déduire de nouvelles connaissances *via* des inférences logiques. Le second est le mécanisme d'induction. À partir d'exemples, l'induction permet d'extraire des connaissances plus généralisées. Les mécanismes d'abduction et d'analogie ne sont pas utilisés dans cette thèse.

Dans cette section, nous commençons par préciser notre notion de connaissances du domaine et le processus d'acquisition auprès de l'expert. Nous détaillons ensuite le processus de création d'exemples et des différents types d'exemples. Enfin, nous étudions le mécanisme de déduction puis d'induction.

2.3.1 Connaissances du domaine

La première source de connaissances est celle de l'expert du domaine de la cartographie des objets urbains (utile pour l'aménagement du territoire et de l'urbanisme). En effet, celui-ci est capable de fournir une interprétation des images à analyser. Il dispose donc *a priori*

FIG. 2.11 – Illustration de la classe *Bâtiment de l'habitat collectif*.

de connaissances suffisantes pour modéliser un algorithme doué de la capacité d'interprétation d'images du domaine.

Malheureusement la réalité est plus complexe. La grande majorité des processus mis en œuvre par l'expert lors d'une interprétation est fondée sur des critères utilisés dans le domaine de la photo-interprétation tel que la couleur/teinte, la forme, la taille, la texture, la structure, le contexte [119]. L'externalisation de la connaissance est une tâche plus difficile qu'il n'y paraît. Il y a un fossé sémantique de représentation entre ce que peut formaliser l'expert et la réalité des images. Un expert a beaucoup de difficultés à s'exprimer en terme de pixels. On pourrait, par exemple, vouloir demander à un expert les valeurs que peuvent prendre les objets d'une classe donnée. Malheureusement, l'expert n'a pas cette connaissance.

Si le photo-interprète peut reconnaître une route immédiatement de part sa couleur grise (clair à foncé) et sa forme (linéaire et continue dans l'espace), il lui est impossible de préciser les intervalles de valeurs des pixels liés à ces caractéristiques. En effet, celles-ci valeurs dépendent à la fois de caractéristiques propres à l'image (résolution spectrale, espace de codage, ...) et à la fois de caractéristiques liées aux conditions externes de prise de vue (date, conditions d'ensoleillement, etc.).

Il est donc possible d'exprimer une connaissance, mais celle-ci n'est pas utilisable directement. Il existe un fossé sémantique entre cette connaissance et les données à traiter qui ne sont pas de même nature (description sémantique versus pixels numériques).

Pour autant, il est possible d'extraire une partie des connaissances de l'expert, c'est ce que nous allons voir par la suite. Celle-ci est répertoriée dans un dictionnaire de données (en partie réalisée dans le cadre de l'ACI Masse de Données Fodomust). Ensuite, nous traduisons ce document sous un formalisme plus structuré séparé en une hiérarchie de classes et un ensemble de connaissances sur ces classes.

Dictionnaire des données

Le recueil des connaissances du domaine par l'expert en photo-interprétation d'images en milieu urbain est résumé dans un dictionnaire de données, réalisé dans le cadre de l'ACI Masse de Données Fodomust par le laboratoire Image, Ville, Environnement. Ce dictionnaire est un document textuel présenté sous la forme de fiche précisant pour chaque classe d'objet susceptible d'être identifiée sur une image THR (ordre métrique) en milieu urbain les éléments suivants :

1. Identification : son nom et sa position dans la hiérarchie de classes.
2. Description dans le monde réel :

- Définition textuelle :

« La classe *bâtiment de l'habitat collectif discontinu* appelé aussi *immeuble* appartient à la catégorie de classe élémentaire *bâtiment*. Il désigne une construction durable importante, destinée à l'habitation collective en appartement ou à des activités secondaires ou tertiaires, à plusieurs étages.

On distingue le plus souvent les immeubles en *barre* et *tour*.

Une *barre* est un bâtiment de forme rectangulaire (étroit et allongé au sol) de plus de 4 étages. Une *tour* est un bâtiment de forme carrée, de volume allongé vers le haut, de plus de 6 étages. Les tours et les barres sont organisées en TU (tissus urbain) discontinu de type Grand Ensemble (GE). Les immeubles de ce type de TU se caractérisent par la régularité des directions principales (parallélisme, orthogonalité).

En général, un immeuble : est situé dans un îlot physique (domaine privé) ; a une emprise au sol de plus de 30 m² ; est associé à des surfaces artificialisées (route, parking, espaces verts, aire de jeux) organisées en quartiers (généralement relativement récents, construits après-guerre) »

- Illustration graphique : voir figure 2.11

3. Description dans l'image :

- Nature de l'objet : quelle forme (simple ou composition) prend l'objet selon la résolution spatiale utilisée.
- Définition textuelle relative à son identification dans l'image. Par exemple :

« Un objet de la classe *immeuble* est représenté graphiquement par un polygone dont la surface correspond à l'emprise au sol du bâtiment. »

- Principales relations : notamment l'adjacence avec d'autres classes, s'il existe une notion d'alignement entre des objets de cette classe ainsi que la distance entre des objets de cette classe si cela est approprié.
- Attributs : type de signature spectrale, longueur, largeur, élongation, surface, indice de Morton, type de texture, ...

4. Commentaires : par exemple des liens vers d'autres sources de données ou d'exemples possibles.

Formalisme

Une fois le dictionnaire de données construit, il faut le convertir sous une forme plus apte à être utilisée par des processus informatiques. En effet, il est difficile de traiter informatiquement des données textuelles non structurées. Nous avons adopté l'utilisation d'une représentation XML². XML est un langage de représentation de données semi-structurées. Son principal intérêt est qu'il est simple à écrire et à lire, que ce soit informatiquement ou humainement. Il autorise aussi une grande liberté sur la manière de représenter la sémantique du fichier.

La base de connaissances présentée ici peut être apparentée à une ontologie. D'après Gruber [52], une ontologie est une spécification explicitée d'une conceptualisation. Plus simplement, toujours d'après Gruber, une ontologie décrit le vocabulaire utilisé pour parler d'un domaine. A *contrario*, une base de connaissances a pour vocation d'être utilisée pour résoudre un problème ou des requêtes particulières dans un domaine. Il existe donc une différence fonctionnelle entre la base de connaissances, utilisée ici, et une ontologie.

²eXtensible Markup Language, voir <http://www.w3.org/TR/REC-xml/>

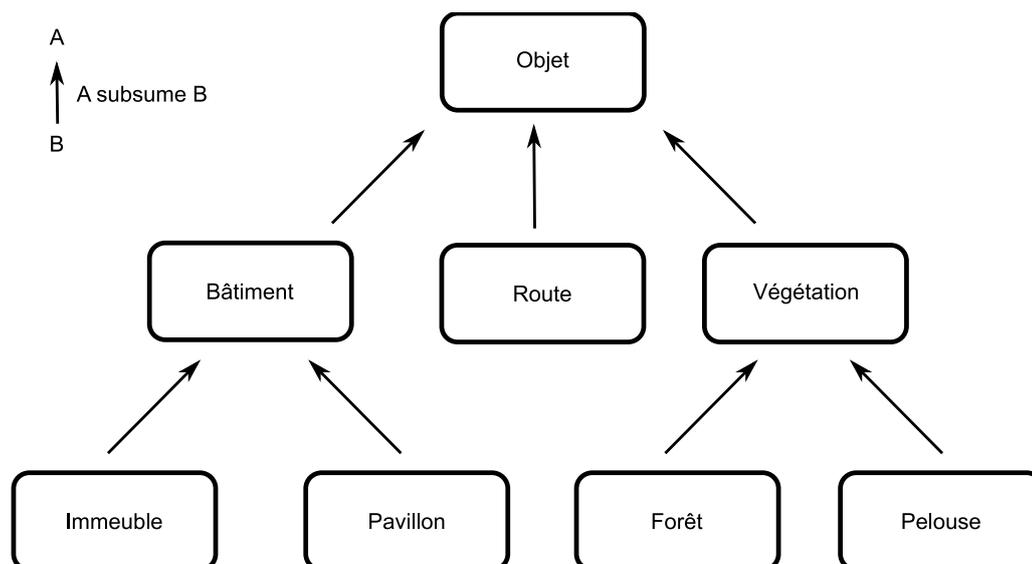


FIG. 2.12 – Exemple de hiérarchie de classes pour le domaine de l'interprétation d'images de télédétection.

La traduction du dictionnaire des données au formalisme de la base de connaissances se fait de façon manuelle. La connaissance extraite du dictionnaire des données peut être séparée en deux parties : la hiérarchie de classes, les connaissances sur les classes.

Hiérarchie de classes

La première étape est la création d'une hiérarchie de classes à partir du champ d'identification du dictionnaire des données pour chaque classe. En effet, chaque classe peut être elle-même une sous-partie d'une classe de plus haut niveau. Par exemple, comme le montre l'exemple précédent, la classe *immeuble* est une spécialisation de la classe *bâtiment*. Inversement, la classe *bâtiment* est subsumée par la classe *immeuble*, i.e. tout objet de la classe *immeuble* appartient aussi à la classe *bâtiment*.

Un exemple de hiérarchie de classes est donné à la figure 2.12. En haut se situe la classe la plus générique, celle d'*objet*. Tous les objets d'une image appartiennent à cette classe. Trois classes spécialisent la classe *objet*, il s'agit de *bâtiment*, *route* et *végétation* qui représentent trois classes d'objets. La classe *bâtiment* peut se spécialiser en *immeuble* et *pavillon*. On peut ainsi inférer qu'un objet de la classe *pavillon* est aussi un objet des classes *bâtiment* et *objet*.

Lors de l'interprétation d'une image, il est possible de définir l'ensemble des classes recherchées. En effet, selon l'interprétation désirée, l'utilisateur sera intéressé par un niveau de précision différent. Il peut être beaucoup plus simple (et donc plus précis) de trouver les objets de la classe *végétation* que de la classe *pommier*. Il est aussi possible que l'utilisateur préfère une méthode d'interprétation qui donne les résultats les plus précis possibles, mais en faisant le moins d'erreur. La hiérarchie de classes contient donc toutes les classes qui peuvent être recherchées dans une image avec des niveaux de généralisation différents.

Connaissances sur les classes

Une fois la hiérarchie de classes créée, il convient d'affecter à chaque classe la connaissance qui a pu être issue du dictionnaire de données. Plusieurs types de connaissances peuvent être

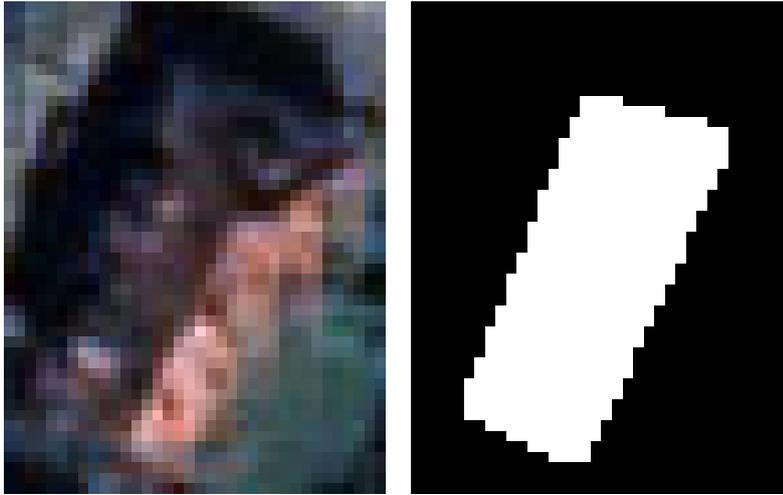


FIG. 2.13 – Exemple de maison individuelle (à gauche) et masque associé (à droite).

ajoutés :

- Intervalle sur des attributs quantitatifs. Nous avons vu avec l'exemple des immeubles, que les objets de cette classe ont leur attribut superficie (emprise au sol) supérieur à 30m^2 d'après l'expert. On peut donc en déduire que pour être défini comme un immeuble (règle nécessaire), il faut que la superficie d'une région soit dans l'intervalle $[30; \infty[\text{ m}^2$.
- Méta-connaissance sur les attributs discriminants de la classe. Pour chaque classe, les attributs discriminants ne sont pas forcément les mêmes. Par exemple, pour discriminer un objets de la classe *forêt*, la forme a peu d'importance. En effet, une forêt est protéiforme. Au contraire, pour un objet de la classe *immeuble*, la forme est très discriminante. Ainsi, à chaque attribut est associé une pondération prise dans l'intervalle $[0; 1]$ avec 0 montrant que l'attribut est sans intérêt.
- Attributs qualitatifs. Cette catégorie comprend des informations qualitatives sur la classe. On y retrouve par exemple les connaissances sur les relations des objets de cette classe avec d'autres objets. Par exemple, l'adjacence ou la proximité avec des objets d'autres classes.

2.3.2 Les exemples d'objet comme source de connaissance

Les exemples sont une façon alternative d'externaliser la connaissance de l'expert. En effet, s'il est difficile pour l'expert de donner un intervalle sur les valeurs des pixels que peut avoir un objet d'une classe, il lui est néanmoins facile de sélectionner des exemples dans des images.

La création d'exemples se fait de la façon suivante. L'expert, à partir d'une image (ou une partie d'image), dessine un masque binaire pour détourer l'exemple. L'intérêt de cette technique est sa simplicité. En effet, cette technique ne nécessite que quelques secondes pour créer un exemple. La figure 2.13 montre un exemple de maison individuelle.

Il est également possible de produire un exemple de zone où toutes les objets des classes recherchées sont détournés. Un tel exemple est donné à la figure 2.14. Puisque plusieurs classes sont représentées dans l'image, l'expert doit fournir un masque par classe. Plusieurs objets d'une même classe peuvent être identifiés sur le même masque. Cette technique est plus coûteuse en temps. Il faut s'occuper de chaque objet de la zone d'exemple. Pour autant, les objets proches spatialement sont souvent assez similaires et donc ajouter plusieurs fois des objets similaires, dans notre exemple plusieurs maisons similaires, n'est pas forcément pertinent au regard du

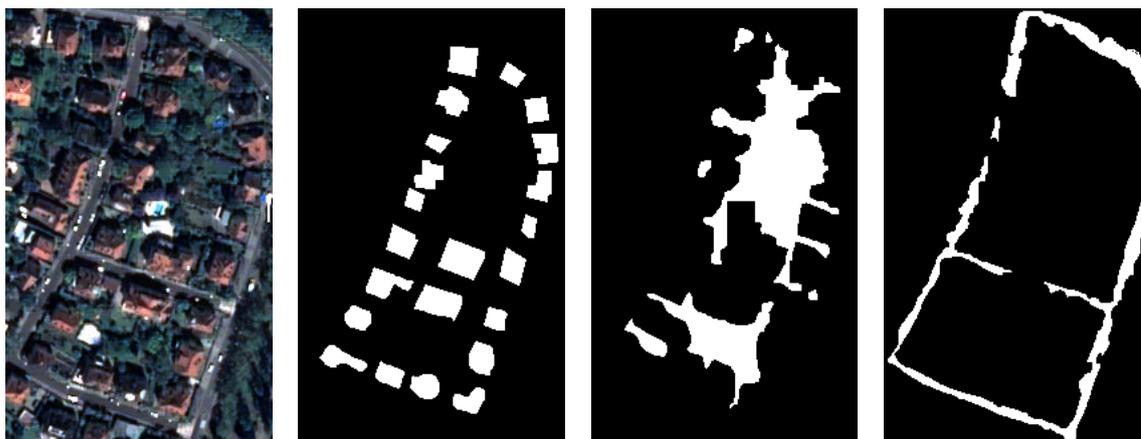


FIG. 2.14 – Zone d'exemples (à gauche), puis masques pour les classes *Maison*, *Végétation* et *Route* (de gauche à droite).

temps passé à créer cette zone d'exemples.

Nous avons donc vu qu'il existe deux types d'exemples : les exemples d'un objet isolé et les exemples d'une zone où tous les objets sont étiquetés.

La seconde forme d'exemples est plus riche. Ce type d'exemple permet d'extraire des informations sur les relations spatiales entre différentes classes d'objets. De plus, nous verrons que le fait de rapprocher les exemples permet d'utiliser certaines mesures d'évaluations de segmentation qui ne sont pas utilisables pour les exemples d'objets isolés. L'inconvénient de cette méthode est qu'elle est coûteuse en temps.

Les exemples d'objets isolés sont beaucoup plus rapides à produire. L'expert est donc plus enclin à en fournir un plus grand nombre. Cela lui permet de choisir des exemples représentatifs spatialement éloignés sur l'image.

2.3.3 Déduction de connaissances

Une fois la base de connaissances alimentée, il est possible d'utiliser des mécanismes d'inférence pour réaliser des déductions. En effet, l'utilisateur a fourni des connaissances du domaine qui ont été formalisées dans la base de connaissances. Il est possible, à partir de ces connaissances, de les combiner pour déduire de nouvelles connaissances qui n'étaient pas explicites. De plus, le mécanisme de déduction a la propriété de préserver la correction des connaissances, c'est-à-dire qu'à partir de connaissances correctes, toute connaissance déduite est nécessairement correcte. Plus formellement, soit c_1 , c_2 et c_3 des connaissances, c_3 est déduite de c_1 et c_2 si et seulement si :

$$c_1, c_2 \mapsto c_3 \quad (2.4)$$

Par exemple, supposons que nous ayons l'ensemble des connaissances suivantes :

1. La classe *bâtiment* subsume la classe *immeuble*. Un objet de la classe *immeuble* est donc aussi un objet de la classe *bâtiment*.
2. Les objets de la classe *bâtiment* ont une superficie supérieure à 12 m².

À partir de ces deux connaissances, nous pouvons déduire qu'un objet de la classe *immeuble* a une superficie supérieure à 12 m². Cette nouvelle connaissance était déjà présente implicitement



FIG. 2.15 – Exemples de cygnes.

dans l'énoncé des deux connaissances initiales. Néanmoins, le fait de l'extraire nous permet de la rendre disponible de manière explicite aux algorithmes utilisant la base de connaissances.

Une deuxième utilisation de la déduction est de compléter les relations d'adjacence ou de proximité. Par exemple, si un objet de la classe *immeuble* peut ou doit être à proximité d'un objet de la classe *route*, on peut en déduire qu'un objet de la classe *route* peut se trouver à proximité d'un objet de la classe *immeuble*. À noter qu'ici, même si l'on a une relation de nécessité de proximité d'un objet de la classe *route* pour un objet classe *immeuble*, on ne peut déduire l'inverse. En effet, un objet de la classe *route* n'est pas forcément à proximité d'un objet de la classe *immeuble*.

Enfin, une dernière utilisation de la déduction est la détection d'incohérences dans la base de connaissances. En effet, les connaissances sont fournies par l'utilisateur, expert dans son domaine. Néanmoins, il est possible qu'il commette une erreur (provenant soit d'une erreur dans ses connaissances, soit d'une erreur durant le processus de transmission de ses connaissances).

Reprenons l'exemple précédent en ajoutant une troisième connaissance :

1. La classe *bâtiment* subsume la classe *immeuble*. Un objet de la classe *immeuble* est donc aussi un objet de la classe *bâtiment*.
2. Les objets de la classe *bâtiment* ont une superficie supérieure à 12 m².
3. Les objets de la classe *immeuble* ont une superficie supérieure à 5 m².

Comme précédemment, on peut déduire des connaissance 1 et 2 que les objets de la classe *immeuble* ont une superficie supérieure à 12 m². Or cet intervalle est plus restrictif que celui qu'a défini l'utilisateur sur le même attribut dans la classe *immeuble*. Il indique qu'un objet ayant une superficie de 6 m² ne peut être un objet de la classe *bâtiment* (connaissance 2) et donc on en déduit (par la connaissance 1) qu'il ne peut être un objet de la classe *immeuble*. Pour autant un tel objet peut être un objet de la classe *immeuble* d'après la connaissance 3. Il y a donc une incohérence qu'il faut corriger.

Nous avons vu l'approche de l'utilisation de l'inférence de déduction sur la base de connaissances : la propagation des connaissances des classes générales sur les classes qu'elles subsument, la complétion des relations spatiales, et enfin la correction d'erreurs.

2.3.4 Induction de connaissances à partir des exemples

Alors que la déduction cherche à extraire des connaissances plus spécifiques de connaissances générales, l'induction cherche à inférer des connaissances plus générales à partir de faits particuliers. Dans notre cas, il s'agit des exemples. En effet, les exemples représentent des objets de chacune des classes. L'objectif de l'induction est, à partir de ces exemples, de trouver des connaissances sur la classe. Par exemple, la figure 2.15 donne des exemples de cygnes. Un lecteur attentif pourra constater que tous ces cygnes sont blancs. Il est alors possible d'induire que toutes les instances de la classe *cygne* sont blanches.



FIG. 2.16 – Un cygne noir. Contradiction de la connaissance induite que tous les cygnes sont blancs.

Nous pouvons ajouter la connaissance que les instances de la classe *cygnes* sont blanches dans la base de connaissances. Ainsi, si nous avons une image à interpréter et que l'on y reconnaît une instance d'animal, il sera possible d'infirmer le fait que cet animal soit un cygne s'il n'est pas blanc. Cette connaissance n'est pourtant pas exacte. Contrairement à la déduction, qui génère des connaissances correctes si l'on suppose les connaissances initiales comme correctes, l'induction peut générer des connaissances fausses. Par exemple, la figure 2.16 représente un cygne, mais celui-ci n'est pas blanc et donc contredit la connaissance qui a été induite des exemples. C'est un contre-exemple. Il est à noter qu'il suffit d'un cygne noir pour contredire cette connaissance, même si cette connaissance a été induite par une infinité d'exemples. Dans les cas où il n'est pas possible de connaître tous les objets d'une classe, l'existence d'un contre exemple ou d'une exception est toujours possible.

Pour être utile, l'induction doit utiliser ce que l'on appelle des *exemples représentatifs*, c'est-à-dire une diversité d'exemples qui couvrent l'ensemble des caractéristiques que peuvent avoir les instances d'une classe. La notion de représentativité est un sujet complexe et dépend du domaine considéré.

Dans le cadre de ce travail, nous utiliserons deux méthodes d'induction : une approche basée sur la caractérisation exhaustive et une approche basée sur la fouille de données.

Induction par caractérisation exhaustive

La première approche pour faire de l'induction est simple. Supposons que chaque objet soit représenté par deux attributs numériques. Il est possible d'induire les intervalles les plus restrictifs pour chaque attribut, définissant un hypercube qui contient tous les exemples de la classe à caractériser. La figure 2.17 montre cela : un carré est créé, qui contient tous les exemples, tout en étant le plus petit possible.

Ainsi, pour un objet de classe inconnue, il est possible de connaître toutes les classes auxquelles il peut possiblement appartenir. En effet si l'objet ne rentre pas dans l'hypercube défini par une classe, cet objet ne peut appartenir à cette classe. Nous faisons l'hypothèse d'un monde fermé, i.e. où chaque objet appartient forcément à une des classes considérées. Si un objet est situé dans l'hypercube d'une seule classe, alors il appartient nécessairement à cette classe.

Cette technique a néanmoins une limite. Deux hypercubes représentant deux classes peuvent avoir une intersection non nulle. Dans ce cas, il est difficile de savoir à quelle classe l'objet appartient. Or, il semble raisonnable de se demander à quelle classe l'objet est le plus vraisemblablement affilié. Nous sommes donc contraints, pour gérer ces cas, d'utiliser des techniques plus avancées.

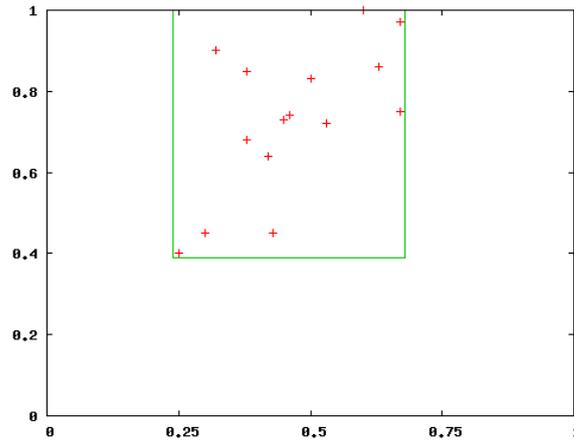


FIG. 2.17 – Exemple de caractérisation d'un ensemble d'exemples. Toutes les exemples sont contenus dans le carré.

Induction par extraction de connaissances

L'extraction de connaissances à partir de données est définie comme un processus non-trivial de découverte d'informations implicites, inconnues auparavant et potentiellement utiles à partir des données [39]. Alors que la caractérisation vue précédemment était un processus fort trivial dont nous avons vu les limites, nous nous intéressons ici à des méthodes plus avancées pour extraire des connaissances.

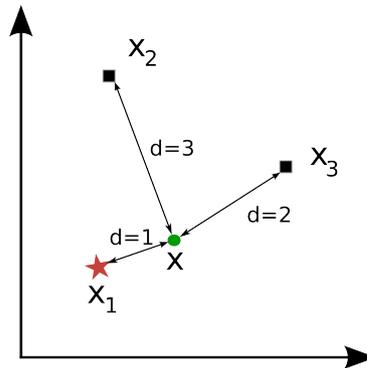
Plus précisément, nous cherchons à induire des modèles de prédiction à partir des exemples. Soit \mathcal{X} l'espace des objets à classer et \mathcal{C} l'ensemble des classes possibles. L'objectif est de créer une fonction f , qui à une instance x de \mathcal{X} , associe une classe C_x de \mathcal{C} . Plus formellement :

$$\begin{aligned} f: \mathcal{X} &\rightarrow \mathcal{C} \\ f(x) &\mapsto C_x \end{aligned} \quad (2.5)$$

Le lecteur intéressé par un descriptif plus exhaustif des algorithmes d'induction de modèle peut se référer au livre de référence de Mitchell [91]. Nous n'abordons ici qu'une méthode, qui sera réutilisée dans le reste de cette thèse : l'apprentissage à base d'exemples et plus précisément, le modèle des k plus proches voisins [1].

Cette méthode fonctionne de la manière suivante : l'ensemble des exemples d'apprentissage est stocké dans une mémoire. Lorsque l'utilisateur désire affecter une classe à un objet, l'algorithme recherche les k exemples d'apprentissage les plus proches de cet objet dans l'espace des données. Plusieurs méthodes existent pour décider de la classe à affecter à l'objet. La méthode la plus simple est d'affecter la classe la plus présente sur les k exemples d'apprentissages sélectionnés (méthode de vote). Il est aussi possible de pondérer ce vote. En effet, un exemple éloigné de l'objet à classer peut se voir affecter une importance moindre. On peut par exemple utiliser l'inverse de la distance de l'exemple à l'objet comme pondération.

Un exemple de classification par k plus proche voisins est donné à la figure 2.18. Dans ce cas, nous prenons $k = 3$. L'objectif est d'affecter une classe à l'objet représenté par un rond vert. Deux classes sont représentées dans les trois plus proches voisins (étoile rouge et carré noir). Si l'on utilise la technique d'un vote non pondéré, la classe prédite sera carré noir. A contrario, si l'on utilise une pondération par l'inverse des distances, la classe prédite sera étoile rouge. En effet, même si à proximité de notre objet à classer il y a une plus forte présence de carrés noirs,

FIG. 2.18 – Exemple de classification par k plus proches voisins.

l'étoile rouge est beaucoup plus proche.

Cette méthode, bien que simple et ancienne, se révèle être performante. Elle est notamment bien adaptée aux problèmes représentés par des attributs numériques.

2.4 Conclusion

Au cours de ce chapitre, nous avons dressé un panorama de méthodes pour l'utilisation d'exemples et de connaissances du domaine pour la construction et la classification d'objets à partir d'images. À partir de ce panorama, nous avons défini l'approche générale que nous allons détailler dans les chapitres suivants. Nous développerons deux approches aux problématiques différentes.

La première approche utilise une segmentation pour la construction des objets puis une classification basée objets. La seconde, l'approche par détecteurs, construit pour chaque classe un détecteur qui aura pour charge de construire et de classer les objets de cette classe.

Ces deux approches se basent sur la description des connaissances que nous avons faites à la fin de ce chapitre. Ces connaissances sont composées d'exemples et de connaissances du domaine.

Au cours des chapitres suivants, nous allons proposer des méthodes pour utiliser ces connaissances dans les processus de construction et de classification d'objets.

Chapitre 3

Segmentation supervisée par des exemples

La chose créée ne peut être égale à l'être créant : est-il possible que la montre soit l'horloger ?

Marquis de Sade

3.1 Segmentation et évaluation

Comme nous l'avons vu au chapitre précédent, la segmentation est une étape qui regroupe les pixels de l'image en composantes connexes ou régions. Cette étape est généralement effectuée de manière non supervisée, c'est-à-dire sans utiliser d'exemples.

Dans ce chapitre, nous allons proposer des méthodes pour introduire des connaissances dans le processus de segmentation. Notre objectif est d'atteindre une segmentation plus proche des objets sous-jacents de l'image grâce à cet ajout de sémantique.

Dans cette section, nous présenterons la problématique de la segmentation. Nous l'illustrerons ensuite à travers deux techniques usuelles de segmentation que nous pensons représentatives de l'état de l'art. Afin d'évaluer la qualité des segmentations obtenues, nous passerons en revue les principaux critères d'évaluation de segmentation. Enfin, nous décrirons les méthodes étudiées durant cette thèse et qui seront présentées à la suite.

3.1.1 Problématique

La segmentation est un problème complexe et mal posé : complexe car pouvant générer de nombreuses solutions et mal posé car il est difficile de savoir si une solution est meilleure qu'une autre.

Une image I est un ensemble de n pixels p_i avec $i \in [1; n]$. Chaque pixel p_i a une abscisse $x(p_i)$ et une ordonnée $y(p_i)$ dans la matrice de pixels composant l'image. Nous définissons les fonctions voisin (eq. 3.1) et voisinage (eq. 3.2) indiquant respectivement si deux pixels sont voisins et l'ensemble des voisins d'un pixel. La fonction $\text{abs}(\cdot)$ renvoie la valeur absolue d'un nombre.

$$\text{voisin}(p_i, p_j) = \begin{cases} \text{vrai} & \text{si } \max(\text{abs}(y(p_i) - y(p_j)), \text{abs}(x(p_i) - x(p_j))) = 1 \\ \text{faux} & \text{sinon} \end{cases} \quad (3.1)$$

$$\text{voisinage}(p_i) = \{p_j \in I \mid \text{voisin}(p_i, p_j) = \text{vrai}\} \quad (3.2)$$

La segmentation consiste à créer un ensemble de régions. Ces régions pourront être considérés comme des objets pour une classification ultérieure. Nous utilisons le terme *segment* pour désigner une région issue d'une segmentation. Nous définissons une segmentation $\mathcal{R} = \{R_j\}_{j \in [1; \Omega(\mathcal{R})]}$ (avec Ω l'opérateur de cardinalité) telle que :

- chaque pixel p_i appartienne à un et un seul segment R_j :
 $\forall p_i, \exists R_j$ tel que $p_i \in R_j$ et $\forall k \neq j, p_i \notin R_k$
- chaque segment R_j soit une composante connexe, c'est-à-dire :
 $\forall p_1, p_2 \in R_j, \text{chemin}(R_j, p_1, p_2) = \text{vrai}$, avec $\text{chemin}(R_j, p_1, p_2)$ défini par la fonction 3.3.

$$\text{chemin}(R_j, p_1, p_2) = \begin{cases} \text{vrai} & \text{si } p_1 = p_2 \\ \text{vrai} & \text{si } \exists p_3 \in \text{voisinage}(p_1) \mid p_3 \in R_j, \text{chemin}(R_j, p_3, p_2) = \text{vrai} \\ \text{faux} & \text{sinon} \end{cases} \quad (3.3)$$

Avec une définition aussi réduite, les segmentations possibles sont nombreuses. Ainsi, avec une image de 2×2 pixels, il existe 12 segmentations possibles.

Si l'être humain a du mal à définir plus précisément le concept de segmentation, c'est car lui-même n'a pas de définition précise universellement reconnue. Pour une même image, plusieurs personnes donneront des segmentations différentes. De même, d'un jour sur l'autre, pour une image et une personne donnée, la segmentation pourra être différente.

L'université de Berkeley ³ a mis à disposition une base de données d'images et de segmentations. Un exemple de segmentations faites par différentes personnes pour une même image est illustré à la figure 3.1. Selon la personne qui a produit la segmentation, le nombre de régions peut varier de 13 à 40, soit du simple au triple. De plus, nous pouvons observer des différences quant à la localisation des frontières des segments.

Si l'on suppose connaître les objets recherchés dans l'image, deux types de problèmes peuvent intervenir dans une segmentation : la sur-segmentation et la sous-segmentation. La sur-segmentation intervient quand l'un des objets recherchés dans l'image est représenté dans la segmentation par plusieurs segments. La sous-segmentation survient quand un segment recouvre plusieurs objets recherchés dans l'image. Ces deux problèmes peuvent réduire la qualité de l'interprétation que l'on peut obtenir à partir de la segmentation.

La segmentation d'une image est donc une tâche problématique. Néanmoins, toutes les segmentations de la figure 3.1 ont un point commun : des pixels similaires sont regroupés dans la même région. De cette constatation découle une hypothèse de similarité des pixels d'une même région : les objets sémantiques de l'image seraient composés de pixels similaires. Cette hypothèse de similarité des pixels à l'intérieur d'une région sert de base à la plupart des algorithmes de segmentation. L'annexe A présente une taxonomie des méthodes de segmentation et deux méthodes usuelles de segmentation.

Afin de pouvoir évaluer des segmentations produites par différents algorithmes, nous allons étudier des critères permettant d'évaluer les segmentations obtenues.

3.1.2 Critères d'évaluations

L'évaluation est une étape essentielle lors de la création d'un algorithme. Sans critère d'évaluation, il est délicat de comparer deux algorithmes. Il est évidemment possible de faire une inspection visuelle des segmentations obtenues mais ce travail serait fastidieux et manquerait

³<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

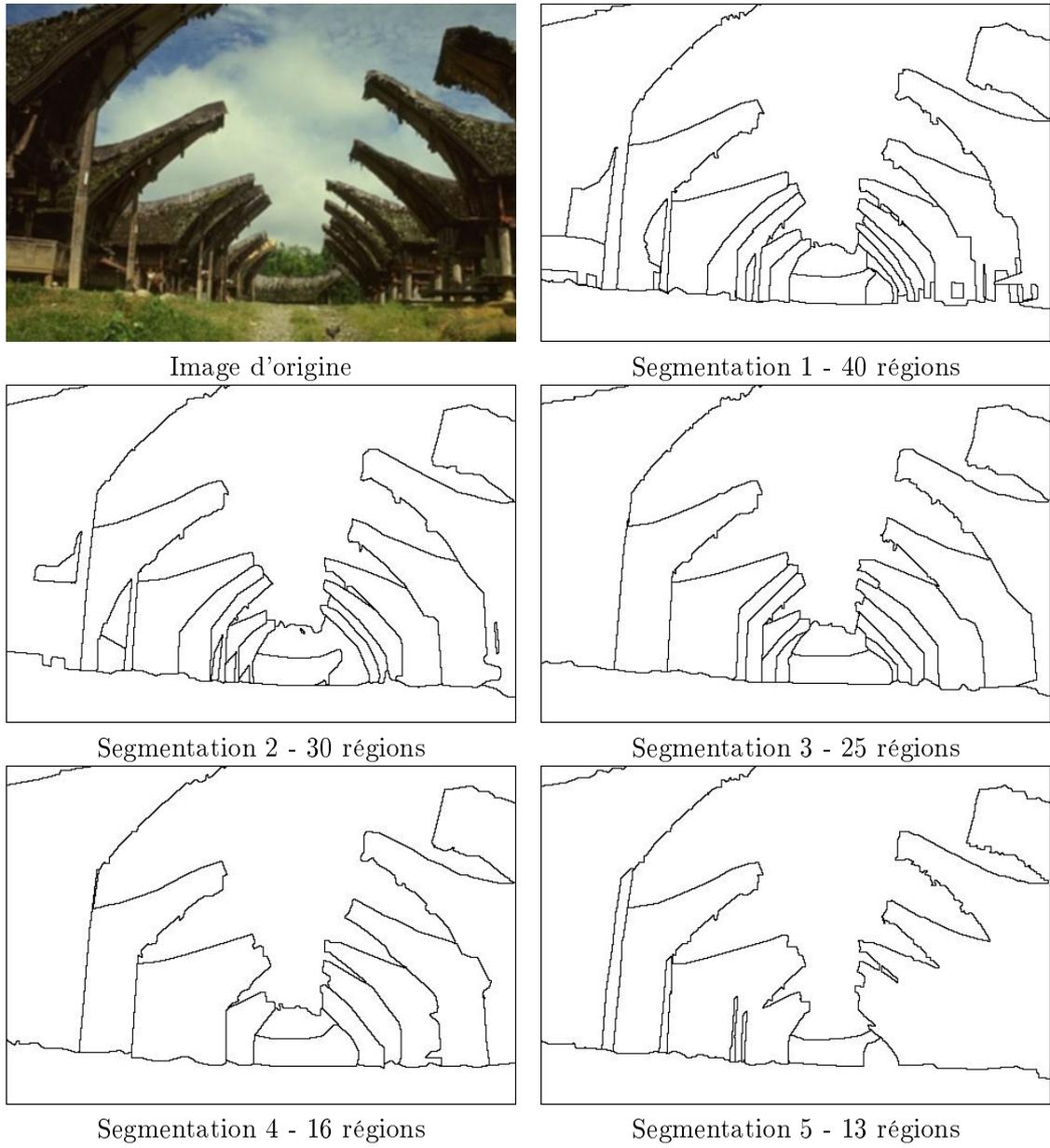


FIG. 3.1 – Exemples de segmentations obtenues par des personnes différentes pour une même image.

d'objectivité. De plus, nous verrons plus loin que l'étape d'évaluation peut-être intégrée directement dans un algorithme de segmentation afin que celui-ci s'oriente vers une meilleure solution.

Dans la littérature, de nombreux critères d'évaluation ont été proposés. Le lecteur peut se référer aux travaux de Zhang [129, 131] et de Cardoso et al. [14]. On distingue trois catégories de critères :

- les **critères analytiques** : ils se basent sur l'algorithme lui-même et étudient ses principes et ses propriétés. L'avantage de cette approche est qu'elle n'est pas dépendante des images sur lesquelles l'algorithme est évalué. Son inconvénient est qu'elle est uniquement qualitative ;
- les **critères empiriques non supervisés** : ils utilisent des mesures comme l'uniformité intra-région ou le contraste inter-région en s'appuyant sur des caractéristiques spectrales ou texturales. Ces critères ne sont pas adaptés aux images complexes car l'uniformité intra-région par rapport aux caractéristiques des pixels n'est pas toujours pertinente car non liée à la sémantique, qui, dans le cas non-supervisé, ne peut pas être connue ;
- les **critères empiriques supervisés** : ces critères nécessitent l'intervention de l'utilisateur pour fournir des régions de référence afin d'évaluer le désaccord entre la segmentation obtenue et les régions fournies par l'utilisateur. Ces critères permettent de donner une idée précise des résultats qui peuvent être attendus par l'algorithme. Ils ont néanmoins une grande dépendance aux images de référence utilisées.

Des critères analytiques seront donnés pour chacun des algorithmes proposés. Nous nous intéressons ici aux critères empiriques supervisés. Ceux-ci peuvent être classés en quatre groupes : les critères d'erreur de classification, les critères de généralisation, les critères de correspondance et enfin les critères hybrides.

Les critères empiriques supervisés nécessitant des exemples, nous notons $\mathcal{E} = \{E_j\}_{j \in [1; \Omega(\mathcal{E})]}$ l'ensemble des régions fournis par l'expert comme régions de référence. La classe d'une région de référence E_i , fournie par l'expert, est définie par $\text{classe}(E_i)$.

Critères d'erreur de classification

Le premier groupe de critères est basé sur le principe d'erreur de classification. Une segmentation d'image peut être vue comme un processus de classification. À chaque pixel devra correspondre une étiquette de région (au lieu de l'étiquette de classe traditionnelle). Nous définissons les fonctions suivantes :

- $\text{etSeg}(p_i)$: l'étiquette du segment associé au pixel p_i
- $\text{etRef}(p_i)$: l'étiquette de la région de référence associé au pixel p_i si elle existe
- $\text{classSeg}(p_i)$: la classe du segment associé au pixel p_i
- $\text{classRef}(p_i)$: la classe de la région de référence associé au pixel p_i

Nous pouvons ainsi utiliser des critères d'évaluations basés sur la corrélation entre $\text{etSeg}(p_i)$ et $\text{etRef}(p_i)$ ou entre $\text{classSeg}(p_i)$ et $\text{classRef}(p_i)$. Toute la problématique réside donc dans la méthode d'affectation de ces étiquettes.

Le premier critère que nous utilisons est dérivé du critère E défini dans Carleer et al. [15]. Ainsi, $\text{etRef}(p_i)$ est défini par :

$$\text{etRef}(p_i) = E_j | p_i \in E_j \quad (3.4)$$

Chaque segment R_l est étiqueté par la région de référence avec laquelle il partage le plus grand nombre de pixels. Ainsi, $\text{etSeg}(p_i)$ est défini par :

$$\text{etSeg}(p_i) = E_j \mid p_i \in R_l, E_j = \arg \max_{E_k \in \mathcal{E}} \Omega(R_l \cap E_k) \quad (3.5)$$

avec Ω l'opérateur de cardinalité.

Suivant nos notations, le critère E s'exprimerait par :

$$E = \frac{\sum_{E_j \in \mathcal{E}} \Omega(\{p_i \in E_j \mid \text{etSeg}(p_i) = E_j\})}{\sum_{E_j \in \mathcal{E}} \Omega(E_j)} \quad (3.6)$$

Ici, nous définissons TMA (Theoretical Maximal Accuracy), critère qui utilise les étiquettes de classe au lieu des étiquettes des régions de référence. Si un segment couvre deux régions de référence de la même classe, notre critère TMA ne comptabilisera pas d'erreur alors que le critère E le ferait. Ce cas est illustré à la figure 3.2. Dans cette situation, le critère E donnerait un score faible, ce qui ne serait pas très représentatif de la réalité. Le critère TMA a pour objectif de donner un score plus représentatif à ce type de situations.



FIG. 3.2 – Correspondance entre deux régions de référence (en blanc) et un segment (illustré par son contour rouge).

Nous définissons $\text{classSeg}(p_i)$ par :

$$\text{classSeg}(p_i) = \text{classe}(\text{etSeg}(p_i)) \quad (3.7)$$

Une matrice de confusion K est calculée, avec K_{ij} défini par :

$$K_{ij} = \sum_{E_k \in \mathcal{E}} \Omega(\{p \in E_k \mid \text{classRef}(p) = C_i, \text{classSeg}(p) = C_j\}) \quad (3.8)$$

Finalement, le critère TMA est défini par :

$$TMA = \frac{1}{\Omega(\mathcal{C})} \times \sum_{j=1}^{\Omega(\mathcal{C})} \frac{K_{jj}}{\sum_{i=1}^{\Omega(\mathcal{C})} K_{ij}} \quad (3.9)$$

avec \mathcal{C} l'ensemble de classes possibles.

Le critère TMA donne donc la précision maximale que l'on peut attendre d'une étape de classification effectuée sur la segmentation évaluée. S'il ne vaut pas 1, c'est qu'au moins un segment s'étend sur deux régions de référence de classes différentes.

Le critère de précision empirique EA (Empirical Accuracy) est similaire au critère TMA à la différence près que l'étiquetage des segments est fait par classification supervisée et non de façon à

optimiser le résultat. La seule modification porte donc sur la définition de la fonction $\text{classSeg}(p)$. Le score de EA est donc toujours inférieur à TMA car la classification peut commettre des erreurs. Il s'agit donc de la précision d'un système complet d'interprétation d'image (étape de segmentation et de classification). Pour un tel critère, plusieurs problèmes doivent être résolus :

- Quels attributs de région utiliser ? Une région peut être caractérisée par de nombreux attributs (par exemple les valeurs moyennes des pixels, la taille, la compacité, ...). Selon l'ensemble des attributs utilisés, les résultats peuvent varier ;
- Quel système de classification utiliser ? Il existe de nombreux algorithmes de classification dans la littérature et aucun n'est reconnu comme significativement meilleur que les autres dans toutes les situations. Selon l'algorithme choisi, la segmentation ayant le meilleur score pour le critère EA ne sera pas forcément la même ;
- Comment définir le jeu d'apprentissage ? Les méthodes de classification supervisée nécessitent des exemples d'apprentissage. Une première idée serait d'utiliser les régions de référence fournies par l'utilisateur. Néanmoins, les segments peuvent avoir des attributs très différents de ceux des régions de référence et ainsi créer des classifications de qualité médiocre. Un exemple est la classe de route. Dans les régions de référence, toute la route d'une image peut être représentée par une seule région (le réseau routier dans son ensemble). En général, une segmentation fournira uniquement des portions de route. Les régions de référence ne sont donc pas représentatives des segments qui devront être classées comme étant de la route. Il semble plus intéressant d'utiliser directement des segments étiquetés, définis à partir des régions de référence. Il reste que définir les segments à utiliser pour l'apprentissage est un problème complexe.

Ces problématiques ayant plutôt trait à l'étape de classification, nous les étudierons dans le chapitre 4.

Les critères d'erreurs de classification que nous avons vu permettent d'évaluer la sous-segmentation mais pas la sur-segmentation. En effet, une segmentation associant une région à chaque pixel donnera les meilleurs scores à ces critères. Pour autant, ce type de segmentation n'est pas pertinent. Nous allons donc voir les critères de généralisation.

Critères de généralisation

Les critères de généralisation mesurent la granularité d'une segmentation, ils permettent d'évaluer la sur-segmentation.

Le critère Gen [15] est le rapport entre le nombre de segments et le nombre de régions de référence. Il mesure la sur-segmentation et est défini par :

$$\text{Gen} = \frac{\Omega(\mathcal{R})}{\Omega(\mathcal{E})} \quad (3.10)$$

Pour faire fonctionner ce critère avec des régions de références ne créant pas une segmentation complète de l'image (i.e. $\exists p_i \in I | p_i \notin E_j, \forall E_j \in \mathcal{E}$), nous considérons uniquement les segments qui recouvrent au moins un pixel d'une région de référence. Nous prenons aussi en compte l'information de classe portée par les régions de référence en calculant la sur-segmentation moyenne des différentes classes. Ainsi, nous définissons le critère OV (over-segmentation) comme :

$$OV = \frac{1}{\Omega(\mathcal{C})} \sum_{i=1}^{\Omega(\mathcal{C})} \frac{\Omega(\{R_j | \exists p \in R_j, \text{classSeg}(p) = C_i\})}{\Omega(\{E_k \in \mathcal{E} | \text{classe}(E_k) = C_i\})} \quad (3.11)$$

Critères de correspondance

Les critères de correspondance mesurent les différences entre les segments issus de la segmentation et les régions de référence. Ces critères se basent sur une fonction d'association $m(E_i, R_j)$ qui calcule le score d'association entre une région de référence E_i et un segment R_j . Nous définissons aussi E_{R_j} l'ensemble des régions de référence qui recouvrent R_j et inversement, R_{E_i} l'ensemble des segments qui recouvrent E_i . Pour agréger cette information de correspondance au niveau de la segmentation entière, nous utilisons la valeur moyenne de correspondance μ_m du meilleur score de correspondance pour chaque région de référence :

$$\mu_m = \frac{1}{\Omega(\mathcal{E})} \sum_{i=1}^{\Omega(\mathcal{E})} \text{best}_{1 \leq j \leq \Omega(\mathcal{R})}(m(E_i, R_j)) \quad (3.12)$$

où la fonction best retourne la meilleure valeur pour le critère donné, i.e. la valeur minimale ou maximale selon le critère de correspondance.

Le premier critère, extrait de Feitosa et al. [40], est défini par :

$$F(E_i, R_j) = \frac{\Omega((E_i \cap R_j) \setminus (E_i \cap R_j))}{\Omega(E_i)} \quad (3.13)$$

Ce critère montre une tendance à favoriser la sur-segmentation et doit être minimisé pour obtenir la meilleure segmentation.

Le second critère est défini par Janssen et al. [63]. Il est semblable au critère F mais n'a pas de biais en faveur de la sur-segmentation, il considère de façon similaire la région de référence et le segment. Ce critère doit être maximisé.

$$J(E_i, R_j) = \sqrt{\frac{\Omega(E_i \cap R_j)^2}{\Omega(E_i) \times \Omega(R_j)}} \quad (3.14)$$

Dans cette formulation, si un segment R_j recouvre deux régions de référence E_i et $E_{i'}$ d'une même classe (un exemple est donné à la figure 3.2) les scores $J(E_i, R_j)$ et $J(E_{i'}, R_j)$ seront faibles. Pour autant, comme E_i et $E_{i'}$ appartiennent à la même classe, il peuvent être regroupés. Le score de correspondance $J(E_i \cup E_{i'}, R_j)$ sous-évalue donc la qualité du segment.

Ce principe nous conduit à créer un nouveau critère JC qui prend en compte les informations de classe. Pour un couple donné (E_i, R_j) , nous considérons la partie du sous-ensemble $E^{c(E_i)} = \{E_{i'} \mid \text{classe}(E_{i'}) = \text{classe}(E_i)\}$ (i.e. l'union de toutes les régions de référence $E_{i'}$ partageant l'étiquette de E_i) recouvrant R_j , soit $E^{c(E_i)} \cap R_j$.

Le critère modifié s'écrit donc :

$$JC(E_i, R_j) = \sqrt{\frac{\Omega(E^{c(E_i)} \cap R_j)^2}{\Omega(E_i) \times \Omega(R_j)}} \quad (3.15)$$

Un critère similaire est l'indice de Jaccard [62] qui doit aussi être maximisé. Il est défini comme la division entre les cardinalités de l'intersection et de l'union de la région de référence et du segment :

$$J'(E_i, R_j) = \frac{\Omega(E_i \cap R_j)}{\Omega(E_i \cup R_j)} \quad (3.16)$$

À nouveau, nous allons étendre ce critère pour qu'il tienne compte de l'information de classe :

$$JC'(E_i, R_j) = \frac{\Omega(E^{c(E_i)} \cap R_j)}{\Omega(E_i \cup R_j)} \quad (3.17)$$

Nous pouvons aussi mentionner le critère de correspondance ultime [132]. Celui-ci mesure l'écart entre les valeurs des caractéristiques extraites de E_i et de R_j . Comme il dépend fortement des caractéristiques extraites des régions, nous n'utiliserons pas ce critère.

Critères hybrides

Certains critères mesurent la sur-segmentation (comme OV) tandis que d'autres mesurent la sous-segmentation (comme TMA). Il est donc intéressant de créer des critères agrégés qui combinent certains des critères précédents. La combinaison peut être vue comme une solution pour résoudre les problèmes d'optimisation multi-objectifs. Or nous cherchons à évaluer les segmentations selon les critères de sur-segmentation et de sous-segmentation simultanément. Nous sommes donc bien dans un paradigme multi-objectif.

Une solution pour les évaluations multi-objectifs est l'utilisation d'un front de Pareto [41]. La comparaison entre différentes solutions est plus complexe à cause de l'absence d'ordre total. Il peut donc être impossible de savoir, parmi deux segmentations différentes, laquelle est la plus pertinente.

Nous définissons ici deux critères multi-objectifs qui combinent les critères TMA et OV . L'objectif de cette combinaison est de pouvoir exprimer dans un seul critère la qualité d'une segmentation, tenant compte des erreurs de sous-segmentation et de sur-segmentation. Le premier critère, TMA/OV , privilégie en premier le critère TMA et, en second, le critère OV . Ce choix est fait car une erreur de sous-segmentation est plus pénalisante qu'une erreur de sur-segmentation puisqu'elle provoque nécessairement des erreurs dans la classification ultérieure. Nous implémentons ce critère en pondérant très faiblement OV :

$$TMA/OV = TMA + \varepsilon \frac{1}{OV} \quad (3.18)$$

Le second critère hybride est $TMA \oplus OV(\alpha)$. Il privilégie aussi en premier le critère TMA , mais limite son importance par un seuil α :

$$TMA \oplus OV(\alpha) = \min(TMA, \alpha) + \varepsilon \frac{1}{OV} \quad (3.19)$$

Évidemment, le paramètre α est dépendant du contexte de l'application. Il représente la quantité d'erreurs de segmentation (mesurée par le critère TMA) que l'utilisateur est prêt à tolérer pour donner plus d'importance à la sur-segmentation (mesurée par le critère OV). Par exemple, si l'utilisateur désire que le critère TMA soit au minimum de 95%, il peut choisir $\alpha = 0.95$.

Nous avons donc défini un ensemble de critères inspirés de la littérature. Nous avons modifié certains de ces critères afin qu'ils prennent en compte la notion de classe, plusieurs régions de référence pouvant être regroupées sous la même classe, et qu'ils puissent fonctionner avec des régions de référence ne formant pas une segmentation complète. Nous avons aussi présenté des critères hybrides basés sur plusieurs critères afin de donner une évaluation d'une segmentation tenant compte simultanément de la sous-segmentation et la sur-segmentation.

Dans cette section, nous avons étudié la problématique de la segmentation. Nous avons ensuite passé en revue deux algorithmes de segmentation, parmi les plus utilisés, et pointé leurs

limites. Enfin, nous avons défini une série de critères d'évaluation pour mesurer la qualité des segmentations et compatibles avec les régions de référence dont nous disposons.

Dans les sections suivantes, nous proposons trois approches qui, se basant sur les algorithmes classiques de segmentation, vont intégrer de la connaissance afin de parvenir à de meilleures segmentations.

Dans la section 3.2, nous étudions une approche où l'algorithme est capable de déterminer automatiquement les meilleurs paramètres pour une segmentation. La détermination se fait à l'aide d'un algorithme évolutionnaire guidé par la base de connaissances et, plus particulièrement, par des exemples. Cette technique permet de simplifier le travail de l'utilisateur. S'il dispose déjà d'exemples sur des images similaires, il pourra utiliser ceux-ci pour trouver les paramètres. Même s'il ne dispose pas d'exemple, il est plus facile de déterminer des exemples que des paramètres d'algorithme. De plus, ces exemples pourront être réutilisés pour optimiser les paramètres d'autres algorithmes de segmentation si, par exemple, l'algorithme utilisé obtient des résultats non satisfaisants même avec les meilleurs paramètres.

Dans la section 3.3, nous remettons en cause l'hypothèse selon laquelle des pixels connexes différents ne peuvent appartenir à un objet sémantique identique. En effet, comme nous l'avons vu, cette hypothèse n'est pas adaptée aux objets hétérogènes. Pour résoudre ce problème, nous introduisons de la connaissance exogène, sous la forme d'une classification basée pixels pour transformer l'espace des données. Au lieu de se placer dans un espace propre à l'image, nous créons un nouvel espace plus sémantique. Ainsi, chaque pixel est représenté non plus par des attributs visuels propres à l'image, mais par des appartenances à différentes classes. Travailler sur un tel espace permet d'obtenir des segmentations plus proches de l'objectif voulu par l'utilisateur.

Enfin, ces deux méthodes peuvent être utilisées conjointement. En effet, la première proposition traite de l'optimisation des paramètres. La seconde propose une étape de transformation des données avant d'appliquer un algorithme de segmentation pouvant nécessiter des paramètres. Ces paramètres peuvent donc être optimisés par la première proposition. Ainsi, la section 3.4 présente l'utilisation simultanée des deux propositions précédentes.

3.2 Apprentissage de paramètres de segmentation par algorithme évolutionnaire

À la section précédente, nous avons introduit une série de critères d'évaluation. Nous avons défini la pertinence d'une segmentation en fonction des scores obtenus par celle-ci pour ces critères. Dans cette section, nous proposons un algorithme qui va chercher à créer des segmentations optimisant l'évaluation par ces critères [29].

Cet algorithme est inspiré du comportement humain. Face à un problème de segmentation, l'utilisateur utilise traditionnellement un algorithme de segmentation dont les paramètres sont ceux par défaut. Selon le résultat de cette première segmentation, il va modifier les paramètres puis relancer l'algorithme. Ainsi, par ce jeu d'essais/erreurs, l'utilisateur va se rapprocher des paramètres les plus pertinents pour résoudre son problème de segmentation. Cette tâche est néanmoins fastidieuse et longue. Nous nous proposons ici de l'effectuer automatiquement.

Pour résoudre cette tâche, nous utilisons un algorithme évolutionnaire [34]. Les algorithmes évolutionnaires sont une classe d'algorithmes utilisés pour résoudre des problèmes d'optimisation. Ils sont réputés pour éviter les pièges des maxima locaux. Un maximum local se produit quand une solution obtient une meilleure évaluation que les autres solutions directement voisines mais n'est pas la meilleure solution sur tout l'ensemble des solutions.

3.2.1 Algorithmes évolutionnaires pour la segmentation

L'utilisation d'algorithmes évolutionnaires pour résoudre des problèmes de segmentation n'est pas nouvelle. Nous considérons ici deux catégories de méthodes de segmentation utilisant des algorithmes évolutionnaires.

- *les méthodes directes* : Cette catégorie d'algorithmes [4, 9, 37, 59, 64, 80] utilise des algorithmes évolutionnaires pour regrouper les pixels de l'image. Très souvent une pré-segmentation de l'image est effectuée (sur-segmentée) afin de réduire l'espace de recherche. Ensuite, l'algorithme évolutionnaire est utilisé pour fusionner ces régions afin de créer des régions qui correspondent aux objets de classes.
- *les meta-méthodes* : Les méthodes appartenant à cette catégorie [10, 40, 85, 105, 116] ne travaillent pas directement sur les données de l'image. Au lieu de cela, ces méthodes opèrent sur la manière dont le processus de segmentation va s'effectuer. Pour un algorithme de segmentation donné, l'algorithme évolutionnaire va rechercher les paramètres qui optimisent un critère d'évaluation. Une autre approche développée est la création d'algorithmes de segmentation par programmation génétique [116].

Pour les méthodes directes, l'algorithme évolutionnaire effectue une tâche d'optimisation uniquement. L'objectif étant d'optimiser le partitionnement des pixels d'une image pour maximiser un critère. Si l'on veut segmenter une autre image, le processus évolutif doit être relancé. À l'inverse, l'approche par meta-méthodes n'effectue pas qu'une simple optimisation mais aussi un apprentissage. Le résultat d'une telle approche n'est pas uniquement une segmentation mais aussi un algorithme de segmentation optimisé pour le contexte considéré. En effet, les résultats obtenus par l'optimisation permettent une bonne segmentation de l'image d'entrée, mais, en plus, il est possible de généraliser ces résultats à d'autres images. Pour espérer de bons résultats, il faut néanmoins que l'image (ou l'ensemble d'images) d'apprentissage soit représentative des autres images sur lesquelles on veut appliquer le résultat (problématique propre à tout problème d'apprentissage et de généralisation).

Une autre limite des méthodes directes est qu'il n'est pas possible d'utiliser les régions de référence dans les critères d'évaluation. Dans ce cas, la meilleure évaluation serait attribuée à n'importe quelle segmentation qui, pour chaque région de référence, ferait correspondre une et une seule région. Comme les méthodes directes peuvent générer n'importe quel partitionnement, il est possible de trouver un grand nombre de solutions optimales. Néanmoins, il est impossible de déduire quoi que ce soit sur la qualité de la segmentation hors des zones des régions de référence. En effet, si l'algorithme segmente bien les régions de référence et produit autant de régions que de pixels restants, la segmentation aura quand même une évaluation optimale sur les critères d'évaluation. *A contrario* de cette approche, les meta-méthodes assurent un traitement homogène sur toute l'image, on peut donc vraisemblablement espérer que le score d'un algorithme sur les régions de référence soit similaire au score sur d'autres zones de l'image.

C'est pour ces raisons que nous avons choisi de nous focaliser sur les meta-méthodes pour élaborer des segmentations en s'appuyant sur les données d'apprentissage. Nous utilisons donc un algorithme de segmentation nécessitant des paramètres et utilisons l'algorithme évolutionnaire pour fixer ces paramètres en utilisant les connaissances portées par les régions de référence fournies.

3.2.2 État de l'art et positionnement

Nous allons à présent voir plus en détails les méthodes similaires et les contributions que nous nous proposons d'apporter. Cet état de l'art inclut aussi des méthodes qui ne se basent pas sur

des algorithmes évolutionnaires, mais qui ont néanmoins le même objectif.

Pignalberi et al. [105] traitent de la segmentation d'images de profondeur. Il s'agit d'images où chaque pixel représente l'éloignement des objets à l'instrument de mesure. La fonction d'évaluation utilisée est une somme pondérée de plusieurs critères tels que l'erreur basée pixel, la sous-segmentation, la sur-segmentation et l'absence d'étiquetage (étant donné qu'un fond est considéré, certains pixels ne sont pas inclus dans la segmentation). Les résultats sont évalués sur deux jeux de données et comparés avec des segmentations obtenues en affectant les paramètres manuellement.

Martin et al. [85] utilisent la méthode du simplex (autre méthode d'optimisation) pour réduire l'erreur entre les segmentations résultantes et des régions de référence données par l'utilisateur. Leur méthode sélectionne, à partir d'une bibliothèque d'algorithmes de segmentation, l'algorithme le plus adéquat pour chaque image. Pour faire ce choix, chaque image est décrite par un ensemble de caractéristiques. Le jeu d'apprentissage est utilisé pour déterminer pour chaque image le choix de l'algorithme de segmentation et de ses paramètres. La fonction d'évaluation à optimiser est basée sur le nombre de pixels mal classifiés en utilisant une correspondance entre les régions de référence et les régions obtenues par segmentation. L'algorithme est évalué par inspection visuelle sur des images n'appartenant pas aux images d'apprentissage.

L'approche proposée par Bhanu et al. [10] caractérise d'abord chaque image avec des statistiques et des variables externes (par exemple : l'heure, les conditions météorologiques). Ensuite, une stratégie d'évolution combinée avec une descente de gradient est utilisée pour optimiser les paramètres de segmentation. L'algorithme de segmentation est évalué sur un ensemble d'images de la même scène d'extérieur mais acquises à plusieurs moments et dans des conditions différentes. La fonction d'évaluation prend en compte cinq critères (trois nécessitant des régions de référence). Dans l'exemple d'application, seule une région de référence est donnée pour chaque image représentant une voiture. Ces critères d'évaluations sont soit combinés pour obtenir une valeur scalaire soit considérés comme un vecteur pour effectuer une évaluation multi-objectifs. L'évaluation de cette approche reste visuelle.

Dans le domaine des images de télédétection, nous pouvons mentionner les travaux de Feitosa et al. [40]. La fonction d'évaluation utilisée se base sur une correspondance entre les régions produites et celles de référence. Le jeu d'apprentissage est composé de cinq à dix régions de référence qui ont été sélectionnées par l'algorithme de segmentation de base (avec des paramètres choisis à la main). Procédant ainsi, les auteurs peuvent garantir que l'algorithme évolutionnaire aura la possibilité de retrouver des paramètres qui produisent exactement les régions de référence. De ce fait, leurs travaux portent plus sur une étude de la capacité d'un algorithme évolutionnaire à retrouver des paramètres cachés. On ne peut donc rien déduire sur les résultats qui peuvent être obtenus si les régions d'apprentissage ne peuvent être construites par l'algorithme de segmentation.

Enfin, la méthode proposée par Trias-Sana et al. [118] sélectionne des bandes d'images à utiliser grâce à l'algorithme Stepwise Forward Selection. Cette sélection est nécessaire car dans leur méthode de nombreuses représentations de couleur et de nombreux indices de texture sont extraits des images et ajoutés à la représentation de l'image. Comme beaucoup de ces bandes sont non-informatives ou corrélées, une sélection de bandes doit être effectuée. Ce processus peut-être vu comme une étape de l'algorithme de segmentation prenant en paramètre les bandes à conserver. Chaque bande est d'abord évaluée individuellement et les meilleures bandes sont conservées pour l'étape suivante. Beaucoup de combinaisons de bandes sont testées puis conservées ou supprimées. Leur algorithme cherche à optimiser un front de Pareto à deux dimensions : les fausses détections et les détections manquées.

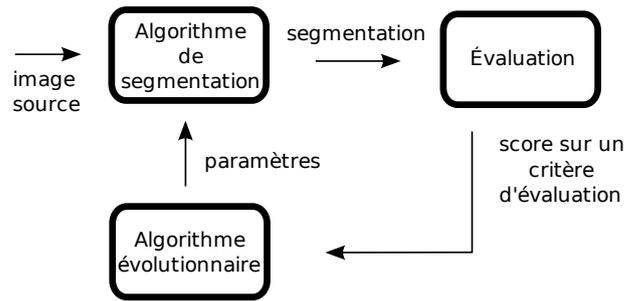


FIG. 3.3 – Illustration du processus d’optimisation des paramètres de segmentation.

Au vu de cet état de l’art sur la paramétrisation d’algorithmes de segmentation, nous pouvons observer que cette problématique reste peu explorée dans le cadre d’images de télédétection. De plus, aucune étude n’a portée sur l’évaluation de telles méthodes, soit en comparant différentes fonctions d’évaluation soit en étudiant la capacité d’apprentissage de tels systèmes.

Dans le reste de cette section nous nous concentrerons sur deux points non encore étudiés dans la littérature. Premièrement, nous allons comparer les différentes fonctions d’évaluation qui peuvent être utilisées dans l’optimisation des paramètres. Deuxièmement, nous analyserons les capacités d’apprentissage de l’algorithme proposé, c’est-à-dire l’importance de la quantité de régions de référence sur le résultat et sur la généralisation des résultats (et donc leur applicabilité à d’autres images).

3.2.3 Présentation de l’algorithme

Le fonctionnement du processus utilisé est présenté à la figure 3.3. L’algorithme se décompose en trois parties : l’algorithme de segmentation de base, l’évaluation des segmentations et enfin l’algorithme évolutif lui-même.

Algorithme de segmentation de base

Il est possible d’utiliser n’importe quel algorithme de segmentation pour cette partie de l’algorithme. Nous utilisons ici un algorithme de ligne de partage des eaux suivi d’un algorithme de fusion des régions. Ces algorithmes sont décrits en annexe A.

Cette méthode de segmentation prend donc en entrée trois paramètres : deux pour la ligne de partage des eaux (*hmin* et *hdyn*) et le seuil de fusion pour l’algorithme de fusion de régions. Nous normalisons ces paramètres afin d’avoir l’intervalle réel $[0; 1]$ comme espace de définition pour chacun des paramètres.

Évaluation

Pour la partie évaluation des segmentations, il s’agit de prendre en entrée une segmentation et de fournir une valeur d’évaluation pour cette segmentation.

Afin d’uniformiser les résultats des critères d’évaluation, nous retraits leurs résultats pour qu’un score élevé représente une bonne qualité d’évaluation. Dans le cas de critères affectant une valeur plus faible aux meilleures segmentations, l’inverse de la valeur attribuée est utilisée.

Algorithme évolutionnaire

L'algorithme évolutionnaire fournit des paramètres à l'algorithme de segmentation et reçoit l'évaluation des segmentations produites par l'algorithme d'évaluation. Son objectif est de fournir, au fur et à mesure de l'avancement de l'algorithme, des paramètres obtenant de meilleures évaluations. L'algorithme évolutionnaire [46] sert à optimiser le processus de segmentation. Le détail de l'algorithme évolutionnaire est défini à l'annexe B.

Pour la suite, nous utiliserons les paramètres suivants pour l'algorithme évolutionnaire : une taille de population de $\Omega(\mathcal{P}) = 15$ solutions, une probabilité de mutation $P_m = 1\%$, et un nombre d'évolutions de $N_e = 30$ générations. Des expériences nous ont montré que laisser plus de générations à l'algorithme n'augmentait pas significativement les résultats. Le choix de $\Omega(\mathcal{P}) = 15$ nous semble cohérent au regard de la complexité de l'espace des solutions \mathbb{G} (3 dimensions dans ce cas). Enfin, la probabilité de mutation $P_m = 1\%$ nous semble un bon compromis entre l'exploration de nouvelles solutions (par mutation) et l'affinage des solutions existantes (par croisement sans mutation).

3.2.4 Évaluation

Nous procédons ici à l'évaluation analytique puis empirique de l'algorithme que nous avons présenté.

Évaluation analytique

Nous évaluons analytiquement l'algorithme proposé en fonction de trois aspects. Premièrement, nous donnons les pré-requis nécessaires à l'application de cet algorithme. Deuxièmement, nous présentons les limites conceptuelles de celui-ci. Troisièmement, nous détaillons certaines de ses particularités.

Pour pouvoir appliquer cet algorithme, plusieurs conditions doivent être remplies :

- Régions de référence : Des régions de référence sont nécessaires au fonctionnement de l'algorithme. Selon les critères utilisés, l'algorithme peut se contenter de régions isolées ou bien nécessiter des parties de segmentations complètes sur une zone (se référer à la section 3.1.2).
- Critère d'évaluation : Pour fonctionner l'algorithme a besoin d'une fonction d'évaluation. Pour obtenir de bons résultats, l'utilisateur doit être en mesure de choisir une fonction d'évaluation qui correspond au mieux à ses objectifs (notamment sur l'arbitrage entre sur-segmentation et sous-segmentation).

L'algorithme contient des limites qu'il convient de connaître :

- Méthode de segmentation utilisée : L'algorithme ne fait qu'optimiser des paramètres pour une méthode de segmentation. De ce fait, l'algorithme proposé hérite des limites de la méthode de segmentation utilisée.
- Faculté de généralisation limitée : Il est possible d'apprendre des paramètres optimaux sur une image et de les appliquer sur une autre. Néanmoins, il est nécessaire que le jeu d'apprentissage soit représentatif des images sur lesquelles les paramètres sont utilisés. Il est utopique d'apprendre des paramètres sur des images médicales et de vouloir les appliquer sur des images satellites.

Enfin, il convient de noter certaines spécificités de cet algorithme :

- Méthode de segmentation utilisée : Il est possible de remplacer l'algorithme de segmentation utilisé par n'importe quel autre algorithme de segmentation tant que celui-ci requiert des paramètres.

TAB. 3.1 – Résultats comparatifs entre les critères utilisés pour l’optimisation et ceux utilisés pour l’évaluation sur le jeu de données urbain. Les cellules en gras montrent les cas où le même critère a été utilisé pour l’apprentissage et pour l’évaluation.

Opt. \ Éval.	TMA	OV	Feitosa	Janssen	Jaccard
<i>TMA/OV</i>	99,82% (±0,04)	69,61 (±4,59)	0,842(±0,011)	0,391(±0,015)	0,302(±0,026)
<i>TMA ⊕ OV</i>	97,58% (±0,08)	13,84 (±0,38)	0,765(±0,014)	0,593(±0,009)	0,590(±0,005)
Feitosa	99,08%(±0,18)	27,85(±1,93)	0,696 (±0,007)	0,588(±0,015)	0,579(±0,008)
Janssen	97,40%(±0,44)	16,27(±1,47)	0,741(±0,012)	0,621 (±0,005)	0,594(±0,004)
Jaccard	97,81%(±0,41)	17,53(±1,91)	0,739(±0,005)	0,606(±0,006)	0,615 (±0,004)

- Complexité : Le temps d’exécution dépend principalement de la méthode de segmentation et du critère d’évaluation utilisé. Le temps total d’apprentissage est de $\Omega(\mathcal{P}) \times (N_e + 1) \times (T_s + T_e)$ avec T_s le temps d’exécution de la méthode de segmentation et T_e le temps d’exécution de l’évaluation. Il est à noter que dans le cas d’une application de paramètres déjà appris à une image, le temps d’exécution total est T_s .

Nous allons à présent procéder à l’évaluation sur des données réelles.

Évaluation empirique

Nous allons évaluer l’algorithme proposé sur deux images satellites : une image urbaine de Strasbourg et une image rurale de Normandie. Chacune de ces images dispose de quatre bandes spectrales.

L’image urbaine de Strasbourg est illustrée à la figure 3.4. Elle a été prise par le satellite Quickbird. Le capteur fournit 4 bandes chromatiques avec une résolution spatiale de 2,7m et une bande panchromatique avec une résolution de 0,7m. La bande panchromatique est fusionnée avec les 4 autres bandes en utilisant la méthode UWT-M2 étudiée dans Puissant et al. [106] pour obtenir une image à 4 bandes chromatiques de résolution 0,7m. L’image est un extrait de 900×900 pixels de cette fusion. Les classes d’intérêt sont les maisons, la route et la végétation.

L’image rurale de Normandie est illustrée à la figure 3.5. Elle a été prise par un satellite Quickbird. Le capteur fournit 4 bandes chromatiques de résolution spectrale 2,7m. Nous avons, là aussi, considéré un extrait de 900×900 pixels. Nous avons défini 6 classes d’intérêt : la route, la terre battue, les bâches agricoles et 3 classes d’agriculture (végétation à faible, moyenne et haute teneur en chlorophylle).

La table 3.1 présente les évaluations des segmentations obtenues sur le jeu de données urbain. Toutes les régions de référence sont utilisées pour l’apprentissage et pour l’évaluation. Il s’agit donc d’un problème d’optimisation pure et la généralisation des paramètres appris n’est pas évaluée.

Pour *TMA ⊕ OV*, le choix du paramètre est $\alpha = 97.5\%$. Il a été choisi après avoir évalué les critères Janssen et Jaccard. Comme le score de ceux-ci pour la sous-segmentation (*TMA*) est d’environ 97.5%, nous avons choisi de contraindre *TMA ⊕ OV* pour obtenir une valeur similaire. Ainsi, comme l’erreur de sous-segmentation est similaire pour *TMA ⊕ OV*, Jaccard et Janssen, il est possible de les comparer selon la sur-segmentation (critère *OV*).

La figure 3.6 donne les résultats visuels de l’application de l’algorithme sur une partie des jeux de données en utilisant différents critères d’évaluation. Les segmentations obtenues en utilisant



FIG. 3.4 – Image satellite d'un quartier pavillonnaire de Strasbourg et son masque d'exemples fourni par l'expert (4 zones).

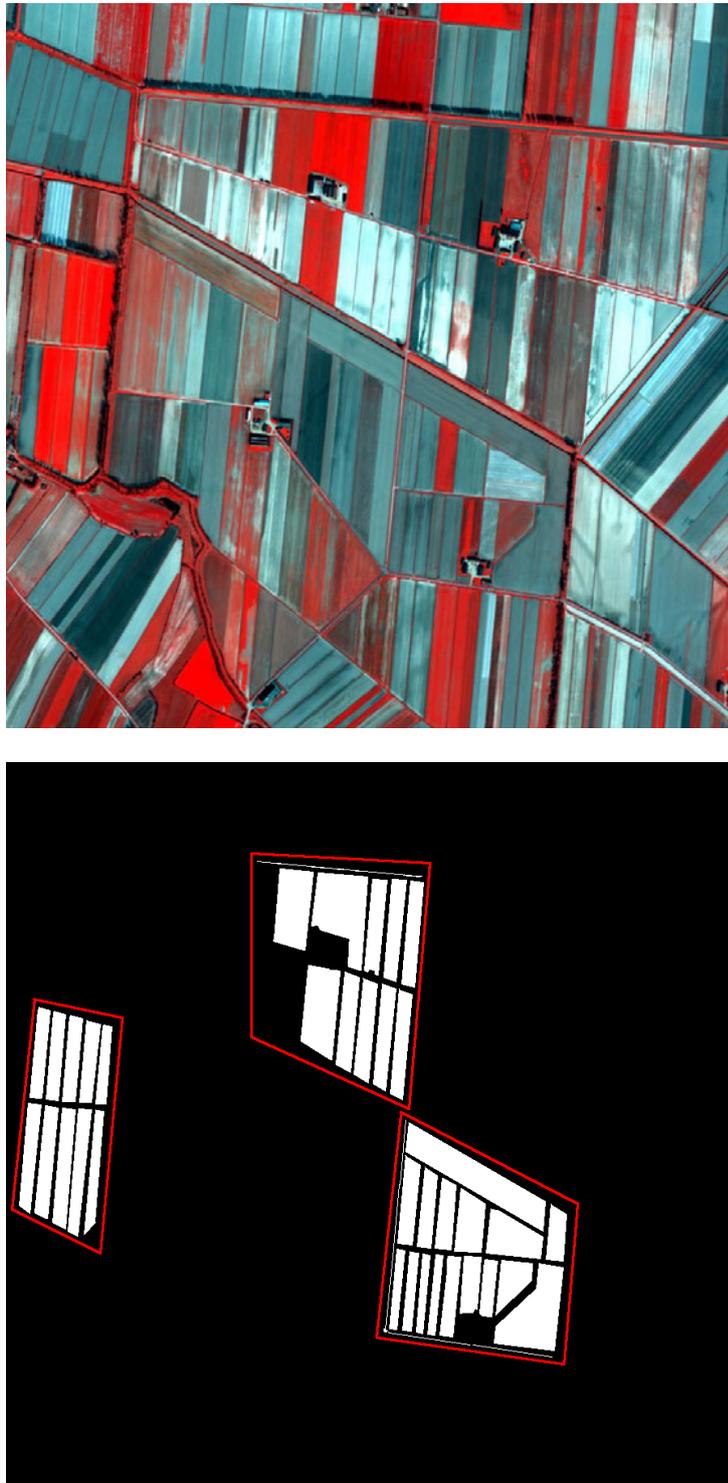


FIG. 3.5 – Image satellite de champs de Normandie (fausses couleurs) et son masque d'exemples fourni par l'expert (3 zones).

le critère OV ne sont pas présentées car elles sont sujettes à une sur-segmentation importante qui rend l'image illisible. Nous pouvons constater qu'une sur-segmentation intervient aussi avec le critère Feitosa. Les critères Janssen et Jaccard donnent des résultats similaires et sont relativement efficaces pour segmenter la route. Le critère $TMA \oplus OV$ (97.5%) produit moins de petites régions, mais sur-segmente la route.

Nous nous sommes ensuite intéressés à la capacité d'apprentissage de l'algorithme, en utilisant un jeu de données différent pour l'apprentissage et l'évaluation. Pour ce faire, nous avons utilisé la technique de la validation croisée. Comme notre jeu de données urbain est composé de quatre zones de régions de référence, nous avons évalué l'algorithme sur chacune de ces zones en apprenant à chaque fois sur une partie des autres zones (une, deux ou les trois zones). La moyenne des quatre évaluations est retenue. Nous avons fait des tests en apprenant sur une, deux ou trois zones. La figure 3.7 illustre les résultats pour chaque couple de critère d'apprentissage et critère d'évaluation en fonction du nombre de zones de régions de référence.

Sur ces graphiques, comme dans le cas de l'optimisation (apprentissage et évaluation sur les mêmes régions de référence), les meilleurs résultats sont obtenus avec un critère d'évaluation quand celui-ci est aussi utilisé pour l'apprentissage. La seule exception est avec le critère de Jaccard, qui obtient la plus mauvaise évaluation de sous-segmentation TMA . Nous pouvons observer que la taille du jeu d'apprentissage a une importance significative sur Janssen, Jaccard et $TMA \oplus OV$. En effet, augmenter le nombre de zones d'apprentissage de une à trois réduit la sous-segmentation de moitié pour Janssen et $TMA \oplus OV$. Janssen montre de très bonnes performances. Il est le meilleur dans l'évaluation sur Jaccard et Janssen, et arrive second (avec $TMA \oplus OV$) sur l'évaluation avec le critère Feitosa.

En conclusion, nous pouvons dire que le critère Janssen semble approprié quand l'utilisateur recherche une correspondance entre les régions de référence et les régions de la segmentation obtenue. A l'inverse, $TMA \oplus OV$ est plus adéquat quand l'utilisateur cherche à obtenir une plus faible sur-segmentation et attache moins d'importance à la correspondance entre les régions.

3.3 Transformation d'espace pour la segmentation

Dans la section précédente, nous avons proposé une méthode de segmentation permettant d'optimiser des paramètres de segmentation. Dans l'évaluation analytique de l'algorithme nous avons vu que les limites de l'algorithme de segmentation de base utilisé ne peuvent être évitées par cette approche. Dans cette section, nous proposons une approche différente, utilisant des méthodes de fouille de données au sein même de l'algorithme de segmentation [27, 26, 28].

Pour introduire cette approche, commençons par rappeler les limites des algorithmes classiques de segmentation. Comme nous l'avons vu, un algorithme de segmentation cherche à créer des regroupements de pixels spatialement connectés selon un critère d'homogénéité. L'hypothèse est donc faite que des pixels connexes ayant des valeurs différentes n'appartiennent pas au même objet. Cette hypothèse est valide pour les images où les objets d'intérêt sont représentées par des pixels similaires. Pour des images complexes, cette hypothèse n'est plus valide. Un objet peut être très hétérogène ; si l'on prend le cas d'un toit de maison, celui-ci peut avoir des fenêtres (de couleur différente au reste du toit) et une illumination différente de chaque côté du toit selon l'exposition au soleil. L'hypothèse faite n'est pas valide dans le cas de ces images et les résultats ne sont donc pas pertinents.

Nous proposons une nouvelle méthode de segmentation qui utilise des régions de référence pour apprendre une nouvelle notion de pixels homogènes. Pour ce faire, nous projetons les données à segmenter dans un espace de représentation plus adéquat où l'hypothèse de similarité est valide.

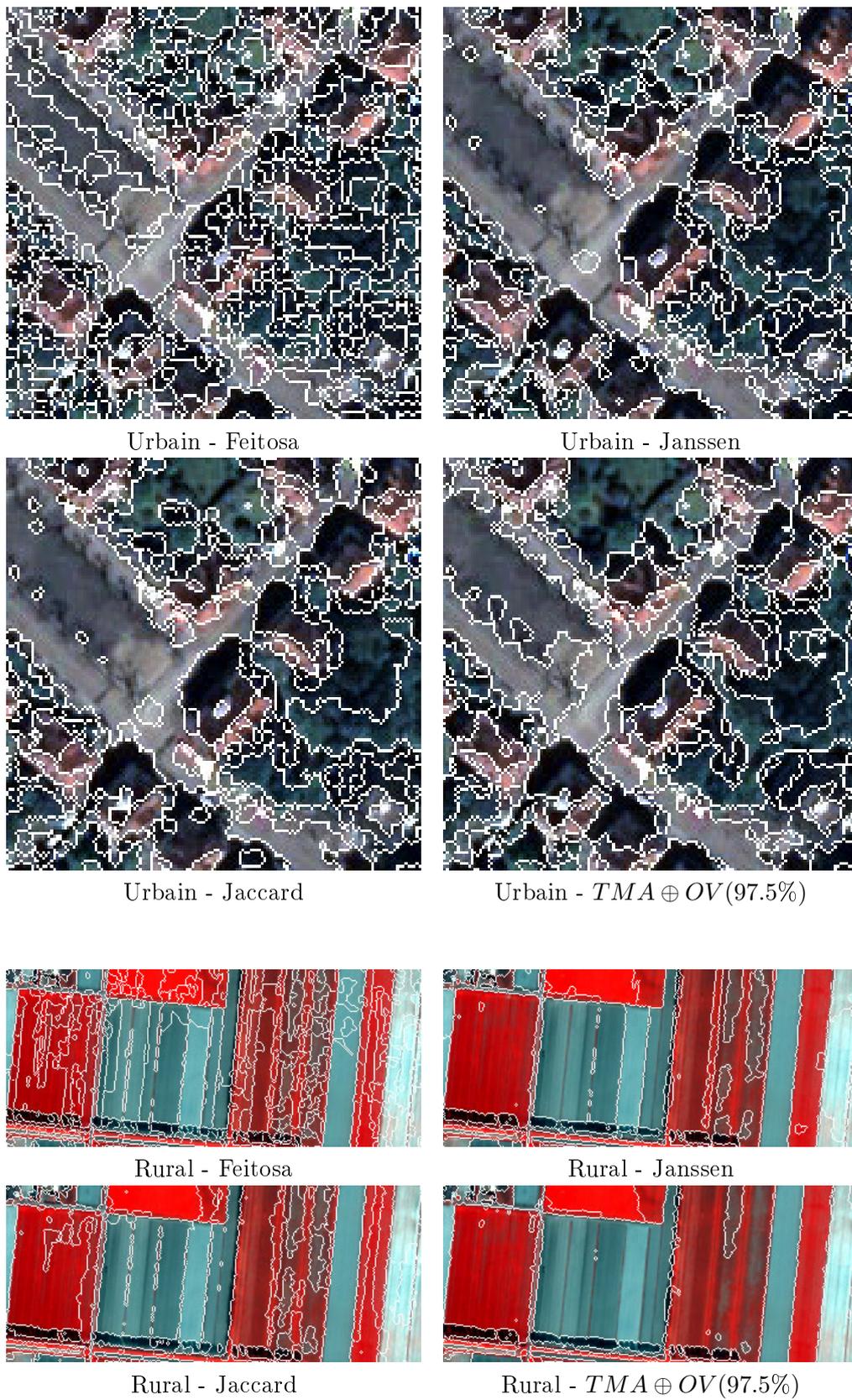


FIG. 3.6 – Segmentations obtenues par l’algorithme proposé sur une image urbaine et une image rurale.

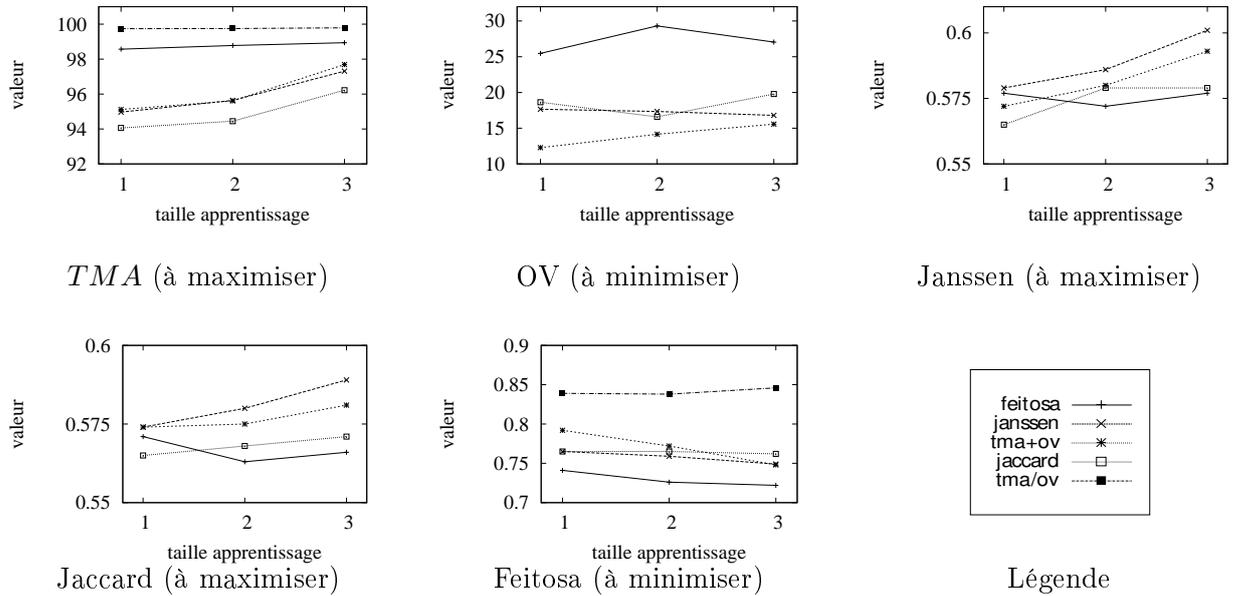


FIG. 3.7 – Moyenne des résultats pour chaque critère en évaluation et en apprentissage en fonction du jeu d'apprentissage. Pour *OV*, Janssen et Jaccard, le critère d'apprentissage *TMA/OV* n'est pas présenté car il est beaucoup plus faible que les autres.

Dans l'espace ainsi créé, les algorithmes classiques de segmentation peuvent être appliqués et doivent fournir des résultats plus proches de ceux escomptés.

Nous allons tout d'abord donner un état de l'art des méthodes pouvant se rapprocher de notre proposition et nous positionner par rapport à celles-ci. Ensuite, nous expliciterons le fonctionnement de l'algorithme avant de l'évaluer.

3.3.1 État de l'art et positionnement

Nous faisons ici un état de l'art regroupant plusieurs méthodes qui se proposent d'introduire des connaissances *via* des régions de références dans un algorithme de segmentation. Dans l'algorithme de segmentation par ligne de partage des eaux, la connaissance est souvent introduite en utilisant des marqueurs [74, 88] et/ou en appliquant la ligne de partage des eaux sur une image modifiée. Les marqueurs modifient légèrement le fonctionnement de la ligne de partage des eaux. Au lieu d'être inondée à partir de ses minima locaux, la surface topographique sera inondée uniquement à partir des marqueurs.

Haker et al. [54] utilisent des images préalablement segmentées afin d'extraire les probabilités *a priori* de l'appartenance aux classes d'intérêt et les combinent selon la règle de Bayes. D'autres connaissances peuvent être intégrées de cette façon comme les positions spatiales approximatives des objets d'intérêt. Si la position d'un objet est connue, il est possible d'augmenter la probabilité *a priori* de la classe de l'objet recherché aux positions où l'objet est attendu. Cette méthode est similaire à une classification supervisée et donc sujette à la sous-segmentation, mais permet d'améliorer les résultats quand l'utilisateur sait déterminer approximativement la localisation des différents objets.

Levner et Zhang [76] proposent une méthode utilisant des cartes de probabilité. Ils utilisent

une première classification, basée sur des régions de référence érodées pour trouver des marqueurs. Une autre classification est effectuée en utilisant les régions de référence non modifiées fournissant une carte de probabilités. L'inverse de cette carte de probabilités est utilisée comme une surface topographique pour appliquer la ligne de partage des eaux en utilisant les marqueurs obtenus précédemment. Cette méthode n'est appliquée que pour de la classification binaire (une classe d'intérêt et un fond). Elle fait l'hypothèse qu'un marqueur sera détecté pour chaque objet d'intérêt. Si aucun marqueur n'est détecté pour un objet, il ne pourra être segmenté.

Grau et al. [51] utilisent une carte de probabilité pour chaque classe d'intérêt. Des marqueurs sont générés en utilisant un atlas. Chaque marqueur a une classe associée. Une approche par croissance de région est utilisée pour simuler le processus d'inondation de la surface topographique. L'élévation entre deux pixels est dépendante du marqueur courant. L'approche utilise la différence de probabilité de ces pixels dans la carte de probabilités associée à la classe du marqueur (c'est un processus markovien). Cette approche nécessite de connaître la position des marqueurs.

D'autres formes d'utilisation de connaissances dans le processus de segmentation ont été proposées. Hamarneh et Li [55] effectuent une ligne de partage des eaux classique. Cette première segmentation est donc affectée par le problème de sur-segmentation inhérent à la ligne de partage des eaux. Ils utilisent ensuite un algorithme de classification non supervisée de type k-means regroupant les régions à partir de leur intensité et leur position. En utilisant des connaissances sur l'apparence des objets recherchés, ils sélectionnent le regroupement de régions approprié et alignent un histogramme de forme pour ôter les régions inadéquates. Cette approche repose fortement sur l'hypothèse que les objets recherchés ont des valeurs d'intensité homogènes.

Chen et al. [18] extraient un modèle de forme et d'intensité de l'objet d'intérêt à partir d'un ensemble de régions de référence. Après cette étape d'apprentissage, ils utilisent un modèle de contour actif pour segmenter l'objet recherché en se basant sur le modèle de forme et d'intensité défini précédemment. Cette méthode n'est applicable que pour les problèmes de détection d'un seul objet dont la localisation approximative est connue.

De cet état de l'art, nous pouvons noter que l'idée de l'introduction de connaissances dans le processus de segmentation a donné lieu récemment à plusieurs propositions. La table 3.2 rassemble les connaissances nécessaires au fonctionnement des méthodes de la littérature que nous venons de voir. Nous pouvons constater que toutes ces méthodes (hormis la classification supervisée) soit nécessitent de connaître la localisation des objets, soit ne peuvent segmenter qu'un seul objet (ou une classe d'objets).

L'approche que nous proposons diffère de ces approches sur plusieurs points :

- la capacité à prendre en compte plusieurs classes d'objets ;
- aucune connaissance n'est nécessaire sur la localisation des objets.

3.3.2 Présentation de l'algorithme

Une représentation graphique de l'algorithme est donnée à la figure 3.8. L'algorithme que nous proposons fonctionne en deux étapes. La première étape consiste à transformer l'espace des données de l'image d'entrée. Nous effectuons cette étape par une classification floue supervisée afin d'introduire de la sémantique dans la représentation des données. Cet algorithme utilise les pixels des régions de référence données et affecte à chaque pixel de l'image une probabilité d'appartenir à chacune des classes possibles. La seconde étape est la segmentation de ces cartes d'appartenances. Nous nous concentrerons ici sur la première étape, la seconde étant classique et exposée à l'annexe A.

Afin d'expliquer le paradigme utilisé dans cet algorithme, considérons que nous disposons

TAB. 3.2 – Intégration de connaissances des différentes méthodes.

Méthode	Caractéristiques des pixels	Localisation	Forme	1 objet
Classification supervisée	X			
Ligne de partage des eaux avec marqueurs [88]		X		
Haker et al. [54]	X	X		
Levner et Zhang [76]	X			1 classe d'objets
Li et Hamarneh [77]	X		X	X
Chen et al. [18]	X		X	X
Grau et al. [51]	X	X		

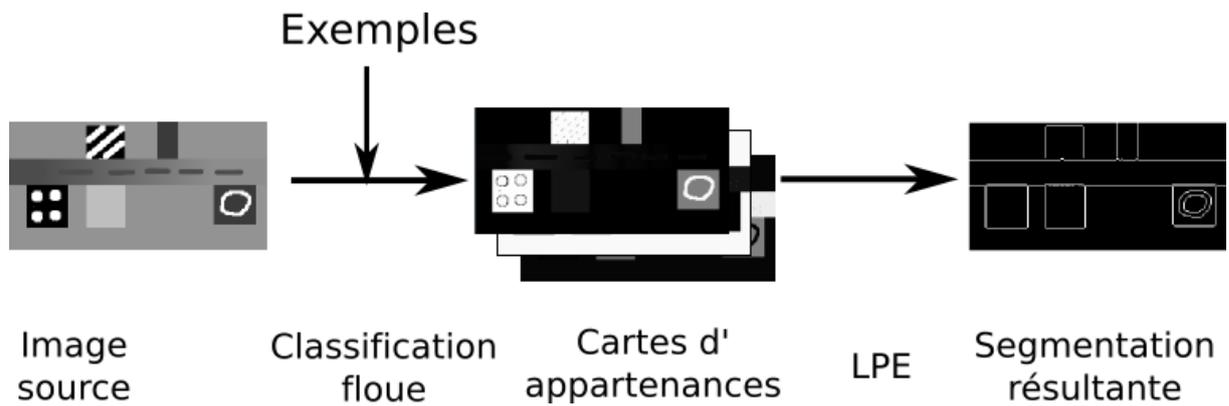


FIG. 3.8 – Processus de segmentation supervisée, classification floue puis application de la ligne de partage des eaux.

d'un classifieur parfait (c'est-à-dire qui ne fait pas d'erreur). Une classe est assignée à chaque pixel en entrée. Ainsi, une région est créée à partir de chaque composante connexe de pixels avec des étiquettes identiques. Il n'y a pas de sous-segmentation, puisque deux pixels de classes différentes ne peuvent être regroupés ensemble. Il n'y a pas de sur-segmentation non plus car deux pixels connexes de la même classe sont toujours regroupés ensemble.

Soit S_i l'espace d'entrée défini par l'équation 3.20 (de dimension $\Omega(i)$). Il n'existe pas de fonction qui pour un pixel de S_i affecte une classe à ce pixel sans jamais faire d'erreur. Comme il n'existe que des fonctions approximatives, soit S_m l'espace des appartenances défini par l'équation 3.21. Dans l'espace des appartenances, chaque classe d'objets (soit $\Omega(\mathcal{C})$ le nombre de classes) contenue dans l'image est une dimension. La valeur dans chaque dimension dénote l'appartenance du pixel à la classe d'objets correspondante.

$$\begin{aligned} S_i : I &\rightarrow \mathbb{R}^{\Omega(i)} \\ p &\mapsto S_i(p) \quad \text{avec } S_i(p) \text{ la valeur} \\ &\quad \text{spectrale du pixel } p \end{aligned} \quad (3.20)$$

$$\begin{aligned} S_m : I &\rightarrow [0; 1]^{\Omega(\mathcal{C})} \\ p &\mapsto S_m(p) \quad \text{avec } S_m(p) \text{ le vecteur} \\ &\quad \text{d'appartenances du pixel } p \end{aligned} \quad (3.21)$$

La transformation de S_i à S_m est apprise en utilisant des outils de fouille de données capables d'apprendre cette fonction à partir des pixels de référence. Dans notre cas, nous utilisons l'algorithme des k plus proches voisins [1]. Pour chaque pixel d'entrée p , les k (avec $k = 5$) plus proches voisins dans l'espace S_i sont sélectionnés. Chaque pixel voisin p_n augmente le niveau d'appartenance du pixel p pour la classe à laquelle il appartient, pondéré par l'inverse de la distance $d(p, p_n)$ dans l'espace des données. L'appartenance du pixel p à la classe C_i , notée m_{p, C_i} , est obtenue par :

$$m_{p, C_i} = \left(\sum_{n=1}^k \sum_{l=1}^{\Omega(\mathcal{C})} w_{n, l} \right)^{-1} \sum_{n=1}^N w_{n, C_i} \quad (3.22)$$

où $w_{n, C_i} = \begin{cases} d(p, p_n)^{-1} & \text{si } p_n \text{ est étiqueté par } C_i \\ 0 & \text{sinon} \end{cases}$

Les données des pixels projetés dans l'espace S_m peuvent être considérées comme des images multi-valuées. En effet, chaque pixel est associé à un vecteur d'appartenances aux classes. On peut donc créer une image où chaque bande contient les appartenances de chaque pixel à une classe donnée.

Cette image ainsi obtenue peut donc être segmentée par des techniques classiques comme la ligne de partage des eaux comme l'illustre la figure 3.8. Nous allons à présent évaluer cette méthode.

3.3.3 Évaluation

Pour évaluer l'algorithme proposé, nous commencerons par une évaluation analytique sur les pré-requis de l'algorithme et ses limites, puis nous présenterons une application sur un exemple synthétique. Nous ne proposons pas ici d'évaluation sur une image réelle. En effet les images réelles demandant l'utilisation de paramètres dans l'algorithme de segmentation pour obtenir de bons résultats (i.e. éviter la sur-segmentation inhérente à l'algorithme de ligne de partage des eaux sans paramètres), nous présenterons ceux-ci à la section 3.4 qui combinera cette technique et la méthode d'apprentissage de paramètres.

Évaluation analytique

L'algorithme proposé nécessite de fournir des connaissances afin d'être en mesure de segmenter l'image :

- Définition des classes : l'utilisateur doit connaître les classes d'objets qu'il recherche dans l'image ;
- Exemples de chaque classe : ils sont nécessaires pour l'étape de classification floue. Il n'y a besoin que de pixels représentatifs des classes recherchées et pas nécessairement de régions entières.

Les limites de l'algorithme sont les suivantes :

- Objets connexes de la même classe : si deux objets de la même classes sont spatialement connexes et ont des valeurs d'appartenance aux classes similaires, ces objets seront fusionnés.
- Objets ayant des valeurs d'appartenance hétérogènes : Dans ce cas, l'algorithme produira des zones sur-segmentées. Par exemple, soit un objet composé des valeurs spectrales v_1 et v_2 . Si v_1 est facilement identifiable comme appartenant à une des classes tandis que v_2 est plus difficilement identifiable et est défini comme pouvant appartenir à plusieurs des classes, alors l'objet sera sur-segmenté.

Enfin, il convient de noter certaines spécificités de cet algorithme :

- Méthode de segmentation utilisée : Le cœur de l'algorithme est d'effectuer une transformation d'espace avant la segmentation proprement dite. Il est donc possible d'utiliser une autre méthode de segmentation que celle qui a été utilisée ici.
- Méthode d'apprentissage de tranformation d'espace utilisée : Nous utilisons ici la méthode des k plus proches voisins. Il est possible d'utiliser d'autres méthodes de classification tant qu'elles peuvent fournir des appartenances aux classes. Néanmoins, nous avons constaté que l'algorithme des k plus proches voisins convient très bien à ce type de problème (attributs numériques, espace des données complexe).
- Complexité : Elle est égale à la somme du temps d'exécution de la tranformation d'espace et de celui de la segmentation. L'apprentissage de la transformation est rapide, il consiste à stocker les exemples d'apprentissage. L'application de la transformation est de l'ordre de $N \times K$ avec N le nombre de pixels dans l'image à transformer et K le nombre de pixels dans le jeu d'apprentissage. Il est possible de descendre à $N \times \log(K)$ en utilisant des structures de données optimisées comme les cover trees [8]. Il est aussi possible de n'utiliser qu'un échantillon des pixels d'apprentissage. Nous avons ainsi limité le nombre d'exemples de chaque classe d'apprentissage à 500 par tirage aléatoire.

Évaluation sur un exemple synthétique

Nous évaluons l'algorithme proposé sur un exemple synthétique afin d'illustrer son fonctionnement et ses différences par rapport à d'autres méthodes. Nous présenterons à la section suivante, qui traite d'une extension de cet algorithme, une évaluation sur des données de télédétection.

L'exemple synthétique, donné à la figure 3.9 (a), représente schématiquement une image de télédétection avec une route et quelques bâtiments. La figure 3.9 (b) montre l'objectif à atteindre dans la segmentation et identifie chaque bâtiment avec une étiquette. Une dizaine de pixels ont été étiquetés pour chaque classe. Aucun exemple n'a été pris pour les bâtiments B et C.

La première segmentation (figure 3.9 (c)) est obtenue avec une ligne de partage des eaux classique. Elle est sujette à de nombreux problèmes de sur-segmentation. L'application de méthodes de réduction de la sur-segmentation conduit au phénomène de sous-segmentation. La figure 3.9

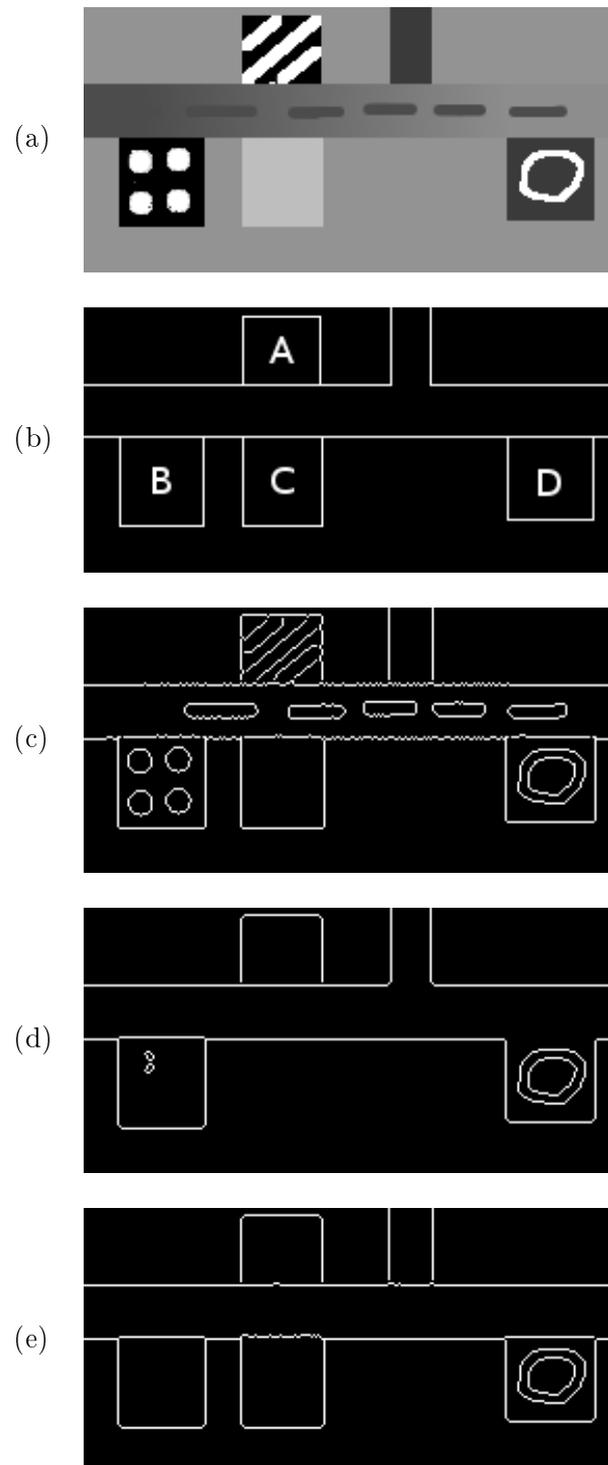


FIG. 3.9 – Application sur une image synthétique. Image originale (a). Segmentation manuelle et étiquetage des bâtiments (b). Segmentation par une ligne de partage des eaux classique (c). Segmentation par classification supervisée et filtrage (d). Résultats obtenus par l’algorithme présenté (e).

(d) illustre le type de résultat que l'on peut obtenir avec des méthodes de classification supervisée comme la méthode des k plus proches voisins (et donc utilisant les exemples). Des problèmes de sous-segmentation interviennent sur une partie du bâtiment D qui est fusionnée avec la route et sur le bâtiment C qui est fusionné avec la végétation. Avec la méthode proposée 3.9 (e), il n'y a pas de problème de sous-segmentation. Il reste deux zones de sur-segmentation : la route est divisée en deux parties et le bâtiment D est divisé en trois parties. Néanmoins, cette méthode donne les résultats les plus probants.

À présent, nous allons étudier comment il est possible d'améliorer cet algorithme en utilisant la méthode d'apprentissage de paramètres de segmentation qui a été vue à la section précédente.

3.4 Approche hybride

Nous avons vu une méthode d'apprentissage des paramètres à la section 3.2 et une méthode de segmentation supervisée à la section 3.3. Dans cette section, nous présentons une combinaison de ces deux méthodes afin de combiner les avantages de chacune d'entre elles.

3.4.1 Présentation de l'algorithme

La méthode [25] est constituée de deux phases comme illustré à la figure 3.10. Dans la première phase, l'algorithme apprend comment segmenter une image en utilisant un jeu d'apprentissage. Le processus d'apprentissage intervient en deux endroits, pour la transformation d'espace et pour le choix des paramètres. Une fois l'apprentissage terminé, l'algorithme de segmentation ainsi créé (composé de l'étape de transformation d'espace et de l'algorithme de segmentation de base) peut être utilisé pour segmenter des images. Il n'y a plus besoin d'exemples d'apprentissage à cette phase. De plus, la méthode ne nécessite aucun paramètre dans les deux phases.

Nous avons déjà vu la méthode de transformation d'espace à la section 3.3. Cette méthode transforme l'espace des valeurs brutes de l'image en un espace d'appartenances aux classes plus sémantique. L'utilisation de la méthode d'apprentissage de paramètres présentée à la section 3.2 va permettre d'introduire une nouvelle étape. En effet, au lieu d'inférer les appartenances des pixels aux classe à partir des valeurs brutes des pixels, nous allons utiliser les valeurs brutes complétées par de nouveaux attributs. Ayant plus d'informations, notre objectif est d'avoir des appartenances aux classes plus précises. Nous introduisons donc l'espace S_e défini par :

$$\begin{aligned} S_e : \mathbb{R}^{\Omega(i)} &\rightarrow \mathbb{R}^{\Omega(e)} \\ S_i(p) &\mapsto S_e(p) \quad \text{avec } S_e(p) \text{ le vecteur} \\ &\quad \text{étendu des attributs du pixel } p \end{aligned} \quad (3.23)$$

En plus des valeurs brutes des pixels de l'image, l'espace S_e contient les attributs suivants :

- intensité relative : chaque bande c , prend la valeur de celle-ci divisée par la somme des valeurs des bandes ;
- intensité moyenne : c'est la moyenne de la valeur des bandes ;
- analyse en composante principale : l'ACP est une méthode qui recherche les directions de l'espace qui représentent le mieux les corrélations [124].

Le problème d'une telle approche tient à la méthode d'apprentissage que nous avons utilisée, celle des k plus proches voisins. Cette méthode a pour défaut d'affecter une importance identique à tous les attributs caractérisant les pixels. Cela pose un problème quand on augmente le nombre d'attributs. En effet, certains attributs sont moins importants que d'autres. Certains peuvent n'être que du bruit (pour le problème donné) ou plusieurs attributs peuvent être corrélés [12].

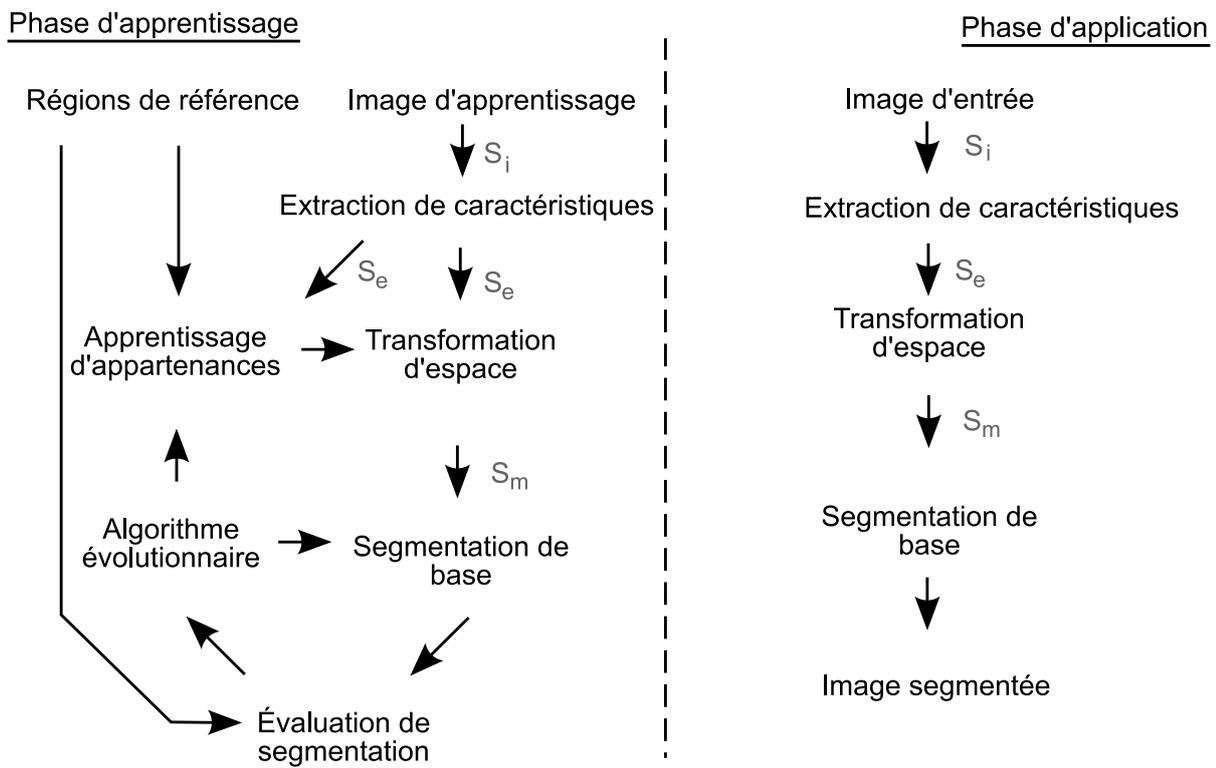


FIG. 3.10 – Diagramme de flux de l'algorithme proposé pour la phase d'apprentissage et la phase d'application.

Pour éviter ce type de problèmes, l'étape d'*extraction de caractéristiques* considère aussi un vecteur de poids, un poids pour chaque dimension de l'espace S_e , afin de pondérer les attributs (par multiplication de la valeur de l'attribut par la pondération). Ainsi, les attributs n'apportant pas d'informations pertinentes pourront voir leur pondération réduite et donc ne pas influencer le résultat. La modification de la fonction de distance permet d'accroître la précision de la méthode des k plus proches voisins [100]. Le nombre d'attributs (nombre de dimensions de S_e) étant $(3 \times B) + 1$, avec B le nombre de bandes de l'image, il faut donc un nombre de poids identique.

Pour l'algorithme de segmentation de base, nous utilisons le même algorithme évolutionnaire que celui présenté à la section 3.2, c'est-à-dire une ligne de partage des eaux suivie d'une étape de fusion de régions. Cet algorithme de segmentation prend trois paramètres. La différence est que le nombre de paramètres à optimiser passe de 3 à $(3 \times B) + 4$ (dont $(3 \times B) + 1$ pour la pondération des attributs et 3 pour l'étape de segmentation).

3.4.2 Évaluation

Pour évaluer l'algorithme proposé, nous commencerons par une évaluation analytique sur les pré-requis de l'algorithme et ses limites. Une partie de l'évaluation analytique sera reprise des algorithmes précédent, l'algorithme proposé ici étant basé sur ceux-ci. Nous présenterons ensuite une application sur des images de télédétection.

Évaluation analytique

L'algorithme proposé nécessite de fournir des connaissances afin d'être en mesure de segmenter l'image :

- Régions de références : elles sont nécessaires au fonctionnement de l'algorithme. Selon le critère utilisé, l'algorithme peut se contenter de régions isolées (critères Feitosa, Jansen et Jaccard) ou bien nécessiter des parties de segmentations complètes sur une zone (critères OV et TMA).
- Critère d'évaluation : pour fonctionner, l'algorithme a besoin d'un critère d'évaluation. Pour obtenir de bons résultats, l'utilisateur doit être en mesure de choisir un critère d'évaluation qui correspond le mieux à ses objectifs (notamment sur l'arbitrage entre sur-segmentation et sous-segmentation).

Les limites de l'algorithme sont les suivantes :

- Objets connexes de la même classe : si deux objets de la même classe sont spatialement connexes et ont des valeurs d'appartenance aux classes similaires, ces objets seront fusionnés.
- Objets ayant des valeurs d'appartenances et des valeurs spectrales hétérogènes : dans ce cas, l'algorithme produira des zones sur-segmentées. dans les méthodes classiques il suffit qu'un objet ait des pixels de valeurs hétérogènes pour qu'il soit sur-segmenté, il s'agit donc d'une limite moins forte que celle des algorithmes classiques.

Enfin, il convient de noter certaines spécificités de cet algorithme :

- Méthode de segmentation utilisée : le cœur de l'algorithme est d'effectuer une transformation d'espace avant la segmentation proprement dite. Il est donc possible d'utiliser une autre méthode de segmentation que celle qui a été utilisée ici.
- Méthode d'apprentissage de transformation d'espace utilisée : nous utilisons ici la méthode des k plus proches voisins. Il est possible d'utiliser d'autres méthodes de classification tant qu'elle peuvent fournir des appartenances aux classes.

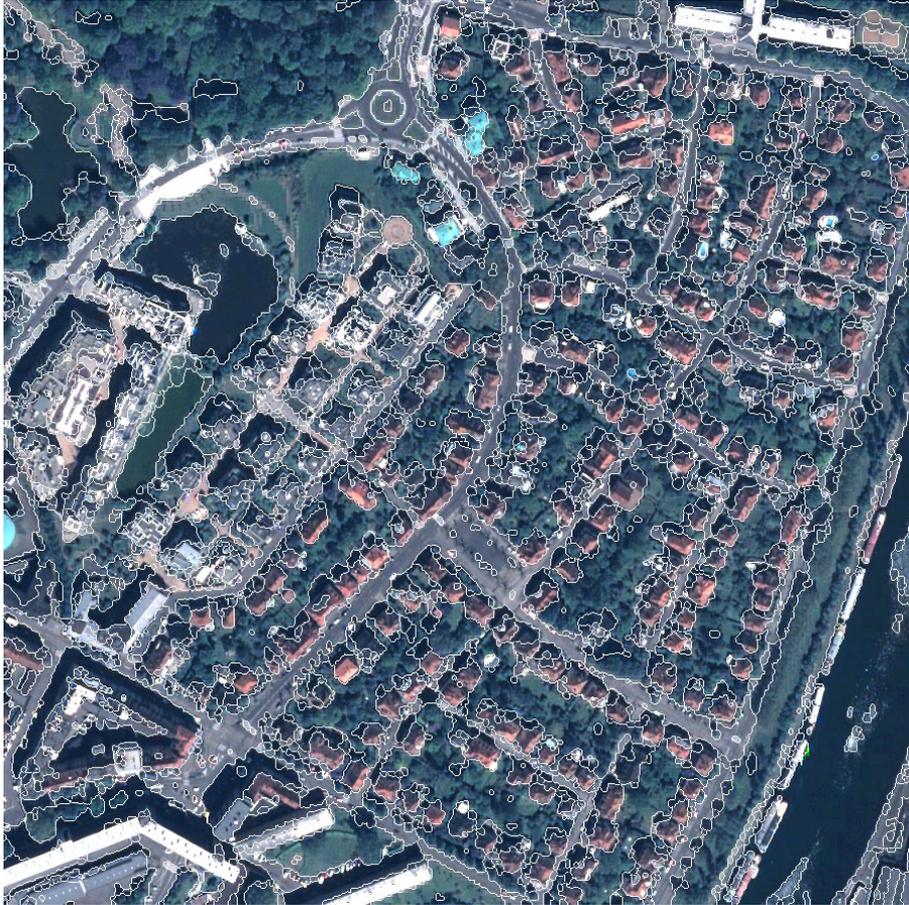


FIG. 3.11 – Segmentation obtenue par l’algorithme proposé sur un extrait du jeu de données urbain.

- Complexité : Comme nous l’avons déjà indiqué, le temps d’exécution de l’algorithme d’apprentissage des paramètres est $\Omega(\mathcal{P}) \times (N_e + 1) \times (T_s + T_e)$ avec T_s le temps d’exécution de la méthode de segmentation, T_e le temps d’exécution de l’évaluation, $\Omega(\mathcal{P})$ le nombre d’individus par générations et N_e le nombre d’évolutions. Ici T_s vaut $N_p \times K$ avec N_p le nombre de pixels dans l’image à transformer et K le nombre de pixels dans le jeu d’apprentissage.

Évaluation empirique

Nous avons appliqué cette méthode sur le jeu de données urbain. La segmentation obtenue en utilisant les quatre zones de référence pour l’apprentissage est illustrée à la figure 3.11. Le critère à optimiser choisi est Janssen. Nous constatons que les objets de chacune des trois classes (route, maison et végétation) sont relativement bien segmentés. Par exemple, sur les maisons, il n’y a plus la séparation habituelle entre la partie du toit à l’ombre et la partie plus exposée au soleil.

De même, nous avons appliqué l’algorithme en utilisant deux zones de référence pour l’apprentissage et deux zones pour l’évaluation. Nous avons ainsi pu évaluer le résultat produit avec des critères d’évaluation (voir table 3.3). Les résultats obtenus sont meilleurs qu’avec les autres approches proposées dans ce chapitre.

TAB. 3.3 – Résultat d'évaluation de segmentations obtenues par l'algorithme proposé selon le critère d'évaluation utilisé pour l'optimisation.

Opt. \ Éval.	TMA(%)	OV	Feitosa	Janssen	Jaccard
Janssen	98,81	4,08	0,847	0,654	0,588
$TMA \oplus OV$	97,11	2,61	0,878	0,627	0,581

Enfin, la figure 3.12 illustre la contribution de l'apprentissage des paramètres et de la transformation d'espace dans l'obtention du résultat final. Nous constatons que la ligne de partage des eaux, avec pour paramètres des valeurs nulles, produit une sur-segmentation importante. L'apprentissage des paramètres permet de réduire cette sur-segmentation, mais les objets spectralement hétérogènes restent sur-segmentés. L'utilisation de la transformation d'espace permet de pallier à ce problème. Il reste néanmoins de nombreuses zones de sur-segmentation (zones de pixels mixtes principalement). Après l'ajout de l'apprentissage de paramètres, ces zones sont fusionnées.

3.5 Conclusion

Au cours de ce chapitre, nous avons étudié deux méthodes pour introduire de la connaissance dans le processus de segmentation.

La première, l'apprentissage de paramètres de segmentation, avait déjà été abordée dans d'autres travaux. Nous avons néanmoins apporté deux contributions. Premièrement, nous avons introduit et comparé différents critères d'évaluation qui peuvent être utilisés dans l'optimisation des paramètres. Deuxièmement, nous avons analysé les capacités d'apprentissage de l'algorithme proposé, c'est-à-dire l'importance de la quantité de régions de référence sur le résultat et sur la généralisation des résultats. Nous avons donc pu voir que même sur des zones qui n'étaient pas dans le jeu d'apprentissage, les résultats étaient concluants.

La seconde méthode vue dans ce chapitre, la segmentation par transformation d'espace, propose un concept original. Il s'agit de faire une classification floue avant l'étape de segmentation. Nous avons remarqué que cette méthode donne de bons résultats sur des images complexes où les méthodes classiques éprouvent des difficultés. Cette méthode a la particularité de pouvoir être utilisée avec n'importe quelle méthode de segmentation déjà existante. Comparée aux autres méthodes de segmentation permettant l'introduction de connaissances, celle-ci ne requiert pas la localisation des objets à segmenter et n'est pas limitée à la segmentation d'un seul objet.

Nous avons choisi de combiner ces deux méthodes. L'algorithme ainsi généré utilise les connaissances à disposition selon deux méthodes complémentaires. Les résultats sont suffisamment probants pour montrer la pertinence de l'utilisation de connaissances dans le processus de segmentation.

Au delà des méthodes proposés dans cette thèse, il nous semble évident que d'autres méthodes d'intégration de connaissances dans le processus de segmentation restent à explorer comme l'utilisation d'algorithme évolutionnaire dans la segmentation. Par exemple, en utilisant la programmation génétique pour induire un programme complet de segmentation optimisé pour le problème particulier à résoudre. Les premiers travaux en ce sens [116] bien que présentant des résultats mitigés montrent la faisabilité d'une telle approche.

Mais la segmentation est rarement le résultat final désiré par l'utilisateur. En général cette



FIG. 3.12 – Segmentations obtenues par une ligne de partage des eaux avec pour paramètres des valeurs nulles (a), ligne de partage des eaux avec apprentissage de paramètres (b), ligne de partage des eaux avec transformation d'espace (c) et enfin combinaison de l'apprentissage des paramètres et transformation d'espace.

étape est suivie d'une classification des régions ainsi obtenues, cette tâche va à présent être abordée.

Chapitre 4

Classification : Identification des régions

Nous classifions trop et ne jouissons pas assez.
Okakura Kakuzo

4.1 Introduction

La classification est une étape postérieure à la segmentation. Elle consiste à affecter une étiquette de classe à chacune des régions issues de l'étape de segmentation. Il a été démontré qu'affecter des étiquettes à des régions est plus intéressant que de travailler directement à partir des pixels, notamment pour l'imagerie satellitaire à très haute résolution [20, 49, 126, 79]. Ce chapitre présente nos travaux sur cette partie du processus d'interprétation d'images.

Dans cette section, nous présenterons la problématique de la classification de régions issues d'une segmentation. Nous aborderons ensuite la définition des différents attributs permettant de caractériser les régions de la segmentation. Nous introduirons aussi des mesures permettant d'évaluer la qualité des classifications obtenues. Enfin, nous décrirons les méthodes de classification qui seront présentées dans les sections suivantes.

4.1.1 Problématique

Soit \mathcal{X} l'espace des objets à classer (dans notre cas des régions issues de la segmentation) et \mathbb{C} l'ensemble des classes possibles. L'objectif est de créer une fonction c , qui, à un objet x de \mathcal{X} , associe une classe C_x de \mathbb{C} . Plus formellement :

$$\begin{aligned} c: \mathcal{X} &\rightarrow \mathbb{C} \\ f(x) &\mapsto C_x \end{aligned} \tag{4.1}$$

Il existe deux approches classiques pour construire cette fonction c . La construction par un expert et l'inférence à partir d'exemples.

La création par un expert, approche utilisée dans les systèmes experts, fonctionne de la manière suivante : l'expert formalise ses connaissances du domaine sous forme de règles, d'ontologies [125], de système à base de frame [90]. Ces connaissances sont ensuite opérationnalisées afin de créer un modèle de classification de régions à partir de ces connaissances.

A contrario, l'approche à partir d'exemples se base sur des objets déjà classés pour inférer la fonction c : systèmes à base de règles (induites des exemples), arbres de décision, réseaux de

neurones, approches par prototypes, . . . Le lecteur intéressé par un descriptif plus exhaustif des algorithmes d'apprentissage à partir d'exemples peut se référer à [91, 127].

Dans les deux cas, la classification se base sur un ensemble d'attributs permettant de caractériser les régions.

4.1.2 Caractérisation des régions

Pour classer les régions, il faut définir un formalisme permettant de les décrire. Initialement, une région est définie par la composition d'un certain nombre de pixels qui forment une composante connexe. Chacun de ces pixels est défini par une position dans l'image (ou une position géolocalisée) et un vecteur de valeurs qui dénote l'intensité de la réponse spectrale du pixel dans différentes longueurs d'onde du spectre. Cette représentation des régions utilise un formalisme multi-relationnel, c'est-à-dire un formalisme où plusieurs types d'entités (régions, pixels) existent et sont reliées entre elles par une relation *un à plusieurs* (une région comporte un ensemble de pixels). Il peut aussi exister une relation *un à plusieurs* entre les régions pour modéliser l'adjacence. Même si des algorithmes de classification capables de traiter ce type de formalisme existent [36], ils n'ont (à notre connaissance) jamais été utilisés pour de la classification de régions.

Une autre approche consiste à propositionnaliser (passage d'un formalisme multi-relationnel à un formalisme attribut-valeur) les données afin de n'avoir que des attributs décrivant l'entité région. Deux familles de propositionnalisation existent : la sélection [71, 68] (par exemple un attribut booléen *existe-t-il un pixel dont l'intensité est inférieure à 0.1 ?*) et l'agrégation [102, 70] (par exemple la surface qui compte le nombre de pixels de la région). La propositionnalisation peut s'avérer être un exercice difficile [72] et entraîne généralement une perte d'information.

Dans notre cas, la classification concerne des régions de segmentation d'images de télédétection à très haute résolution. Nous utilisons principalement l'agrégation pour définir les propriétés internes des régions, et la sélection pour définir des propriétés d'adjacences entre les régions. En effet, les caractéristiques propres d'un pixel d'une région ont peu d'intérêt, a priori, pour classer des régions. Nous utilisons aussi des attributs qui ne sont pas issus d'agrégation ou de sélection, mais qui sont spécifiques au domaine. Par exemple, nous construisons l'attribut *élongation* qui permet de mesurer l'élongation d'une région.

L'ensemble des attributs ainsi utilisés peut se décomposer en trois types d'attributs :

- attributs spectraux : regroupe des attributs statistiques sur les valeurs spectrales des pixels (la moyenne et l'écart-type sur chacune des bandes de l'image) ;
- attributs spatiaux : constitué d'attributs caractérisant la forme de la région (élongation par exemple), sa taille, . . . ;
- attributs contextuels : permet de caractériser l'environnement de la région (présence d'objets d'une classe particulière à proximité, . . .).

Nous allons à présent détailler ces différents attributs.

Attributs spectraux

Les attributs spectraux calculés pour les régions sont constitués des moyennes et écart-types de valeurs des pixels composant la région pour chacune des bandes suivantes :

- bandes présentes initialement dans l'image ;
- bande d'intensité : moyenne des valeurs des pixels des bandes initiales de l'image ;
- bandes de ratio : il s'agit de chacune des bandes initiales normalisées afin que la somme des valeurs des bandes de ratio soit égale à 1.

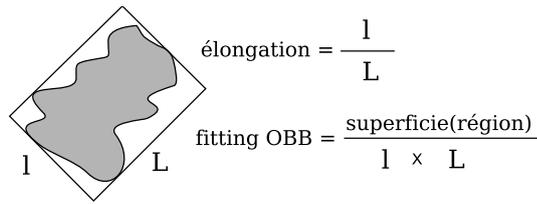


FIG. 4.1 – Illustration des attributs spatiaux élongation et fitting OBB.

- bande NDVI : Le NDVI (Normalized Difference Vegetation Index) est un indice couramment utilisé en géomatique, il se calcule avec la bande rouge (*RED*) et proche infra-rouge (*NIR*) de l'image initiale. La formule est donnée par :

$$NDVI = \frac{(NIR - RED)}{NIR + RED} \quad (4.2)$$

Pour une image qui contient initialement quatre bandes spectrales, on obtient donc dix bandes où sont calculées la moyenne et l'écart-type, ce qui correspond à vingt attributs.

Attributs spatiaux

Nous calculons trois attributs spatiaux :

- superficie : surface couverte par la région (nombre de pixels) ;
- élongation : rapport de la largeur sur la longueur de la région. Le calcul est effectué en prenant la plus petite élongation d'un ensemble de boîtes englobantes avec des orientations variant de 22.5° ;
- correspondance avec la boîte englobante (Fitting OBB) : rapport de la superficie de la région par rapport à la superficie de la boîte englobante. Le calcul est effectué en prenant la valeur de la boîte la plus englobante par rapport à un ensemble de boîtes englobantes avec des orientations variant de 22.5° .

Le choix de ces attributs s'est fait au regard des classes d'intérêts recherchées. Nous n'avons calculé d'attributs ne semblant pas, a priori, discriminants.

Une illustration de ces indices est présentée à la figure 4.1.

Attributs contextuels

Les attributs contextuels représentent l'environnement de la région et son rapport à cet environnement. Il est, par exemple, intéressant de modéliser le fait qu'une région est adjacente à une région d'une classe donnée. Comme exemple concret, on peut vouloir modéliser le fait qu'une route doit être adjacente à une autre route.

Le problème de cette caractérisation est qu'avant l'étape de classification, la classe des régions avoisinant une région à classer n'est pas disponible.

Il est possible d'utiliser des attributs de base des régions avoisinantes (surface, statistiques spatiales, ...), mais cela reviendrait à apprendre en même temps que la classe d'une région donnée, la classe des régions avoisinantes. De plus, le problème de propositionnalisation devient plus complexe, le nombre d'attributs constructibles devenant important.

Le lecteur peut se référer à Debeir [23], pour un exemple de classification en intégrant des connaissances exogènes (notamment sur les réseaux routiers, canaux, ...) et des attributs inter-

régions (statistiques sur les premiers voisins de la région). L'étude montre que l'utilisation des attributs inter-régions n'améliore pas les performances.

Nous verrons dans le chapitre sur la détection (section 5.3), une utilisation du contexte pour définir la classe d'un groupement de régions.

À présent que les attributs permettant de classer une région sont établis, nous allons étudier les différents critères possibles pour évaluer une classification.

4.1.3 Critères d'évaluation

Les critères d'évaluation d'une classification d'image peuvent être assez similaires à ceux d'évaluation d'une segmentation d'image déjà vus à la section 3.1.2 de cette thèse. Pour la classification, il est néanmoins possible de se limiter à quatre critères d'évaluation : la précision, le rappel, la F-mesure et le Kappa.

Ces trois critères sont basés sur une matrice de confusion. Nous rappelons ici comment créer une matrice de confusion K entre une classification et des régions de référence. Pour chaque pixel d'évaluation d'une classe C_i , étant étiqueté par la classe C_j dans la classification, la valeur de la cellule K_{ij} est incrémentée de 1 (les cellules étant initialisées à zéro). Dans une classification, une région (et donc les pixels sous-jacents) peut n'avoir aucune étiquette attribuée. Dans ce cas, les pixels sous-jacents de la région ne seront pas ajoutés dans la matrice de confusion.

Précision

La précision (qui était nommée précision empirique dans le chapitre 3) est le nombre de pixels bien étiquetés divisé par le nombre total de pixels étiquetés, c'est-à-dire :

$$EA = \frac{1}{\Omega(\mathcal{C})} \times \sum_{i=1}^{\Omega(\mathcal{C})} \frac{K_{ii}}{\sum_{j=1}^{\Omega(\mathcal{C})} K_{ji}} \quad (4.3)$$

Il peut aussi être intéressant de se focaliser sur la précision pour une classe donnée. On utilise alors les pixels qui ont été étiquetés par la classe donnée. Ce critère est défini par :

$$EA(C_i) = \frac{K_{ii}}{\sum_{j=1}^{\Omega(\mathcal{C})} K_{ji}} \quad (4.4)$$

Rappel

Le rappel est le nombre de pixels bien classés divisé par le nombre de pixels d'exemples, c'est-à-dire :

$$R = \frac{1}{\Omega(\mathcal{C})} \times \sum_{i=1}^{\Omega(\mathcal{C})} \frac{K_{ii}}{\Omega(C_i)} \quad (4.5)$$

avec $\Omega(C_i)$ le nombre de pixels des régions de référence de la classe C_i .

À nouveau, on peut choisir de se focaliser sur le rappel pour une classe donnée. Ce critère est défini par :

$$R(C_i) = \frac{K_{ii}}{\Omega(C_i)} \quad (4.6)$$

F-Mesure

La F-mesure est un critère qui combine le rappel et la précision afin de fournir une évaluation unique. Sa formule est la suivante :

$$\text{F-mesure} = \frac{2 \times R \times EA}{R + EA} \quad (4.7)$$

À nouveau il est possible de l'utiliser pour une classe en particulier :

$$\text{F-mesure}(C_i) = \frac{2 \times R(C_i) \times EA(C_i)}{R(C_i) + EA(C_i)} \quad (4.8)$$

Kappa

Le Kappa est un critère d'évaluation qui permet d'évaluer une classification par rapport à une classification aléatoire. Il est défini par :

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \quad (4.9)$$

avec P_0 la proportion d'accord entre la classification et les exemples et P_e la proportion d'accord aléatoire. Nous utilisons la précision pour P_0 et nous calculons P_e par :

$$P_e = \frac{\sum_{i=1}^{\Omega(C)} K_i \cdot K_i}{\left(\sum_{i=1}^{\Omega(C)} \sum_{j=1}^{\Omega(C)} K_{ij} \right)^2} \quad (4.10)$$

La valeur de ce critère peut aller de -1 à +1. Un score proche de zéro indique que la classification équivaut à une classification aléatoire. Plus la valeur est proche de 1, plus le résultat est corrélé aux exemples. Inversement, un score inférieure à zéro indique une corrélation inverse aux exemples.

À présent que les bases nécessaires à l'évaluation des méthodes de classification sont posées, nous allons commencer par étudier une approche utilisant une classification supervisée basée sur des exemples (section 4.2). Nous nous attarderons, à la section 4.3, sur une approche utilisant les connaissances des experts pour effectuer cette étape de classification. À la section 4.4, nous constaterons la complémentarité de ces deux approches et proposerons une méthode d'unification de leurs résultats. Enfin, à la section 4.5, nous développerons une méthode pour remettre en cause la segmentation à partir du résultat de la classification.

4.2 Classification à l'aide d'exemples

Dans cette section, nous allons aborder la classification de régions par des méthodes de fouille de données. Ces méthodes utilisent des régions dont la classe est connue pour induire un modèle de prédiction afin de pouvoir prédire la classe d'autres régions. Ce sont des méthodes de classification supervisée.

4.2.1 Problématique du jeu d'exemples d'apprentissage

Une des difficultés dans la classification de régions vient du fait qu'il existe une différence non négligeable entre la segmentation obtenue et les régions fournies par l'expert. Les méthodes de classification se basant sur des exemples d'apprentissage, le choix et la qualité de ces exemples impactent directement la qualité des résultats. Pour obtenir une bonne classification, les exemples d'apprentissage doivent être corrects (une région de route ne doit pas être utilisée comme région d'apprentissage de maison) et représentatifs.

Deux approches sont possibles pour s'assurer de la qualité du jeu d'apprentissage. La première consiste à prendre les exemples fournis par l'expert comme ensemble d'apprentissage. Dans ce cas, ils peuvent être considérés comme corrects (on considère que l'expert fait un nombre d'erreurs négligeable ou nul). La représentativité créée par l'expert est par contre discutable. Rien n'indique en effet que les régions obtenues par segmentation seront similaires à celles fournies par l'expert. Nous qualifions cette méthode de *brute* comme elle utilise directement les données de l'expert.

La seconde approche consiste à utiliser des régions issues de la segmentation comme régions d'apprentissage. Le problème se situe dans l'étiquetage des régions de la segmentation à partir des régions fournies par l'expert. Dans ce cas, les exemples qui seront choisis seront plus représentatifs (car choisis directement dans la segmentation), mais nous introduisons le risque de faire des erreurs sur l'étiquetage des régions d'apprentissage.

Pour cette approche, nous définissons quatre méthodes :

- méthode Janssen ; pour chaque région donnée par l'expert, on étiquette dans la segmentation, la région qui a le meilleur score de Janssen [63] avec celle-ci.
- méthode vote ; chaque région est étiquetée par la classe majoritaire des pixels, si aucun pixel n'est étiqueté, la classe n'a pas d'étiquette.
- méthode couverture ; toute région de la segmentation dont plus de 50% des pixels sont d'une classe est étiquetée par cette classe. C'est un cas particulier de vote. L'objectif est d'éviter de créer des régions d'apprentissage mal étiquetés.
- méthode fusion ; consiste en l'union des ensembles de régions fournies par les méthodes brute, Janssen et couverture.

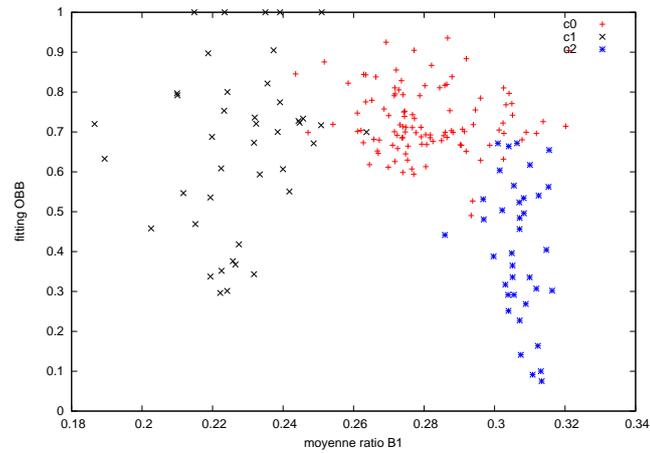
Une dispersion des valeurs des attributs de région est donnée pour chacune des méthodes de création de jeu d'apprentissage à la figure 4.2. La méthode vote n'est pas représentée car son espace de données est trop confus. On constate que plus la méthode accepte des régions s'éloignant des exemples de l'expert, plus l'espace des données devient complexe et les classes non séparables. Il est donc, a priori, de plus en plus difficile d'induire une méthode de classification.

La figure 4.3 représente les dispersions des objets pour une classe donnée selon les méthodes de génération d'exemples. On y constate que la méthode Janssen, et encore plus la méthode couverture, créent des exemples sensiblement différents de la méthode brute pour les attributs considérés. Il est à noter qu'une grande partie des exemples de la méthode Janssen sont aussi repris par la méthode couverture.

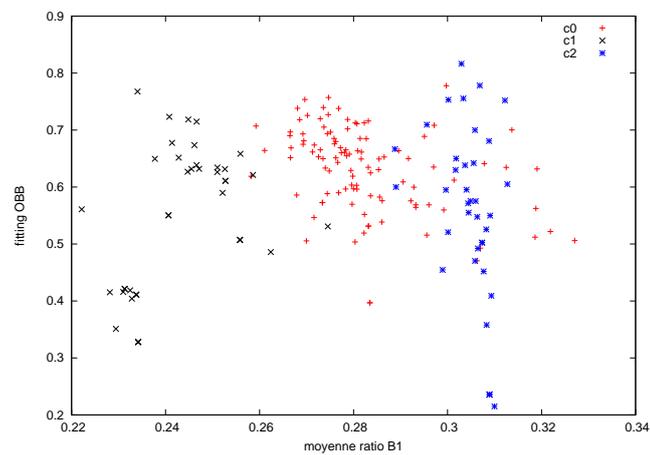
Nous allons à présent appliquer ces méthodes de génération d'exemples pour plusieurs algorithmes de classification supervisée. Il s'agit de la méthode des k plus proches voisins, du perceptron multicouche et du bayésien naïf. Ces méthodes étant usuelles en fouille de données, le lecteur peut se référer à l'annexe C pour un descriptif de celles-ci.

4.2.2 Résultats

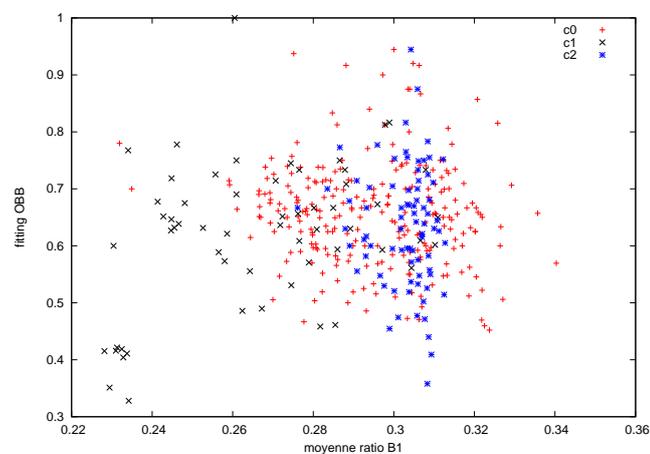
Nous allons à présent appliquer les différentes méthodes sur la segmentation présentée à la figure 4.4. Cette segmentation a été obtenue en utilisant la méthode de transformation d'espace



Espace méthode brute

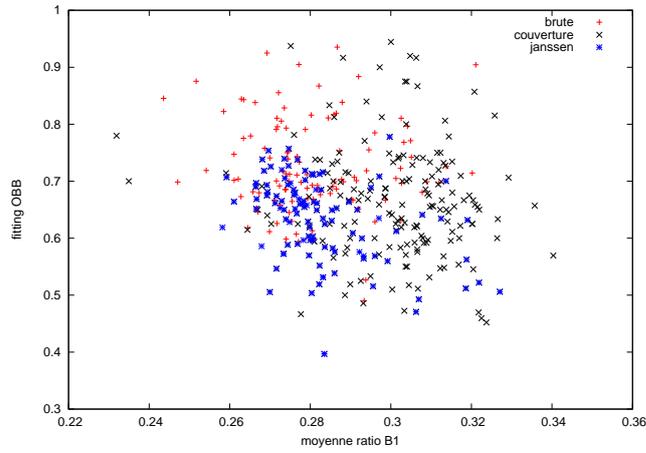


Espace méthode Janssen

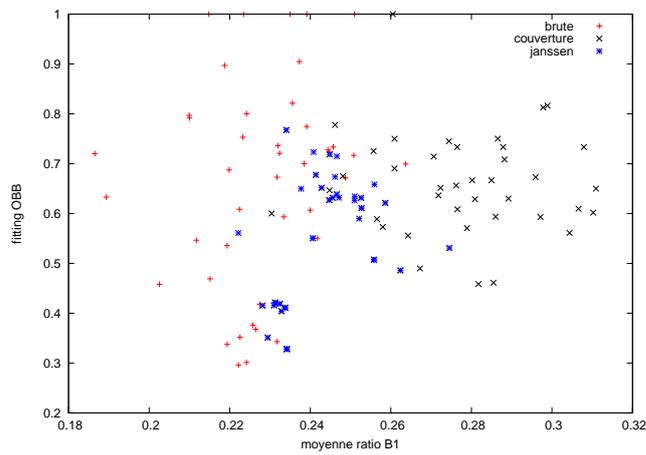


Espace méthode couverture

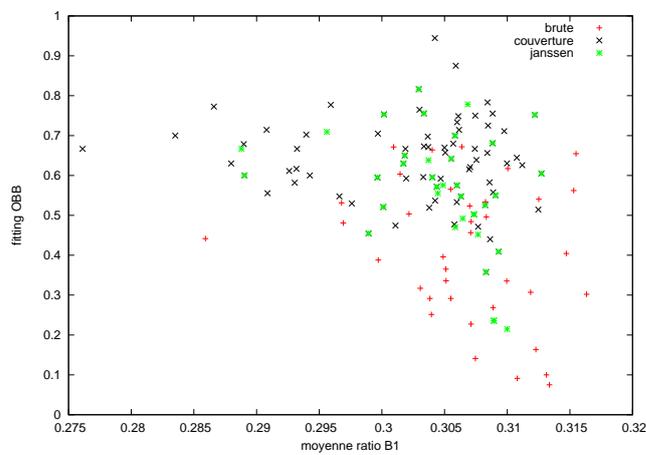
FIG. 4.2 – Représentation des données selon les méthodes sur les attributs *fitting OBB* et *ratio B1* (c0 = maison, c1 = végétation, c2 = route).



Dispersion des objets de la classe *Maison Orange* selon la méthode de génération



Dispersion des objets de la classe *Végétation* selon la méthode de génération



Dispersion des objets de la classe *Route* selon la méthode de génération

FIG. 4.3 – Représentation des données selon les classes sur les attributs *fitting OBB* et *ratio B1*.



FIG. 4.4 – Segmentation utilisée pour les évaluations empiriques de la classification basée régions.

pour la segmentation (voir section 3.3). Sur cette segmentation, une dizaine de régions ont été modifiées par l'expert afin de corriger certaines erreurs.

La segmentation finale du chapitre 3 peut sembler plus intéressante, notamment grâce à l'utilisation de la méthode d'apprentissage des paramètres (voir section 3.2). Nous avons néanmoins préféré cette segmentation car l'expert a séparé les maisons qui étaient regroupées sous une même région. Nous avons ainsi des régions plus conformes à la réalité pour cette classe. Nous verrons au chapitre 6, l'utilisation de segmentations de qualité similaire à celle obtenue en fin du chapitre 3.

L'application des différentes méthodes de création du jeu d'apprentissage et des différentes méthodes d'apprentissage est illustrée à la table 4.1. Sauf pour la méthode de classification 3-ppv biaisée, les résultats sont obtenus par validation croisée (*ten folds cross-validation*) sur les quatre zones d'exemples de l'image. Chaque zone est évaluée séparément en utilisant les trois zones restantes comme zones d'apprentissage de la méthode de classification. La moyenne sur les quatre zones est prise comme résultat.

La méthode 3-ppv biaisée utilise toutes les zones comme apprentissage et évaluation en même temps. Sa tâche est simplifiée car elle dispose ainsi d'exemples sur les zones où elle est évaluée, ce qui limite le besoin de généralisation de l'algorithme. De plus, dans le cas des méthodes de création d'exemples janssen, couverture et fusion, les exemples à classer se retrouvent en grande

TAB. 4.1 – Résultats comparatifs pour les différentes méthodes de création du jeu d’apprentissage et pour les différentes méthodes d’apprentissage. Les résultats sont donnés pour la précision (en %) sur une validation croisée des 4 zones d’apprentissage (sauf 3-ppv biaisée).

Méthode de création	3-ppv biaisée	3-ppv	Réseau de neurones	Bayésien naïf
brute	91,65	91,04	90,90	89,80
Janssen	93,01	92,36	92,18	85,58
couverture	96,29	92,54	92,59	89,91
fusion	96,42	92,53	92,49	89,13
vote	98,90	91,68	85,17	81,35

partie dans le jeu d’apprentissage. Or, dans le cas de la méthode 3-ppv, si un exemple à classer se retrouve dans le jeu d’apprentissage, il sera forcément bien classé. La distance à lui-même étant nulle, la pondération de sa classe réelle sera infinie et sera donc utilisée. L’utilisation de cette méthode permet d’établir le score maximal atteignable par la méthode 3-ppv. Il est en effet improbable que la méthode 3-ppv non biaisée obtienne de meilleurs résultats comme elle dispose d’exemples a priori moins représentatifs de la zone d’évaluation.

Les résultats montrent que l’utilisation directe des exemples (méthode brute) donne de moins bons résultats que de prendre les exemples correspondant issus de la segmentation (méthodes Janssen, couverture, fusion et vote). Même avec la méthode 3-ppv biaisée, le résultat avec la méthode brute n’est que de 91.65%. Ceci dénote la moindre représentativité des régions données par l’expert par rapport aux régions de la segmentation. L’utilisation de la méthode Janssen améliore le résultat, mais les exemples créés par cette méthode ne sont toujours pas complètement représentatifs de l’ensemble des régions à classer. Plus la méthode de génération d’exemples inclut de régions, plus le résultat de la méthode 3-ppv biaisée est bon. Il va jusqu’à 98.90% en intégrant toutes les régions qui ont au moins un pixel appartenant à un exemple d’une classe (méthode vote). Le score obtenu n’est pas de 100% car certaines régions de la segmentation ont des pixels appartenant à différentes classes (erreur dans la segmentation ou dans les exemples de l’expert).

Pour les autres méthodes de classification, les résultats sont différents. On constate à nouveau l’inadéquation des exemples de l’expert pour l’apprentissage. La méthode vote obtient de mauvais résultats avec toutes les méthodes de classification. Ces mauvais résultats sont dû à la faible qualité de certains exemples ainsi créés. En effet, il suffit qu’un pixel soit d’une classe pour qu’une région soit utilisée comme exemple d’apprentissage. Si l’on prend en compte la possibilité d’erreur dans la segmentation (qui est réelle), il paraît ambitieux de penser qu’un pixel suffit à définir la classe d’une région. Cela entraîne une complexification de l’espace de données des exemples d’apprentissage et une baisse de la généralisation possible (puisqu’on se met à douter de la véracité des exemples d’apprentissage). Ceci impacte fortement les méthodes qui font de l’induction de modèles, comme le réseau de neurones et le bayésien naïf. La méthode 3-ppv, qui ne fait pas de généralisation, est impactée dans une moindre mesure. Les meilleurs résultats sont obtenus avec la méthode fusion ou de couverture.

Départager les méthodes couverture et fusion nous semble un exercice difficile. En effet, les résultats sont assez similaires sauf pour la méthode de classification bayésien naïf. Nous privilégions néanmoins la méthode fusion pour la raison suivante : les régions fournies par l’expert sont, a priori, de bonne qualité. Même si l’imprécision de la segmentation réduit l’intérêt de ces exemples, leur qualité ne peut que servir sur des zones possiblement mieux segmentées. Il ne nous semble donc pas pertinent de les supprimer.



FIG. 4.5 – Classification du jeu d'évaluation utilisant la méthode des 3-ppv avec la méthode de génération des exemples fusion.

Les résultats de la méthode des 3 plus proches voisins combinée à la méthode fusion est illustrée à la figure 4.5. La zone 2 a été utilisée comme jeu d'apprentissage. La table 4.2 détaille les résultats pour chaque classe évaluée sur les zones restantes. On constate que la végétation est très bien classée. La classification commet des erreurs entre les maisons et les routes, certaines parties de routes sont classées en tant que maisons, ce qui réduit sensiblement la précision de la classe *maison* et le rappel de la classe *route*.

4.2.3 Conclusion sur l'apprentissage à partir d'exemples

Dans cette section, nous avons étudié la classification de régions à partir de méthodes basées sur des exemples. Nous avons vu qu'un élément important pour la classification de régions est la définition des exemples d'apprentissage et la prise en compte de l'imperfection de la segmentation et donc des objets construits. Cette définition des exemples semble plus importante que le choix des méthodes de classification, qui, si elles sont appliquées sur des données d'apprentissage représentatives des régions à classer, donnent des résultats proches.

À partir des résultats obtenus, nous pouvons conclure que la méthode de classification des k plus proches voisins, alliée à la méthode de création d'exemple fusion est la plus pertinente

TAB. 4.2 – Résultats détaillés pour chaque classe en utilisant la méthode de génération d'exemples *fusion* avec l'algorithme de classification 3-ppv.

Classe	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Maisons	86,73	94,78	94,78
Végétation	98,28	96,07	97,17
Route	94,09	87,94	90,56
Moyenne	93,04	92,93	92,98

pour le classement des régions. La méthode de classification des k plus proches voisins a montré sa robustesse même sur des données d'apprentissage très imparfaites, tandis que la méthode de génération d'exemples fusion obtient de très bons résultats et une bonne robustesse à la méthode de classification utilisée.

Nous allons à présent aborder la classification à l'aide de connaissances du domaine fournies par l'expert.

4.3 Classification à l'aide de connaissances du domaine

Tandis qu'à la section précédente nous avons vu une méthode de classification basée prioritairement sur les exemples contenus dans la base de connaissances, nous proposons ici une approche de classification des régions basée sur les connaissances fournies par l'expert dans la base de connaissances. Ce travail a été mené en collaboration avec l'ensemble des membres participants au projet FoDoMuSt ⁴.

4.3.1 Méthodes de classification par connaissances du domaine

Les systèmes à base de connaissances ont prouvé leur efficacité pour l'identification d'objets complexes [78, 19], l'analyse d'image [87, 103] ou l'interprétation d'images [94]. Par exemple, les systèmes SIGMA [87] et Schema [32] effectuent des analyses d'images aériennes en utilisant plusieurs descripteurs de régions. Ces systèmes permettent d'accéder à un niveau sémantique élevé. Néanmoins, comme le soulignent Crevier et al. [21], ces systèmes sont très dépendants du domaine car ils intègrent des connaissances a priori sur l'image à analyser. Leurs inconvénients sont que les connaissances du domaine ne sont pas clairement séparées de la partie procédurale et que la base de connaissances est difficile à produire.

C'est pourquoi des travaux récents ont proposé d'utiliser des systèmes d'ontologies pour décrire plus clairement les connaissances du domaine. Zlattof et al. [135] utilisent les relations spatiales entre les concepts pour fusionner les régions et identifier les objets de classes. L'utilisation quasi-exclusive des relations spatiales se prête difficilement à notre domaine d'application (imagerie satellitaire à très haute résolution).

Panagi et al. [98] proposent un algorithme génétique d'analyse sémantique basé sur une ontologie. Des attributs de bas niveau sont extraits de l'image et utilisés pour correspondre avec l'ontologie. Un ensemble d'hypothèses (région, liste des classes possibles et degré de confiance) sont testées avec un algorithme génétique pour déterminer l'interprétation acceptable de l'image. Seules les relations spatiales sont utilisées dans ce système.

⁴ACI Masse de Données — Fodomust <http://fodomust.u-strasbg.fr>

Nous allons à présent proposer un algorithme d'identification par connaissance du domaine inspiré de ces travaux.

4.3.2 Algorithme d'identification par connaissances du domaine

L'algorithme proposé calcule un score de correspondance entre une région à identifier et des classes contenues dans la base de connaissance. Le résultat final est donc, pour chaque région, une liste d'étiquettes de classes possibles et un score de confiance pour chaque attribution d'étiquette. Nous allons d'abord présenter le calcul du score de correspondance utilisé entre une région et une classe. Nous présentons ensuite l'algorithme de recherche dans la hiérarchie de classes.

Formalisation de la base de connaissances

Comme nous l'avons vu au chapitre 2, la base de connaissances contient une hiérarchie des classes recherchées. Nous allons présenter ici la formalisation des connaissances utilisées dans cette section.

Parmi les travaux présentés dans [109] pour modéliser de la connaissance, les approches à base de *frames* semblent les plus appropriées pour les problèmes de classification d'image [83]. Dans ces systèmes [90], les connaissances sont regroupées dans une hiérarchie de *frames*. Un *frame* est un prototype (c'est-à-dire un objet représentatif d'une famille) composée d'un ensemble de *slot* (attributs) décrivant les propriétés du prototype. Nous nous basons sur cette représentation dans nos travaux.

Ainsi la formalisation des connaissances retenue est la suivante :

- Θ est l'ensemble des classes ; $\rho(C_i)$ est la profondeur de la classe C_i dans la hiérarchie
- \mathcal{A} est l'ensemble de tous les attributs présents dans la base de connaissances

Définition 1 (Sous-classe) *La relation de sous-classe \preceq_{Θ} entre deux classes est un ordre partiel défini par :*

$$\forall (C_i, C_j) \in \Theta^2, C_i \preceq_{\Theta} C_j \text{ signifie que } C_i \text{ est une sous-classe de } C_j.$$

Un tel concept C_j est appelé super-classe de C_i , c'est-à-dire que C_j est une classe plus générique que C_i .

Par exemple, $Maison\ Orange \preceq_{\Theta} Maison$, signifie que la classe *Maison* est plus générique que *Maison Orange*. En effet, la classe *Maison* regroupe toutes les maisons y compris celles de couleur orange.

Définition 2 (Attributs) *L'ensemble des attributs \mathcal{A}_{C_i} associés à une classe $C_i \in \Theta$ est défini comme l'ensemble des attributs de la classe C_i sur lesquels un ensemble de définition (plus restreint que l'espace de définition original) a été donné par l'expert.*

$\omega(a, C_i)$ est le poids de l'attribut a associé à la classe C_i . Ce poids reflète l'importance de l'attribut a pour décrire la classe C_i d'après l'expert.

Définition 3 (Valeurs et espace de définition des attributs) *Soit $a \in \mathcal{A}$ un attribut, $\mathcal{V}(R, a)$ est la valeur de l'attribut a pour la région R . $\min(C_i, a)$ (respectivement $\max(C_i, a)$) représente la valeur minimale (respectivement maximale) que peut prendre l'attribut a pour pouvoir décrire un objet de la classe C_i .*

Voir pour exemple la classe *Maison Orange* décrite au tableau 4.3.

TAB. 4.3 – Classe *Maison Orange*.

Type d'attribut	Attribut	Poids	Intervalle	
			min	max
Spectral	Bleu	1	21.7	62.3
	Vert	1	19.4	80.1
	Rouge	1	29.7	135.1
	Proche IR	1	34.8	139
	NDVI	1	50.2	108
Spatial	Diamètre(m)	0.8	13	61
	Surface (m ²)	1	10	600
	Élongation (m)	0.6	1	3.1

Définition 4 (Attributs spécifiques et hérités) *L'ensemble des attributs spécifiques \mathcal{AS}_{C_i} est l'ensemble des attributs dont l'espace de définition est défini dans la classe C_i . L'ensemble des attributs hérités \mathcal{AH}_{C_i} est l'ensemble des attributs dont l'espace de définition n'est pas défini dans la classe C_i mais s'applique à cette classe car défini dans une super-classe de C_i .*

Opérationnalisation de la base de connaissances

Une fois les connaissances formalisées, nous pouvons nous occuper de les opérationnaliser, c'est-à-dire de créer une méthode permettant de les utiliser. Nous allons d'abord présenter le calcul du score de correspondance utilisé entre une région et une classe. Nous présentons ensuite l'algorithme de recherche dans la hiérarchie de classe.

Le calcul du score de correspondance proposé est basé sur une approche orientée attributs. Il correspond à vérifier la validité des valeurs des attributs d'une région selon les intervalles définis par l'expert dans la base de connaissances. Néanmoins, la description de chaque classe n'a pas une structure sémantique définie. Il n'est donc pas possible d'utiliser les mesures comme MDSM [108] ou les mesures présentées dans Tversky et al. [121] et Schwering et al. [114].

Une région peut, a priori, être comparée à n'importe quelle classe. L'ensemble des attributs utilisés comme description peut être différent pour chaque classe. Par exemple, la classe *Maison Orange* est définie par un grand nombre d'attributs de forme (élongation, Miller, ...) et certains attributs spectraux, alors que la classe *Ombre* n'est définie que par des attributs spectraux, les attributs de forme étant non pertinents pour cette classe.

Sans connaissance *a priori*, cette dissymétrie implique de calculer tous les attributs d'une région, même si une partie de ces attributs est non pertinente relativement à la classe de la région non encore connue.

Pour prendre en compte cette spécificité (les classes n'ayant pas toutes le même ensemble d'attributs), nous proposons un score de correspondance basé sur la distance entre les valeurs des attributs pour une région donnée et l'espace de définition de cet attribut pour la classe donnée. Le score ici proposé est composé d'une similarité locale (correspondance d'une région avec une classe donnée) et une similarité globale (correspondance d'une région avec une hiérarchie de concepts).

La mesure de similarité locale compare les valeurs des attributs d'une région avec les attributs spécifiques d'une classe. Pour définir la mesure de similarité locale, nous avons besoin de définir le degré de validité entre une classe C_i et une région R pour un attribut a donné qui mesure à quel point la classe C_i est possible pour la région R selon l'attribut a .

Définition 5 (Degré de validité) *Soit $\text{Valid}(a, C_i, R)$ le degré de validité pour un attribut a entre une région R et une classe C_i .*

$$\text{Valid}(a, C_i, R) = \begin{cases} 1 & \text{si } \mathcal{V}(R, a) \in [\min(C_i, a); \max(C_i, a)] \\ \frac{\mathcal{V}(R, a)}{\min(C_i, a)} & \text{si } \mathcal{V}(R, a) < \min(C_i, a) \\ \frac{\max(C_i, a)}{\mathcal{V}(R, a)} & \text{si } \mathcal{V}(R, a) > \max(C_i, a) \end{cases} \quad (4.11)$$

Définition 6 (Similarité locale) Soit $\text{Sim}_\alpha(R, C_i)$ la similarité locale entre une région R et une classe C_i .

$$\text{Sim}(R, C_i) = \frac{\sum_{a \in \mathcal{AS}(C_i)} \omega(a, C_i) \text{Sim}(a, C_i, R)}{\sum_{a \in \mathcal{AS}(C_i)} \omega(a, C_i)} \quad (4.12)$$

Le score de similarité globale évalue la pertinence de la correspondance entre une région R et une classe C_i en prenant en compte la hiérarchie de classes.

Définition 7 (Score de correspondance) Soit $\mathcal{P}(C_i)$ le chemin dans la hiérarchie de classe commençant à la racine et finissant sur la classe C_i défini par :

$$\mathcal{P}(C_i) = \{C_j \mid C \preceq_\Theta \cdots \preceq_\Theta C_2 \preceq_\Theta C_1\}$$

Le score de correspondance $\text{Score}(R, C_i)$ entre une région R et une classe C_i est défini par :

$$\text{Score}(R, C_i) = \frac{\sum_{C_j \in \mathcal{P}(C_i)} \rho(C_j) \text{Sim}(R, C_j)}{\sum_{C_j \in \mathcal{P}(C_i)} \rho(C_j)} \quad (4.13)$$

Le score de correspondance est une combinaison linéaire entre les similarités locales. Les similarités locales des classes génériques sont propagées par héritage aux concepts plus spécifiques. Dans ce calcul, nous intégrons un coefficient de spécialisation ρ égal à la profondeur de la classe dans la hiérarchie. Par ce moyen, la mesure favorise la spécialisation des classes, indiquant que toute information additionnelle apporte une nouvelle sémantique.

À présent que le score de correspondance entre une région et une classe donnée a été défini, il nous faut détailler l'algorithme d'exploration de la hiérarchie de classes pour trouver la ou les meilleures classes pour une région. L'algorithme que nous proposons est basé sur des heuristiques afin de réduire l'espace de recherche.

Le processus général de l'exploration est le suivant : si la région peut correspondre à la classe courante, l'algorithme va descendre d'un niveau dans la hiérarchie définie par l'ordre partiel \preceq_Θ . Si la région ne peut pas correspondre à la classe courante, celle-ci est abandonnée et ses sous-classes ne seront pas explorées.

Nous définissons le seuil minScore comme étant la valeur minimale du score de correspondance entre une région et une classe pour considérer possible que la région soit effectivement un objet de cette classe.

Soit $\mathcal{L} : \Theta \rightarrow \Theta$ tel que $\mathcal{L}(R)$ est l'ensemble des classes qui peuvent correspondre à la région R tenant compte du seuil minScore .

$$\mathcal{L}(R) = \{(C_i, \text{Score}(R, C_i)) \mid \text{Score}(R, C_i) \geq \text{minScore}\} \quad (4.14)$$

L'exploration de la hiérarchie de classe est présentée dans l'algorithme 1. Ce processus peut être répété pour chaque région d'une image segmentée afin de donner une interprétation de l'image complète.

Algorithme 1 Exploration de la hiérarchie de classe.

Entrées : une région R , une hiérarchie de classes Θ , un seuil $minScore$.

Sortie : $\mathcal{L}(R)$.

$\mathcal{L}(R) = \emptyset$;

$\mathcal{RC} = \{racine\}$;

tant que ($\mathcal{RC} \neq \emptyset$) **faire**

 meilleursProfondeur = \emptyset ;

pour tout $C_i \in \mathcal{RC}$ **faire**

$s = \text{Score}(R, C_i)$;

si ($s \geq minScore$) **alors**

$\mathcal{L}(R) = \mathcal{L}(R) \cup \{(C_i, s)\}$;

 meilleursProfondeur = meilleursProfondeur $\cup \{C_i\}$;

fin si

fin pour

$\mathcal{RC} = \emptyset$;

pour tout $C_j \in \text{meilleursProfondeur}$ **faire**

$\mathcal{RC} = \mathcal{RC} \cup \{C_i | C_i \preceq_{\Theta} C_j\}$;

fin pour

fin tant que

return $\mathcal{L}(R)$

4.3.3 Évaluation

Nous allons évaluer l'algorithme de classification proposé par une approche analytique tout d'abord, puis par une évaluation empirique sur une image réelle. Nous finirons par une discussion sur l'importance de la qualité de la segmentation sur cet algorithme de classification.

Évaluation analytique

Nous évaluons analytiquement l'algorithme proposé sous trois aspects. Premièrement, nous donnons les pré-requis nécessaires à l'application de cet algorithme. Deuxièmement, nous présentons les limites conceptuelles de celui-ci. Troisièmement, nous détaillons certains de ces aspects.

Pour pouvoir appliquer cet algorithme, une condition doit être remplie. L'ensemble des classes d'intérêt doit être structuré sous la forme d'une hiérarchie. A chaque classe doit correspondre un ensemble d'espaces de définition sur des attributs (au moins sur un attribut, sinon toute région sera reconnue).

L'algorithme contient des limites qu'il convient de connaître :

- Précision de la caractérisation des classes : L'expert spécifie pour chaque classe des espaces de définition sur un ou plusieurs attributs. Ceux-ci prennent la forme d'intervalles. Il s'agit donc de définir des hypercubes dans l'espace de données. Même en étant définis au mieux, une grande partie de ces hypercubes peut ne pas correspondre à la classe définie à cause de la simplicité de la représentation. Cette simplicité est néanmoins nécessaire pour que l'expert puisse fournir ses connaissances.
- Qualité de la segmentation : Si les valeurs des attributs des régions sont différentes des espaces de définition pour la classe de ces régions, l'algorithme ne pourra classifier correctement ces régions. Cela peut se produire si la segmentation n'est pas de qualité suffisante. Ce problème sera étudié en détail plus loin. Il est à noter que le paramètre $minScore$ permet de relâcher en partie cette contrainte.

Enfin, il convient de noter certaines spécificités de cet algorithme :

TAB. 4.4 – Résultats de l'évaluation selon la valeur de minScore, basée sur les pixels des exemples d'évaluation.

minScore	Précision (en %)	Rappel (en %)	F-Mesure (en %)
0,75	85.4	85.9	85.7
0.80	85.4	85.9	85.7
0.85	85.6	85.8	85.7
0.90	86.2	85.4	85.8
0.95	87.3	83.7	85.5
1.00	88.3	66.2	75.7

- Espace de définition des attributs : Nous avons ici utilisé uniquement des intervalles. Il est possible d'utiliser des représentations plus complexes (gaussiennes simples ou multiples, unions d'intervalles, ...). Néanmoins, des formes plus avancées sont difficilement modélisables par l'expert.
- Complexité : Le temps d'exécution dépend principalement du nombre de classes N_c et du nombre d'attributs maximal N_a par classe. Au maximum, le classement d'une région prendra $N_c \times N_a$ opérations de comparaison entre la valeur d'un attribut de la région et l'espace de définition pour un attribut d'une classe.

Nous allons à présent procéder à l'évaluation sur des données réelles.

Évaluation empirique

L'évaluation empirique s'est déroulée sur la même segmentation qu'utilisée à la section précédente (voir figure 4.4). Les exemples disponibles sur cette image sont intégralement utilisés pour l'évaluation (la méthode ne nécessitant pas d'exemples d'apprentissage).

Comme l'image contient principalement quatre classes (eau, maisons, route, végétation), nous travaillons uniquement sur un extrait de la hiérarchie de classes contenant ces classes en feuilles.

Les résultats sont donnés à la table 4.4 en fonction du seuil minScore utilisé.

La première information qu'il faut relever, c'est l'importance d'utiliser le paramètre minScore. Avec minScore = 1, 0, seuls 66% des pixels du jeu d'évaluation sont reconnus comme appartenant à une classe. Cela implique qu'il existe de nombreuses régions qui ne correspondent pas exactement à la classe à laquelle elles appartiennent. Le problème vient de l'étape de segmentation. Celle-ci ne produisant pas des régions identiques à ce que ferait un expert du domaine, les valeurs des attributs sont possiblement hors des espaces de définition de la base de connaissances. Pour traiter ceci, l'utilisation du paramètre minScore est nécessaire.

On peut aussi noter que plus la valeur de minScore augmente, plus l'algorithme devient sélectif et donc la précision augmente, alors que le rappel baisse. L'algorithme fait moins d'erreurs, mais il classe moins de régions. C'est donc un paramètre intéressant pour l'utilisateur. Selon son objectif, il peut privilégier la précision ou le rappel.

La figure 4.6 illustre une telle classification de notre image de test avec un minScore = 0,90. Les résultats obtenus sont rappelés à la table 4.6 et comparés avec les résultats obtenus pour la méthode d'apprentissage à partir d'exemples. On constate que la simplicité du modèle employé, afin de rester intelligible pour l'expert limite l'efficacité de la méthode. Néanmoins, cette méthode permet d'obtenir une classification pour l'eau ce qui n'avait pas de sens pour la méthode à base

TAB. 4.5 – Résultats de l'évaluation par classes selon la valeur de minScore, basée sur les pixels des exemples d'évaluation.

Classe	Précision (en %)					
	1,00	0,95	0,90	0,85	0,80	0,75
Maison	81,2	81,3	81,9	82,8	83,6	85,2
Végétation	98,2	98,2	98,2	98,2	98,3	98,4
Route	63,9	63,9	63,9	63,9	67,4	69,7
Eau	98,4	98,4	98,5	99,7	100	100

Classe	Rappel (en %)					
	1,00	0,95	0,90	0,85	0,80	0,75
Maison	69,6	69,4	69,0	67,5	62,0	59,8
Végétation	97,6	97,6	97,6	97,6	97,3	96,9
Route	82,4	82,4	82,4	82,3	81,5	80,5
Eau	94,1	94,1	94,1	94,1	94,1	27,6

Classe	F-Mesure (en %)					
	1,00	0,95	0,90	0,85	0,80	0,75
Maison	74,9	74,9	74,9	74,4	71,2	70,3
Végétation	97,9	97,9	97,9	97,9	97,8	97,7
Route	71,9	71,9	71,9	72,0	73,8	74,7
Eau	96,2	96,2	96,3	96,8	97,0	43,3

TAB. 4.6 – Comparaison entre la méthode de classification à l'aide d'exemples et celle à base de connaissances du domaine (on procède à la moyenne des trois premières classes pour rester comparable entre les deux méthodes).

Classe	Méthode	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Maison	Exemples	86,73	94,78	94,78
	Domaine	82,45	67,46	74,20
Végétation	Exemples	98,28	96,07	97,17
	Domaine	98,55	94,98	96,74
Route	Exemples	94,09	87,94	90,56
	Domaine	67,63	82,63	74,38
Moyenne	Exemples	93,04	92,93	92,98
	Domaine	82,88	81,69	82,28
Eau	Exemples	n/a	n/a	n/a
	Domaine	99,68	94,14	96,83

d'exemples à cause du faible nombre d'exemples d'eau.

Selon les classes, nous constatons les erreurs suivantes :

- *route* : le réseau routier est relativement bien reconnu. Le paramètre `minScore` a peu d'effet sur cette classe tant pour le rappel que pour la précision.
- *végétation* : la classe végétation étant très bien prise en compte, il n'y a que certaines régions de végétation ombrée qui ne sont pas reconnues.
- *maisons* : une bonne partie des maisons est détectée. Pour cette classe une valeur de `minScore` faible est préférable. En effet, ce sont les régions de maisons qui sont le plus mal segmentées. Elles sont donc les plus éloignées des espaces de définition fournis par l'expert. Certaines maisons ont été classées prioritairement comme de la route. Ceci est dû à une similarité spectrale et des formes elles aussi similaires (certaines régions maisons sont suffisamment allongées).
- *eau* : certaines grandes régions d'ombre ont été classées comme de l'eau. Les réponses spectrales entre ces deux classes ne sont pas discriminantes. Cette classe étant peu décrite par des indices de forme, ceux-ci ne peuvent servir à supprimer ces régions. À partir d'un seuil de `minScore = 0.95`, les erreurs sur les petites régions d'ombre classées par erreur comme de l'eau ne sont plus commises grâce à la connaissance de la taille minimale d'une surface d'eau.

Comme nous pouvons le constater, il y a deux principales formes d'erreurs. Les premières sont dues au manque de précision des connaissances. Ce problème se manifeste dans la difficulté de discrimination entre les classes *maisons* et *route*. Alors que ces classes semblent différenciables spectralement, l'algorithme proposé n'arrive pas clairement à les séparer.

Le second type d'erreur vient de la qualité des régions construites lors de la segmentation.

Influence de la qualité de la segmentation

Nous présentons ici une illustration de l'effet de la qualité de la segmentation utilisée sur les résultats.

Nous proposons d'utiliser l'algorithme sur trois segmentations qui sont illustrées à la figure 4.7. La première segmentation (qualité basse) est obtenue par l'application d'une ligne de partage des eaux suivie d'une fusion des régions (voir chapitre 3). Les différents paramètres sont

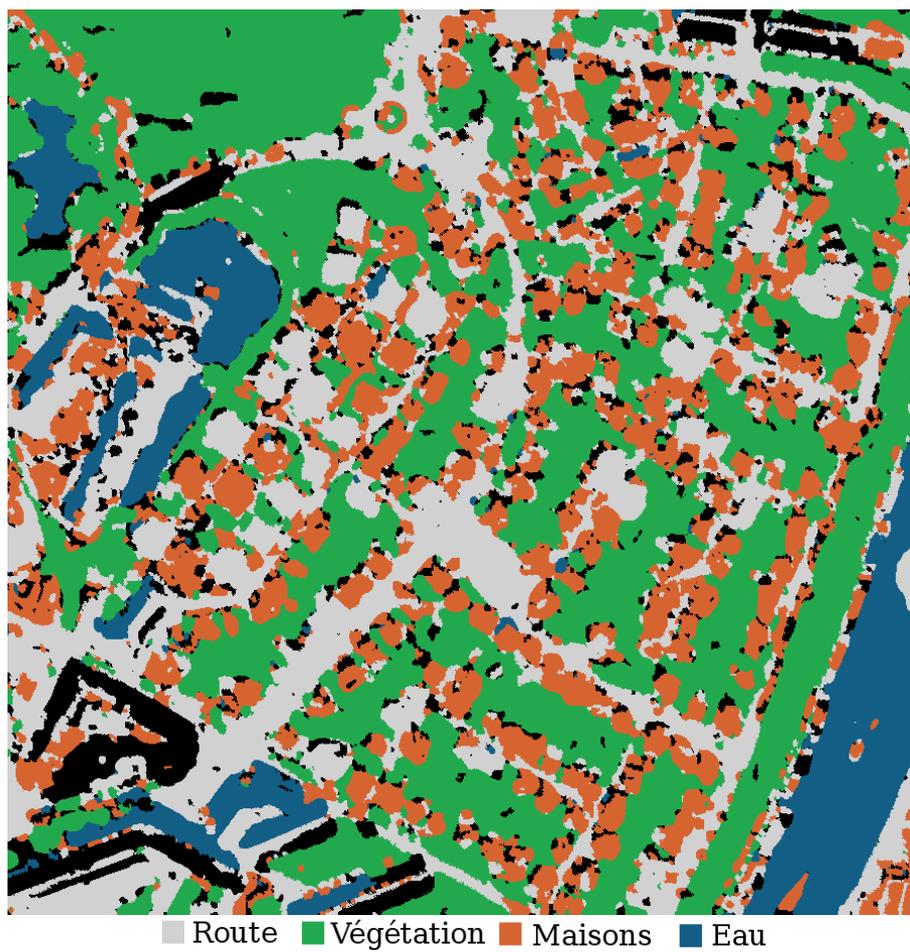


FIG. 4.6 – Résultat obtenu par classification utilisant les connaissances du domaine avec $\text{minScore} = 0.90$.

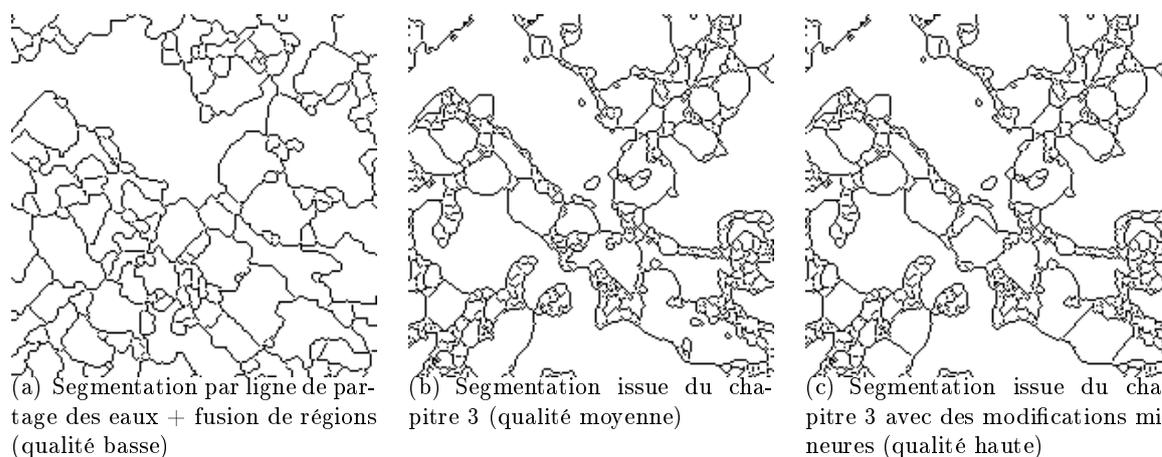


FIG. 4.7 – Illustration des trois qualités de segmentation utilisées.

choisis manuellement. La seconde segmentation (qualité moyenne) est issue de l'application de la méthode de segmentation par transformation d'espace proposée au chapitre 3. Enfin la dernière segmentation (qualité haute, utilisée dans les expérimentations précédentes) est la seconde segmentation légèrement retouchée manuellement (une dizaine de régions ont été modifiées).

Les différents résultats, en fonction de la valeur de `minScore`, sont présentés à la table 4.7.

Comme nous pouvons le constater, la qualité de la classification augmente avec la qualité de la segmentation. La précision augmente sensiblement jusqu'à 8,4% (selon la valeur de `minScore`) selon la segmentation de qualité basse et de qualité haute. Cet accroissement de la précision signifie que la segmentation génère moins de régions qui auront des attributs indiquant l'appartenance à une classe lorsque ce n'est pas le cas.

Le différentiel est encore plus important pour le rappel ou l'utilisation de la segmentation de qualité haute et permet de gagner jusqu'à 12,2%. Ceci indique que les régions devant appartenir à une classe ont des attributs qui sont plus en accord avec les connaissances fournies sur cette classe.

La qualité de la segmentation joue donc un rôle essentiel pour cet algorithme. Une mauvaise segmentation produit des régions non semblables à la réalité et dont les valeurs des attributs ne correspondent pas à ceux de la base de connaissances.

4.4 Fusion de résultats de classifications

Précédemment, nous avons vu une approche à base d'exemples et une à base de connaissances fournies par l'expert. Ces deux approches ont chacune leurs avantages et leurs inconvénients. Dans cette section, nous allons étudier la possibilité de fusionner ces deux méthodes et évaluer si cette approche hybride a un intérêt plus grand que les deux méthodes utilisées séparément.

Nous commencerons par rappeler les avantages et inconvénients de chacune des méthodes et énumérerons les synergies qui peuvent être créées [24]. Nous étudierons ensuite les méthodes usuelles de fusion de résultats de classification. Enfin, nous présenterons la méthode utilisée et finirons par une évaluation de celle-ci.

TAB. 4.7 – Résultats selon les segmentations, basées sur les pixels des exemples d'évaluation.

Qualité (voir fig. 4.7)	minScore	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Basse	0,75	81,3	81,6	81,5
	0,80	81,3	81,5	81,4
	0,85	81,3	81,4	81,3
	0,90	82,3	75,8	78,9
	0,95	82,7	73,5	77,8
	1,00	78,9	64,9	71,2
Moyenne	0,75	83,8	84,2	84,0
	0,80	83,9	84,2	84,0
	0,85	84,1	84,0	84,1
	0,90	84,6	83,7	84,1
	0,95	85,6	82,0	83,7
	1,00	86,5	64,4	73,9
Haute	0,75	85,4	85,9	85,7
	0,80	85,4	85,9	85,7
	0,85	85,6	85,8	85,7
	0,90	86,2	85,4	85,8
	0,95	87,3	83,7	85,5
	1,00	88,3	66,2	75,7

TAB. 4.8 – Avantages et inconvénients de la méthode basée exemples et celle basée connaissances de l’expert.

Méthode	Avantages	Inconvénients
Exemples	Relative indépendance à la qualité de la segmentation Bonne classification des régions similaires aux exemples d’apprentissage	Nécessite des exemples représentatifs Classification incontrôlée des régions sans représentant dans le jeu d’apprentissage (pas de rejet)
Expert	Classifications contrôlées, les régions n’appartenant pas aux classes définies sont non étiquetées Bonne classification des régions compatibles avec la définition donnée par l’expert	Dépendance à la qualité de la segmentation Difficulté d’expression des connaissances pour certaines classes.

4.4.1 Comparaison des deux méthodes précédentes

Nous avons vu, aux sections précédentes, une méthode de classification de régions basée sur des exemples et une autre basée sur des connaissances d’experts. Nous rappelons les avantages et inconvénients de chacune de ces méthodes à la table 4.8 .

Nous pouvons constater une complémentarité des deux méthodes. Dans chacune, il faut soit des exemples représentatifs, soit des connaissances de l’expert. Selon les classes, il est plus simple d’avoir l’un ou l’autre. Dans le cas de l’eau, il est difficile d’avoir des exemples représentatifs ; les surfaces d’eau pouvant avoir des formes et des tailles variées. Pour cette classe, il est donc plus simple de donner des connaissances de l’expert. En effet, celui-ci peut indiquer la taille minimale nécessaire pour qu’une région soit considérée comme une surface d’eau. Si une région correspond spectralement à de l’eau mais dispose d’une surface très petite, il s’agira probablement plus d’une zone d’ombre (l’ombre et l’eau ayant des comportements spectraux pouvant être identiques). Comme la surface de l’eau peut être importante, l’expert peut indiquer qu’il n’y a pas de limitation de superficie.

Inversement, pour la classe *maison*, la classification basée connaissances de l’expert commet quelques erreurs : certaines maisons sont reconnues comme de la route. Ces erreurs sont dues en partie au manque de précision et de distinction entre les classes *maison* et *route* dans la base de connaissances. Une autre raison de ces erreurs est l’imprécision de la segmentation. Cette imprécision entraîne que les régions obtenues ne correspondent plus exactement à la définition de la classe qu’elles représentent, telle qu’elle est formulée dans la base de connaissances. Dans ce cas, une approche de classification basée exemples permet d’obtenir de meilleurs résultats. En effet, quand bien même les régions seraient mal segmentées, si l’apprentissage s’est fait sur des régions segmentées de manière identique, la reconnaissance sera possible.

4.4.2 Méthodes de fusion de résultats de classifications

Les méthodes de fusion d'un ensemble de classifications est un domaine de recherche ancien [56] mais qui devient de plus en plus actif comme le montre la création des ateliers *Multiple Classifier Systems* en 2000 et *Supervised and Unsupervised Ensemble Methods and Their Applications* en 2007.

Différentes méthodes ont été mises au point pour réaliser cette fusion d'informations utilisant des fonctions d'agrégations simples [69], la théorie des probabilités [69], des possibilités [110] et des méta-méthodes [113, 117]. Ces différentes méthodes de fusion nécessitent de fusionner des résultats différents et complémentaires [112], cette notion pouvant être définie par plusieurs critères [93]. Fusionner des classifications fortement corrélées réduit considérablement l'apport de ces méthodes.

L'hétérogénéité des différentes méthodes de classification fusionnées peut être créée de diverses manières :

- indépendance cognitive des méthodes de classification : utilisation de différents experts n'ayant pas les mêmes connaissances ou, dans le cas de la classification supervisée, utilisation d'algorithmes d'apprentissage différents ;
- ensemble d'apprentissages différents : pour la classification supervisée, l'utilisation de jeux d'apprentissages différents permet d'obtenir des résultats différents et donc possiblement complémentaires ;
- ensemble d'attributs différents [120] : si deux méthodes de classification utilisent des ensembles d'attributs différents (par exemple l'un uniquement les attributs sur les réponses spectrales et l'autre sur la forme), les résultats obtenus seront différents.

Il est possible d'améliorer encore les résultats de la fusion en utilisant, au lieu de l'ensemble des méthodes de classification disponibles, uniquement un sous-ensemble de celles-ci. Plusieurs algorithmes ont été proposés pour faire cette sélection, notamment via l'ordonnement des méthodes par réduction d'erreur [86], par regroupement automatique [48], par algorithme évolutionnaire [67, 134].

Beaucoup de ces approches ont déjà été appliquées au domaine de la télédétection [6, 47, 96]. Nous nous intéresserons ici à une approche différente basée sur la fusion de méthodes de classification ayant des ensembles de classes différentes dont certaines n'ont pas d'exemples associés (et donc où les méthodes précédentes ne peuvent s'appliquer). Nous mettons en avant des connaissances expertes plutôt que de l'induction à partir d'exemples (déjà bien traitée dans la littérature).

Notre approche se base néanmoins sur une indépendance cognitive et une différence des exemples d'apprentissage entre les deux méthodes de classification. En effet, le fonctionnement des algorithmes est très différent : l'un utilise une notion de distance entre des régions, tandis que l'autre définit des intervalles valides pour chaque classe et chaque attribut. Les jeux d'apprentissage aussi sont différents, une méthode se basant sur des exemples donnés par l'utilisateur et la seconde se basant sur des connaissances de l'expert.

4.4.3 Algorithme de fusion

Soit \mathbb{C} l'ensemble des méthodes de classification utilisées pour la fusion. Pour chaque méthode de classification \mathbb{C}_j et pour chaque classe C_i , une pondération $w(\mathbb{C}_j, C_i)$ est prise en compte (discutée ultérieurement).

Soit $p(R_k, \mathbb{C}_j, C_i)$ le score qu'affecte la classification \mathbb{C}_j à l'étiquetage de la région R_k par la classe C_i défini par :

$$p(R_k, \mathbb{C}_j, C_i) = \begin{cases} w(\mathbb{C}_j, C_i) & \text{si } \mathbb{C}_j \text{ affecte la classe } C_i \text{ à la région } R_k \\ 0 & \text{sinon} \end{cases} \quad (4.15)$$

Soit $p(R_k, C_i)$ le score de l'affectation de la région R_k à la classe C_i défini par :

$$p(R_k, C_i) = \sum_{\mathbb{C}_j \in \mathcal{C}} p(R_k, \mathbb{C}_j, C_i) \quad (4.16)$$

La classe finalement affectée par la fusion de classification sera :

$$classe(R_k) = \begin{cases} \text{sans classe} & \text{si } p(R_k, C_i) = 0, \forall C_i \in \mathcal{C} \\ C_l \mid p(R_k, C_l) > p(R_k, C_i) \forall C_i \neq C_l \in \mathcal{C} & \text{sinon} \end{cases} \quad (4.17)$$

Un point important de la méthode est le réglage des pondérations $w(\mathbb{C}_j, C_i)$. Nous avons choisi de laisser le soin à l'expert de les définir. L'expert nous semble en effet le mieux placé pour décider si la connaissance du domaine qu'il donne sur les différentes classes permet de bien les discriminer. De même, en donnant des exemples, il peut se rendre compte de la difficulté d'apprentissage d'une classe et de la représentativité des exemples.

Dans le cas des surfaces d'eau, la superficie peut être très variable, de même que la forme. De plus, il est assez fastidieux de chercher des exemples très représentatifs de toutes les formes et superficies. Il est donc logique d'affecter une faible pondération à une classification basée sur les exemples pour cette classe. Inversement, l'expert peut facilement définir les critères pour qu'une région soit une surface d'eau, la classification basée sur les connaissances du domaine aura donc une forte pondération pour cette classe.

Il est néanmoins possible d'estimer les pondérations à utiliser avec un jeu d'exemples. La pondération peut être la précision de la classification obtenue sur ce jeu d'exemples. Nous considérons le score de précision et non pas celui de rappel, car la méthode de fusion n'est pas impactée par une région qui n'est pas étiquetée. A contrario, donner une forte pondération à une classification qui génère beaucoup de faux positifs (des régions sont étiquetées par une mauvaise classe) est pénalisant pour la classification.

4.4.4 Résultats

Nous avons appliqué la méthode proposée afin de fusionner les résultats de classification obtenus par les approches basées sur des exemples (section 4.2) et basées sur des connaissances du domaine (section 4.3).

Pour la classification basée sur les connaissances du domaine, nous avons utilisé $\text{minScore} = 0,95$. Pour la classification basée exemples, la méthode des k plus proches voisins a été utilisée avec la méthode de génération des exemples d'apprentissage fusion.

Les pondérations $w(\mathbb{C}_j, C_i)$ sont définies à la table 4.9. Pour les classes n'ayant pas d'exemple (dans notre cas, la classe surfaces d'eau), seule la classification basée connaissances du domaine peut être utilisée. Pour l'eau, seule la classification basée connaissances du domaine est utilisée avec une pondération égale à 1, cette classe pouvant être détectée avec une bonne précision. Pour la végétation, les contributions de la classification basée connaissances du domaine et celles basées sur les exemples sont égales, les deux classification ayant une bonne précision sur cette classe. Pour les maisons et la route, la pondération de la classification basée connaissances est mise à zéro et donc les contributions pour ces classes ne sont pas utilisées. En effet, ces deux classes sont difficilement discriminables par la méthode basée connaissances et induiraient trop

TAB. 4.9 – Pondération des résultats pour chaque classe de chaque classification. Les couples (classes, classification) n'ayant pas de sens sont notées *na*.

Classe	Classif. basée exemples	Classif. basée connaissances
Végétation	0.95	0.95
Eau	<i>na</i>	1
Maison	0.8	0
Route	0.8	0

TAB. 4.10 – Comparaison entre les méthodes de bases et la méthode de fusion de classifications.

Classe	Méthode	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Maison	Exemples	86,73	94,78	94,78
	Domaine	82,45	67,46	74,20
	Fusion	86,73	94,78	90,78
Végétation	Exemples	98,28	96,07	97,17
	Domaine	98,55	94,98	96,74
	Fusion	98,28	96,07	97,17
Route	Exemples	94,09	87,94	90,56
	Domaine	67,63	82,63	74,38
	Fusion	94,09	87,94	90,56
Moyenne	Exemples	93,04	92,93	92,98
	Domaine	82,88	81,69	82,28
	Fusion	93,04	92,93	92,98
Eau	Exemples	n/a	n/a	n/a
	Domaine	99,68	94,14	96,83
	Fusion	99,68	94,14	96,83

d'erreurs. Ces classes ont une pondération de 0,8 pour la classification basée sur les exemples pour tenir compte de la moindre précision sur ces deux classes comparées aux autres.

Le résultat de l'application de la méthode est illustré à la figure 4.8. Il est à noter que la classification basée connaissances du domaine rajoute quelques régions de végétation qui n'ont pas été détectées par la méthode basée sur les exemples. La table 4.10 compare les résultats de la méthode de fusion et de ses méthodes de base. La méthode de fusion obtient toujours les meilleurs résultats. Sur les classes *maison*, *végétation* et *route* les résultats sont identiques à la classification à l'aide d'exemples. Enfin pour la classe *eau*, les résultats sont indentiques à la classification à l'aide des connaissances du domaine. La méthode de fusion de classifications permet donc d'utiliser les meilleure parties de chacune des classifications.

4.4.5 Conclusion

À cette section, nous avons présenté une méthode de combinaison de résultats de méthodes de classification. Contrairement aux méthodes de combinaison usuelles, notre méthode porte sur la fusion de résultats de classifications hétérogènes au sens des classes recherchées. Comme un certain nombre de ces classes n'ont pas d'exemples associés, les méthodes classiques, qui se basent sur des exemples, ne fonctionnent pas. Dans ce cas, des méthodes se basant exclusivement sur les



FIG. 4.8 – Classification issue de la fusion des résultats des méthodes par exemples et par connaissances du domaine.

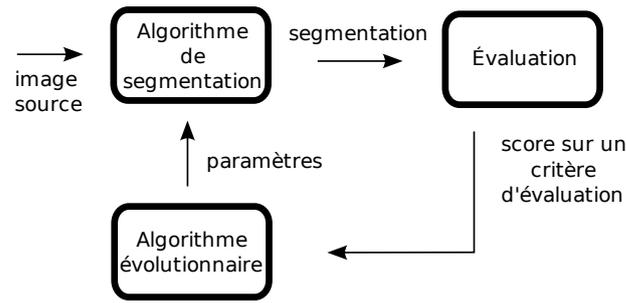


FIG. 4.9 – Illustration de l’algorithme d’optimisation des paramètres de segmentation.

connaissances du domaines peuvent être utilisées. Inversement, les connaissances du domaines peuvent fournir des classifications de moindre qualité qu’en utilisant des exemples. Dans ces cas, l’utilisation de méthode de classification basées sur les exemples est adéquate.

Pour résoudre ce problème, nous proposons de laisser l’expert définir, a priori, quelle méthode de classification utiliser pour chaque classe. Comme nous l’avons vu, cette méthode permet de fusionner des résultats très complémentaires issus d’une classification basée sur des exemples et une autre basée sur des connaissances de l’expert.

4.5 Remise en cause de l’étape de segmentation par la classification

Les section précédentes traitaient de méthodes de classification basée sur une segmentation qui n’était pas remise en cause. Au chapitre 3, nous avons vu des méthodes basées sur des exemples qui permettaient d’obtenir une segmentation plus proche des attentes des utilisateurs. Néanmoins, de tels exemples ne sont pas toujours disponibles.

Dans cette section, nous proposons d’utiliser une classification par connaissances du domaine afin d’améliorer la qualité de la segmentation. Il s’agit donc de permettre à l’étape de classification de remettre en cause la segmentation.

Nous allons commencer par présenter la méthode [43, 44] avant de faire une présentation sur l’image de test.

4.5.1 Présentation de la méthode

La méthode se base sur celle présentée à la section 3.2 dont le processus est rappelé à la figure 4.9. La différence se situe sur la fonction d’évaluation. Dans notre cas, nous considérons que nous ne disposons pas d’exemples.

Nous proposons ici d’utiliser une classification basée sur les connaissances du domaine (voir section 4.3). L’avantage de cette méthode est qu’elle ne nécessite pas d’exemple. Comme nous ne pouvons pas utiliser les mesure classiques d’évaluation qui utilisent des exemples, nous introduisons une autre mesure de qualité de la classification.

Nous allons utiliser comme fonction d’évaluation le pourcentage de la surface de l’image qui est identifiée par la classification. Soit \mathcal{X} les régions d’une segmentation et \mathcal{X}_o les régions identifiées par la classification ($\mathcal{X}_o \subseteq \mathcal{X}$). Le pourcentage de la surface de l’image reconnue par la classification est défini par :

$$\text{Reco}(\mathcal{X}, \mathcal{X}_o) = \frac{\sum_{R \in \mathcal{X}_o} \text{Aire}(R)}{\sum_{R \in \mathcal{X}} \text{Aire}(R)} \quad (4.18)$$

avec $\text{Aire}(R)$ étant la surface de la région R . La surface des régions identifiées a été préférée à leur nombre pour évaluer le résultat. En effet, une segmentation peut produire de nombreuses petites régions qui n'ont pas de signification sémantique et qui ne sont donc pas reconnues par la méthode de classification. Ces petites régions peuvent perturber un calcul basé sur le nombre de régions. Une grande partie de l'image peut être reconnue si de grandes régions (e.g. étendue végétale) sont reconnues par la classification. Ce résultat peut être faussé si de petites régions sans sémantique sont présentes dans le reste de l'image segmentée.

Le critère basé sur la surface des régions permet de quantifier la qualité de la segmentation par rapport à la classification basée sur les connaissances du domaine. L'augmentation de ce critère signifie que les régions construites par la segmentation correspondent de mieux en mieux à la description des objets géographiques présents dans la base de connaissances. En maximisant ce critère nous nous assurons de construire une segmentation en accord avec les connaissances du domaine fournies par l'expert.

Il est à noter que pour que cette méthode fonctionne, il est essentiel de mettre le seuil `minScore` de la méthode de classification basée sur les connaissances du domaine à 1. Ce paramétrage oblige la méthode à ne reconnaître que les régions qui y correspondent parfaitement. Réduire cette contrainte aurait pour effet de permettre la classification de plus de régions, mais réduirait la confiance que l'on peut exprimer sur une telle classification.

4.5.2 Expérimentations

Une série d'expérimentations a été menée sur l'image urbaine déjà utilisée tout au long du chapitre 3. L'objectif est de montrer la pertinence de l'approche.

La figure 4.10 (a) présente un extrait de l'image fusionnée. Les figures 4.10 (b) (c) (d) (e) et (f) présentent des extraits de segmentations (avec le contour des régions en blanc) avec les paramètres obtenus au cours d'une évolution respectivement aux générations 1, 2, 7, 12 et 15. Nous observons une réduction de la sous segmentation (trop peu de régions) au cours des générations et une amélioration de la construction des régions, la superficie de l'image reconnue par la méthode de classification augmentant. Pour valider ces résultats, nous avons évalué la qualité des segmentations obtenues avec les exemples disponibles de trois classes, *maison*, *végétation* et *route*.

Nous avons comparé notre critère d'évaluation (surface reconnue par la méthode de classification) par rapport à trois critères classiques pour évaluer une segmentation ou une classification, les critères Janssen, Feitosa (décrit à la section 3.1.2) et le rappel (décrit à la section 4.1.3). Pour chacun des critères une moyenne est calculée sur l'ensemble des régions fournies par l'expert. Le but de cette analyse est de vérifier que l'amélioration de notre critère conduit bien à une amélioration de la segmentation. Au cours d'une évolution, nous avons évalué chaque individu en fonction de ces différents critères. Ainsi, 145 paramétrages possibles ont été évalués (tous les paramétrages testés par l'algorithme évolutionnaire). La figure 4.11 (a) présente les nuages de points avec l'évaluation sur notre critère en abscisse et le rappel en ordonnée, la figure 4.11 (b) utilise l'indice de Janssen en ordonnée, et la figure 4.11 (c) l'indice de Feitosa. Nous pouvons constater que dans les trois cas, les nuages de points indiquent des corrélations entre l'évaluation et ces indices. Les coefficients de corrélation sont de 0,97 pour l'évaluation avec le rappel, 0,98 avec l'indice Janssen et -0,93 avec l'indice Feitosa. Ces résultats montrent qu'optimiser notre

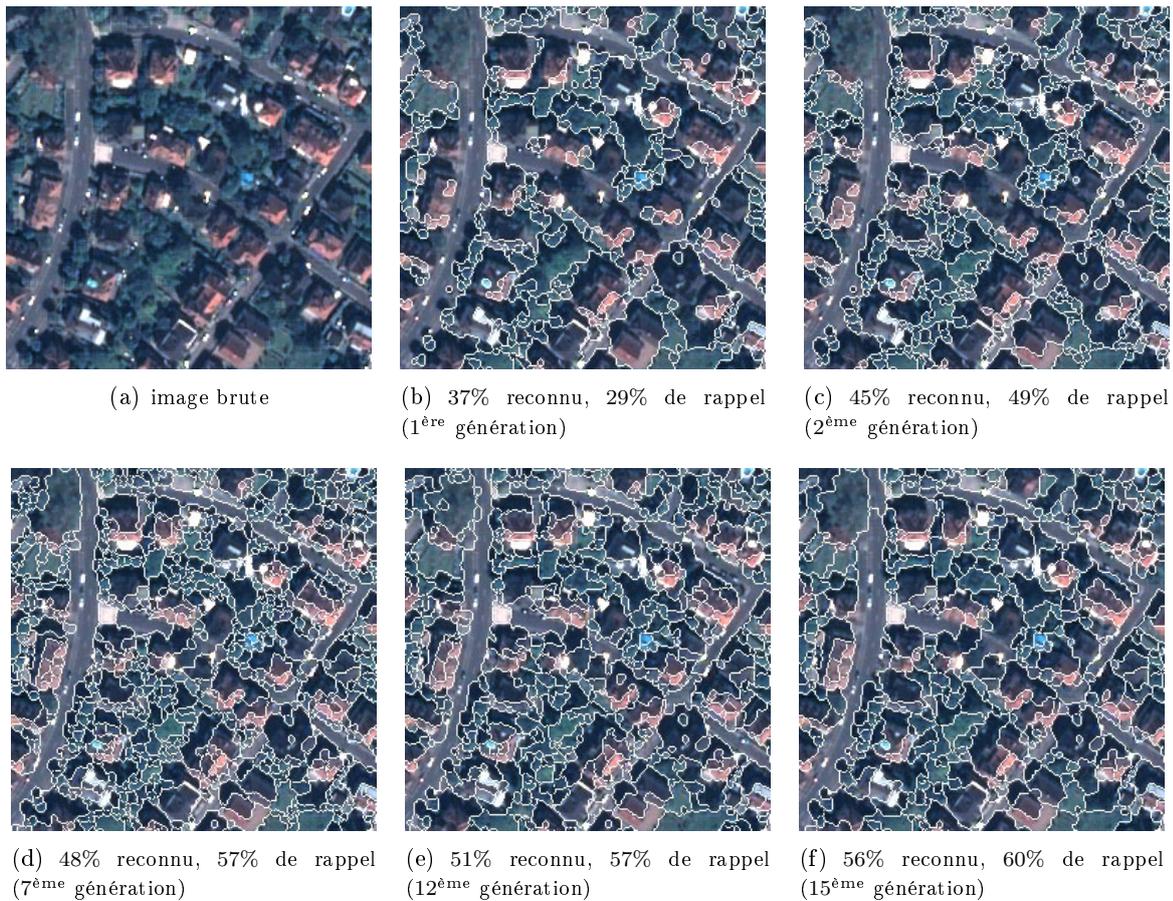


FIG. 4.10 – Extraits de segmentations obtenues à différentes générations au cours d’une évolution. Le contour des régions est affiché en blanc.

critère est pertinent et permet d’effectuer des segmentations qui donneront de bons résultats de classification.

Enfin, le tableau 4.11 présente les valeurs de ces indices pour les segmentations avec les paramètres trouvés au cours des générations 1, 2, 7, 12 et 15 lors d’une évolution (la qualité n’évoluant plus significativement après). On constate que les différentes mesures d’évaluation de classification (rappel et F-mesure) sont optimisées. Pour les mesures d’évaluation de segmentation (Janssen et Feitosa), on a une optimisation sur le début du processus, mais sur la fin les valeurs se dégradent légèrement. On peut en conclure qu’une segmentation optimale pour la classification n’est pas forcément identique à une segmentation optimale au sens des critères d’évaluation de segmentations.

À la fin du processus évolutif, la méthode de classification obtient un score de rappel de 59,6%. Ceci est à comparer avec le score de rappel de 53,8% qu’a obtenu la segmentation de basse qualité pour un seuil $\text{minScore} = 1$ dans la section 4.3.3. Cette segmentation était obtenue avec la même méthode de segmentation que celle utilisée ici, mais avec des paramètres choisis par l’utilisateur. De même pour la précision, nous obtenons ici un score de 87,2% contre 85,3% et un score de F-Mesure de 70,8% contre 66,0%. La méthode proposée permet donc d’obtenir un résultat meilleur sans demander à l’utilisateur de trouver manuellement le bon paramétrage .

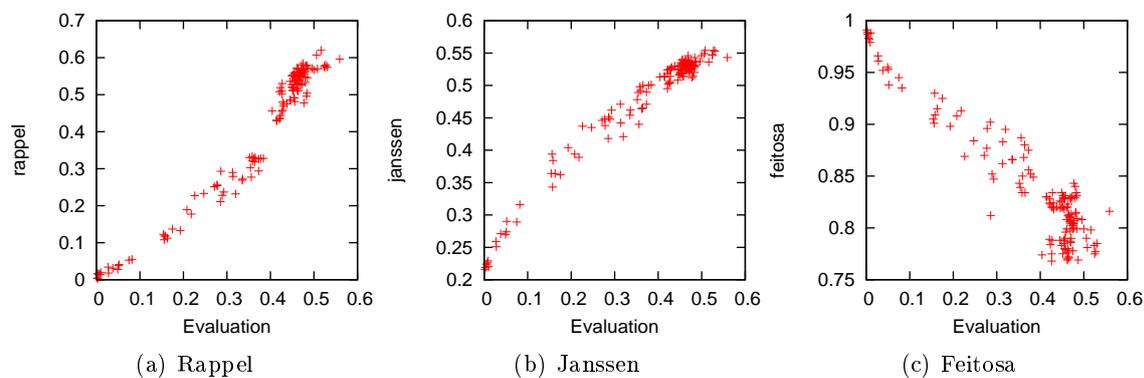


FIG. 4.11 – Évolution des fonctions d'évaluation pour 145 individus ordonnés par le taux de reconnaissance de la méthode de classification basée sur les connaissances du domaine.

TAB. 4.11 – Résultat de l'évaluation de la méthode sur les régions d'exemples pour cinq générations. Les valeurs sont exprimées en pourcentages à l'exception de Janssen et de Feitosa

Génération	Reconnaissance	Rappel	F-Mesure	Kappa	Janssen	Feitosa
1 ^{ère}	37,3	29,4	42,4	6,0	0,490	0,856
2 ^{ème}	44,5	48,5	62,5	16,8	0,525	0,826
7 ^{ème}	47,7	56,6	68,9	21,6	0,529	0,800
12 ^{ème}	50,8	57,0	68,8	26,2	0,554	0,781
15 ^{ème}	55,9	59,6	70,8	39,4	0,543	0,816

4.5.3 Conclusion sur la remise en cause de la segmentation

Dans cette section, nous avons présenté un mécanisme permettant de guider un processus automatique d'interprétation d'images se basant uniquement sur les connaissances du domaine. Ce processus se base sur une remise en cause de la segmentation par l'étape de classification. La méthode utilise un algorithme évolutionnaire pour trouver les paramètres à utiliser pour la méthode de segmentation. L'algorithme évolutionnaire utilise la surface détectée par une classification par connaissances du domaine pour juger de la pertinence des paramètres. Ceci est utilisé à la place des évaluations de segmentation utilisant des exemples évitant ainsi le besoin de créer des exemples, tâche pouvant être coûteuse en temps.

Les résultats montrent que la méthode est capable de fournir des segmentations plus intéressantes pour obtenir une classification par connaissances du domaine qu'un utilisateur qui se lassera plus vite de faire des essais/erreurs. Dans cette méthode, une fois la base de connaissances créée, tout le processus se fait sans besoin d'interaction avec l'utilisateur.

4.6 Conclusion sur la classification des régions

Dans ce chapitre, nous avons étudié deux méthodes de classification des régions issues d'une segmentation. La première méthode est basée sur des exemples et pratique un apprentissage sur ces exemples pour classer des régions. Nous avons illustré cette approche par plusieurs algorithmes de classification aux fonctionnements hétérogènes : k plus proches voisins, réseaux de neurones et bayésien naïf. L'algorithme des k plus proches voisins a montré une performance meilleure par rapport aux autres méthodes et surtout une robustesse plus grande au jeu de données d'apprentissage. En effet, plus que la méthode de classification utilisée, la méthode de création du jeu d'apprentissage est primordiale. L'utilisation directe des exemples, méthode la plus simple, n'est pas la meilleure et peut induire en erreur les méthodes de classification basées sur des modèles. Nous avons vu que parmi les méthodes possibles, la plus pertinente semble être de créer un jeu d'apprentissage avec les exemples bruts et les régions qui correspondent le plus à celles-ci dans la segmentation (méthodes Janssen et couverture). Cette combinaison permet d'induire des modèles qui ne fonctionnent pas seulement sur des données parfaites, mais aussi sur les données fournies par une segmentation, généralement de qualité moindre que celles de l'expert.

Nous avons ensuite vu une méthode de classification basée uniquement sur les connaissances du domaine. Cette approche permet d'éviter la création d'exemples qui est une tâche fastidieuse pour l'utilisateur. Nos expérimentations ont mis en avant la variabilité des résultats en fonction de la qualité de la segmentation. Afin d'obtenir de meilleurs résultats, il est indiqué de reconnaître aussi des régions qui ne correspondent pas tout à fait aux connaissances du domaine (paramètre `minScore`) afin de palier aux imperfections de la segmentation. Selon les besoins de l'utilisateur, il est donc possible de privilégier le rappel ou la précision.

Après avoir présenté ces deux familles de méthodes de classification, nous avons constaté une complémentarité de ces familles et avons proposé une méthode pour combiner leurs résultats afin de tirer avantage de cette complémentarité. Cette méthode permet de faire une classification en fournissant des exemples uniquement pour les classes difficilement séparables.

Enfin, nous avons proposé d'utiliser la méthode de classification basée sur les connaissances du domaine afin de remettre en cause, et d'améliorer, la segmentation. Les résultats ont montré que cette méthode permettait d'obtenir des résultats plus intéressants par rapport à la même méthode de segmentation mais paramétrée par l'utilisateur.

Les résultats obtenus dans ce chapitre mettent en avant l'importance de la segmentation et la

pertinence de remettre en cause la segmentation afin d'obtenir une meilleure classification. Nous avons aussi vu que selon les classes et les connaissances a priori sur elles (exemples et/ou connaissances du domaine), la méthode de classification à utiliser (ou même son paramétrage) n'est pas forcément la même. Nous allons à présent voir une approche différente de celle de segmentation puis de classification vue durant ce chapitre et le précédent, l'approche par détecteurs.

Chapitre 5

Détection : Extraction des objets d'une classe

*Il faut toujours se réserver le droit de rire le lendemain
de ses idées de la veille.*
Napoléon Bonaparte

5.1 Introduction

La détection consiste à rechercher des objets d'intérêt dans une image. On appelle détecteur un algorithme capable, à partir d'une image, d'associer un ensemble d'instances de la classe qu'il recherche. Dans ce chapitre, nous présentons l'intérêt que peut avoir une approche utilisant un ensemble de détecteurs par rapport à l'approche plus classique consistant en une segmentation globale suivie d'une classification des régions, présentée dans les chapitres 3 et 4.

La littérature contient une multitude de méthodes [65, 75, 73, 104, 128, 133] pour rechercher les objets de certaines classes d'objets particulières (route, immeubles, ...). Ces méthodes sont très spécifiques et intègrent généralement les connaissances implicites de leurs concepteurs sur les classes d'objets recherchées. Ces connaissances sont ensuite difficiles à extraire si l'on souhaite les modifier ou les améliorer.

Nous proposons ici une approche générique permettant d'extraire des détecteurs pour chaque classe d'objets à partir d'une base de connaissances. Nous séparons ainsi clairement les connaissances de l'algorithmique. L'objectif est de permettre à l'expert du domaine d'exprimer facilement ses connaissances sans être spécialiste en traitement d'image ou en extraction de connaissances.

Enfin, nous aborderons la communication entre plusieurs détecteurs. Les détecteurs peuvent en effet être vus comme des agents capables de créer de l'information. Il apparaît naturel que cette information puisse être utilisée par d'autres détecteurs. Nous verrons notamment le cas des détecteurs d'objets composites (par exemple un îlot résidentiel est composé de maisons, routes et végétation).

5.1.1 Interprétation par ensemble de détecteurs

Un détecteur d_i , spécialisé pour la classe C_i , peut être formalisé comme une fonction qui à un pixel p_j associe une valeur binaire issue de $\mathbb{B} = \{\text{vrai}, \text{faux}\}$ signifiant l'appartenance ou la non appartenance du pixel p_j à la classe C_i :

$$d_i(p_j) = \begin{cases} \text{vrai} & \text{si } p_j \text{ appartient à } C_i \\ \text{faux} & \text{sinon} \end{cases} \quad (5.1)$$

Par extension, en appliquant la fonction à tous les pixels de l'image, il est possible d'obtenir la liste des composantes connexes, les régions, appartenant à C_i . On note ainsi \mathcal{X}_i l'ensemble des objet que le détecteur d_i propose comme étant de la classe C_i .

Chaque détecteur permet la détection d'une seule classe. Afin de classifier l'image, il faut utiliser un détecteur par classe d'intérêt dans l'image. On obtient un ensemble de détecteurs $\mathcal{D} = \{d_i \mid C_i \in \mathcal{C}\}$ (avec \mathcal{C} l'ensemble des classes d'objets).

Il est à noter que certains pixels de l'image peuvent n'appartenir à aucune classe. Plus formellement, il peut exister un pixel p_j tel que $\forall i \mid C_i \in \mathcal{C}, d_i(p_j) = \text{faux}$. De même, un pixel peut être considéré comme appartenant à plusieurs classes. C'est-à-dire, qu'il peut exister un pixel p_j tel que $d_a(p_j) = \text{vrai}$ et $d_b(p_j) = \text{vrai}$ avec $a \neq b$ et $C_a, C_b \in \mathcal{C}$.

Le fait que des pixels puissent n'être associés à aucune classe a déjà été rencontré pour la classification d'objets par connaissances du domaine à la section 4.3. C'est naturel car les images, notamment de télédétection, peuvent contenir des objets non recherchés par l'utilisateur ou des pixels mixtes (pixels appartenant partiellement à plusieurs classes) ou de bruit.

De même, nous pouvons avoir deux objets x_a et x_b tels que $x_a \in \mathcal{X}_i$ et $x_b \in \mathcal{X}_j$ avec $x_a \neq x_b$ et $x_a \cap x_b \neq \emptyset$. Ces deux objets de classes différentes se chevauchent donc partiellement. Cet effet peut provenir d'une erreur d'un des algorithmes de détection ou d'une difficulté intrinsèque à l'image pour fixer une frontière entre deux objets. Cela peut arriver dans le cas de pixels mixtes. Comme les différents détecteurs ne tiennent pas compte les uns des autres, ils ne cherchent pas à résoudre ce type d'incohérences.

5.1.2 Comparaison avec l'approche par segmentation suivie d'une classification

La différence principale entre l'approche par ensemble de détecteurs et celle de la segmentation suivie d'une classification des régions est le type de séparation effectué dans les traitements. Cette différence est illustrée à la figure 5.1. L'approche segmentation puis classification opère une séparation séquentielle, c'est-à-dire qu'elle divise en deux le traitement, d'un côté la segmentation et de l'autre la classification. De son côté, l'approche par ensemble de détecteurs propose une séparation parallèle, i.e. le traitement de chaque classe est séparé. Il n'est pas fait mention d'un éventuel découpage au niveau du traitement qui peut être différent pour chaque classe.

L'avantage de l'approche par détecteurs est justement la possibilité d'appliquer un traitement spécifique à chaque classe. Dans l'approche par segmentation puis classification, toutes les classes sont traitées par le même algorithme. C'est pourquoi, il est difficile de prendre en compte les spécificités de chaque classe, ou bien les connaissances particulières disponibles. Nous avons vu, à la section 4.4, une utilisation de la connaissance de l'expert sur les classes pour choisir quel type d'algorithme de classification utiliser. Cela reste une utilisation limitée car contrainte à des algorithmes non spécifiques et n'intervenant que pour la classification.

L'approche par ensemble de détecteurs a aussi un avantage calculatoire. En effet, le traitement de chaque classe se faisant à part, le processus de détection est facilement parallélisable. Si l'on suppose avoir à disposition un nombre d'unités de calcul supérieur au nombre de classes recherchées dans l'image, la durée d'exécution sera celle de l'application du détecteur prenant le plus de temps. De plus, la complexité algorithmique d'un détecteur peut être au pire égale à l'ensemble du processus de segmentation puis classification (en utilisant les mêmes méthodes). Elle dépend cependant des méthodes utilisées. En effet, la détection d'une seule classe est un

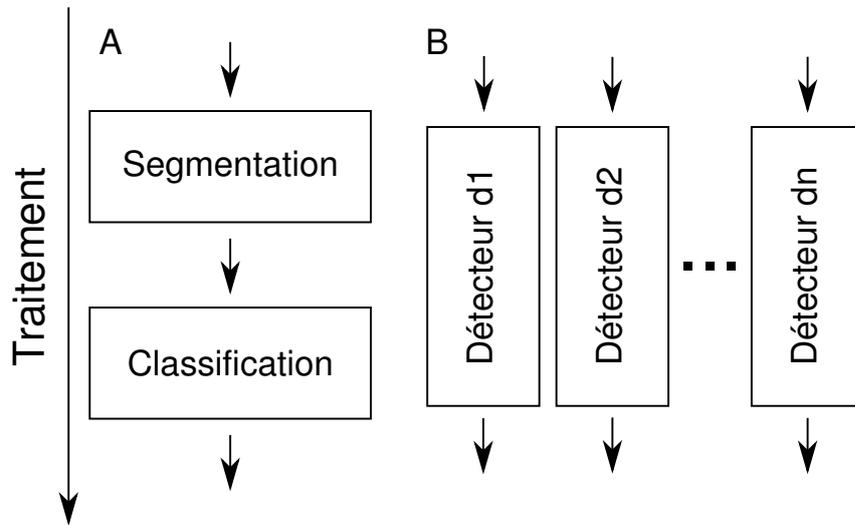


FIG. 5.1 – Représentation conceptuelle de l’approche segmentation puis classification (A) et de l’approche par ensemble de détecteurs (B).

problème plus simple que la détection d’une multitude de classes. Il est de plus possible d’utiliser des détecteurs dont la complexité algorithmique est corrélée à la complexité de détection de la classe. Par exemple, la végétation est une classe facilement détectable et ne demande pas un détecteur algorithmiquement très complexe (nous le verrons à la prochaine section).

Enfin, cette approche a un avantage conceptuel. Il est en effet plus simple de se focaliser sur la création d’un détecteur pour une classe que de créer un algorithme capable de découvrir toutes les classes d’intérêt.

Le seul inconvénient de l’approche par détecteurs, est que deux régions de classes différentes peuvent se chevaucher partiellement. Dans le cas de la segmentation suivie d’une classification, par définition, cela ne peut pas se produire, chaque pixel n’appartenant qu’à une région après la segmentation. Cet inconvénient est néanmoins à relativiser. En effet, si l’étape de segmentation n’affecte qu’une seule région à chaque pixel, ce n’est pas forcément la région la plus appropriée qui est choisie. Comme nous l’avons vu au chapitre 3, une image peut être sous-segmentée, c’est-à-dire qu’une région peut couvrir deux objets d’intérêt de classes différentes.

Nous pouvons donc conclure que malgré cet inconvénient, l’approche par détecteurs dispose de nombreux atouts qui doivent permettre d’obtenir de meilleures performances qu’une approche par segmentation puis classification. Nous allons à présent présenter une application de cette approche.

5.2 Extraction de détecteurs spécifiques à partir de la base de connaissances

Nous présentons ici une approche d’extraction de détecteurs à partir d’une base de connaissances. Nous commençons par expliquer le principe sous-jacent à cette approche qui est le *design pattern* dit de la séparation des préoccupations [61]. Nous détaillons ensuite l’algorithme d’extraction de détecteurs et finissons par l’évaluation de cet algorithme.

5.2.1 Séparation des préoccupations

Dans cette section, nous nous intéressons à la création des détecteurs. De nombreuses méthodes spécifiques existent pour détecter des objets particuliers dans une image, tels que la route [104, 128, 133] ou le bâti [65, 73]. Ces méthodes intègrent les connaissances implicites qu'ont leurs concepteurs sur les types d'objets recherchés. Cette connaissance n'a pas été modélisée et se retrouve imbriquée dans l'algorithmique des détecteurs ainsi créés. Il en résulte que ces connaissances sont ensuite difficiles à retrouver si l'on souhaite améliorer ou adapter l'algorithme.

Si l'on prend le cas de deux détecteurs de routes créés par des concepteurs différents, il est très difficile d'exprimer les connaissances qui ont été utilisées pour chaque algorithme. De ce fait, il est difficile de fusionner les connaissances de ces deux détecteurs pour en former un plus efficace. La seule solution est de faire une fusion des résultats de ces détecteurs en espérant obtenir un meilleur résultat final (voir section 4.4 relative à la fusion de résultats).

D'un autre côté, afin de modéliser ces connaissances implicites, l'utilisation d'ontologies ou de bases de connaissances devient de plus en plus courante, par exemple dans les systèmes d'informations géographiques [42] ou pour l'analyse d'image [11]. L'objectif est d'obtenir une spécification abstraite, une vue simplifiée du monde représentée dans un but précis [53]. Avec la base de connaissances définie au chapitre 2, nous disposons d'un ensemble de classes d'intérêt, leurs caractéristiques et des relations entre elles. La plupart des méthodes [35, 82, 89, 98] formalisent les classes pouvant être présentes dans une image puis proposent une analyse sémantique de celle-ci en cherchant à identifier les objets de ces classes dans l'image.

C'est pourquoi, nous proposons une méthode d'extraction de détecteurs pour des objets spécifiques à partir d'une base de connaissances. L'originalité de cette approche est d'utiliser le *design pattern* de séparation des préoccupations [61]. Alors que dans les solutions existantes la connaissance est mêlée au processus d'identification, notre approche permet de séparer la partie connaissance de la partie algorithmique.

Si l'expert du domaine a pour tâche le remplissage d'une base de connaissances, le développement de la méthode d'extraction de détecteurs est la tâche de l'expert en algorithmique. La méthode que nous proposons ici s'inspire des travaux menés dans les classifications de régions à l'aide de connaissances (section 4.3). Les connaissances sont vues comme des contraintes qui permettent d'affirmer que des parties de l'image ne peuvent appartenir à une classe donnée. En effet, une zone de l'image qui ne valide pas une contrainte ne peut appartenir à la classe du détecteur. Une fois toutes les contraintes appliquées, il ne reste plus que les zones pouvant appartenir à cette classe.

5.2.2 Structure générale des détecteurs

Les détecteurs que nous allons créer ont une structure de base identique. Ils commencent par considérer que toute l'image correspond à la classe qu'ils recherchent. Ensuite, une succession de filtres sont appliqués afin de retirer les zones de l'image qui ne correspondent pas à la classe recherchée. Une fois ces filtres appliqués, il ne reste plus que les zones de l'image pouvant appartenir à la classe recherchée. Ce processus est illustré à la figure 5.2.

Les filtres sont extraits de la base de connaissances, en prenant les connaissances comme des contraintes qui doivent être vérifiées par les objets de la classe recherchée. La problématique consiste à traduire les différentes connaissances en des contraintes à respecter et enfin en des filtres opérationnels.

Nous avons vu au chapitre 2 que les connaissances sont souvent exprimées sous la forme d'intervalles sur des attributs. Dans ce cas il est fort aisé de développer un filtre associé. En effet,

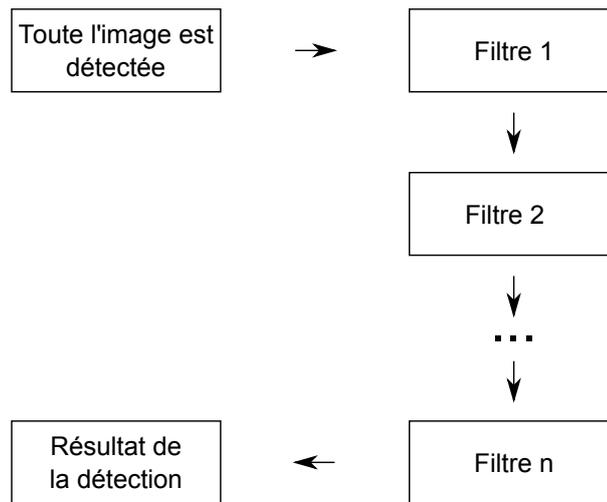


FIG. 5.2 – Structure de base des détecteurs générés.

il suffit de vérifier si la valeur pour cet attribut est bien dans l'intervalle permis. Si ce n'est pas le cas, il faut supprimer la partie de l'image. Nous allons étudier en détail les différents cas.

5.2.3 Description des filtres pouvant composer un détecteur

Pour le niveau spectral, deux méthodes de filtrage sont possibles selon les connaissances disponibles. La première méthode, détaillée à l'algorithme 2, utilise les intervalles des valeurs spectrales admises pour la classe recherchée. Pour chaque pixel, si ses valeurs spectrales ne sont pas dans l'intervalle défini, ce pixel est supprimé de la liste des pixels détectés. La seconde méthode est utilisable si des exemples sont disponibles et que l'expert a indiqué qu'il est possible d'en extraire des connaissances spectrales. Dans ce cas, il est possible d'effectuer une classification supervisée floue basée pixel en utilisant les attributs spectraux. Le détail du processus de filtrage est donné à l'algorithme 3. La fonction `classificationFloue(p , e_1 , e_2)` renvoie la probabilité d'appartenance du pixel p à la classe représentée par les exemples e_1 , les exemples e_2 étant utilisés comme exemples négatifs. Le paramètre seuil permet de définir à partir de quelle probabilité d'appartenance à la classe recherchée on conserve le pixel. Nous prendrons ici comme valeur arbitraire 0,4, c'est-à-dire que même si un pixel semble appartenir plus à la classe représentée par les exemples e_2 , il peut quand même être conservé dans celle représentée par e_1 . La raison de ce choix est d'éviter de supprimer des pixels alors qu'ils appartiennent à la classe, la classification basée pixels pouvant commettre des erreurs. Dans les deux cas de filtrage spectral, nous appliquons une fermeture binaire, c'est-à-dire que nous remettons dans la liste des pixels détectés toutes les zones connexes non détectées qui sont plus petites qu'une forme prédéfinie, ici un carré de 3×3 pixels, ceci afin de gérer le bruit poivre et sel ou les petits objets sans intérêt présents sur l'image.

Pour la forme, nous ne pouvons utiliser directement les intervalles. En effet, nous avons à notre disposition une image binaire (pixels détectés comme étant de la classe recherchée et ceux non détectés). Nous ne pouvons utiliser directement les composantes connexes présentes dans l'image. Différentes composantes connexes représentant des objets peuvent être reliées par des pixels détectés par erreur comme appartenant à la classe recherchée. Dans ce type de situations, utiliser les attributs des composantes connexes ne permet pas de détecter ces objets. Nous proposons de

Algorithme 2 Filtrage spectral par contraintes.

Entrée : C_c la classe recherchée

Entrée : d_c le détecteur de la classe C_c

Entrée : \mathcal{P} l'ensemble des pixels de l'image à interpréter

Entrée : \mathcal{A}_s l'ensemble des attributs spectraux avec un intervalle spécifié dans la base de connaissances pour la classe C_c .

Sortie : d_c le détecteur de la classe C_c après prise en compte du filtre spectral

pour tout $p \in \mathcal{P}$ **faire**

$d_c(p) := \begin{cases} \text{vrai} & \text{si } \forall a \in \mathcal{A}_s, \text{ valeur}(p, a) \in [\min(C_c, a), \max(C_c, a)] \\ \text{faux} & \text{sinon} \end{cases}$

fin pour

Algorithme 3 Filtrage spectral par classification floue supervisée.

Entrée : C_c la classe recherchée

Entrée : d_c le détecteur de la classe C_c

Entrée : \mathcal{P} l'ensemble des pixels de l'image à interpréter

Entrée : \mathcal{C}_c l'ensemble des classes dont les exemples doivent être utilisés pour détecter C_c .

Sortie : d_c le détecteur de la classe C_c après prise en compte du filtre spectral

pour tout $p \in \mathcal{P}$ **faire**

pour tout $C_a \in \mathcal{C}_c \setminus C_c$ **faire**

si $\text{classificationFloue}(p, \text{exemples}(C_c), \text{exemples}(C_a)) > \text{seuil}$ **alors**

$d_c(p) := \text{faux}$

fin si

fin pour

fin pour

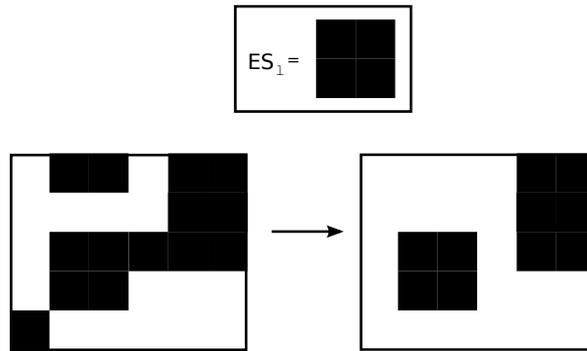


FIG. 5.3 – Illustration d’une ouverture par un élément structurant, passage de l’image de gauche à celle de droite. Cette ouverture permet de supprimer les régions trop petites et de séparer des régions liées par erreur. Le résultat est ici de deux régions.

contruire des filtres sous forme de traitements morphologiques sur l’image binaire.

Par exemple, les classes de bâti sont souvent de forme rectangulaire. On peut déduire ceci de l’attribut *fitting OBB* (qui mesure la similarité des objets avec des rectangles) : si la valeur minimale est très proche de 1, cela signifie que les objets de cette classe ont des formes de rectangle. Nous avons donc défini le traitement morphologique suivant pour découvrir ces objets. Notons l la longueur minimale d’un des côtés d’un objets de cette classe. On définit l’élément structurant ES_l comme un carré de coté l . On applique une ouverture binaire de cet élément structurant afin de ne conserver que les zones de l’image pouvant contenir cet élément structurant. On supprime ainsi les zones de l’image qui ne vérifient pas cette contrainte de rectangularité. Une illustration de ce processus est proposée à la figure 5.3. Afin de conserver les zones carrées quelle que soit leur orientation, on applique en parallèle plusieurs ouvertures avec une rotation différente de l’élément structurant. L’unification de ces résultats se fait selon la règle suivante : si un pixel est conservé dans au moins une des ouvertures, alors il reste dans le résultat final.

Enfin, il reste les attributs de régions. Pour les traiter, une segmentation de l’image est effectuée, chaque composante connexe de l’image de détection correspondant à une région. Des attributs de régions peuvent alors être calculés. Si la valeur d’un des attributs ne correspond pas à l’intervalle autorisé pour la classe donnée, la composante connexe est retirée de la détection.

À la fin de l’application de ces différents filtres, l’image binaire ne contient plus que les zones considérées par le détecteur comme étant des objets de la classe recherchée.

Il est à noter que l’ordre d’exécution des filtres a un impact direct sur le résultat. Il est évident qu’appliquer un filtre basé sur des attributs de régions en premier n’est pas pertinent. En effet, pour le premier filtre, toute l’image est détectée comme appartenant à la classe recherchée. Il n’y a donc qu’une région qui est l’image entière.

On définit donc trois types de filtre selon leur priorité d’exécution : filtre de pixel puis filtre de forme et enfin filtre de région. Les filtres de pixel ne travaillent que sur des pixels pris isolément (filtres spectraux) et sont appliqués en premier. Comme ils n’utilisent pas le voisinage du pixel, ils n’ont pas besoin que certains pixels soient déjà filtrés. Le second type de filtres qui peut être appliqué est celui des filtres de forme. Ces filtres nécessitent qu’une bonne partie des pixels n’appartenant pas à la classe recherchée soient supprimés mais peut s’accomoder d’un bruit assez important (voir figure 5.3). Enfin, les filtres de régions doivent être appliqués en dernier. En effet, une trop grande quantité de bruit pourrait agréger plusieurs régions ensemble et donc modifier

TAB. 5.1 – Liste des filtres développés pour la création des détecteurs.

Nom	Type	Connaissance utilisée	Implémentation
Intervalle spectral	Pixel	Intervalles spectraux	Chaque pixel dont les valeurs spectrales ne correspondent pas
Classification pixel	Pixel	Exemples spectraux	Un algorithme de classification supervisée (ici un 5 plus proches voisins) est utilisé pour détecter les pixels pouvant appartenir à la classe recherchée
Rectangularité	Forme	Connaissances de forme	Des ouvertures à partir d'éléments structurant dont les caractéristiques dépendent des connaissances sont effectuées selon plusieurs orientations
Surface	Région	Intervalle de surface	Les régions dont la surface n'est pas comprise dans l'intervalle des surfaces admises pour la classe recherchée sont supprimées

fortement les valeurs des attributs de régions. Par exemple à la figure 5.3, les deux régions de droite étaient fusionnées avant l'application du filtre.

Un résumé des filtres développés est présenté à la table 5.1. Cette liste n'est pas exhaustive et d'autres filtres peuvent sans doute être extraits à partir de la base de connaissances. Nous nous limitons pour l'instant à ceux-ci car ils nous semblent suffisamment complémentaires. L'ajout de nouveaux filtres est une opération simple. Une fois une connaissance identifiée comme pouvant produire un filtre, il suffit de déterminer le type de ce filtre (pixel, forme, région) et de l'ajouter à la liste des filtres disponibles. Tant qu'un filtre ne supprime pas des pixels appartenant à la classe recherchée, l'ajout de filtre ne peut qu'améliorer la qualité du résultat. Il n'y a donc pas besoin de connaître les autres filtres. Un concepteur de filtre peut créer des filtres de manière indépendante du travail d'autres personnes que ce soit d'autres concepteurs de filtres ou des experts du domaine modélisant leurs connaissances.

Il est à noter que dans notre proposition, les connaissances sont considérées comme des filtres. L'ajout de connaissances réduit le nombre de pixels qui seront considérés comme appartenant à la classe recherchée. De ce fait, si un objet de la classe recherchée n'est pas détecté, l'ajout de connaissances ne permettra pas de corriger cette erreur. Une perspective envisagée pour résoudre ce problème consiste à permettre de transcrire certaines connaissances en étapes correctives, qui redétecteraient certains pixels qui auraient été supprimés par des filtres précédents.

5.2.4 Unification des résultats de détecteurs

Selon l'utilisation faite des résultats, il peut être nécessaire de faire une unification des différentes cartes de détection afin de fournir une classification de l'image. Nous allons utiliser ici la même méthode que celle présentée pour la fusion de classification (section 4.4).

Cette méthode consiste à attribuer une pondération à chaque résultat pour déterminer la priorité de chacun des résultats sur les autres. Le problème est ici simplifié car chaque résultat à fusionner ne comporte qu'une seule classe. Chaque pixel, détecté par au moins un détecteur, est associé à la classe dont le détecteur a la plus grande priorité.

TAB. 5.2 – Pondérations des résultats de chaque détecteur (ordonnées selon la pondération).

Détecteur	Pondération
Eau	1.00
Maison	0.95
Végétation	0.90
Route	0.85

5.2.5 Évaluation

La méthode proposée a été évaluée sur le même jeu de test que celui utilisé dans le chapitre 4 sur la classification afin de pouvoir établir des comparaisons. L'image de travail, ainsi que les exemples d'apprentissage et les exemples d'évaluation sont donnés à la figure 5.4. Pour les classes *végétation*, *route* et *maison*, nous avons défini un jeu d'apprentissage et un jeu d'évaluation. Nous nous permettons d'utiliser les exemples de la classe *eau* à la fois pour l'apprentissage et pour l'évaluation. N'ayant que trois surfaces d'eau différentes et chacune spectralement différente l'une de l'autre, il n'est pas envisageable de faire un jeu d'apprentissage et un jeu d'évaluation séparé.

Évaluation de l'unification des résultats

La table 5.3 montre les résultats obtenus pour les différentes classes. À nouveau, la moyenne ne porte que sur les 3 premières classes afin d'être comparable aux résultats du chapitre précédent. Les pondérations utilisées pour l'unification de détecteurs sont données à la table 5.2. Elles sont choisies (arbitrairement) afin de maximiser le score de F-Mesure de la fusion sur les exemples d'apprentissage.

On constate que la méthode des détecteurs privilégie la précision au détriment du rappel. Elle obtient en effet un meilleur score de précision sur les maisons et la route par rapport à la méthode de fusion des résultats des classifications par exemple et par connaissances du domaine présenté à la section 4.4. Pour la végétation et l'eau, les scores de précision sont excellents pour les deux méthodes. *A contrario*, les scores de rappel sont sensiblement moins bons sauf pour l'eau. Chacune des deux méthodes a donc ses avantages et inconvénients. Selon l'objectif recherché (le compromis précision/rappel), la méthode à utiliser peut changer.

Le résultat de l'unification des détecteurs est donné à la figure 5.5. On y constate des zones noires indiquant des pixels non étiquetés, ceci expliquant le score de rappel inférieur à la fusion des classifications. Par exemple, plusieurs maisons n'ont pas été détectées sur la partie *est* de l'image. Par comparaison avec le résultat de la fusion des classifications, on obtient une image épurée de certaines erreurs grâce aux différents filtres mis en place.

Nous allons à présent appliquer l'algorithme pour chaque concept recherché. L'évaluation sera faite par inspection visuelle et quantitative dans l'objectif de vérifier :

- la qualité de la base de connaissances : s'il manque certaines connaissances ;
- la généralité de l'algorithme : s'il prend en compte toutes les connaissances disponibles.

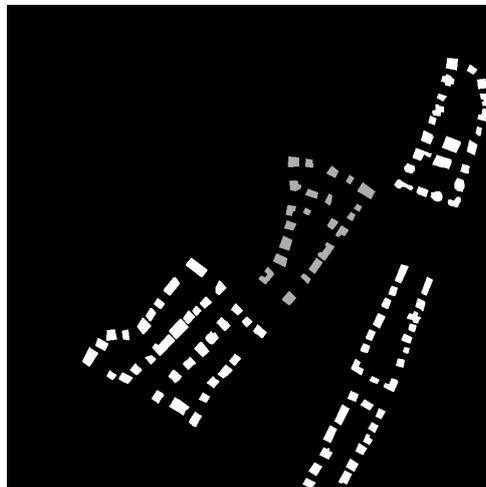
L'évaluation quantitative donnée à la table 5.4 est effectuée sur les exemples d'évaluation de toutes les classes en utilisant à chaque fois uniquement le résultat du détecteur évalué. On constate à nouveau la très bonne précision obtenue par les différents détecteurs à l'exception du détecteur de la classe *route*. Si l'on compare avec les résultats obtenus après l'unification des



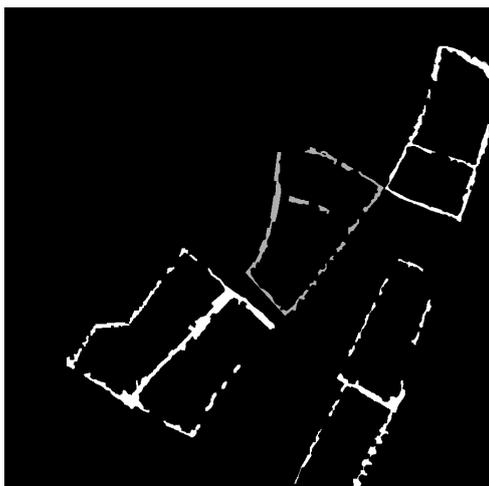
(a) Image



(b) Exemples de végétation



(c) Exemples de maisons



(d) Exemples de routes



(e) Exemples d'eau

FIG. 5.4 – Image d'évaluation de la méthode et exemples des différentes classes d'objets considérés. Pour les exemples de végétation, de maisons et de routes, les exemples en gris sont utilisés en apprentissage et ceux en blanc pour l'évaluation. Les exemples d'eau sont utilisés tant pour l'apprentissage que pour l'évaluation du fait de leur nombre limité.



■ Route ■ Végétation ■ Maisons ■ Eau

FIG. 5.5 – Résultat obtenu par unification des détecteurs (en haut) et par fusion des classifications (en bas).

TAB. 5.3 – Comparaison entre les méthodes d'ensemble de détecteurs et la méthode de fusion de classification.

Classe	Méthode	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Maison	Classification	86.73	94.78	90.78
	Détecteur	91.20	81.79	86.24
Végétation	Classification	98.28	96.07	97.17
	Détecteur	98.34	95.82	97.06
Route	Classification	94.09	87.94	90.56
	Détecteur	95.06	81.05	87.50
Moyenne	Classification	93.04	92.93	92.98
	Détecteur	94.87	86.22	90.34
Eau	Classification	99.99	94.08	96.95
	Détecteur	99.96	98.27	99.11

TAB. 5.4 – Évaluation individuelle des détecteurs.

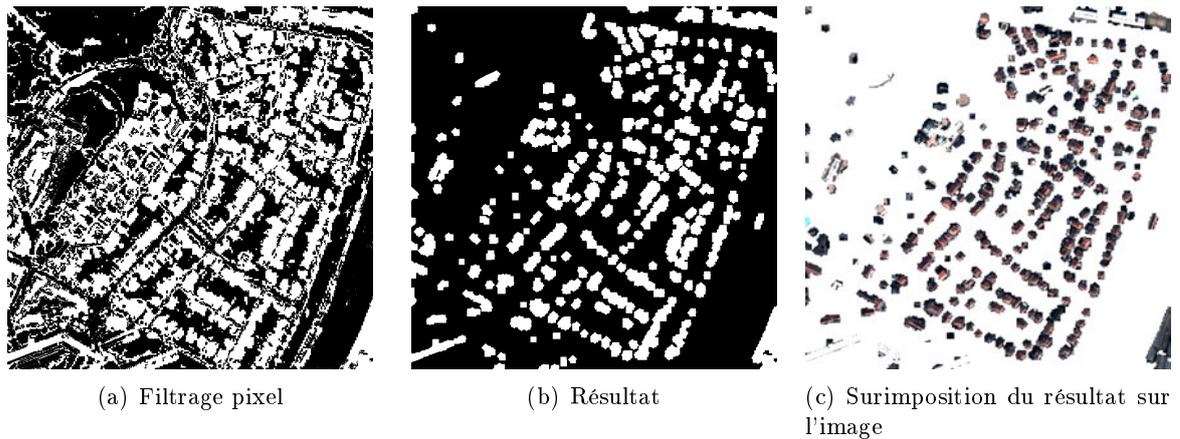
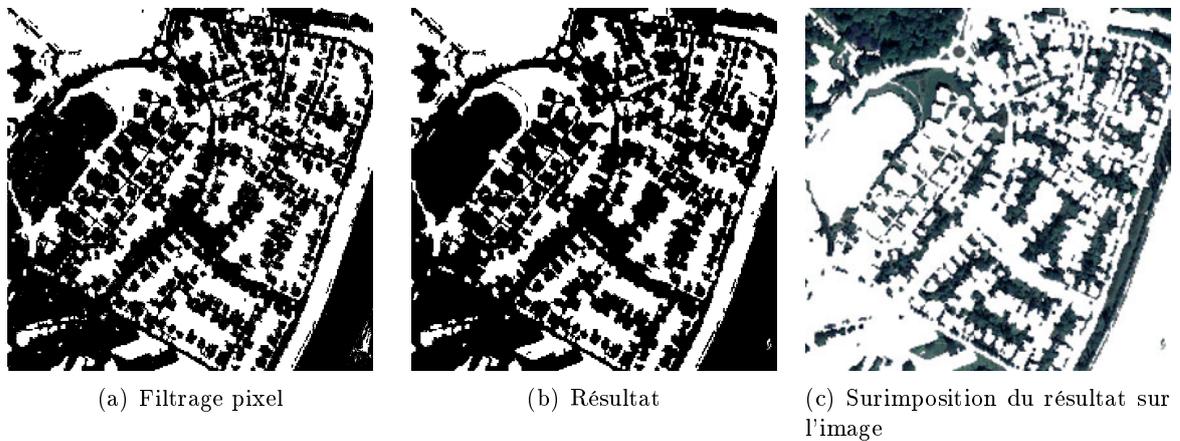
Classe	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Maison	91.20	81.79	86.24
Végétation	96.54	96.68	96.61
Route	82.77	85.19	83.96
Eau	99.95	98.27	99.11

détecteurs (table 5.3), on constate une augmentation de la précision pour les classes *végétation* et *route* et une diminution du rappel pour ces deux classes. La fusion des résultats permet donc de réduire certaines erreurs mais en commet d'autres. Ceci est dépendant des pondérations de chaque classe pour la fusion. Nous pouvons aussi constater que l'utilisation des résultats de certains détecteurs peut améliorer les performances d'autres (par exemple la F-Mesure de la route passe de 83.96% à 87.50% quand on utilise les autres détecteurs). Nous reviendrons sur ce point plus tard.

Évaluation pour la classe *maison*

La classe *maison* est globalement bien définie : étant subsumée par la classe *bâti*, elle hérite des connaissances sur celle-ci (qui se limitent à la contrainte d'une forme rectangulaire). De plus, l'expert fournit un intervalle de surface (entre 100 et 1000 m²) et indique que les bandes spectrales des exemples sont pertinentes et pourront donc être utilisées pour le filtre de classification des pixels.

Le résultat de l'application de ce détecteur est donné en figure 5.6. L'étape de filtrage par classification basée pixels conserve une grande partie des pixels, notamment certains représentant des routes, immeubles ou des pixels bruités. L'utilisation de la connaissance sur la forme rectangulaire du bâti permet néanmoins d'améliorer considérablement le filtrage. Les maisons non détectées sont principalement le fait d'erreurs de détection : la classification basée pixels a détecté trop peu de pixels de la maison sous-jacente, elle est donc éliminée du fait de sa surface trop faible ou de sa forme inadéquate.

FIG. 5.6 – Application du détecteur pour la classe *maison*.FIG. 5.7 – Application du détecteur pour la classe *végétation*.

Évaluation pour la classe *végétation*

La végétation est un concept très simple à extraire des images. Elle peut en effet être détectée en utilisant directement l'indice pixel *NDVI* (Normalized Difference Vegetation Index) dont la formule est :

$$NDVI = \frac{PIR - R}{PIR + R} \in [-1; 1]$$

avec *PIR* étant la valeur du pixel dans la bande spectrale du proche infra-rouge et *R* la valeur du pixel dans la bande spectrale de la couleur rouge. Pour cet indice, l'intervalle reconnu de la végétation est $[0, 3; 0, 8]$. Le détecteur utilisera donc un filtrage sur la valeur *NDVI*. Il n'y aura ensuite qu'un filtrage de superficie qui retirera certains groupements de pixels trop petits pour être considéré comme des zones de végétation.

Le résultat de la figure 5.7 montre que le résultat obtenu est visuellement satisfaisant.

Évaluation pour la classe *route*

La classe *route* est définie par des exemples et par une contrainte de superficie. Comme on peut le constater sur la figure 5.8, le filtrage pixel conserve beaucoup trop de pixels. Le filtrage

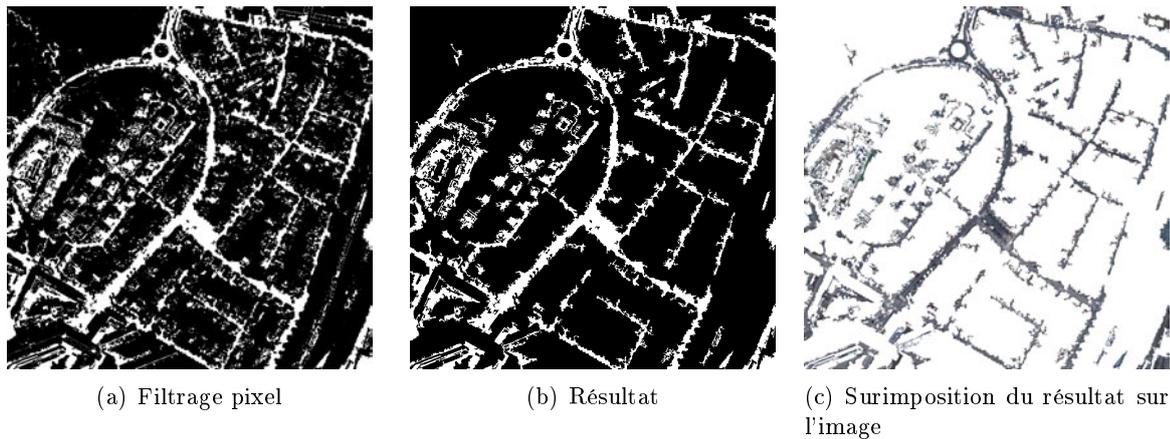


FIG. 5.8 – Application du détecteur pour la classe route.

région sur la superficie permet néanmoins de réduire sensiblement les faux positifs.

Cette classe nous semble pouvoir bénéficier de plus de connaissances afin de supprimer par exemples les zones d'immeubles qui sont reconnues comme de la route. Ces formes sont humainement identifiables comme n'étant pas de la route. La difficulté est donc de supprimer ces régions sans pour autant supprimer ni les régions de petites portions de routes ni les larges portions de réseau routier. De ce fait, aucune contrainte de région ne permet de parvenir à un meilleur résultat.

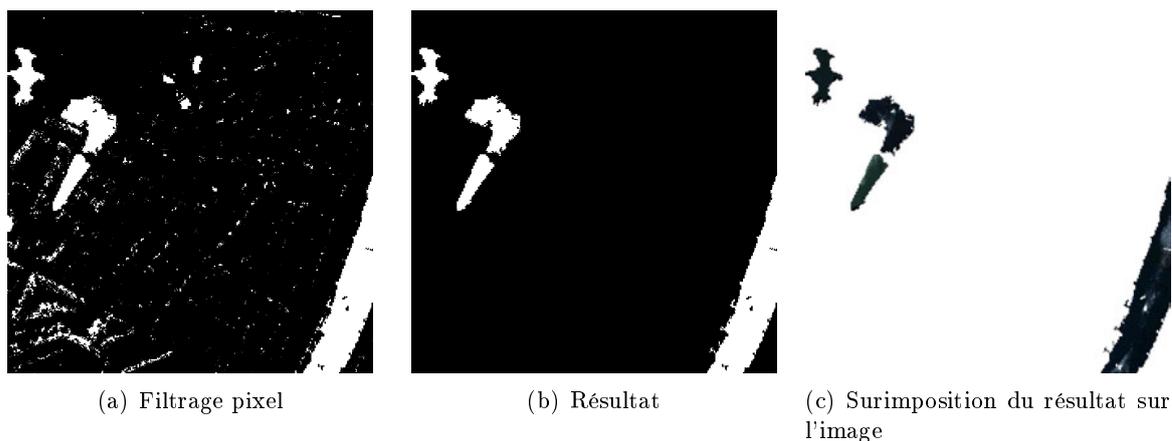
Le problème vient ici du fait que la classe *route* représente deux types d'objets ; les petites portions de routes et les réseaux routiers plus grands. Ces deux types d'objets ont des propriétés très différentes (une petite portion de route étant de nature très compacte contrairement à un réseau par exemple). Une possibilité serait de créer des sous-classes opérationnelles dans la base de connaissances ; c'est-à-dire des classes représentant des types d'objets différents d'une même classe. La détection d'une classe définie par des sous-classes opérationnelles se ferait donc par l'union des détections de ces sous-classes. Ceci peut être une piste d'amélioration de notre proposition.

Évaluation pour la classe *eau*

La difficulté de la classe d'eau est sa proximité spectrale avec l'ombre. Cette difficulté est bien illustrée à la figure 5.9, où l'on constate qu'après le filtrage spectral, une grande quantité de pixels d'ombre est conservée. L'utilisation de la contrainte de superficie suffit ici à supprimer les pixels ne représentant pas la classe *eau*.

5.2.6 Conclusion

Dans cette section, nous avons proposé une méthode automatique de construction de détecteurs à partir d'une base de connaissances comprenant des connaissances du domaine et des exemples. Nos évaluations montrent que cette approche a tout son intérêt comparée à l'approche de classification de régions. Cette approche a l'avantage d'être plus simple à réaliser et à étendre. Les résultats montrent qu'elle privilégie la précision au dépend du rappel.

FIG. 5.9 – Application du détecteur pour la classe *eau*.

5.3 Collaboration de détecteurs et objets composites

À la section précédente, nous avons vu comment extraire des détecteurs à partir de la base de connaissances. Nous allons étudier ici comment un détecteur peut utiliser le résultat d'autres détecteurs comme une source de connaissance.

La première étape pour pouvoir faire collaborer les détecteurs est d'ajouter des relations entre les classes dans la base de connaissances. La table 5.5 définit une liste non exhaustive des relations possibles entre les classes exprimables dans la base de connaissances. Encore une fois, il ne s'agit que de connaissances descriptives des classes. L'opérationnalisation de ces relations, c'est-à-dire leur traduction en algorithmes n'est pas spécifiée dans la base de connaissances construite par l'expert du domaine.

La seconde étape est de prévoir un mécanisme capable d'utiliser ces connaissances. La difficulté consiste en l'ordonnancement des détecteurs. Pour pouvoir fonctionner certains détecteurs ont besoin que d'autres détecteurs aient déjà fourni un résultat. Nous proposons ici d'utiliser un système de producteurs/consommateurs.

Chaque détecteur d_i est un producteur. Il produit une carte de détection pour la classe C_i . Les détecteurs utilisant des relations entre les classes sont aussi des consommateurs. Ils nécessitent en effet un ensemble de cartes de détection afin de pouvoir produire leur propre carte de détection. Il faut qu'un consommateur soit informé quand une carte de détection dont il a besoin est produite. Pour ce faire, lors de la création d'un détecteur consommateur, celui-ci va s'enregistrer chez les détecteurs producteurs dont il a besoin. Lorsqu'un producteur crée une carte de détection, il en notifie tous les consommateurs qui se sont enregistrés chez lui. Ainsi chaque détecteur peut commencer ses traitements dès que les données dont il a besoin sont disponibles.

Nous allons à présent voir les utilisations de ces différentes formes de collaboration à travers des exemples d'applications. Nous proposerons aussi des algorithmes pour opérationnaliser les relations.

5.3.1 Utilisation de détecteurs comme filtres pour d'autres détecteurs

Le premier type de collaboration que nous allons voir est l'utilisation du résultat d'un détecteur comme un filtre pour d'autres détecteurs. Nous allons voir l'utilisation de la relation *n'est pas* pour simplifier la définition de la classe *ombre*, et l'utilisation de la relation *proximité* pour les classes *maison avec végétation* et *péniche*.

TAB. 5.5 – Connaissances contextuelles que peut avoir une classe.

Nom	Attributs	Description
<i>n'est pas</i>	classe C	Si un pixel a été détecté comme appartenant à la classe c , il ne peut être détecté pour la classe du détecteur.
<i>proximité</i>	classe C , distance maximale d_m	Exprime la contrainte qu'un objet de la classe recherchée doit se trouver à proximité (distance inférieure à d_m entre deux pixels de chaque objet) d'un objet d'une classe C .
<i>composition</i>	liste d'éléments, pour chaque élément : classe, cardinalité (minimale et maximale), superficie (minimale et maximale)	La classe à détecter est une composition de l'ensemble des classes définies dans la liste des éléments. Pour être détecté, chaque groupement doit vérifier les contraintes sur les cardinalités (nombre d'objets de chaque classe) et les surfaces relatives de chaque classe.
<i>entourée par</i>	classe C , distance maximale d_m	Exprime la contrainte qu'un objet de la classe recherchée doit être entourée par un ou plusieurs objets de la classe C , i.e. dans chaque direction un objet de classe C est présent à une distance inférieure à d_m

Le cas de la classe *ombre* nous permet d'introduire la relation *n'est pas*. L'ombre est une classe difficile à définir car on ne peut utiliser que des connaissances spectrales : elle n'a pas de forme ou caractéristique de région spécifiques. De ce fait, il n'est pas possible de séparer l'ombre de l'eau qui a une réponse spectrale similaire. Ce n'est pas le cas de la classe *eau* qui est aisément détectable comme nous l'avons vu à la section précédente grâce à des connaissances sur sa superficie.

On peut ainsi utiliser le résultat de la détection de la classe *eau*. Grâce à la connaissance indiquant qu'une zone d'eau ne peut pas être aussi de l'ombre (relation *n'est pas*), il est possible de supprimer les pixels étant déjà détectés comme de l'eau.

Comme nous pouvons le voir à l'image 5.10, le résultat de la détection en utilisant uniquement les connaissances spectrales conserve les pixels d'eau. L'utilisation du résultat de la détection de l'eau permet de supprimer ces pixels superflus.

Nous allons maintenant voir l'utilisation de la relation *proximité* afin de détecter la classe *maison avec végétation*. Cette classe est identique à la classe *maison* sauf qu'elle a une relation de proximité en plus. La relation de proximité prend en attribut, une classe C et une distance maximale d_m . Pour être détectée, un objet doit donc être à une distance inférieure à d_m d'un objet de la classe C . Dans le cas de la classe *maison avec végétation*, la classe C est la classe *végétation*. En effet, une maison a en général un jardin associé. L'ajout de cette connaissance permet de supprimer certaines zones détectées comme des maisons par erreur comme l'illustre la figure 5.11. Des maisons étaient détectées sur des zones occupées par des immeubles. L'utilisation de la connaissance de proximité des maisons avec de la végétation permet de supprimer certaines de ces erreurs.

La relation *proximité* peut aussi s'utiliser pour indiquer qu'une gare doit se trouver à proximité de rails ou qu'un port doit être à proximité d'une étendue d'eau. La figure 5.12 illustre l'utilisation

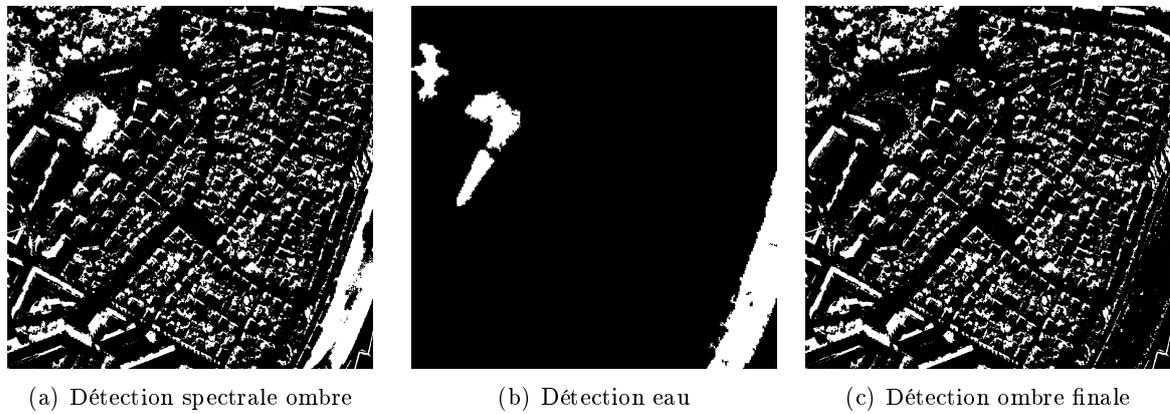


FIG. 5.10 – Application de la collaboration de détecteurs pour l'ombre.

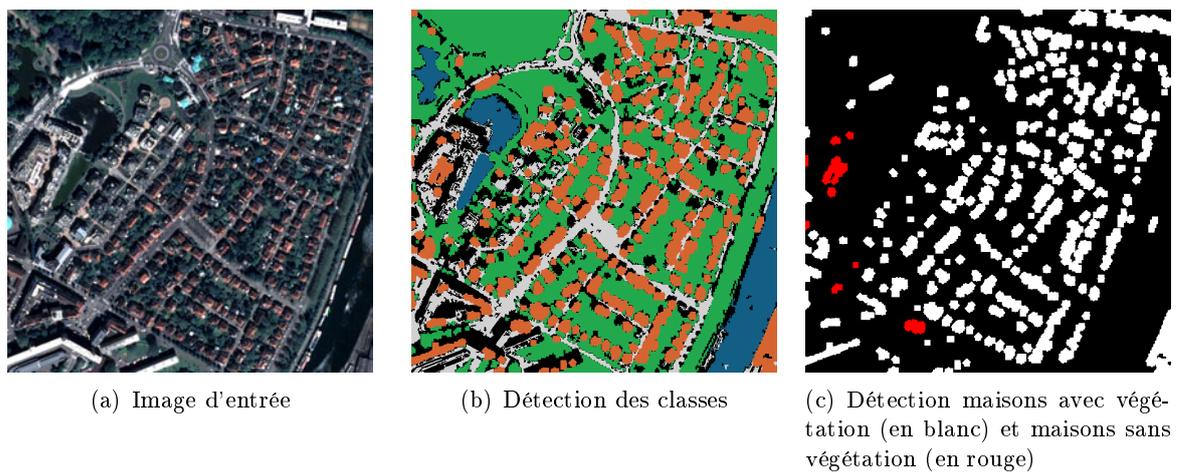


FIG. 5.11 – Application de la collaboration de détecteurs pour les maisons avec végétation.

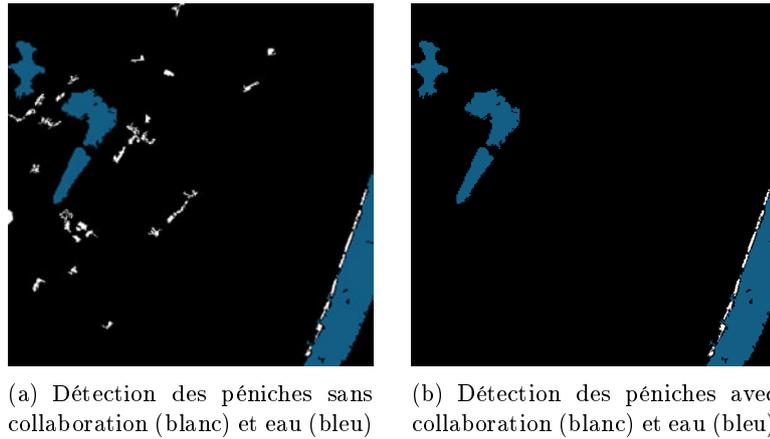


FIG. 5.12 – Application de la collaboration pour la détection de péniches.

de la proximité à l'eau pour détecter la classe péniche. Grâce à l'utilisation de cette connaissance, il est possible de passer de la détection 5.12 (a) (utilisant certains exemples de péniches et des connaissances sur la superficie), conservant plusieurs zones n'étant pas des péniches, à la détection 5.12 (c) supprimant ces erreurs.

5.3.2 Utilisation de détecteurs pour la détection d'objets composites

La collaboration de détecteurs permet aussi de détecter des objets composites, c'est-à-dire des objets composés d'objets de plusieurs classes. Nous allons illustrer ici ce type de détecteur pour rechercher les classes *îlot d'habitations de faible densité* et *îlot d'habitations de moyenne densité*. Pour ce faire, nous utilisons les relations *composition* et *entouré par*.

Un îlot d'habitations est un ensemble de maisons et de zones de végétation entourées par de la route. Nous pouvons donc l'exprimer comme une composition d'objets des classes *maison* et *végétation*. Comme indiqué à la table 5.5, la relation *composition* prend la forme d'une liste de composants correspondant chacun à une classe et des attributs. La définition de la relation de composition pour chacune des classes recherchées est donnée à la table 5.6. La principale différence entre les îlots de faible densité et ceux de densité moyenne est la proportion de la surface occupée par la végétation et les maisons. Dans le cas d'un îlot à densité moyenne, la superficie recouverte de maisons est plus importante. L'autre différence porte sur la cardinalité minimale du nombre de maisons. Pour les îlots à densité moyenne, le nombre minimal de maisons pour former l'îlot est de un. Cette particularité permet de tenir compte du fait que dans les îlots les maisons ont une plus forte propension à être très proches et donc non séparables.

L'algorithme 4 détaille la méthode de détection des compositions. La fonction $\text{nbObjets}(C_i, x_j)$ retourne le nombre d'objets de la classe C_i contenues dans l'objet x_j . La fonction $\text{proportion}(C_i, x_j)$ retourne la proportion de la surface de x_j occupée par des objets de la classe C_i . Enfin les fonctions minCard , maxCard , minProportion et maxProportion renvoient la valeur minimale ou maximale pour les attributs de cardinalité ou de proportion de surface.

Afin de modéliser le fait qu'un îlot d'habitation soit entouré de route, nous utilisons la relation *entouré par avec route* comme classe. Nous utiliserons ici une méthode présentée à l'algorithme 5. La fonction $\text{dilatation}(x, E)$ effectue une dilatation morphologique de la région x en utilisant E comme élément structurant.

L'application de ces algorithmes sur un quartier de Strasbourg (France), différent du précédent pour avoir différentes densités de zones d'habitation, est illustrée à la figure 5.13. La figure

TAB. 5.6 – Propriétés des différents types d’îlots d’habitations.

Type	Composant	Attributs	Intervalle
faible densité	maison	cardinalité superficie	min = 2, max = $+\infty$ min = 10%, max = 50%
	végétation	cardinalité superficie	min = 1, max = $+\infty$ min = 50%, max = 90%
moyenne densité	maison	cardinalité superficie	min = 1, max = $+\infty$ min = 50%, max = 70%
	végétation	cardinalité superficie	min = 1, max = $+\infty$ min = 30%, max = 50%

Algorithme 4 Algorithme de détection des compositions.

Entrée : \mathcal{P} l’ensemble des pixels de l’image à interpréter

Entrée : C_c la classe recherchée

Entrée : \mathcal{C}_c l’ensemble des classes d’objets composant les objets de la classe C_c .

Sortie : \mathcal{X}_c l’ensemble des objets de la classe C_c détectés.

pour tout $p \in \mathcal{P}$ **faire**

$$d_c(p) := \begin{cases} \text{vrai} & \text{si } \exists i \mid d_i(p) = \text{vrai avec } C_i \in \mathcal{C}_c \\ \text{faux} & \text{sinon} \end{cases}$$

fin pour

$\mathcal{X}_c :=$ les composantes connexes de d_c

pour tout $x_j \in \mathcal{X}_c$ **faire**

pour tout $C_i \in \mathcal{C}_c$ **faire**

si nbObjets(C_i, x_j) \notin [minCard(C_i, C_c), maxCard(C_i, C_c)] **alors**

$\mathcal{X}_c := \mathcal{X}_c \setminus \{x_j\}$

sinon si proportion(C_i, x_j) \notin [minProportion(C_i, C_c), maxProportion(C_i, C_c)] **alors**

$\mathcal{X}_c := \mathcal{X}_c \setminus \{x_j\}$

fin si

fin pour

fin pour

Algorithme 5 Algorithme de détection de la relation *entouré par*.

Entrée : x la région à tester

Entrée : d_e la détection de la classe e dont des objets doivent entourer x

Entrée : distMax la distance maximale autorisée entre x et les objets qui l'entourent

Sortie : vrai si x est entouré par des objets de la classe e , faux sinon

$E :=$ disque de rayon distMax

$x' :=$ dilatation(x, E)

$\mathcal{Q} :=$ division de x' en quatre quadrants par rapport au centre de la région x'

pour tout $q_i \in \mathcal{Q}$ **faire**

si $\nexists p \in q_i \mid d_e(p) = \text{vrai}$ **alors**

retourne faux

fin si

fin pour

retourne vrai

5.13 (a) représente l'image initiale et la figure 5.13 (b) le résultat des détecteurs pour les classes *maison*, *végétation* et *route*. Les différentes composantes connexes comportant des objets des classes *maison* et/ou *route* sont présentées à la figure 5.13 (c). On y constate notamment que plusieurs îlots ont été regroupés ensemble. Cela est dû au fait que ces îlots sont reliés ensemble soit par un arbre soit par une maison qui est détectée comme plus grande qu'elle ne l'est en réalité. La figure 5.13 (d) représente les îlots détectés comme des îlots d'habitations (d'après la définition donnée précédemment) selon la densité d'habitation. On constate un îlot d'habitations à faible densité (nous préférons ne pas considérer les détections à cheval sur le bord de l'image), les autres îlots étant à densité moyenne. Ce résultat est cohérent avec le contenu de l'image.

La détection d'objets composites est donc facilitée par la collaboration entre détecteurs. Nous avons vu une utilisation des objets composites pour détecter des îlots d'habitation. Nous allons voir à présent comment utiliser les résultats de la détection des îlots pour remettre en cause la détection des maisons.

5.3.3 Collaboration plus complexe entre détecteurs

Dans la situation précédente, nous avons vu des cas où les détecteurs ne sont pas interrompus, c'est-à-dire qu'une fois qu'ils ont les résultats de détection nécessaires à leur fonctionnement, ils sont capables de s'exécuter complètement. De plus, ils considèrent que les résultats des détecteurs qu'ils utilisent sont définitifs.

Face à la complexité des images de télédétection, il semble vain d'espérer un résultat optimal en un seul traitement sans retour en arrière. Au contraire, comme nous l'avons vu à la section 4.5, il peut être pertinent que la classification basée régions remette en cause la segmentation afin de délivrer de meilleurs résultats. Nous prônons la même méthodologie itérative pour l'approche à base de détecteurs.

Prenons comme exemple la détection de la classe *maison*. Une fois la détection des maisons réalisée, il est possible d'effectuer la détection des îlots d'habitations. Or il est possible de vouloir définir qu'une maison doit appartenir à un îlot d'habitations. On peut ainsi utiliser le résultat de la détection des îlots pour filtrer les maisons qui ne sont pas dans des îlots. Cela pose un problème d'ordonnancement. Tant que les maisons n'ont pas été détectées, on ne peut détecter les îlots. Et tant que les îlots ne sont pas détectés, on ne peut détecter les maisons grâce à

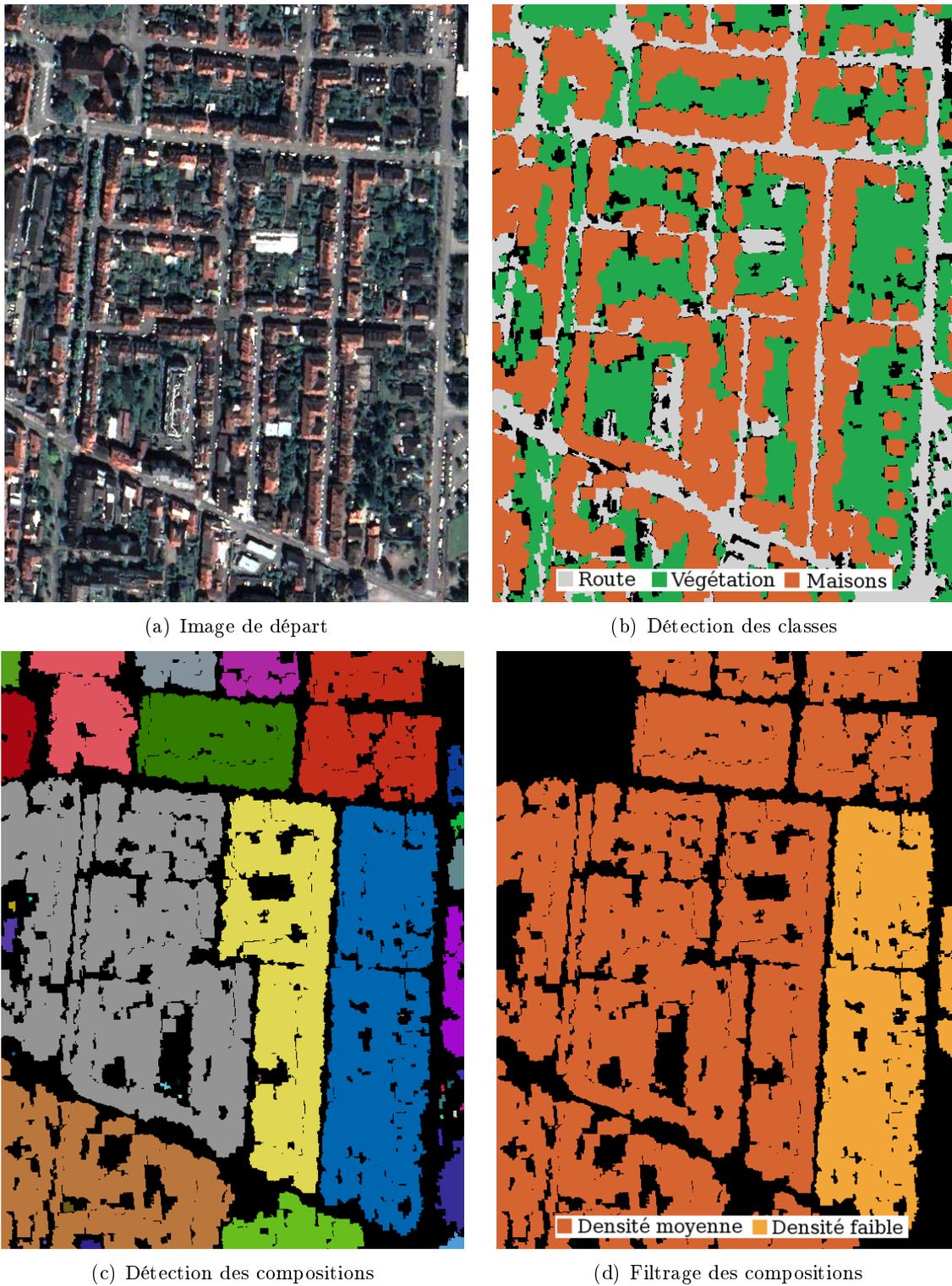
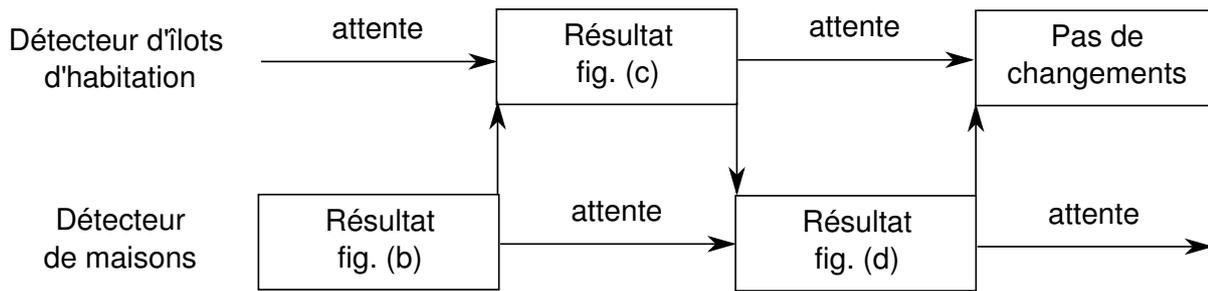


FIG. 5.13 – Utilisation des détecteurs des classes *maison*, *végétation* et *route* pour la détection des classes *îlot d'habitations de faible densité* et *îlot d'habitations de moyenne densité*.



(a) Processus de collaboration ente détecteurs



(b) Première détection de maisons



(c) Détection d'îlots



(d) Seconde détection de maisons

FIG. 5.14 – Collaboration bidirectionnelle pour l'amélioration de la détection des maisons.

leur appartenance aux îlots. Nous résolvons ce problème en permettant aux détecteurs de fournir des résultats intermédiaires. Lorsqu'un détecteur se retrouve bloqué dans l'attente de résultats d'autres détecteurs, il peut fournir son résultat actuel aux détecteurs consommateurs qui se sont enregistrés auprès de lui.

Ce processus est illustré à la figure 5.14 pour la détection des maisons en deux temps. Dans cet exemple, nous considérons qu'une maison doit appartenir à un îlot d'habitations pour être détectée. Dans un premier temps, le détecteur de maisons fonctionne comme il a été défini précédemment dans ce chapitre. Le détecteur d'îlots d'habitations est en attente d'obtenir un résultat du détecteur de maisons. Au bout d'un moment, le détecteur de maisons fournit un résultat intermédiaire de détection. En effet, pour avancer plus loin, il a besoin du résultat de la détection d'îlots qui n'est pas encore disponible. Il notifie donc le détecteur d'îlots qu'une première version de la détection de maisons est disponible. Le détecteur d'îlots peut donc fonctionner. Une fois la détection des îlots effectuée, le détecteur de maisons en est notifié et peut donc supprimer les maisons ne faisant pas parties d'un îlot d'habitations, il génère un nouveau résultat et en notifie le détecteur d'îlot. Celui-ci recommence sa détection car ses données ont changé. Néanmoins, comme la détection reste identique, il n'en informe pas le détecteur de maisons.

Cette méthode permet donc de régler le cas des dépendances circulaires où deux détecteurs ont besoin chacun l'un de l'autre. Elle permet aussi la mise en place des processus de détection itératifs où au fur et à mesure des itérations, plus d'informations sont disponibles fournissant un meilleur résultat. Elle permet aussi de donner à l'utilisateur un résultat dès que possible dans le processus. Ce résultat est provisoire et sera affiné au fil du processus.

5.4 Conclusion

Dans ce chapitre nous avons présenté l'approche d'interprétation d'images par détecteurs et les différences de cette approche vis à vis de l'approche de segmentation puis classification basée régions. Plusieurs avantages à cette approche ont été cités. Premièrement, cette approche permet d'utiliser des détecteurs spécifiques à chaque classe. C'est un problème moins complexe de détecter une classe que de détecter toutes les classes en même temps. Cela permet d'avoir des algorithmes plus simples et plus rapides pour les classes faciles à détecter comme la végétation. Deuxièmement, il est plus simple d'utiliser les connaissances contenues dans la base de connaissances grâce à l'approche par filtre que nous proposons. En effet, il suffit d'associer à chaque type de connaissance un algorithme de filtre pour que les connaissances de ce type puissent être prises en compte. Il n'y a pas besoin de faire des modifications à un algorithme monolithique comme cela peut être le cas dans l'approche segmentation puis classification qui se compose de deux algorithmes monolithiques.

Au niveau des résultats, nous avons vu que l'interprétation par détecteurs permet d'obtenir une précision plus grande grâce à l'utilisation de plus de connaissances d'une part et la non distorsion des régions par la segmentation d'autre part (cette approche n'utilisant pas de segmentation comme premier traitement). Cependant, le rappel est en léger recul.

Nous avons aussi proposé une approche de réutilisation des connaissances produites par les détecteurs avec une collaboration entre les différents détecteurs. Cette collaboration peut s'extraire de la base de connaissances grâce à l'utilisation de relations entre les classes. Nous avons vu que cette collaboration permet de détecter bien plus précisément certaines classes et des objets composites.

Les perspectives d'évolution de cette approche sont nombreuses. L'ajout d'autres types de connaissances sur les classes permettra le développement de nouveaux types de filtres. Pour les rares paramètres (notamment le seuil du filtre par classification floue supervisée), il est possible d'utiliser une méthode évolutionnaire et des exemples pour apprendre les meilleures valeurs de ces paramètres pour chaque classe (en reprenant les travaux présentés à la section 3.2).

Une autre perspective est de rendre plus flexible la structure des détecteurs. Pour l'instant, elle est basée sur un empilement de filtres, un filtre servant de brique de base. Au fur et à mesure du processus, de moins en moins de pixels sont détectés. Il serait intéressant d'ajouter un nouveau type de brique de base, la brique de correction. La brique de correction fonctionnerait de manière inverse à un filtre, c'est-à-dire qu'au lieu d'enlever des pixels, elle en rajouterait. Par exemple, certains îlots d'habitation détectés contiennent des trous de formes carrées, des maisons qui ont été supprimées par un filtre du détecteur de maisons. Si ces zones ne sont détectées par aucun autre détecteur et tenant compte de leur forme et de leur voisinage, il semble raisonnable de pouvoir les reclasser en maisons.

Chapitre 6

Application à l'interprétation d'images satellites

Everyone wants results, but no one is willing to do what it takes to get them.

Harry Callahan – Dirty Harry

Après avoir présenté différents algorithmes pour la construction et la classification d'objets au cours des chapitres précédents, nous allons étudier une application de ceux-ci dans un contexte plus proche de la réalité. En effet, nos algorithmes ont été pour le moment évalués uniquement sur des images de test de petite taille. Nous allons à présent utiliser des images plus grandes et de composition plus hétérogène de Strasbourg et de Marseille.

Dans un premier temps nous présentons en détail l'image de Strasbourg, ainsi que les connaissances du domaine et les exemples mis à notre disposition. Nous évaluons ensuite nos propositions, les approches par segmentation puis classification et par détecteurs, par comparaison avec une approche classique de classification par pixel. Tandis que nos deux propositions se basent sur le paradigme de la construction et de la classification d'objets, la classification par pixel ne cherche pas au contraire à construire d'objets. Nous étudions ensuite la robustesse de ces méthodes en les appliquant sur l'image de Marseille sans nouvel apport de connaissances. Enfin, nous concluons ce chapitre par une discussion d'analyse de cette application.

6.1 Présentation du jeu de données

Le jeu de données que nous allons utiliser dans ce chapitre se décompose en 4 parties : l'image elle-même, les connaissances du domaine, les exemples utilisés pour l'apprentissage et enfin les exemples utilisés destinés à l'évaluation.

6.1.1 Description de l'image

L'image Quickbird ©DigitalGlobe (2008) utilisée dans cette application représente l'agglomération de Strasbourg. Elle est représentative d'une agglomération urbaine constituée de tissus urbains hétérogènes. Cette image, mise à disposition par le laboratoire Image, Ville, Environnement, dispose de 4 bandes spectrales de résolution spatiale 2,7m et d'une bande panchromatique de résolution 0,7m. Chaque bande a une résolution spectrale de 11 bits. Pour nos tests, une fusion entre la bande panchromatique et les bandes multispectrales a été réalisée en utilisant une mé-

thode UWT-M2 étudiée dans Puissant et al. [106]. L'image ayant pour dimensions 22000×19000 pixels soit 418 millions de pixels, la zone d'étude couvre 204km^2 .

L'image est présentée dans son intégralité à la figure 6.1. Le centre de l'image est occupé par Strasbourg, le fleuve à l'Est étant le Rhin. Dans la partie supérieure Nord-Est, des nuages, ainsi que leur ombre portée, sont présents et obstruent certaines zones.

Étant donné l'inopérabilité des traitements sur l'image dans son intégralité (l'image ayant une taille numérique de plusieurs giga octets), celle-ci est subdivisée en 418 carrés de dimensions 1000×1000 pixels. Cette technique peut occasionnellement provoquer des erreurs sur les objets qui seraient à cheval entre deux carrés. Même si ces cas sont rares, nous avons évité d'utiliser des exemples sur les bords des carrés que ce soit pour l'apprentissage ou pour l'évaluation.

Nous allons à présent nous intéresser aux classes recherchées dans cette image et aux connaissances du domaine associées.

6.1.2 Description des connaissances du domaine

Nous décrivons ici les connaissances fournies par l'expert dont nous disposons sur les différentes classes d'objets recherchées. Six classes d'intérêt ont été considérées. Il s'agit des maisons, des immeubles, des routes, de la végétation, de l'eau et des sols nus.

La table 6.1 présente les différentes connaissances du domaine relative à chaque classe. Pour chacune des classes, nous donnons des exemples visuels et les connaissances du domaine fournies par l'expert.

Les experts fournissent une liste de critères (hiérarchisés selon leur importance) classiquement utilisés lors de la photo-interprétation d'une image (couleur/teinte, forme, taille, texture, structure, contexte). Il est difficile de donner des intervalles de valeurs permettant d'identifier de manière univoque un objet car ces attributs varient le plus souvent selon le contexte géographique. Le critère de couleurs/teintes étant très dépendant de la date de prise de vue, de l'illumination ou des capteurs utilisés, les critères les plus souvent utilisés sont ceux relatifs aux dimensions et à la forme des objets.

Pour les classes *maison* et *immeuble*, nous avons à notre disposition des connaissances portant sur la superficie possible pour les objets de cette classe ainsi que sur la longueur minimale d'un côté d'un objet. Une maison ou un immeuble doivent en effet avoir une certaine emprise au sol pour être définis comme tels (*une superficie supérieure à 100m^2 et une largeur supérieure à 10m pour une maison, et 500m^2 et 10m pour un immeuble* cf. dictionnaire de données). Ces intervalles fournis par l'expert sont issus du calcul d'une moyenne effectuée sur au moins 30 exemples pour chaque classe d'objets.

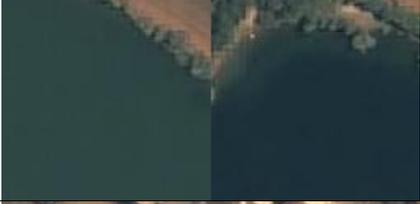
Pour la classe *végétation*, nous avons à notre disposition des connaissances sur la surface minimale d'un objet ainsi qu'une connaissance sur la réponse spectrale attendue d'une surface végétale (l'indice NDVI extrait des réponses spectrales doit être compris entre 0,3 et 0,8).

Pour les classes *eau* et *sols nus*, des connaissances sur la surface minimale et des connaissances spectrales sont fournies. Ces connaissances spectrales sont le résultat de l'analyse de l'image par l'expert et de banques de réponses spectrales. Elles sont donc partiellement dépendantes de l'image. Cependant, ces connaissances sont bien différentes des exemples fournis en apprentissage car l'expert a spécifiquement choisi les bornes des intervalles de réponses spectrales. Celles-ci ne correspondent pas nécessairement aux bornes que l'on peut extraire des exemples. C'est par ce biais que la connaissance de l'expert est introduite.

Enfin, pour les routes, seule la connaissance sur la surface minimale est utilisée.

Nous allons à présent détailler les connaissances fournies sous la forme d'exemples.

TAB. 6.1 – Liste des classes recherchées avec des exemples d’objets ainsi que les connaissances du domaine associées.

Classe	Représentation	Connaissance du domaine
Maisons		superficie, largeur
Végétation		informations spectrales, superficie
Route		superficie
Eau		informations spectrales, superficie
Immeubles		superficie, largeur
Sols nus		informations spectrales, superficie

TAB. 6.2 – Nombre d'exemples de pixels et d'objets par classe.

Classe	Nombre de pixels	Nombre d'objets
Maisons	2578	89
Végétation	376	35
Route	1291	24
Eau	1557	3
Immeubles	2789	19
Sols nus	545	4

6.1.3 Exemples d'apprentissage

Parmi les exemples d'apprentissage, nous pouvons noter deux types d'exemples différents : les exemples pixels et les exemples de zones. Les premiers sont des pixels, c'est-à-dire ne représentant pas un objet complet. Les secondes sont des zones qui ont été en partie étiquetées par les classes recherchées.

Exemples pixels

Les exemples pixels se composent, pour chaque classe, d'un ensemble de pixels choisis sur l'image. La création de ces exemples est très rapide. Dans le cas présent, elle a duré moins de cinq minutes. Pour chaque classe, l'utilisateur utilise un pinceau pour noter des pixels des objets de cette classe qui lui semblent représentatifs. Pour chacun de ces objets, une dizaine à une centaine de pixels sont sélectionnés. Il ne sont pas sélectionnés un par un, mais en passant la souris par dessus les pixels ce qui est très rapide.

Tous les exemples sont pris en dehors des zones qui serviront à l'évaluation. La table 6.2 indique pour chaque classe le nombre de pixels d'exemples.

Exemples de zones

Les exemples de zones représentent des zones qui ont été en partie étiquetées par un expert. À partir de ces exemples, il est possible d'apprendre des connaissances relatives à la forme des objets. Cela permet aussi l'utilisation de critères d'évaluation pour la méthode d'apprentissage des paramètres de segmentation proposée à la section 3.2. Contrairement aux exemples pixels, produire des exemples de zones est très coûteux en temps. Une heure a été nécessaire pour produire les deux zones illustrées à la figure 6.2. La difficulté réside dans la précision nécessaire pour délimiter les bords des objets afin de donner un exemple le plus représentatif possible de l'objet sous-jacent.

Comme pour les exemples pixels, ces exemples sont pris en dehors des zones qui serviront à l'évaluation. La table 6.2 indique pour chaque classe le nombre d'objets exemples de chaque classe. Nous constatons que malgré le temps utilisé pour faire ces exemples, le nombre est très faible pour les classes *eau* et *sols nus*. Ces classes sont représentées par un nombre d'objets relativement plus faible que les autres classes dans l'intégralité de l'image. Afin de nous placer dans un contexte réaliste, nous avons décidé de ne pas accorder plus de temps pour la réalisation des exemples de zones.

6.1.4 Exemples d'évaluation

Nous avons choisis trois zones d'évaluation afin de représenter la diversité et l'hétérogénéité des tissus urbains d'une agglomération urbaine telle que Strasbourg.

Nous distinguons trois types de zones d'évaluation.

Zones pavillonnaires

Ces zones correspondent à des zones d'habitat discontinu de type individuel constituées de maisons entourées de surfaces végétales (jardins, pelouse, arbres) et d'un réseau routier. Elles sont organisées le plus souvent en lotissement.

Ce type de zone présente l'avantage que chaque maison est souvent clairement séparée des autres par une surface végétale. Cela évite de construire des objets qui englobent plusieurs maisons. Un tel objet pourrait avoir des caractéristiques, notamment la surface, éloignées de celles attendues pour un objet de la classe *maison*.

Nous avons défini deux zones d'évaluation à dominante pavillonnaire qui sont présentées, avec leurs exemples associés, à la figure 6.3.

Zones périurbaines

Les zones périurbaines sont constituées de villages ou bourgs au contact direct avec le centre urbain. Elles sont composées principalement de maisons, de routes, et de surfaces végétales assez vastes (jardins, champs, forêts). Ces zones peuvent aussi contenir des immeubles. La différence majeure avec les zones pavillonnaires est la concentration plus importante des maisons. De plus, ces maisons sont de formes plus hétérogènes (présence de corps de ferme par exemple).

Nous avons défini deux zones d'évaluation à dominante périurbaine qui sont présentées, avec leurs exemples associés, à la figure 6.4.

Zones urbaines

Les zones urbaines sont des zones d'habitations très denses des centres urbains. Elles sont composées principalement d'immeubles, d'un réseau routier et de petites surfaces végétales (arbres le long des routes, parcs).

Ces zones sont difficiles à interpréter car les objets qui les composent sont très hétérogènes et présentent de fortes interactions. Le toit des immeubles peut être construit à partir de matériaux très différents, notamment en asphalte, les rendant similaires à de la route goudronnée. Les immeubles peuvent également avoir des terrasses sur le toit. Les immeubles et les arbres projettent leur ombre sur les routes. Les arbres au bord des routes peuvent cacher partiellement ou totalement celle-ci. Enfin de nombreuses voitures circulent ou sont garées sur la route.

Après avoir présenté notre jeu de données, nous allons appliquer trois méthodes d'interprétation sur celui-ci et commenter les résultats obtenus.

6.2 Application des méthodes d'interprétation

Au cours de cette thèse, nous avons proposé deux méthodes d'interprétation d'images basées sur la construction et la classification d'objets. La première méthode est la segmentation suivie d'une classification basée objets, présentée aux chapitres 3 et 4. La seconde est l'approche par détecteurs discutée au chapitre 5. Nous allons évaluer ces deux méthodes, ainsi qu'une méthode plus classique à titre de comparaison, sur le jeu de données défini à la section précédente.

L'évaluation sera faite au niveau pixels car chaque méthode crée des objets différents. Pour chacune des zones d'évaluation, seules les classes présentes dans les exemples sont considérées. Ainsi, dans les zones pavillonnaires et périurbaines, il n'y a pas d'exemple de la classe *immeuble*. Il n'y a aucun exemple de la classe *eau* et *sols nus* car les objets de ces classes sont très grands par rapport à la taille de nos zones d'évaluation, très peu représentés dans ces dernières et toujours bien détectés par les trois méthodes.

Nous décrivons tout d'abord l'approche classique, la classification basée pixels. Nous nous intéressons ensuite à la segmentation suivie d'une classification basée objet. Enfin, nous finissons avec l'approche par détecteurs. Pour chacune de ces méthodes, nous rappelons son fonctionnement, les connaissances qu'elle utilise et procédons enfin à son évaluation.

6.2.1 Classification basée pixels

La première méthode d'interprétation d'image que nous allons utiliser sur notre jeu de données est une classification basée pixels. Plus précisément, nous utilisons l'algorithme des k plus proches voisins, déjà plusieurs fois utilisé durant cette thèse, pour apprendre l'association entre un pixel et sa classe.

Pour ce faire, cette méthode utilise les exemples d'apprentissage donnés sous forme de pixels. Cette méthode ne fait pas usage des exemples sous forme de région, ni d'aucune connaissance du domaine. Après plusieurs expérimentations, nous avons décidé de prendre $k = 5$, valeur obtenant la meilleure F-mesure sur le jeu d'apprentissage.

Les résultats de l'application de cet algorithme sur le jeu de données sont présentés à la table 6.3. Nous y constatons que la classe *végétation* est très bien traitée dans les trois types de zones (la valeur de F-mesure, agrégation entre la précision et le rappel, étant supérieure à 95% dans les trois cas). La classe *route* obtient des résultats de qualité inférieure. En effet, sur les images périurbaines, des pixels de classe *maison* sont classés en tant que *route*, engendrant des erreurs de précision. À l'inverse, sur les images de zones pavillonnaires et urbaines, le problème est que nombre de pixels *route* sont classés soit en *maison* soit en *immeuble*, engendrant des erreurs de rappel. De nombreuses maisons sont étiquetées par la classe *immeuble*, ce qui entraîne de mauvais scores de rappel. Le mauvais score de précision sur les zones urbaines provient du fait qu'il y a peu de maisons sur ces images et que de nombreux pixels de classe *immeuble* sont classés en *maison*. Le même problème est observé pour la classe *immeuble*, où de nombreux pixels de cette classe sont classés comme *route* ou *eau* impliquant là encore, un faible rappel.

Plus généralement, on remarque dans les résultats produits par cette méthode une grande hétérogénéité des classifications de pixels proches. Des pixels voisins de route ou de toit peuvent être classés différemment du fait d'une différence de réponses spectrales très faible. Cet aspect est induit par le niveau pixel de la méthode, celle-ci ne prenant pas en compte le voisinage.

6.2.2 Segmentation suivie d'une classification basée objets

La méthode de segmentation suivie d'une classification basée objets a été l'objet des chapitres 3 (segmentation) et 4 (classification basée objets). Elle représente notre première proposition pour la construction et la classification d'objets à partir d'image.

Cette méthode utilise l'ensemble des connaissances mises à disposition. Les exemples d'apprentissage sous forme de pixels sont utilisés pour l'algorithme de segmentation supervisée par transformation d'espace. Les exemples d'apprentissage sous forme de zones sont quant à eux utilisés pour l'apprentissage des paramètres de la segmentation et pour la classification à l'aide

TAB. 6.3 – Résultats pour la classification basée pixels selon le type de zone et la classe.

Type de zone	Classe	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Pavillonnaire	Maison	71,45	57,71	63,82
	Route	90,38	77,90	83,68
	Végétation	98,93	97,28	98,10
	Moyenne	86,92	77,63	82,01
Périurbain	Maison	95,52	42,31	80,02
	Route	59,54	90,54	58,65
	Végétation	98,24	95,28	96,74
	Moyenne	84,44	76,04	80,02
Urbain	Maison	11,10	37,16	17,10
	Immeuble	88,47	42,15	57,07
	Route	51,58	72,58	60,13
	Végétation	98,06	93,87	95,92
Moyenne	70,89	65,70	68,16	
Moyenne	Moyenne	80,74	73,12	76,07

d'exemples (abrégée CE). Enfin, les connaissances du domaine sont utilisées par la classification à l'aide des connaissances du domaine (abrégée CD).

L'ordre de fusion des deux classifications (CD et CE) est le suivant : *végétation* par CD et CE, *eau* par CD, *sols nus* par CD, et enfin *route*, *maison* et *immeuble* par CE. Nous avons utilisé la classification par connaissances du domaine pour les classes avec peu d'exemples et où la séparation avec les autres classes par les connaissances du domaine est possible. Les classes difficilement séparables et bien pourvues en exemples sont traitées par la classification basée sur les exemples. Les classes *sols nus* et *eau* n'étant pas utilisées dans la fusion pour la classification à l'aide d'exemples, elles ne sont pas apprises dans celle-ci. Ce point est important, car apprendre ou non des classes qui ne seront pas utilisées n'est pas neutre. Une classe qui ne sera finalement pas utilisée peut en effet réduire le score de rappel des autres classes dans le cas d'une mauvaise classification. Le seuil pour la classification à l'aide de connaissances du domaine est fixé, selon les expérimentations du chapitre 4, à 0,95.

Les résultats de l'application de cet algorithme sur le jeu de données sont présentés à la table 6.4. Plusieurs problèmes peuvent être observés. Premièrement, la segmentation construit des objets relatifs à la végétation qui empiètent légèrement sur les maisons adjacentes. Ceci diminue le rappel de la classe *maison* et la précision de la classe *végétation*. Inversement, les objets construits pour certaines maisons sont plus grands que l'objet sous-jacent. De ce fait, certains de ces objets sont classés comme *immeubles*. Ceci apparaît plus fortement pour les zones périurbaines où les maisons sont plus proches et entourées de surfaces minérales (cours de graviers ou garage extérieur par exemple). Ceci réduit le rappel de la classe *maison* et la précision de la classe *immeuble*. Enfin, certains objets de classe *immeuble* sont classés en tant que *route* et inversement. Ceci est dû à la forte similarité, tant au niveau forme et superficie qu'au niveau spectral, de certains objets construits de ces deux classes.

6.2.3 Approche par détecteurs

L'approche par détecteurs a été détaillée au chapitre 5. Contrairement à l'approche par segmentation suivie d'une classification basée objets, qui construit tous les objets dans un premier temps et les classes dans un second temps, il existe ici une séparation entre les traitements de

TAB. 6.4 – Résultats pour la segmentation puis classification basée objets selon le type de zone et la classe.

Type de zone	Classe	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Pavillonnaire	Maison	92,96	50,31	63,90
	Route	93,20	87,88	90,46
	Végétation	89,98	99,14	94,32
	Moyenne	92,04	79,11	85,04
Périurbain	Maison	99,38	55,81	71,48
	Route	58,04	94,29	71,85
	Végétation	86,40	96,15	91,02
	Moyenne	81,27	82,08	81,68
Urbain	Maison	21,19	40,76	27,89
	Immeuble	84,55	66,91	74,65
	Route	62,14	68,43	65,00
	Végétation	90,20	94,17	92,14
	Moyenne	71,86	72,27	72,05
Moyenne	Moyenne	81,72	77,82	79,59

chaque classe.

Nous avons vu que, pour chaque classe d'objets recherchés, un détecteur spécifique est construit à partir de l'ensemble des connaissances disponibles (exemples et connaissances du domaine). Chacun de ces détecteurs est responsable de la construction et classification des objets de la classe à laquelle il est dédié. Chaque détecteur contient un ensemble de filtres de niveau pixel, forme et/ou région. Les filtres pixels et forme se chargent de construire les objets qui sont ensuite classés comme appartenant ou non à la classe recherchée par les filtres de région.

Cette méthode utilise les exemples sous forme de pixels pour les filtres de niveau pixel et les connaissances du domaine pour les filtres de niveau forme et région.

Les résultats de l'application de cet algorithme sur le jeu de données sont présentés à la table 6.5. On y constate de bons résultats sur la classe *végétation*. Ceci est dû à la simplicité de sa détection avec les connaissances du domaine sur l'indice NDVI (voir section 5.2). Pour les zones périurbaines, plusieurs maisons sont classées en *immeuble*. Ceci est dû au fait que les filtres de niveau spectral et de forme peuvent construire des objets plus grands qu'une seule maison, dans le cas où plusieurs maisons sont très proches. Dans ce cas, ces objets ne sont pas détectés comme appartenant à la classe *maison* à cause de leur superficie trop importante, réduisant ainsi le rappel de cette classe. Ils sont même souvent détectés comme appartenant à la classe *immeuble*. Le problème inverse se pose dans les zones urbaines. Du fait de la grande hétérogénéité des toits des immeubles, les objets construits par le détecteur *immeuble* peuvent être bien plus petits que les immeubles sous-jacents et sont donc rejetés sur ce critère, réduisant le rappel de cette classe. Le détecteur *maison* peut, quant à lui, construire des objets sur les immeubles. À cause, à nouveau, de l'hétérogénéité des toits des immeubles, ces objets peuvent être petits et donc ne pas être rejetés à cause de leur taille. Ceci diminue la précision pour la classe *maison*.

6.3 Comparatif des approches

Nous venons d'appliquer les méthodes de construction et de classification d'objets à un cas concret : une image satellite à très haute résolution spatiale de l'agglomération de Strasbourg.

TAB. 6.5 – Résultats de l’approche par détecteurs selon le type de zone et la classe.

Type de zone	Classe	Précision (en %)	Rappel (en %)	F-Mesure (en %)
Pavillonnaire	Maison	96,45	75,72	84,65
	Route	95,96	82,21	88,54
	Végétation	98,62	98,36	98,49
	Moyenne	97,01	85,43	90,82
Périurbain	Maison	99,47	38,15	55,15
	Route	69,42	93,32	79,61
	Végétation	98,11	95,86	96,97
	Moyenne	89,00	75,78	81,86
Urbain	Maison	20,45	53,49	29,59
	Immeuble	97,86	34,11	50,57
	Route	56,21	68,43	65,00
	Végétation	97,98	99,54	98,75
Moyenne	75,95	68,85	72,20	
Moyenne	Moyenne	87,32	76,69	81,63

TAB. 6.6 – Résultats obtenus par différentes méthodes sur l’image de l’agglomération de Strasbourg. Les résultats sont donnés en pourcentages.

Méthode	Précision	Rappel	F-mesure	Kappa
Approche par détecteurs	87,32	76,69	81,63	84,78
Seg. suivie d’une classif.	81,72	77,82	79,59	78,06
Classification basée pixels	80,74	73,12	76,07	76,88

Les résultats de nos méthodes et d’une méthode classique sont rappelés à la table 6.6.

L’approche par détecteurs obtient les meilleurs scores tant en précision qu’en F-mesure, qui tient compte de la précision comme du rappel. La méthode de segmentation suivie d’une classification basée objets obtient quant à elle le meilleur score en rappel. Elle détecte donc un plus grand nombre de pixels mais commet aussi plus d’erreurs que l’approche par détecteurs. Enfin, la méthode de classification basée pixels est la plus mauvaise dans les trois mesures d’évaluation, ce qui confirme l’intérêt des approche objets. En effet, certaines classes sont difficilement discriminables en utilisant uniquement l’information spectrale. De plus, cette information spectrale est très variable au niveau pixel. L’introduction des objets permet d’utiliser de nouveaux attributs (forme, superficie, . . .) et d’avoir des informations spectrales plus stables car prises sur l’ensemble des pixels d’un objet.

La figure 6.7 illustre les résultats des différentes méthodes sur une zone pavillonnaire. On y constate l’aspect très bruité du résultat de la classification basée pixels, surtout sur la classe *maison* qui est difficilement discriminée de la classe *immeuble*.

Sur le résultat de la méthode de segmentation suivie d’une classification basée objets, on remarque que la végétation s’étend un peu hors des zones de végétation. Ceci est dû en partie à la fusion de bandes de l’image. La bande proche infra rouge, très discriminante pour la végétation, conserve une résolution très grossière, elle a donc tendance à empiéter sur des pixels dont l’indication spectrale des autres bandes les classerait en tant que *maison*.

La méthode de segmentation puis classification a aussi tendance à mal classer certaines petites

régions. Ces régions présentent des réponses spectrales atypiques pour les classes recherchées et peuvent donc générer des erreurs. Il semble possible, dans une phase de post-traitement, de supprimer ces petits objets selon un ensemble de règles, possiblement extraites des connaissances du domaine. Malgré la présence de ces petits objets mal classés, ce résultat est d'aspect moins bruité que la classification pixel grâce à l'utilisation d'objets au lieu de pixels.

L'approche par détecteurs est la moins affectée par cet effet de bruit. Ceci est dû à l'application forte des filtres de niveau région qui composent les détecteurs. Si un objet construit ne valide pas une contrainte de taille par exemple, il n'est pas conservé, même s'il ne semble pas appartenir aux autres classes non plus. On constate donc que certaines maisons ne sont pas classées. Ce comportement ne peut arriver pour les deux autres méthodes qui affectent toujours une classe aux objets ou aux pixels qui les composent, quitte à commettre une erreur. Il y a donc structurellement un choix de faire primer le score de précision sur le score de rappel.

La figure 6.6 illustre, quant à elle, les résultats obtenus sur une zone urbaine. Par rapport à la zone pavillonnaire, la zone urbaine présente plus de difficultés pour la construction et la classification d'objets et pour la méthode basée pixels. Les immeubles projettent leur ombre sur la route. Ces régions sont souvent assimilées à la classe *eau* pour la méthode pixel, *immeuble* pour la méthode de segmentation suivie d'une classification, et *maison* pour l'approche par détecteurs. On note aussi que plusieurs toits d'immeuble sont considérés comme des toits de maison. Nous avons vu que pour la classification pixel, l'erreur était due à la faible séparation entre les pixels de classe *maison* et *immeuble*. Pour la méthode de segmentation suivie de classification, qui semble la moins affectée par ce problème, l'erreur intervient sur les régions créées par la segmentation qui ont à la fois des réponses spectrales et une superficie semblables aux objets de la classe *maison*. Pour l'approche par détecteurs, c'est la taille trop petite des objets construits qui les empêche d'appartenir à la classe *immeuble*.

6.4 Robustesse de l'apprentissage

Après avoir étudié les résultats des méthodes d'interprétation sur une image de Strasbourg, nous allons nous intéresser à la robustesse de l'apprentissage ainsi effectué. Pour ce faire, nous allons utiliser les données de Strasbourg pour faire l'apprentissage des méthodes. Les algorithmes sont appliqués sur une image de Marseille. Nous pouvons donc étudier la robustesse des différentes approches. Cette image Quickbird (© Digital Globe 2004), mise à disposition par le CNES dans le cadre du programme d'accompagnement ORFEO, représente également une agglomération urbaine avec des tissus urbains similaires à ceux de Strasbourg et d'une manière générale typiques des grandes agglomérations françaises. Elle représente une région côtière au climat plus aride. Nous allons étudier les résultats obtenus sur une zone pavillonnaire et une zone urbaine.

Les résultats obtenus sur la zone pavillonnaire de Marseille sont illustrés à la figure 6.8. Une évaluation quantitative est donnée à la table 6.7 calculée sur les moyennes des classes présentes effectivement dans la zone (*maison*, *végétation*, *route* et *sols nus*). Pour la classification pixel, nous retrouvons les mêmes problèmes que sur l'image de Strasbourg. La séparabilité entre la classe *immeuble* et *maison* est problématique. Une grande partie des pixels qui devraient être classés en *maison* sont classés en *immeuble*. On obtient ainsi un rappel pour la classe *maison* de 49%. On y constate aussi une grande hétérogénéité de classification entre des pixels proches. Enfin, cas particulier pour cette image, une grande partie des pixels de *végétation* sont classés en tant que *maison*, réduisant le rappel de la classe de végétation à 64% et la précision de la classe *maison* à 15%. Cet effet est dû à la différence entre les signatures spectrales de l'image de Strasbourg (utilisé comme apprentissage) et celle de Marseille, le terrain étant plus aride.

TAB. 6.7 – Résultats obtenus par différentes méthodes sur une zone pavillonnaire de l'agglomération de Marseille. Les résultats sont donnés en pourcentages et représentent une moyenne pour chaque classe.

Méthode	Précision	Rappel	F-mesure	Kappa
Approche par détecteurs	84,26	61,34	71,00	79,01
Seg. suivie d'une classif.	92,10	81,62	86,55	89,47
Classification basée pixels	71,23	57,78	63,81	61,64

TAB. 6.8 – Résultats obtenus par différentes méthodes sur une zone urbaine de l'agglomération de Marseille. Les résultats sont donnés en pourcentages.

Méthode	Précision	Rappel	F-mesure	Kappa
Approche par détecteurs	88,91	74,66	81,16	77,82
Seg. suivie d'une classif.	76,18	63,91	69,51	52,36
Classification basée pixels	84,03	49,22	62,08	68,06

La méthode segmentation puis classification semble avoir une robustesse plus grande sur la reconnaissance de la végétation (98% de précision et 99% de rappel). Visuellement, le résultat semble néanmoins peu convaincant. Cette méthode classe par exemple un bon nombre de régions comme étant de l'*eau*. Sa principale erreur réside en un score de rappel faible pour la classe *maison* (55% de rappel). Certaines maisons sont classées comme des immeubles et la végétation empiète sur les maisons. Certaines zones semblent mal classées, mais correspondent à des zones sans classe clairement définissable. Ainsi, cette méthode obtient les meilleurs résultats de F-Mesure pour toutes les classes.

Enfin, la méthode par détecteurs attribue la classe *immeuble* de manière erronée. Cela réduit le rappel des classes *maison* à 53% et celui de la classe *végétation* à 78%. De plus, malgré le fait qu'il détecte de nombreuses zones de sols nus dans l'extrait de la figure 6.8, il obtient un faible score de rappel dans cette classe (32%, les zones de sols nus étant situés majoritairement hors de l'extrait de la figure 6.8).

Pour cette zone pavillonnaire de Marseille, la méthode par détecteurs obtient de meilleurs résultats quantitatifs que la méthode pixels. Elle se place par contre derrière la méthode segmentation puis classification. Néanmoins, il est important de noter que visuellement, les résultats de la méthode par détecteurs semblent plus exploitables que la méthode de segmentation puis classification.

Après avoir vu une zone pavillonnaire, nous étudions les résultats sur une zone urbaine de Marseille qui sont illustrés à la figure 6.9. De même, une évaluation quantitative est donnée à la table 6.8 se basant sur les classes présentes dans la zone (*immeuble* et *route*). La méthode pixel obtient un résultat peu lisible visuellement. La majorité de l'image est interprétée comme appartenant à la classe *maison*, alors que celle-ci n'est pas présente. On obtient donc un Kappa assez faible (52%), ce qui montre une corrélation faible avec les exemples.

La méthode par segmentation puis classification se trompe en classant plusieurs immeubles en *maison*, en *eau* ou en *route*, le rappel étant de 57%. La présence des régions classées en tant qu'*eau* est due aux zones d'ombre qui ne sont pas référencés dans les exemples. De fait, la précision est inférieure à la méthode pixel.

Enfin, la méthode par détecteur offre le meilleur résultat visuel et les meilleures performances quantitatives. Le seul bémol se situe sur le faible score de rappel pour la classe *immeuble* de 63%. Il est néanmoins contrebalancé par une précision très forte de 99%.

L'application des méthodes sur une image de Marseille après un apprentissage sur les connaissances de Strasbourg a permis d'évaluer la robustesse des méthodes proposées. On y constate que les méthodes pixels et segmentation puis classification sont fortement impactées par le changement d'image. La méthode pixel obtient de mauvais résultats sur nos deux tests, tandis que pour la méthode segmentation puis classification cela se limite à la zone urbaine. D'un autre côté, la méthode par détecteurs montre une grande régularité.

Ces constats sont explicables par le fonctionnement intrinsèque des méthodes. La méthode basée pixel se base uniquement sur des réponses spectrales qui sont variables d'une image à l'autre selon l'éclairage, les différences de type de végétation ou de matériaux de construction. Avec la méthode de segmentation puis classification, des informations de régions sont aussi acquises lors de l'apprentissage et peuvent se baser sur des connaissances du domaine. Néanmoins, cette méthode comporte beaucoup d'apprentissage qui peut avoir du mal à se transposer sur une autre image. Enfin la méthode par détecteurs utilise certes des connaissances extraites par apprentissage d'exemples, mais fait aussi un large usage de connaissances du domaine. De fait, elle est plus facilement transposable entre les images.

En conclusion de ce chapitre d'application, nous pouvons dire que le principe de construction et de classification d'objets, de même que l'utilisation conjointes d'exemples et de connaissances du domaine, a montré son intérêt par rapport à une approche purement pixel n'utilisant que les exemples. Ceci permet l'utilisation d'attributs de régions et ainsi une meilleure séparabilité des classes. Notamment, les connaissances du domaine intégrées dans le processus permettent une meilleure séparation entre les classes *maison* et *immeuble* et d'éviter les erreurs de classification pour les classes où l'apprentissage à partir d'exemples est moins bon que l'utilisation des connaissances du domaine.

Il convient de noter que l'approche que nous présentons à partir de détecteurs nous semble à la fois moins gourmande en connaissances (elle ne nécessite pas d'exemples de régions), plus performante en moyenne et plus robuste.



FIG. 6.1 – Image satellite de l'agglomération de Strasbourg représentant 15,4km (22000 pixels à 0,7m) en hauteur et 13,3km (19000 pixels à 0,7m) en largeur.



(a) Zone (2,13)



(b) Exemples de la zone (2,13)



(c) Zone (11,11)



(d) Exemples de la zone (11,11)

■ Route ■ Végétation ■ Maisons ■ Eau ■ Immeubles ■ Sols nus

(e) Légende

FIG. 6.2 – Exemples des zones utilisées en apprentissage.



(a) Zone (5,7)



(b) Exemples de la zone (5,7)



(c) Zone (5,18)



(d) Exemples de la zone (5,18)

■ Route ■ Végétation ■ Maisons ■ Eau ■ Immeubles ■ Sols nus

(e) Légende

FIG. 6.3 – Exemples d'évaluation pour les zones pavillonnaires.



(a) Zone (3,11)



(b) Exemples de la zone (3,11)



(c) Zone (5,2)



(d) Exemples de la zone (5,2)

■ Route ■ Végétation ■ Maisons ■ Eau ■ Immeubles ■ Sols nus

(e) Légende

FIG. 6.4 – Exemples d'évaluation pour les zones périurbaines.



(a) Zone (11,9)



(b) Exemples de la zone (11,9)



(c) Zone (13,9)



(d) Exemples de la zone (13,9)

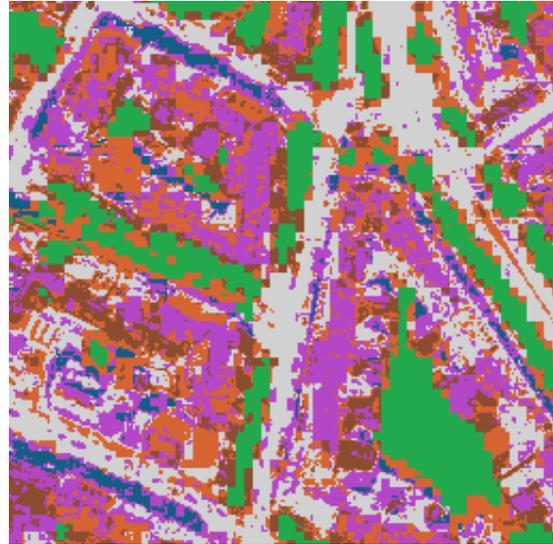
■ Route ■ Végétation ■ Maisons ■ Eau ■ Immeubles ■ Sols nus

(e) Légende

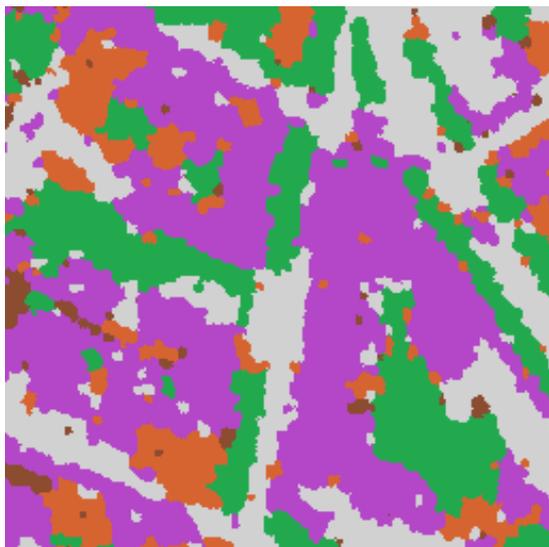
FIG. 6.5 – Exemples d'évaluation pour les zones urbaines.



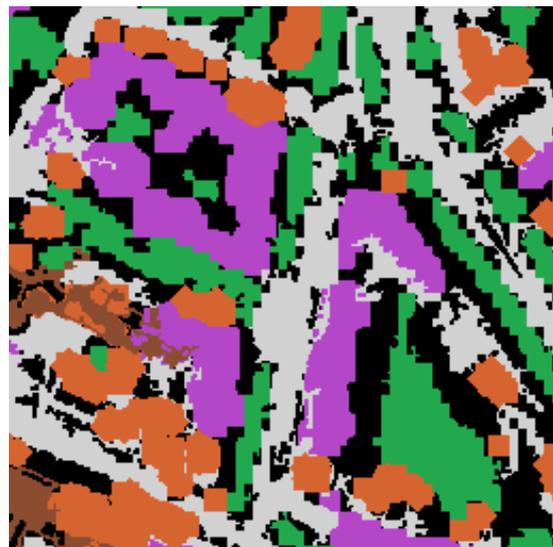
(a) Image source



(b) Classification basée pixels



(c) Segmentation suivie d'une classification basée objets



(d) Approche par détecteurs

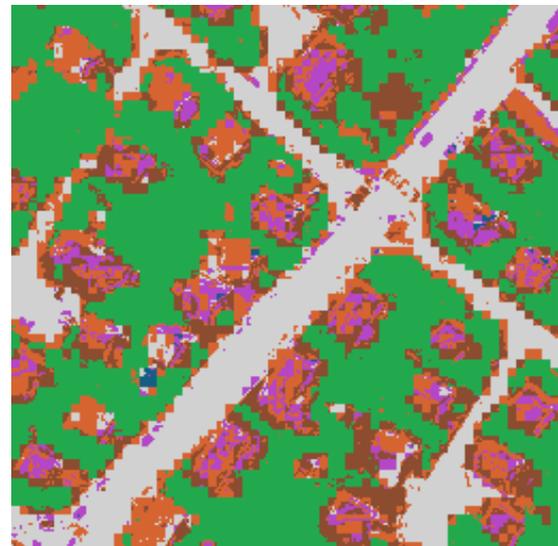
Route
 Végétation
 Maisons
 Eau
 Immeubles
 Sols nus

(e) Légende

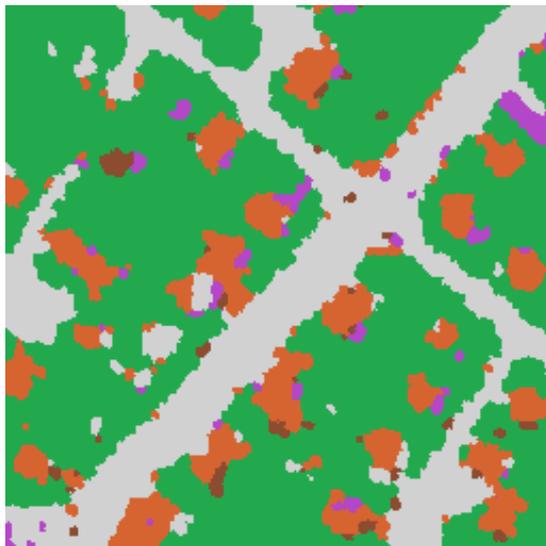
FIG. 6.6 – Illustration des résultats sur une zone urbaine.



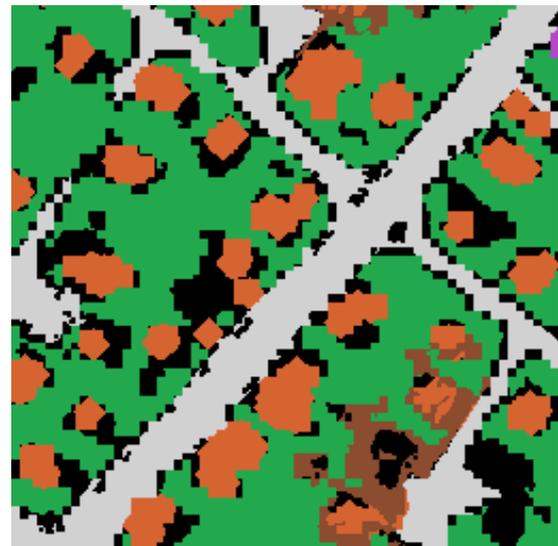
(a) Image source



(b) Classification basée pixels



(c) Segmentation suivie d'une classification basée objets

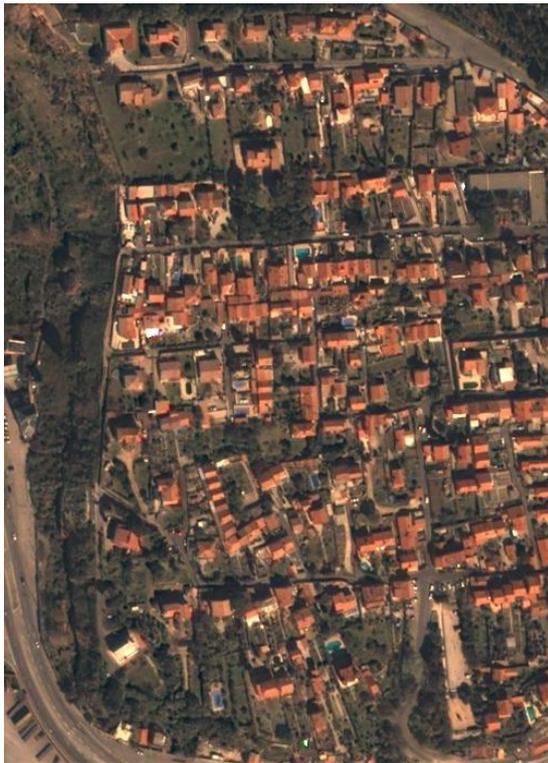


(d) Approche par détecteurs

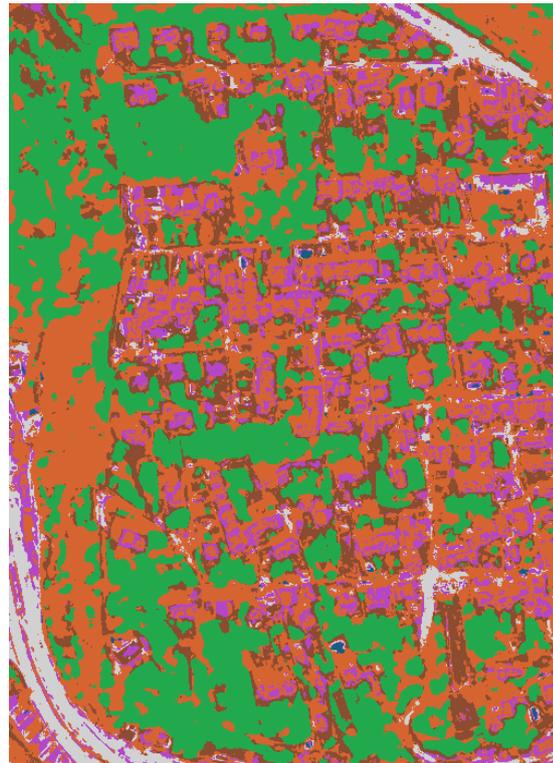
■ Route ■ Végétation ■ Maisons ■ Eau ■ Immeubles ■ Sols nus

(e) Légende

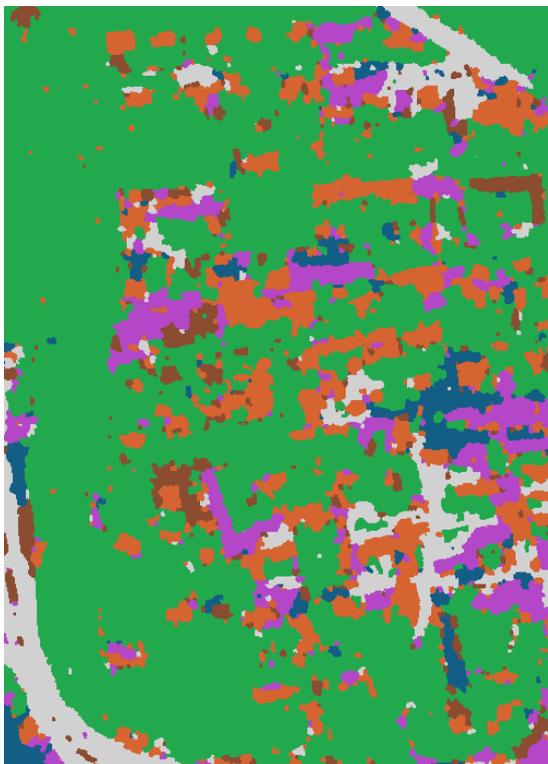
FIG. 6.7 – Illustration des résultats sur une zone pavillonnaire.



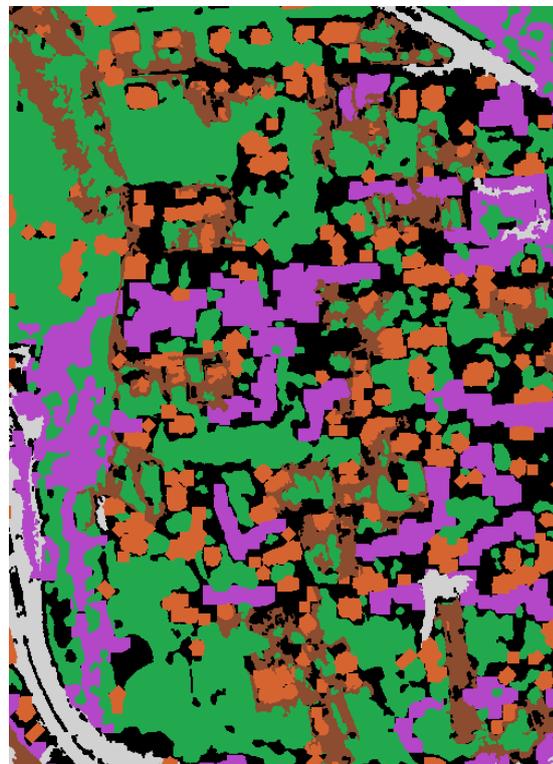
(a) Image source



(b) Classification basée pixels



(c) Segmentation suivie d'une classification basée objets



(d) Approche par détecteurs

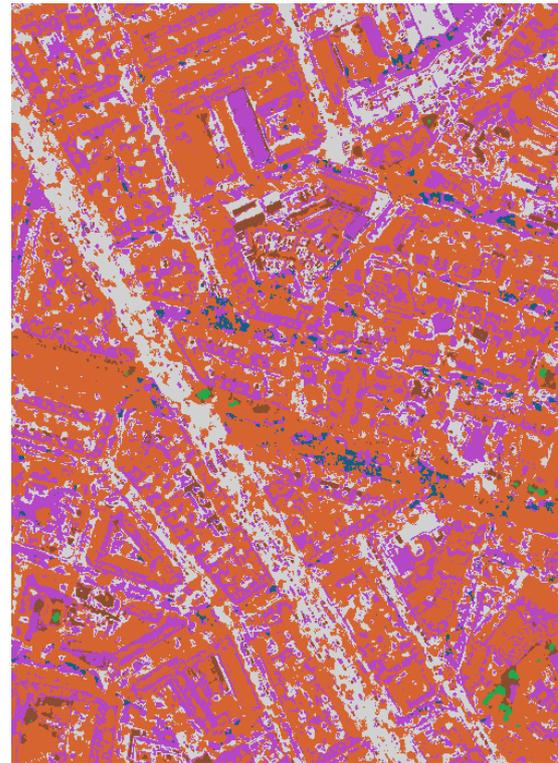
■ Route ■ Végétation ■ Maisons ■ Eau ■ Immeubles ■ Sols nus

(e) Légende

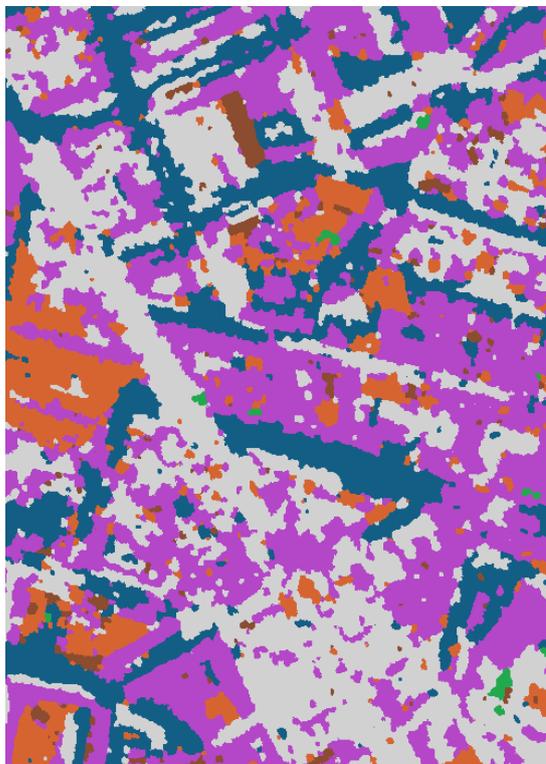
FIG. 6.8 – Illustration des résultats sur une zone pavillonnaire de Marseille.



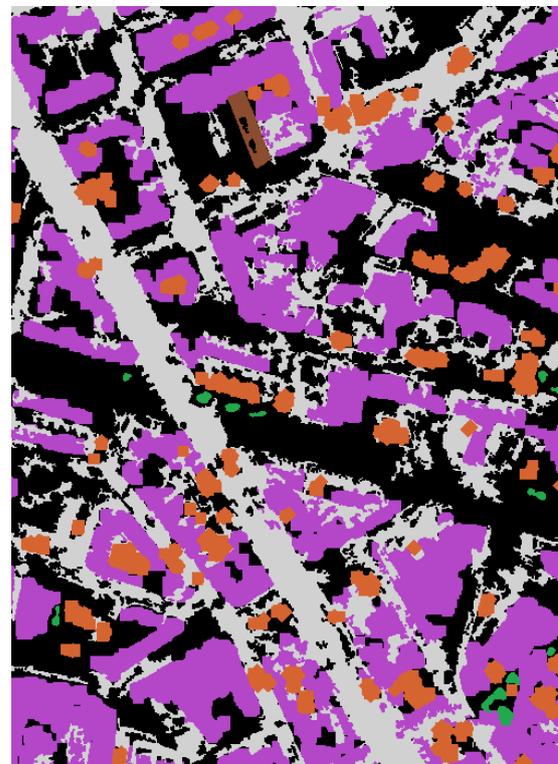
(a) Image source



(b) Classification basée pixels



(c) Segmentation suivie d'une classification basée objets



(d) Approche par détecteurs

Route
 Végétation
 Maisons
 Eau
 Immeubles
 Sols nus

(e) Légende

FIG. 6.9 – Illustration des résultats sur une zone urbaine de Marseille.

Chapitre 7

Conclusion générale

Évidemment même abrégeant au possible, je vous ai demandé beaucoup... lecteur patient, certes, presque attentif, ami ou ennemi, vous approchez de la millième page, vous n'en pouvez plus... butant, ma faute!... de-ci, de-là, d'un mot... un autre... au trop long cours de ce pensum...

Céline – Nord

Dans cette thèse, nous avons proposé des approches pour résoudre le problème de construction et de classification d'objets à partir d'images. Nous avons fait un usage important des connaissances, qu'il s'agisse d'exemples ou de connaissances du domaine. Nous avons mis en place des processus itératifs permettant une remise en cause des résultats pour les améliorer (section 4.5 et 5.3).

Nous recensons dans ce chapitre l'ensemble des contributions que nous avons apportées, dressons une liste de perspectives à court et long terme de nos travaux et, enfin, élaborons une réflexion sur les limites de l'intégration de connaissances dans le processus de construction et de classification d'objets à partir d'images.

7.1 Contributions

Nous classons les contributions apportées selon leur étape dans le processus d'interprétation d'image et selon les domaines dans lesquelles elles s'inscrivent. Ce classement est présenté à la table 7.1. Nous constatons que pour chacune des étapes d'interprétation d'image abordées (segmentation, classification et détection), nous avons fait usage de la plupart des domaines que nous avons décidé d'aborder. Dans un souci de cohérence de nos travaux, chacune de ces contributions fait appel à la base de connaissances.

Nous allons à présent revenir plus en détail sur les apports pour chacun des domaines.

Fouille de données

Le domaine de la fouille de données pour l'interprétation d'images a été exploré de quatre manières. Lors de l'étape de segmentation nous avons exploité la fouille de données, dans un premier temps, afin d'apprendre/d'extraire des paramètres pour des algorithmes de segmentation. Nous avons ensuite étudié l'intérêt que pouvait avoir une classification supervisée floue, non pas

TAB. 7.1 – Apport de cette thèse (références par les numéros de section de cette thèse) selon l'étape dans le processus d'interprétation d'image et les domaines de recherche associés.

	Segmentation	Classification	Détection
Fouille de données	3.2, 3.3	4.2	5.2
Opérationnalisation des connaissances	<i>cf perspectives</i>	4.3	5.2, 5.3
Optimisation par algorithmes évolutionnaires	3.2, 3.4	4.5	<i>cf perspectives</i>
Morphologie mathématique	3.3	<i>cf perspectives</i>	5.2
Combinaison de méthodes	3.4	4.4	5.3

pour produire un résultat définitif, mais pour définir un nouvel espace de données plus sémantique sur lequel une segmentation peut être appliquée.

Dans le cadre de la classification, nous avons approfondi le problème de la création des exemples et nous avons montré que considérer uniquement les exemples fournis par l'expert n'est pas pertinent.

Enfin, pour la détection, nous avons proposé un processus de fusion d'un ensemble de classifications floues afin de détecter les zones de l'image où la présence d'objets d'une classe est possible.

Opérationnalisation des connaissances

Le domaine de la gestion de connaissances est abordée dans trois points. Pour la classification de régions, nous avons proposé une opérationnalisation de la base de connaissances. Malgré des résultats inférieurs à la classification par exemples, cette alternative reste pertinente en cas d'absence d'exemples ou bien pour les classes difficiles à caractériser par des exemples.

Les deux autres apports s'appliquent à la détection. D'une part, nous avons présenté des méthodes pour exprimer des connaissances sous forme de filtres pour la détection. D'autre part, nous avons proposé d'extraire des collaborations entre détecteurs à partir des connaissances contextuelles contenues dans la base de connaissances.

Optimisation par algorithmes évolutionnaires

Des algorithmes évolutionnaires ont été utilisés à deux reprises. La première a eu pour objet l'estimation de paramètres d'algorithmes de segmentation. Nous nous sommes basés sur des mesures d'évaluation de qualité de segmentation à partir d'exemples.

La seconde a utilisé un algorithme évolutionnaire afin de permettre à l'étape de classification de remettre en cause le résultat de la segmentation.

Morphologie mathématique

Le domaine de la morphologie mathématique a été abordé par l'application d'une ligne de partage des eaux sur le relief construit à l'aide d'une classification floue permettant de réduire le phénomène de sur-segmentation et d'introduire des connaissances dans le processus de segmentation.

Nous avons aussi utilisé des opérateurs morphologiques afin de créer des filtres pour des détecteurs.

Combinaison de méthodes

Pour chaque étape de l'interprétation d'image, nous avons proposé une approche de combinaison de méthodes. Nous avons proposé une approche enveloppe où les paramètres d'un algorithme de segmentation basée sur des exemples sont optimisés par une méthode évolutionnaire utilisant elle aussi des exemples. Nous avons utilisé en parallèle deux méthodes de classification basées régions et fusionné leurs résultats pour donner un résultat définitif. Enfin, nous proposons une approche collaborative de détecteurs. Ceux-ci communiquent aux autres leurs résultats intermédiaires et, utilisent les résultats des autres détecteurs pour s'améliorer.

La combinaison des divers apports de cette thèse a été l'objet du chapitre 6 qui a évalué l'impact de ces améliorations. Nous avons évalué nos propositions dans le cadre de la classification d'images de télédétection. Il convient de noter que notre approche est générique et peut s'appliquer à tout domaine où des images contiennent des objets de classes d'intérêt et où de la connaissance, sous forme d'exemples et/ou de connaissances du domaine, est disponible. Notre originalité tient en l'introduction de connaissances dans un processus de construction et de classification d'objets. Les résultats obtenus dans notre application sur des images de télédétection sont concluants. Les deux approches de construction et de classification d'objets à l'aide de connaissances obtiennent des résultats significativement meilleurs qu'une approche non objets et utilisant moins de connaissances. Nous avons aussi montré que parmi les deux approches que nous proposons, l'approche par détecteurs semble plus pertinente.

Cette conclusion de nos travaux est importante car la majorité des travaux actuels portent sur l'utilisation d'une segmentation suivie d'une classification [7, 23, 82]. Nous avons montré qu'une approche intégrant la construction et la classification pour chaque classe de manière séparée est à la fois moins gourmande en connaissances, plus performante en moyenne et plus robuste.

7.2 Perspectives

Après avoir présenté les contributions de cette thèse, nous allons à présent discuter de certaines perspectives de nos travaux. Nous aborderons d'abord des perspectives à court terme, simples extensions des travaux présentés, puis des perspectives à plus long terme, nécessitant un travail de recherche plus important.

7.2.1 Perspectives à court terme

Parmi les perspectives à court terme, nous proposons d'étendre les travaux sur la segmentation. Nous avons vu l'utilisation d'algorithmes évolutionnaires pour diriger une segmentation en paramétrant un algorithme de segmentation. Les résultats montrent que cette méthode reste limitée par les possibilités de l'algorithme de segmentation sous-jacent utilisé. Deux méthodes pour s'affranchir de cette contrainte peuvent être développées.

La première consiste à ne plus optimiser les paramètres, mais l'algorithme de segmentation lui-même. Les algorithmes évolutionnaires peuvent en effet être utilisés pour construire des programmes, notamment des algorithmes de segmentation [116]. Le processus serait le même que celui détaillé dans cette thèse. L'algorithme évolutionnaire ferait évoluer des solutions possibles (dans ce cas des algorithmes de segmentation) selon leur évaluation sur des exemples. La difficulté consiste à définir quels blocs de base peuvent être utilisés pour créer l'algorithme de segmentation.

Une seconde méthode permet d'être sûr de pouvoir segmenter correctement l'image (mais pas nécessairement d'y arriver). Elle consiste à ne plus utiliser l'algorithme évolutionnaire pour définir les paramètres d'un algorithme de segmentation, ni même de créer un algorithme de segmentation, mais de créer directement la segmentation d'une image donnée [64, 37, 9, 80, 59, 4]. La difficulté de cette approche est son inadéquation avec des mesures d'évaluation supervisées par des exemples (section 3.2). Nous pouvons néanmoins utiliser la classification à l'aide de connaissances (section 4.3) pour l'évaluation, le même principe ayant été utilisé à la section 4.5 pour l'optimisation de paramètres.

Pour la classification, nous pensons qu'il serait possible d'obtenir de nouveaux attributs de régions par l'utilisation de la morphologie mathématique. L'utilisation des connaissances du domaine sur les formes possibles des différentes classes permettrait de définir, à l'aide de la morphologie mathématique, des attributs pouvant indiquer l'adéquation de la forme des régions à classer avec les formes connues pour les différentes classes.

L'introduction d'une classification basée région itérative afin de pouvoir prendre en compte le contexte est aussi une perspective intéressante. Nous avons vu à la section 4.1 que les attributs contextuels sont difficiles à intégrer : on ne peut prendre en compte la classe des régions avoisinantes qu'après l'étape de classification. Une voie possible serait de faire une classification itérative. Une première classification serait obtenue sans utiliser les attributs contextuels. Une seconde classification pourrait ensuite être obtenue en utilisant des attributs contextuels extraits de la première classification.

Enfin, pour l'interprétation d'image par ensemble de détecteurs, une perspective à court terme serait d'intégrer de nouveaux filtres et de nouveaux types de relations entre classes. Nous avons vu à la section 5.2 que l'approche que nous proposons est prévue pour être facilement extensible. Une autre perspective réside dans l'usage d'une optimisation de paramètres par algorithmes évolutionnaires. Certains filtres, par exemple le filtre spectral par classification supervisée floue, peuvent avoir recours à un paramétrage. Nous avons fixé manuellement ces paramètres. Un apprentissage de ces paramètres pourrait néanmoins améliorer les résultats.

7.2.2 Perspectives à long terme

D'après nous, trois perspectives à long terme sont envisageables et pertinentes : l'intégration de plus de connaissances dans la segmentation, la collaboration plus forte entre la segmentation et la classification et, pour finir, la poursuite des travaux en détection.

Comme nous pouvons le voir à la table 7.1, nous n'avons pas opérationnalisé les connaissances du domaine pour la segmentation. Cette intégration est difficile car lors de l'étape de segmentation nous ne connaissons pas encore la classe des régions qui sont créées. Ceci semble pourtant nécessaire pour utiliser les connaissances sur des classes particulières. Par exemple, nous disposons de connaissances sur la forme de certaines classes. Or, la forme peut difficilement être utilisée dans lors d'une segmentation car les régions ne sont pas encore créées. Une piste possible serait l'utilisation des contours actifs [66].

Nous avons conclu le chapitre sur la classification en mettant en avant l'importance de la

segmentation et nos résultats ont montré la pertinence d'une remise en cause de la segmentation afin d'obtenir une meilleure classification. La remise en cause est néanmoins peu précise, seuls les paramètres pour la segmentation sont contrôlés. Cette approche ne permet pas de conserver les régions qui étaient facilement identifiées et donc bien segmentées. Ce mécanisme semble donc pouvoir être amélioré. Une piste serait de conserver les régions identifiables et de ne segmenter à nouveau que les zones restantes. Cette approche permettrait donc d'utiliser les meilleurs paramètres de segmentation selon les objets à segmenter, au lieu d'utiliser un unique jeu de paramètres. Il serait ainsi possible d'utiliser différents algorithmes de segmentation.

La seconde perspective concerne l'approche par ensemble de détecteurs. Nous avons conclu le chapitre 5 par la perspective consistant à rendre plus flexible la structure des détecteurs. Nous avons proposé l'ajout d'une nouvelle brique de base dite corrective car elle permettrait de rajouter des pixels comme étant détectés alors que la brique de base actuelle fonctionne comme un filtre, c'est-à-dire que chaque brique supprime des pixels détectés. L'autre perspective est la subdivision des détecteurs. Nous avons vu par exemple que la classe *route* regroupe des ensembles d'objets très différents. Nous pensons que l'introduction de classes opérationnelles, classes n'étant pas recherchées par l'utilisateur, peut se révéler intéressante. Une classe pourrait avoir peu de connaissances attachées directement (celles qui s'appliquent à l'ensemble de ces objets) mais disposerait d'un ensemble de classes opérationnelles. Les objets d'une telle classe seraient donc l'ensemble des objets détectés par ces sous-classes opérationnelles. Il deviendrait aussi possible de rechercher des classes sans intérêt direct pour l'utilisateur, ou n'ayant pas d'objets réels associés dans l'image comme l'ombre ou des zones d'illuminations. Ces classes opérationnelles seraient utilisées pour détecter des classes d'intérêt pour l'utilisateur (l'ombre permettant par exemple de détecter des immeubles).

7.3 Réflexion sur les limites d'intégration de connaissances

Notre principal objectif dans cette thèse a été l'intégration de connaissances dans le processus de construction et de classification d'objets à partir d'images. Pour l'atteindre, nous avons proposé plusieurs contributions qui viennent d'être énumérées. Au regard des résultats obtenus, ces contributions semblent pertinentes. Pour autant, malgré l'intégration de connaissances, la classification des images n'est pas encore parfaite.

La question qu'il convient donc de se poser est de savoir d'où vient cette imperfection. Une première cause possible est une intégration de connaissances insuffisante, c'est-à-dire que les connaissances sont bien exprimées dans la base de connaissances mais mal exploitées. Même avec les propositions de cette thèse, nous ne doutons pas que l'on puisse encore explorer de nombreuses pistes.

Une seconde cause peut être le manque de connaissances ou des imprécisions dans l'expression des connaissances. Nous avons déjà déploré la difficulté de transmission de certaines connaissances implicites qu'à l'expert. De même, il est utopique d'espérer pouvoir extraire une connaissance parfaite des exemples.

Ces deux points peuvent être améliorés. Pour autant sera-t-il possible d'atteindre un résultat parfait? Nous en doutons. Dans nos travaux nous nous sommes limités aux connaissances du domaine, c'est-à-dire relatives aux objets recherchés. Les expérimentations nous font penser que cela n'est pas suffisant. Il faudrait inclure des connaissances relatives à la physique (pour prendre en compte l'illumination de certaines surfaces par le soleil, gérer l'ombre), sur des objets plus petits qui perturbent l'image (les voitures par exemple qui obstruent la route). Ainsi, même pour interpréter une image dans un domaine donné, il semble indispensable d'exploiter de la connais-

sance d'autres domaines. Il faut donc d'une part que des experts de ces domaines expriment leurs connaissances et, d'autre part, concevoir des méthodes pouvant les exploiter.

De plus, dans le cas des images de télédétection (mais cela se retrouve dans une large gamme d'images), l'imprévu est monnaie courante. Le concept même de « connaissances du domaine » suppose que le domaine en question soit clairement définissable. Si l'on prend l'exemple du bâti, c'est difficilement le cas. L'utilisation de matériaux de plus en plus variés pour les constructions, avec des styles de plus en plus atypiques rend impossible une définition pour l'ensemble des bâtiments existants et à venir.

Pour essayer d'égaliser l'expert humain en classification d'images, il faut pouvoir prendre en compte les connaissances du domaine mais aussi la capacité à pouvoir remettre en cause ces connaissances. Cela nécessite de disposer de méta-connaissances qui sont encore plus intuitives, et donc plus dures à formaliser, que les connaissances du domaine.

Nous en arrivons donc à la conclusion suivante : pour réussir à interpréter une image dans un domaine particulier, il est nécessaire de formaliser et d'exploiter des connaissances bien plus vastes à ce simple domaine.

Bibliographie

- [1] D. W. AHA, D. F. KIBLER et M. K. ALBERT : Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
- [2] E. APTOULA et S. LEFÈVRE : A comparative study on multivariate mathematical morphology. *Pattern Recognition*, 40(11):2914–2929, 2007.
- [3] N. S. ARINI : Post-segmentation feature-based classification of synthetic aperture radar data. *Dans IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages 1615–1617. IEEE Computer Society, 2002.
- [4] M. AWAD, K. CHEHDI et A. NASRI : Multicomponent image segmentation using a genetic algorithm and artificial neural network. *IEEE Geoscience and Remote Sensing Letters*, (99):571–575, 2007.
- [5] T. BAYES : An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53, 1763.
- [6] J.A. BENEDIKTSSON, J.A. PALMASON, J.R. SVEINSSON et J. CHANUSSOT : Decision level fusion in classification of hyperspectral data from urban areas. *Dans Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages 73–76. IEEE Computer Society, 2004.
- [7] U. BENZ, P. HOFMANN, G. WILLHAUCK, I. LINGENFELDER et M. HEYNEN : Multi-resolution, object-oriented fuzzy analysis of remote sensing data for gis-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3–4):239–258, 2004.
- [8] A. BEYGELZIMER, S. KAKADE et Langford J. : Cover trees for nearest neighbor. *Dans International Conference on Machine Learning*, 2006.
- [9] S. M. BHANDARKAR et H. ZHANG : Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 3(1):1–21, 1999.
- [10] B. BHANU, S. LEE et S. DAS : Adaptive image segmentation using genetic and hybrid search methods. *IEEE Transactions on Aerospace and Electronic Systems*, 31(4):1268–1291, 1995.
- [11] T. BITTNER et S. WINTER : On ontology in image analysis. *Dans International Workshop On Integrated Spatial Databases ISD'99*, volume 1737, pages 168–191. Lecture Notes in Computer Science, 1999.
- [12] A. BLANSCHÉ, P. GAŃCARSKI et J. J. KORCZAK : Genetic algorithms for feature weighting : Evolution vs. coevolution and darwin vs. lamarck. *Dans Proceedings of the 4th Mexican International Conference on Artificial Intelligence*, volume 3789 de *Lecture Notes in Computer Science*, pages 682–691, 2005.
- [13] J. BÜCKNER, M. PAHL et O. STAHLHUT : Semantic interpretation of remote sensing data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):62–66, 2002.

- [14] A. S. CARDOSO et L. CORTE-REAL : Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14(11):1773–1782, 2005.
- [15] A. P. CARLEER, O. DEBEIR et E. WOLFF : Assessment of very high spatial resolution satellite image segmentations. *Photogrammetric Engineering and Remote Sensing*, 71(11): 1285–1294, 2005.
- [16] A.P. CARLEER et E. WOLFF : Region-based classification potential for land-cover classification with very high spatial resolution satellite data. *Dans Proceedings of the 1st International Conference on Object-Based Image Analysis (OBIA 2006)*, 2006.
- [17] G. CASTILLA : Size-constrained region merging : A new tool to derive basic landcover unit from remote sensing imagery. *Dans Proceedings of ESA-EUSC 2004*, 2004.
- [18] Y. CHEN, F. HUANG, H.D. TAGARE, M. RAO, D. WILSON et E.A. GEISER : Using prior shape and intensity profile in medical image segmentation. *Dans Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1117–1124. IEEE Computer Society, 2003.
- [19] G. CHETTY et N. DESHPANDE : Knowledge-based object recognition system. *Dans Proceedings of the 8th international conference on industrial and engineering applications of artificial intelligence and expert systems*, pages 459–468. Gordon and Breach Science Publishers, Inc., 1995.
- [20] C. CLEVE, M. KELLY, F. R. KEARNS et M. MORITZ : Classification of the wildland-urban interface : A comparison of pixel- and object-based classifications using high-resolution aerial photography. *Computers, Environment and Urban Systems*, 32(4):317–326, 2008.
- [21] D. CREVIER et R. LEPAGE : Knowledge-based image understanding systems : a survey. *Computer Vision and Image Understanding*, 67(2):161–185, 1997.
- [22] A. DARWISH, K. LEUKERT et W. REINHARDT : Image segmentation for the purpose of object-based classification. *Dans Proceedings of the IEEE International Geoscience and Remote Sensing Symposium 2003*, volume 3, pages 2039–2041. IEEE Computer Society, 2003.
- [23] O. DEBEIR : *Segmentation supervisée d'images*. Thèse de doctorat, Université Libre de Bruxelles, 2001.
- [24] S. DERIVAUX, N. DURAND et C. WEMMERT : On the complementarity of an ontology and a nearest neighbour for remotely sensed image interpretation. *Dans Proceedings of IEEE International Geosciences and Remote Sensing Symposium*, pages 3983–3986.
- [25] S. DERIVAUX, S. LEFÈVRE, Wemmert C. et J. KORCZAK : On machine learning in watershed segmentation. *Dans IEEE Workshop on Machine Learning for Signal Processing*, pages 187–192. IEEE Computer Society, 2007.
- [26] S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : On the use of a fuzzy unsupervised classification for watershed segmentation. *Dans Proceedings of the International Conference on Spatial Analysis and Geomatics (SAGEO)*, 2006.
- [27] S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : Watershed segmentation of remotely sensed images based on a supervised fuzzy pixel classification. *Dans Proceedings IEEE International Geoscience And Remote Sensing Symposium (IGARSS 2006)*, pages 3712–3715. IEEE Computer Society, 2006.
- [28] S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : Segmentation par ligne de partage des eaux basée sur des connaissances texturales. *Dans XXIème colloque GRETSI, Traitement du Signal et des Images*, Septembre 2007.

- [29] S. DERIVAUX, C. WEMMERT, S. LEFÈVRE et J. KORCZAK : Paramétrisation de méthodes de segmentation par utilisation de connaissances et approche génétique. *Dans Atelier Extraction de Connaissance à partir d'Images (ECOI), Journées Francophones Extraction et Gestion des Connaissances (EGC)*, Namur, Belgium, Janvier 2007.
- [30] T. G. DIETTERICH : Ensemble methods in machine learning. *Dans Proceedings of the First International Workshop on Multiple Classifier Systems*, volume 1857 de *Lecture Notes in Computer Science*, pages 1–15, 2000.
- [31] E. W. DIJKSTRA : *Selected writings on Computing : A Personal Perspective*, chapitre On the role of scientific thought. Springer-Verlag New York, Inc., 1982.
- [32] B. DRAPER, A. COLLINS, J. BROLIO, A. HANSON et E. RISEMAN : The schema system. *International Journal of Computer Vision*, 2(3):209–250, 1989.
- [33] C DREYFUS, J.-M. MARTINEZ, M. SAMUELIDES, M. B. GORDON, F. BADRAN, S. THIRIA et L. HÉRAULT : *Réseaux de neurones*. Eyrolles, 2002.
- [34] J. DRÉO, A. PETROWSKI, E. TAILLARD et P. SIARRY : *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003.
- [35] N. DURAND, S. DERIVAUX, G. FORESTIER, C. WEMMERT, P. GAŃCARSKI, O. BOUSSAÏD et A. PUISSANT : Ontology-based object recognition for remote sensing image interpretation. *Dans Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 472–479. IEEE Computer Society, 2007.
- [36] S. DZEROSKI et N. LAVRAC : *Relational Data Mining*. Springer, 2001.
- [37] M. E. FARMER et D. SHUGARS : Application of genetic algorithms for wrapper-based image segmentation and classification. *Dans Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1300–1307. IEEE Computer Society, 2006.
- [38] U. M. FAYYAD et K. B. IRANI : Multi-interval discretization of continuous valued attributes for classification learning. *Dans Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, volume 2, pages 1022–1027. Morgan Kaufmann Publishers, 1993.
- [39] U. M. FAYYAD, G. PIATETSKY-SHAPIRO et P. SMYTH : Knowledge discovery and data mining : Towards a unifying framework. *Dans Proceedings of KDD and DM*, pages 82–88, 1996.
- [40] R. Q. FEITOSA, G. A. COSTA, T. B. CAZES et B. FEIJO : A genetic approach for the automatic adaptation of segmentation parameters. *Dans Proceedings of the International Conference on Object-based Image Analysis*, 2006.
- [41] C. M. FONSECA et P. J. FLEMING : An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 1(3):1–16, 1996.
- [42] F. FONSECA, M. EGENHOFER, P. AGOURIS et G. CAMARA : Using ontologies for integrated geographic information systems. *Transactions in Geographic Information Systems*, 6(3): 231–257, 2002.
- [43] G. FORESTIER, S. DERIVAUX, C. WEMMERT et P. GAŃCARSKI : An evolutionary approach for ontology driven image interpretation. *Dans Tenth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing*, volume 4974 de *Lecture Notes in Computer Sciences*, pages 295–304, Napoli, Italy, Mars 2008. Springer.
- [44] G. FORESTIER, S. DERIVAUX, C. WEMMERT et P. GAŃCARSKI : Interprétation d'images basée sur une approche évolutive guidée par une ontologie. *Dans Journées Francophones*

- Extraction et Gestion des Connaissances (EGC 2008)*, volume 2, pages 469–474, Sophia Antipolis, France, January 2008.
- [45] F. FÜRST : *Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation*. Thèse de doctorat, Université de Nantes, 2004.
- [46] A. GERSHO et Gray R. : *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
- [47] G. GIACINTO et F. ROLI : Ensembles of neural networks for soft classification of remote sensing images. *Dans Proceedings of the European Symposium on Intelligent Techniques*, pages 166–170, 1997.
- [48] G. GIACINTO et F. ROLI : An approach to the automatic design of multiple classifier systems. *Pattern Recognition Letters*, 22(1):25–33, 2001.
- [49] X. GIGANDET, M.B. CUADRA, A. POINTET, L. CAMMOUN, R. CALOZ et J.-Ph. THIRAN : Region-based satellite image classification : method and validation. *Dans IEEE International Conference on Image Processing*, volume 3, pages 832–836. IEEE Computer Society, 2005.
- [50] D. GOLDBERG : *Genetic Algorithms in Search, Optimization and Machine Learning*. Kluwer Academic Publishers, 1989.
- [51] V. GRAU, A.U.J. MEWES, M. ALCANIZ, R. KIKINIS et S.K. WARFIELD : Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on Medical Imaging*, 23(4):447–458, 2004.
- [52] T. R. GRUBER : *Formal Ontology in Conceptual Analysis and Knowledge Representation*, chapitre Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Kluwer Academic Publishers, 1993.
- [53] T.R. GRUBER : Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5/6):907–928, 1995.
- [54] S. HAKER, G. SAPIRO et A. TANNENBAUM : Knowledge-based segmentation of SAR data with learned priors. *IEEE Transactions on Image Processing*, 9(2):299–301, 2000.
- [55] G. HAMARNEH et Xiaoxing LI : Watershed segmentation using prior shape and appearance knowledge. *Image and Vision Computing*, 2007.
- [56] L. K. HANSEN et P. SALAMON : Neural network ensemble. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
- [57] K. HARIS, S. N. EFSTRADIADIS, N. MAGLAVERAS et A. K. KATSAGGELOS : Hybrid image segmentation using watersheds and fast region merging. *IEEE Transaction On Image Processing*, 7(12):1684–1699, 1998.
- [58] J.-P. HATON, N. BOUZID, F. CHARPILLET, M.-C. HATON, B. LAASRI, H. LAASRI, P. MARQUIS, T. MONDOT et A. NAPOLI : *Le raisonnement en intelligence artificielle*. InterEditions, 1991.
- [59] S.-Y. HO et K.-Z. LEE : An efficient evolutionary image segmentation algorithm. *Dans Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1327–1334. IEEE Computer Society, 2001.
- [60] C. HUDELLOT : *Towards a Cognitive Vision Platform for Semantic Image Interpretation ; Application to the Recognition of Biological Organisms*. Thèse de doctorat, Université de Nice, 2005.

- [61] W. HÜRSCH et C. LOPES : Separation of concerns. *Dans Technical report by the College of Computer Science*, Northeastern University, 1995.
- [62] P. JACCARD : The distribution of flora in the alpine zone. *The New Phytologist*, 11(2):37–50, 1912.
- [63] L.L.F JANSSEN et M. MOLENAAR : Terrain objects, their dynamics and their monitoring by the integration of gis and remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 33:749–758, 1995.
- [64] X. JIN et C. H. DAVID : A genetic image segmentation algorithm with a fuzzy-based evaluation function. *Dans IEEE International Conference on Fuzzy Systems*, volume 2, pages 938–943. IEEE Computer Society, 2003.
- [65] Xiaoying JIN et C. H. DAVIS : Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information. *EURASIP Journal on Applied Signal Processing*, (14):2196–2206, 2005.
- [66] M. KASS, A. WITKIN et D. TERZOPOULOS : Snakes : Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [67] Y.-W. KIM et I.-S. OH : Classifier ensemble selection using hybrid genetic algorithms. *Pattern Recognition Letters*, 29(6):796–802, 2008.
- [68] L. A. KIRA, K. and Rendell : A practical approach to feature selection. *Dans Proceedings of the ninth international workshop on Machine learning*, pages 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [69] J. KITTLER, M. HATEF, R. P. W. DUIN et J. MATAS : On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [70] A. KNOBBE, M. de HAUS et A. SIEBES : Propositionalisation and Aggregates. *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-2001)*, pages 277–288, 2001.
- [71] M.-A. KROGEL et S. WROBEL : Feature selection for propositionalization. *Dans Proceedings of the 5th International Conference on Discovery Science*, Lecture Notes in Computer Science, pages 430–434, 2002.
- [72] N. LACHICHE : Good and bad practices in propositionalisation. *Dans Proceedings of Advances in Artificial Intelligence, 9th Congress of the Italian Association for Artificial Intelligence (AI*IA'05)*, pages 50–61, 2005.
- [73] S. LEFÈVRE, J. WEBER et D. SHEEREN : Automatic building extraction in vhr images using advanced morphological operators. *Dans Proceedings of the IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (URBAN)*, 2006.
- [74] S. LEFÈVRE : Knowledge from markers in watershed segmentation. *Dans IAPR International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 4673 de *Lecture Notes in Computer Sciences*, 2007.
- [75] S. LEFÈVRE, J. WEBER et D. SHEEREN : Automatic building extraction in vhr images using advanced morphological operators. *Dans IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (URBAN)*. IEEE Computer Society, 2007.
- [76] I. LEVNER et H. ZHANG : Classification-driven watershed segmentation. *IEEE Transaction on Image Processing*, 16(5):1437–1445, 2007.
- [77] X. LI et G. HAMARNEH : Modeling prior shape and appearance knowledge in watershed segmentation. *Dans Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, pages 27–33. IEEE Computer Society, 2005.

- [78] S. LIU, M. THONNAT et M. BERTHOD : Automatic classification of planktonic foraminifera by a knowledge-based system. *Dans Proceedings 10th Conference on Artificial Intelligence for Applications*, pages 358–364. IEEE Computer Society, 1994.
- [79] Y. LIU, M. LI, L. MAO, F. XU et S. HUANG : Review of remotely sensed imagery classification patterns based on object-oriented image analysis. *Chinese Geographical Science*, 16(3):282–288, 2006.
- [80] G. LO BOSCO : A genetic algorithm for image segmentation. *Dans Proceedings of the IEEE International Conference on Image Analysis and Processing*, pages 262–266. IEEE Computer Society, 2001.
- [81] E. LOPEZ-ORNELAS, F. SÈDES, G. FLOUZAT et F. LAPORTERIE-DEJEAN : Une approche objet pour la description des images à très haute résolution spatiale et l'extraction de connaissances. *Téledétection*, 5(1):261–273, janvier 2006. ISSN 1028-7736.
- [82] N. MAILLOT : *Ontology Based Object Learning and Recognition*. Thèse de doctorat, Université de Nice, 2005.
- [83] O. MARINO DREWS : *Raisonnement classificatoire dans une représentation à objets multi-points de vue*. Thèse de doctorat, Université J. Fourier, Grenoble, 1993.
- [84] V. MARTIN : *Cognitive Vision : Supervised Learning for Image and Video Segmentation*. Thèse de doctorat, Université de Nice, 2007.
- [85] V. MARTIN, N. MAILLOT et M. THONNAT : A learning approach for adaptive image segmentation. *Dans IEEE International Conference on Computer Vision Systems*, pages 40–48. IEEE Computer Society, 2006.
- [86] G. MARTINEZ-MUNOZ et A. SUAREZ : Pruning in ordered bagging ensembles. *Dans Proceedings of the 23rd International Conference on Machine learning*, pages 609–616, 2006.
- [87] T. MATSUYAMA et V.-S. HWANG : *SIGMA - A Knowledge-Based Aerial Image Understanding System*. Plenum Press New York USA, 1990.
- [88] F. MEYER et S. BEUCHER : Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, 1990.
- [89] V. MEZARIS, I. KOMPATSIARIS et M. G. STRINTZIS : Region-based image retrieval using an object ontology and relevance feedback. *EURASIP Journal on Advances in Signal Processing*, 2004(1):886–901, 2004.
- [90] M. MINSKY : A framework for representing knowledge. *Dans The Psychology of Computer Vision*, pages 211–281. McGraw-Hill, 1975.
- [91] T. M. MITCHELL : *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1993.
- [92] L. NAJMAN et M. SCHMITT : Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1163–1173, 1996.
- [93] A. NARASIMHAMURTHY : Evaluation of diversity measures for binary classifier ensembles. *Dans Multiple Classifier Systems*, volume 3541 de *Lecture Notes in Computer Science*, pages 267–277, 2005.
- [94] M. R. OGIELA et R. TADEUSIEWICZ : *Modern Computational Intelligence Methods for the Interpretation of Medical Images*. Springer, 2008.
- [95] S. M. OMOHUNDRO : Five balltree construction algorithms. Rapport technique TR-89-063, International Computer Science Institute, Decembre 1989.

- [96] M. PAL : Ensemble of support vector machines for land cover classification. *International Journal of Remote Sensing*, 29(10):3043–3049, 2008.
- [97] N. R. PALN et S. K. PAL : A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.
- [98] P. PANAGI, S. DASIOPOULOU, G. Th. PAPADOPOULOS, I. KOMPATSIARIS et M. G. STRINTZIS : A genetic algorithm approach to ontology-driven semantic image analysis. *Dans Proceedings of the 3rd IEE International Conference of Visual Information Engineering (VIE 2006)*, pages 132–137. IEEE Computer Society, 2006.
- [99] G. T. PAPADOPOULOS, V. MEZARIS, S. DASIOPOULOU et I. KOMPATSIARIS : Semantic image analysis using a learning approach and spatial context. *Dans Proceedings of the International Conference on Semantics And digital Media Technologies (SAMT)*, volume 4306 de *Lecture Notes in Computer Science*, pages 199–211, 2006.
- [100] R. PAREDES et E. VIDAL : Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1100–1110, 2006.
- [101] V. PATHAK et O. DIKSHIT : Segment based classification of indian urban environment. *Dans Proceedings of the MAP India Conference*, 2004.
- [102] C. PERLICH et F. PROVOST : Aggregation-based Feature Invention and Relational Concept Classes. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, 2003.
- [103] P. PERNER : A knowledge-based image-inspection system for automatic defect recognition, classification, and process diagnosis. *Machine vision and applications*, 7(3):135–147, 1994.
- [104] R. PETERI, J. CELLE et T. RANCHIN : Detection and extraction of road networks from high resolution satellite images. *Dans Proceedings of the IEEE International Conference on Image Processing*, pages 301–304. IEEE Computer Society, 2003.
- [105] G. PIGNALBERI, R. CUCCHIARA, L. CINQUE et S. LEVIALDI : Tuning range image segmentation by genetic algorithm. *EURASIP Journal on Applied Signal Processing*, 2003(8):780–790, 2003.
- [106] A. PUISSANT, T. RANCHIN, C. WEBER et A. SERRADJ : Fusion of quickbird ms and pan data for urban studies. *Dans Proceedings of the EARSeL Symposium*, pages 77–83, 2003.
- [107] A. RENOUF : *Modélisation de la formulation d'applications de traitement d'images*. Thèse de doctorat, Université de Caen, 2007.
- [108] M. A. RODRIGUEZ et M. J. EGENHOFER : Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, 2003.
- [109] C. ROUSSEY : *Une méthode d'indexation sémantique adaptée au corpus multilingues*. Thèse de doctorat, Insa de Lyon, 2001.
- [110] L. ROUX et J. DESACHY : Information fusion for supervised classification in a satellite image. *Dans Proceedings of the IEEE International Conference on Fuzzy Systems*, volume 3, pages 1119–1124. IEEE Computer Society, 1995.
- [111] M. SADGAL, A. EL FAZZIKI et A. A. OUAHMAN : Aerial image processing and object recognition. *The Visual Computer*, 21:118–123, 2005.
- [112] A. SARANLI et M. DEMIREKLER : On output independence and complementariness in rank-based multiple classifier decision systems. *Pattern Recognition*, 34:2319–2330, 2001.

- [113] P. SAVICKY et J. FURNKRANZ : Combining pairwise classifiers with stacking. *Dans Cryptographic Hardware and Embedded Systems*, volume 2779 de *Lecture Notes in Computer Science*, pages 219–229, 2003.
- [114] A. SCHWERING et M. RAUBAL : Measuring semantic similarity between geospatial conceptual regions. *Dans Proceedings 1st International Conference on GeoSpatial Semantics (GeoS)*, volume 3799 de *Lecture Notes in Computer Science*, pages 90–106, Mexico City, Mexico, 2005.
- [115] P. SOILLE : *Morphological Image Analysis*. 2nd ed. Springer-Verlag, 2003.
- [116] A. SONG et V. CIESIELSKI : Fast texture segmentation using genetic programming. *Dans Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 2126–2133. IEEE Computer Society, 2003.
- [117] L. TODOROVSKI et S. DZEROSKI : Combining classifiers with meta decision trees. *Machine Learning*, 50(3):223–249, 2003.
- [118] R. TRIAS-SANZA, G. STAMON et J. LOUCHET : Using colour, texture, and hierarchial segmentation for high-resolution remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):156–168, 2008.
- [119] J. TRICART, S. RIMBERT et G. LUTZ : *Introduction à l'utilisation des photographies aériennes*. Sédes, 1970.
- [120] K. TUMER et Ghosh J. : Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–403, 1996.
- [121] A. TVERSKY : Features of similarity. *Psychological Review*, 84:327–352, 1977.
- [122] J. K. UHLMANN : Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40(4):175–179, November 1991.
- [123] L. VINCENT et P. SOILLE : Watersheds in digital spaces : An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [124] M. VOLLE : *Analyse des données*. Economica, 1997.
- [125] L. WANG, L. KHAN et C. BREEN : Object boundary detection for ontology-based image classification. *Dans Proceedings of the 3rd International Workshop on Multimedia Data Mining (MDM/KDD2002)*, pages 51–61, 2002.
- [126] T. WHITESIDE et W. AHMAD : A comparison of object-oriented and pixel-based classification methods for mapping land cover in northern australia. *Dans Proceedings Spatial Sciences Institute Biennial Conference (SSC 2005)*, pages 1225–1231, 2005.
- [127] I. H. WITTEN et E. FRANK : *Data mining : pratical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [128] N. YAGER et A. SOWMYA : Support vector machines for road extraction from remotely sensed images. *Dans Computer Analysis of Images and Patterns*, pages 285–292, 2004.
- [129] Y. J. ZHANG : A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.
- [130] Y. J. ZHANG : Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, 18:963–974, 1997.
- [131] Y. J. ZHANG : A review of recent evaluation methods for image segmentation. *Dans International Symposium on Signal Processing and its Applications*, volume 1, pages 148–151, 2001.

- [132] Y. J. ZHANG et J. J. GERBRANDS : Segmentation evaluation using ultimate measurement accuracy. *Dans Proceedings SPIE, Image Processing Algorithms and Techniques III*, volume 1657, pages 449–460, 1992.
- [133] H.J. ZHAO, J. KUMAGAI, M. NAKAGAWA et R. SHIBASAKI : Semi-automatic road extraction from high-resolution satellite image. *Dans Proceedings of the ISPRS Symposium on Photogrammetry and Computer Vision*, page A : 406, 2002.
- [134] Z.-H. ZHOU et W. TANG : *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, volume 2639 de *Lecture Notes in Computer Science*, chapitre Selective Ensemble of Decision Trees. Springer-Verlag New York, Inc., 2003.
- [135] N. ZLATOFF, B. TELLEZ et A. BASKURT : Image understanding using domain knowledge. *Dans Proceedings of Recherche d'information Assistée par Ordinateur (RIAO)*, pages 277–290, 2004.

Annexe A

Méthodes usuelles de segmentation

Il existe une multitude d’algorithmes de segmentation qui ont été développés depuis plusieurs décennies. Le lecteur désirant une vision plus large des algorithmes de segmentation peut se référer à Pal et Pal [97], Zhang [130] et, pour une évaluation dans le domaine des images satellites, à Carleer et al. [15].

Dans la taxonomie utilisée par Zhang [130], les algorithmes de segmentation sont regroupés en deux familles : les algorithmes basés frontières et les algorithmes basés régions. Les algorithmes de segmentation basés frontières recherchent les frontières (zones d’hétérogénéité) présentes dans l’image et en déduisent les régions. Les algorithmes de segmentation basés régions utilisent la propriété de similarité pour regrouper des pixels et former les régions.

Nous allons illustrer chacune de ces deux familles par l’algorithme qui nous semble le plus représentatif : la fusion de région pour la famille des algorithmes basés régions et la ligne de partage des eaux pour la famille des algorithmes basés frontières.

Segmentation par fusion de régions

Le premier algorithme présenté est celui de la fusion de régions [17]. Son mécanisme est simple et repose sur une hypothèse : *les régions connexes similaires doivent être fusionnées*. La connexité entre deux régions R_i et R_j est définie par l’équation A.1.

$$\text{connexite}(R_1, R_2) = \begin{cases} \text{vrai} & \text{si } \exists p_1 \in R_1, p_2 \in R_2 \mid \text{voisin}(p_1, p_2) = \text{vrai} \\ \text{faux} & \text{sinon} \end{cases} \quad (\text{A.1})$$

La notion de similarité peut être définie de différentes façons : texture, réponse spectrale, forme, ... Nous utilisons ici une mesure de dissimilarité *dis* (inverse de la similarité) comme la distance Euclidienne des signatures spectrales moyennes de chaque région. Soit B le nombre de bandes dans l’image et $\text{moy}(R_i, b)$ la fonction qui retourne la moyenne des pixels de la région R_i pour la bande b . La fonction $\text{dis}(R_i, R_j)$ peut être définie par l’équation suivante :

$$\text{dis}(R_i, R_j) = \begin{cases} \sqrt{\sum_{b=1}^B (\text{moy}(R_i, b) - \text{moy}(R_j, b))^2} & \text{si } \text{connexite}(R_i, R_j) = \text{vrai} \\ +\infty & \text{sinon} \end{cases} \quad (\text{A.2})$$

L’algorithme 6 détaille le processus de segmentation. Il est à noter que dans cet algorithme, l’ensemble des régions \mathcal{R} est initialisé avec des régions représentant un seul pixel. Il est possible

d'initialiser \mathcal{R} différemment, par exemple en s'appuyant sur le résultat d'une première segmentation.

Algorithme 6 Segmentation par fusion de régions.

Entrée : $\mathcal{R} = \{R_l\}$ l'ensemble des régions tel que $R_l = \{p_m\}$ avec p_m le pixel m de l'image
Entrée : t le seuil de dissimilarité maximale toléré pour fusionner deux régions
Sortie : \mathcal{R} la segmentation obtenue

tant que $\min_{R_i, R_j \in \mathcal{R}} \text{dis}(R_i, R_j) < t$ **faire**
 Soient R_1, R_2 tels que $(R_1, R_2) = \arg \min_{R_i, R_j \in \mathcal{R}} \text{dis}(R_i, R_j)$
 $R_m := R_1 \cup R_2$
 $\mathcal{R} := (\mathcal{R} \setminus \{R_i, R_j\}) \cup \{R_m\}$
fin tant que

L'application de cet algorithme sur l'image vue précédemment est illustrée à la figure A.1. Puisque l'algorithme nécessite un paramètre t , différentes valeurs de t produiront différentes segmentations. Nous pouvons ainsi observer sur la figure A.1 la très forte dépendance de la segmentation produite à la valeur de t choisie. Plus précisément, pour $t = 0,3$ la sur-segmentation est visiblement très importante. Néanmoins, il existe déjà certaines frontières qui sont manquantes par rapport aux images segmentées par les experts. À l'autre extrémité, pour $t = 0,7$, la plupart des objets de l'image sont fusionnés, il ne reste quasiment plus qu'une frontière entre le ciel et le reste de l'image.

Segmentation par ligne de partage des eaux

Contrairement à l'algorithme de fusion de régions que nous venons de voir, l'algorithme de segmentation par ligne de partage des eaux [123] est basé sur les frontières des régions plutôt que les caractéristiques globales des régions. La ligne de partage des eaux est la méthode de segmentation principale de la morphologie mathématique. Elle considère l'image à traiter comme une surface topographique, souvent (ce sera notre cas) en utilisant le gradient de l'image. Une image de gradient contient en chaque pixel la dérivée des valeurs locales de l'image originale, de telle sorte qu'un contour dans l'image originale correspondra à une forte valeur dans l'image de gradient. Inversement, une zone homogène dans l'image originale correspondra à une faible valeur dans l'image de gradient. Un exemple est donné à la figure A.2.

Pour le calcul du gradient, il est possible d'utiliser la méthode du gradient morphologique [115]. Dans le cas d'une image multi-valuée (par exemple une image multi-spectrale), le gradient peut-être calculé marginalement sur chaque bande de l'image. Afin de générer une image de gradient mono-valuée, les résultats marginaux sont alors combinés via le calcul pour chaque pixel de la norme Euclidienne tel qu'indiqué à l'équation A.3 avec $g(i, b)$ le gradient du pixel d'indice i pour la bande b d'une image à B bandes. D'autres techniques de gradient sont possibles, le lecteur pourra se référer à [2] pour un panorama des approches en morphologie mathématique multivariée. Toutefois, dans notre cas, le choix de l'opérateur de gradient a un impact très limité sur le processus global.

$$g(i) = \sqrt{\sum_{b=1}^B g(i, b)^2} \quad (\text{A.3})$$

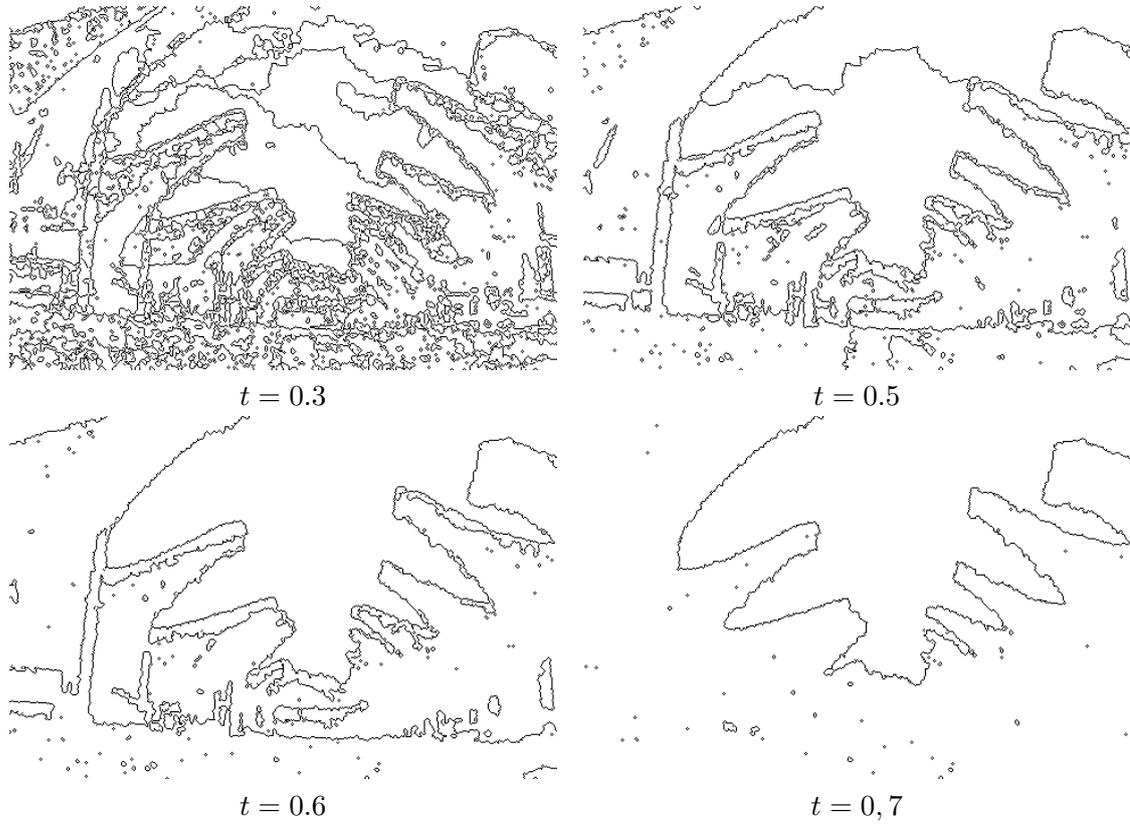


FIG. A.1 – Exemples de segmentations obtenues par l'algorithme de fusion des régions avec différentes valeurs de t .

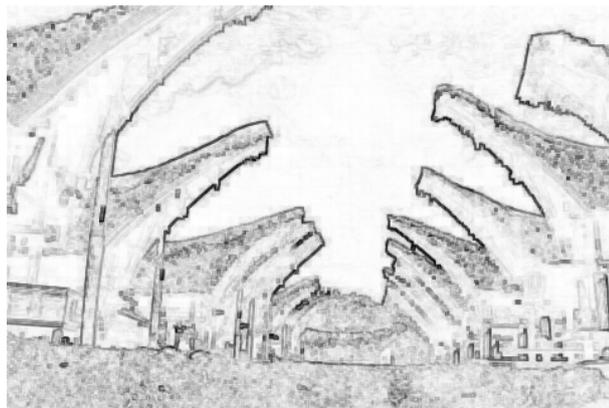


FIG. A.2 – Exemple de gradient (le blanc dénotant une valeur faible et le noir une valeur forte du gradient).

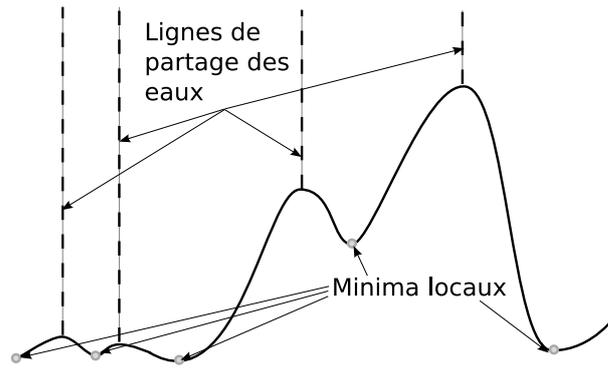


FIG. A.3 – Exemple de coupe de surface topographique avec les minima locaux et les lignes de partages des eaux créés.

Une fois cette surface topographique définie, elle est inondée à partir de ses minima locaux générant ainsi des bassins de rétention qui s'étendent au fur et à mesure du processus d'inondation. Des barrages sont construits aux endroits de contact d'eau de bassins différents. L'algorithme 7 détaille ce mécanisme d'inondation. La fonction *niveau*(p) retourne l'altitude du pixel p (ici la valeur du gradient de ce pixel). La fonction *croissance*(I, R_i, n), quant à elle, fait croître la région R_i en inondant les pixels voisins ayant une altitude n et n'appartenant pas déjà à une autre région. Une représentation graphique est donnée à la figure A.3. Le résultat de la segmentation est défini par la localisation de ces barrages (i.e. les lignes de partage des eaux).

Algorithme 7 Inondation de la surface topographique.

Entrée : I image topographique dont les niveaux d'altitude vont de `niveauMin` à `niveauMax`

Sortie : \mathcal{R} la segmentation obtenue

$\mathcal{R} := \emptyset$

pour tout $n \in [\text{niveauMin}, \dots, \text{niveauMax}]$ **faire**

pour tout $R_i \in \mathcal{R}$ **faire**

croissance(I, R_i, n)

fin pour

pour tout $p \in I \mid \text{niveau}(p) = n$ **faire**

si $\nexists R_i \in \mathcal{R} \mid p \in R_i$ **alors**

$R_i := \{p\}$

croissance(I, R_i, n)

$\mathcal{R} := \mathcal{R} \cup \{R_i\}$

fin si

fin pour

fin pour

La ligne de partage des eaux est connue pour sa tendance à favoriser les phénomènes de sur-segmentation. En effet, chaque minimum local créera une région. Cette méthode est donc très sensible au bruit. Afin de limiter son impact, une solution est de lisser chaque bande de l'image en appliquant un prétraitement tel qu'un filtre médian. Le filtre médian est connu pour sa capacité à conserver les arêtes de l'image. Dans nos expérimentations, nous utiliserons comme

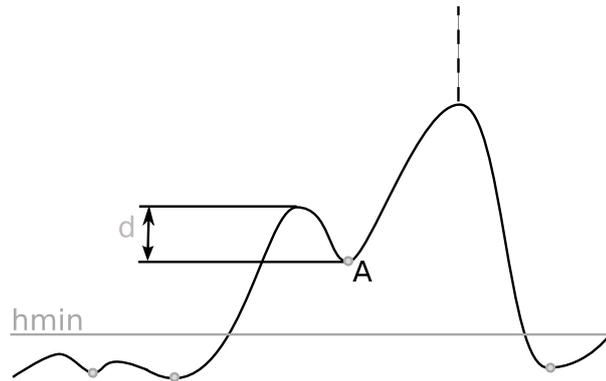


FIG. A.4 – Exemple d’application des méthodes de réduction de la sur-segmentation. La ligne *hmin* représente le seuil au dessous duquel la surface topographique est mise à zéro. Le point A montre un exemple de minima local supprimé par la méthode de réduction de dynamique car sa dynamique (illustrée par *d*) est insuffisante.

élément structurant un carré de taille 3×3 pixels.

Une fois la sur-segmentation issue du bruit de l’image supprimée, il peut encore rester une sur-segmentation liée à la complexité de l’image et des objets recherchés. Si chaque objet est représenté dans l’image par une couleur uniforme, il n’y aura pas de sur-segmentation. Néanmoins, les images sont souvent beaucoup plus complexes. L’objectif est donc ici de supprimer les régions les moins importantes. Nous rappelons ici deux techniques propres au paradigme de la ligne de partage des eaux pouvant être utilisées.

La première méthode est le seuillage de la surface topographique [57]. Cette méthode opère une simplification de cette surface. Soit $g(i)$ la valeur pixel d’indice i de la surface topographique et $hmin$ le paramètre du seuillage. La valeur $g'(i)$ du pixel d’indice i de la surface topographique résultante est définie par :

$$g'(i) = \begin{cases} g(i) & \text{si } g(i) > hmin \\ 0 & \text{sinon} \end{cases} \quad (\text{A.4})$$

Ainsi, les petites variations au sein des zones homogènes, qui se traduisent par de faibles valeurs de g , sont supprimées. Sur la figure A.4, la ligne *hmin* représente le seuil en dessous duquel les pixels doivent être considérés comme ayant une valeur nulle.

Une autre technique consiste à utiliser la profondeur des bassins [92]. Soit m_r le minimum local du bassin r et d_r l’altitude minimale à partir de laquelle il sera séparé d’un autre bassin par une ligne de partage des eaux. Tous les minima locaux pour lesquels on a $d_r - m_r < h_{dyn}$, avec h_{dyn} un seuil donné, ne seront pas considérés pendant l’étape d’immersion. Sur la figure A.4, le minimum local A ne sera pas pris en compte lors de l’immersion car sa dynamique d , la différence entre m_r et d_r , n’est pas assez importante. Cette technique permet de supprimer les petites variations d’hétérogénéité y compris dans les zones hétérogènes.

D’autres approches ont été explorées comme Lopez-Ornelas et al. [81] qui proposent deux techniques. La première consiste à effectuer des ouvertures et fermetures morphologiques sur la segmentation obtenue. La seconde effectue une fusion de régions selon des caractéristiques spatiales (comparaison de surface et longueur de contact entre les régions).

L’application de cet algorithme sur l’image vue précédemment est illustrée à la figure A.5. Sans l’utilisation de méthodes de réduction de sur-segmentation, l’image montre une très forte

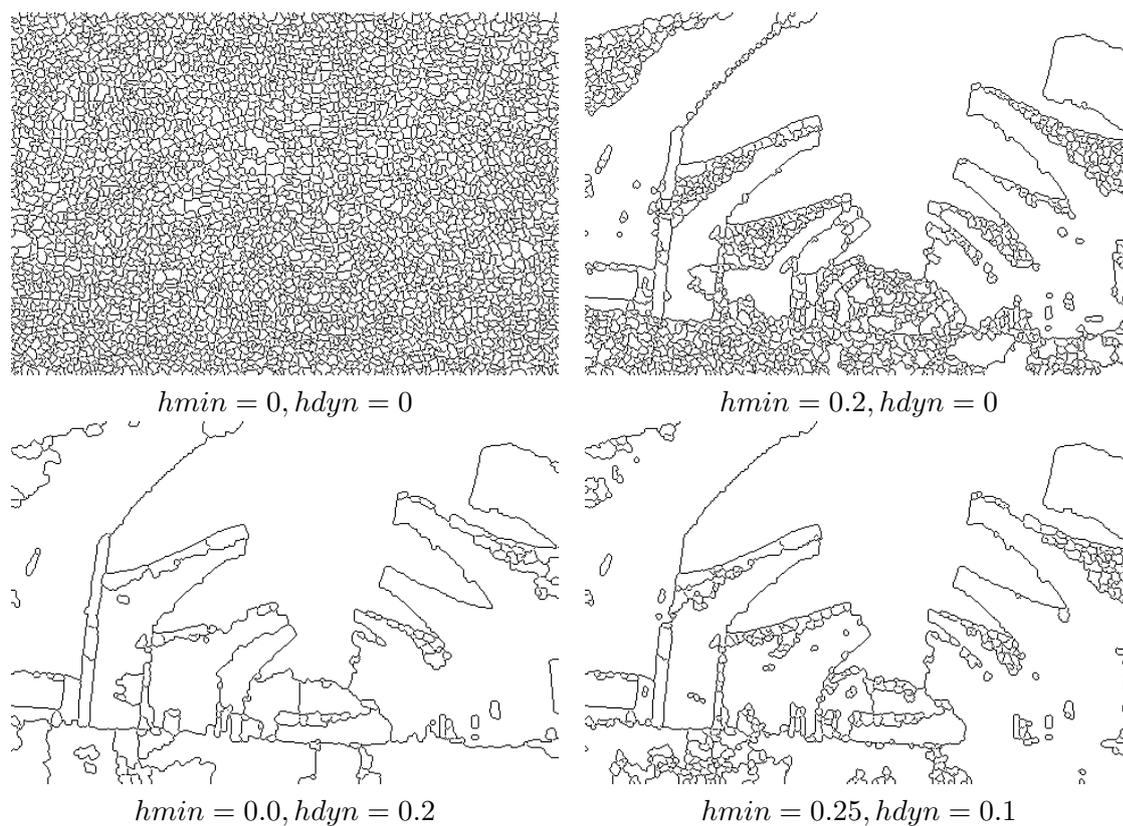


FIG. A.5 – Exemples de segmentations obtenues par l'algorithme de ligne de partage des eaux avec différentes valeurs de $hmin$ et $hdyn$.

sur-segmentation. En utilisant les méthodes de réduction de sur-segmentation, les résultats sont plus proches des segmentations faites par les experts.

Annexe B

Algorithme évolutionnaire

Un algorithme évolutionnaire [46] cherche à trouver, dans un espace de solution, la meilleure solution.

Plus formellement, si l'on considère une fonction f d'évaluation définie à l'équation B.1, avec G une solution prise dans l'espace des solutions \mathbb{G} , l'algorithme évolutionnaire cherche à implémenter la fonction d'optimisation définie à l'équation B.2.

$$\begin{aligned} f : \mathbb{G} &\rightarrow \mathbb{R} \\ G &\mapsto f(G) \quad \text{avec } f(G) \text{ l'évaluation de la solution } G \end{aligned} \tag{B.1}$$

$$\text{best}(\mathbb{G}, f) = \arg \max_{G \in \mathbb{G}} f(G) \tag{B.2}$$

Cette optimisation peut être considérée comme un processus d'apprentissage si, et seulement si, la solution G retenue peut aussi s'appliquer avec une bonne évaluation $f(G)$ sur d'autres jeux de données (non compris dans le jeu d'apprentissage).

Dans notre cas, nous définissons la solution G comme un vecteur contenant les paramètres qui doivent être automatiquement définis pour l'algorithme de segmentation donc $\mathbb{G} = [0; 1]^\rho$ avec ρ dépendant de l'algorithme de segmentation utilisé ($\rho = 3$ dans notre cas). On note $G[i]$ la valeur du paramètre i pour la solution G .

Une représentation graphique du processus de l'algorithme évolutionnaire est donnée à la figure B.1⁵.

L'algorithme évolutionnaire nécessite une population initiale de solutions pour débiter le processus évolutionnaire. Dans ce processus, la population va évoluer afin de contenir des solutions de plus en plus proches de la solution optimale. Pour créer la population initiale, chaque solution est choisie aléatoirement dans l'espace \mathbb{G} .

L'étape de sélection va créer des couples de solutions pour l'étape ultérieure de croisement. La sélection des solutions est biaisée par le résultat de leur évaluation. La probabilité de sélection d'une solution est définie à l'équation B.3 avec \mathcal{P} la population courante. Ainsi pour une population de taille $\Omega(\mathcal{P}) = 10$, la meilleure solution aura environ 18% de chance d'être sélectionnée tandis que la plus mauvaise n'en aura que 1.8%. Le rang d'une solution est le classement de son évaluation entre 1 pour la meilleure solution et $\Omega(\mathcal{P})$ pour la plus mauvaise.

$$\text{prob}(G, \mathcal{P}) = \frac{(\Omega(\mathcal{P}) - \text{rang}(G, \mathcal{P}) - 1)}{\sum_{i=1}^{\Omega(\mathcal{P})} i} \tag{B.3}$$

⁵Image sous licence CeCILL, copyleft Johann "nojhan" Dréo [34]

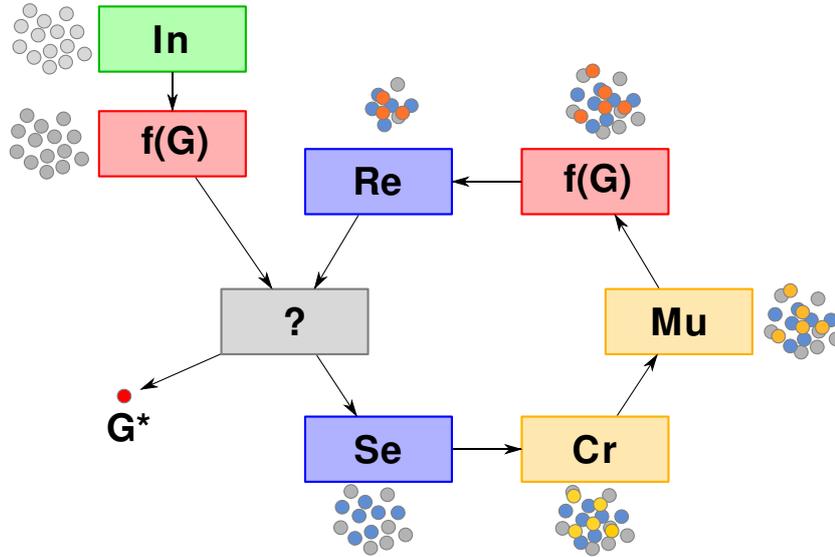


FIG. B.1 – Algorithme évolutionnaire. In : initialisation, $f(G)$: évaluation, ? : critère d'arrêt, Se : sélection, Cr : croisement, Mu : mutation, Re : remplacement, G^* : optimum trouvé.

Pour chacun de ces couples (G_1, G_2) , l'étape de croisement va combiner leurs paramètres pour donner une nouvelle solution G' avec $G'[i] = G_{p_i}[i] + \alpha_i \times |G_1[i] - G_2[i]|$ où α_i et p_i sont choisis aléatoirement dans $[-1; 1]$ et $\{1, 2\}$ respectivement.

L'étape de mutation peut remplacer chacun des paramètres de chacune des solutions par une valeur aléatoire avec une probabilité P_m . Cette technique est utilisée afin d'éviter d'être piégé dans un minimum local grâce à une exploration continue de l'espace des solutions.

Enfin, l'étape de remplacement décide quelles solutions vont former la population de la génération suivante. Nous prenons comme nouvelle population l'ensemble des nouvelles solutions ($\Omega(\mathcal{P}) - 1$ solutions) plus la meilleure solution de la génération précédente. Cette stratégie élitiste permet de toujours conserver le meilleur individu dans les générations.

Ces opérations se déroulent jusqu'à ce qu'un nombre pré-défini d'évolutions N_e soient effectuées. À ce moment, la meilleure solution obtenue jusqu'alors est retournée (G^*).

Annexe C

Méthodes usuelles de classification supervisée

Parmi les méthodes d'apprentissage à partir d'exemples utilisées pour classifier des régions, on peut séparer celles basées sur des prototypes (plus proches voisins [16] et Sadgal [111]) et celles basées sur des modèles (maximum likelihood [3], Mahalanobis distance classifier [49], séparateur à vaste marge [49] et perceptron multicouche [101]).

La première catégorie de méthodes conserve des exemples tels quels (une partie ou l'intégralité). Quand l'utilisateur cherche à connaître la classe d'une région, la méthode va rechercher quels sont les prototypes qui ressemblent à cette région, en fonction de la classe de ces prototypes. À partir de cette classe, la méthode va affecter une classe à la région.

La seconde catégorie, basée sur la génération de modèles, génère d'abord, à partir des exemples, une généralisation des données. La méthode des séparateurs à vaste marge [49] va par exemple chercher des hyperplans qui séparent les différentes classes. Les perceptron multicouche [101], vont eux chercher une modélisation mathématique à base de neurones (fonctions de base).

Nous nous baserons sur trois méthodes : k plus proches voisins, bayésien naïf et perceptron multicouche. La première méthode appartient à la catégorie qui utilise les exemples directement, tandis que les deux dernières génèrent un modèle. Le choix de ces trois méthodes tient au fait qu'elles sont sensiblement différentes au niveau de la capacité de généralisation ainsi que de leurs inconvénients et limitations spécifiques. Nous allons présenter brièvement ces méthodes et discuter leur adéquation aux données.

Méthode des k plus proches voisins

La méthode des k plus proches voisins [1] (ou k -NN) fonctionne de la façon suivante : lors de l'étape d'apprentissage, elle ne fait que stocker les exemples du jeu d'apprentissage. Le stockage se fait en utilisant l'algorithme *Balltree* [95, 122] afin de pouvoir retrouver rapidement des exemples proches d'un point de l'espace des données ($O(\log(n))$ au lieu de $O(n)$ avec n le nombre d'exemples). Lorsqu'une région doit être étiquetée par une classe, l'algorithme recherche les k régions du jeu d'apprentissage qui sont les plus proches dans l'espace des données. Soit x la région dont la classe est recherchée. Soit x_j pour $j = 1 \dots k$ les k régions les plus proches du jeu d'apprentissage dans l'espace des données. Soit $d(x, x_j)$ la distance entre x et x_j . On définit p_{xi} la probabilité d'appartenance de x à la classe C_i (avec $|C|$ classes possibles) par :

$$p_{xi} = \frac{1}{W} \sum_{j=0}^k w_{ji} \quad (\text{C.1})$$

$$\text{où } w_{ji} = \begin{cases} d(x, x_j)^{-1} & \text{si } x_j \text{ a pour classe } C_i \\ 0 & \text{sinon} \end{cases}$$

$$\text{et } W = \sum_{j=0}^k \sum_{i=1}^{|C|} w_{ji}$$

La région x a pour classe celle dont la probabilité p_{xi} est la plus grande.

La méthode des k plus proches voisins (abrégée en k -ppv) est relativement simple et connue pour donner souvent de bons résultats. Il a notamment été montré que si le nombre d'exemples d'apprentissage n augmente vers l'infini de telle sorte que $k/n \rightarrow 0$, la probabilité d'erreur approche le minimum théorique du jeu de données [127]. La méthode s'applique directement sur des attributs numériques. Le principal inconvénient de cette méthode est qu'elle ne peut détecter si un attribut a une importance plus grande qu'un autre. Si plusieurs attributs sont corrélés, ils auront un pouvoir de décision de la classe finale plus grand que d'autres attributs.

Enfin le choix du paramètre k dépend de la complexité du jeu de données, notamment de la séparabilité des classes et de la présence de bruit. Nous prendrons une valeur de $k = 3$ dans l'exemple d'évaluation que nous utiliserons. Ce nombre tient compte de la faible quantité d'exemples pour certaines classes (37 pour la route) mais permet néanmoins une certaine résistance au bruit et aux erreurs (car $k > 1$).

Perceptron multicouche

Les réseaux de neurones (nous nous limitons ici aux perceptrons multicouches) [33] peuvent être vus comme des outils de modélisation non linéaire par apprentissage. Un perceptron est composé de neurones qui sont des fonctions algébriques non linéaires, paramétrées et à valeurs bornées. Dans notre cas, un neurone prend en entrée un potentiel qui est une combinaison linéaire d'un ensemble de n valeurs $\{x_i | 1 \leq i \leq n\}$ pondérées par des paramètres (des poids) $\{w_i | 1 \leq i \leq n\}$ et d'un paramètre de biais w_0 . Ce potentiel, noté v s'écrit donc :

$$v = w_0 + \sum_{i=1}^n w_i x_i \quad (\text{C.2})$$

Ce potentiel v est passé dans une fonction d'activation sigmoïde pour donner le résultat y du neurone :

$$y = \frac{1}{1 + e^{-v}} \quad (\text{C.3})$$

Un perceptron multicouches est un réseau de neurones non bouclés à couches. Il est composé d'une couche de neurones de sortie (un par classe recherchée), d'une couche d'entrée composée des valeurs des attributs de l'exemple à classifier et d'une ou plusieurs couches de neurones cachés. Les résultats des neurones de la couche de sortie du réseau sont donc des fonctions algébriques non linéaires des entrées et des paramètres du réseau. Une illustration d'un réseau de neurones est donnée à la figure C.1.

La classe affectée à une région est celle dont la valeur de sortie du neurone associé est la plus grande.

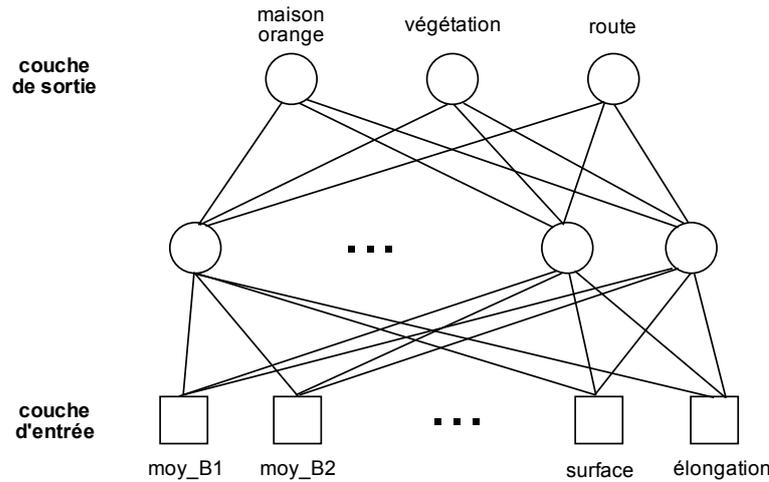


FIG. C.1 – Illustration d’un réseau de neurones. Chaque rond représente un neurone, chaque carré une entrée du réseau et chaque arc a une pondération associée.

L’apprentissage se fait par remontée du gradient. Au début, toutes les pondérations sont aléatoires. Chaque exemple d’apprentissage est appliqué successivement. On compare à chaque fois les valeurs de sortie du réseau et celles à obtenir (1 sur le bon neurone de sortie et -1 sur les autres). On modifie les poids afin de faire se rapprocher le résultat de celui attendu. Afin d’éviter le sur-apprentissage (le réseau est efficace sur le jeu d’apprentissage, mais mauvais sur le jeu de test), on garde 33% du jeu d’apprentissage à part et on évalue de temps en temps la qualité du réseau de neurones sur cette partie. Si le résultat baisse, on stoppe l’apprentissage du réseau de neurones. Pour plus d’informations, le lecteur peut se référer à Dreyfus [33].

Les réseaux de neurones ont l’avantage de pouvoir apprendre des fonctions très complexes. De plus, grâce aux pondérations, l’algorithme peut très bien gérer des groupes d’attributs corrélés ou bruités. Une des limites de l’algorithme est l’apprentissage des pondérations qui peut se finir sur un minimum local, c’est-à-dire trouver la meilleure solution d’une sous-partie de l’espace des pondérations, mais pas la meilleure solution de tout l’espace.

Bayésien naïf

L’apprentissage par méthode bayésienne naïve se base sur le théorème de Bayes [5] qui fournit la probabilité a posteriori d’une hypothèse h sachant les données D notée $P(h|D)$ par la formule :

$$P(h|D) = \frac{P(D|h) \times P(h)}{P(D)} \quad (\text{C.4})$$

On détermine la probabilité qu’une région appartienne à une classe (hypothèse h) en fonction de la valeur de ses attributs (les données D) à partir de la probabilité a priori de la classe $P(h)$, de la probabilité de la valeur des attributs pour une région $P(D)$ et de la probabilité qu’une région de la classe h ait pour valeurs d’attributs D , c’est-à-dire $P(D|h)$.

Comme les données sont en fait un vecteur d’attributs, $P(D|h)$ est calculé de façon naïve, c’est-à-dire en supposant que les attributs (l’ensemble des attributs est noté A) sont indépendants. Nous ne nous intéressons pas non plus à avoir une probabilité, il n’est donc pas nécessaire de normaliser par $P(D)$. La classe ayant la plus forte probabilité est celle dont la valeur $B(h|D)$ donnée par la formule suivante est la plus grande :

$$B(h|D) = P(h) \prod_{a \in A} P(a|h) \quad (\text{C.5})$$

Comme les attributs utilisés sont numériques, l'algorithme effectue d'abord une discrétisation supervisée [38].

La méthode bayésienne naïve permet des estimations plus locales qu'un modèle global comme les réseaux de neurones, mais généralise plus que la méthode des plus proches voisins. Cette méthode montre de bonnes capacités d'apprentissage même lorsque l'hypothèse d'indépendance des attributs n'est pas vérifiée [91].

Annexe D

Publications

D.1 Chapitre de livres

C. WEMMERT, G. FORESTIER et S. DERIVAUX : *Supervised and Unsupervised Ensemble Methods and their Applications*, chapitre Improving Supervised learning with Multiple Clusterings. Springer, 2009.

D.2 Conférences internationales avec comité de lecture

G. FORESTIER, S. DERIVAUX, C. WEMMERT et P. GANÇARSKI : An Evolutionary Approach for Ontology Driven Image Interpretation. *Dans Tenth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing*, volume 4974, pages 295–304. Springer, 2008.

S. DERIVAUX, G. FORESTIER et C. WEMMERT : Improving Supervised Learning with Multiple Clusterings. *Dans Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications in conjunction with ECAI*, pages 57–60. 2008.

S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : On Machine Learning in Watershed Segmentation. *Dans Proceedings of the IEEE Workshop on Machine Learning for Signal Processing*, pages 187–192. IEEE Computer Society, 2007.

N. DURAND, S. DERIVAUX, G. FORESTIER, C. WEMMERT, P. GANÇARSKI, O. BOUSSAÏD et A. PUISSANT : Ontology-based Object Recognition for Remote Sensing Image Interpretation. *Dans Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 472–479. IEEE Computer Society, 2007.

S. DERIVAUX, N. DURAND et C. WEMMERT : On the Complementarity of an Ontology and a Nearest Neighbour for Remotely Sensed Image Interpretation. *Dans Proceedings of IEEE International Geosciences and Remote Sensing Symposium*, pages 3983–3986. IEEE Computer Society, 2007.

S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : On the use of a fuzzy unsupervised classification for watershed segmentation. *Dans Proceedings of the International Conference on Spatial Analysis and Geomatics (SAGEO)*. 2006.

S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : Watershed Segmentation of Remotely Sensed Images Based on a Supervised Fuzzy Pixel Classification. *Dans Proceedings of IEEE International Geosciences and Remote Sensing Symposium*, pages 3712–3715. IEEE Computer Society, 2006.

N. LACHICHE, S. FUCHS, P. GAŃCARSKI et S. DERIVAUX : Discriminative power of related controls in the STULONG project. *Dans Discovery Challenge 2004, European Conference on Machine Learning and European Conferences on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'04)*, pages 139–145. 2004.

D.3 Conférences nationales avec comité de lecture

S. DERIVAUX, G. FORESTIER, C. WEMMERT et S. LEFÈVRE : Extraction de détecteurs d'objets urbains à partir d'une ontologie. *Dans Atelier Extraction de Connaissance à partir d'Images (ECOI), Journées Francophones Extraction et Gestion des Connaissances (EGC 2008)*, pages 71–81. 2008.

G. FORESTIER, S. DERIVAUX, C. WEMMERT et P. GAŃCARSKI : Interprétation d'images basée sur une approche évolutive guidée par une ontologie. *Dans Journées Francophones Extraction et Gestion des Connaissances (EGC 2008)*, volume 2, pages 469–474. 2008.

S. DERIVAUX, C. WEMMERT, S. LEFÈVRE et J. KORCZAK : Paramétrisation de méthodes de segmentation par utilisation de connaissances et approche génétique. *Dans Atelier Extraction de Connaissance à partir d'Images (ECOI), Journées Francophones Extraction et Gestion des Connaissances (EGC 2007)*. 2007.

S. DERIVAUX, S. LEFÈVRE, C. WEMMERT et J. KORCZAK : Segmentation par ligne de partage des eaux basée sur des connaissances texturales. *Dans XXIème colloque GRETSI, Traitement du Signal et des Images*. 2007.

S. DERIVAUX, A. BRAUD et N. LACHICHE : Prédiction de solubilité de molécules à partir des seules données relationnelles. *Dans Journées Francophones Extraction et Gestion des Connaissances (EGC 2006)*, pages 669–674. 2006.