



**HAL**  
open science

# Contributions à la segmentation d'image et de nuages de points. Application à l'analyse de scènes urbaines

Beatriz Marcotegui

## ► To cite this version:

Beatriz Marcotegui. Contributions à la segmentation d'image et de nuages de points. Application à l'analyse de scènes urbaines. Traitement des images [eess.IV]. Institut Pascal, UMR 6602 CNRS, Université Blaise Pascal, 2015. tel-01527577

**HAL Id: tel-01527577**

**<https://minesparis-psl.hal.science/tel-01527577v1>**

Submitted on 24 May 2017

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

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

Numéro d'ordre : 424

# UNIVERSITÉ BLAISE PASCAL – CLERMONT II

U.F.R. Sciences et Technologies

## HABILITATION À DIRIGER DES RECHERCHES

Institut Pascal, UMR 6602 CNRS, Université Blaise Pascal,  
Axe ISPR : Image, Systèmes de Perception pour la Robotique

Spécialité : Vision par ordinateur

Présentée et soutenue publiquement par

**Beatriz Marcotegui**

le 9 octobre 2015

### **Contributions à la segmentation d'image et de nuages de points. Application à l'analyse de scènes urbaines**

Synthèse des travaux scientifiques 1998 – 2015

Devant le jury composé de

Président :	PASCAL MONASSE	Ponts ParisTech
Rapporteurs :	ISABELLE BLOCH	Telecom ParisTech
	NICOLAS PAPANODITIS	IGN
	LAURENT TRASSOUDAINÉ	Université Blaise Pascal, Institut Pascal
Examineurs :	PAUL CHECCHIN	Université Blaise Pascal, Institut Pascal
	FERNAND MEYER	MINES ParisTech

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Segmentation morphologique à l'aide de graphes</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Segmentation et arbre de poids minimum (1998-2005) . . . . .	5
1.2.1 Calcul de l'arbre de poids minimum . . . . .	5
1.2.2 Segmentation interactive . . . . .	6
1.2.3 Algorithme de cascades et APM . . . . .	11
1.3 Ouverture ultime par attribut . . . . .	15
1.3.1 Définition . . . . .	15
1.3.2 Application à l'extraction de texte . . . . .	16
1.3.3 Evaluation sur la base Weizmann . . . . .	20
1.3.3.1 Description de la base et protocole d'évaluation . . . . .	20
1.3.3.2 Résultats . . . . .	22
1.4 Transitions graduelles : ouverture ultime par $\Delta$ -attribut . . . . .	26
1.5 Structures imbriquées : ouverture ultime avec une information de forme	30
1.5.1 Information de forme . . . . .	31
1.5.2 Ouverture ultime avec information forme . . . . .	33
1.5.3 Analyse de façades . . . . .	35
1.5.4 Localisation de texte enfoui . . . . .	37
1.5.5 Segmentation de cellules . . . . .	39
1.6 Algorithme rapide d'ouverture ultime utilisant l'arbre des maxima . . .	42
1.6.1 Max-Tree . . . . .	42
1.6.2 Calcul de résidus . . . . .	42
1.6.3 Calcul de l'ouverture ultime à partir des résidus . . . . .	43
1.6.4 Génération des images résultantes . . . . .	45
1.6.5 Temps de calcul . . . . .	46
1.6.6 Implémentation de $\Delta UAO$ sur le max-tree . . . . .	47
1.7 Profils d'attributs . . . . .	52
1.7.1 Décomposition par seuils et profil d'attributs . . . . .	52
1.7.2 Image de caractéristique liée à un attribut . . . . .	53
1.7.3 Un nouvel attribut : l'élongation stable . . . . .	54

1.8	Conclusion . . . . .	57
<b>2</b>	<b>Analyse de scènes urbaines</b>	<b>59</b>
2.1	Segmentation de façades (2006-2009) . . . . .	59
2.1.1	Segmentation de bâtiment . . . . .	60
2.1.2	Segmentation verticale et horizontale . . . . .	64
2.1.3	Localisation de fenêtres . . . . .	64
2.1.4	Résultats sur la base Teboul . . . . .	65
2.2	Analyse de nuages de points . . . . .	70
2.2.1	Structures de données . . . . .	71
2.2.2	Génération des images d'élévation . . . . .	72
2.2.3	Filtrage de points redondants . . . . .	73
2.2.4	Interpolation . . . . .	74
2.2.5	Segmentation de blocs de maisons . . . . .	80
2.2.6	Détection du sol . . . . .	83
2.2.7	Segmentation d'objets 3D . . . . .	83
2.2.8	Classification d'objets 3D . . . . .	88
2.2.8.1	Résultats sur la base TerraMobilita/iQmulus . . . . .	88
2.2.9	Diagnostic d'accessibilité . . . . .	91
2.2.9.1	Détection des bords de trottoir . . . . .	93
2.2.9.2	Reconnexion des bords de trottoir . . . . .	94
2.2.9.3	Résultats sur la base Enschede . . . . .	100
<b>3</b>	<b>Conclusion et perspectives</b>	<b>105</b>
<b>A</b>	<b>Curriculum Vitae</b>	<b>1</b>
<b>B</b>	<b>Publications</b>	<b>5</b>
B.1	Articles de journaux . . . . .	5
B.2	Articles de Conférence . . . . .	6
B.3	Synthèse de la production scientifique . . . . .	10
B.4	Activités d'enseignement . . . . .	11
<b>C</b>	<b>Activités d'encadrement</b>	<b>13</b>
C.1	Encadrement de doctorants . . . . .	13
C.2	Encadrement de post-doctorants . . . . .	14
C.3	Encadrement de stagiaires . . . . .	14

# Introduction

Ce document présente une synthèse des activités de recherche que j'ai menées depuis mon arrivée au Centre de Morphologie Mathématique (CMM) de MINES Paristech, à Fontainebleau, en 1998. Le CMM est un laboratoire d'analyse d'images créé en 1967, fondateur de la théorie éponyme. Le CMM a contribué à la croissance et la dissémination du corpus théorique et algorithmique de la morphologie mathématique. L'équipe est composée d'environ 25-30 personnes, dont une dizaine de permanents. Ses activités s'articulent autour de l'enseignement, de la recherche et de collaborations avec l'industrie. Toujours motivée par les applications industrielles, la recherche au CMM a produit une expertise originale et efficace dans de nombreux domaines (multimédia, science des matériaux, électronique, biologie, médecine, contrôle industriel, etc.). Ces domaines d'application fournissent un riche terrain d'expérimentation pour le développement de nouveaux concepts et outils. Comme on verra dans ce mémoire, mon activité au CMM correspond parfaitement au modèle de fonctionnement du centre, avec des contributions aussi bien dans l'enseignement que dans la recherche académique et partenariale. La plupart de ces travaux ont été réalisés en collaboration avec des étudiants en thèse ou post-doctorants. Je tiens à les remercier non seulement pour leur contribution scientifique mais également pour leurs qualités humaines.

Ce mémoire comprend deux parties. La première, décrite dans le chapitre 1, porte sur la segmentation morphologique à l'aide de graphes et s'inscrit dans la continuité des axes de recherche historiques du centre. Les apports principaux concernent l'utilisation d'une hiérarchie de partitions dans un contexte de segmentation interactive, l'utilisation de l'arbre de poids minimum pour accélérer les processus de segmentation fondés sur la ligne de partage des eaux et la proposition des opérateurs de segmentation intrinsèquement multi-échelle ainsi que leur implementation rapide à l'aide de l'arbre des maxima. La deuxième partie de ce manuscrit, détaillée dans le chapitre 2, porte sur l'analyse de scènes urbaines que nous avons développée depuis 2006. Nous détaillons ces travaux par la richesse, diversité et maturité des approches proposées. La stratégie qui consiste à appliquer la morphologie mathématique à une image d'élévation s'est révélée efficace aussi bien en termes de qualité de résultats que de temps de calcul, par rapport à l'état de l'art. Elle constitue une nouvelle thématique au CMM avec des perspectives de développement intéressantes. Finalement, nous dresserons un bilan et ouvrirons des perspectives dans le chapitre 3.



# Chapitre 1

## Segmentation morphologique à l'aide de graphes

### 1.1 Introduction

La segmentation a pour objectif d'isoler les différents objets d'intérêt présents dans une image. Elle constitue souvent une étape essentielle aussi bien pour l'interprétation sémantique que pour la quantification d'images. Depuis plusieurs décennies la segmentation est au cœur des activités en traitement d'images. Une multitude de méthodes existe dans la littérature telles que la ligne de partage des eaux [1, 2], les coupes de graphes [3, 4], mean-shift [5, 6], Maximally Stable Extremal Regions (MSER) [7, 8, 9] et les ensembles d'échelles [10, 11, 12]. Parmi ces méthodes, celles fondées sur des graphes sont particulièrement intéressantes. En effet, elles réunissent plusieurs pixels dans un seul nœud du graphe réduisant ainsi la complexité algorithmique et donnant lieu à des techniques efficaces en temps de calcul. Par ailleurs, elles permettent de caractériser les nœuds avec des informations de plus haut niveau, comme par exemple la forme, et aboutissent à des résultats plus pertinents que les approches locales manipulant des informations au niveau du pixel. Une littérature abondante existe sur ces méthodes et attire un intérêt croissant de nombreux chercheurs.

Par conséquent, je me suis concentrée sur les méthodes morphologiques à l'aide des graphes, qui offrent des performances intéressantes en termes de qualité des résultats et de temps de calcul. En particulier, j'ai proposé des méthodes de segmentation s'appuyant sur deux représentations d'arbre : l'Arbre de Poids Minimum (APM) et l'arbre des maxima (MaxTree).

Le chapitre est structuré en ordre chronologique. En arrivant au CMM, en 1998, j'ai pris la responsabilité du développement d'un prototype de segmentation interactive, dans le cadre du projet MoMuSys. Ces travaux s'inscrivent dans la continuité d'un des axes de recherche privilégiés au CMM : la segmentation morphologique. C'est dans ce contexte qui s'est déroulée la thèse de Francisca Zanoguera, que j'ai co-encadrée avec Fernand Meyer. La section 1.2 détaille les contributions de cette thèse. Premiè-

rement, l'intérêt d'utiliser l'arbre de poids minimum (APM) est rappelé et son calcul est explicité dans la section 1.2.1. L'APM synthétise les informations nécessaires pour implémenter de manière efficace tout processus lié aux inondations [13]. Ensuite, une boîte à outils originale et performante de segmentation interactive en utilisant l'APM est introduite dans la section 1.2. Après la thèse de F. Zanoguera, j'ai continué à exploiter les avantages de l'APM et j'ai développé un algorithme rapide de segmentation hiérarchique par cascades. Cet algorithme est présenté dans la section 1.2.3. J'ai été ensuite amenée à travailler sur la segmentation de texte enfoui. La grande variabilité des zones de texte à détecter rendait difficile la paramétrisation des méthodes de segmentation introduites précédemment. Je me suis alors tournée vers les approches résiduelles introduites au CMM par Serge Beucher et qui ont donné lieu à une littérature très abondante dans la communauté scientifique. Ces méthodes généralisent le concept de chapeau haut de forme dans un contexte multi-échelle. J'ai commencé par proposer une version de cet opérateur en utilisant des ouvertures par attribut, décrit dans la section 1.3. J'ai ensuite analysé le comportement de cet opérateur dans des situations problématiques. Cette analyse a donné lieu à la proposition de l'opérateur d'ouverture ultime par  $\Delta$ -attributs qui offre une solution aux transitions graduelles (décrit dans la section 1.4), et l'ouverture ultime avec information de forme qui permet d'aborder le problème lié aux structures imbriquées (section 1.5). La section 1.6 introduit un algorithme pour calculer l'ouverture ultime, ainsi que toutes les améliorations proposées, de manière efficace, en utilisant l'arbre des maxima. Finalement, la section 1.7 introduit des techniques qui explorent l'évolution d'attributs tels que l'élongation à travers une décomposition par seuils.



## 1.2 Segmentation et arbre de poids minimum (1998-2005)

L'Arbre de Poids Minimum (APM) [14, 15] d'un graphe  $G$  contenant  $N$  nœuds et  $E$  arêtes,  $G(N, E)$ , est un sous-ensemble de  $G$  contenant tous les nœuds,  $N$ , et  $N - 1$  arêtes. Les arêtes de l'APM doivent vérifier deux conditions : 1.- il existe un et seulement un chemin dans l'APM reliant n'importe quel paire de nœuds du graphe 2.- la condition précédente doit être remplie en choisissant les arêtes du graphe de plus petite valuation. Si toutes les arêtes du graphe ont des valuations différentes, l'APM est unique. Sinon, un choix entre arêtes de même valuation peut avoir lieu. Dans [13] Meyer montre que l'APM du graphe de voisinage d'une image est une structure efficace permettant de simuler tout processus d'inondation associé à l'approche de segmentation de ligne de partition des eaux. Des nombreux auteurs confirment l'intérêt de l'arbre de poids minimum pour les techniques de segmentation [13, 16, 17, 18].

Dans le cadre de la thèse de F. Zanoguera, que j'ai co-encadrée avec F. Meyer, l'APM est combiné aux valeurs d'extinction proposées par C. Vachier [19]. Nous avons utilisé les valeurs d'extinction volumiques qui ont un comportement se rapprochant du système visuel humain, en associant une importance aux régions qui dépend à la fois de leur taille et de leur contraste. Nous avons proposé de nombreux outils de segmentation interactive dans le cadre du projet MoMuSys (Mobile Multimedia Systems), de 1998 à 2000. J'ai porté la responsabilité du workpackage 5.2, dont le but était le développement du prototype de segmentation interactive, VOGUE.

Nous détaillons ci-dessous les points clés de l'algorithme. En particulier, le calcul de l'arbre de poids minimum de manière efficace lors du processus d'inondation ainsi que la manipulation de l'APM pour développer des outils de segmentation interactive.

### 1.2.1 Calcul de l'arbre de poids minimum

Plusieurs algorithmes existent dans la littérature pour calculer l'APM à partir d'un graphe de voisinage dont les arêtes sont valuées : les algorithmes de Prim [20], Kruskal [15] ou Dijkstra [21]. Néanmoins, nous proposons de le calculer en même temps que le processus d'inondation utilisé lors de l'obtention de la partition fine (le niveau le plus bas de la hiérarchie). Pour ceci, une arête est rajoutée chaque fois que deux régions se rencontrent pour la première fois. Une table d'équivalence permet d'éviter les cycles en gérant les régions déjà fusionnées. Cette procédure est décrite dans [22]. Par ailleurs, les valeurs d'extinction sont calculées lors de cette même inondation. L'intérêt de cette approche est d'obtenir une hiérarchie de partitions à moindre coût algorithmique par rapport à une simple ligne de partage des eaux.

La figure 1.1 illustre la création de l'arbre de poids minimum avec les arêtes valuées avec les valeurs d'extinction volumiques [19] lors de l'inondation. Le signal 1D contient 4 minima nommés de  $A$  à  $D$ . Le processus comporte trois étapes :

1. Les premiers minima qui se rencontrent lors de l'inondation sont les minima B et C. Le volume de ces deux lacs est comparé à cet instant. Le volume de B est 4 alors que celui de C est 6. Le lac C absorbe alors le lac B et une arête est rajoutée entre les noeuds correspondants du graphe. Cette arête prend comme valuation le volume du lac éteint (=4).
2. L'inondation continue et le lac D rencontre le lac BC. Une arête est alors rajoutée entre C et D avec la valeur 7 ( $7 = V_D < V_{BC} = 17$ ).
3. Finalement le lac A rencontre BCD. L'arête A – B est alors ajoutée avec une valuation de 9 ( $9 = V_A < V_{BCD} = 40$ ).

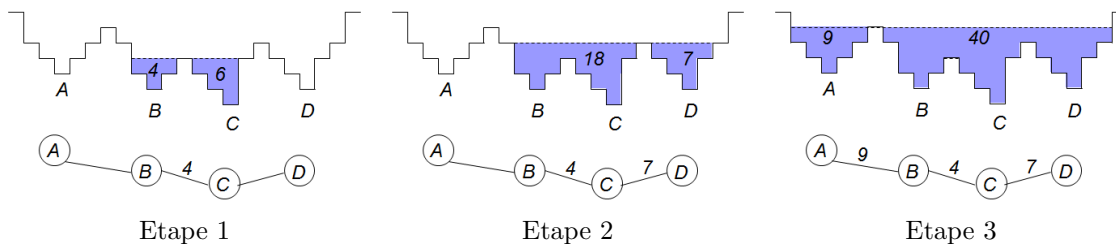


FIGURE 1.1 – Illustration de la création de l'arbre de poids minimum pendant l'inondation.

A la fin de l'inondation l'APM est disponible. Les valuations des arêtes correspondent aux valeurs d'extinction volumiques.

### 1.2.2 Segmentation interactive

Une fois l'APM calculé et les arêtes valuées avec les valeurs d'extinction, nous avons développé des outils interactifs pour naviguer dans la hiérarchie de partitions (voir figure 1.2). L'image de la figure 1.2(a) et son APM associé (figure 1.2(b)) sont segmentés en  $N$  régions ( $N=3$ ). La segmentation en  $N$  régions nécessite seulement de supprimer  $N-1$  arêtes et de relabeliser les différents sous-graphes obtenus (voir figure 1.2(b)). Une re-segmentation locale est aussi possible, en considérant seulement les arêtes au sein d'une région sélectionnée par l'utilisateur (voir figure 1.2(c)).

La segmentation à partir de marqueurs [2] est une technique largement utilisée pour éviter le problème de sur-segmentation de la ligne de partage des eaux. Dans un contexte de segmentation interactive, les marqueurs constituent un moyen simple d'interaction : l'utilisateur dessine des traits quelconques identifiant les régions d'intérêt. La localisation précise de la frontière est réalisée grâce à une procédure automatique. Cette technique peut aussi être implémentée en s'appuyant seulement sur l'APM au lieu de sur toute l'image, avec un gain en efficacité car l'arbre contient beaucoup moins de noeuds que l'image des pixels. La figure 1.3 illustre ce processus. Deux labels, jaune et rouge sont propagés dans l'ensemble du graphe, prenant les arêtes en contact avec les marqueurs en ordre croissant d'évaluation et transmettant la couleur du marqueur

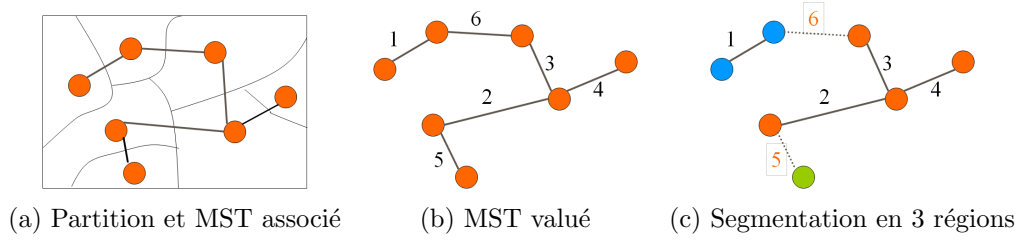


FIGURE 1.2 – Segmentation hiérarchique et Minimum Spanning Tree.

à l'autre extrémité de l'arête. Ce processus sera utilisé pour le calcul des cascades sur l'APM décrit dans la section 1.2.3.

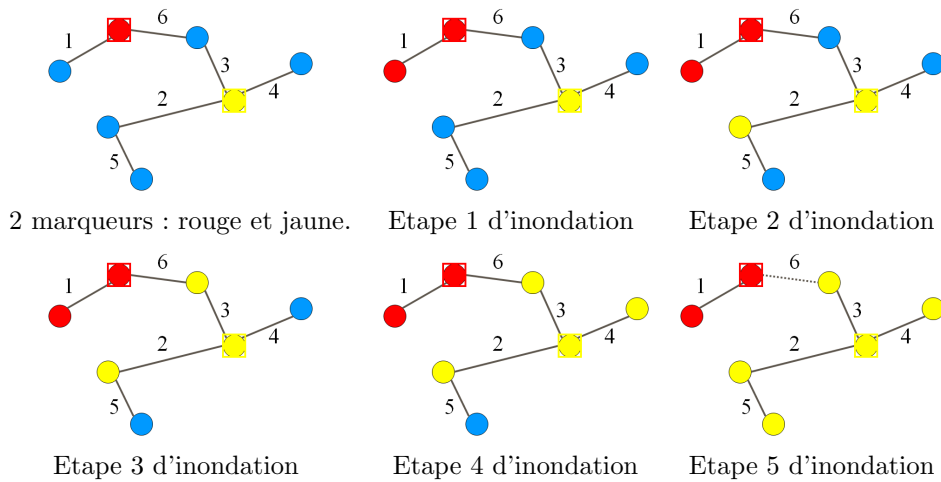
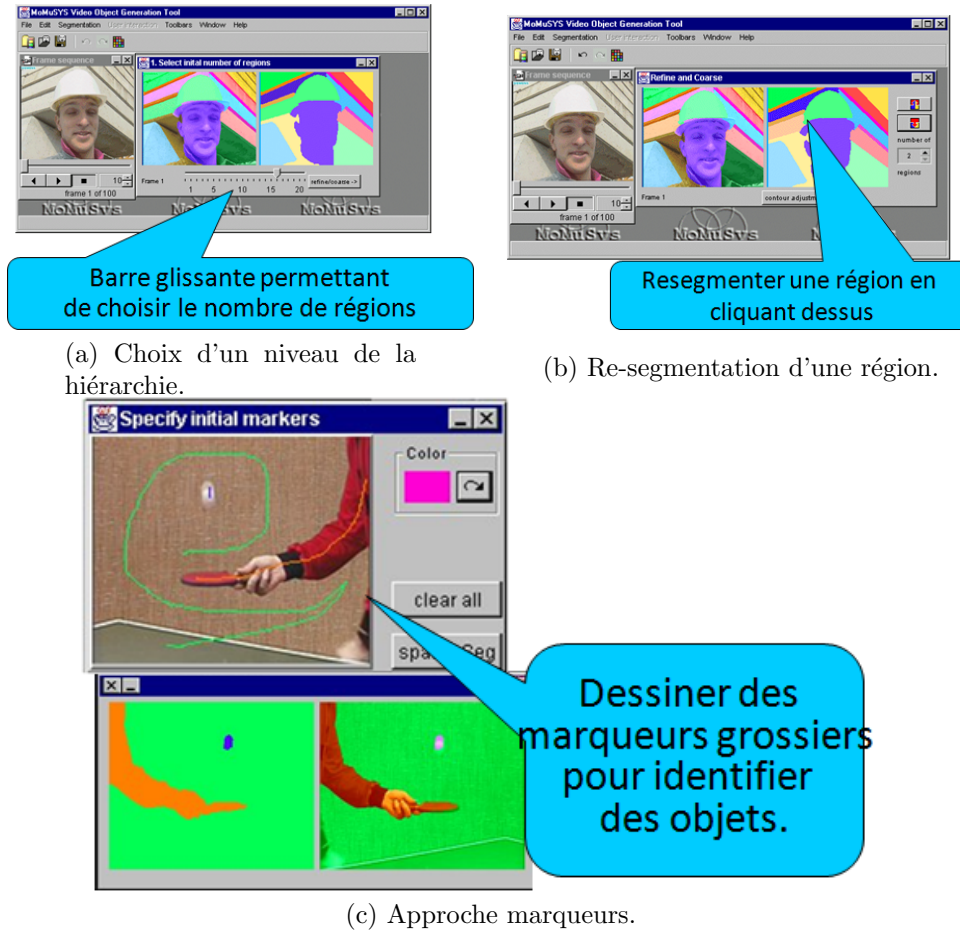


FIGURE 1.3 – Segmentation à partir de marqueurs sur l'APM.

Ces outils ont été intégrés dans une interface homme-machine, VOGUE, dans le cadre du projet MoMuSys. Des exemples d'utilisation de ce logiciel sur des images réelles sont illustrés sur la figure 1.4. Une barre glissante permet de naviguer dans la hiérarchie et de choisir un nombre de régions adapté au contenu de l'image (figure 1.4(a)). Cette procédure peut être appliquée localement, à l'intérieur d'une région préalablement identifiée. Dans notre exemple (figure 1.4(b)) le casque, d'une couleur proche au bâtiment derrière a été re-segmenté. Finalement l'utilisation de marqueur est montrée sur la figure 1.4(c).

La combinaison des valeurs d'extinction et l'APM donne lieu à des méthodes de segmentation interactive efficaces en termes de résultats et de temps de calcul. Ces méthodes sont appliquées dans un premier temps au domaine multimédia [22, 23] et ensuite elles sont utilisées dans le domaine médical [24]. Ces travaux ont par ailleurs fait l'objet d'un contrat de valorisation avec la société TDK.



(a) Choix d'un niveau de la hiérarchie.

(b) Re-segmentation d'une région.

(c) Approche marqueurs.

FIGURE 1.4 – Exemples de segmentation interactive intégrés dans VOGUE, logiciel développé dans le cadre du projet MoMuSys.

Nous avons étendu ces techniques aux séquences d'images, considérant des régions spatio-temporelles dans la thèse de Francisca Zanoguera [25, 26]. Par ailleurs, dans la prolongation de ce projet, j'ai développé d'autres outils d'édition, parmi lesquels la baguette magique s'appuyant sur une hiérarchie de partitions [27] est particulièrement intéressante.

**Baguette magique sur une segmentation hiérarchique** La baguette magique est un outil classique d'édition d'image. Elle est conçue pour sélectionner les objets de couleur uniforme. Son fonctionnement est le suivant : l'utilisateur clique sur un pixel à l'intérieur de l'objet. La couleur du pixel cliqué est sélectionnée. Un seuil de tolérance (paramètre de l'algorithme) est appliqué autour de la valeur obtenue. Le résultat est la région connexe autour du pixel cliqué, sélectionnée par le seuil de tolérance précédent. Si le résultat n'est pas conforme aux attentes, le seuil de tolérance peut être modifié

ou alors des clics supplémentaires peuvent rajouter ou supprimer les parties de l'objet erronées.

Le contour de l'objet ainsi obtenu peut être imprécis. En effet, la frontière est localisée sur une zone de transition entre l'objet et son fond où les changements de couleur sont importants. L'hypothèse d'uniformité de couleur n'est donc pas vérifiée. Augmenter la tolérance donne lieu souvent à des fuites avant d'obtenir l'objet complet. Par ailleurs, cliquer sur les zones de transition pour rajouter les parties manquantes peut être laborieux. J'ai proposé dans [27] une version améliorée de cet outil. Elle s'appuie sur la hiérarchie de partitions utilisant les valeurs d'extinction introduite ci-dessus. Agissant au niveau région au lieu d'au niveau pixel donne une meilleure précision des contours obtenus. Le principe de la baguette magique s'appuyant sur une hiérarchie de partitions est le suivant :

- premièrement, la baguette magique classique est appliquée ;
- le résultat précédent est étendu aux régions de la partition. Le choix du niveau de la hiérarchie peut être fait par l'utilisateur. Cependant, nous proposons une méthode adaptative pour sélectionner le bon niveau de partition. Cette sélection est faite en deux étapes :
  - sélection de régions de la partition fine dont l'intersection avec le résultat de la baguette magique classique n'est pas vide ;
  - l'ensemble de ces régions sélectionnées doivent être fusionnées car elles sont dans la gamme de couleurs sélectionnée par l'utilisateur. Dans le but d'exploiter l'information de la hiérarchie et non seulement d'une partition, nous recherchons le niveau de la hiérarchie (L) le plus bas pour lequel l'ensemble des régions touchées par la baguette magique classique sont réunies. La région du niveau L contenant le pixel sélectionné sera le résultat final. Autrement dit, les régions de la partition fine touchées par la baguette magique classique sont étendues aux régions voisines similaires (suivant l'ordre croissant de la hiérarchie) jusqu'à ce qu'elles soient toutes réunies dans une seule région.

La figure 1.5 illustre la comparaison entre la baguette magique classique et la baguette magique multi-échelle. L'objectif fixé est d'extraire la pomme de l'image "fruits" (fig. 1.5(a)). L'utilisateur clique alors à l'intérieur de la pomme. La figure 1.5(b) montre le résultat de la baguette magique classique. Nous observons que beaucoup de trous restent dans la sélection et le contour n'est pas précis. En augmentant la tolérance pour essayer d'améliorer le résultat (figure 1.5(c)) nous voyons que le contour est plus précis sur certaines parties de la pomme mais une fuite importante s'est produite au niveau des raisins. La figure 1.5(d) montre le résultat de la combinaison du résultat de la baguette classique (figure 1.5(b)) avec celui de la segmentation multi-échelle où le niveau de hiérarchie est choisi automatiquement. Nous constatons un bien meilleur résultat sans intervention supplémentaire de l'utilisateur.

**Publications associées :** ConfI{ [22], [23], [25], [28], [27].}

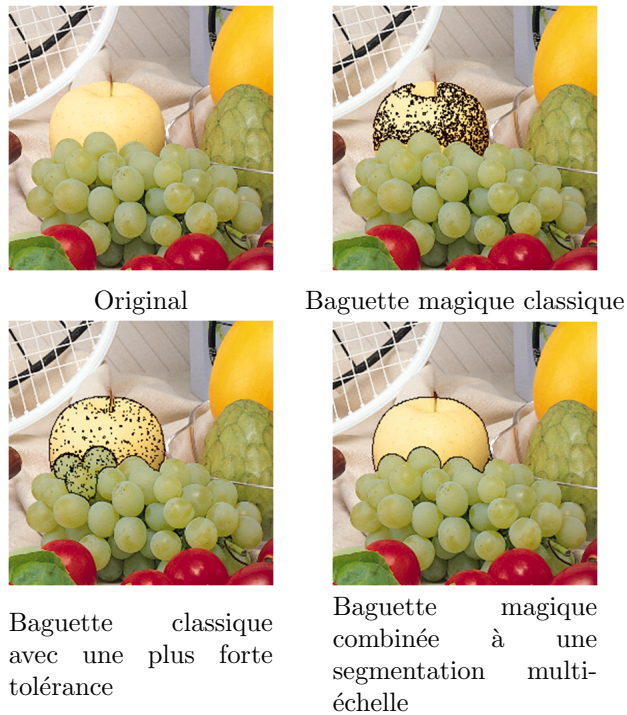


FIGURE 1.5 – Comparaison entre la baguette magique classique et celle combinée à une segmentation multi-échelle.

- [22] F. Zanoguera, **B. Marcotegui**, F. Meyer. "A Toolbox for Interactive Segmentation Based on Nested Partitions" *ICIP-99*, Kobe (Japan), October 1999.
- [23] **B. Marcotegui**, P. Correia, F. Marqués, R. Mech, R. Rosa, M. Wollborn, F. Zanoguera. "A Video Object Generator Tool Allowing Friendly User Interaction" *ICIP-99*, Kobe (Japan), October 1999.
- [25] F. Zanoguera, **B. Marcotegui** and F. Meyer. A segmentation pyramid for the interactive segmentation of 3-D images and video sequences, *n proceedings of the International Symposium of Mathematical Morphology*, Palo Alto, California, USA, 2000, pages 263-272. Kluwer Academic Publishers, 2000.
- [28] F. Marqués, **B. Marcotegui**, F. Zanoguera, P. Correia, R. Mech, M. Wollborn. Partition-based image representation as basis for user-assisted segmentation, in : Image Processing, 2000. Proceedings. 2000 International Conference on, Vol. 1, IEEE, 2000, pp. 312-315.
- [27] **B. Marcotegui**, F. Zanoguera. Image editing tools based on multi-scale segmentation. In H. Talbot and R. Beare (eds.), *Mathematical Morphology VI*, (Proc. ISMM02, Sydney, Australia, April 2002). CSIRO, pp. 127-135.

### 1.2.3 Algorithme de cascades et APM

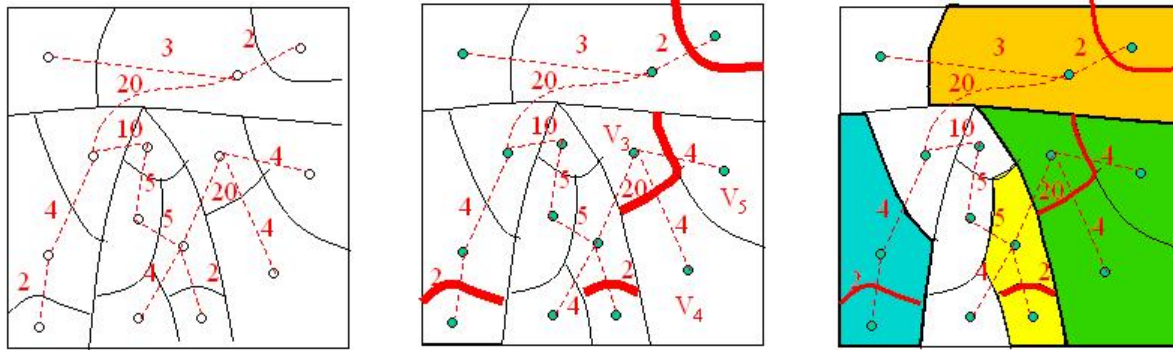
Etant donné que l'APM contient toute l'information nécessaire pour implémenter de manière efficace les processus associés à l'inondation, j'ai proposé une implementation rapide de l'algorithme de cascades, algorithme de segmentation hiérarchique introduit en 1994 par S. Beucher [29]. Le point de départ est une partition fine. Celle-ci peut être le résultat d'une ligne de partage des eaux, algorithme connu par sa fréquente sur-segmentation, ou toute autre partition initiale. Chaque contour de la partition fine (frontière commune entre deux régions) a une valuation, par exemple la valeur minimale du gradient sur la frontière. Ces contours sont alors classés selon leur importance, en fonction des valeurs des contours environnants. A chaque itération les contours entourés par des contours dont le gradient est plus élevé sont supprimés, établissant ainsi une hiérarchie entre eux. L'algorithme de cascades peut alors être interprété comme une ligne de partage des eaux mais appliquée sur les contours d'une partition au lieu de l'appliquer à l'ensemble des pixels de l'image. Cette procédure est itérée jusqu'à la suppression de tous les contours.

L'algorithme proposé dans [29], nécessite pour chaque itération une reconstruction d'image suivie d'une inondation. Dans [30], nous proposons un algorithme plus rapide, utilisant l'APM. Chaque région est alors traitée comme une entité au lieu de chaque pixel séparément, d'où le gain en efficacité. La procédure sur l'APM comporte deux étapes :

- Recherche des arêtes minimales. Chaque arête minimale identifie une région du niveau hiérarchique suivant ;
- Un processus d'inondation de l'arbre donnera la zone d'influence de chaque minimum.

La figure 1.6 illustre une étape de l'algorithme sur l'APM. La partition initiale et son APM associé sont sur la figure 1.6(a). Deux arêtes de l'arbre sont dites voisines, si elles ont un nœud en commun. Les arêtes minimales sont celles dont la valuation est inférieure à la valuation des arêtes voisines. Elles sont marquées en gras sur la figure 1.6(b). Dans cet exemple il y a 4 arêtes minimales. Chacune de ces arêtes devient un marqueur d'une région dans la partition résultante. Un label différent est ainsi assigné à chacune d'elles. Le label est associé aux nœuds reliant chaque arête, le même label de part et d'autre de l'arête minimale, montrant que cette frontière doit être supprimée de la partition. Un sous-ensemble des régions de la partition initiale reçoit ainsi un label sur la figure 1.6(c). Ces labels sont étendus à l'ensemble des régions suivant un processus d'inondation sur graphe (comme celui illustré sur la figure 1.3), donnant lieu à la partition finale sur la figure 1.6(d).

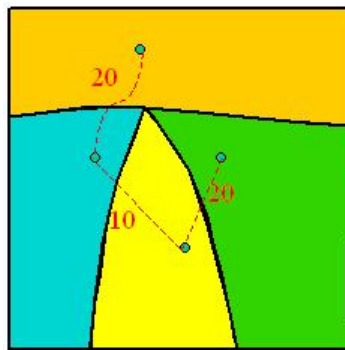
Cet algorithme a été développé dans le cadre d'un projet industriel avec Thales, dont l'objectif était de trouver les bords de routes non structurées pour le pilotage automatique de drones terrestres. La détection comprenait deux étapes : 1) initialisation, pour localiser les bords de route sur la première image ou après une erreur de détection ; 2) le suivi temporel, à partir de la détection sur l'image précédente. Les schémas



(a) Partition initiale et APM associé

(b) Arêtes minimales

(c) Labellisation d'arêtes minimales



(d) Propagation de labels à partir de (c)

FIGURE 1.6 – Une étape de cascade sur l'APM : (a) hiérarchie de niveau  $N$  et APM associé ; (b) arêtes minimales en gras ; (c) labellisation d'arêtes minimales et noeuds associés, en blanc les régions sans label associé à cette étape ; (d) propagation des labels à partir de (c) selon l'ordre marqué par les valuations des arêtes jusqu'à l'obtention d'une partition.

de ces deux étapes sont montrés sur la figure 1.7 et les images intermédiaires sur les figures 1.8, 1.9 et 1.10. Le temps de calcul était un paramètre critique de l'algorithme car le guidage du drone devait, bien entendu, être réalisé en temps réel. L'objectif a été atteint avec la puissance de calcul d'un processeur *Pentium 4*, cadencé à  $1,8\text{ GHz}$ , avec  $512\text{ Mo}$  de mémoire, tout à fait en phase avec la période de développement du projet.

#### Publications associées : **ConfI**{ [31], [30].}

- [31] V. Marion, O. Lecointe, C. Lewandowski, J.G. Morillon, R. Aufrere, **B. Marcotegui**, R. Chapuis, S. Beucher. Robust perception algorithm for road and track autonomous following. *Unmanned ground vehicle technology, Conference No 6*, vol. 5422, pp. 55-66, Orlando FL, USA, 2004.
- [30] **B. Marcotegui**, S. Beucher. Fast implementation of waterfall based on graphs. in *Mathematical Morphology : 40 Years on : Proc. 7th ISMM*. Paris, April 2005. Ronse C., Najman L., and Decencière E., Eds. Springer : p. 177-186.



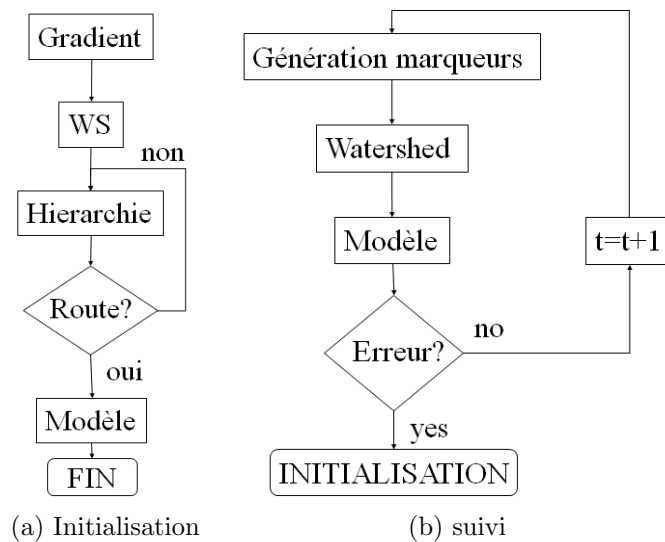


FIGURE 1.7 – Détection de bords de route : schémas de la phase d'initialisation et du suivi.

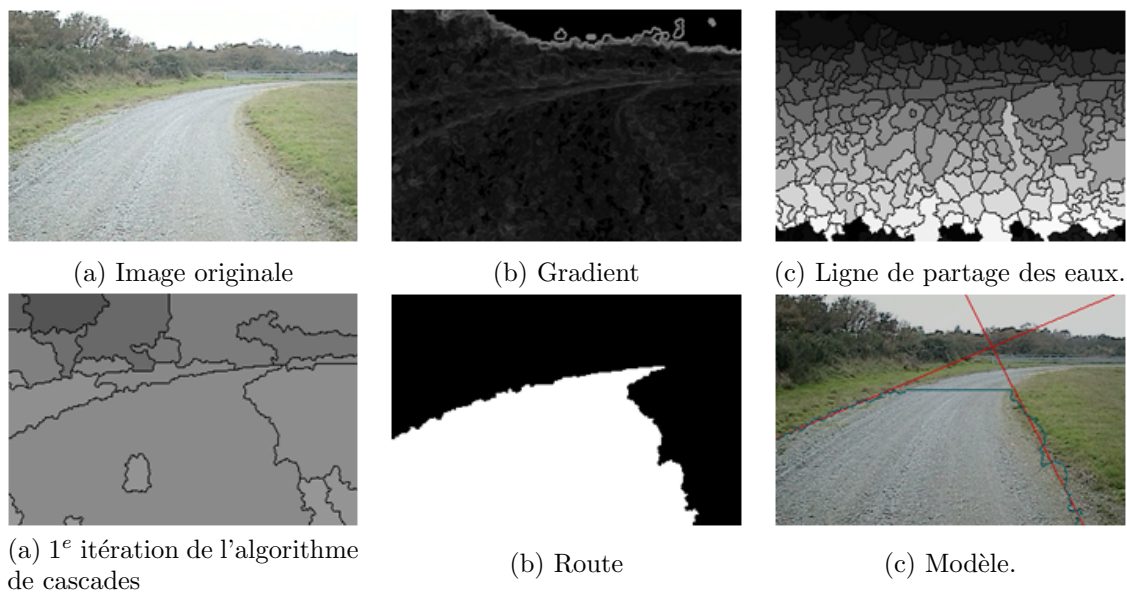


FIGURE 1.8 – Détection de bords de route : phase d'initialisation.

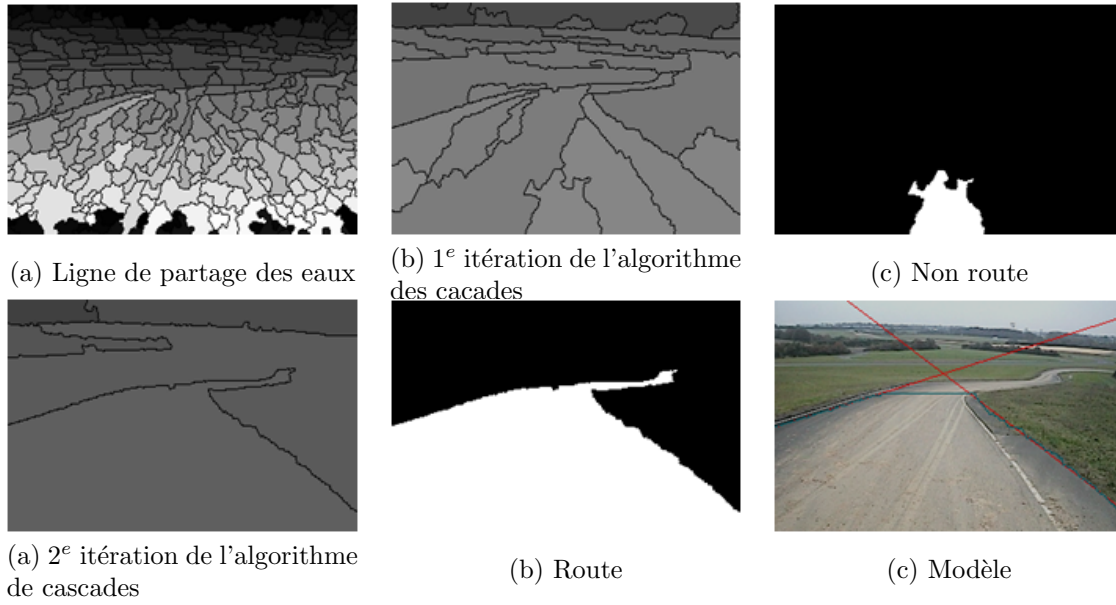


FIGURE 1.9 – Détection de bords de route : phase d'initialisation.

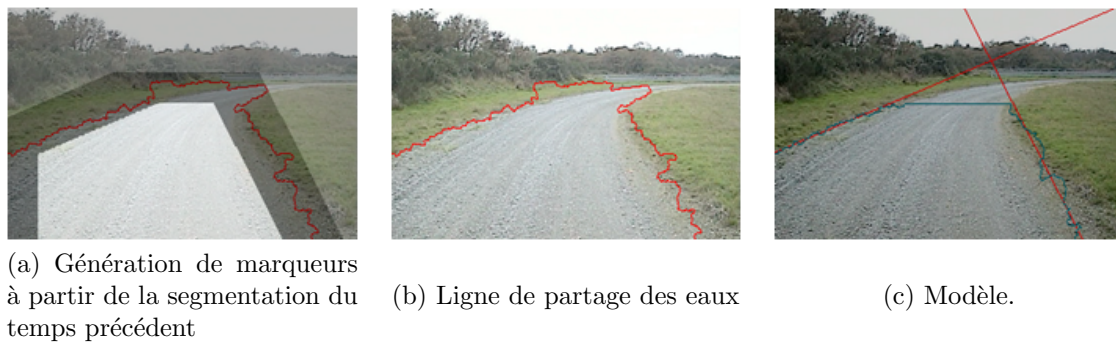


FIGURE 1.10 – Détection de bords de route : phase de suivi.

## 1.3 Ouverture ultime par attribut

### 1.3.1 Définition

Le top-hat associé à une ouverture est un opérateur très utilisé pour extraire des objets d'intérêt [32]. Il nécessite néanmoins d'une information a priori sur la taille des objets recherchés. L'ouverture ultime [33, 34] peut être interprétée comme une généralisation de cet opérateur. En effet, il s'agit d'un résidu d'ouvertures dont la taille s'adapte localement au contenu de l'image.

L'ouverture ultime,  $\theta$ , s'exprime de la manière suivante :

$$\begin{aligned} R_\theta(I) &= \max_\lambda(r_\lambda(I)) = \max_\lambda(\gamma_\lambda(I) - \gamma_{\lambda+1}(I)) \\ q_\theta(I) &= \begin{cases} \max\{\lambda + 1 \mid r_\lambda(I) = R_\theta(I)\} & \text{if } R_\theta(I) > 0 \\ 0 & \text{if } R_\theta(I) = 0 \end{cases} \end{aligned} \quad (1.1)$$

Cet opérateur analyse l'évolution de chaque pixel avec une série d'ouvertures de taille de plus en plus grande ( $\gamma_\lambda$ , avec  $\lambda = \{0, 1, \dots, N - 1\}$ ). La série de différences entre deux ouvertures successives  $r_\lambda = \gamma_\lambda - \gamma_{\lambda+1}$  est considérée et deux informations en sont extraites pour chaque pixel : 1.-  $R_\theta(I)$ , le résidu maximal et 2.-  $q_\theta(I)$  la taille de l'ouverture produisant le plus grand résidu.  $R_\theta(I)$  véhicule une information de contraste alors que  $q_\theta(I)$  indique la taille de la structure la plus contrastée contenant le pixel. Il est à noter que ce maximum peut ne pas être unique. Dans ce cas, Beucher propose de garder la valeur de  $\lambda$  la plus grande pour laquelle le résidu est maximal [33, 34].

Trois versions de cet opérateur ont été proposées dans la littérature par des chercheurs travaillant séparément, sur des domaines différents :

- en 1994, Vogt, publie l'ouverture adaptative [35] appliquée à l'analyse de documents ;
- en 2000 Pesaresi et Benediktsson, publient les profils de dérivées morphologiques [36] [37], appliqués à la télédétection ;
- en 2005 Beucher, publie l'ouverture ultime [33, 34] appliquée à la granulométrie des roches.

Les trois versions s'appuient sur la même idée de fond : les structures contrastées sont détectées grâce au résidu important qu'elles produisent quand elles sont filtrées par une ouverture de la taille adaptée.

J'ai proposé, dans le cadre de la thèse de Thomas Retornaz, l'utilisation d'une ouverture par attribut (*UAO*) dans le cadre résiduel décrit ci-dessus, donnant lieu à l'ouverture ultime par attribut, *UAO*. L'*UAO* offre les avantages suivants :

- comme l'ouvert ultime, il s'agit d'un opérateur intrinsèquement multi-échelle, capable d'extraire des objets d'intérêt sans information a priori sur leur taille.
- les ouvertures par attribut étant des opérateurs connexes, l'*UAO* en est un aussi.
- nous avons proposé une implémentation efficace. Nous démontrons dans [38] que l'*UAO* peut être calculée en une simple passe de l'arbre des maxima [39].

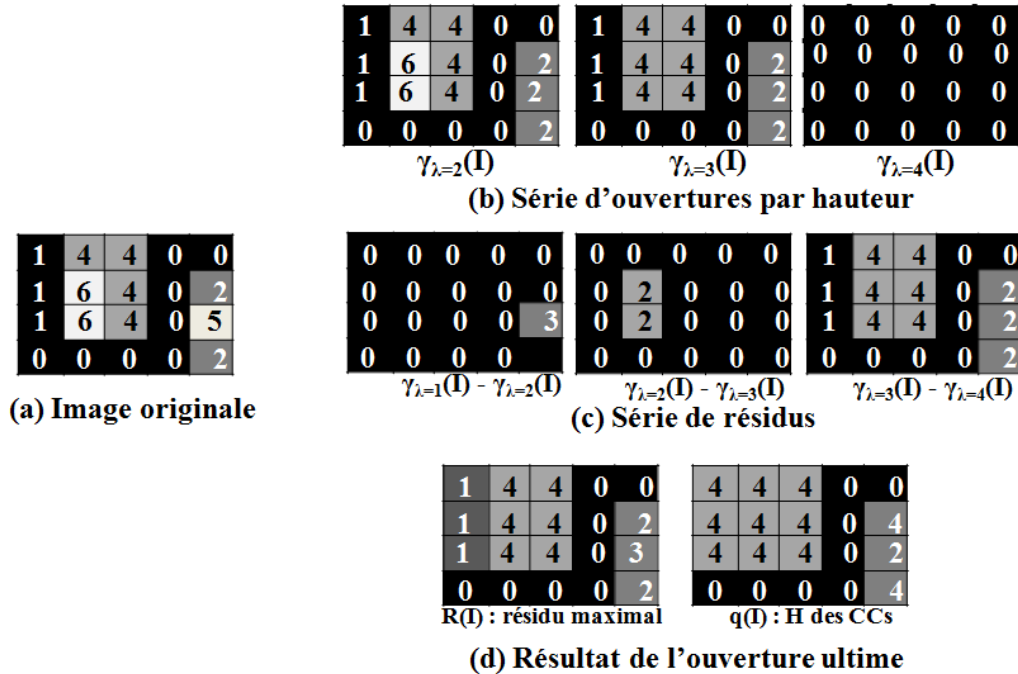


FIGURE 1.11 – Opérateur  $v^\gamma$ . (a) image originale. (b) série d'ouvertures par hauteur. (c) série de résidus. (d) résultat de l'ouverture ultime par hauteur :  $R_\theta$  et  $q_\theta$ .

La figure 1.11 illustre les étapes intermédiaires du calcul de l'ouverture ultime par attribut. L'attribut utilisé est  $H$ , la hauteur de la composante connexe. L'image 1.11(a) montre l'image originale. Les différentes ouvertures par hauteur, sont illustrées sur la figure 1.11(b). On observe la suppression des structures de plus en plus grandes. Sur la figure 1.11(c) nous pouvons observer les résidus successifs et sur 1.11(d) les images résultantes de l'opérateur : le résidu maximal pour chaque pixel et la taille associée. La labellisation de  $q_\theta(I)$  donne lieu à une partition partielle [40, 41] : les pixels sont regroupés en segments qui ne s'intersectent pas et dont l'union ne recouvre pas tout le domaine de l'image ( $q_\theta(I)$  est égale à 0 pour les pixels où tous les résidus sont nuls).

L'ouverture ultime segmente des objets clairs sur fond sombre. Une fermeture ultime est nécessaire pour extraire des objets sombres sur fond clair. L'ouverture ultime peut aussi être appliquée au gradient, traitant ainsi les deux polarités en même temps. L'utilisation du gradient est néanmoins déconseillée pour les images contenant des objets fins.

### 1.3.2 Application à l'extraction de texte

L'UAO s'est révélée particulièrement utile pour l'extraction de texte sur des images naturelles [42]. Sur la figure 1.12, nous pouvons observer le résultat de la fermeture

ultime par hauteur ( $UHC$ ) appliquée à une image réelle 1.12(a). Les lettres sur l’affiche ne sont pas homogènes, elles sont remplies d’une texture venant d’une autre image. La figure 1.12(b) illustre le résultat  $q_\theta(I)$  et la figure 1.12(c)  $R_\theta(I)$ . Malgré le fait que les lettres sont texturées, la plupart d’entre elles est correctement segmentée. Ceci est dû grâce au fait que la texture à l’intérieur des lettres est moins contrastée que les lettres avec le fond.

Pour bien comprendre la procédure, analysons l’évolution de la série de fermetures (les caractères sont plus sombres que leur fond) de deux pixels à l’intérieur de la même lettre. Deux pixels sont choisis aléatoirement à l’intérieur de la lettre “N” de “GENERATION”. La série de valeurs de ces deux pixels après leur avoir appliqué des fermetures de taille croissante est représentée en ligne discontinue bleue sur les figures 1.13(a) et 1.13(b). La courbe des résidus (dérivée de la courbe bleue) est superposée en vert. Les résidus produits par des fermetures de petite taille correspondent aux fluctuations internes de la lettre. Ces résidus sont différents pour chaque pixel. En revanche, ces deux pixels ont le résidu maximal pour une fermeture de taille 83. La raison de ceci est que la lettre a une hauteur de 82. Elle est donc filtrée par une fermeture de taille 83. Ceci peut être observé sur la figure 1.13(c), fermeture de taille 82, et la figure 1.13(d), fermeture de taille 83, avec le N filtré. De ce fait, le même résidu important est vu par tous les pixels de la structure. Ainsi, la fermeture ultime capture la présence d’objets contrastés lors de leur suppression par une fermeture d’une taille adaptée, menant à une segmentation intéressante sans information a priori sur la taille des objets d’intérêt.

Nous avons développé un système de localisation de texte enfoui basé sur l’ouverture ultime par attribut, qui a remporté la première position dans le cadre du projet ImagEVAL 2006 [42], du programme TechnoVision. Le but d’ImagEVAL était d’évaluer les technologies de filtrage et d’indexation d’images, de recherche d’images par le contenu dans des bases de données constituées par les détenteurs de fonds professionnels. Ce travail a fait l’objet d’un contrat industriel avec EADS, dont le but était d’extraire de mots clés des images sur internet.

L’ouverture ultime produit des résultats intéressants dans une grande variété de situations. Elle a été utilisée dans des applications comme la granulométrie de roches [43], la localisation de texte enfoui [44], l’analyse de façades [45], l’analyse d’images de fond de l’oeil [46]. Les performances de l’ouverture ultime sont testées sur la base Weizmann dans la section 1.3.3. Par ailleurs, nous avons identifié quelques situations problématiques. Deux configurations posent principalement des problèmes : 1) les transitions graduelles liées par exemple au flou de l’image, étudiées dans la section 1.4, et 2) les structures imbriquées que nous analysons dans la section 1.5. Nous proposons des solutions à ces problèmes dans les sections respectives.

**{Publications associées : ConfI{[47].}, ACLI{[48] } }**

- [47] A. Serna and B. Marcotegui. ”Attribute controlled reconstruction and adaptive mathematical morphology.”, Proceedings of 11th International Sym-



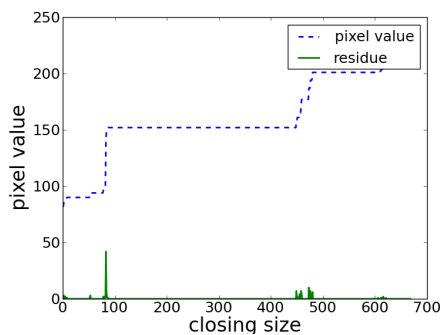
(a) Image originale.]

(b)  $q_\theta(I)$  (information de taille).(c)  $R_\theta(I)$  (information de contraste).

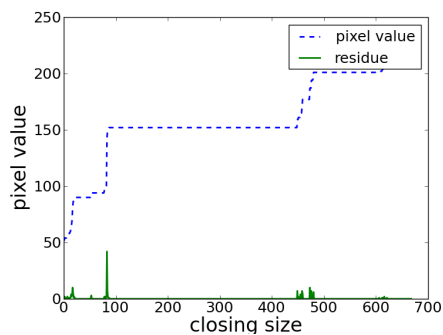
FIGURE 1.12 – Fermeture ultime par hauteur (*UHC*) : (b)  $q_\theta(I)$ , en fausses couleurs pour observer les régions segmentées et (c)  $R_\theta(I)$ .

posium on Mathematical Morphology, May 27-31 2013, Uppsala, Sweden. pp 207-218

- [48], A. Serna, B. Marcotegui, E. Decencière, Th. Baldewck, A.M. Pena, S. Brizion. "Segmentation of elongated objects using attribute profiles and area stability : application to melanocyte segmentation in engineered skin". Pattern Recognition Letters, Volume 47, 1 October 2014, Pages 172-182.



(a) Evolution du niveau de gris (en ligne bleue discontinue) et les résidus associés (en vert) d'un pixel à l'intérieur de la lettre "N".



(b) Evolution du niveau de gris (en ligne bleue discontinue) et les résidus associés (en vert) d'un autre pixel à l'intérieur de la lettre "N".



(d) fermeture de taille  $H = 82$ .



(d) Fermeture de taille  $H = 83$ .

FIGURE 1.13 – Images intermédiaires de la fermeture ultime par hauteur. La fermeture de taille 83 supprime le "N" (entouré d'une ellipse), produisant un résidu important sur l'ensemble de pixels de cette lettre.

### 1.3.3 Evaluation sur la base Weizmann

#### 1.3.3.1 Description de la base et protocole d'évaluation

La segmentation automatique d'image est un problème mal posé. Son évaluation est par conséquent problématique. Une méthode peut produire des résultats excellents sur une base de données mais échouer dans d'autres conditions : régions texturées, conditions d'éclairage difficiles, contours flous... Par ailleurs, la notion de "segmentation correcte" sur la même image peut varier en fonction de l'application. Par exemple, face à une scène urbaine on peut s'intéresser à compter les voitures, lire les plaques d'immatriculation, reconnaître les conducteurs ou modéliser les bâtiments derrière la scène. Sans une spécification claire de ce qui doit être segmenté et le niveau de détail souhaité (façade en entier ou un segment pour chaque fenêtre) différentes personnes produiront des partitions très différentes. Quelques tentatives d'évaluation de méthodes de segmentation d'images génériques existent dans la littérature. La plus connue est la base de Berkeley [49] (<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>). 300 images ont été choisies et annotées par plusieurs personnes. Les contours reposent sur des critères sémantiques et ne correspondent pas toujours à des contours contrastés. Par exemple, la frontière entre deux chevaux de la même couleur n'est pas visible sur l'image 1.14. En revanche tous les annotateurs ont marqué la frontière, se basant sur la connaissance a priori de la forme d'un cheval.

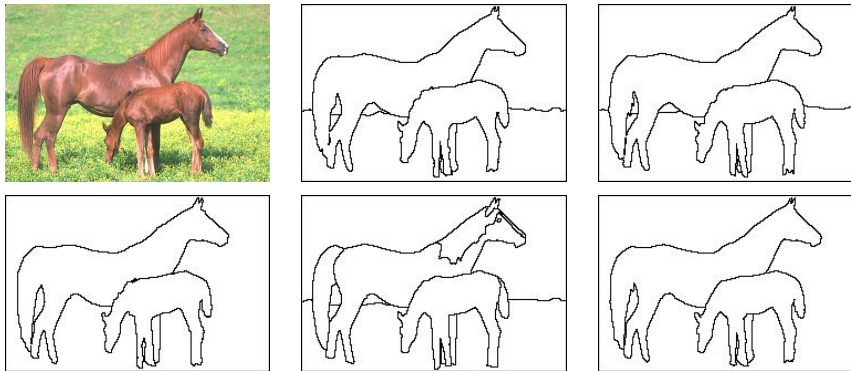


FIGURE 1.14 – Annotation sémantique.

Une base d'images alternative a été proposée par l'Institut Weizmann [50]. Elle contient une variété d'images avec des objets qui diffèrent de leur environs soit en intensité, en texture ou une caractéristique bas niveau. Pour éviter des ambiguïtés potentielles, seulement des images avec un seul objet ont été sélectionnées. Pour obtenir la vérité terrain (segmentation de référence), 50 personnes ont segmenté des images en deux classes : fond et forme. Chaque image a été segmentée par trois personnes différentes. Finalement, le pixel est déclaré comme forme si il a été marqué comme tel par au moins deux des trois personnes. Cette base contient aussi quelques images



discutables, avec des vérités terrain reposant sur des notions sémantiques. Voir par exemple la figure 1.15 et sa vérité terrain. Il n’y a pas de critère bas niveau identifiant l’objet d’intérêt mais une connaissance a priori de la forme d’un pingouin.

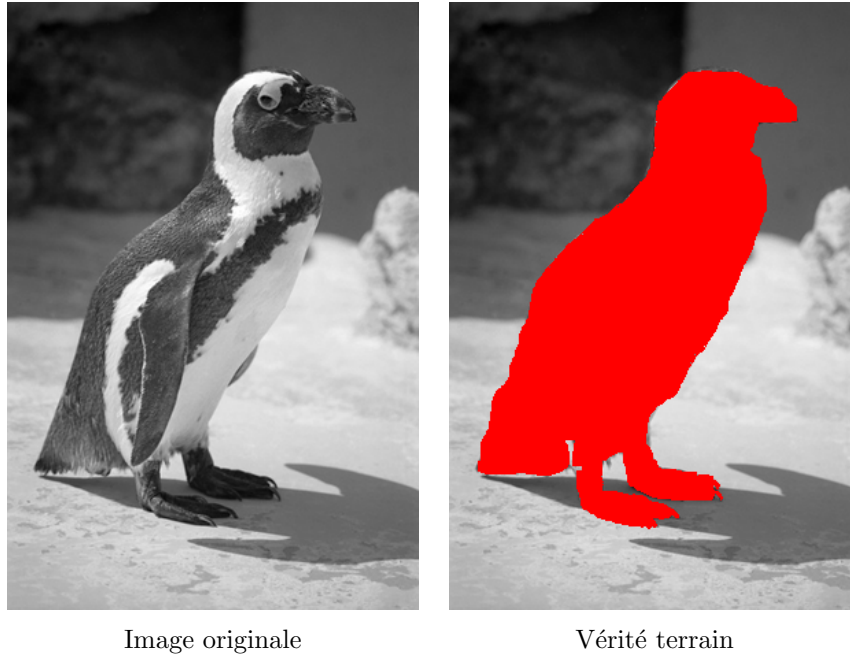


FIGURE 1.15 – Exemple de vérité terrain reposant sur des critères sémantiques.

Les performances de l’ouverture ultime sont testées sur la base Weizmann. Les résultats quantitatifs utilisent les mesures classiques de *precision*, *recall* et  $f_{mean}$  selon les formules suivantes :

$$Precision = \frac{N_{OK}}{N_{detected}} \quad (1.2)$$

$$Recall = \frac{N_{OK}}{N_{GT}} \quad (1.3)$$

$$f_{mean} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1.4)$$

avec  $N_{detected}$  étant le nombre de pixels de l’objet détecté,  $N_{OK}$  le nombre de pixels détectés qui appartiennent à l’objet d’intérêt et  $N_{GT}$  le nombre de pixels de l’objet de référence. Ainsi, la *precision* est élevée si l’objet détecté ne dépasse pas la frontière de l’objet de référence, alors que le *recall* est élevé si la plupart de l’objet d’intérêt est couvert par l’objet détecté.  $f_{mean}$  est la moyenne harmonique de la *precision* et du *recall*, utilisé pour classer les performances de différents systèmes.  $f_{mean}$  pénalise les

systèmes qui ont des valeurs de *precision* et de *recall* non balancées. Les méthodes de segmentation produisent des partitions avec plusieurs régions alors que l'image est supposée contenir un seul objet. Dans ce cas, tel que proposé par les créateurs de la base d'images [50], le  $f\_mean$  est calculé pour chaque région et la région maximisant cette mesure est choisie. Si l'algorithme sur-segmente, la *precision* sera élevée mais le *recall* sera faible et à l'inverse si l'algorithme sous-segmente, le *recall* sera élevé mais la *precision* sera faible.

### 1.3.3.2 Résultats

La table suivante 1.1 résume les performances de l'ouverture ultime par attribut avec différents attributs, appliqué à l'ensemble d'images de la base Weizmann :

	P	R	F_mean
UO H	79.36	56.52	57.18
UO W	86.77	46.97	55.97
UO S	87.18	39.09	49.03
UO A	73.18	46.40	46.08

TABLE 1.1 – Performance de l'ouverture ultime par attribut, avec différents attributs, sur la base Weizmann. Les attributs sont les suivants : H=hauteur, W=largeur, S=diamètre= $\max(H,W)$ , A=area. P=*precision*, R=*recall*.

L'attribut de hauteur donne le meilleur résultat avec un  $f\_mean=0.57$ , le même score rapporté dans [50] avec la populaire méthode de mean-shift [51]. Nous avons essayé de reproduire ce résultat avec le logiciel cité sur le papier [52] (EDISON implémentation disponible sur [www.caip.rutgers.edu](http://www.caip.rutgers.edu)). Nous avons obtenu un score de  $f\_mean=0.47$ , ce qui veut dire que les paramètres ont été optimisés pour la base. Ces paramètres ne sont pas donnés dans l'article.

L'ouverture ultime avec des attributs de hauteur et de largeur produit des résultats similaires ( $f\_mean = 0.56$  pour la largeur et  $0.57$  pour la hauteur) sans besoin de régler des paramètres. Les figures 1.16 et 1.17 illustrent des résultats qualitatifs avec l'*UHO* sur quelques images. Des bons résultats sont obtenus si l'objet d'intérêt est contrasté par rapport à son fond. Des exemples de ces situations sont l'arbre illustré sur les figures 1.16(a-b), le chat sur les figures 1.16(o-p) ou le papillon sur les figures 1.16(m-n). Ces objets sont bien segmentés et ont un  $f\_mean$  supérieur à 0.9 malgré leur texture interne. Ceci est vrai grâce au fait que le contraste entre l'objet et son fond est plus important que le contraste des motifs de texture interne. Ceci n'est pas le cas de l'étoile de mer des figures 1.17(g-h). C'est pourquoi les motifs de cette texture sont segmentés par l'*UHO*, donnant lieu à une sur-segmentation et un  $f\_mean$  inférieur à 0.2. Des résultats médiocres sont obtenus pour des objets composés de plusieurs parties, comme l'exemple de la figure 1.17(a-b). Un autre exemple de résultat avec un  $f\_mean$  faible est montré sur les figures 1.17(e-f). Dans ce cas, l'enveloppe de la couronne de

l'arbre a été annotée alors que l'UHO a segmenté les détails des branches. Par ailleurs l'UHO a fusionné le tronc de l'arbre avec le sol, ce qui est compréhensible du point de vue du contraste entre eux. C'est un autre exemple de vérité terrain reposant non seulement sur des indices bas niveaux mais aussi sur des critères sémantiques.

Nous observons une relative bonne *precision* (0.70 - 0.80) par rapport au *recall* (0.40 - 0.50), ce qui veut dire qu'en général les images sont sur-segmentées : la région qui mieux représente la vérité terrain est globalement incluse dans l'objet d'intérêt mais elle couvre moins de la moitié de la surface de celui-ci. Selon cette remarque, une étape de prétraitement, filtrant les détails contenant une texture fine ou du bruit permettrait d'améliorer le résultat. A cet effet, nous utilisons les nivellements alternés [53, 54]. La figure 1.19 montre la *precision*, le *recall* et le  $f_{mean}$  pour différentes tailles de filtre. Etant donné que l'étape de filtrage agrandit les régions, le *recall* augmente avec la taille du filtre alors que la *precision* diminue avec celle-ci. Nous observons néanmoins que le score  $f_{mean}$ , utilisé généralement pour classer les performances de différents algorithmes, augmente jusqu'à atteindre un maximum pour de tailles de filtre entre 8 et 10. Cela veut dire que les régions grandissent majoritairement à l'intérieur des objets d'intérêt, augmentant davantage le *recall* que la perte en *precision*. La valeur maximale de  $f_{mean}$  est obtenue pour l'attribut W (largeur) et un filtre de taille 8. Les valeurs précises de ces performances sont : P= 0.8326, R= 0.6405 et  $f_{mean} = 0.678$ .

Ce résultat est très intéressant pour une méthode bas niveau, tenant compte seulement des informations de contraste et avec des temps de traitement particulièrement intéressants, comme nous verrons dans la section 1.6.

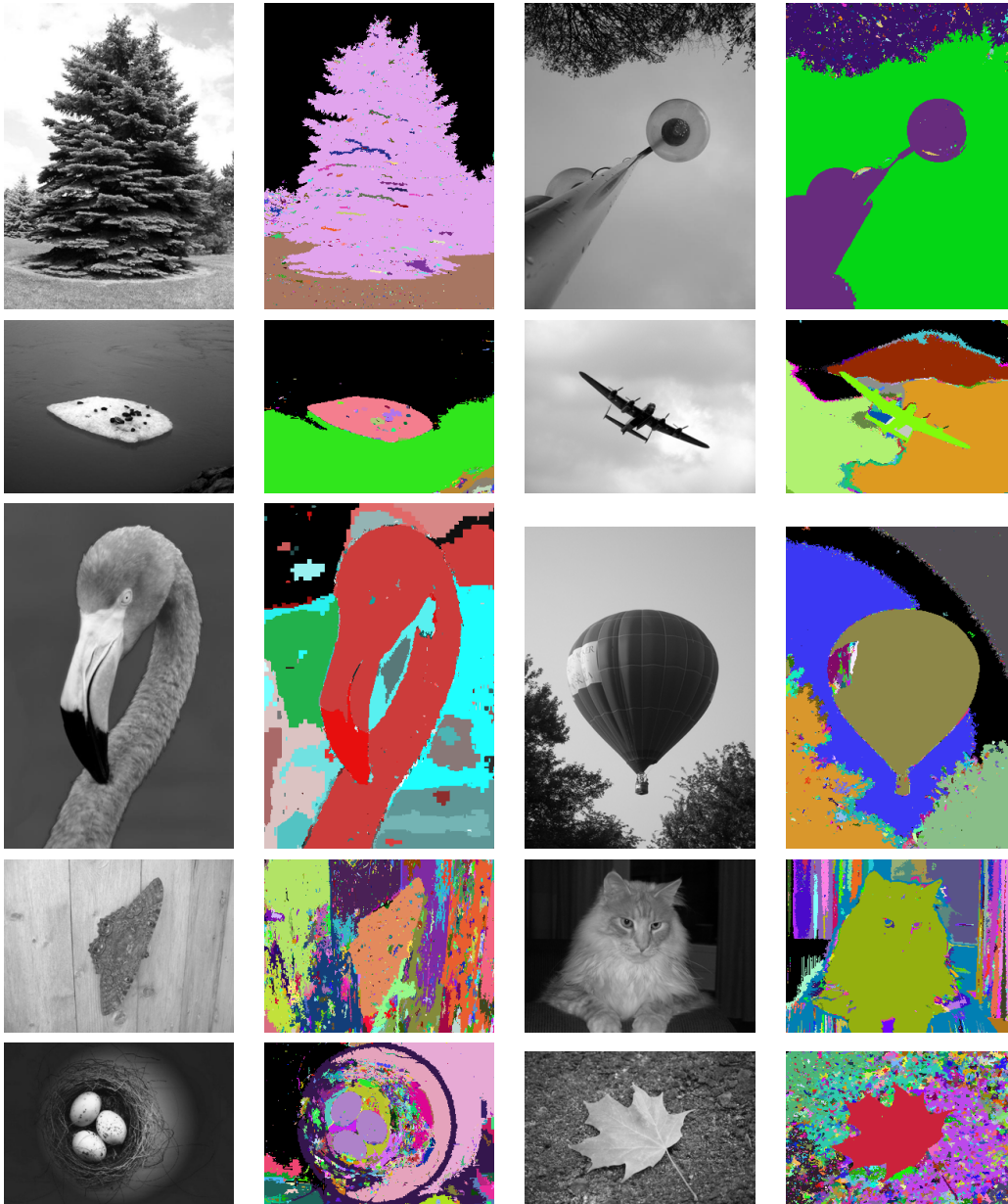


FIGURE 1.16 – Exemples d'utilisation de l'UHO avec un  $f\_mean$  supérieur à 0.9.

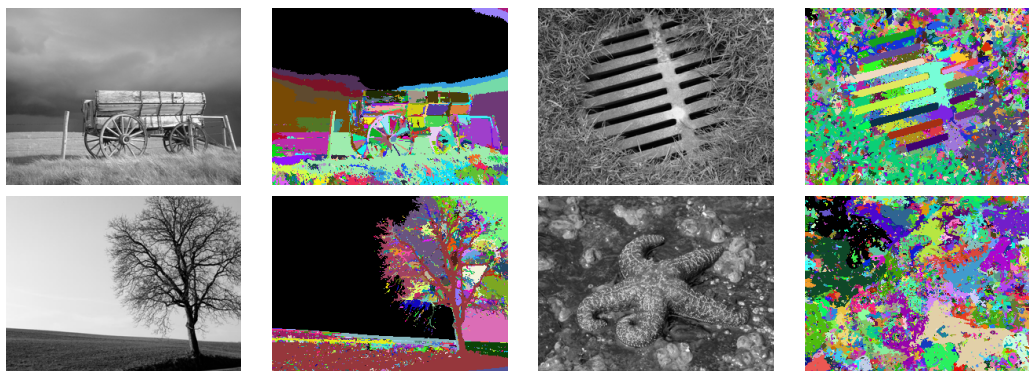


FIGURE 1.17 – Exemples d'utilisation de l'UHO avec un  $f_{mean}$  inférieur à 0.2.



FIGURE 1.18 – Vérité terrain de l'image dscn6805.png.

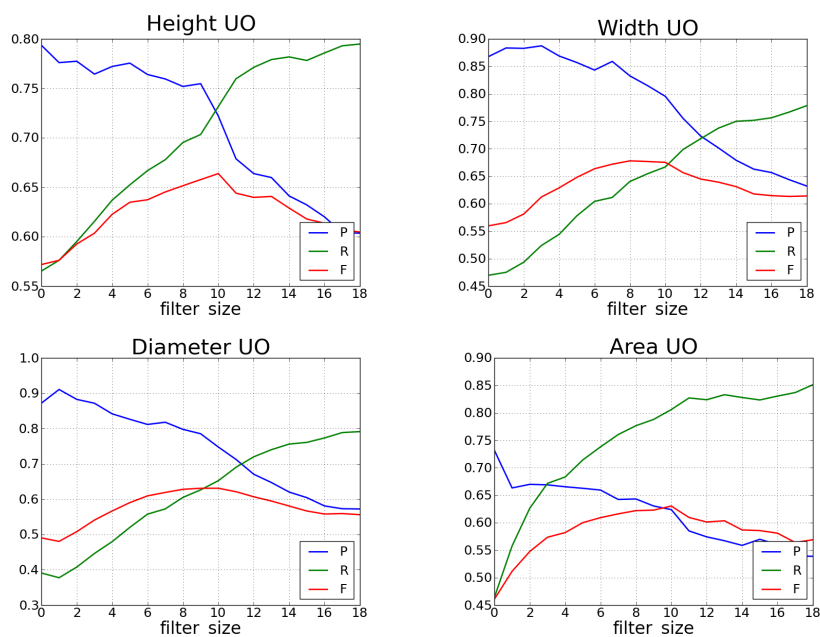


FIGURE 1.19 – Performances de l'ouverture ultime par attribut avec différents attributs sur la base Weizmann.

## 1.4 Transitions graduelles : ouverture ultime par $\Delta$ -attribut

Le changement de niveau de gris d'une frontière floue se fait de manière progressive, sur une zone de transition d'autant plus étendue que le phénomène est important. Ainsi, le changement de niveau de gris entre régions voisines se fait en plusieurs pas. Le contraste associé par l'ouverture ultime à une telle transition sera le plus grand de ces échelons, par définition de plus petite amplitude que le réel contraste entre les deux structures. Ainsi l'ouverture ultime sous-estime le contraste associé aux frontières floues.



FIGURE 1.20 – Sous-estimation du contraste pour une image floue.

Un exemple de cette situation est illustré sur la figure 1.20. Analysons sur un profil 1D le comportement de cet opérateur. L'image originale est sur la figure 1.21(a) et le profil horizontal de la ligne surimposée en blanc est sur la figure 1.21(b). Le niveau de gris des lettres est d'environ 160 alors que le fond a une valeur aux alentours de 10. Le contraste entre les lettres et leur fond est alors de 150. Cependant, l'ouverture ultime estime un contraste de seulement 20 (voir  $R_\theta$  sur la figure 1.21(d)). L'évolution de niveau de gris d'un pixel à l'intérieur d'une lettre avec une série d'ouvertures de taille de plus en plus grande est sur la figure 1.21(c), en ligne discontinue bleue. Les résidus associés sont illustrés en ligne verte continue. Nous pouvons observer une série de résidus non nuls pour des ouvertures de tailles consécutives. Ces résidus correspondent au filtrage des transitions graduelles d'une frontière floue. En fait, les zones de transition sont caractérisées par leur petite taille. C'est pourquoi elles donnent lieu à des résidus non nuls pour des tailles proches. Si on intègre les résidus non nuls consécutifs, supposant qu'ils correspondent aux régions de transition, nous obtenons une meilleure estimation du contraste. La ligne rouge sur la figure 1.21(c) montre l'intégration de résidus consécutifs non nuls et la figure 1.23 montre les pixels qui ont bénéficié de cette intégration. Nous constatons que ces pixels sont sur la frontière des régions floues.

Cette idée a été introduite par Thomas Retornaz et généralisée par Jorge Hernández avec l'opérateur d'ouverture ultime par  $\Delta$ -attribut ( $\Delta UAO$ ). L'ouverture ultime par  $\Delta$ -attribut intègre les résidus jusqu'à rencontrer une série de  $\Delta$  résidus nuls. Pour  $\Delta = 0$  on obtient l'ouverture ultime classique, pour  $\Delta = 1$  intègre les séries de résidus non

nuls,  $\Delta = 2$  intègre les séries de résidus séparés par au moins deux résidus consécutifs nuls, et ainsi de suite. Plus  $\Delta$  est grand, plus les zones considérées comme transition seront grandes.

La figure 1.22 compare les résultats d'ouverture ultime avec un attribut de hauteur appliqué à l'image 1.21(a) : les figures 1.22(a) et 1.22(b) montrent  $R_\theta$  et  $q_\theta$  avec  $\Delta = 0$  et les figures 1.22(c) et 1.22(d) montrent  $R_\theta^\Delta$  and  $q_\theta^\Delta$  pour  $\Delta = 1$ . On peut apprécier une bien meilleure estimation du contraste avec  $\Delta = 1$ . Les zones de transition peuvent être identifiées comme celles contribuant à l'intégration de résidus (voir figure 1.23).

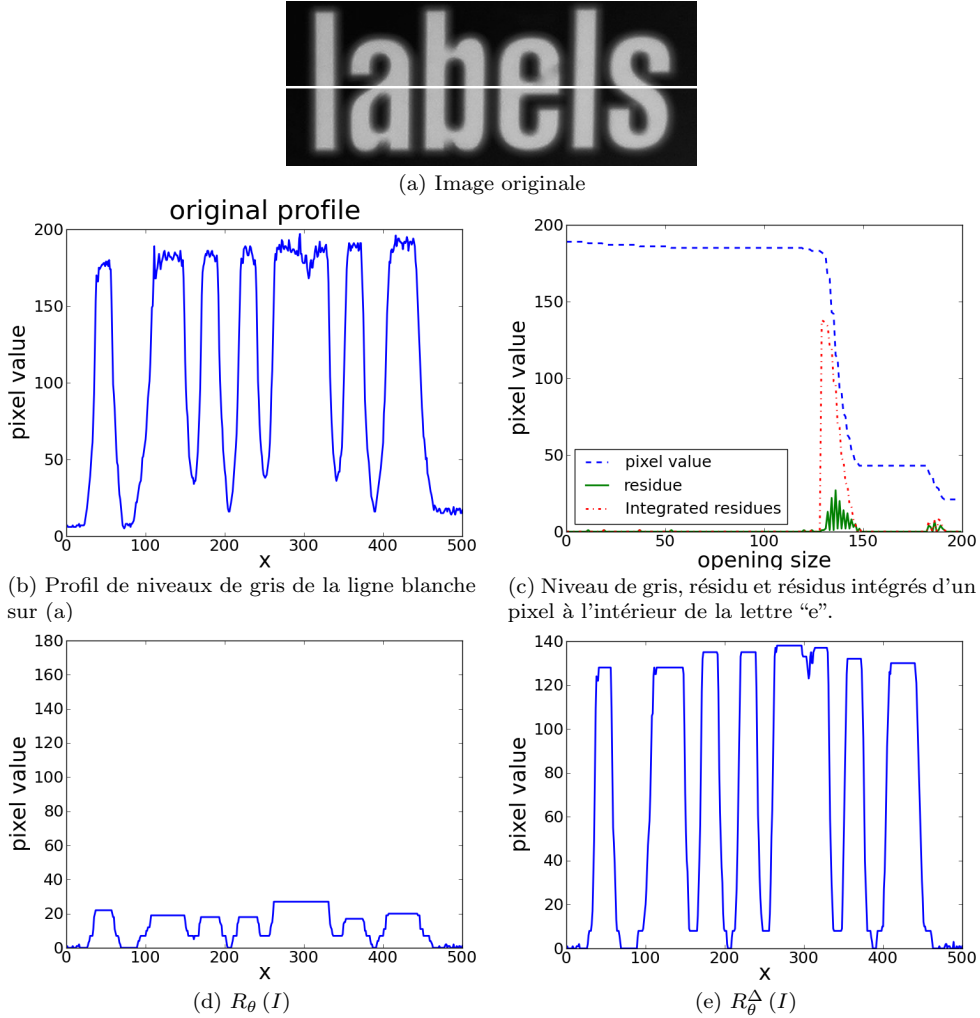


FIGURE 1.21 – Résidus d'une transition graduelle (objets flous).

La figure 1.24 illustre quelques exemples supplémentaires comparant  $UAO$  classique avec  $\Delta UAO$ . Pour chaque image, la première ligne montre l'image originale (à gauche) et le résidu associé à l'ouverture ultime classique. Nous constatons que le contraste est clairement sous-estimé. Sur la deuxième ligne,  $q_\theta^\Delta$  (à gauche), représenté en fausses

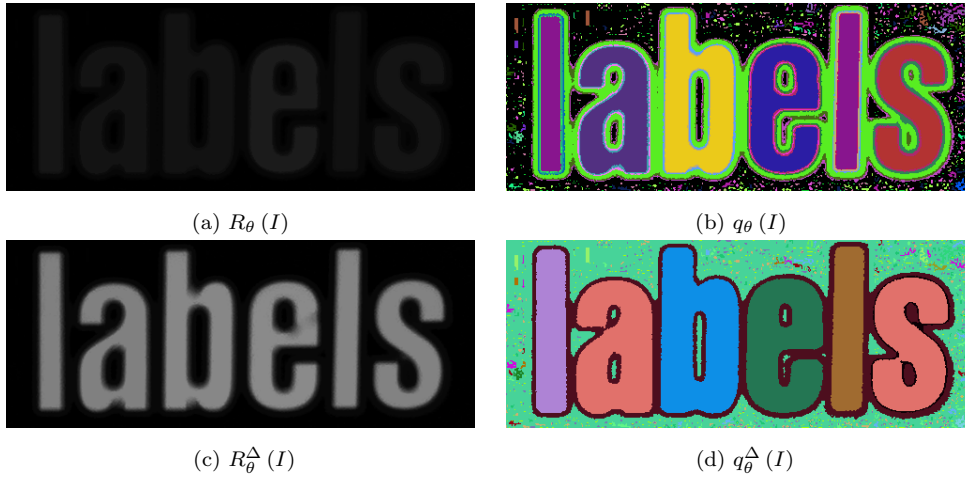
FIGURE 1.22 – Comparaison de  $UAO$  et  $\Delta UAO$ .

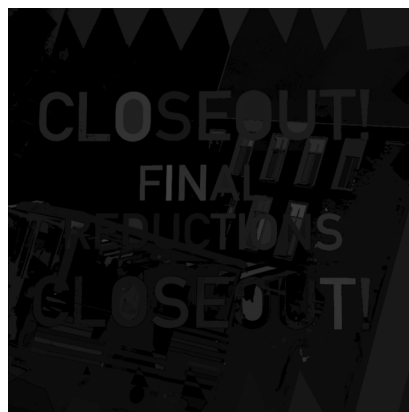
FIGURE 1.23 – Zones de transition détectées comme celles qui bénéficient de l'intégration de résidus.

couleurs pour distinguer les régions segmentées et leur information résiduelle associée (à droite). Nous constatons un résultat satisfaisant dans des situations difficiles : fond complexe, problèmes d'éclairage. La plupart de lettres ont été bien segmentées et leur contraste associé correspond au contraste perçu.





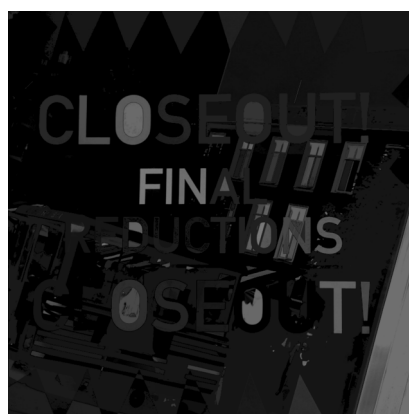
(a) Original image



(b)  $R_\theta$



(c)  $q_\theta^\Delta$



(d)  $R_\theta^\Delta$



(e) Original image



(f)  $R_\theta$



(g)  $q_\theta^\Delta$



(h)  $R_\theta^\Delta$

FIGURE 1.24 –  $\Delta UAO$  Résultats de  $\Delta UHO$ . Pour chaque image, première ligne : image originale et  $R_\theta$  ; deuxième file :  $q_\theta^\Delta$  et  $R_\theta^\Delta$ .

## 1.5 Structures imbriquées : ouverture ultime avec une information de forme

L'ouverture ultime sélectionne les résidus de plus grande amplitude et détecte ainsi les structures les plus contrastées d'une image. Toute structure incluse dans une autre de plus fort contraste est alors considérée comme du bruit et n'est donc pas présente sur le résultat de segmentation. La figure 1.25 montre un exemple de cette situation. L'image 1.25(a) contient trois formes imbriquées : un rectangle (de dimensions  $120 \times 40$ ,  $t = 250$ ), un carré (de dimensions  $30 \times 30$ ,  $t = 200$ ) et un cercle (de diamètre 90,  $t = 175$ ). L'image 1.25(b) montre cette image en relief. Ces structures sont sur un socle de niveau  $t = 130$ . Analysons pas à pas les différents résidus (figures 1.25(c) et 1.25(d)) obtenus avec des ouvertures par hauteur : la première forme filtrée est le carré (pour un attribut  $H = 31$ ) produisant un résidu de valeur 25. Ensuite, cette forme est masquée par le résidu de valeur 45 du cercle qui se produit pour une ouverture de taille  $H = 91$ . Une ouverture de taille  $H = 121$  produit un résidu de valeur 120 pour le rectangle. Jusqu'à cette étape deux formes sont alors sélectionnées : le cercle, avec un résidu maximal de  $175 - 130 = 45$  et le rectangle avec un résidu de  $250 - 130 = 120$ . L'ouverture de taille 161 produit un résidu de 130, et ceci sur une région incluant les deux formes précédentes. Ces deux formes sont alors masquées par la région qui les contient. Le résidu de l'ouverture  $H = 161$  produit un résidu plus fort que les précédents et masque une information significative.

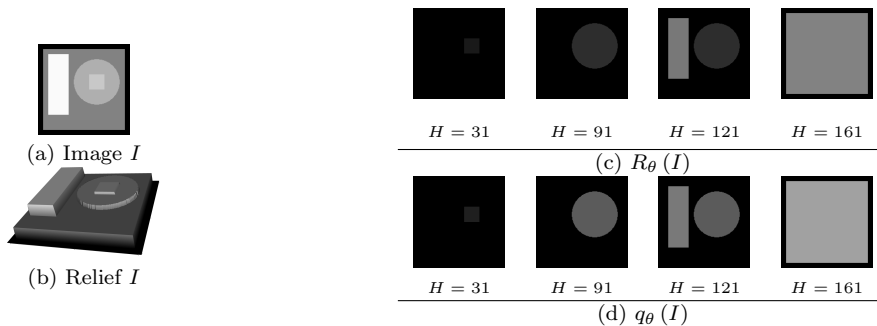


FIGURE 1.25 – Exemple de structures imbriquées : (a) Image de synthèse, (b) représentation de  $I$  sous forme de relief, (c) and (d) images intermédiaires de  $R_\theta(I)$  et de  $q_\theta(I)$ , respectivement.

La figure 1.26 illustre plusieurs exemples de l'opérateur  $UHO$  appliqué sur des images réelles. Sur la figure 1.26(a), la plupart des structures internes de la façade sont masquées. En effet, le contraste entre le mur et les fenêtres est inférieur au contraste entre la façade et le ciel. Un exemple de transition graduelle est illustré sur la figure 1.26(b). Nous observons que toutes les lettres sont masquées. Finalement, sur la figure 1.26(c) deux cellules visuellement bien séparées sont fusionnées à cause du flou présent sur la frontière des cellules. Une analyse détaillée de ces problèmes sera présentée dans la section suivante.

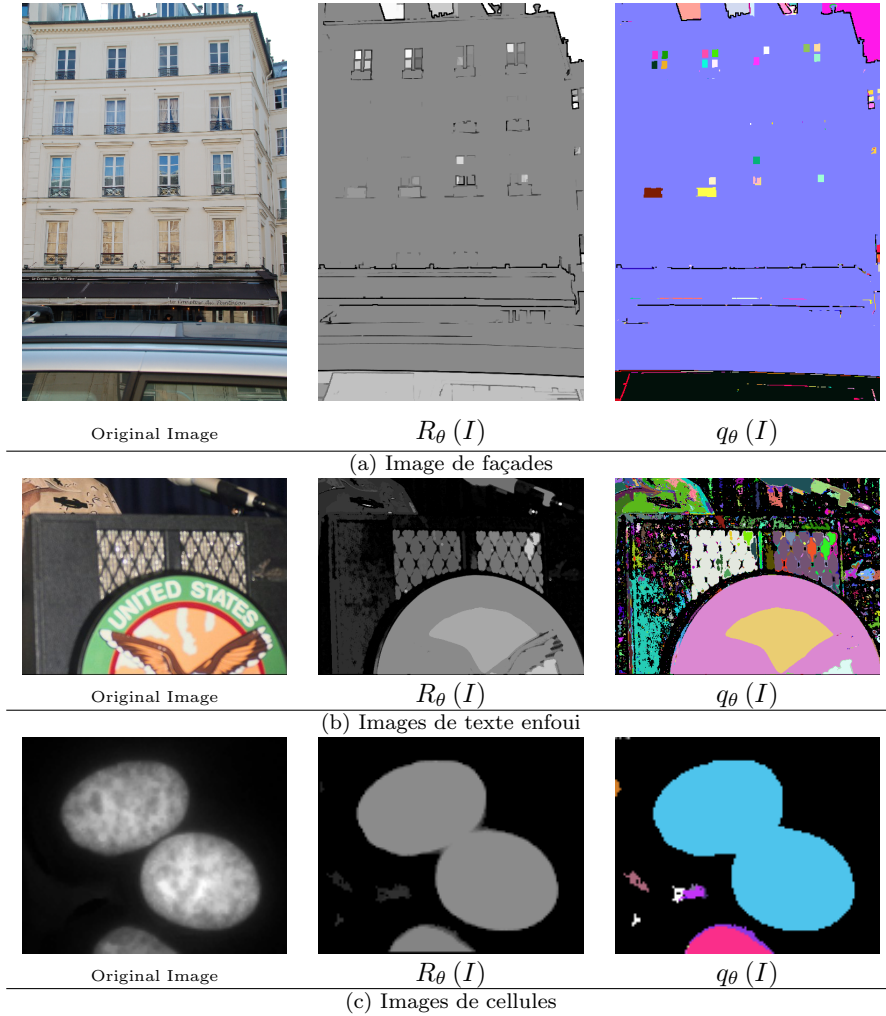


FIGURE 1.26 – Exemples de structures imbriquées sur des images réelles.

L'opérateur ( $SUAO$ ), introduit dans la thèse de Jorge Hernández, combine l'information de niveau de gris avec une information de forme de façon à extraire les structures les plus contrastées et en même temps similaires à une forme de référence.

### 1.5.1 Information de forme

Parfois la segmentation est appliquée dans un contexte particulier, où une information a priori est disponible sur les objets d'intérêt. Nous proposons d'exploiter cette information en introduisant un facteur de similarité de forme dans la définition de l'ouverture ultime. De cette façon, le masquage lié par exemple aux régions imbriquées peut être limité.

La définition de forme a été étudiée abondamment dans la littérature. Des revues

des méthodes d'analyse de forme peuvent être trouvées dans [55, 56, 57].

Dans notre contexte, nous cherchons à comparer deux formes  $(\Omega_i, \Omega_j)$  via une fonction de similarité  $\psi()$ .

La plupart des approches de segmentation utilisant la forme travaillent au niveau de contours dans un cadre de modèles déformables. Nous proposons de définir une fonction de similarité via des attributs  $\kappa_\Omega$  de régions. Cette similarité servira à pondérer les résidus de manière à favoriser certaines formes de référence. Les régions  $\Omega$  sur lesquelles sont calculés les descripteurs de forme sont les composantes connexes  $CC$ s associées à une décomposition par seuils. Comme nous verrons dans la section 1.6.1, ces composantes seront représentées sous forme d'arbre, le max-tree [39]. Chaque nœud de l'arbre correspond à une composante connexe  $CC$  de la décomposition par seuils (voir figure 1.36). Des descripteurs géométriques simples, comme la hauteur ou la largeur sont utilisés. Des descripteurs plus complexes, comme par exemple les moments de Fourier ou d'autres à base des caractéristiques multiples des  $CC$ s peuvent être utilisés pour décrire la forme.

**Définition 1** *La similarité entre deux formes,  $\Omega_i$  et  $\Omega_j$ , utilisant un attribut  $\kappa$ , est défini comme une fonction  $\psi_\kappa : \Omega \times \Omega \rightarrow [0, 1] \in \mathbb{R}$  et vérifie les conditions suivantes [58] :*

- *Identité* :  $\psi_\kappa(\Omega_i, \Omega_i) = 1$ .
- *Unicité* :  $\psi_\kappa(\Omega_i, \Omega_j) = 1$  implique  $\kappa_{\Omega_i} = \kappa_{\Omega_j}$
- *Symétrie* :  $\psi_\kappa(\Omega_i, \Omega_j) = \psi_\kappa(\Omega_j, \Omega_i)$

Un exemple de fonction de similarité est présenté sur l'équation 1.5 :

$$\psi_\kappa(\Omega_i, \Omega_j) = \begin{cases} \left(1 - \frac{|\kappa_{\Omega_i} - \kappa_{\Omega_j}|}{\tau}\right)^\varsigma & |\kappa_{\Omega_i} - \kappa_{\Omega_j}| \leq \tau \\ 0 & \text{autrement} \end{cases} \quad (1.5)$$

où  $\tau$  est un seuil de similarité selon l'attribut  $\kappa$ . La fonction atteint son maximum quand la différence entre les attributs est nulle ( $\kappa_{\Omega_i} - \kappa_{\Omega_j} = 0$ ), la valeur minimale quand la différence d'attributs est supérieure à un certain seuil  $\tau$  et une valeur dans l'intervalle  $[0,1]$  quand  $|\kappa_{\Omega_i} - \kappa_{\Omega_j}| < \tau$ .  $\varsigma$  détermine si la dynamique est étendue ( $\varsigma < 1$ ), linéaire ( $\varsigma = 1$ ) ou étroite (sélective) ( $\varsigma > 1$ ).

Nous avons défini une fonction de similarité utilisant un seul attribut  $\kappa$ . Dans la pratique, plusieurs attributs sont utilisés pour décrire une forme. Dans ce cas, une simple multiplication des fonctions de similarité peut être appliquée :

$$\psi = \prod_{\forall \kappa} \psi_\kappa(\Omega_i, \Omega_j) \quad (1.6)$$

Les critères suivants doivent être pris en considération pour le choix de la fonction de similarité :

- **Temps de calcul** : Ces mesures sont calculées pour chaque nœud de l'arbre. Pour garder des temps de calcul raisonnables les attributs les plus simples, ceux qui peuvent être calculés pendant la création de l'arbre, sont préférables.
- **Invariance avec la position, l'échelle et la rotation** : l'invariance de la méthode proposée dépendra de celle des attributs choisis. Par exemple, l'ouverture ultime avec un attribut de facteur de remplissage ne sera pas invariant à la rotation mais si nous choisissons la compacité, l'opérateur résultant sera invariant aux trois critères.

### 1.5.2 Ouverture ultime avec information forme

Nous proposons de nous appuyer sur les CCs,  $\Omega$ , correspondent aux nœuds de l'arbre, pour estimer la forme des régions segmentées.

Le facteur de forme  $f(\Omega, \Omega_{ref})$  par rapport à une forme de référence  $\Omega_{ref}$  est noté  $f(\Omega)$  car  $\Omega_{ref}$  est fixe.  $f(\Omega)$  est d'autant plus grand que la forme  $\Omega$  est proche de la forme de référence  $\Omega_{ref}$ . Ainsi, nous proposons de modifier la définition de résidu, en le pondérant par le facteur de forme (Eq. 1.7).

$$r_\lambda^\Omega \leftarrow f(\Omega) r_\lambda \quad (1.7)$$

De cette façon le résidu d'une forme similaire à la forme de référence est artificiellement augmenté. Ainsi, sa présence dans le résultat de segmentation est favorisé, même si elle est imbriquée à l'intérieur d'une autre région plus contrastée, à condition que la région qui la contient ait une forme éloignée de celle de référence. Le facteur de forme  $f(\Omega)$  dépend de la fonction de similarité,  $\psi(\Omega)$ , selon la fonction suivante :  $f(\Omega) = 1 + \alpha\psi(\Omega)$

Un offset de 1 est ajouté pour basculer à la définition de l'ouverture ultime classique quand la forme est très éloignée de la forme de référence. Par ailleurs, un facteur de pondération  $\alpha$  est introduit pour contrôler l'influence de l'information de forme par rapport au contraste. Finalement la fonction  $f_\Omega$  est enregistrée sur l'image  $F_\theta^\Omega(I)$  quand le résidu maximal ( $R_\theta(I)$ ) est généré. Nous pouvons maintenant proposer la définition suivante :

**Définition 2** *L'ouverture ultime avec information de forme,  $\theta^\Omega$ , d'une image  $I$  est :*

$$\theta^\Omega(I) : I \xrightarrow{\theta^\Omega} (R_\theta^\Omega(I), q_\theta^\Omega(I), F_\theta^\Omega(I)) \quad (1.8)$$

$$R_\theta^\Omega(I) = \max_\lambda (r_\lambda^\Omega(I)) = \max_\lambda (f(\Omega) \times r_\lambda(I)) \quad (1.9)$$

$$q_\theta^\Omega(I) = \begin{cases} \max \{ \lambda + 1 \mid r_\lambda^\Omega(I) = R_\theta^\Omega(I) \} & R_\theta^\Omega(I) > 0 \\ 0 & R_\theta^\Omega(I) = 0 \end{cases} \quad (1.10)$$

$$F_{\theta}^{\Omega}(I) = \begin{cases} f(\Omega) & | r_{\lambda}^{\Omega}(I) = R_{\theta}^{\Omega}(I) & R_{\theta}^{\Omega}(I) > 0 \\ 0 & & R_{\theta}^{\Omega}(I) = 0 \end{cases} \quad (1.11)$$

où,  $\Omega$  est une CC associée à un nœud de l'arbre (voir figure 1.36) et  $f(\Omega)$  est le facteur de forme.  $F_{\theta}^{\Omega}(I)$  indique la similarité des régions segmentées avec la forme de référence.

**Exemple sur une image de synthèse** Le fonctionnement de l'approche est montré dans un premier temps sur une image de synthèse (figure 1.25). Tout d'abord, nous favorisons les formes rectangulaires. Pour ceci, nous utilisons le facteur de remplissage  $\Upsilon_{\Omega} = \frac{A_{\Omega}}{Abbox_{\Omega}}$ , où  $A_{\Omega}$  est la surface de la forme et  $Abbox_{\Omega}$  celle de sa boîte englobante. Le rapport entre les deux est compris dans l'intervalle  $[0,1]$ ; si la valeur est proche de 1, la forme correspond à un rectangle dont les côtés sont parallèles aux axes de l'image. Ensuite, nous avons imposé une surface maximale (90 % de la surface de l'image  $A_I$ ) pour rejeter les trop grandes régions. La fonction utilisée est résumée sur l'équation 1.12 :

$$\begin{aligned} f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \\ \text{où, } \alpha &= \max(I)/2 \\ \psi_{\kappa_1}(\Omega) &= \Upsilon_{\Omega} \\ \psi_{\kappa_2}(\Omega) &= \begin{cases} 1 & A_{\Omega} < 90\%A_I \\ 0 & \text{autrement} \end{cases} \end{aligned} \quad (1.12)$$

La figure 1.27 présente le résultat sur l'image de synthèse. Le problème de masquage est supprimé et trois formes sont segmentées. L'importance de  $\psi_{\kappa_2}(\Omega)$  est remarquable dans cet exemple, car la région qui produit le masquage a un facteur de remplissage élevé  $\Upsilon_{\Omega} \approx 1$ .

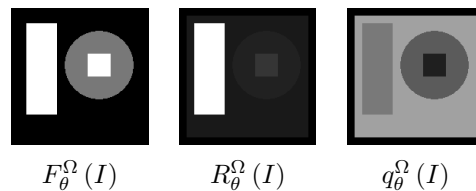


FIGURE 1.27 – *SUAO* de la figure 1.25 avec l'équation 1.12 comme facteur de forme, favorisant les formes rectangulaires.

Malgré le fait que les formes rectangulaires soient favorisées, le cercle est également segmenté car son facteur de remplissage est suffisamment important. Si nous voulons être plus sélectifs nous pouvons utiliser une fonction  $\psi_{\kappa_1}(\Omega)$  plus étroite. Par exemple nous remplaçons  $\Upsilon_{\Omega}$  dans Eq. 1.12 par  $\Upsilon_{\Omega}^3$  :

$$\begin{aligned}
f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \\
\text{où, } \alpha &= \max(I)/2 \\
\psi_{\kappa_1}(\Omega) &= (\Upsilon_{\Omega})^3 \\
\psi_{\kappa_2}(\Omega) &= \begin{cases} 1 & A_{\Omega} < 90\% A_I \\ 0 & \text{autrement} \end{cases}
\end{aligned} \tag{1.13}$$

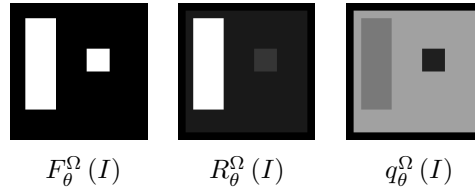


FIGURE 1.28 – *SUAO* de la figure 1.25 avec l'équation 1.13 comme facteur de forme, favorisant plus sélectivement les formes rectangulaires.

L'exemple suivant favorise la segmentation de régions circulaires. Le facteur de forme utilisé (1.14) est donc la circularité :  $\frac{4\pi A_{\Omega}}{(L_{\Omega})^2}$ , avec  $A$  la surface et  $L$  le périmètre de  $\Omega$ . Les figures 1.28 et 1.29 confirment la capacité de l'algorithme à segmenter des régions d'une certaine forme.

$$\begin{aligned}
f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \\
\text{où, } \alpha &= \max(I)/2 \\
\psi_{\kappa_1}(\Omega) &= \frac{4\pi A_{\Omega}}{(L_{\Omega})^2}
\end{aligned} \tag{1.14}$$

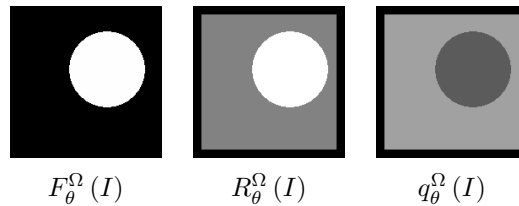


FIGURE 1.29 – *SUAO* de la figure 1.25 avec l'équation 1.14 comme facteur de forme, favorisant les régions de forme circulaire.

Nous allons montrer l'intérêt de cet opérateur à travers trois applications : l'analyse de façades, la localisation de texte enfoui et la segmentation de cellules.

### 1.5.3 Analyse de façades

Cette étude se situe dans le cadre de la modélisation de façades [59, 60] qui nécessite la détection des éléments structurant celles-ci, principalement des fenêtres. Cette

analyse a été réalisée dans le cadre du projet TerraNumerica (<http://cmm.ensmp.fr/TerraNumerica>) du Pôle de compétitivité Cap Digital. Son objectif était d'augmenter la productivité et le réalisme de la modélisation urbaine.

Comme illustré sur la figure 1.30(a-c), la plupart des structures internes de la façade sont masquées par l'ouverture ultime à cause du fort contraste entre la façade et le ciel. La présence du ciel contrasté est courante dans les images urbaines, c'est pourquoi le problème de masquage a un fort impact sur la robustesse de l'algorithme.

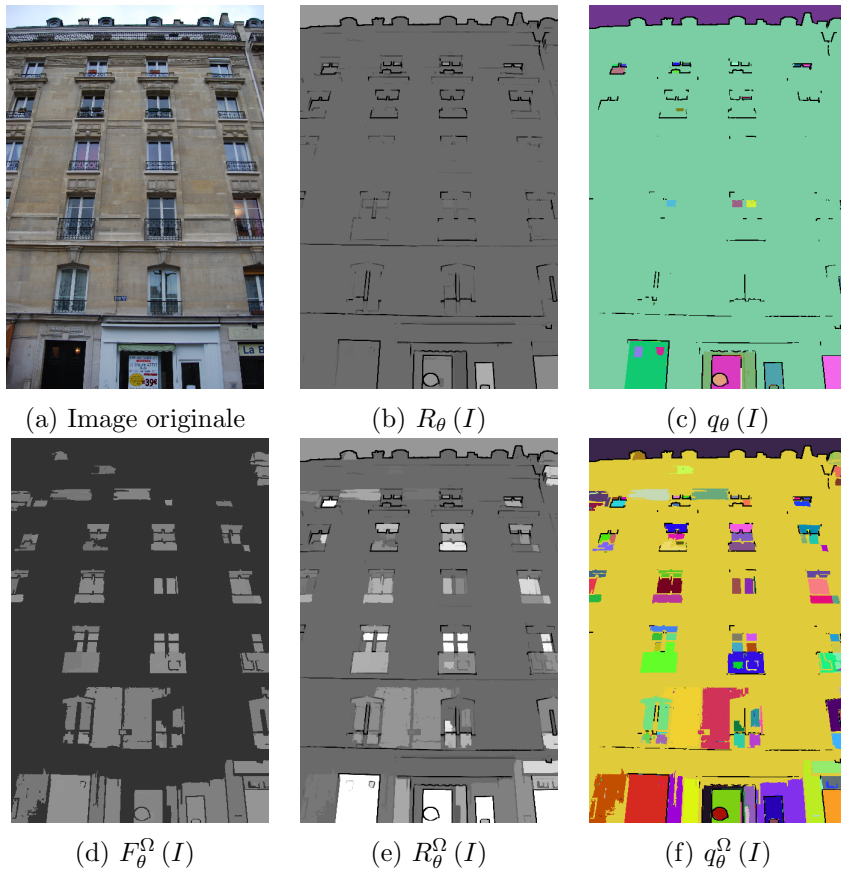


FIGURE 1.30 – (a) Original image, (b)-(c)  $R_\theta(I)$  et  $q_\theta(I)$  de  $UAO$ , (d)-(f)  $F_\theta^\Omega(I)$ ,  $R_\theta^\Omega(I)$  et  $q_\theta^\Omega(I)$  de  $SUAO$ .

Les fenêtres et les portes sont majoritairement de forme rectangulaire. Mayer et Reznik [61] proposent d'utiliser le taux de remplissage,  $(\Upsilon_\Omega)$ , et le rapport hauteur/largeur,  $(\aleph_\Omega = \frac{height}{width})$ , comme des caractéristiques discriminantes. Ils constatent que les valeurs habituelles de cette dernière caractéristique se situent entre 0.2 et 0.5. Avec ces informations, nous définissons le facteur de forme suivant :



$$\begin{aligned}
 f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \\
 \psi_{\kappa_1}(\Omega) &= (\Upsilon_\Omega)^2 \\
 \psi_{\kappa_2}(\Omega) &= \begin{cases} (\aleph_\Omega)^{0.5} & \aleph_\Omega > 0.2 \\ 0 & \text{autrement} \end{cases}
 \end{aligned} \tag{1.15}$$

Le premier attribut est le taux de remplissage ( $\kappa_1 = \Upsilon_\Omega$ ). Une valeur quadratique est utilisée pour le rendre plus sélectif  $\varsigma = 2$ . Le deuxième attribut  $\kappa_2$  est le rapport d'aspect  $\aleph$ , habituellement défini comme le rapport ( $h/w$ ). Ce rapport est très sensible à des faibles variations de la largeur. Nous avons préféré la formulation suivante  $\aleph = \frac{\min(h,w)}{\max(h,w)}$ , qui est plus stable. Une dynamique large, avec  $\varsigma = 0.5$  et tronquée à une valeur  $\aleph > 0.2$  est utilisée. De manière heuristique, nous avons constaté que le gradient entre la façade et le ciel est environ 10 fois plus fort que le contraste entre les fenêtres et le mur. Pour cette raison, nous choisissons une valeur  $\alpha = 9$ , i.e.  $1 + \alpha = 10$ .

Le résultat de la segmentation est illustré sur la figure 1.30(d-f). Nous constatons que les structures d'intérêt ont été correctement segmentées.

Pour illustrer la robustesse du facteur de forme aux effets de perspective, la figure 1.31 montre l'évolution du facteur pour une fenêtre tournant autour de son axe Z (angle de vue entre  $0^\circ$  et  $80^\circ$ ). Le facteur de forme décroît avec la perspective. Il est néanmoins d'environ 0.5 pour une inclinaison de  $40^\circ$ . La figure 1.32 montre le résultat de segmentation avec un bon résultat, y compris pour les fenêtres qui ne sont pas fronto-parallèles.

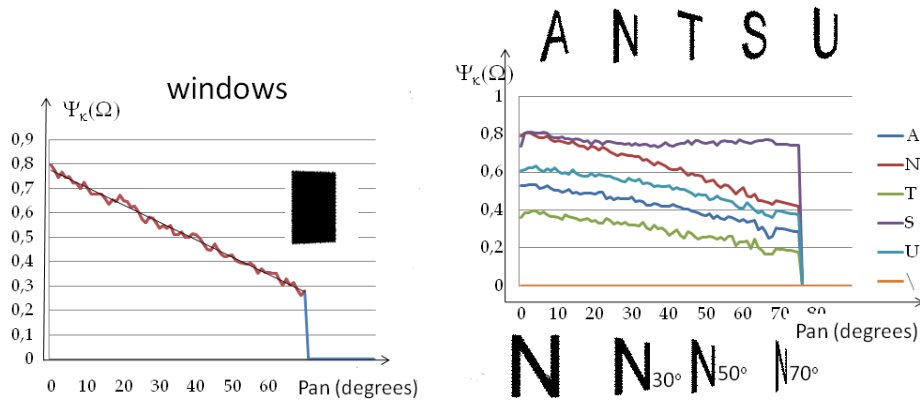


FIGURE 1.31 – Evolution du facteur de forme avec l'angle de rotation autour de l'axe Z, pour les fenêtres (à gauche) et les lettres (à droite).

#### 1.5.4 Localisation de texte enfoui

Le texte dans une scène est souvent relié au contenu sémantique de l'image et constitue un descripteur haut niveau pour une indexation basée sur le contenu [63].

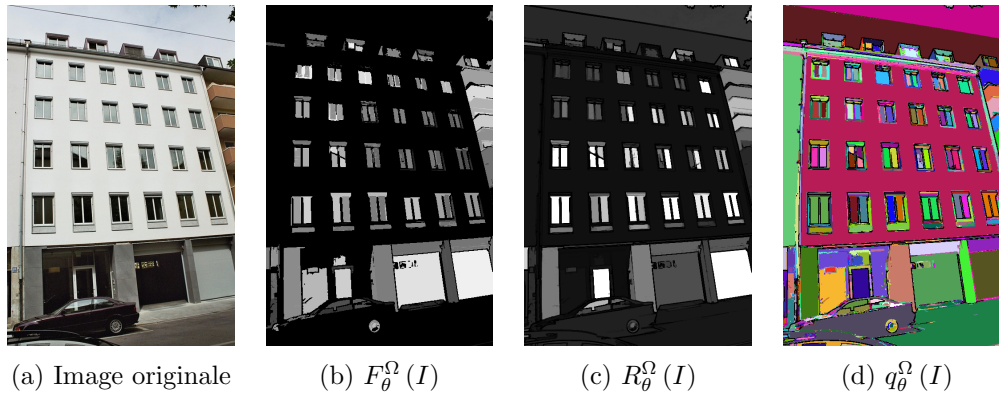


FIGURE 1.32 – (a) Image originale [62], (b)-(d)  $F_{\theta}^{\Omega}(I)$ ,  $R_{\theta}^{\Omega}(I)$  et  $q_{\theta}^{\Omega}(I)$  de *SUAO*.

Cette étude a été réalisée dans le cadre du projet iTowns (Image-based Town On-line Web navigation and Search Engine, <http://www.itowns.fr/>). L'objectif de ce projet est de développer une nouvelle génération d'outils web multimédia qui combinent un navigateur géographique 3D avec un moteur de recherche basé sur une base de données indexée et le contexte visuel.

Le texte est parfois sur un support (pancarte, poster,...) qui favorise sa visibilité. Ce support est souvent contrasté par rapport à son environnement (Figure 1.33 et Figure 1.34) et peut provoquer le phénomène de masquage dont nous avons parlé précédemment.

L'utilisation d'une information a priori des lettres permettra de réduire ce problème de masquage. Les attributs des lettres, extraits de l'analyse de 5000 caractères, sont décrits comme suit :

1. 97% des lettres ont un rapport d'aspect supérieur à 0.4,
2. 84% des lettres ont un taux de remplissage entre 0.2 et 0.9 et
3. la hauteur la plus grande (respectivement largeur) d'un caractère est 1/3 de la hauteur  $H_I$  (resp. largeur  $W_I$ ) de l'image.

L'équation 1.16 résume le facteur de forme utilisé pour la segmentation de texte. La figure 1.31 montre l'évolution de ce facteur avec l'angle de rotation, pour différentes lettres. Certaines lettres, comme le S ou le N, ont un facteur de forme élevé qui décroît lentement avec l'angle de rotation. Ainsi, le résidu de ces lettres sera renforcé, augmentant leur probabilité d'être correctement segmentées. D'autres lettres, celles avec un taux de remplissage plus bas, comme le T, ont un facteur de forme plus faible, particulièrement pour des angles de rotation importants. Dans ces cas, la combinaison avec le contraste est nécessaire pour les sélectionner comme *CCs* significatives.

$$\begin{aligned}
& f(\Omega) = 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \psi_{\kappa_3}(\Omega) \\
\text{où, } & \alpha = 9 \\
& \psi_{\kappa_1}(\Omega) = \begin{cases} (\aleph_\Omega)^{0.5} & \aleph_\Omega > 0.4 \\ 0 & \text{autrement} \end{cases} \\
& \psi_{\kappa_2}(\Omega) = \begin{cases} \left(1 - \left(\frac{\Upsilon_\Omega - 0.55}{0.35}\right)^{0.5}\right) & 0.2 < \Upsilon_\Omega < 0.9 \\ 0 & \text{autrement} \end{cases} \\
& \psi_{\kappa_3}(\Omega) = \begin{cases} 1 & h_\Omega < \frac{H_I}{3} \wedge w_\Omega < \frac{W_I}{3} \\ 0 & \text{autrement} \end{cases}
\end{aligned} \tag{1.16}$$

Par ailleurs, le texte enfoui peut être clair ou sombre. Pour tenir compte des deux polarités l'opérateur est appliqué à l'image et à son inverse. Pour chaque pixel, la polarité de plus fort résidu est sélectionnée (voir Eq. 1.17).

$$\begin{aligned}
R_\theta(I, I^c) &= \max(R_\theta(I), R_\theta(I^c)) \\
q_\theta(I, I^c) &= \begin{cases} q_\theta(I) & R_\theta(I) > R_\theta(I^c) \\ q_\theta(I^c) & \text{autrement} \end{cases}
\end{aligned} \tag{1.17}$$

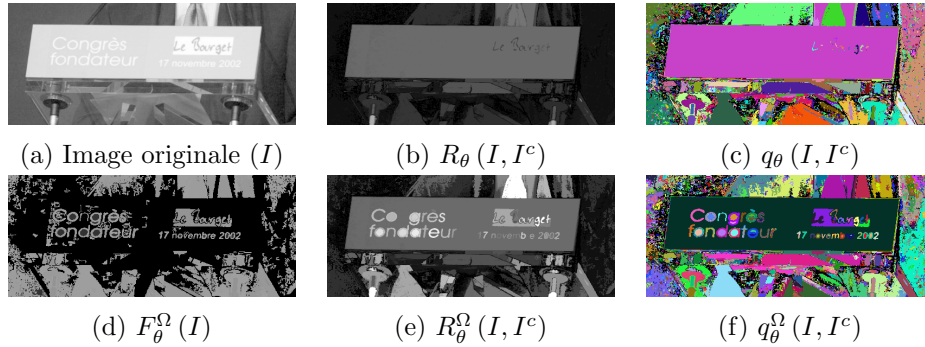


FIGURE 1.33 – (a) Image originale, (b)-(c)  $R_\theta(I, I^c)$  et  $q_\theta(I, I^c)$  de  $UAO$ , (d)-(f)  $F_\theta^\Omega(I)$ ,  $R_\theta^\Omega(I, I^c)$  et  $q_\theta^\Omega(I, I^c)$  de  $SUAO$ .

Les figures 1.33 et 1.34 montrent les résultats de la détection de texte utilisant l'ouverture ultime par attribut  $UHO$  et  $SUHO$  d'une image contenant un panneau avec du texte dessus. En utilisant  $UAO$ , la plupart des lettres sont masquées. Avec  $SUHO$ , les lettres sont correctement segmentées. Par ailleurs, le fond du panneau n'est pas sur-segmenté. La méthode proposée, l'ouverture ultime avec une information de forme ( $SUHO$ ) donne de meilleurs résultats que l'ouverture ultime par attribut  $UHO$ .

### 1.5.5 Segmentation de cellules

L'émergence récente de technologies automatisées d'acquisition d'images à haut débit a changé à jamais la manière dont les biologistes cellulaires collectent et analysent

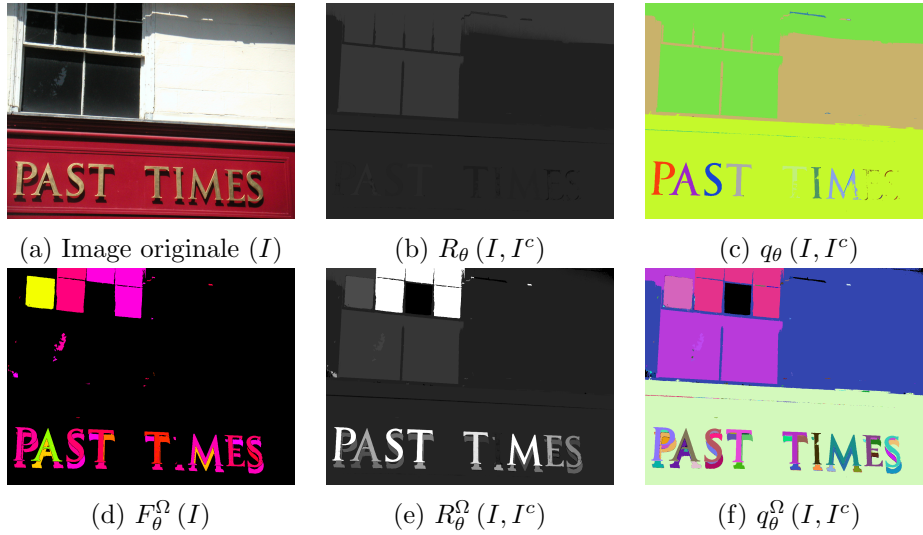


FIGURE 1.34 – (a) Image originale, (b)-(c)  $R_\theta(I, I^c)$  et  $q_\theta(I, I^c)$  de *UAO*, (d)-(f)  $F_\theta^\Omega(I, I^c)$ ,  $R_\theta^\Omega(I, I^c)$  et  $q_\theta^\Omega(I, I^c)$  de *SUAQ*.

les données [64]. L'interprétation des phénotypes cellulaires est maintenant un sujet majeur de recherche en biologie. Ainsi, le développement de méthodes automatisées qui visent à identifier des nouveaux phénotypes pertinents est un défi majeur. Par exemple, le blocage d'un gène avec un ARN interférence (ARNi) peut mettre en évidence son effet sur le fonctionnement normal de la cellule. L'observation d'une molécule toxique peut montrer de nouveaux modes d'action et mettre sur la bonne voie pour développer de nouveaux médicaments. Cette étude est réalisée dans le cadre du projet RAMIS (Rock Analysis Module and Interface for Screening) du Pôle Cancer-Bio- Santé de Toulouse. Une collection d'images acquises par marquage multi-paramétrique des cellules est analysée.

En analyse d'images, les propriétés morphologiques (taille, forme, texture, etc.) de cellules fournissent des informations précieuses à explorer pour mettre au point une étape de segmentation. Nous proposons une approche de segmentation basée sur l'*UAO*. Le problème de masquage est également présent sur ces images. En effet, le halo autour des cellules provoque la fusion de celles qui sont proches entre elles. En analysant les histogrammes d'environ 1200 cellules, les cellules sont décrites comme suit : 1- la circularité est un attribut discriminant, 2- la surface ( $A$ ) est entre 500 et 3000 pixels et 3- la hauteur des cellules ( $h$ ) et la largeur ( $w$ ) se situent entre 15 et 110 pixels. L'équation 1.18 montre la fonction de facteur de forme résultante :

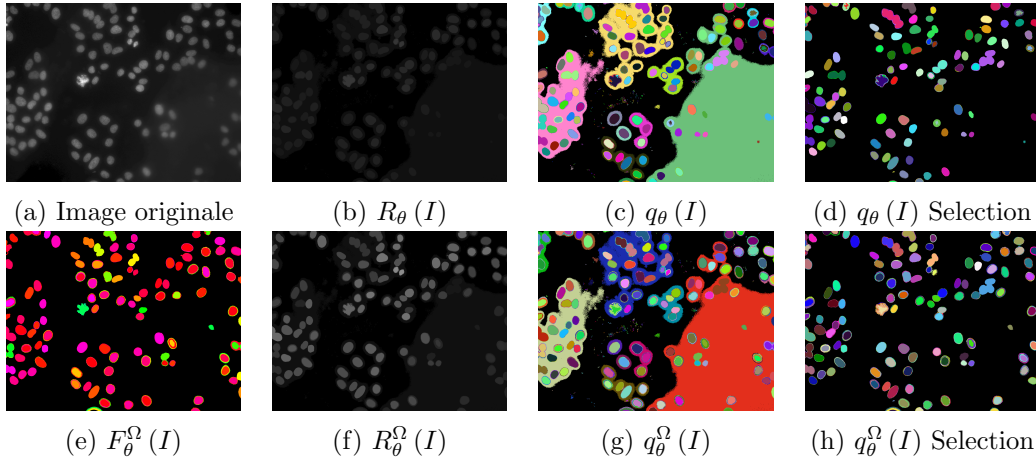


FIGURE 1.35 – (a) Image originale : exemple d’image de fluorescence RNAi , (b)-(d)  $R_\theta(I)$   $q_\theta(I)$  et  $q_\theta(I)$  + filtrage grossier sur  $UAO$ , (e)-(h)  $F_\theta^\Omega(I)$ ,  $R_\theta^\Omega(I)$ ,  $q_\theta^\Omega(I)$  et  $q_\theta^\Omega(I)$  + filtrage grossier sur  $SUAO$ .

$$\begin{aligned}
 f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \psi_{\kappa_3}(\Omega) \\
 \text{où, } \alpha &= 10, \psi_{\kappa_1}(\Omega) = \frac{4\pi A_\Omega}{(L_\Omega)^2} \\
 \psi_{\kappa_2}(\Omega) &= \begin{cases} 1 & 500 < A_\Omega < 3000 \\ 0 & \text{autrement} \end{cases} \\
 \psi_{\kappa_3}(\Omega) &= \begin{cases} 1 & h_\Omega \text{ et } w_\Omega \in (15, 110) \\ 0 & \text{autrement} \end{cases}
 \end{aligned} \tag{1.18}$$

La figure 1.35 montre la comparaison entre  $UAO$  et  $SUAO$ . Noter que  $q_\theta(I)$  et  $q_\theta^\Omega(I)$  présentent beaucoup de  $CCs$  dues au bruit. Ainsi, une fois le  $SUAO$  effectué, un filtrage grossier des  $CCs$  obtenues est appliqué afin d’éliminer les structures de faible contraste. Ce filtrage consiste en un simple seuil, de valeur 2, sur la moyenne de l’image résiduelle ( $R_\theta(I)$ ) à l’intérieur des composantes connexes de  $q_\theta(I)$ . Les deux segmentations sont similaires (figure 1.35(c) et figure 1.35(g)). Toutefois, le  $SUAO$  a démasqué plusieurs cellules sur la région en bas à droite et a séparé des cellules qui étaient réunies. Notre approche ne peut pas éviter un problème de fuite lorsque celle-ci a lieu alors que les  $CCs$  n’ont pas un facteur de forme élevé. En outre, elle réduit la segmentation des noyaux et favorise les structures correspondant aux cellules entières. Les résultats expérimentaux indiquent que l’approche proposée peut être utilisée pour segmenter les cellules pour une analyse ultérieure, en réduisant les problèmes de la  $UAO$  classique.

## 1.6 Algorithme rapide d'ouverture ultime utilisant l'arbre des maxima

L'implémentation naïve de l'ouverture ultime à partir de sa définition est très coûteuse car elle nécessite le calcul de nombreuses ouvertures. Nous avons proposé, dans le cadre du post-doc de Jonathan Fabrizio [38], une implémentation rapide de l'ouverture ultime par attribut, fondée sur une représentation de max-tree [39]. La vitesse de calcul atteinte devient alors compatible avec des applications temps réel pour des images de taille relativement importante (1000x1000). Nous décrivons ci-dessous cette implémentation, ainsi que les modifications à apporter pour dériver les différentes améliorations de l'opérateur que nous avons proposées.

### 1.6.1 Max-Tree

Le max-tree, par dualité min-tree, a été introduit par Salembier [39] comme une structure pour implémenter des opérateurs connexes. Ici, nous introduisons succinctement le concept de max-tree. Pour une définition complète se référer à [39]. Le max-tree est un arbre, dans le sens de la théorie des graphes, qui donne une représentation multi-échelle d'une image. Les nœuds représentent les  $CCs$  de  $X_t(I)$ , les versions seuillées de l'image au niveau  $t$ . Les liens parents-enfants décrivent les relations d'inclusion entre les  $CCs$  correspondantes. Ainsi, le nœud racine correspond à la totalité de l'image tandis que les nœuds feuille correspondent aux maxima de l'image. Un ou plusieurs attributs, calculés sur les  $CCs$ , peuvent être associés à chaque nœud, ce qui permet un calcul efficace des ouvertures par attribut. Un exemple est présenté sur la figure 1.36. L'attribut  $\kappa$  utilisé est la hauteur de la  $CC$  associée. Le nœud racine,  $A_0^4$ , de niveau 0 et avec un attribut de hauteur  $\kappa = 4$  a deux enfants  $B_1^3$  au niveau 1 et  $C_2^3$  au niveau 2, les deux avec un attribut  $\kappa = 3$ . La composante connexe  $C$  demeure inchangée à partir de  $X_1(I)$  à  $X_2(I)$ . Dans ce cas, le seuil maximal pour lequel la composante connexe est la même, est affecté au nœud (niveau 2 dans notre exemple). Chaque branche conduit à un nœud feuille ( $E_5^1$  et  $F_6^2$ ) correspondant aux maxima régionaux de l'image, à des niveaux 5 et 6 respectivement.

Une ouverture par attribut  $\gamma_\lambda$  supprime les nœuds,  $n$ , dont l'attribut  $\kappa(n)$  est inférieur à  $\lambda$ . La figure 1.37 montre les résultats d'ouverture par attribut, avec de valeurs de  $\lambda$  croissantes. Ces ouvertures sont obtenues par l'élagage de l'arbre aux lignes discontinues correspondantes. Cette opération est extrêmement rapide, c'est pourquoi la structure de max-tree est appropriée pour la mise en oeuvre de l'ouverture ultime.

### 1.6.2 Calcul de résidus

Le résidu entre deux ouvertures consécutives est calculé comme la différence de niveau de gris d'un nœud et le niveau de gris de son premier ancêtre avec un attribut différent, comme suit :

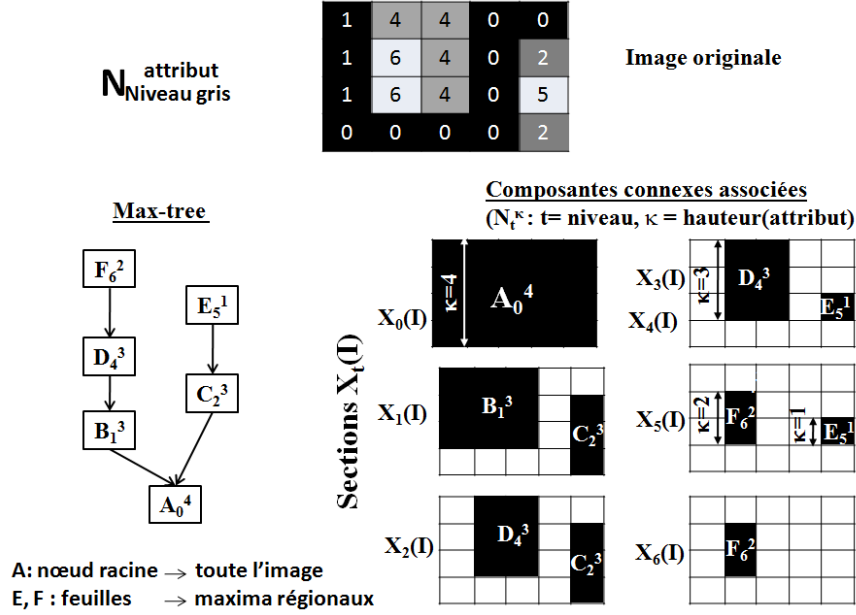


FIGURE 1.36 – Représentation de l'arbre de maxima (max-tree).

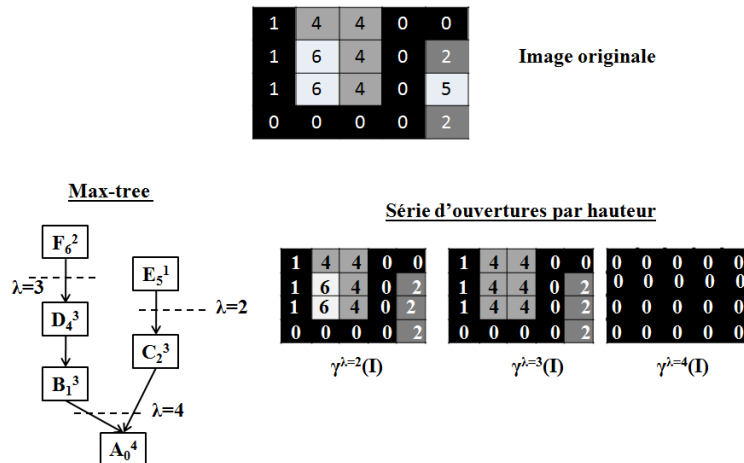
$$r_\lambda(child) = \begin{cases} t(child) - t(parent) & \kappa(parent) \neq \kappa(child) \\ t(child) - t(parent) + r_\lambda(parent) & \text{autrement} \end{cases} \quad (1.19)$$

où  $child$  et  $parent$  sont deux nœuds reliés du max-tree,  $t(n)$  le niveau de gris associé au nœud  $n$  et  $\kappa(n)$  son attribut correspondant.

La figure 1.38 illustre un exemple de calcul de résidus sur le max-tree. L'attribut de hauteur est choisi pour l'exemple (notons  $H_k$  la hauteur de la région de  $k$ ). Chaque nœud de l'arbre contient l'information du niveau de gris et de l'attribut de la région correspondante. Notez que les régions  $B$  et  $D$  ont le même attribut. Les deux régions sont alors éliminées par la même ouverture, de taille  $H_B + 1$  ( $= H_D + 1$ ). Le résidu généré sur la région  $D$  par  $\gamma_{H_B+1}$  n'est pas  $l_D - l_B$  (la différence de niveau de gris avec son nœud parent), mais  $l_D - L_A$ ,  $A$  étant son premier ancêtre avec un attribut différent.

### 1.6.3 Calcul de l'ouverture ultime à partir des résidus

Le  $UAO$  analyse le résidu des ouvertures successives de taille croissante et le résidu le plus élevé est enregistré (figure 1.38, Eq. 1.1). Pendant le processus, une ouverture  $\gamma_i$  peut générer un résidu  $r_1$  pour un nœud  $n_1$ . Plus tard, une ouverture plus grande  $\gamma_j$  générera un résidu  $r_2$  pour un nœud  $n_2$  ancêtre de  $n_1$  dans l'arbre. Comme  $n_2$  englobe  $n_1$ , si  $r_2$  est plus grand que  $r_1$ ,  $r_2$  doit également être attribué à

FIGURE 1.37 – Max-Tree et ouvertures par hauteur  $\gamma_T(I)$ , avec  $T : \kappa \geq \lambda$ .

$n_1$  (sinon  $n_1$  maintient  $r_1$ ). Pour trouver le résidu le plus élevé de tous les nœuds, chaque ancêtre  $n_k$  de chaque nœud doit être vérifié. C'est pourquoi la structure de données de max-tree est adaptée à ce traitement : le résidu le plus élevé ne sera pas recherché parmi les ancêtres, mais tous les ancêtres transmettront leur propre résidu maximal. Le pseudo-code est présenté dans l'algorithme 1. L'arbre est traité de manière récursive : sur un nœud donné, le résidu calculé est transmis à chaque enfant (voir l 21-22 du pseudo-code). Chaque enfant va comparer son propre résidu ( $r[node]$ ) avec celui transmis par ses ancêtres,  $R_\theta(parent)$  (l 14) et garde le maximum d'entre eux ( $R_\theta(node) = \max(r[node], R_\theta(parent))$ ) ainsi que la taille de l'attribut d'ouverture associé au résidu maximal. Plus précisément, si  $r[node] > R_\theta(parent)$ ,  $R_\theta(node)$  prendra la valeur du résidu actuel ( $R_\theta(node) = r[node]$ ) et  $q_\theta(node)$  l'attribut du nœud +1 (l 15-16). Sinon, les valeurs  $R_\theta(parent)$  et  $q_\theta(parent)$  sont propagées (l 18-19). Ce maximum est transmis à nouveau aux enfants et ainsi de suite. Ce processus est en  $O(n)$ , avec  $n$  le nombre de nœuds.

La figure 1.38 montre le calcul complet de l'ouverture ultime étape par étape sur notre image synthétique de la figure 1.36. Dans un premier temps, le résidu de chaque nœud  $n$  est calculé. Ce résidu est la différence entre son niveau de gris  $t_n$  et le niveau de gris de son premier ancêtre  $a$  dont l'attribut  $\kappa(a)$  est différent de  $\kappa(n)$ . Dans notre exemple, le résidu de chaque nœud est la différence entre son niveau de gris et celui de son parent direct, sauf pour le nœud  $D$ . En effet, la seule paire de nœuds père-fils avec le même attribut est  $D$  et  $B$ . Par conséquent, les deux nœuds sont éliminés par la même ouverture et le résidu de  $D$  est alors  $r_D = t_D - t_A = 4 - 0 = 4$  (et non pas  $t_D - t_B$ ). Une fois le résidu de chaque nœud calculé,  $R_\theta$  est obtenu par propagation du résidu maximum en partant de la racine :  $R_\theta(n) = \max(R_\theta(parent), r(n))$ . En ce qui concerne  $q_\theta$ , si  $R_\theta(n) = r(n)$ , alors  $q_\theta(n) = \kappa(n) + 1$ . Sinon,  $q_\theta(n) = q_\theta(parent)$ .



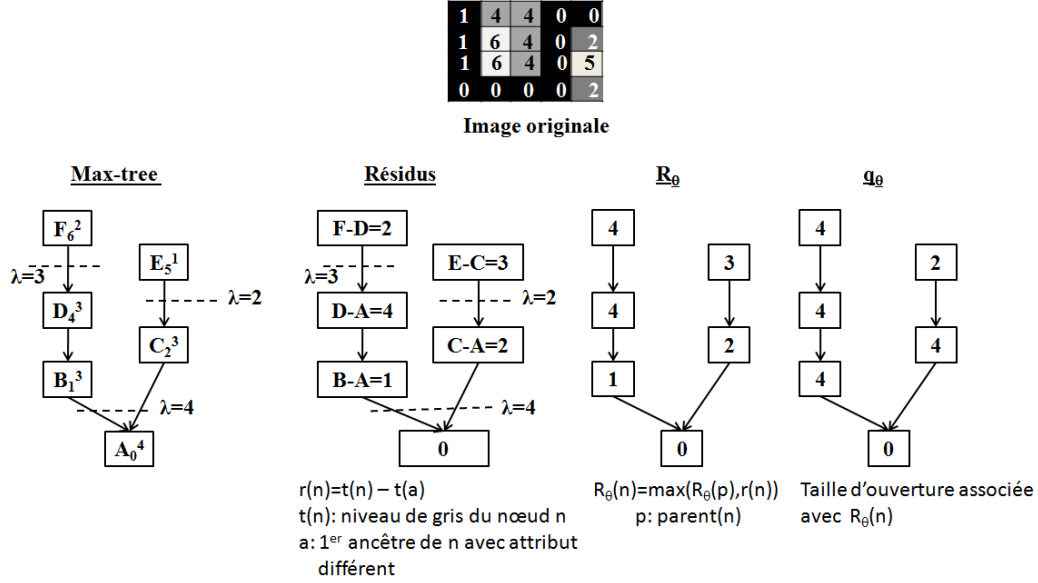


FIGURE 1.38 – Calcul des résidus et de l'ouverture ultime sur l'arbre. L'attribut utilisé est la hauteur de la composante connexe.

Pour notre exemple synthétique,  $R_\theta(n)$  est égal à  $r(n)$  pour tous les nœuds sauf pour le nœud  $F$  où  $R_\theta(\text{parent}) = R_\theta(D) = 4 > r(F) = 2$ . C'est pourquoi  $R_\theta(D)$  est propagé au nœud  $F$ , et que  $q_\theta(F) = q_\theta(D)$ .

---

**Algorithme 1** : Calcul Ouverture ultime par attribut.

---

- 1 //Initialisation
  - 2  $R_\theta(\text{root}) = 0$
  - 3  $q_\theta(\text{root}) = 0$
  - 4 //Compute children
  - 5 **foreach** *child* in  $Children(\text{root})$  **do**
  - 6     ComputeNode(*child*,*root*)
- 

### 1.6.4 Génération des images résultantes

La dernière étape consiste à générer les deux résultats (la transformée et la fonction associée). Ceci est une étape facile. Lors de la création de l'arbre, une image labélisée ( $lab_{img}$ ) permet d'établir la correspondance entre les nœuds et les composantes connexes associées. Une simple look-up-table entre les valeurs  $R_\theta[node]$  et  $q_\theta[node]$  calculées suffit pour générer les images résultantes. L'image transformée et l'image associée sont ainsi générées de la manière suivante :

---

**Algorithme 2** : Compute Node.

---

```

7 ComputeNode(node, parent)
8 //Verify attribute //Compute residue  $r_\lambda(\textit{node})$ 
9 if ( $(\kappa(\textit{parent}) == \kappa(\textit{node}))$ ) then
10    $r_\lambda(\textit{node}) = t(\textit{node}) - t(\textit{parent}) + r_\lambda(\textit{parent})$ 
11 else
12    $r_\lambda(\textit{node}) = t(\textit{node}) - t(\textit{parent})$ 
13 //Compute  $R_\theta(\textit{node})$  and  $q_\theta(\textit{node})$ 
14 if ( $r_\lambda(\textit{node}) > R_\theta(\textit{node})$ ) then
15    $R_\theta(\textit{node}) = r_\lambda(\textit{node})$ 
16    $q_\theta(\textit{node}) = \kappa(\textit{node}) + 1$ 
17 else
18    $R_\theta(\textit{node}) = R_\theta(\textit{parent})$ 
19    $q_\theta(\textit{node}) = q_\theta(\textit{parent})$ 
20 //Compute children
21 foreach child in Children(node) do
22   ComputeNode(child, node)

```

---

```

01 for all pixel p {
02   transform[p]= $R_\theta[\textit{lab}_{img}[\textit{p}]]$ ;
03   associate[p]= $q_\theta[\textit{lab}_{img}[\textit{p}]]$ ;
04 }

```

### 1.6.5 Temps de calcul

L'efficacité de cette mise en œuvre est évaluée sur une base d'images personnelle d'environ 500 photographies. L'ouverture ultime par hauteur est appliquée 10 fois à chaque image et le temps moyen est donné. Le tableau 1.39 donne le temps de calcul obtenu pour différentes tailles d'images. Le résultat est à peu près linéaire par rapport au nombre de pixels dans l'image (figure 1.39.) Sauf pour les très petites images; le cache peut introduire un biais. Tous les tests ont été effectués sur un ordinateur portable DELL D630 avec processeur 2,4 GHz T7700 et la mise en œuvre est en C. Les temps indiqués comprennent l'ensemble du processus : toutes les allocations, la création de l'arbre, le calcul de l'*UAO*, la génération des images résultantes et le nettoyage de la mémoire (données intermédiaires tels que l'arbre, la structure LIFO ...). Seuls les opérations E/S ne sont pas incluses dans les temps donnés. Par ailleurs, la distribution des temps d'exécution par étape de traitement est la suivante : 72 % du temps en moyenne est passé pour construire l'arbre, 9 % et 19 % est consacré à traiter l'arbre et générer les images résultantes respectivement. Cette mise en œuvre permet l'utilisation de l'ouverture ultime par attribut dans différents contextes et même en temps réel pour des images relativement grandes.

Format	128x128	256x256	512x512	1024x1024	2048x2048
Nombre de pixels	16384	65536	262144	1048576	4194304
Temps (ms)	0,18	2,39	12,01	52,04	235,53

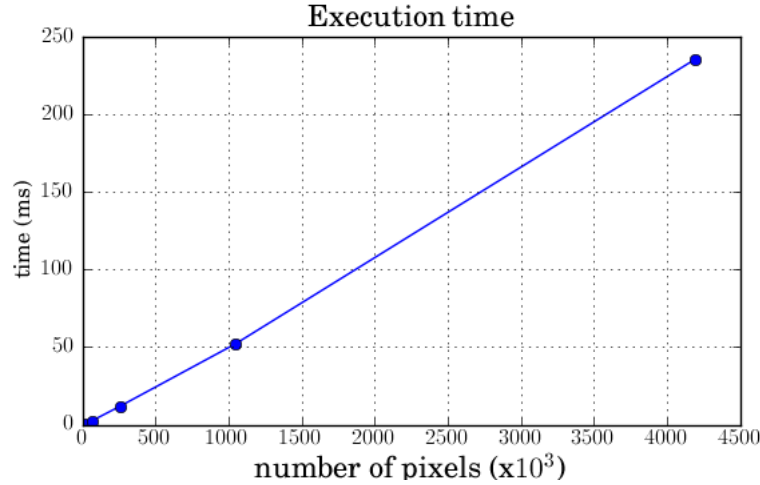


FIGURE 1.39 – Temps d'exécution de l'UAO, selon le nombre de pixels [38].

### 1.6.6 Implémentation de $\Delta UAO$ sur le max-tree

Afin d'implémenter  $\Delta UAO$ , la condition ( $\kappa(\text{parent}) == \kappa(\text{node})$ ) de l'équation 1.19 est remplacée par  $(\kappa(\text{parent}) - \kappa(\text{node})) \leq \Delta$ . La raison en est la suivante : une ouverture de la taille  $\kappa(\text{node}) + 1$  va filtrer la région correspondant au nœud  $\text{node}$ , lui attribuant la valeur de son  $\text{parent}$ . Ainsi, l'ouverture suivante produisant un résidu dans cette région serait l'ouverture de taille  $\kappa(\text{parent}) + 1$ , qui supprime le nœud  $\text{parent}$ . Donc, les ouvertures allant de taille  $\kappa(\text{node}) + 2$  à la taille  $\kappa(\text{parent})$  (toutes les deux incluses) ne modifieront pas la région, menant à une série de  $\kappa(\text{parent}) - \kappa(\text{node}) - 1$  résidus nuls. Étant donné que, selon la définition de  $\Delta UAO$ , les résidus doivent être intégrés jusqu'à trouver une série de  $\Delta$  résidus nuls, la condition à cette intégration est alors  $(\kappa(\text{parent}) - \kappa(\text{node})) \leq \Delta$ . La formule pour le calcul de  $r_\lambda^\Delta$  devient :

$$r_\lambda^\Delta(\text{node}) = \begin{cases} t(\text{node}) - t(\text{parent}) + r_\lambda^\Delta(\text{parent}) & \kappa(\text{parent}) - \kappa(\text{node}) \leq \Delta \\ t(\text{node}) - t(\text{parent}) & \text{otherwise} \end{cases} \quad (1.20)$$

Le pseudo-code pour  $\Delta UAO$  est donné par l'algorithme 3.  $R_\theta^\Delta$  est calculé à partir de  $r_\lambda^\Delta$  tel que décrit dans [38, 65]. Le processus commence à la racine de l'arbre :  $R_\theta^\Delta(\text{root})$  et  $q_\theta^\Delta(\text{root})$  sont initialisés à zéro. Ensuite, la fonction `ComputeDeltaNode` est appelée pour chaque nœud enfant de la racine. La fonction `ComputeDeltaNode` (voir algorithme 4) calcule  $r_\lambda^\Delta(\text{node})$  et le compare avec  $R_\theta^\Delta(\text{parent})$ .  $R_\theta^\Delta(\text{node})$  enregistre le maximum

entre ces deux valeurs :  $R_\theta^\Delta(node) = \max(r_\lambda^\Delta(node), R_\theta^\Delta(parent))$ .  $q_\theta^\Delta$  demande une attention particulière. Par définition, il doit prendre la taille de l'ouverture produisant le résidu maximal. Mais plusieurs ouvertures contribuent au calcul de  $R_\theta^\Delta$ . Quelle taille doit alors être choisie ? Si  $R_\theta^\Delta(parent)$  est supérieur à  $r_\lambda^\Delta(node)$ ,  $R_\theta^\Delta(parent)$  et  $q_\theta^\Delta(parent)$  sont propagés vers le nœud  $node$  (lignes 36 et 37 de l'algorithme 4). Sinon, si  $r_\lambda^\Delta(node)$  devient  $R_\theta^\Delta(node)$ ,  $q_\theta^\Delta(node)$  prendra sa valeur selon les cas suivants :

- $\kappa(node) + 1$ , si le nœud  $node$  n'appartient pas à une série d'intégration ( $is\Delta$  est false) ou si  $node$  est le premier nœud d'une série d'intégration quand le résidu maximal est produit ( $is\Delta$  est true mais  $q\_propagation$  est false ; lignes 42 et 43).
- $q_\theta^\Delta(parent)$ , jusqu'à la fin du processus d'intégration ( $is\Delta$  et  $q\_propagation$  sont tous les deux true ; ligne 37).

Ainsi,  $q_\theta^\Delta$  est mis à la valeur de la plus grande ouverture impliquée dans le processus d'intégration, quand un résidu maximal est produit. Cette taille correspond à la taille réelle de la région détectée.

---

**Algorithme 3** : Compute Delta Ultimate Attribute Opening.

---

```

23 ComputeDeltaUAO()
24 //Initialization
25  $R_\theta^\Delta(root) = 0$ 
26  $q_\theta^\Delta(root) = 0$ 
27  $q\_propagation = 0$ 
28 //Compute children
29 foreach  $child$  in  $Children(root)$  do
30   ComputeDeltaNode( $child, root, q\_propagation$ )

```

---

Un exemple de calcul de  $r_\lambda^\Delta$ , avec  $\Delta = 1$ , pour un profil de synthèse de la figure 1.40 est illustré sur la figure 1.41. Pour des raisons de simplicité, les nœuds de l'arbre sont labélisés avec leur niveau de gris (ce qui évidemment n'est pas le cas pour l'implémentation logicielle utilisée). La procédure comprend les étapes suivantes :

- le résidu du nœud racine (0) est initialisé à zéro :  $r_\lambda^\Delta(0) = 0$ .  $R_\theta^\Delta(0)$  et  $q_\theta^\Delta(0)$  sont mis à zéro ;
- ensuite, le nœud 3 est traité :  $r_\lambda^\Delta(3) = r_\lambda(3) = 3$ . Il n'y a pas d'intégration car  $\kappa(3) - \kappa(0) = 10 - 8 = 2 > \Delta$ .  $r_\lambda^\Delta(3) > R_\theta^\Delta(0)$ , alors  $R_\theta^\Delta(3) = r_\lambda^\Delta(3)$  et  $q_\theta^\Delta(3) = \kappa(3) + 1 = 9$  ;
- après ceci, le nœud 5 est traité :  $r_\lambda^\Delta(5) = r_\lambda(5) = 2$ . Pas d'intégration, car  $\kappa(5) - \kappa(3) = 8 - 6 = 2 > \Delta$ . Comme  $r_\lambda^\Delta(5) < R_\theta^\Delta(3)$ ,  $R_\theta^\Delta$  et  $q_\theta^\Delta$  sont propagés du père (nœud 3) vers le fils (nœud 5) :  $R_\theta^\Delta(5) = R_\theta^\Delta(3) = 3$  et  $q_\theta^\Delta(5) = q_\theta^\Delta(3) = 9$  ;
- quand on calcule  $r_\lambda^\Delta(7)$ ,  $r_\lambda(7)$  doit être ajouté à  $r_\lambda^\Delta(5)$  :  $r_\lambda^\Delta(7) = r_\lambda(7) + r_\lambda^\Delta(5) = 2 + 2$ , car  $\kappa(7) - \kappa(5) = 6 - 5 = 1 \leq \Delta$ . Ce résidu est plus grand que  $R_\theta^\Delta(parent)$ , alors  $R_\theta^\Delta(7) = r_\lambda^\Delta(7)$  et  $q_\theta^\Delta(7) = \kappa(7) + 1 = 6$ . La variable  $q\_propagation$  est activée pour le reste de la série d'intégration.

---

**Algorithme 4** : Compute  $\Delta$  Node.

---

```

31 ComputeDeltaNode(node, parent, q_propagation)
32 //Vérifier  $\Delta$  attribut
33  $is\Delta = (\kappa(parent) - \kappa(node)) \leq \Delta$ 
34 //Calcul résidu  $r_\lambda^\Delta(node)$ 
35 if ( $is\Delta$ ) then
36    $r_\lambda^\Delta(node) = t(node) - t(parent) + r_\lambda^\Delta(parent)$ 
37 else
38    $r_\lambda^\Delta(node) = t(node) - t(parent)$ 
39 //Calcul  $R_\theta^\Delta(node)$  et  $q_\theta^\Delta(node)$ 
40  $R_\theta^\Delta(node) = R_\theta^\Delta(parent)$ 
41  $q_\theta^\Delta(node) = q_\theta^\Delta(parent)$ 
42  $isMaxT = 0$ 
43 if ( $r_\lambda^\Delta(node) > R_\theta^\Delta(node)$ ) then
44    $isMaxT = 1$ 
45    $R_\theta^\Delta(node) = r_\lambda^\Delta(node)$ 
46   if not( $q_{propagation}$  and  $is\Delta$ ) then
47      $q_\theta^\Delta(node) = \kappa(node) + 1$ 
48 //Compute children
49 foreach child in Children(node) do
50   ComputeDeltaNode(child, node,  $isMaxT$ )

```

---

- cette intégration continue jusqu’au nœud 9 (les attributs ont des valeurs consécutives : de 6 à 3), donnant lieu à  $r_\lambda^\Delta(9) = 2+2+1+1 = 6$ . Ainsi,  $R_\theta^\Delta(8) = r_\lambda^\Delta(8) = 5$  et  $R_\theta^\Delta(9) = r_\lambda^\Delta(9) = 6$ .

**Publications associées** : **ConfI**{ [66], [44], [38], [67], [68], [69], [65].}, **ACLI**{ [70], [71], [46]}.

- [66] T. Retornaz, **B. Marcotegui**. Ultimate opening implementation based on a flooding process. *In the 12th International Congress for Stereology, ICS XII*, Septembre 2007, Saint-Etienne, France.
- [44] T. Retornaz, **B. Marcotegui**. Scene-text localization based on ultimate opening. *International Symposium on Mathematical Morphology ISMM'07*. October 10-13, 2007. Rio de Janeiro. Bresil.
- [38] J. Fabrizio, **B. Marcotegui**. Fast Implementation of the Ultimate Opening. *ISMM '09 : Proceedings of the 9th International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing 2009*. pp. 272–281. Groningen, The Netherlands.
- [67] J. Fabrizio, M. Cord, **B. Marcotegui**. Text Extraction from Street Level

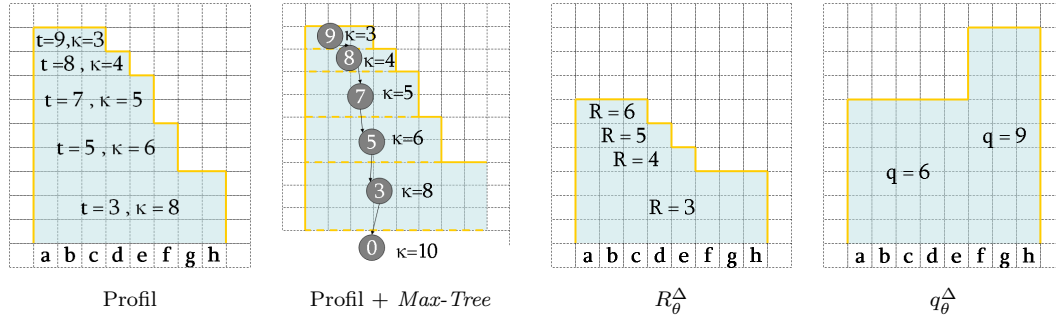


FIGURE 1.40 – (a) Profil avec niveaux de gris ( $t$ ) et attributs ( $\kappa$ ), (b) Max-Tree. (c)-(d)  $\Delta UAO$  résultant.

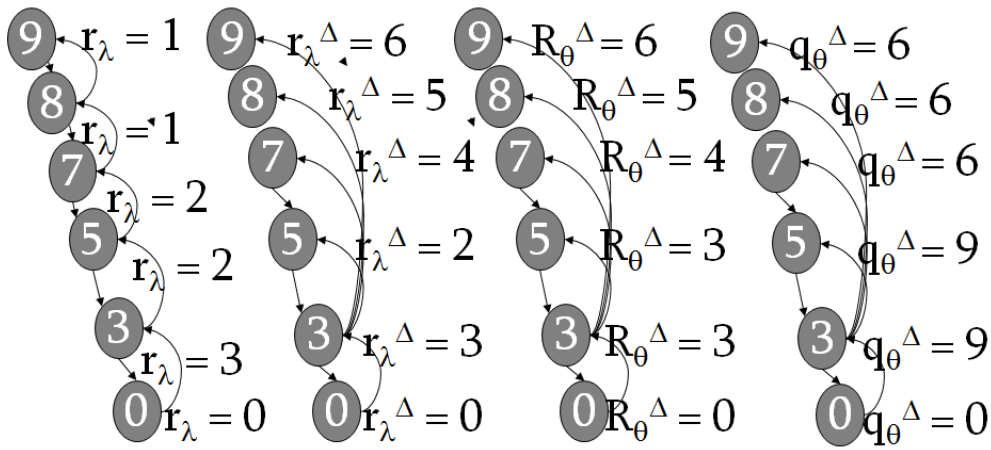


FIGURE 1.41 – Calcul de  $\Delta UAO$  sur le Max-tree. A gauche les résidus, au milieu les résidus  $\Delta$ , et à droite le résultat  $\Delta UO$  ( $R_\theta^\Delta$  et  $q_\theta^\Delta$  respectivement).

Images, *CMRT09 - CityModels, Roads and Traffic 2009*. Paris, France.

- [68] J. Fabrizio, **B. Marcotegui**, M. Cord. Text Segmentation in Natural Scenes Using Toggle-Mapping. *ICIP09 - IEEE International Conference on Image Processing 2009*. Cairo, Egypt.
- [69] R. Minetto, N. Thome, M. Cord, J. Fabrizio, **B. Marcotegui**. SNOOPERTEXT : A Multiresolution system for text detection in complex visual scenes. *ICIP10 - IEEE International Conference on Image Processing 2010*, Hong Kong.
- [65] **B. Marcotegui** and J. Hernández and T. Retornaz. Ultimate Opening and Gradual Transitions. ISMM'2011, *Proceedings of 10th International Symposium on Mathematical Morphology*, 6th-8th July 2011 - Intra, Lake Mag-

giore, Italy.

- [70] (JCR) J. Fabrizio, **B. Marcotegui**, M. Cord. Text Detection in Street Level Images. *Pattern Analysis and Applications*, Volume 16, Issue 4, pp 519-533 November 2013. doi=10.1007/s10044-013-0329-7,
- [71] E. Decencière, G. Cazuguel, X. Zhang, G. Thibault, J.C. Klein, F. Meyer, B. Marcotegui, G. Quellec, M. Lamard, R. Danno, D. Elie, P. Massin, Z. Viktor, A. Erginay, B. Lay and A. Chabouis. (2013). TeleOphta : Machine learning and image processing methods for teleophthalmology. IRBM, 34(2), 196-203.
- X. Zhang, G. Thibault, E. Decencière, B. Marcotegui, B. Lay, R. Danno, G. Cazuguel, G. Quellec, M. Lamard, P. Massin, A. Chabouis, Z. Victor, A. Erginay. "Exudate Detection in Color Retinal Images for Mass Screening of Diabetic Retinopathy". Medical Image Analysis, 2014.

## 1.7 Profils d'attributs

### 1.7.1 Décomposition par seuils et profil d'attributs

La décomposition par seuils est représentée sous forme de max-tree, chaque branche de l'arbre correspondant à l'évolution d'une composante connexe avec les différents seuils. En associant un attribut à chaque composante connexe de la décomposition, un profil d'attributs résume l'évolution des régions de chaque branche de l'arbre.

Les attributs sont classés en deux catégories : croissants et non croissants [72]. Un attribut est croissant si sa valeur pour une région donnée est plus grande ou égale que celle de n'importe quel sous-ensemble de cette région. La surface d'une région est l'attribut croissant le plus commun. Les attributs non croissants ne vérifient pas cette condition. La plupart des attributs de forme, tels que la circularité, la tortuosité et l'élongation, sont des attributs non croissants.

La figure 1.42 illustre la décomposition par seuils d'une image 1D, son max-tree et les profils de l'attribut de largeur ( $W$ ) pour les deux maxima de la fonction ( $p_A$  et  $p_B$ ). Des événements sur ces profils sont utiles pour segmenter des objets [73] ou extraire des descripteurs [37]. Si l'attribut est non croissant, sa valeur maximale est un événement significatif. Cette valeur maximale sera atteinte par des seuils différents à différents emplacements de l'image. Il s'agit donc d'une approche locale, qui maximise un attribut indépendamment de son contraste. En revanche, si l'attribut est croissant, la valeur maximale sera atteinte par le seuil minimal. D'autres événements significatifs seront ainsi à étudier dans le cas d'un attribut croissant. Par exemple, les régions extrémales de stabilité maximale (Maximally Stable Extremal Regions, MSER) introduites dans [8] détectent des régions dont la variation de surface est minimale.

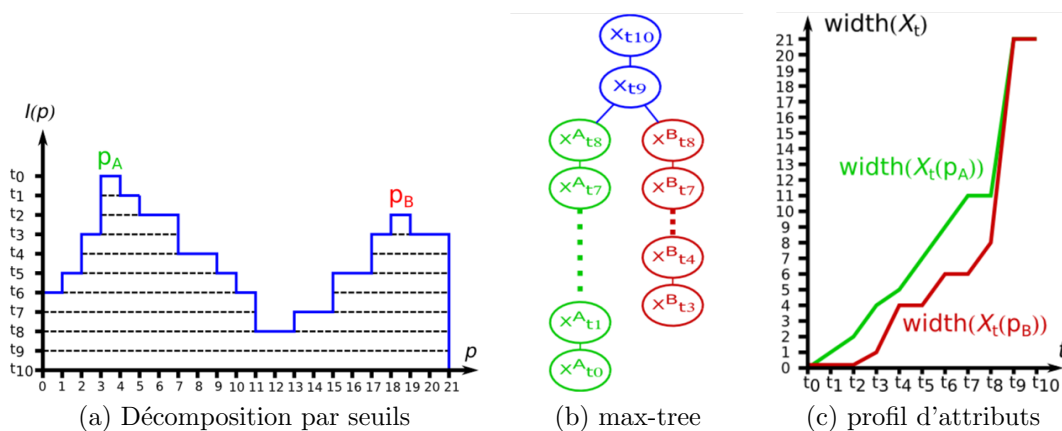


FIGURE 1.42 – Illustration des profils d'attributs. (a) Décomposition par seuils, (b) max-tree, et (c) profil de l'attribut largeur.

L'élongation géodésique  $E(X_t)$  d'un objet  $X_t$  est un descripteur de forme utile



pour caractériser les structures longues et fines. Il s'agit d'un attribut non croissant, proportionnel au rapport entre le diamètre géodésique au carré  $L^2(X_t)$  et la surface de l'objet  $A(X_t)$ , comme indiqué dans l'équation. 1.21.

$$E(X_t) = \frac{\pi L^2(X_t)}{4 A(X_t)} \quad (1.21)$$

Le diamètre géodésique  $L(X_t) = \sup_{x \in X_t} \{l_x(X_t)\}$  est la longueur de l'arc géodésique le plus long à l'intérieur de l'objet [74]. La figure 1.43 illustre le principe du diamètre géodésique.

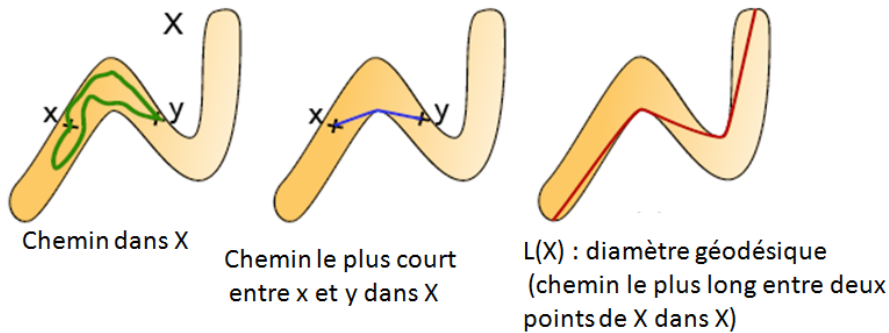


FIGURE 1.43 – Illustration du calcul du diamètre géodésique, qui intervient dans la définition d'élongation.

Les meilleurs algorithmes de calcul du diamètre géodésique ont une complexité élevée. Récemment, une bonne approximation de celui-ci, le diamètre barycentrique, a été proposée dans [75]. Non seulement elle donne en pratique des valeurs proches du diamètre géodésique, mais en plus elle peut être calculée à l'aide d'un algorithme dont la complexité est linéaire.

## 1.7.2 Image de caractéristique liée à un attribut

L'image caractéristique associe à chaque pixel de l'image la valeur de l'attribut maximal du profil d'attributs correspondant à sa branche. En d'autres termes, pour un pixel donné  $p$ , toutes les régions de la décomposition par seuils contenant  $p$  sont analysées. Le pixel  $p$  prend alors la valeur de l'attribut maximal de toutes ces régions.

La figure 1.44 montre un exemple d'image d'attribut maximal, en utilisant l'élongation géodésique comme attribut analysé. Nous observons que cette image fait ressortir les structures allongées.

Nous utilisons l'image d'élongation maximale pour segmenter les façades sur l'image d'élévation. L'image 1.45 montre un exemple de résultat. A gauche nous observons le nuage de points segmenté (la façade en bleu, le sol en gris et les objets 3D de la scène en

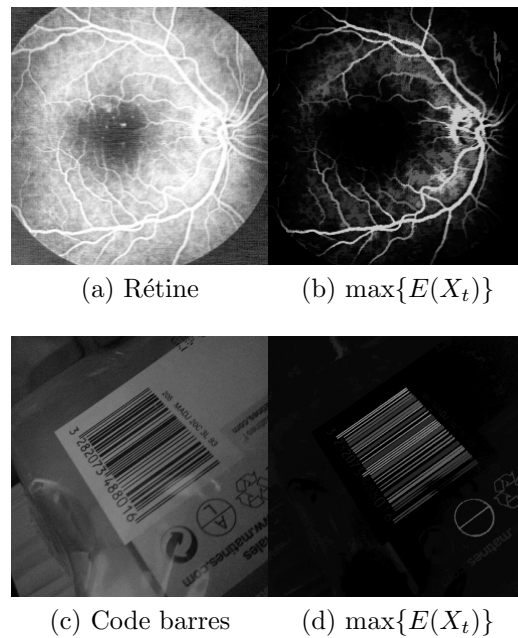


FIGURE 1.44 – Elongation maximale utilisant une décomposition par seuils.

jaune). A droite l'image couleur associée permet d'interpréter le contenu de la scène. Nous pouvons constater que des objets proches, comme les motos garées près de la façade, ne sont pas confondues avec celle-ci.

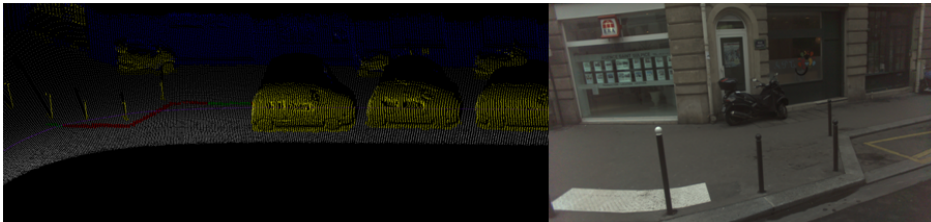


FIGURE 1.45 – Segmentation de façades (en bleu) en maximisant l'attribut d'élongation sur l'image d'élévation. Les objets proches à celle-ci, comme les motos, ne sont pas fusionnés.

### 1.7.3 Un nouvel attribut : l'élongation stable

Les filtres connexes utilisant un attribut d'élongation [75] ont tendance à reconnecter des structures parasites liées au bruit. Les régions extrema de stabilité maximale (Maximally Stable Extremal Regions, MSER) introduites dans [8] détectent des régions stables mais favorisent les régions arrondies[9]. Afin de détecter des objets allongés en

tenant compte de leur stabilité, nous proposons un nouvel attribut l'*élongation stable*, noté  $\Phi(X_t)$ . Cet attribut combine la stabilité d'une région  $\Psi(X_t)$  avec son élongation  $E(X_t)$  :

$$\Phi(X_t) = \Psi(X_t)E(X_t) = \frac{A(X_t)}{\frac{dA(X_t)}{dt}} \frac{\pi L^2(X_t)}{4 A(X_t)} = \frac{\pi L^2(X_t)}{4 \frac{dA(X_t)}{dt}} \quad (1.22)$$

Les maxima de  $\Phi(X_t)$  correspondent aux régions stables avec une élongation significative. L'image caractéristique en utilisant le maximum  $\Phi(X_t)$  fournit : 1) une partition de l'espace utile pour la segmentation ; 2) chaque pixel contient des informations sur la forme et la stabilité de son voisinage.

La figure 1.46 illustre le comportement de cette méthode sur une image réelle d'ADN. L'objectif est de segmenter la fibre allongée de l'arrière-plan bruitée. La figure 1.46b montre l'image d'élongation maximale où les objets de forme allongée ont été mis en évidence. Toutefois, des structures liées au bruit sont fusionnés dans le fond donnant lieu à des détections parasites, comme par exemple la structure poreuse au centre de l'image. Le maximum de stabilité (figure 1.46c) conserve également de nombreuses structures liées au bruit et non allongées dans le fond. Finalement, la figure 1.46d montre l'image d'élongation stable maximale, où la plupart des structures liées au bruit ont été éliminées en raison de leur faible stabilité.

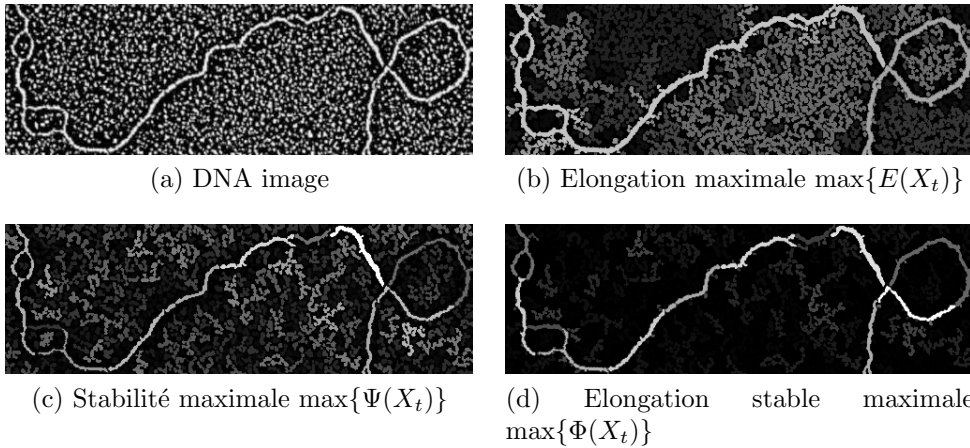


FIGURE 1.46 – Elongation maximale, stabilité maximale et élongation stable maximale image.

Cette méthodologie a été utilisée pour segmenter des cellules de la peau dans le cadre d'un projet industriel avec L'Oréal. L'image 1.47 illustre un résultat expérimental. La figure 1.47(a) illustre l'image originale et la figure 1.47(b) la vérité terrain correspondante. La figure 1.47(c) montre l'image d'élongation maximale,  $\max\{E(X_t)\}$ . Nous pouvons constater que tous les mélanocytes présentent une élongation significative mais le bruit produit des structures parasites dans le fond. La plupart des structures

liées au bruit ne sont pas stables (dans le sens MSER). Alors leur élongation stable  $\Phi(X_t)$  constitue un attribut adapté pour la segmentation de ce type d'objet.

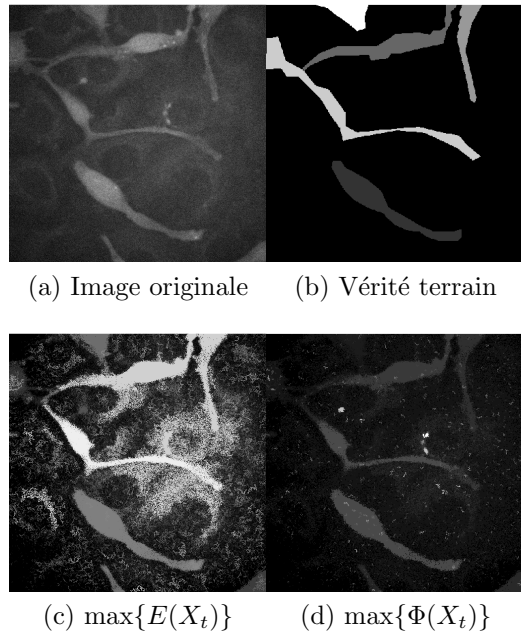


FIGURE 1.47 – Image caractéristique utilisant la valeur maximale du profil d'attributs.

## 1.8 Conclusion

Dans ce chapitre j'ai décrit mes contributions dans le domaine de la segmentation morphologique. Dans un premier temps, j'ai développé des outils de segmentation interactive, capables d'aborder la segmentation d'images génériques. L'approche proposée, repose sur une segmentation hiérarchique. Les outils d'interaction proposés permettent par exemple de choisir un niveau de résolution adapté au contenu de l'image ou de tracer un simple trait pour signaler un objet d'intérêt. La localisation précise des contours de l'objet signalé est alors confiée à l'ordinateur. Les moyens d'interaction proposés sont intuitifs et ne requièrent pas des compétences en traitement d'images. Une approche interactive constitue un bon compromis entre une segmentation purement manuelle, très laborieuse, et une approche complètement automatique, difficile dans un contexte général. Par ailleurs, la segmentation idéale peut ne pas être bien définie et dépendre de l'application. L'approche interactive combine la robustesse des méthodes manuelles avec l'efficacité des méthodes automatiques.

Dans un deuxième temps je me suis orientée vers les méthodes résiduelles du fait de leur caractère générique, multi-échelle et faiblement paramétrique. Ces méthodes extraient les régions les plus contrastées d'une image sans information a priori sur leur taille, forme ou contraste. Nous avons proposé l'ouverture ultime avec des ouvertures par attribut. Elles ont l'avantage d'être des opérateurs connexes qui préservent la forme des objets segmentés, simplifiant leur analyse ultérieure. Nous avons utilisé cet opérateur pour la détection de texte enfoui, qui a une forte variabilité de couleur, de taille et de forme et nous avons obtenu des résultats satisfaisants.

La confrontation de ces méthodes avec des cas réels a mis en évidence un certain nombre de problèmes. L'analyse de ces problèmes nous a conduit à la définition de nouveaux opérateurs proposant des solutions aux problèmes rencontrés. Par ailleurs, nous avons proposé un algorithme rapide, à l'aide du max-tree. Il permet l'utilisation de ces approches en temps réel pour des images de relative grande taille.

Finalement, nous avons exploré des techniques qui étudient l'évolution des attributs tels que l'élongation ou l'élongation pondérée par la stabilité. Nous avons montré l'intérêt de ces techniques pour la segmentation de façades et la segmentation de cellules de la peau pour le compte de L'Oréal.



# Chapitre 2

## Analyse de scènes urbaines 2006-

J'ai abordé l'analyse sémantique de scènes urbaines dans le cadre du projet collaboratif TerraNumerica, qui a été ensuite suivi par le projet TerraMobilita. Ce nouvel axe de recherche a été l'occasion de proposer des méthodes originales et efficaces d'analyse sémantique de façades ainsi que de nuages de points 3D, combinant une représentation sous forme d'image d'élévation et leur traitement efficace au moyen d'outils morphologiques.

### 2.1 Segmentation de façades (2006-2009)

Le but de TerraNumerica est d'augmenter le réalisme et la productivité de la modélisation urbaine. Dans ce contexte, j'ai développé en collaboration avec Jorge Hernández, dans le cadre de sa thèse, des méthodes de segmentation de façades de bâtiments haussmanniens. Nous présentons dans un premier temps les principes des techniques de l'état de l'art adressant ce problème. Ensuite nous identifions les points faibles et proposons des solutions.

Dans la littérature, les méthodes les plus robustes de segmentation de façades exploitent la structure régulière des bâtiments étudiés, avec les fenêtres bien alignées, aussi bien verticalement qu'horizontalement. Par exemple, Lee et Nevatia [76] et Wang et al [77] utilisent des profils 1D obtenus par intégration horizontale ou verticale de gradients directionaux. Les fenêtres alignées produisent des pics d'accumulation de gradient. Elles sont ainsi détectées par un simple seuil du gradient projeté. Muller [60] utilise l'information mutuelle pour trouver des structures architecturales répétitives sur une image de façade. Ces méthodes sont appliquées à des bâtiments isolés, ayant une unité architecturale. Cette première étape d'extraction d'une façade isolée est généralement réalisée manuellement.

Cependant, lorsque nous prenons une photo au niveau de la rue dans une ville, plusieurs problèmes apparaissent pour modéliser une seule façade :

1. variations de la résolution causées par des effets de perspective,

2. façades multiples dans la même image,
3. présence du ciel,
4. présence de la route et du trottoir,
5. détails architecturaux, balcons en fer forgé, branches d'arbres devant la façade, perturbant le caractère répétitif des éléments constitutifs.
6. rez de chaussée, considéré comme un problème en raison de structures non répétitives [60].

Ces problèmes sont illustrés sur l'image de la figure 2.1.



FIGURE 2.1 – Image acquise au sol, avec plusieurs bâtiments côte à côte de caractéristiques différentes ne pouvant faire l'objet d'une seule étape de modélisation, présence du ciel, de la route et du rez-de-chaussée.

Nous proposons une extraction automatique de la façade d'un seul bâtiment et son analyse sémantique tel qu'illustré sur la figure 2.2. Cette procédure, développée dans le cadre de la thèse de Jorge Hernández, est décrite dans les sections suivantes.

### 2.1.1 Segmentation de bâtiment

Premièrement, nous avons proposé une méthode automatique capable d'isoler les différentes façades de bâtiments d'une image acquise dans des conditions réalistes dans une ville. Cette segmentation facilite l'extraction d'information sémantique/grammaticale pour une modélisation procédurale efficace à grande échelle. La figure 2.3 montre un schéma de notre méthode. La première étape est la rectification d'image. Les lignes de fuite sont détectées, fondées sur la transformée de Hough [78]. Ensuite la division verticale des façades est réalisée. Finalement le ciel et le rez-de-chaussée sont supprimés.



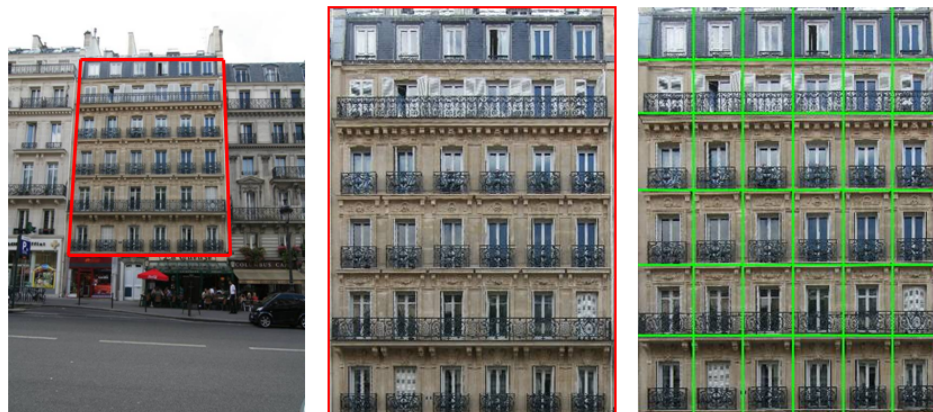


FIGURE 2.2 – Segmentation de façades de bâtiments haussmanniens : à gauche, segmentation d’un bâtiment, au milieu, zone d’intérêt extraite, avec un seul bâtiment à analyser, à droite, résultat de l’analyse.

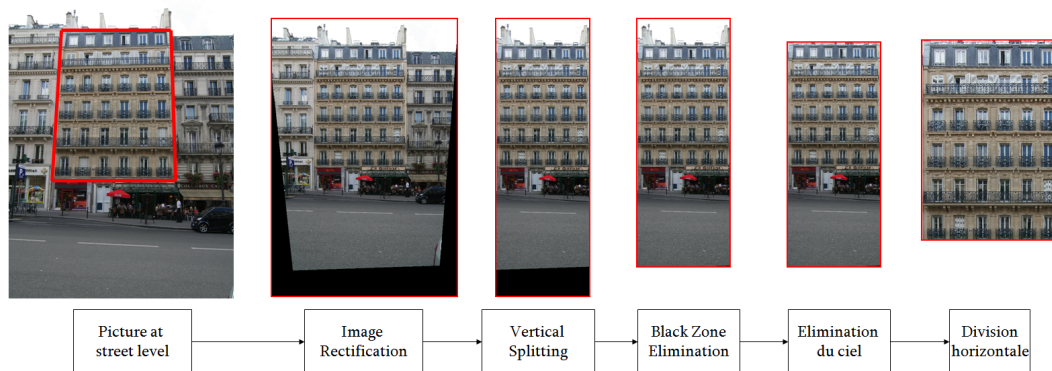


FIGURE 2.3 – Méthode de segmentation de façades de bâtiments. Image d’entrée : image acquise au niveau du sol dans une rue parisienne. Sortie : image faisant l’objet d’une procédure de modélisation.

La division verticale de façades est inspirée de l’approche de Lee et Nevatia [76]. Premièrement le gradient vertical est calculé, détectant des contours horizontaux (figure 2.4(a)). Avant intégration, des filtres morphologiques sont utilisés pour supprimer les détails non significatifs. Une ouverture morphologique avec un élément structurant horizontal est utilisé. La taille de cette ouverture est déterminée en fonction de la taille de la fenêtre la plus petite à détecter, si la résolution de l’image est connue. Sinon cette taille est déduite de l’analyse granulométrique du gradient, tel que nous avons proposé dans [79]. Nous observons une forte simplification de ce profil tout en préservant l’information utile. Les pics de ce profil se situent aux emplacements des fenêtres. Un filtre alterné séquentiel est alors appliqué sur ce profil. Finalement, la ligne de partage des eaux du profil inversé détermine les divisions verticales. Parmi ces divisions, certaines sont intra-façade (en vert) et d’autres inter-façades (rouge). Comme nous pouvons voir

sur la figure 2.4(c) les divisions inter-façades correspondent aux vallées les plus profondes. En effet, il n’y a pas de structures horizontales qui traversent la limite entre deux bâtiments. Afin de classer les deux types de division, un algorithme de k-means avec deux classes est utilisé. La classe associée à la valeur minimale correspond aux divisions inter-façades. Cet algorithme de k-means permet de choisir le bon nombre de divisions inter-façades sans besoin d’appliquer un seuil ni de connaître a priori le nombre de bâtiments présents dans l’image, à condition qu’il y ait plus d’un.

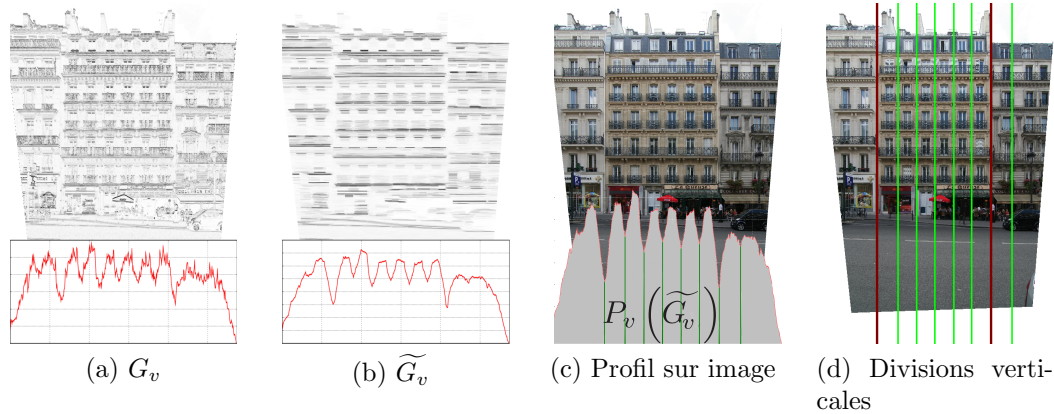


FIGURE 2.4 – (a) Gradient couleur vertical (élément structurant vertical, détectant des contours horizontaux) et sa projection. (b) Gradient filtré et sa projection. (c) Projection verticale  $P_v(\tilde{G}_v)$  sur l’image de départ. (d) Divisions verticales de la façade.

La segmentation horizontale est identique à la verticale, remplaçant simplement vertical par horizontal et vice-versa. La figure 2.5 illustre un exemple de cette procédure. Le gradient horizontal, filtré par une ouverture verticale, est projeté horizontalement. Ensuite un filtre alterné séquentiel suivi d’une ligne de partage des eaux détecte les différents étages du bâtiment. Une étape supplémentaire de raffinement est rajoutée pour positionner la division au niveau de la dalle : la division est déplacée au maximum du gradient vertical projeté horizontalement. En effet, le gradient vertical positionne de manière plus précise la division horizontale.

Afin d’éliminer le rez de chaussée et la section de route, nous comparons la projection totale ( $P_v(\tilde{G}_h)$ ), sur toute l’image et la projection locale ( $P_v(\tilde{G}_h(\text{étage}))$ ), sur chaque étage. Ces projections sont normalisées pour pouvoir être comparées. A noter que la différence entre la projection locale et la projection globale est inférieure pour un étage de fenêtres que pour le rez-de-chaussée ou la route (Fig. 2.6). Nous utilisons à nouveau l’algorithme k-means pour classer les projections locales (par étage) en deux catégories : celles qui ressemblent à la projection globale et celles qui en sont plus éloignées. Les sections sont alors supprimées du bas vers le haut, jusqu’à trouver une section de la classe proche de la projection globale. Dans cet exemple, deux sections sont supprimées.

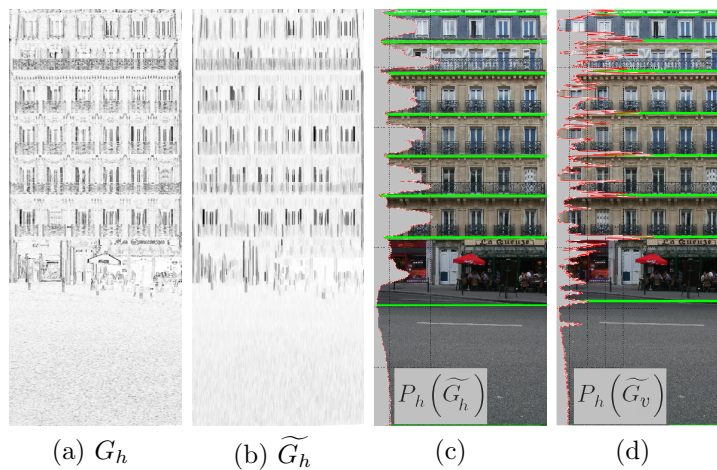


FIGURE 2.5 – (a) Gradient couleur horizontal et (b) gradient filtré. (c) Projection horizontale de  $\widetilde{G}_h$  et division horizontale de façade, (d) Projection horizontale de  $\widetilde{G}_v$  et division horizontale raffinée de façade.

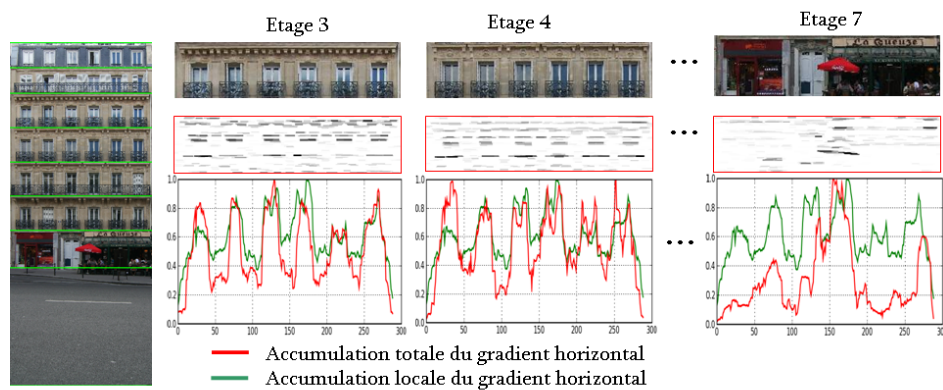


FIGURE 2.6 – Analyse des sections horizontales.

L'image 2.7 montre d'autres exemples d'images ainsi que les bâtiments isolés automatiquement à partir de ces images.

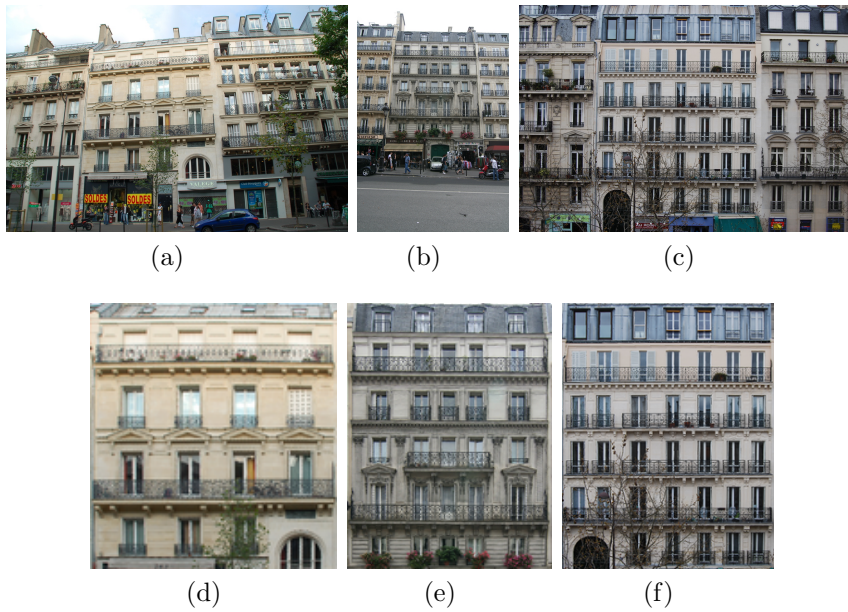


FIGURE 2.7 – (a)-(c) Images de façades acquises en conditions réalistes. (d)-(f) bâtiments isolés automatiquement à partir des images (a)-(c).

### 2.1.2 Segmentation verticale et horizontale

Une fois la façade isolée, la segmentation verticale et horizontale est appliquée, utilisant la même technique décrite dans la section précédente. Le gradient vertical, filtré par une ouverture horizontale, est accumulé verticalement. L'inclusion du filtre morphologique avant l'accumulation permet de mieux gérer les façades texturées. Un filtre alterné séquentiel est ensuite appliqué sur le profil d'accumulation. Finalement une ligne de partage des eaux est appliquée au profil inversé. La même procédure est appliquée horizontalement. La figure 2.8 illustre les résultats sur plusieurs images, y compris pour des cas où un arbre cache une partie importante de la façade.

### 2.1.3 Localisation de fenêtres

Une fois la division de la façade effectuée, la localisation précise des bords de fenêtres est réalisée de la manière suivante : pour les bords verticaux, les extrema de  $P_v(\widetilde{G}_v)$  sont utilisés comme marqueurs. La figure 2.9 illustre cette procédure. Les maxima (bleu) sont les marqueurs des fenêtres et les minima (vert) les marqueurs du mur de

la façade. Ces marqueurs sont introduits dans  $P_v(\widetilde{G}_h)$  pour la localisation précise des bords de fenêtre verticaux. De manière analogue les contours horizontaux sont obtenus, remplaçant horizontal par vertical et viceversa.

D'autres résultats de détection de fenêtres sont illustrés sur la figure 2.10.

### 2.1.4 Résultats sur la base Teboul

L'analyse sémantique de façades a donné des résultats satisfaisants sur les images issues du projet TerraNumerica. Nous avons cherché également à nous comparer avec d'autres techniques de l'état de l'art. Pour ceci, nous avons testé la performance de notre méthode sur la base publique introduite par Teboul et al. [80]. Dans [79] nous proposons de régler automatiquement les paramètres de notre méthode. Nous choisissons la taille de filtre qui minimise l'écart type des tailles de division des façades. Le réglage adaptatif des paramètres offre une robustesse au bruit, aux changements de résolution de l'image, des ombres et des textures. Ces filtres adaptatifs conduisent à la meilleure performance par rapport à des paramètres de filtre testés de façon exhaustive. Ainsi, notre approche est validée. Les résultats que nous obtenons sont  $P = 0,82$ ,  $R = 0,79$  et  $f_{mean} = 0.80$ , ce qui est beaucoup mieux que d'autres résultats rapportés dans la littérature sur cette même base. En effet, les résultats publiés par Teboul et al sont  $P = 0.58$ ,  $R = 0.81$  and  $f_{mean} = 0.68$  [80]. Cependant, leur approche nécessitant une phase d'apprentissage, ils donnent leurs résultats sur seulement 10 images. alors que nous donnons les performances sur l'ensemble de la base. Les résultats qualitatifs sont présentés sur les figures 2.12 et 2.11. La figure 2.12 présente des exemples dans lesquels la méthode proposée échoue. Ces images ne respectent pas l'hypothèse de régularité sur laquelle se fonde notre approche : certaines colonnes de la fenêtre sont presque adjacentes. L'écart-type de la largeur de division est plus petite lorsque les colonnes sont fusionnées que quand elles sont séparées. La figure 2.11 illustre la robustesse de notre système pour les ombres, les textures et les images sur lesquelles la distance entre les fenêtres est pseudo-régulière.

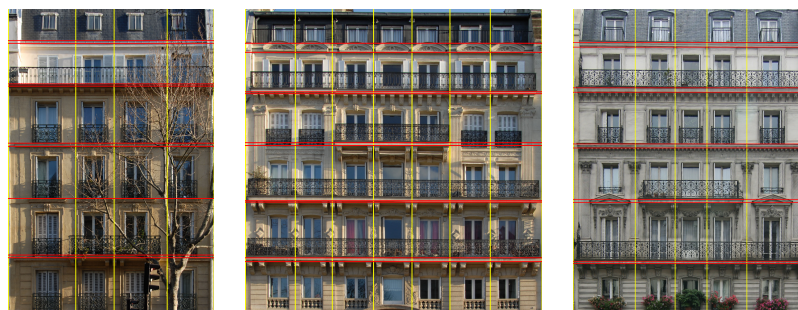


FIGURE 2.8 – Division de la façade en étages + dalle et en travées.

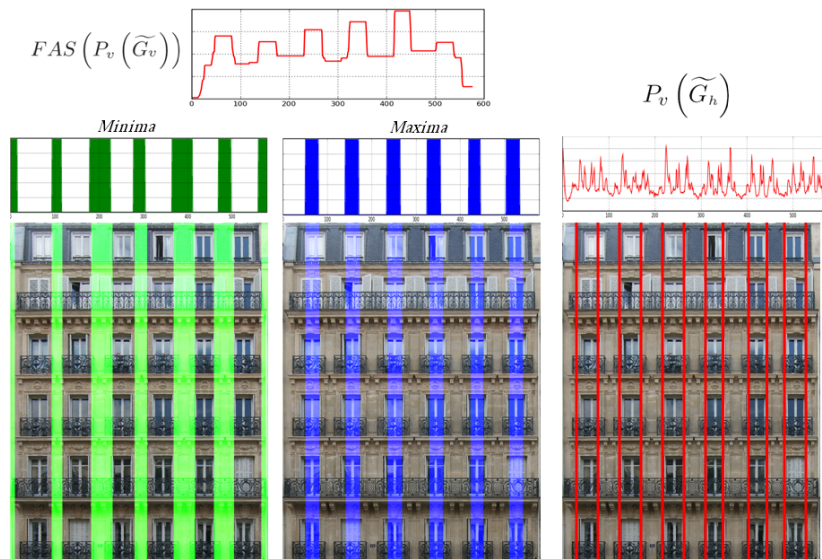


FIGURE 2.9 – Localisation de la bordure verticale des fenêtres.



FIGURE 2.10 – Localisation de fenêtres.

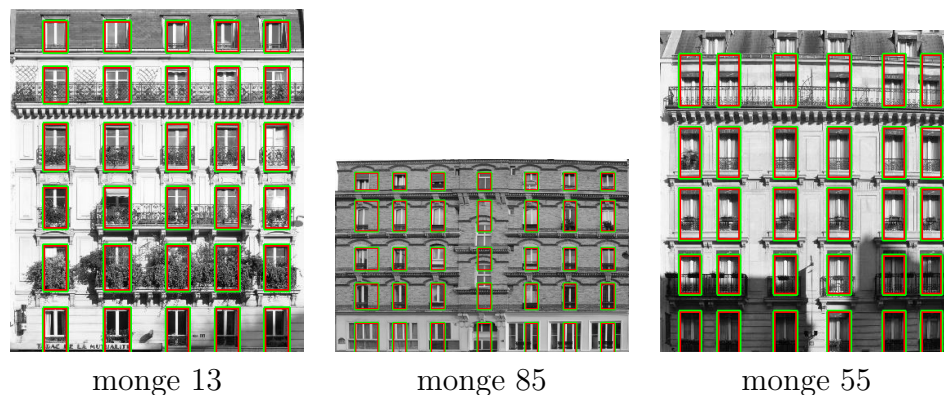


FIGURE 2.11 – Exemples de résultats satisfaisants.

Les bons résultats de ce projet ont permis d'obtenir un financement pour la suite des travaux, dans le cadre du projet TerraMobilita.

**Publications associées :** **ConfI**{ [81], [82], [83], [45], [79], [65].}, **ACLI**{ [84] }.

- [81] J. Hernández and B. Marcotegui. Ultimate Opening Segmentation with Shape Constraints. The 8th IASTED International Conference on Visualization, Imaging, and Image Processing 2008, 630-120, September 1-3, 2008, Palma de Mallorca, Spain.
- [83] J. Hernández and B. Marcotegui. Morphological Segmentation of Building Facade Images. ICIP09 - IEEE International Conference on Image Processing 2009. Cairo, Egypt.
- [45] J. Fabrizio, B. Marcotegui. Fast Implementation of the Ultimate Opening. ISMM09 : Proceedings of the 9th International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing 2009. pp. 272-281. Groningen, The Netherlands.
- [79] A. Serna, J. Hernández and B. Marcotegui. Adaptive Parameter Tuning For Morphological Segmentation of Building Façade Images. Proceeding of 20th European Signal Processing Conference (EUSIPCO 2012), August 27-31 2012, Bucharest, Romania. pp. 2268-2272
- [65] 5. B. Marcotegui and J. Hernández and T. Retornaz. Ultimate Opening and Gradual Transitions. ISMM'2011, Proceedings of 10th International Symposium on Mathematical Morphology, 6th-8th July 2011 - Intra, Lake Maggiore, Italy.
- [84] J. Hernández and B. Marcotegui. Shape Ultimate Attribute Opening. Image and Vision Computing. Image Vision Computing, vol. 29, n. 8. pp : 533-545. July 2011.

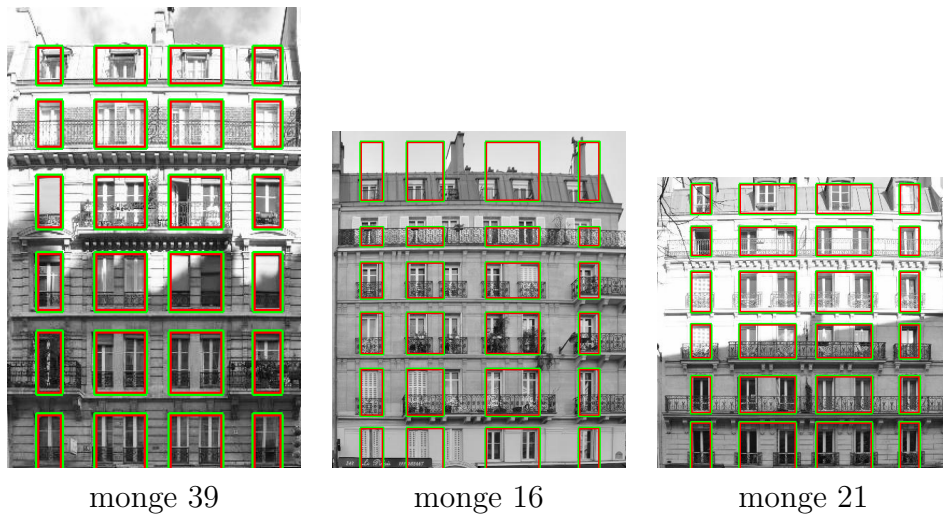


FIGURE 2.12 – Exemples de résultats problématiques.



FIGURE 2.13 – Détection de ciel et lignes de cadrage de la zone intérêt.



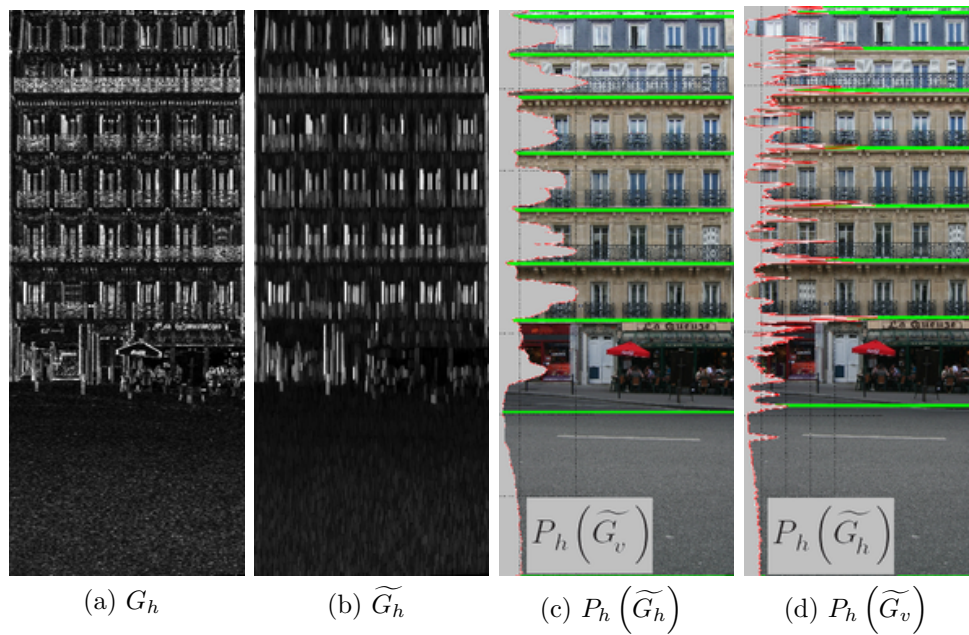


FIGURE 2.14 – (a) Gradient couleur horizontal. (b) Gradient filtré. (c) Projection horizontale de  $\widetilde{G}_h$  et division horizontale de la façade. (d) Projection horizontale de  $\widetilde{G}_v$  et raffinement de la division horizontale de la façade.

## 2.2 Analyse de nuages de points. TerraNumerica (2006-2009) et TerraMobilita (2011-2014)

Les systèmes d'acquisition d'images ne cessent de se diversifier. Depuis plusieurs décennies sont apparus les scanners 3D. Ils balayent l'espace avec un faisceau et fournissent une représentation des objets présents dans la scène sous forme de nuages de points 3D. Les scanners actuels, fondés sur une technologie de temps de vol, sont capables d'acquérir jusqu'à 1 million de points par seconde, avec une précision de l'ordre du centimètre, sur une portée de plus de 100 m. Ces systèmes peuvent être embarqués sur des véhicules équipés de dispositifs de positionnement précis, permettant une acquisition dense et précise à grande échelle. Des villes entières peuvent ainsi être numérisées en quelques jours.

De nombreuses villes au monde disposent de plans 2D très détaillés des rues et des espaces publics. Ces plans contiennent des informations de l'ensemble du mobilier urbain (lampadaires, panneaux de signalisation, abribus, entre autres) ainsi que les bords de trottoir, les arbres ou les barrières. Les scanners 3D permettent une acquisition tridimensionnelle des villes et ouvrent des perspectives très intéressantes. Le canton de Genève a été pionnier dans l'intégration de la 3D dans la cartographie du territoire. En 2010 il s'est doté d'un SIG 3D, décrivant de manière fine l'environnement urbain, comprenant le bâti, les infrastructures de communications, la végétation, le mobilier urbain, le modèle de terrain et les ouvrages d'art. La 3D permet un rendu visuel de la ville qui simplifie la compréhension des projets urbains et mène à une meilleure concertation entre la population et les élus. Au delà d'une navigation immersive pour une meilleure compréhension de l'environnement, la 3D offre la possibilité de prendre des mesures à l'aide d'une interface appropriée (telle que celle développée dans le cadre d'itowns, <http://www.itowns.fr/>) sans avoir à se rendre sur place, de réaliser des diagnostics liés à la mobilité, au bruit, entre autres, et facilite les expertises de l'administration dans le domaine de l'énergie (calcul des volumes chauffés, exposition d'une façade au soleil,...). Les collectivités territoriales s'intéressent aujourd'hui à l'évolution de cette thématique et prévoient une transformation de la gestion des villes en s'appuyant sur ces nouvelles technologies.

L'objectif de TerraMobilita est de mettre au point de nouveaux processus automatisés de création et de mise à jour de cartes 3D de voirie urbaine. Le but est non seulement de réduire le temps et les coûts, mais aussi de permettre une mise à jour plus fréquente des données. Plus celles-ci seront récentes plus elles seront fiables, et plus elles pourront être réutilisées dans différentes applications, et ceci, de manière plus pertinente. L'utilisation de la 3D permet d'évaluer les pentes et les dévers (qui sont critiques dans certains cas de mobilité, comme par exemple le vélo) et d'en tenir compte pour les calculs d'itinéraires adaptés à différents types de mobilité. La 3D permet de cartographier les obstacles, par exemple de prévoir la largeur critique de passage pour les fauteuils roulants. Dans le contexte de la loi de 2005, sachant que les villes doivent être aux normes pour 2015, la possibilité de faire des diagnostics automatisés d'accessibilité,

en particulier pour les Personnes à Mobilité Réduite (PMR) à court terme, est un vrai sujet d'actualité.

Les nuages de points bruts suffisent pour permettre une navigation immersive. Par contre, une interprétation sémantique de ces nuages est nécessaire pour développer d'autres applications. Par exemple, pour cartographier le mobilier urbain le nuage de points doit être nettoyé de tout objet mobile : piétons, voitures, vélos,... L'analyse sémantique des nuages de points permettrait le développement de nombreuses applications. Cette analyse est encore aujourd'hui un challenge scientifique. Les méthodes semi-automatiques existantes ne sont pas suffisantes pour un déploiement à grande échelle. C'est en ce sens que des méthodes automatiques pour l'analyse urbaine sémantique en 3D sont nécessaires et c'est pourquoi nous nous y sommes intéressés.

Le reste du chapitre est organisé de la manière suivante : la section 2.2.1 discute des structures de données existantes pour analyser les nuages de points non structurés. Ensuite, nous détaillons la génération des images d'élévation dans la section 2.2.2, l'élimination des points redondants dans la section 2.2.3 et la méthode d'interpolation pour mieux gérer les zones occultées dans la section 2.2.4. Les sections suivantes décrivent les différentes contributions que nous avons proposées sur l'analyse sémantique des nuages de points : la segmentation en blocs de maisons (section 2.2.5), la segmentation du sol (section 2.2.6), la segmentation d'objets 3D (section 2.2.7) et leur classification (section 2.2.8) et finalement le diagnostic d'accessibilité (section 2.2.9).

### 2.2.1 Structures de données

Analyser un nuage de points est délicat car il n'y a pas de relations topologiques prédéfinies entre ceux-ci. Des nombreuses structures de données ont été utilisées pour le traitement de nuages de points : le maillage, l'octree [85], le k-d tree [86], les points à une distance inférieure à un certain rayon sont considérés voisins, ce rayon pouvant être adaptatif [87], le point structurant [88], les images d'élévation [89, 90]. Nous avons choisi de travailler sur les images d'élévation. Les raisons sont les suivantes :

- la surface recouvrant une zone contient l'information nécessaire pour analyser la quasi-totalité des objets présents sur la scène urbaine. Seulement les objets sous un objet en hauteur (un piéton sous un arbre, par exemple) peuvent être occultés ;
- la projection constitue une réduction d'information, donnant lieu à des méthodes efficaces en temps de calcul ;
- les relations de voisinage entre points deviennent implicites ;
- des nombreuses techniques de filtrage et segmentation efficaces sont disponibles dans l'état de l'art pour les images résultantes.

### 2.2.2 Génération des images d'élévation

Les images d'élévation sont des images à deux dimensions dont la valeur du pixel contient une information relative à la troisième dimension. On les appelle parfois des images à 2,5 dimensions. Elles sont générées par une projection orthographique, chaque point prenant comme valeur la distance au plan de projection. Ce plan de projection est le plan horizontal, avec une normale  $\vec{n} = (0, 0, 1)$  et contenant le point  $(0, 0, z_{min})$ . Le seul paramètre à définir lors de cette projection est la taille du pixel de l'image résultante. Ce paramètre doit être choisi en fonction de la résolution des données. Si la taille du pixel est trop grande, plusieurs points seront projetés sur le même pixel, avec une éventuelle perte d'information. En revanche, si le pixel est trop petit la connexité des pixels requise par nos méthodes ne sera pas assurée. Par ailleurs, la taille de ce pixel a un impact direct sur le temps de calcul de nos méthodes.

Lors de la projection, plusieurs points 3D peuvent être projetés sur le même pixel, en particulier pour les objets verticaux comme par exemple les façades. Plusieurs informations peuvent être conservées :

- l'image d'élévation maximale ( $f$ ), ou simplement nommée image d'élévation, contient la distance du point le plus éloigné du plan de projection. Cette image contient la surface de la scène du point de vue nadir ;
- l'élévation minimale ( $f_{min}$ ), contient la distance la plus petite des tous les points 3D projetés sur le même pixel. Cette image contient le sol ou le point le plus bas des objets ;
- image de hauteur relative ( $f_{height} = f - f_{min}$ ), contient la différence entre l'image maximale et minimale. Cette image permet d'estimer la hauteur des objets indépendamment de l'éventuelle pente de la scène. Notez cependant que cette image ne contient pas la hauteur de tous les objets car le sol n'est pas toujours visible sous les objets ;
- l'image d'accumulation ( $f_{acc}$ ), contient le nombre de points 3D qui se projettent sur chaque pixel. Cette image est très utile pour détecter les structures verticales de la scène. Elle est utile aussi pour détecter les arrêts du véhicule d'acquisition aux feux rouges, par exemple. Nous reviendrons sur ce point plus tard.

La figure 2.15 montre un exemple de projection d'un nuage de points (a) : élévation (b), hauteur (c) et accumulation (d). Dans ce cas, le nuage de points a une résolution moyenne d'environ 5[cm]. Nous observons que les valeurs les plus élevées se trouvent sur les façades, les structures les plus hautes où la plupart des points sont projetés.

L'exemple présenté est un tronçon de rue de dimensions XY égales à 103.5[m] × 48.4[m], et donc les images produites ont une taille de 2070[pix] × 968[pix]. Dans le cas réel, les systèmes d'acquisition fournissent des nuages de points de plusieurs centaines de mètres. Une étape de découpage préalable à partir des images d'élévation en blocs de maisons est mise en place.

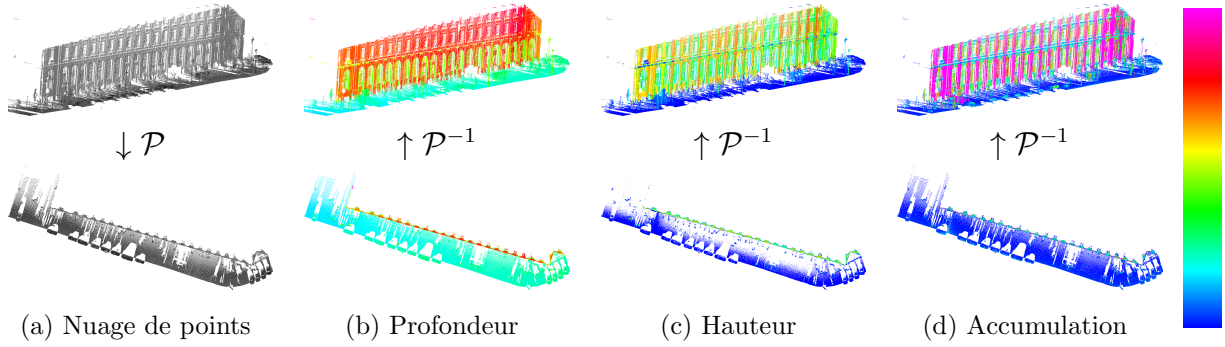


FIGURE 2.15 – (a) Nuages de points - vue 3D et vue supérieure. Projection sur images (code couleurs au rang) : (b) la valeur maximale, (c) la différence max – min et (d) l’accumulation . Données ©IGN Tronçon de la Rue Soufflot (Paris) côté pair.

### 2.2.3 Filtrage de points redondants

La densité de nuages de points dépend des spécifications techniques des capteurs, de la distance capteur-objet et de la vitesse du véhicule lors de l’acquisition. Lorsque le véhicule va lentement ou quand il s’arrête à cause d’un feu par exemple, la densité de points augmente puisque les lignes de balayage se chevauchent, ce qui donne de grandes valeurs sur l’image d’accumulation.

Afin de réduire les données redondantes, nous proposons une stratégie de filtrage en s’appuyant sur la façon dont chaque profil est acquis. Le capteur tourne pour acquérir chaque profil. Labelisant chaque profil, il est possible de déterminer quand différents profils se superposent. Quand des points de deux ou plus de profils se projettent sur le même pixel, seulement les points du profil majoritaire sont considérés.

Illustrons cette stratégie de filtrage sur un exemple. Considérons l’acquisition d’un véhicule à l’arrêt dans l’intervalle  $[t_0, t_2]$  (2.16). Les trois profils  $f_{t=t_0}(x)$ ,  $f_{t=t_1}(x)$  et  $f_{t=t_2}(x)$  se superposent (voir figure 2.16a, 2.16b et 2.16c). La scène contient les objets labélisés de ① à ④ : ① poteau, ② piéton, ③ piéton et ④ façade.

Les objets ① et ④ sont des objets fixes, présents sur les trois profils. Leur accumulation est ainsi trois fois plus grande que l’accumulation d’un seul profil. Les objets ② et ③ sont par contre mobiles et apparaissent sur certains profils. Leur accumulation  $f_{acc}$  ne dépend pas que de leur géométrie, mais également du temps qu’ils nécessitent pour traverser le rayon laser (2.16d). En choisissant pour chaque pixel le profil le plus représentatif nous obtenons une information d’accumulation  $f_{acc}$  plus pertinente.

La figure 2.17 illustre ce phénomène sur des données réelles, acquises par l’IGN sur la place *St. Sulpice* à Paris. Nous observons des valeurs d’accumulation très élevées lorsque le véhicule prend le tournant ou quand il s’arrête. La figure 2.17b montre l’élévation maximale et 2.17c la labélisation des profils pour détecter la superposition de plusieurs profils. En utilisant la stratégie décrite ci-dessus, développée dans le cadre de la thèse d’Andrés Serna, nous observons des valeurs d’accumulation plus fiables (voir

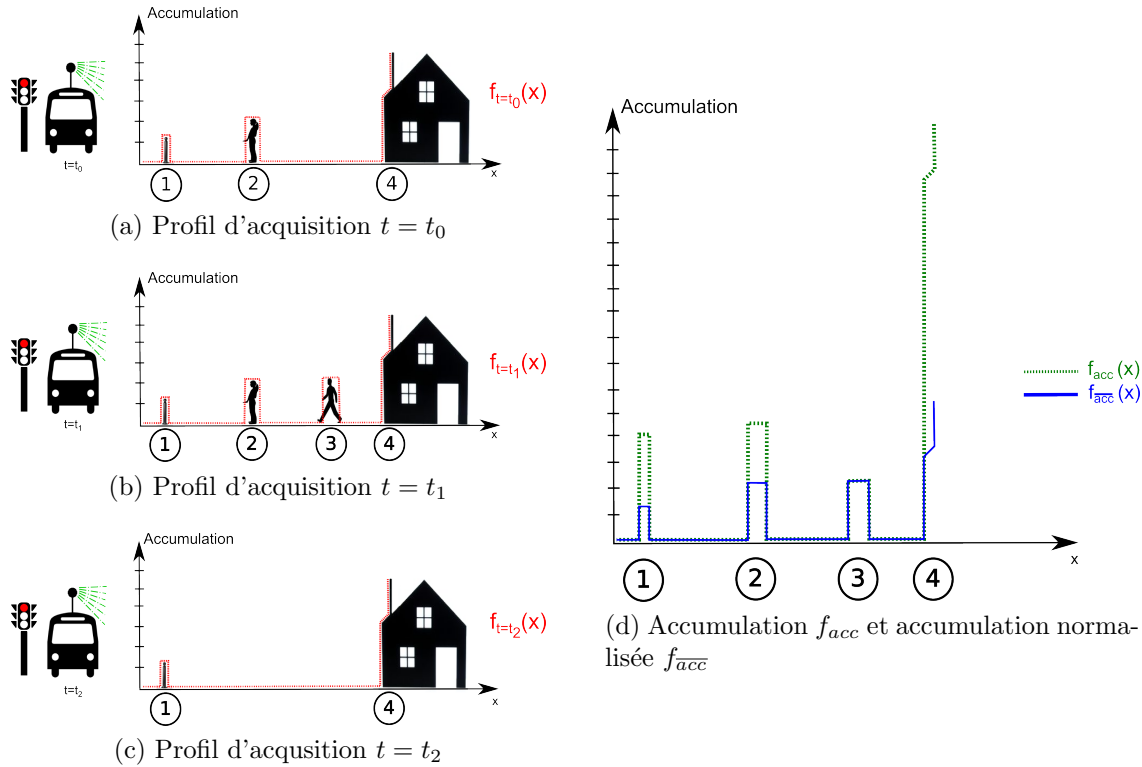


FIGURE 2.16 – Filtrage des points redondants, choisissant pour chaque pixel le profil de plus grande accumulation.

figure 2.17e).

## 2.2.4 Interpolation

L'image d'élévation contient des zones sans information dans les zones d'ombre du capteur. Ces zones peuvent créer des problèmes de déconnexion lors de l'analyse ultérieure. Nous avons proposé une méthode d'interpolation morphologique utilisant la transformation de remplissage de trous [91]. Cette transformation est choisie car elle ne requiert pas de paramètres, remplit des zones d'occlusion de n'importe quelle taille et ne crée pas de nouveaux extrema parasites, qui perturberont les étapes postérieures de la procédure de segmentation.

Illustrons la procédure d'interpolation avec un exemple. La figure 2.18 montre un profil typique d'acquisition. Il contient 7 objets : ① voiture, ② piéton, ③ bruit, ④ chien, ⑤ piéton, ⑥ façade, et ⑦ cheminée. Notez que la procédure est illustrée sur un profil à une dimension mais elle est appliquée à la totalité de l'image d'élévation à deux dimensions. La transformation de remplissage de trous, remplira chaque trou (vallée ne touchant pas le bord de l'image) avec la valeur minimale entourant la vallée. Par

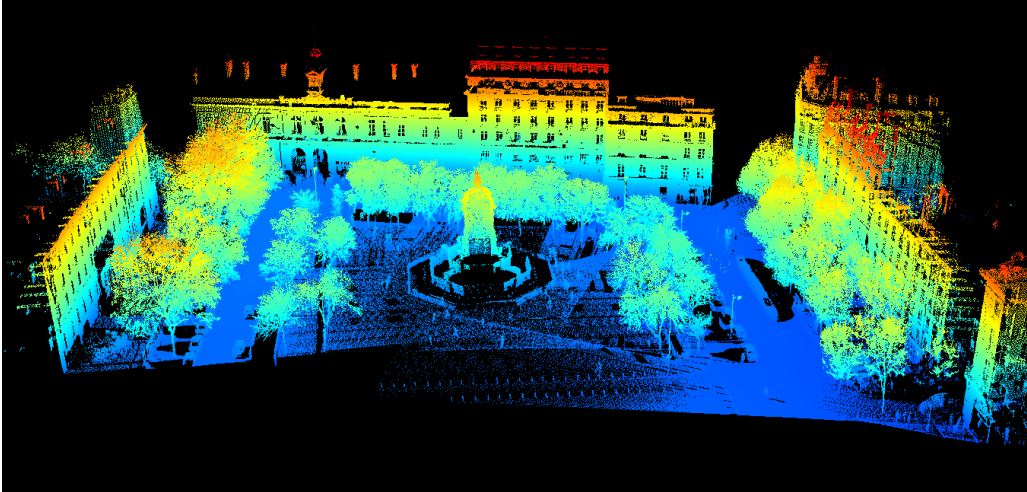
exemple, le trou de la partie gauche, entre les objets ③ et ④ sera rempli à la hauteur du sol. Par ailleurs le trou entre les objets ② et ③, et celui entre les objets ⑤ et ⑥ seront également remplis avec la valeur du sol. En effet, même si sur le profil 1D, ces trous ne touchent pas le sol, sur l'image 2D ils sont connectés aux profils voisins du sol. Les piétons ② et ⑤ n'occulent pas le sol sur les profils voisins (avant ou après eux).

Appliquons maintenant cette méthode sur des exemples réels. Trois jeux de données sont illustrés sur la figure 2.19b. Nous pouvons apprécier sur la figure 2.19b, que beaucoup de données manquantes n'ont pas pu être interpolées. Ceci est dû aux problèmes suivants :

- dans le cas des données MENSİ, les données du sol sont de petites régions isolées et éparpillées.
- dans le cas des données de systèmes mobiles, pour une scène avec beaucoup d'obstacles (par exemple des voitures garées) ou de grands obstacles (des lampadaires), les trous sont connectés au bord de l'image.

Pour résoudre ce problème, nous proposons de relier les régions, en utilisant le chemin de plus courte distance entre elles [92, 93]. Autrement dit une région sera reliée à sa voisine la plus proche. Cette technique ne garantit pas le remplissage de tous les trous, mais en pratique donne des bons résultats. Les connexions rajoutées sont illustrées dans la figure 2.19c. Le remplissage de trous sur cette image réussit cette fois-ci à combler les trous dus aux occultations (voir 2.19d).

Cette méthode d'interpolation, mise en place dans le cadre de la thèse de Jorge Hernández, est rapide, interpole des trous de n'importe quelle taille sans information a priori. En revanche, le remplissage se fait par des zones plates, ce qui peut donner des résultats peu réalistes sur des données en pente comme ceux de l'image 2.20.



(a) Nuage de points 3D coloré selon la coordonnée Z

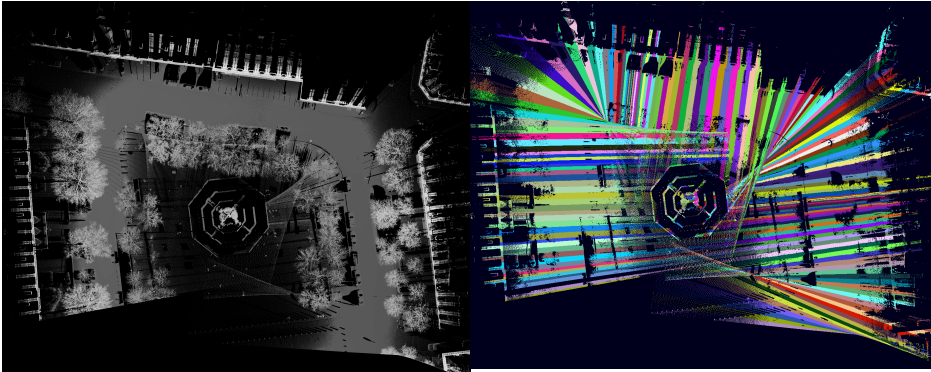
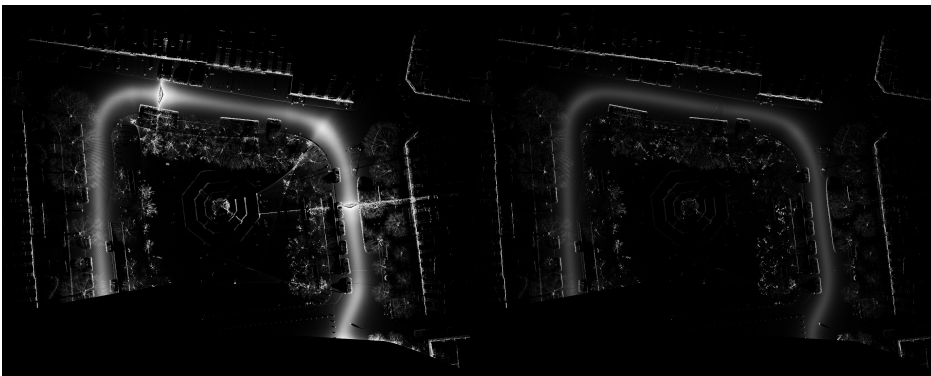
(b) Image d'élévation maximale  $f$  (c) Labelisation de chaque profil d'acquisition(d) Image d'accumulation  $f_{acc}$  (e) Filtrage d'information redondante. Cette image est nommée image d'accumulation normalisée  $f_{acc}$ 

FIGURE 2.17 – Filtrage d'information redondante reposant sur l'information d'accumulation. Place *St. Sulpice* à Paris, France. Données acquises par Stereopolis II, IGN©France.



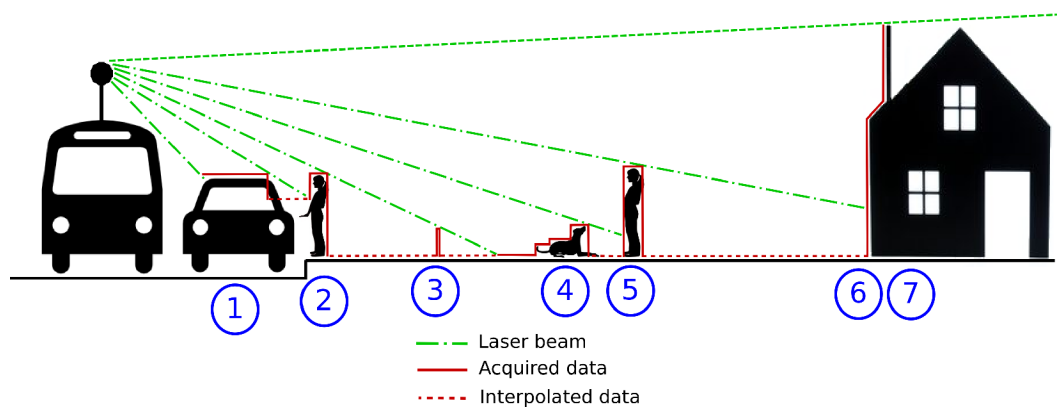


FIGURE 2.18 – Schéma d’acquisition et méthode d’interpolation sur un profil 1D. Le profil urbain contient sept objets : ① voiture, ② piéton, ③ bruit, ④ chien, ⑤ piéton, ⑥ façade, et ⑦ cheminée.

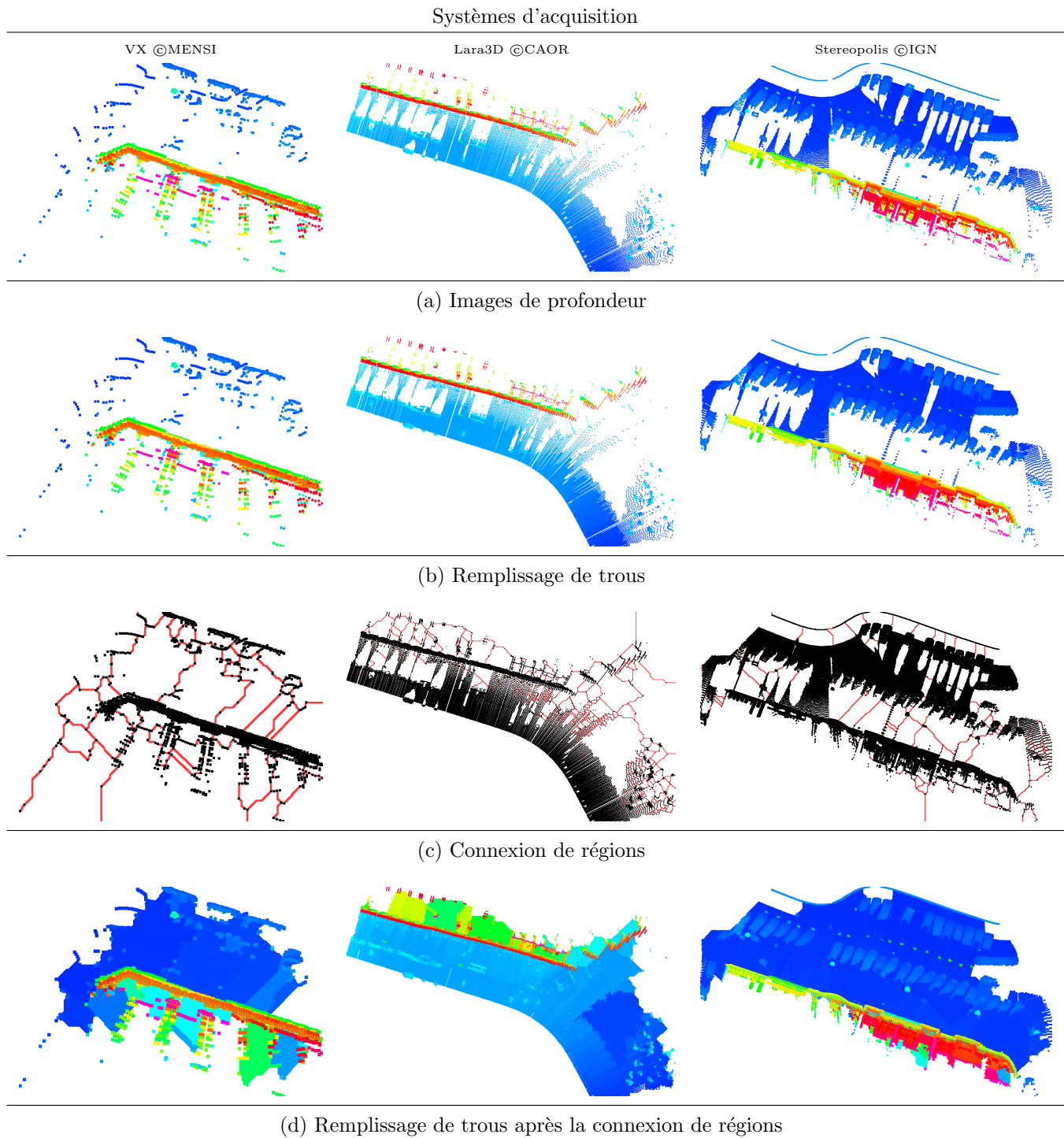


FIGURE 2.19 – Estimation des pixels manquants en utilisant le remplissage de trous.

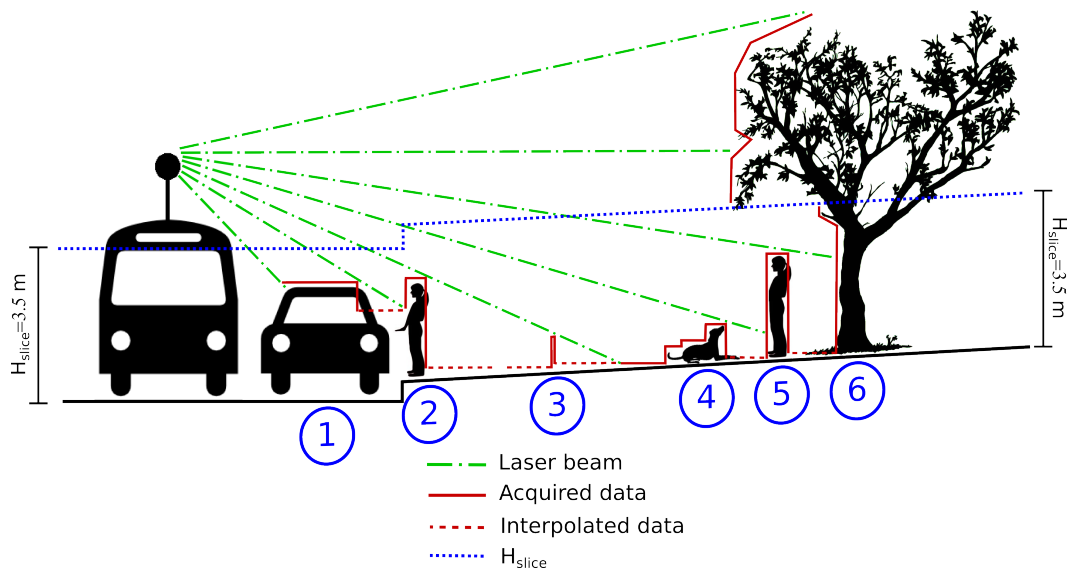


FIGURE 2.20 – Projection par tranches sur un profil 1D. La scène urbaine contient les objets suivants : ① voiture, ② piéton, ③ bruit, ④ chien, ⑤ piéton et ⑥ arbre. Le traitement par tranches évite que les objets hauts tels que les arbres (objet ⑥) occulte des objets en dessous tels que les piétons (objet ⑤).

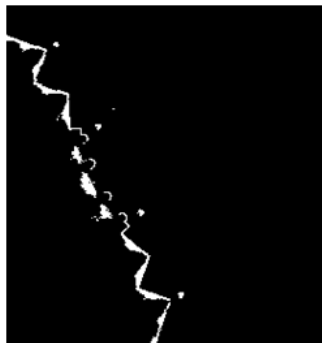
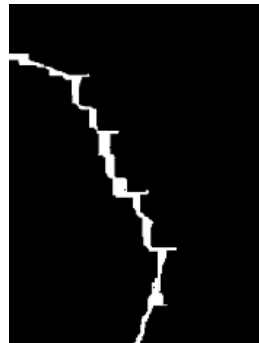
### 2.2.5 Segmentation de blocs de maisons

Les données sont acquises en flux continu, au fur et à mesure que le véhicule avance. La segmentation en blocs de maison présente les avantages suivants : 1. chaque îlot constitue une entité sémantique, autour de laquelle se situe le trottoir. 2. le flux continu est divisé en tronçons d'une taille raisonnable, évitant des problèmes de dépassement de mémoire.

Dans la littérature plusieurs approches modélisent par des plans les façades des bâtiments [94, 95, 96, 97, 98]. Ces méthodes sont alors limitées à des bâtiments composés de façades planes. Les structures comme les balcons peuvent mettre à défaut cette hypothèse et surtout des bâtiments avec des empreintes circulaires, par exemple, font échouer ces méthodes.

Dans le cadre de la thèse de Jorge Hernández, nous avons proposé des méthodes de segmentation automatique de nuages de points en îlots. Celle qui a été utilisée dans le cadre du projet est celle qui n'impose pas de contraintes d'alignement des façades. Les contraintes suivantes sont utilisées : 1) la hauteur d'un bâtiment est d'au moins 4 mètres, 2) la distance minimale entre îlots est supérieure à 2 mètres et 3) la longueur d'un îlot est d'au moins 10 mètres. La procédure est la suivante : 1) un seuil de 4 m est appliqué à l'image de hauteur ( $f - f_{min}$ ). L'image résultat présente des *CCs* de bâtiments et des objets hauts tels que les lampadaires ; néanmoins dans un même îlot, il peut y avoir de petites déconnexions entre façades, dues à des données manquantes ou à des colonnes qui occultent des parties de façade provoquant leur déconnexion (voir figure 2.21(a) et son image de projection 2.21(b)) ; 2) pour fusionner les façades dans les îlots, nous appliquons une fermeture d'une taille de 2 mètres (figure 2.21(c)) ; 3) puis nous éliminons les *CCs* de longueur inférieure à 10 mètres. Les *CCs* résultantes correspondent aux îlots de la scène. Pour tracer la séparation de manière équidistante entre îlots voisins, nous calculons la fonction distance aux *CCs* détectées et lui appliquons la ligne de partage des eaux. L'image segmentée est également rétro-projetée pour découper le nuage de points original.

Les figures 2.22 et 2.23 illustrent la segmentation par îlots de données 3D issues des deux systèmes d'acquisition, Lara3D et Stereopolis respectivement. Deux cas sont illustrés : deux côtés de rues et plusieurs îlots en diverses directions. Comme l'on peut observer, la méthode présente de bons résultats dans les deux cas. Néanmoins, dans le cas de la figure 2.22 correspondant aux données CAOR, on a détecté séparément les régions 1 et 2 qui appartiennent au même îlot (de même pour les régions 5, 6 et 7), car le profil d'acquisition du capteur laser est perpendiculaire à la direction de la façade et il y a trop de données manquantes. Ces régions sont séparées par une distance supérieure à 2 mètres et la fermeture utilisée n'arrive pas à les relier.

(a) Mairie du  $V^{eme}$ (b) Colonnes déconnectant  
une façade

(c) Fermeture 2 m

FIGURE 2.21 – (a) Mairie du  $V^{me}$  arrondissement de Paris. (b) Déconnexion de la façade à cause des colonnes (c) Reconnexion grâce à une fermeture de 2 m.

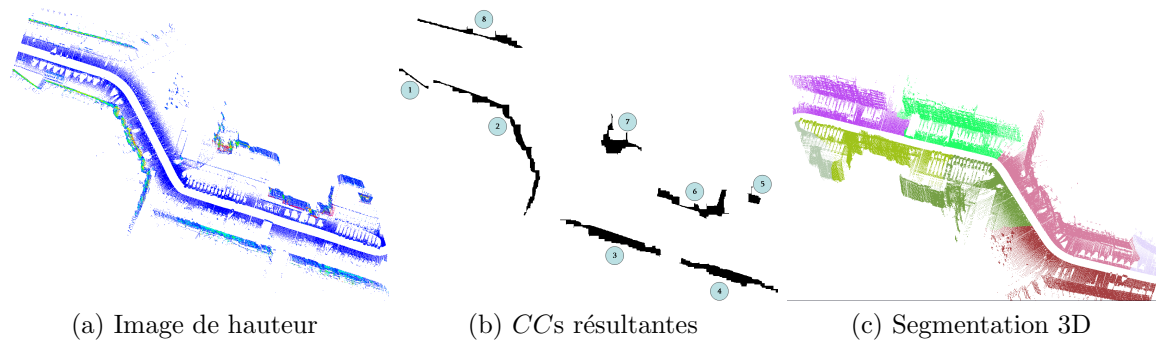


FIGURE 2.22 – Segmentation par îlots de rues non-alignées. Données ©CAOR Fin de la Rue Soufflot - début de la Place du Panthéon.

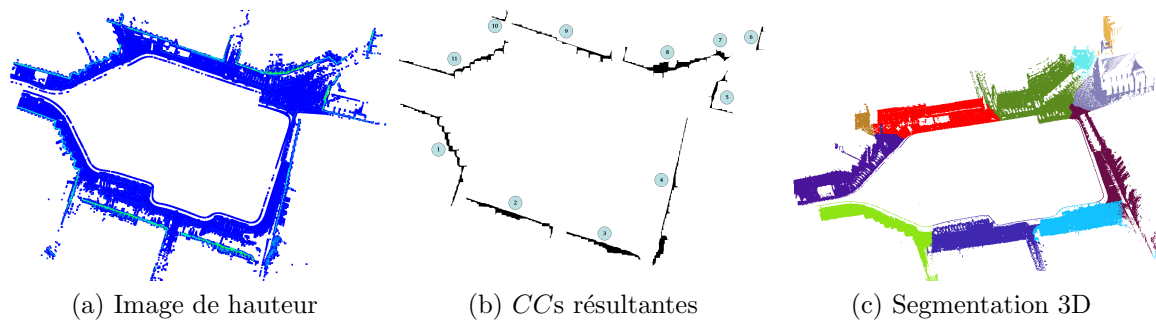


FIGURE 2.23 – Segmentation par îlots de rues non-alignées. Données ©IGN Place du Panthéon.

### 2.2.6 Détection du sol

Des nombreuses méthodes de l'état de l'art abordent ce problème en ajustant des plans [99, 90, 100, 101, 102, 103, 104] ou des polynômes [105]. Ces approches peuvent échouer dans des scénarios complexes avec des rampes d'accès ou de ralentisseurs. Nous proposons de considérer le sol comme une région étendue, avec des transitions d'élévation douces entre pixels voisins [93, 82]. L'algorithme de zones quasi-plates (voir définition 3) est donc adapté pour sa détection.

**Définition 3** *Deux pixels voisins  $p$  et  $q$  appartiennent à la même zone quasi-plate d'une image  $I$ , si leur différence de profondeur  $|I_p - I_q|$  est inférieure à un certain seuil  $\lambda$  donné.*

$$\forall (p, q) \text{ voisins} : |I_p - I_q| \leq \lambda \quad (2.1)$$

Le sol est ainsi détecté comme la région la plus grande après avoir appliqué l'algorithme de zones quasi-plates à l'image d'élévation interpolée. Les éventuels trous de cette région correspondent à des objets très hauts (comme les lampadaires) et ils sont rajoutés au masque du sol. Cette méthode, proposée dans le cadre de la thèse de Jorge Hernández, est simple, rapide, sans contraintes de planarité du sol et robuste à des configurations complexes sans besoin d'identifier le modèle le plus approprié. La figure 2.24a montre le masque du sol des images 2.19d. Le masque final est rétro-projeté sur le nuage de points, permettant la segmentation de nuages de points 3D de façades et points du sol (voir figure 2.24b).

### 2.2.7 Segmentation d'objets 3D

L'un des apports majeurs de nos travaux est la méthode de détection et de classification automatique d'objets 3D. Elle permet d'un côté le filtrage des artefacts afin de faciliter la segmentation de façades et de la chaussée, de l'autre côté la réintroduction des certains objets du mobilier urbain (comme par exemple les lampadaires) afin d'incrémenter le réalisme visuel d'une scène modélisée. L'atout principal par rapport à l'état de l'art est la généralité vis-à-vis du système d'acquisition (trois systèmes différents ont été utilisés dans le cadre du projet) ainsi que la prise en compte de plusieurs classes d'objets simultanément. La généralité relative au type d'objet peut être abordée grâce à une segmentation performante que nous avons développée. Une première version de cette méthode a été mise en place dans la thèse de Jorge Hernández et améliorée par Andrés Serna en rajoutant plusieurs hypothèses dans la phase de sélection de candidats.

La méthode de détection d'objets consiste en une première sélection de candidats suivie d'une étape de validation. La sélection de candidats est l'union des objets vérifiant une des deux hypothèses suivantes :

1. les objets ne font pas partie du masque du sol. Autrement dit, une discontinuité d'élévation évite la propagation du sol vers l'intérieur de l'objet. Sur le profil 1D

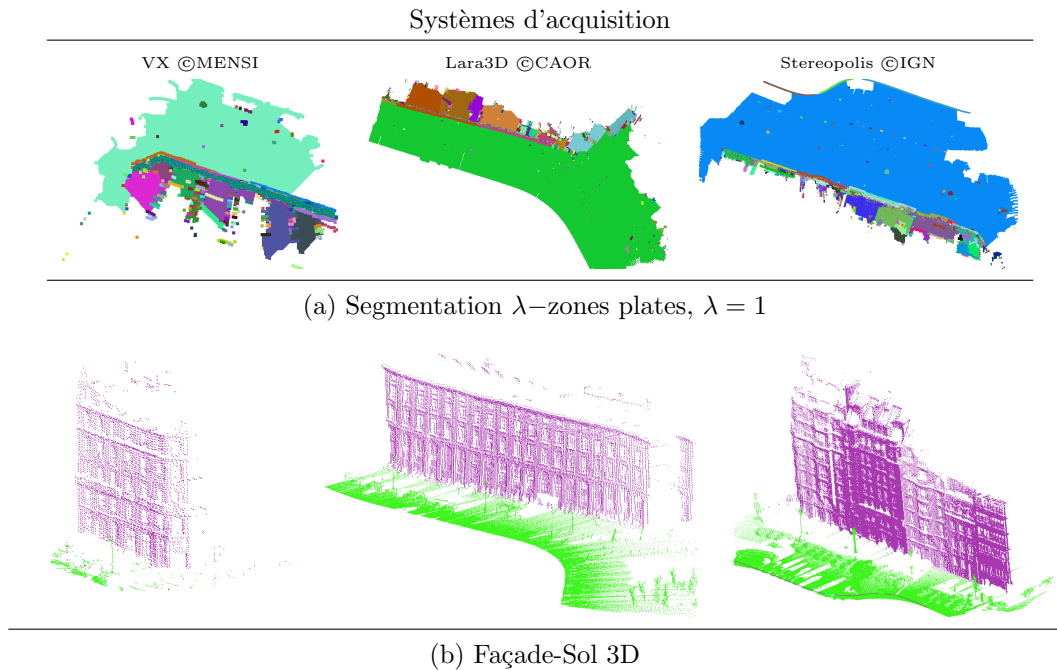


FIGURE 2.24 – Segmentation du masque du sol (comprenant les objets urbains).

de la figure 2.25b tous les objets sauf le chien identifié avec le numéro ④ sont détectés.

2. les objets 3D apparaissent sur le nuage de points comme des “bosses” au niveau du sol. Le résidu d’un opérateur de remplissage de trous sur le profil inversé donne le deuxième jeu des objets candidats, illustrés sur la figure 2.25d. Le chien ④ qui n’avait pas été détecté avec l’hypothèse précédente est maintenant bien détecté. En revanche, les objets ①, ②, ⑥ et ⑦ qui se trouvent sur le bord des données acquises ne sont pas détectés ou seulement en partie.

L’union des deux jeux de candidats, illustré sur la figure 2.25e, donne l’ensemble d’objets candidats. Ces candidats sont filtrés, utilisant un filtre surfacique qui supprime les objets de très petite taille, correspondant au bruit. Une contrainte sur la valeur d’accumulation est rajoutée à ce filtrage pour éviter de supprimer des objets de petite extension surfacique mais significatifs pour l’application, tels que les poteaux.

La méthode que nous venons d’introduire, détecte correctement les objets présents dans la scène. En revanche, ces objets doivent être correctement individualisés pour bien les caractériser et ainsi simplifier leur classification ultérieure. Sur l’exemple de la figure 2.25, nous constatons que les objets ① et ② font partie de la même composante connexe. Ceci pose deux problèmes. D’une part, on détecte un objet en moins. D’autre part, la classification de l’union des deux (ou plusieurs) objets sera plus difficile, car ses caractéristiques s’éloigneront des caractéristiques du modèle d’objet isolé. Nous



revenons à la définition d'objet 3D que nous avons introduite : un objet apparaît comme une sorte de bosse sur le sol. Nous analysons le contenu de chaque composante connexe détectée comme étant un objet et nous appliquons une segmentation contrainte par le nombre de maxima significatifs de celle-ci. L'algorithme de h-maxima [106] est utilisé pour sélectionner les maxima significatifs. La figure 2.26 illustre le fonctionnement de cette re-segmentation de composantes connexes sur un exemple réel. Nous constatons que plusieurs voitures qui avaient été détectées dans une seule composante connexe sont correctement séparées.

Le principal inconvénient de cette méthode est la segmentation d'objets "à plusieurs bras", comme les lampadaires, les clôtures, etc. En effet, ces objets présentent plusieurs maxima significatifs et seront sur-segmentés. Par ailleurs, si un vélo est garé à côté d'un lampadaire, le vélo ne sera pas segmenté car il ne contiendra pas de maximum significatif. Pour résoudre ce problème, des analyses de forme ou contextuelles pourront aider à prendre la décision de re-segmenter un objet ou pas.

Cette étape de segmentation est essentielle pour bien caractériser les objets urbains et aboutir à une bonne classification. Nous verrons dans la section suivante que notre approche produit de très bons résultats sur la base TerraMobilita/IQmulus (<http://data.ign.fr/benchmarks/UrbanAnalysis/>) en s'appuyant sur la méthode de segmentation que nous venons de décrire.

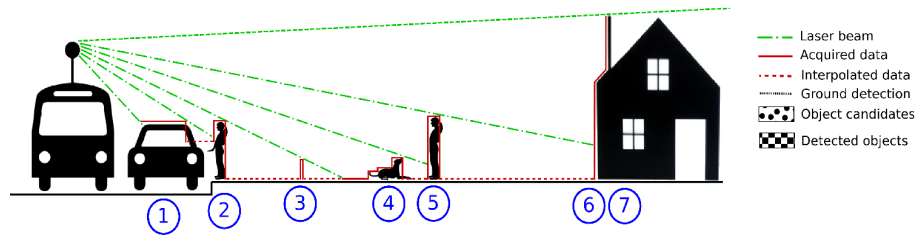
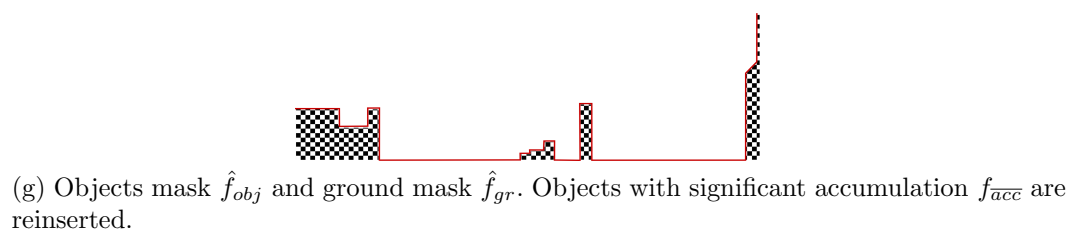
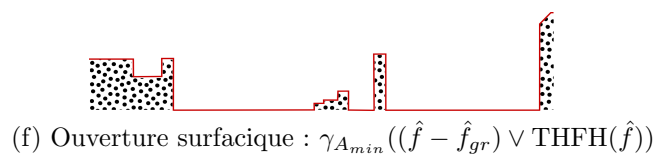
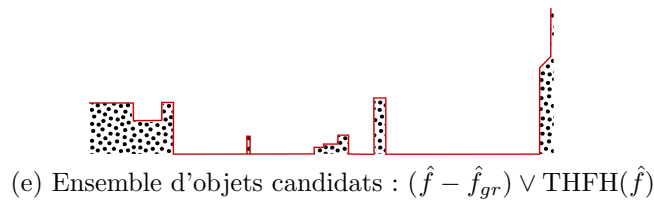
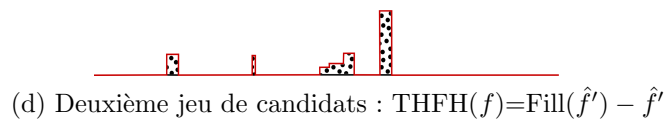
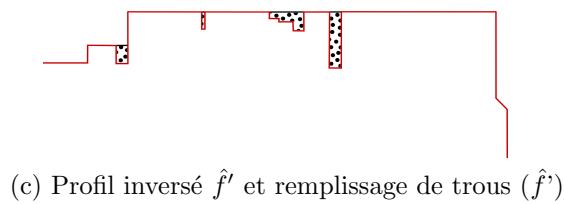
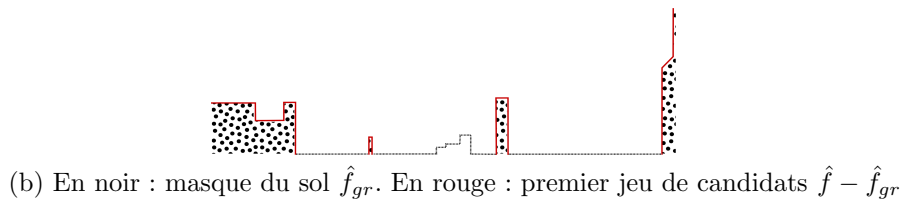
(a) Système d'acquisition et profil interpolé  $\hat{f}$ .

FIGURE 2.25 – Méthode de détection d'objets urbains, illustrée sur un profil 1D. Cette scène urbaine contient les objets suivants : ① voiture, ② piéton, ③ bruit, ④ chien, ⑤ piéton, ⑥ façade et ⑦ cheminée.

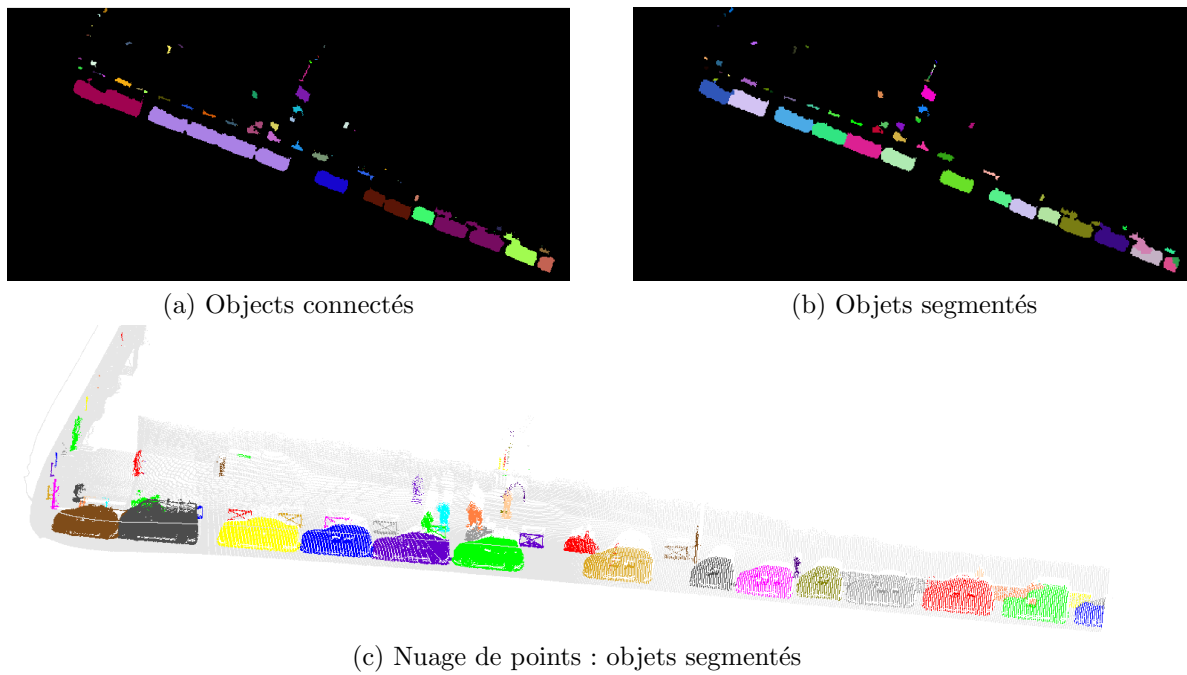


FIGURE 2.26 – Segmentation d’objets 3D basée sur une ligne de partage des eaux contrainte par les maxima d’élévation. Chaque couleur correspond à un objet différent. Données sur la *rue Vaugirard* à Paris. IGN©France.

### 2.2.8 Classification d’objets 3D

Une fois les objets segmentés, nous utilisons des techniques d’apprentissage pour les classer dans différentes catégories de manière à rajouter une sémantique à la scène et pouvoir développer des applications utiles. Nous avons choisi d’utiliser la méthode de classification classique de Support Vector Machine (SVM) pour ses capacités à gérer des jeux de données de grande taille, des ensembles d’apprentissage limités. Par ailleurs, elle est facile à implémenter et produit des résultats précis dans des applications similaires rapportées dans la littérature [107, 82, 108, 109].

Trois types de caractéristiques sont utilisées pour la classification :

- **Géométriques** : surface, volume (calculé comme l’intégrale de l’image “hauteur” sous le masque de l’objet) et périmètre de l’objet ; surface de la boîte englobante ; axes moyens ; maximum, moyenne, déviation standard et mode (valeur la plus fréquente) de la hauteur de l’objet ;
- **Contextuelles** : Nombre d’objets voisins  $N_{neigh}$ , défini comme le nombre de régions touchant l’objet, utilisant une connectivité 8 sur l’image d’élévation. Cette caractéristique est très discriminante pour les groupes d’arbres et de voitures garées les unes à côté des autres ; l’indice de confiance  $C_{ind} = n_{real}/(n_{real}+n_{interp})$ , où  $n_{real}$  et  $n_{interp}$  correspondent au nombre de pixels non vides de l’objet avant et après interpolation, respectivement. En général, les objets éloignés ou occultés ont des indices de confiance faibles.
- **Intensité** : Moyenne de l’intensité du laser sur les points appartenant à l’objet. Cette caractéristique est utilisée seulement si disponible.
- **Couleur** : Moyenne rouge, vert et bleu sur les points appartenant à l’objet. Ces caractéristiques sont utilisées seulement si disponibles.

La fiabilité de ces caractéristiques dépend du système d’acquisition. Les capteurs précis et bien calibrés contribuent à la robustesse de la méthode de classification. Il est à noter que les caractéristiques géométriques peuvent être adaptées à n’importe quel nuage de points XYZ, tenant compte de la résolution du système d’acquisition.

Les bases de cette étape de classification ont été établies dans la thèse de Jorge Hernández, avec des descripteur géométriques principalement. Andrés Serna a rajouté d’autres descripteurs rendant la méthode plus robuste.

#### 2.2.8.1 Résultats sur la base TerraMobilita/iQmulus

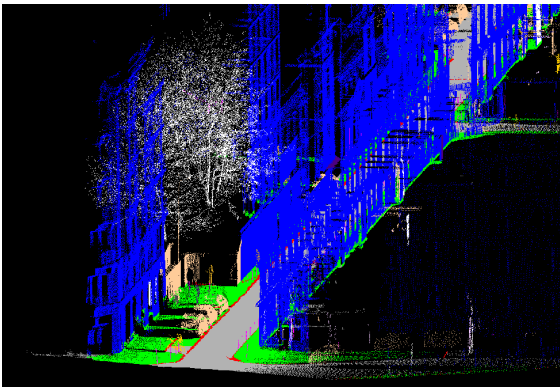
L’évaluation des résultats est réalisée selon le protocole TerraMobilita/iQmulus décrit dans [110]. La première étape de ce protocole divise le nuage de points en 3 classes : *surface* (comprenant les façades et le sol), *objet* et *autre*. Une quatrième classe est rajoutée, *non classifié* incluant les points 3D qui ne sont pas annotés dans la vérité terrain. Ces points correspondent généralement au bruit ou des points dont l’appartenance à une classe est ambiguë, comme par exemple les points derrière les façades. 18.31 % des points ne sont pas classés dans la vérité terrain.

La table 2.1 présente la matrice de confusion de nos résultats sur les 3 catégories.

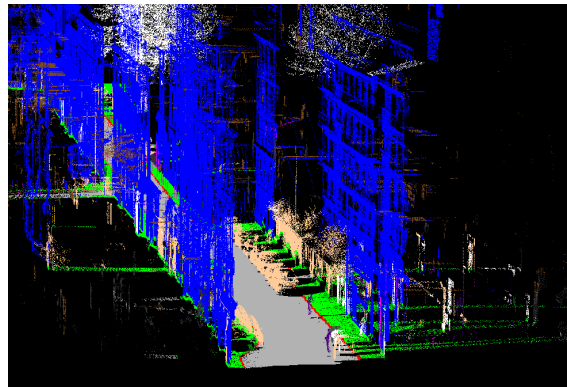
TABLE 2.1 – Résultats de classification pour les 3 catégories générales de la base TerraMobilita/iQmulus. GT : vérité terrain, AR : Résultat automatique. N.C. : non classé. Sur la matrice de confusion, les résultats sont présentés en pourcentage par rapport au nombre total de points 3D du nuage de points (12 millions de points).

GT/AR	N.C.	autre	surface	objet	Sum	R	P	$f_{\text{mean}}$
N.C.	-	-	-	-	18.31 %	-	-	-
autre	0.00 %	0.00 %	0.13 %	0.04 %	0.17 %	0.59 %	0.05 %	0.08 %
surface	1.90 %	2.19 %	70.81 %	0.91 %	75.82 %	93.40 %	98.82 %	96.03 %
objet	0.09 %	0.02 %	0.72 %	4.88 %	5.70 %	85.49 %	83.72 %	84.59 %
Sum	1.99 %	2.21 %	71.66 %	5.82 %	81.69 %	<b>Taux de bonne classification : 92.65 %</b>		

En utilisant notre méthode, les *objets* sont détectés avec un  $f_{\text{mean}}$  de 84.59 %. Les objets sont ainsi séparés des *surfaces* incluant les façades et la route. 75.82 % du nombre total de points appartiennent aux surfaces alors que seulement 5.7 % correspondent à la classe *objet*. La classe *surfaces* est détectée avec un  $f_{\text{mean}}$  égal à 96.03 %. Les erreurs classiques de classification sont illustrées sur la figure 2.27 : façades basses considérées comme des objets (figure 2.27(a)). La classe *autre* n'est pas bien classifiée par notre méthode. Cela n'est pas critique car cela ne représente que 0.17 % du nombre total de points de la scène. Les points de la classe, *non classifié*, ne sont pas considérés pour l'évaluation. Cela n'est pas un réel problème car elle contient principalement des points 3D derrière les façades et ne sont pas dans l'espace public, zone d'intérêt de notre étude. Le taux de bonne classification avec les 4 catégories principales est de 92.65 %.



(a) La façade basse sur la partie gauche en bas, l'arbre a été classé comme objet.



(b) La façade basse sur la partie droite de l'image, derrière les voitures est classée comme objet.

FIGURE 2.27 – Erreurs de classification sur la base TerraMobilita/iQmulus database. Façades (bleu), trottoir (vert), route (gris), voitures (rose), poteaux (magenta), arbres (blanc), piétons (indigo). Rue Cassette à Paris, France. Stereopolis II, IGN©.

Nous avons évalué la qualité de la segmentation de la classe *objet*. Pour cela, une

mise en correspondance des objets est réalisée, imposant des pourcentages d'inclusion et de recouvrement réciproque, tel que décrit dans [110]. Après cette mise en correspondance, les valeurs classiques de *precision*, *recall* et  $f_{\text{mean}}$  sont calculées, en fonction du seuil d'inclusion imposé  $m$ .  $P(m)$ ,  $R(m)$  and  $f_{\text{mean}}(m)$  diminuent quand  $m$  augmente. En effet, si les contraintes d'appariement sont plus fortes moins de correspondances pourront être établies. La vérité terrain contient 189 objets. Nous détectons 142 avec  $m = 0.1$ , ce qui correspond à 75.13 % du total. En imposant  $m=0.9$  nous détectons 127 objets (67.20 %). La qualité de la segmentation diminue lentement en augmentant  $m$  ( $f_{\text{mean}}(0.1) = 85.80$  % et  $f_{\text{mean}}(0.9) = 76.74$  %), ce qui prouve une bonne correspondance entre les contours des objets segmentés et ceux de la vérité terrain.

Les erreurs topologiques de la segmentation de la classe *objet* sont données sur la figure 2.28b. Les courbes (1-to- $n$ ) et ( $n$ -to-1) indiquent les erreurs de sur-segmentation et de sous-segmentation, respectivement. Ces erreurs dépendent de la valeur  $m$  utilisée pour l'appariement. Des valeurs faibles induisent des erreurs topologiques (sur et sous-segmentation). Une valeur de  $m=0.5$  est un bon compromis : la *precision* et le *recall* restent élevés (2.28a) alors que les erreurs topologiques ne sont pas possibles.

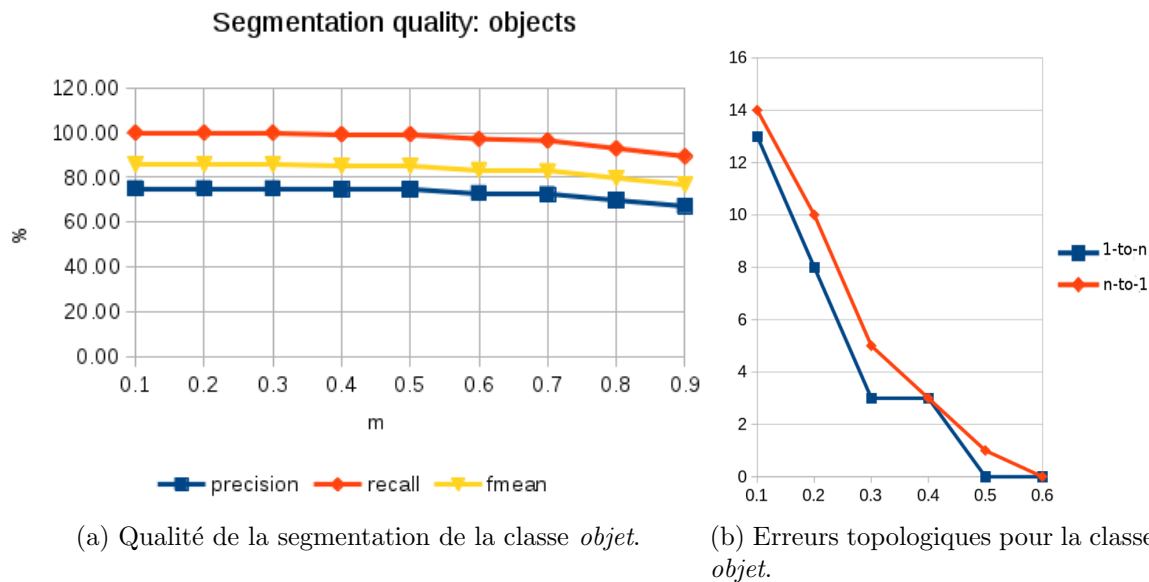


FIGURE 2.28 – Qualité de la segmentation et erreurs topologiques de la classe *objet* sur la base TerraMobilita/iQmulus.

La table 2.2 montre les résultats de classification pour le sous-arbre *objets* incluant 3 catégories : *statique*, *dynamique* et *naturel*. Grâce à notre méthode, les objets *dynamiques* et *naturels* sont classifiés avec un  $f_{\text{mean}}$  de 92,56 % et 95,49 %, respectivement. Notez que les classes *dynamique* et *naturel* représentent les structures les plus grandes de la scène avec 92,90 % de tous les points 3D du sous-arbre. Le principal inconvénient

de la méthode est dû aux objets statiques qui sont confondus avec des objets dynamiques, comme c'est le cas des barrières et de murets classifiés comme voitures. Autres erreurs de faible importance sont dus aux parcmètres qui sont classés comme piétons ou les arbustes classés comme voitures. Cependant, ces erreurs ne sont pas critiques puisqu'ils ne représentent que 7,10 % du nombre total de points du sous-arbre. La *precision* globale de notre méthode de classification des *objets* est de 91,84 %.

TABLE 2.2 – Résultats de classification pour le sous-arbre *objet* de la base TerraMobilita/iQmulus. GT : ground truth, AR : Résultat Automatique. N.C : non classé. dyn : dynamique. Sur la matrice de confusion, les résultats sont présentés en pourcentage par rapport au nombre total de points 3D du nuage de points (12 millions de points).

GT/AR	N.C.	statique	dyn	natural	Sum	R	P	$f_{\text{mean}}$
N.C.	-	-	-	-	0,0 %	-	-	-
statique	0,0 %	0,75 %	6,09 %	0,26 %	7,10 %	10,58 %	50,85 %	17,51 %
dyn	0,15 %	0,72 %	83,63 %	0,85 %	85,36 %	97,98 %	93,13 %	95,49 %
natural	0,0 %	0,0 %	0,08 %	7,45 %	7,54 %	98,86 %	87,01 %	92,56 %
Sum	0,15 %	1,48 %	89,80 %	8,57 %	100,0 %	<b>Taux de bonne classification : 91,84 %</b>		

La figure 2.29 présente la qualité de segmentation de la classe *objets dynamiques*. Le nombre total d'*objets dynamiques* annotés dans GT est de 113. Selon les résultats, notre segmentation retrouve correctement 97 objets dynamiques ( $P(0,1)=85,84\%$ ) avec  $m=0,1$ , tandis que 88 objets dynamiques ( $P(0,9)=77,88\%$ ) sont récupérés pour  $m=0,9$ . La qualité géométrique de notre segmentation est bonne puisque les performances diminuent lentement. Par exemple,  $f_{\text{mean}}(m)$  décroît de 92,38 % à 83,81 % pour quand  $m$  varie de 0,1 à 0,9. Notez que dans la gamme  $m=[0,1, 0,5]$  nos performances sont constantes, et dans la gamme  $m=[0,5, 0,9]$ ,  $f_{\text{mean}}(m)$  diminue moins de 9%, ce qui prouve la robustesse de notre processus de segmentation.

La figure 2.3 présente les résultats de classification du sous-arbre *objets dynamiques* considérant 3 catégories : *piétons*, *2 roues* et *4+ roues*. Le taux de bonne classification est de 99,34 %, ce qui prouve le bon comportement de notre approche. La classe *4+ roues* incluant les voitures ont un  $f_{\text{mean}}$  égal à 99,86 %. La classe *4+ roues* contient 97,20 % des points dans la classe *objet dynamique*. La classe *piétons* et *2 roues* le  $f_{\text{mean}}$  sont 83,87 % et 75,71 %, respectivement. Le principal inconvénient est lié aux motos qui peuvent être sur-segmentées et classifiées alors comme piétons. Un autre problème est dû aux piétons qui marchent près des voitures hautes, type fourgonnettes, qui peuvent ne pas être correctement segmentés, menant ainsi à un problème de sous-segmentation.

### 2.2.9 Diagnostic d'accessibilité

Environ 80 millions de personnes en Europe souffrent d'un handicap modéré ou sévère. Les obstacles sur la voie publique limitent leur capacités de déplacement et les rend vulnérables à l'exclusion. En 2007, L'UE a signé *La convention des Nations*

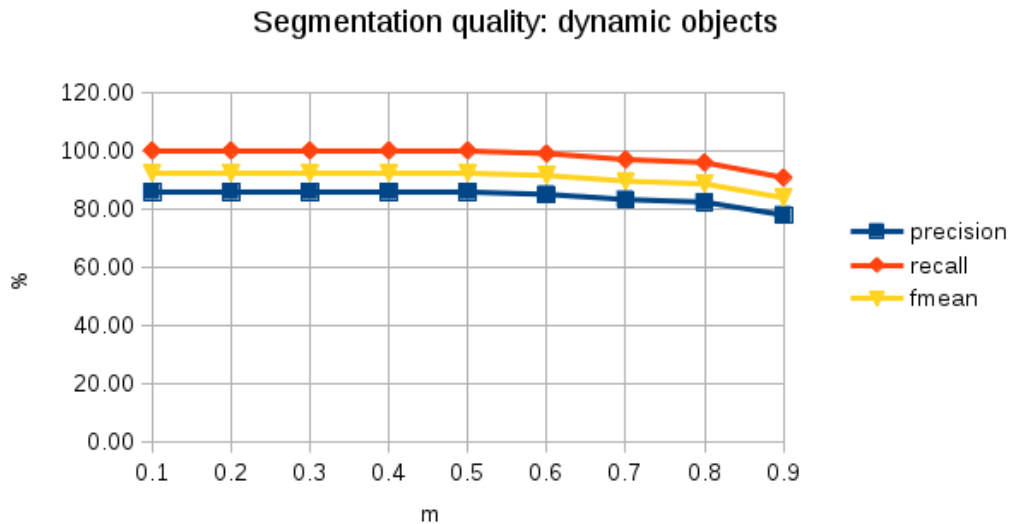


FIGURE 2.29 – Qualité de segmentation pour la classe *objets dynamiques* sur la base TerraMobilita/iQmulus.

TABLE 2.3 – Résultats de classification des *objets dynamiques* de la base TerraMobilita/iQmulus. GT : vérité terrain, AR : résultats automatiques. La table de confusion contient les pourcentages de points par rapport au nombre de points 3D du nuage de points initial (12 millions de points).

GT/AR	piétons	2 roues	4+ roues	Sum	Recall	Precision	$f_{\text{mean}}$
piétons	1,63 %	0,00 %	0,12 %	1,76 %	92,80 %	76,51 %	83,87 %
2 roues	0,39 %	0,65 %	0,00 %	1,04 %	62,71 %	95,51 %	75,71 %
4+ roues	0,11 %	0,03 %	97,06 %	97,20 %	99,86 %	99,87 %	99,86 %
Sum	2,13 %	0,68 %	97,18 %	100,0 %	<b>Taux de bonne classification : 99,34 %</b>		

*Unies sur les droits de personnes handicapées* [111]. L’objectif est de donner accès aux droits (en particulier le droit à la mobilité pour se déplacer dans leur vie courante) à tous les citoyens y compris ceux qui ont un handicap moteur. En France, selon la loi 2005-102<sup>1</sup>, les autorités locales sont tenues de garantir l’accessibilité aux espaces publics. Ainsi, être en mesure de faire des diagnostics d’accessibilité à grande échelle dans les milieux urbains afin d’identifier les endroits nécessitant une adaptation est un vrai sujet d’actualité.

L’utilisation de la 3D permet de visualiser les pentes et les dévers (qui sont critiques dans certains cas de mobilité, comme par exemple le vélo) et d’en tenir compte pour

1. Loi 2005-102 du 11 février 2005 : “Pour l’égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées”.



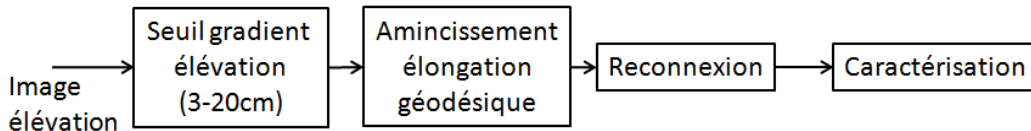
les calculs d'itinéraires adaptés à différents types de mobilité. La 3D permet de cartographier les obstacles, par exemple de prévoir la largeur critique de passage pour les fauteuils roulants.

Différentes approches sont proposées dans la littérature pour localiser les bords de trottoir à partir des données laser. Dans [112, 113] une fonction sigmoïde est ajustée aux différents profils d'acquisition. Le voisinage implicite entre différents profils n'est donc pas pris en compte. Dans [105], le flanc descendant entre le trottoir et la route est détecté grâce au changement de la direction normale aux données. La taille du voisinage pour l'estimation de la normale reste un point fragile de la méthode. Nous avons proposé, dans le cadre de la thèse de Jorge Hernández, une technique utilisant la détection de zones quasi-plates suivie d'une segmentation contrôlée par marqueurs [114]. Cette technique reste fragile en présence de rampes d'accès où la propagation de zones quasi-plates peut fusionner la route et le trottoir.

Dans le cadre de la thèse d'Andrés Serna, nous avons mis au point une technique fondée sur le filtrage du gradient de l'image d'élévation suivi d'une étape de reconnexion des bords détectés. Cette technique a été publiée dans [115] et a reçu le prix au meilleur papier ISPRS 2013. Nous décrivons cette technique ci-dessous.

### 2.2.9.1 Détection des bords de trottoir

L'approche que nous proposons exploite la proximité des profils voisins pour rendre la procédure plus robuste et sans échouer à cause des rampes d'accès dont la discontinuité en hauteur est de l'ordre du niveau de bruit des données 3D [115]. Nous définissons un bord de trottoir comme une structure longue qui a une discontinuité en hauteur entre 3 et 20 cm. La méthode que nous avons développée est donc la suivante (voir diagramme de la figure 2.30) :



(a) Schéma pour la localisation du bord du trottoir

FIGURE 2.30 – Schéma de localisation du bord du trottoir.

- le gradient de l'image d'élévation est calculé ;
- un seuil entre 3 et 20 cm sélectionne les structures candidates à être des bords de trottoir ;
- un amincissement par élévation géodésique supprime les éventuelles fausses détections liées au bruit ;
- finalement les tronçons détectés sont reconnectés. En effet, les accès piétons peuvent avoir une différence d'élévation inférieure à 3cm. Par ailleurs, les voi-

tures garées occultent une grand partie des bords de trottoirs, c'est pourquoi cette re-connexion est indispensable.

La figure 2.31 illustre l'effet du seuil d'élongation utilisé. Un seuil égal à zéro préserve toutes les structures détectées par le gradient. Le seuil  $E_{min}=5$  ne supprime pas tout le bruit, alors que une élongation  $E_{min}=20$  élimine quelques tronçons correspondant aux trottoirs (qui sont un peu courts à cause des occultations). Une valeur d'élongation  $E_{min}=10$  semble un bon compromis pour éliminer le bruit tout en préservant les tronçons utiles. Nous remarquons quelques détections près des bâtiments. Elles correspondent aux marches d'accès à ceux-ci, car elles vérifient nos hypothèses. Si ces marches ne doivent pas être détectées, une contrainte de distance minimale aux bâtiments devra être rajoutée.

### 2.2.9.2 Reconnexion des bords de trottoir

Le principal inconvénient de cette méthode est l'absence de détection des rampes d'accès dont la hauteur est inférieure à 3 cm ainsi que les bords de trottoir occultés. Des solutions pour combler ces zones existent dans la littérature, mais elles ne sont pas suffisantes, car elles supposent que les reconnexions sont rectilignes alors que les bateaux peuvent se trouver à l'angle de deux rues, comme celle de la figure 2.34c. Nous observons sur cette figure que le gradient d'élévation n'est pas assez précis pour localiser la reconnexion. Une information a priori est nécessaire pour combler ces zones manquantes.

Notre stratégie utilise les courbes de Bézier. Deux bordures plus proches qu'une certaine distance  $d_{min}$  sont reconnectées en traçant une courbe de Bézier entre les extrémités géodésiques [116, 117]. La direction des segments est utilisée pour définir les paramètres de ces courbes.

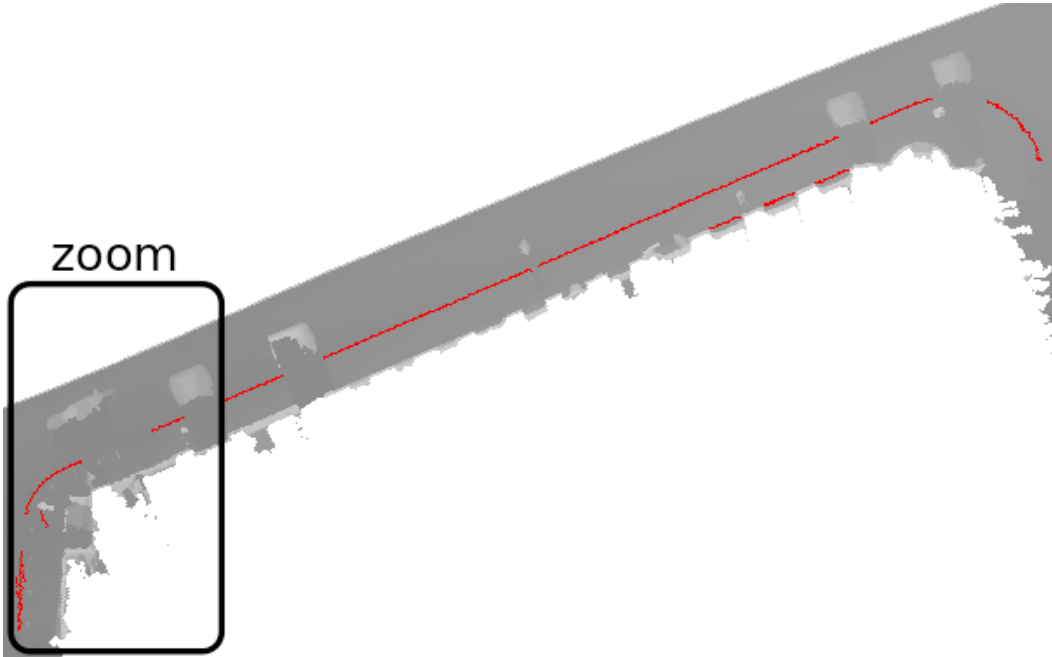
La courbe de Bézier,  $B(t)$ , étant donnés les points  $P_0$ ,  $P_1$ , and  $P_2$  est définie selon l'équation suivante :

$$B(t) = (1 - t)^2 P_0 + 2(1 - t)t P_1 + t^2 P_2, \forall t \in [0, 1] \quad (2.2)$$

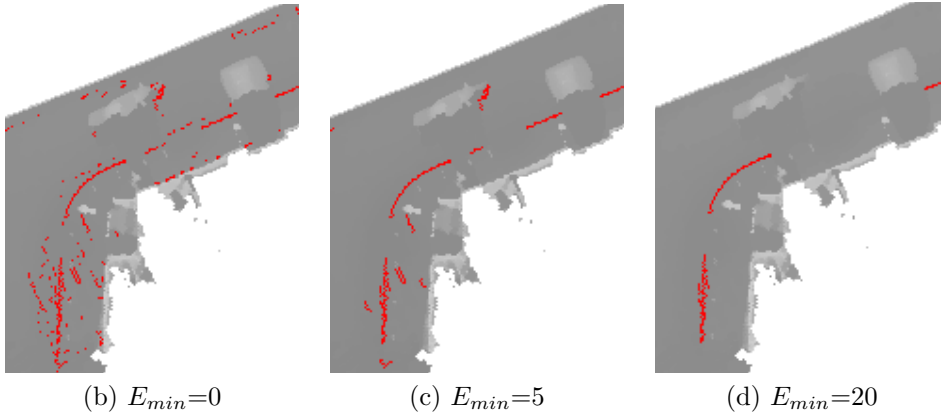
Elle part de  $P_0$  vers  $P_1$  et après elle tourne pour arriver à  $P_2$ . Les tangentes de cette courbe en  $P_0$ , et  $P_2$  passent par  $P_1$ , le point de contrôle. Cette courbe rejoint les deux extrémités sans changement abrupte de direction.

Nous devons donc déterminer les 3 points qui définissent la courbe.  $P_0$  et  $P_2$  seront les extrémités des segments à rejoindre.  $P_1$  sera l'intersection de la prolongation des deux segments à reconnecter ( $C_0$  et  $C_2$ ). Ainsi, le problème est réduit à trouver le point de contrôle  $P_1$ .

- Si les segments sont co-linéaires nous positionnerons  $P_1$  au milieu du segment  $\overline{P_0 P_2}$ . Les trois points de contrôle seront co-linéaires et la reconnexion sera une ligne droite (voir figure 2.32a).
- Si les segments à reconnecter ne sont pas co-linéaires,  $P_1$  sera localisé à l'intersection de la prolongation des segments, selon la direction estimée à proximité des



(a) Détection du bord de trottoir avec un amincissement d'élongation géodésique avec  $E_{min}=10$



(b)  $E_{min}=0$

(c)  $E_{min}=5$

(d)  $E_{min}=20$

FIGURE 2.31 – Segmentation du bord de trottoir basé sur amincissement par élongation géodésique. Une élongation  $E_{min}=10$  est choisie pour l'ensemble de nos résultats. Données acquises par l'IGN©France.

extrémités (voir figure 2.32b).

Voyons sur l'exemple de la figure 2.33. la méthode de reconnexion. Les courbes de la figure 2.33a montrent trois sections de bords de trottoir ( $A$ ,  $B$  et  $C$ ) à reconnector. Nous commençons par trouver les extrémités géodésiques ( $a_1$ ,  $a_2$ ,  $b_1$ ,  $b_2$ ,  $c_1$  et  $c_2$ ) selon la méthode barycentrique proposée par [117]. Ensuite, pour chaque extrémité, l'extrémité la plus proche appartenant à un autre segment est cherchée. Seulement les

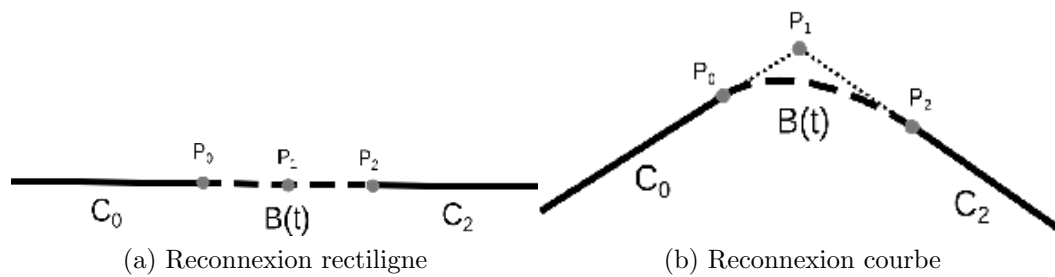


FIGURE 2.32 – Reconnexions par courbes de Bézier quadratiques : cas rectiligne et courbe.

appariements réciproques sont acceptés :  $(a_2, b_1)$  et  $(c_1, b_2)$  sont candidats à reconnector. Ces appariements sont considérés seulement si la distance entre extrémités est inférieure à  $d_{min}$ . Sur cet exemple, nous supposons que la distance euclidienne entre les extrémités  $d(a_2, b_1)$  et  $d(c_1, b_2)$  sont toutes les deux inférieures à  $d_{min}$ . L'orientation de chaque segment à chaque extrémité est calculée, utilisant les pixels du segment à moins d'un mètre de l'extrémité. Ces orientations sont prolongées pour trouver les points de contrôle  $P_{AB}$  et  $P_{CB}$ . Finalement, les courbes de Bézier sont tracées  $R_{AB}$  et  $R_{BC}$  (voir figure 2.33c), reconnectant les 3 segments initiaux.

La figure 2.34 illustre cette procédure dans un cas réel. Le résultat obtenu est lisse et fidèle à la réalité.

$d_{min}$  est fixé empiriquement à 8m, ce qui correspond approximativement à la longueur de deux voitures qui pourraient être garées très proches. Ce paramètre peut être adapté en fonction des conditions d'occultation de chaque base de données. La figure 2.35 présente une zone urbaine avec plusieurs voitures garées sur le côté gauche de la rue, ce qui compromet la visibilité du trottoir. Nous avons proposé précédemment une technique pour segmenter et classifier les objets urbains, et en particulier les voitures. Nous proposons d'exploiter cette information, en permettant des reconnexions plus longues quand on détecte une suite de voitures garées à proximité. Pour ce faire, les zones occultées (zones dont les projections ne contiennent aucun point) en contact direct avec les voitures sont rajoutées comme candidats trottoirs dans la procédure de reconnexion décrite ci-dessus.

La figure 2.35b montre le résultat de cette reconnexion, où les bords de trottoir détectés sont en rouge, les candidats rajoutés derrière les voitures en magenta et les reconnexions en cyan. Le résultat sur le nuage de points 3D est montré sur la figure 2.35c. L'information sur le fait que le bord de trottoir a été détecté sur le nuage de points, ou reconnecté ou reconnecté à travers une zone occultée est gardée, de façon à associer à chaque tronçon un indice de confiance.

Une fois le bord du trottoir détecté, la hauteur de chaque point est calculée et est intégrée dans un SIG 3D. Cette information permet de rajouter des options de

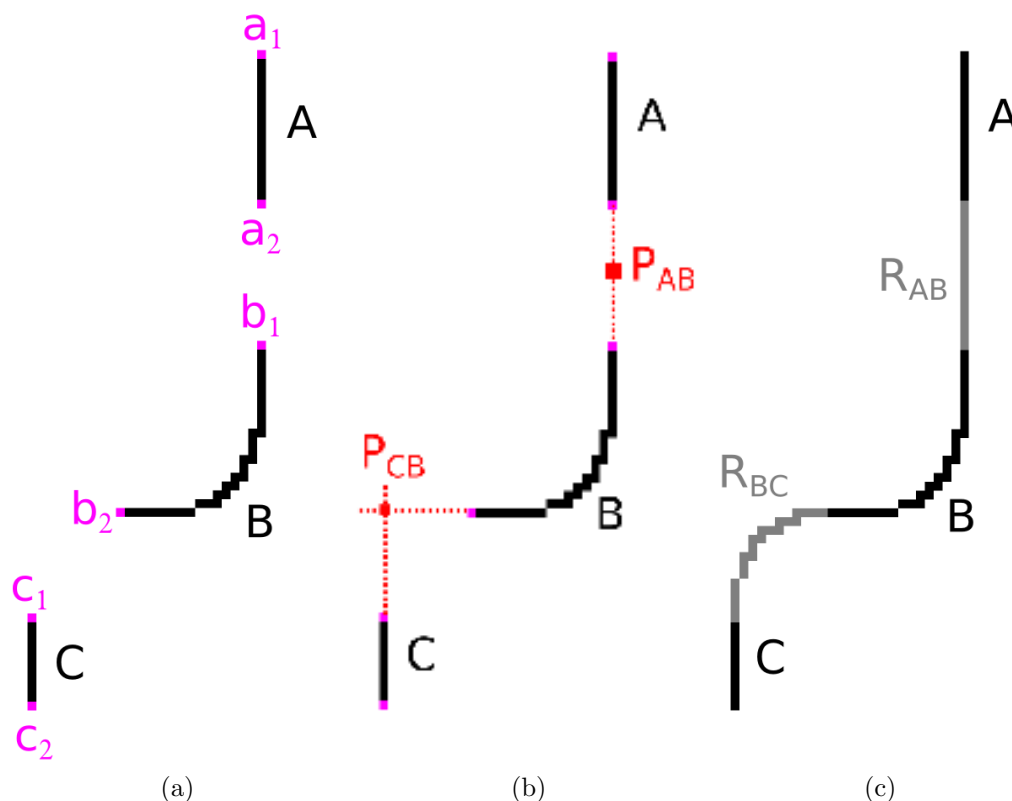


FIGURE 2.33 – Exemple de reconnexion par courbes de Bézier. (a) segments initiaux (en noir) et ses extrémités géodésiques (magenta); (b) points de contrôle définis par l'intersection de la prolongation de tangentes aux extrémités des segments (en rouge); (c) reconnexions utilisant des courbes de Bézier quadratiques (gris).

navigation pour différents types de mobilité. Elle permet par exemple de définir une largeur de passage et une hauteur franchissable adaptées de manière fine à chaque utilisateur : un fauteuil roulant motorisé est plus lourd qu'un fauteuil standard. Le dénivelé franchissable sera inférieur et le système pourra s'adapter de manière précise à chaque utilisateur. Des expérimentations ont été réalisées par l'IGN dans le cadre du projet TerraMobilita à l'échelle du VI<sup>e</sup> arrondissement de Paris, avec des résultats probants.

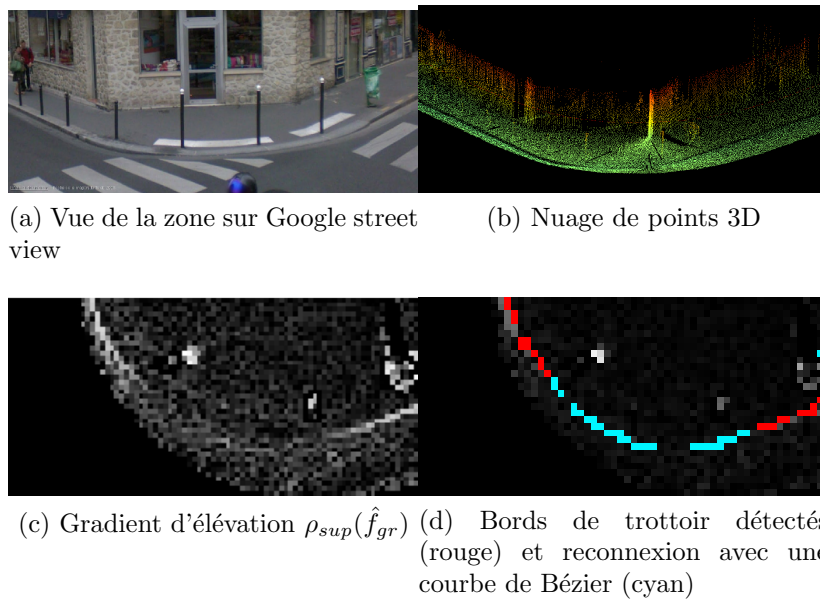


FIGURE 2.34 – Exemple de reconnexion utilisant les courbes de Bézier. *Rue Vaugirard* à Paris, France. Stereopolis II, IGN©.

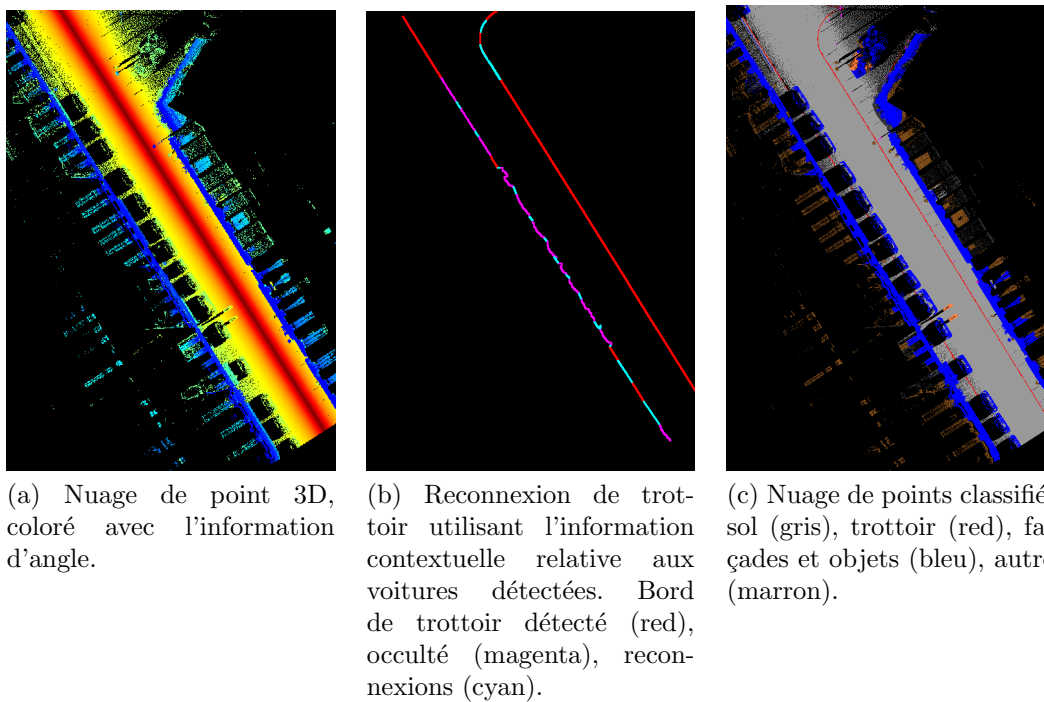


FIGURE 2.35 – Reconnexion de trottoirs utilisant l'information sémantique contextuelle, relative aux voitures garées et à la trajectoire du véhicule. Acquisition réalisée par l'IGN©France.

### 2.2.9.3 Résultats sur la base Enschede

Dans le but de nous comparer avec l'état de l'art, nous avons testé notre approche sur la base Enschede. Cette base contient environ 1000 m de données MLS, acquises à Enschede, The Netherlands, ainsi que des annotations 2D manuelles. Deux informations ont été recueillies : 1) les lignes de bordure de route dont la hauteur est supérieure à 7 cm et 2 cm) les lignes d'une hauteur inférieure à 7cm, correspondant à des rampes d'accès. L'analyse quantitative est effectuée par comparaison entre les lignes extraites automatiquement et les annotations manuelles. Les détections automatiques sont considérées bien détectées si elles se trouvent à moins de 50cm d'une ligne de la vérité terrain et des faux positifs si elles sont plus loin. Cette stratégie d'évaluation est identique à celle utilisée dans [112, 118, 115], ce qui permet des comparaisons entre les différentes approches. Les mesures classiques de *precision* et *recall* (complétude) sont calculées. Le *recall* est défini comme la longueur des lignes extraites correctement  $l_{OK}$  (selon la distance à la vérité terrain) divisé par la longueur des lignes de référence  $l_{GT}$  ( $R = \frac{l_{OK}}{l_{GT}}$ ). La *precision* est définie comme le rapport entre la longueur des lignes extraites correctement et la longueur des lignes détectées ( $P = \frac{l_{OK}}{l_{detecte}}$ ).

La figure 2.36 montre notre détection automatique de bords de trottoir sur les trois sites. La table 2.4 donne la *precision*, le *recall* et le temps de calcul.

Nous constatons que notre méthode est plus rapide et présente un faible taux de fausses alarmes. D'une part, la *precision* est supérieure à 90% pour les trois sites, ce qui montre le faible taux de fausses alarmes. Celles-ci sont dues principalement à la végétation (Zones H et I sur la figure 2.36b). Par ailleurs, la *precision* est jusqu'à 7% meilleure par rapport aux travaux rapportés dans la littérature sur la même base de données. D'autre part, le *recall* sur les sites 2 et 3 sont meilleurs que ceux rapportés dans la littérature. Seulement le *recall* du site 1 est moins bon. La raison de ce faible *recall* est due aux bordures au milieu de la route (Zones A and B in 2.36a). En effet, le système d'acquisition est passé d'un seul côté de la rue, la bordure de l'autre côté n'est donc pas visible sur le nuage de points. Notre méthode ne traite pas les trottoirs polygonaux en aucune manière alors que les scores publiés dans [118] prennent en compte ces polygones. Un autre problème du site 1 est dû à la rampe d'accès très longue (environ 45 m) que nous n'avons pas reconnectée étant donné ses grandes dimensions.

TABLE 2.4 – *Precision*, *recall* et temps de traitement sur les trois sites de la base Enschede, aux Pays Bas. Entre parenthèses les résultats rapportés dans [118]

	Site 1	Site 2	Site 3
<i>Precision</i>	95% (91%)	94% (92%)	91% (84%)
<i>Recall</i>	65% (83%)	54% (53%)	60% (54%)
Time	8.6 min (1 heure)		



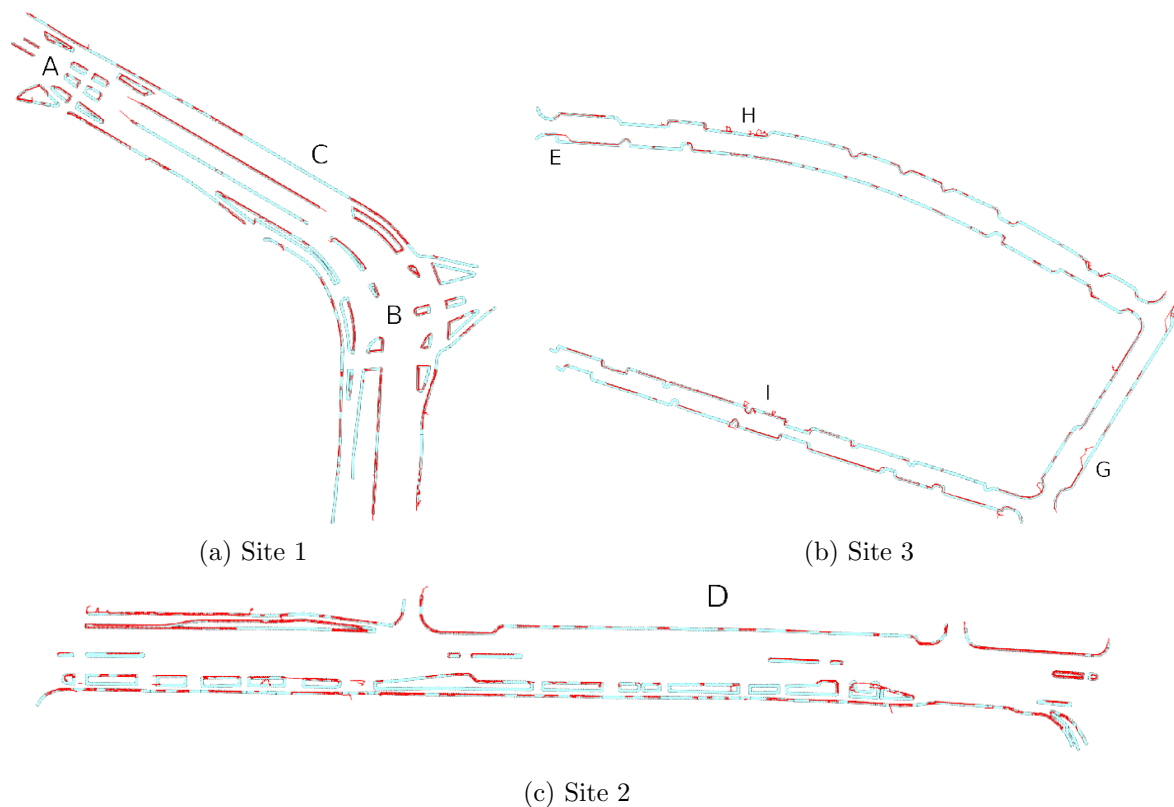


FIGURE 2.36 – Détection de bords de trottoir sur la base *Enschede*. Notre détection en rouge et la vérité terrain en cyan.

Le traitement a été effectué sur un Intel Core i7 @2.93 GHz avec 8 GB RAM. Notre méthode prend moins de 9 minutes pour traiter les 3 sites, sept fois plus rapide que la méthode proposée dans [112] qui réalise le traitement en 1 heure. La différence conceptuelle est liée au fait que [112] traite le nuage de points 3D ligne par ligne alors que nous projetons tous les points 3D sur une image d'élévation et nous traitons l'information 2D.

La table 2.5 permet une analyse plus fine, selon le type de trottoir. Le meilleur *recall* est obtenu pour les bords de trottoirs de plus de 7cm. Les faibles valeurs de *recall* sont dus en général aux occultations, qui affectent de manière similaire à tous types de trottoirs et aux polygones dont un seul côté est visible, qui nécessiteraient un traitement particulier.

{Publications associées : ConfI{ [47], [119].}, ACLI{ [115], [48], [120]  
ACLN{ [93] }

TABLE 2.5 – *Recall* pour chaque type de trottoir de la base *Enschede*.

	Site 1	Site 2	Site 3
Route (hauteur > 7cm)	82%	67%	64%
Gaps (hauteur < 7cm)	55%	48%	46%
Polygones	46%	45%	N/A

- [82] J. Hernández and B. Marcotegui. Point Cloud Segmentation towards Urban Ground Modeling. URBAN’09, 5th GRSS/ISPRS Joint workshop on remote sensing and data fusion over urban areas, Shangai, China. May 2009.
- [47] A. Serna and B. Marcotegui. "Attribute controlled reconstruction and adaptive mathematical morphology.", Proceedings of 11th International Symposium on Mathematical Morphology, May 27-31 2013, Uppsala, Sweden. pp 207-218
- [119] A. Serna, B. Marcotegui, F. Goulette, J.E. Deschaud. "Paris-rue-Madame database : a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods.", 4th International Conference on Pattern Recognition, Applications and Methods, March 2014, Angers, France.
- [121] D Craciun, A. Serna-Morales, J. E. Deschaud, B. Marcotegui, & F. Goulette. (2014, September). Scalable and Detail-Preserving Ground Surface Reconstruction from Large 3D Point Clouds Acquired by Mobile Mapping Systems. In PCV2014 (Photogrammetric Computer Vision) (Vol. 3, pp. 73-80). Zurich, Switzerland.
- [122] M. Brédif, B. Vallet, A. Serna, B. Marcotegui, N. Paparoditis. "TERRA-MOBILITA/IQMULUS Urban point cloud classification benchmark.", Workshop on Processing Large Geospatial Data. Data, 8th July 2014, Cardiff, UK.
- [123] A.K. Aijazi, A. Serna, B. Marcotegui, P. Checchin, and L. Trassoudaine. Segmentation and Classification of 3D Urban Point Clouds : Comparison and Combination of Two Approaches. 10th Conference on Field and Service Robotics. 2015
- [93] J. Hernández and B. Marcotegui. Segmentation et Interprétation des Nuages de Points pour la Modélisation d’Environnements Urbains. Revue française de Photogrammétrie et de Télédétection. n.191. mai 2008. pp. 28-35.
- [115] A. Serna and B. Marcotegui. "Urban accessibility diagnosis from mobile laser scanning data". ISPRS Journal of Photogrammetry and Remote Sensing, Volume 84, October 2013, Pages 23-32 (2013). **meilleur papier IS-**

**PRS 2013.** <http://www.isprs.org/society/awards/helava/2013.aspx>.

- [48], A. Serna, B. Marcotegui, E. Decenci re, Th. Baldeweck, A.M. Pena, S. Brizion. "Segmentation of elongated objects using attribute profiles and area stability : application to melanocyte segmentation in engineered skin". Pattern Recognition Letters, Volume 47, 1 October 2014, Pages 172-182.
- [120] A. Serna and B. Marcotegui. "Detection, segmentation and classification of 3D urban objects using morphological and learning techniques". ISPRS Journal of Photogrammetry and Remote Sensing (2014), Volume 93, July 2014, Pages 243-255.
- [124] B. Vallet, M. Br dif, A. Serna, B. Marcotegui, N. Paparoditis. (2015). TerraMobilita/iQmulus urban point cloud analysis benchmark, Computers & Graphics, Volume 49, June 2015, Pages 126-133.



# Chapitre 3

## Conclusion et perspectives

Ce manuscrit présente mes travaux de recherche organisés en deux parties : contributions méthodologiques à la segmentation morphologique et application à l'analyse de scènes urbaines. D'un point de vue méthodologique, j'ai développé des approches de segmentation morphologique à l'aide des graphes. Dans un premier temps, j'ai utilisé l'arbre de poids minimum (APM). Cette structure contient toute l'information nécessaire pour simuler un processus d'inondation. Cela m'a permis de travailler au niveau de la région au lieu du niveau du pixel, donnant lieu à des méthodes efficaces en temps de calcul. L'efficacité de ces méthodes a été démontrée dans le contexte de la segmentation interactive qui requiert un temps de réponse rapide compatible avec le temps d'attente réduit souhaité par l'utilisateur. J'ai ainsi, dans le cadre du projet MoMuSys, développé une interface graphique pour la segmentation interactive intégrant plusieurs outils d'édition s'appuyant sur l'APM, y compris pour les séquences d'images définissant des régions spatio-temporelles. L'algorithme de cascades a également été revisité et j'ai proposé une version fondée sur l'APM, donnant lieu à un algorithme efficace, compatible avec les applications temps réel telle que la détection des bords de route non structurées pour le compte de Thales.

Ces méthodes posent néanmoins un problème de paramétrisation : on doit en effet choisir l'ensemble des marqueurs initiaux, le nombre de régions à segmenter ou le nombre d'itérations des méthodes hiérarchiques, comme celle des cascades. Ce choix n'est pas toujours facile dans le cadre général. Dans un deuxième temps, j'ai donc cherché à mettre au point des méthodes non paramétriques de segmentation, qui s'adaptent au contenu de l'image. Je me suis orientée vers les méthodes résiduelles, telle que l'ouverture ultime, qui extraient les régions les plus contrastées d'une image en analysant l'évolution de celle-ci quand on lui applique une série d'ouvertures. J'ai en particulier introduit l'ouverture ultime avec des ouvertures par attribut, un opérateur générique et puissant. J'ai montré qu'il produit des résultats intéressants dans des applications variées. Par ailleurs, l'utilisation de l'arbre de maxima m'a permis de proposer un algorithme efficace de calcul de cet opérateur. J'ai ensuite proposé une stratégie générale faisant ressortir les régions d'une certaine forme. Pour ceci, les résidus sont pondérés

par une fonction de similarité des régions à une forme de référence.

Depuis 2006, je me suis intéressée à une nouvelle source d'information : les nuages de points 3D, acquis par des systèmes de cartographie mobiles en milieu urbain. Ces données contiennent une information géométrique très riche mais son exploitation pose plusieurs problèmes. A la différence des images classiques, l'échantillonnage n'est pas régulier et il n'existe pas une définition de voisinage bien établie. Ces données sont souvent très denses et à grande échelle. Des méthodes efficaces en temps de calcul sont de rigueur. Nous avons proposé une stratégie combinant la morphologie mathématique et l'image d'élévation. Les contributions dans ce domaine sont les suivantes :

- segmentation en îlots, sans contraintes d'alignement ni de forme de l'empreinte des bâtiments. Néanmoins, une hauteur et une longueur minimales sont imposées, ce qui n'est pas contraignant dans un contexte d'immeubles mais nécessiterait une adaptation en zone pavillonnaire.
- segmentation sol-façade : une approche de propagation par zone quasi-plates est proposée. C'est une méthode plus générique et moins paramétrique que les approches de l'état de l'art qui cherchent à ajuster un modèle, soit plan ou polynomial.
- segmentation d'objets 3D : notre méthode s'appuie sur les méthodes de filtrage et segmentation morphologique. L'apport principal par rapport à l'état de l'art est la prise en compte de plusieurs classes d'objets simultanément et la généralité vis-à-vis du système d'acquisition. L'hypothèse d'un maximum significatif par objet est généralement vérifiée mais il y a des exceptions. Par exemple, un vélo garé au pied d'un lampadaire ne sera pas correctement segmenté. Une analyse contextuelle, accompagnée d'une étude de l'image couleur associée permettra d'améliorer les résultats.
- classification d'objets : une caractérisation des objets segmentés suivie d'une application classique de techniques d'apprentissage est proposée. L'étape de segmentation précédente simplifie de manière significative cette phase de classification.
- la détection du bord de trottoir. Nous avons proposé une méthode fondée sur le filtrage géodésique du gradient d'élévation, suivi d'une phase de reconnexion pour tenir compte des rampes d'accès et des zones occultées. Par rapport à l'état de l'art, nous exploitons la connexité des profils d'acquisition voisins, donnant lieu à des approches plus robustes. Par ailleurs, le fait de travailler sur l'image d'élévation (2D) au lieu de directement sur le nuage de points (3D), réduit la complexité de la méthode, aboutissant à des approches plus rapides que celles de l'état de l'art.

Nous avons ainsi démontré l'efficacité de la stratégie proposée, qui se compare favorablement à l'état de l'art aussi bien du point de vue de la qualité que des résultats que des temps de calcul.

Toutes ces contributions ont été intégrées dans un prototype qui a été livré au consortium du projet TerraMobilita. Ce prototype a suscité l'intérêt des autres partenaires du projet et a permis de développer des applications réelles : réaliser des enquêtes

de stationnement, des diagnostics de dégradation du mobilier urbain ainsi que des diagnostics d'accessibilité pouvant s'adapter à différents types de mobilité. Ce dernier point s'inscrit dans le cadre de la Loi 2005-102 du 11 février 2005 : « Pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées ». Toutes ces solutions intéressent les collectivités locales et en particulier le service de voirie de la mairie de Paris, ainsi que Lille métropole où des ateliers d'utilisation de ces techniques auront lieu dans le cadre du projet TerraMobilita. Ces résultats ont atteint une maturité opérationnelle à grande échelle : l'Institut Géographique National (IGN) a obtenu des résultats concluants sur tout le VIème arrondissement parisien s'appuyant sur les outils cités ci-dessus.

### Perspectives

Nous venons de résumer nos contributions dans le domaine de la segmentation d'images et de l'analyse de scènes urbaines. Nous avons accompli des progrès aussi bien d'un point de vue méthodologique qu'applicatif. Des nombreuses pistes d'amélioration sont envisageables et nous énumérons ci-dessous celles qui nous semblent les plus prometteuses à court ou moyen terme.

L'ouverture ultime par attribut est un opérateur versatile et faiblement paramétrique. Le principal paramètre de la méthode est l'attribut utilisé pour définir les ouvertures sur lesquelles repose le calcul des résidus. Nous pouvons généraliser la procédure en combinant plusieurs attributs. Plusieurs pistes seraient à explorer pour cette combinaison. Par exemple les résidus selon les attributs  $H$  et  $W$  pourraient être calculés et la sélection des régions se ferait en maximisant le résidu. Le coût algorithmique d'utiliser plusieurs attributs est marginal car l'étape la plus coûteuse de l'approche est la création de l'arbre. A ce propos, il serait intéressant d'intégrer notamment des mécanismes de parallélisation [125] permettant de tirer parti des architectures modernes multi-cœurs.

Une autre piste d'amélioration concerne la segmentation adaptative. La stratégie proposée où le facteur de pondération est indépendant du calcul des résidus offre une grande flexibilité à cette approche, et ouvre de nombreuses perspectives. Nous avons pour l'instant utilisé des descripteurs de forme simples, avec un faible coût de calcul. Si le temps de calcul n'impose pas des contraintes fortes, des descripteurs plus évolués peuvent être utilisés. Par exemple, des techniques d'apprentissage permettent d'estimer la probabilité d'une région d'appartenir à une classe de formes. Cette probabilité peut alors pondérer les résidus, de façon à favoriser la sélection de ces régions dans la procédure de segmentation. D'autres caractéristiques, comme la stabilité des régions, définie par la populaire méthode de segmentation MSER, peuvent intervenir dans le processus de segmentation. Cette caractéristique tient compte du changement de surface entre deux ouvertures successives pour sélectionner les régions les plus stables d'une image.

Enfin, j'utilise seulement l'information de luminance. Des stratégies permettant d'utiliser l'information couleur peuvent apporter un gain significatif de qualité de résultats. Des techniques récentes proposent la création d'arbres représentant les images couleur [126]. A partir de ces arbres nous pouvons mettre en place la méthodo-

logie résiduelle définissant les résidus en fonction des distances couleur perceptuelles.

Dans le domaine de l'analyse de scènes urbaines, une première amélioration à apporter est d'augmenter le nombre de classes d'objets à traiter. Cela entraîne une complexité accrue liée à la diversité d'objets qu'on peut trouver dans l'espace public. Les pistes suivantes mériteraient d'être explorées pour faire face à cette complexité. Actuellement nous utilisons principalement des informations géométriques. D'autres sources d'information sont disponibles et devraient être mises à contribution. D'une part l'information couleur et/ou l'intensité du laser rajoute une dimension intéressante au problème et permet de lever certaines ambiguïtés des données 3D. Les données étant calibrées, chaque point 3D dispose de sa couleur et de son intensité. Il est donc direct d'intégrer ces informations comme descripteurs supplémentaires dans l'étape de classification. En ce qui concerne l'étape de segmentation ces informations devraient être combinées pour définir un gradient plus performant/pertinent. D'autre part, les systèmes d'acquisition en temps réel, type Velodyne, permettent une meilleure gestion des occultations mais aussi une augmentation du bruit et une géométrie plus complexe des objets en mouvement. La prise en compte de la dimension temporelle, mettant à jour régulièrement les cartes urbaines, permettrait d'améliorer la qualité de l'analyse en exploitant la redondance des acquisitions successives. Par exemple, les techniques de détection de changement pourraient cibler les zones à analyser par l'étape de segmentation et classification. Par ailleurs, des informations contextuelles peuvent être pertinentes. Nous avons déjà utilisé une information contextuelle : nous avons augmenté la longueur de reconnexion du bord de trottoir si la zone se trouve occultée par les voitures. D'autres règles contextuelles peuvent être introduites. Par exemple : une bicyclette peut être attachée au pied d'un lampadaire, les motos sont souvent garées en batterie, les arbres sont souvent alignés et de taille similaire, à côté d'un abribus il y a des piétons...

Nous avons proposé une méthode très simple et efficace pour résoudre les problèmes des vitesses variables du véhicule d'acquisition. Cette méthode mériterait d'être approfondie pour définir une taille du pixel adaptative localement à la scène en fonction de la vitesse et de la distance au capteur.

Les profils d'attributs utilisés pour la segmentation d'objets allongés, en particulier les façades, est un outil flexible et performant. Appliqués sur l'image d'élévation, chaque catégorie d'objet ou ensemble d'objets (dans une logique contextuelle) aurait un profil caractéristique qui peut servir à prendre la décision de resegmenter ou pas un objet. Cette analyse de profil fournirait une hypothèse de type d'objet rencontré. Elle pourrait être ensuite corroborée ou invalidée en s'appuyant sur d'autres sources d'information.

Nos méthodes d'analyse de nuages de points permettent d'ores et déjà de développer des applications à haut contenu sémantique. Dans l'optique d'une industrialisation, il serait néanmoins nécessaire de rajouter une étape de validation/vérification des résultats. Pour cela, une estimation de la qualité de détection serait nécessaire, celle-ci pouvant s'appuyer par exemple sur la différence d'élévation de l'objet par rapport aux voisins, le pourcentage de points interpolés par rapport aux points mesurés ou la probabilité d'appartenance à une classe lors de l'étape de classification. Une approche de



segmentation interactive, inspirée des méthodes décrites dans le premier chapitre de ce manuscrit, pourrait être utilisée pour corriger les zones avec une faible fiabilité.



# Annexe A

## Curriculum Vitae

**Beatriz MARCOTEGUI ITURMENDI**  
**Ingénieur en Télécommunications**  
**Docteur en Morphologie Mathématique**

Née le 08/07/1968 à Pamplona (Espagne)  
Nationalité espagnole  
Mariée, 4 enfants

ADRESSE :  
35, rue St Honoré  
77305 Fontainebleau Cedex  
e-mail : [beatriz.marcotegui@mines-paristech.fr](mailto:beatriz.marcotegui@mines-paristech.fr)

### FORMATION

- 1993-1996 : Formation doctorale dans le domaine du codage de séquences d'images au Centre de Morphologie Mathématique de l'Ecole des Mines de Paris.
- 1986-1992 : Diplôme d'ingénieur en télécommunications à l'Université Polytechnique de Catalogne à Barcelone (Espagne).

### DOMAINES DE COMPÉTENCE

- Analyse et traitement d'images et de vidéo, Morphologie Mathématique.
- Filtrage, segmentation, segmentation interactive.
- Indexation d'image basée sur le contenu.
- Télécommunications.

## EXPÉRIENCE PROFESSIONNELLE

### Chargée de recherche puis maître de recherche au Centre de Morphologie Mathématique (depuis 1998).

- Développement d'un système temps réel de suivi de bords de route pour Thales Airborne Systems (Septembre 2002 -Septembre 2003)
- Validation des résultats et documentation d'un système de segmentation interactive de séquences vidéo pour TDK (2000-2002).
- Responsable du développement d'un logiciel de génération d'objets vidéo, dans le cadre du projet européen MOBILE MULTIMEDIA SYSTEMS (MoMuSys-WP5.2. 1998-2000).
- Responsable des activités d'indexation au CMM depuis Mai 2004, dans le cadre du réseau d'excellence MUSCLE (Multimedia Understanding through Semantics, Computation and Learning) du 6ème PCRD.
- Développement d'un système de localisation de texte enfoui, remportant la première position dans le cadre du projet ImagEVAL 2006, du programme TechnoVision. Le but d'ImagEVAL était d'évaluer les technologies de filtrage et d'indexation d'images, de recherche d'images par le contenu dans des bases de données constituées par les détenteurs de fonds professionnels.
- Développement d'un système d'analyse sémantique de façades Hausmaniennes et d'un système d'interprétation de nuages de points 3D, dans le cadre du projet TerraNumerica (2006-2009). Le but de TerraNumerica était d'augmenter la productivité et le réalisme de la modélisation 3D urbaine.
- Développement d'un système de binarisation de documents, remportant la deuxième place dans la compétition internationale « Document Image Binarization Contest- DIBCO 2009 ».
- Développement d'un système de localisation de texte en milieu urbain, dans des systèmes de navigation enrichis, dans le cadre du projet iTowns (2007-2010).
- Cartographie 3D de la voirie et de l'espace public urbains accessibilité et circulations douces (TerraMobilita, 2011-2015).
- Intégration et maintenance de la librairie de traitement d'image Xlim3D du CMM.

### Enseignement au CMM

- Co-responsable de l'option Vision et Morphologie (1999-2008) et de l'option MA-REVA de MINES ParisTech (2008-Présent)
- Montage d'une formation en segmentation avancée, adaptée aux besoins d'une équipe de recherche de la société Michelin (2007-2008).
- Responsable de l'enseignement spécialisé « Analyse d'image, de la théorie à la pratique » à MINES ParisTech depuis 2008.
- Responsable d'une école d'été de 5 jours à destination d'industriels et de cher-

cheurs extérieurs (2008-2012).

### **Professeure Associée à l'Université Publique de Navarre (Octobre 1996-Juin 1998).**

- Enseignement à l'Ecole d'Ingénieurs en Télécommunications.
- Développement du module de "eye tracking" dans le cadre du projet SIVHA synthèse visuelle de la parole (1996-1998).
- Participation au développement d'un système de télésurveillance dans le cadre du projet "Televigilancia inteligente" financé par le Comité Interministériel de Science et Technologie Espagnol (CICYT).

### **Doctorant au Centre de Morphologie Mathématique (1992-1996)**

- Développement d'un système de codage de séquences vidéo dans le cadre des projets européens MORPHECO, MAVT et COST 211ter, en collaboration avec des laboratoires universitaires et des partenaires industriels.

## **DISTINCTIONS**

- Nominée au prix à la meilleure thèse de l'Ecole des Mines de Paris 1996.
- Première position dans la compétition de localisation de texte ImagEVAL 2006.
- Deuxième place dans la compétition internationale « Document Image Binarization Constest- DIBCO 2009 ».
- Chevalier dans l'ordre des Palmes académiques.
- Meilleur papier ISPRS Journal of Photogrammetry and Remote Sensing 2013 : "Urban accessibility diagnosis from mobile laser scanning data", ISPRS 84, 23-32 (2013). The U. V. Helava Award- meilleur papier 2013.



# Annexe B

## Publications

Les articles dans des revues internationales indexées JCR (Journal Citation Reports) sont indiqués par un **(JCR)** au début de la référence correspondante. Mon nom est en gras pour mieux le repérer dans la liste des auteurs. Il est souligné quand j'ai présenté oralement le travail.

### B.1 Articles de journaux

1. B. Vallet, M. Brédif, A. Serna, **B. Marcotegui**, N. Paparoditis, TerraMobilita/i-Qmulus urban point cloud analysis benchmark, *Computers & Graphics*, Volume 49, Pages 126-133, June 2015.
2. **(JCR)** X. Zhang, G. Thibault, E. Decencièrre, **B. Marcotegui**, B. Lay, R. Danno, G. Cazuguel, G. Queller, M. Lamarde, P. Massin, A. Chabouis, Z. Victor, A. Erginay. "Exudate Detection in Color Retinal Images for Mass Screening of Diabetic Retinopathy". *Medical Image Analysis*, Volume 18, number 7, pages 1026 - 1043, 2014.
3. **(JCR)** A. Serna, **B. Marcotegui**, E. Decencièrre, Th. Baldeweck, A.M. Pena, S. Brizion. "Segmentation of elongated objects using attribute profiles and area stability : application to melanocyte segmentation in engineered skin". *Pattern Recognition Letters*, Volume 47, Pages 172-182, October 2014.
4. **(JCR)** A. Serna and **B. Marcotegui**. "Detection, segmentation and classification of 3D urban objects using morphological and learning techniques". *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 93, Pages 243-255, 2014.
5. **(JCR)** A. Serna and **B. Marcotegui**. "Urban accessibility diagnosis from mobile laser scanning data". *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 84, October 2013, Pages 23-32, 2013.
6. **(JCR)** J. Fabrizio, **B. Marcotegui**, M. Cord. Text Detection in Street Level Images. *Pattern Analysis and Applications*, Volume 16, Issue 4, pp 519-533 November 2013.

7. E. Decencière, G. Cazuguel, X. Zhang, G. Thibault, J.C. Klein, F. Meyer, B. Marcotegui, G. Quellec, M. Lamard, R. Danno, D. Elie, P. Massin, Z. Viktor, A. Erginay, B. Lay and A. Chabouis. TeleOphta : Machine learning and image processing methods for teleophthalmology. *IRBM*, 34(2), 196-203, 2013.
8. **(JCR)** J. Hernández and **B. Marcotegui**. Shape Ultimate Attribute Opening. *Image and Vision Computing. Image Vision Computing*, vol. 29, n. 8. pp : 533–545. July 2011.
9. **(JCR)** **B. Marcotegui** and Salembier, P., Compression, in *Mathematical morphology from theories to applications*, L. Najman and H. Talbot (Eds.), Wiley, 2010, pp. 385-391.
10. **(JCR)** A. Hanbury, **B. Marcotegui**. Morphological segmentation on learned boundaries. *Image Vision Computing*, vol. 27, n. 4. pp : 480–488. March 2009.
11. J. Hernández and **B. Marcotegui**. Segmentation et Interprétation des Nuages de Points pour la Modélisation d’Environnements Urbains. *Revue française de Photogrammétrie et de Télédétection*. n.191. mai 2008. pp. 28-35.
12. **(JCR)** E. Decencière Ferrandière, **B. Marcotegui** and F. Meyer. Content dependent image sampling using mathematical morphology : application to texture mapping. *Signal Processing : Image Communication*, vol. 16, issue 6, pages 567-584, February 2001.
13. **(JCR)** P. Salembier, F. Marqués, M. Pardàs, R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, **B. Marcotegui**. Segmentation-based video coding system allowing the manipulation of objects. *IEEE Transactions on Circuits and Systems for Video Technology, (RACE/MAVT and MORPHECO Projects)*, 7, 1. February 1997, pp. 60-74.
14. **B. Marcotegui**, F. Marqués, F. Meyer. Allowing content-based functionalities in segmentation-based coding schemes. *Annales des Télécommunications, Numéro spécial "Compression and image processing"*, Vol. 52, No 7-8, July-August 1997.
15. **B. Marcotegui**, F. Marqués, R. Morros, M. Pardàs, P. Salembier. Segmentation of video sequences and rate control. *Annales des Télécommunications, Numéro spécial "Compression and image processing"*, Vol. 52, No 7-8, July-August 1997.
16. **B. Marcotegui**, F. Meyer. Bottom-up segmentation of image sequences for coding. *Annales des Télécommunications, Numéro spécial "Compression and image processing"*, Vol. 52, No 7-8, July-August 1997.

## B.2 Articles de Conférence

1. A.K. Aijazi, A. Serna, **B. Marcotegui**, P. Checchin, and L. Trassoudaine. Segmentation and Classification of 3D Urban Point Clouds : Comparison and Combination of Two Approaches. 10th Conference on Field and Service Robotics. 2015



2. **Th. Chabardès, B. Marcotegui**. Local blur estimation based on toggle mapping”. Proceedings of 12th International Symposium on Mathematical Morphology, May 2015, Reikjavik, Iceland.
3. M. Brédif, B. Vallet, A. Serna, **B. Marcotegui**, N. Paparoditis. ”TERRAMOBILITA/IQMULUS Urban point cloud classification benchmark.”, Workshop on Processing Large Geospatial Data. Data, 8th July 2014, Cardiff, UK.
4. A. Serna, **B. Marcotegui**, F. Goulette, J.E. Deschaud. ”Paris-rue-Madame database : a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods.”, *4th International Conference on Pattern Recognition, Applications and Methods*, March 2014, Angers, France.
5. A. Serna and **B. Marcotegui**. ”Attribute controlled reconstruction and adaptive mathematical morphology.”, *Proceedings of 11th International Symposium on Mathematical Morphology*, May 27-31 2013, Uppsala, Sweden. pp 207-218
6. A. Serna, J. Hernández and **B. Marcotegui**. Adaptive Parameter Tuning For Morphological Segmentation of Building Façade Images. *Proceeding of 20th European Signal Processing Conference (EUSIPCO 2012)*, August 27-31 2012, Bucharest, Romania. pp. 2268-2272.
7. **B. Marcotegui** and J. Hernández and T. Retornaz. Ultimate Opening and Gradual Transitions. ISMM’2011, *Proceedings of 10th International Symposium on Mathematical Morphology*, 6th-8th July 2011 - Intra, Lake Maggiore, Italy.
8. R. Minetto, N. Thome, M. Cord, J. Fabrizio, **B. Marcotegui**. SNOOPERTEXT : A Multiresolution system for text detection in complex visual scenes. *ICIP10 - IEEE International Conference on Image Processing 2010*, Hong Kong.
9. J. Hernández and **B. Marcotegui**. Segmentation et Interprétation des Nuages de Points pour la Modélisation d’Environnements Urbain. *Colloque Techniques laser pour l’étude des environnements naturels et urbains*, Janvier 20 -21,2009, Le Mans, France.
10. J. Hernández and **B. Marcotegui**. Point Cloud Segmentation towards Urban Ground Modeling. *URBAN’09, 5th GRSS/ISPRS Joint workshop on remote sensing and data fusion over urban areas*, Shangai, China. May 2009.
11. J. Hernández and **B. Marcotegui**. Ultimate Attribute Opening Segmentation with Shape Information. *ISMM’09 Proceedings of the nine International Symposium on Mathematical Morphology*, pp : 205–214, Groningen, The Netherlands, August 24-27, 2009.
12. J. Hernández and **B. Marcotegui**. Filtering of Artifacts and Pavement Segmentation from Mobile LiDAR Data. *LaserScanning’09*, Paris, France. September 2009.
13. J. Hernández and **B. Marcotegui**. Morphological Segmentation of Building Façade Images. *ICIP09 - IEEE International Conference on Image Processing 2009*. Cairo, Egypt.

14. J. Fabrizio, **B. Marcotegui**, M. Cord. Text Segmentation in Natural Scenes Using Toggle-Mapping. *ICIP09 - IEEE International Conference on Image Processing 2009*. Cairo, Egypt.
15. J. Fabrizio, M. Cord, **B. Marcotegui**. Text Extraction from Street Level Images, *CMRT09 - CityModels, Roads and Traffic 2009*. Paris, France.
16. J. Fabrizio, **B. Marcotegui**. Fast Implementation of the Ultimate Opening. *ISMM '09 : Proceedings of the 9th International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing 2009*. pp. 272–281. Groningen, The Netherlands.
17. J. Hernández and **B. Marcotegui**. Ultimate Opening Segmentation with Shape Constraints. *The 8th IASTED International Conference on Visualization, Imaging, and Image Processing - 2008*, 630-120, September 1-3, 2008, Palma de Mallorca, Spain.
18. T. Retornaz, **B. Marcotegui**. Scene-text localization based on ultimate opening. *International Symposium on Mathematical Morphology ISMM'07*. October 10-13, 2007. Rio de Janeiro. Bresil.
19. T. Retornaz, **B. Marcotegui**. Ultimate opening implementation based on a flooding process. *In the 12th International Congress for Stereology, ICS XII*, September 2007, Saint-Etienne, France.
20. A. Hanbury, **B. Marcotegui**. Colour Adjacency Histograms for Image Matching. *In 12th International Conference on Computer Analysis of Images and Patterns*, Vienna, Austria, August 27th-29th, 2007. CAIP 2007 p 424-431
21. A. Hanbury, **B. Marcotegui**. Waterfall segmentation of complex scenes. *In Asian Conference on Computer Vision - ACCV 2006*, January 13-16 2006, Hyderabad, India. p. 888-897.
22. J. Angulo and **B. Marcotegui**. Sur l'influence des conditions d'éclairage dans la segmentation morphologique couleur par LPE . *In Actes de CORESA 2005 (Compression et Représentation des Signaux Audiovisuels)*, Rennes, France, November 2005.
23. **B. Marcotegui**, S. Beucher. Fast implementation of waterfall based on graphs. *in Mathematical Morphology : 40 Years on : Proc. 7th ISMM*. Paris, April 2005. Ronse C., Najman L., and Decencière E., Eds. Springer : p. 177-186.
24. V. Marion, O. Lecoïnte, C. Lewandowski, J.G. Morillon, R. Aufrère, **B. Marcotegui**, R. Chapuis, S. Beucher. Robust perception algorithm for road and track autonomous following. *Unmanned ground vehicle technology, Conference No 6*, vol. 5422, pp. 55-66, Orlando FL, USA, 2004.
25. **B. Marcotegui**, F. Zanoguera. Image editing tools based on multi-scale segmentation. In H. Talbot and R. Beare (eds.), *Mathematical Morphology VI*, (Proc. ISMM02, Sydney, Australia, April 2002). CSIRO, pp. 127-135.

26. F. Marqués, **B. Marcotegui**, F. Zanoguera, P. Correia, R. Mech, and M. Wollborn. Partition-Based Image Representation as Basis for User-Assisted Segmentation, *Int. Conference on Image Processing (ICIP 2000)*, Vancouver, Canada, September 10-13, 2000, pp. 312-315
27. F. Zanoguera, **B. Marcotegui** and F. Meyer. A segmentation pyramid for the interactive segmentation of 3-D images and video sequences, *n proceedings of the International Symposium of Mathematical Morphology*, Palo Alto, California, USA, 2000, pages 263-272. Kluwer Academic Publishers, 2000.
28. E. Decencière, **B. Marcotegui** and F. Meyer. Content dependent image sampling using mathematical morphology, in *Mathematical morphology and its applications to image and signal processing (proceedings of the International Symposium of Mathematical Morphology, Palo Alto, California, USA, 2000)* pages 263-272. Kluwer Academic Publishers, 2000.
29. F. Zanoguera, **B. Marcotegui**, F. Meyer. "A Toolbox for Interactive Segmentation Based on Nested Partitions" *ICIP-99*, Kobe (Japan), October 1999.
30. **B. Marcotegui**, P. Correia, F. Marques, R. Mech, R. Rosa, M. Wollborn, F. Zanoguera. "A Video Object Generator Tool Allowing Friendly User Interaction" *ICIP-99*, Kobe (Japan), October 1999.
31. E. Decencière, **B. Marcotegui**, and F. Meyer. Application de la morphologie mathématique au mipmapping. *CORESA '98*, June 1998.
32. Y. Blanco., M. Cuellar, A. Villanueva, F. Lacunza, R. Cabeza, **B. Marcotegui**. "SIVHA, visual speech synthesis system". The 5th International Conference on Spoken Language Processing, November 30th - December 4th, 1998.
33. F. Marqués, **B. Marcotegui** and F. Meyer. Tracking areas of interest for content-based functionalities in segmentation-based video coding. *ICASSP'96*, Atlanta, May 1996.
34. **B. Marcotegui**. Segmentation algorithm by multicriteria region merging. Boston : Kluwer Ac. Publ., *Proceedings ISMM'96*, P. Maragos, R.W. Schafer, M.A. Butt (Eds.), Atlanta, May 1996.
35. **B. Marcotegui** J. Crespo and F. Meyer. Morphological Segmentation Using Texture and Coding Cost. *IEEE Workshop on Non Linear Signal/Image Processing*, June 95, Thessaloniki (Greece).
36. F. Marqués, P. Salembier, M. Pardàs, R. Morros, I. Corset, S. Jeannin, **B. Marcotegui**, F. Meyer. A segmentation-based coding system allowing manipulation of objects (SESAME). *Proc. ICIP'96, (IEEE International Conference on Image Processing)*, Lausanne, September 16-19, 1996, Vol. III. pp 679-682.
37. **B. Marcotegui**, F. Meyer. Morphological Segmentation of Image Sequences. *Workshop on Mathematical Morphology and its Applications to Image Processing*, Sept 94, Fontainebleau.

### B.3 Synthèse de la production scientifique

Le tableau ci-dessous dresse le bilan de mes publications depuis 1997.

La nomenclature est la suivante :

- ACLI : Articles de revues internationales (JCR) avec comité de lecture.
- ACLN : Articles de revues nationales avec comité de lecture.
- ConfI : Congrès internationaux avec actes et comité de sélection sur le texte complet.
- ConfN : Congrès nationaux avec actes et comité de sélection sur le texte complet.

Type	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
ACLI	1				1								1		1		3	3	1+2*
ACLN												1							
ConfI		1	2	3		2		1	1	1	3	1	6	1	1	1	1	2	2
ConfN		1							1				1						

TABLE B.1 – Synthèse de la production scientifique

\* : en révision.

## B.4 Activités d'enseignement

La scolarité du cycle ingénieur civil à MINES ParisTech dure 3 ans et comporte des enseignements de 4 types :

- tronc commun portant sur les mathématiques, la physique, les sciences économiques et sociales...
- enseignements personnalisés incluant les MIG (Modules d'Initiation aux métiers d'ingénieur Généraliste), l'acte d'entreprendre, la mécatronique, le stage d'observation en géologie,...
- les enseignements au choix, ou enseignements spécialisés. Une offre très étendue de cours permet à chaque élève de construire un parcours "à la carte", en fonction de ses intérêts.
- les activités d'option comprenant 2 semaines en deuxième année de prise de contact avec la discipline, un mois d'approfondissement au début de la troisième année et 4 mois de stage en entreprise.

L'option Vision et Morphologie, portant sur le traitement d'images et la morphologie mathématique, existait à l'Ecole depuis très longtemps. J'ai été co-responsable de cette option, avec Fernand Meyer, de 1999 à 2008. Etant donné le nombre important d'options (une vingtaine) pour une promotion d'une centaine d'élèves, nous avons décidé de fusionner les options autour des mathématiques appliquées : Automatique, Robotique et Vision et Morphologie ; L'option MAREVA - Mathématiques Appliquées, Robotique, Vision et Automatique, est ainsi née en 2008. L'intérêt de cette fusion est multiple : proposer des compétences dans plusieurs disciplines pour répondre à des projets multidisciplinaires (robotique médicale, aide à la conduite, par exemple) et mutualiser les efforts évitant les promotions à trop faible effectif. Actuellement, un total de 15 options est proposée aux élèves ingénieurs civils. Le cours de morphologie mathématique est devenu un enseignement spécialisé lors de la création de l'option MAREVA. Cela a permis d'ouvrir le cours à des élèves d'autres options qui ont affaire à des images telle les l'options matériaux ou bio-technologie.

J'ai ainsi assumé les tâches suivantes liées à l'enseignement :

- Co-responsable de l'option Vision et Morphologie (1999-2008). L'organisation de l'option s'étend sur 6 semaines (2 en deuxième année et 4 en troisième). Elle comprend 4 semaines de cours (introduction à la morphologie mathématique, segmentation avancée, compression, indexation), un voyage d'option d'une semaine et une semaine de mini-projets travaillant sur un projet en binôme. Les promotions sont petites, entre 3 et 4 élèves.
- Co-responsable, avec Brigitte d'Andréa-Novel, de l'option MAREVA - Mathématiques Appliquées Robotique, Vision et Automatique, prenant en charge la responsabilité de la partie vision. (2008-Présent). Dans le cadre de MAREVA j'assure deux journées de cours (compression, imagerie médicale, estimation de mouvement), deux semaines de mini-projets (une en deuxième année et autre en troisième) et l'organisation du voyage d'option d'une semaine. Promotions d'entre

10 et 15 élèves.

- Responsable de l’enseignement spécialisé « Analyse d’image, de la théorie à la pratique » depuis 2008. (35 heures : 50% cours, 50% travaux dirigés). Environ 20 élèves ingénieurs inscrits, auxquels on ajoute les nouveaux arrivants au CMM (doctorants ou post-docs).
- Responsable d’une école d’été de 5 jours (35 heures : 50% cours, 50% travaux dirigés) à destination d’industriels et de chercheurs extérieurs (2008-2012). Environ 10 participants chaque année.
- Montage d’une formation en segmentation avancée, adaptée aux besoins d’une équipe de trois chercheurs de la société Michelin. (2007-2008). 3 modules de 3 jours (50% cours, 50% travaux dirigés).

# Annexe C

## Activités d'encadrement

### C.1 Encadrement de doctorants

#### 1. Francisca ZANOQUERA

Thèse menée dans le cadre du projet européen MoMuSys, intitulée : "Segmentation interactive d'images fixes et de séquences vidéo basée sur des hiérarchies de partitions".

Début & fin	Noms et % encadrement des co-directeurs
Septembre 1998 - 13 Décembre 2001.	F. Meyer (60%) - B. Marcotegui (40%)

**Publications associées : ConfI{ [22], [23], [25], [28], [27], [27].}**

Situation actuelle : Expedia Inc. Genève-Suisse. Analyse des données et stratégie.

#### 2. Thomas RETORNAZ

Encadrement de la thèse de Thomas Retornaz, dans le cadre du projet EADS-SCOOOP et ImagEval. "Détection de textes enfouis dans des bases d'images généralistes : un descripteur sémantique pour l'indexation".

Début & fin	Directeur de thèse *
Septembre 2003 - 23 Octobre 2007.	B. Marcotegui

\* Seul encadrant, sans HDR, en accord avec les règles de l'Ecole Doctorale numéro 431 : Inforamtion, Communication, Modélisation et Simulation en 2007.

**Publications associées : ConfI{ [66], [44], [65].}**

Situation actuelle : Manager R&D A2iA.

#### 3. Jorge HERNANDEZ

Encadrement de la thèse de Jorge Hernández, dans le cadre du projet TerraNumerica. "Analyse morphologique d'images pour la modélisation d'environnements urbains".

Début & fin	Noms et % encadrement des co-directeurs
Janvier 2006 - 14 Décembre 2009.	B. Marcotegui (80%) - D. Jeulin (20%)

**Publications associées : ConfI**{ [81], [82], [83], [45], [79], [65].}, **ACLI**{ [84] }. **ACLN**{ [93]}

Situation actuelle : Ingénieur R&D Traitement Signal/Image, R&T Center, Safran SA.

#### 4. Andrés SERNA

Encadrement de la thèse de Andrés Serna, dans le cadre du projet TerraMobilita.

Début & fin	Directeur de thèse *
Novembre 2011 - 16 Décembre 2014.	B. Marcotegui

\* Seul encadrant, sans HDR, en tant que Maître de recherche en accord avec l'Ecole Doctorale SMI - Sciences des Métiers de l'Ingénieur.

**Publications associées : ConfI**{ [79], [47], [119], [122], [123].}, **ACLI**{ [115], [48], [120], [124]}.

Situation après la thèse : Ingénieur R&D chez Trimble.

## C.2 Encadrement de post-doctorants

- Encadrement du post-doc de Jonathan Fabrizio (Septembre 2007-Décembre 2008), dans le cadre du projet iTowns. Situation actuelle : Enseignant chercheur au Laboratoire de Recherche et Développement de l'EPITA (LRDE).
- Encadrement du post-doc de Amira Belhedi (Janvier-Juillet 2015), dans le cadre du projet LINX.
- Encadrement du post-doc de Luc Gillibert (Juillet-Décembre 2015), dans le cadre du projet LINX.

## C.3 Encadrement de stagiaires

- Encadrement des stages suivants : Myriam Huberman "Capteur d'environnement destiné à un service télématique"(PSA-2005), Mercè Almuni "Indexation d'images médicales" (CEA-LIST 2006), Pierre Bertrand "Automatic Image Annotation" (LTU-2006), Victor FUNG "Detecting and tracking players in sports scenes by image analysis" (Grass Valley-2007), Youssouf Chherawala (valeo + MUSCLE-2007), Simon Le Moal "Quality control of photovoltaic glass through mechanical test and automatic optical inspection "(Saint Gobain 2008), Fawzi Houfaf "Développement d'une application de traitement d'image pour la détection et le suivi de cibles volantes" (thales-2008), Marion Scoazec "Quantification des caractéristiques de bâtiments à partir d'images" (EDF 2015).



- Suivi des nombreux stages d'option MAREVA.



# Bibliographie

- [1] S. Beucher, C. Lantuéjoul, Use of watersheds in contour detection, in : International Workshop on Image Processing : Real-time Edge and Motion Detection/Estimation, Rennes, France., 1979.
- [2] F. Meyer, S. Beucher, Morphological segmentation, *Journal of Visual Communication and Image Representation* 1 (1) (1990) 21–46.
- [3] Z. Wu, R. Leahy, An optimal graph theoretic approach to data clustering : Theory and its application to image segmentation, *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* 15 (11) (1993) 1101–1113.
- [4] V. Kolmogorov, R. Zabini, What energy functions can be minimized via graph cuts?, *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* 26 (2) (2004) 147–159.
- [5] Y. Cheng, Mean shift, mode seeking, and clustering, *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* 17 (8) (1995) 790–799.
- [6] D. Comaniciu, P. Meer, Mean shift analysis and applications, in : *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Vol. 2, IEEE, 1999, pp. 1197–1203.
- [7] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in : *BMVC*, 2002.
- [8] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, *Image Vision Comput.* 22 (10) (2004) 761–767.
- [9] R. Kimmel, C. Zhang, A. M. Bronstein, M. M. Bronstein, Are mser features really interesting?, *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* 33 (11) (2011) 2316–2320.
- [10] L. Guigues, Modèles multi-échelles pour la segmentation d’images, Ph.D. thesis, Université Cergy Pontoise, France. (December 2003).
- [11] Y. Xu, Tree-based shape spaces : definition and applications in image processing and computer vision, Ph.D. thesis, Université Paris-Est (2013).
- [12] B. R. Kiran, Energetic-lattice based optimization, Ph.D. thesis, Paris Est (2014).

- [13] F. Meyer, Minimum spanning forests for morphological segmentation, in : Workshop on Mathematical Morphology and its Applications to Image Processing, Fontainebleau, 1994, pp. 77,84.
- [14] J. Nešetřil, E. Milková, H. Nešetřilová, Otakar boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history, *Discrete Mathematics* 233 (1) (2001) 3–36.
- [15] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, *Proceedings of the American Mathematical society* 7 (1) (1956) 48–50.
- [16] Y. Xu, E. C. Uberbacher, 2d image segmentation using minimum spanning trees, *Image and Vision Computing* 15 (1) (1997) 47–57.
- [17] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient graph-based image segmentation, *International Journal of Computer Vision* 59 (2) (2004) 167–181.
- [18] J. Cousty, G. Bertrand, L. Najman, M. Couprie, Watershed cuts : Minimum spanning forests and the drop of water principle, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (8) (2009) 1362–1374.
- [19] C. Vachier, F. Meyer, Extinction value : a new measurement of persistence, in : IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece, 1995, pp. 254–257.
- [20] R. C. Prim, Shortest connection networks and some generalizations, *Bell system technical journal* 36 (6) (1957) 1389–1401.
- [21] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische mathematik* 1 (1) (1959) 269–271.
- [22] F. Zanoguera, B. Marcotegui, F. Meyer, A toolbox for interactive segmentation based on nested partitions., in : *ICIP* (1), 1999, pp. 21–25.
- [23] B. Marcotegui, F. Zanoguera, P. Correia, R. Rosa, F. Marqués, R. Mech, M. Wollborn, A video object generation tool allowing friendly user interaction, in : *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, Vol. 2, IEEE, 1999, pp. 391–395.
- [24] J. Stawiaski, *Mathematical morphology and graphs : Application to interactive medical image segmentation.*, Ph.D. thesis, Ecole des Mines de Paris (2008).
- [25] F. Zanoguera, B. Marcotegui, F. Meyer, A segmentation pyramid for the interactive segmentation of 3-d images and video sequences, in : *Mathematical Morphology and its Applications to Image and Signal Processing*, Springer, 2000, pp. 223–232.
- [26] F. Zanoguera, *Segmentation interactive d’images fixes et de séquences vidéo basée sur des hiérarchies de partitions.*, Ph.D. thesis, Ecole des Mines de Paris (2001).
- [27] F. Z. B. Marcotegui, Image editing tools based on multi-scale segmentation, in : *International Symposium of Mathematical Morphology 6th ISMM*, 2002, pp. 127–135.

- [28] F. Marqués, B. Marcotegui, F. Zanoguera, P. Correia, R. Mech, M. Wollborn, Partition-based image representation as basis for user-assisted segmentation, in : Image Processing, 2000. Proceedings. 2000 International Conference on, Vol. 1, IEEE, 2000, pp. 312–315.
- [29] S. Beucher, Watershed, hierarchical segmentation and waterfall algorithm, in : J. Serra, P. Soille (Eds.), Mathematical morphology and its applications to image processing, Kluwer Academic Publishers, 1994, pp. 69–76.
- [30] B. Marcotegui, S. Beucher, Fast implementation of waterfall based on graphs, in : Mathematical Morphology : 40 Years On, Springer, 2005, pp. 177–186.
- [31] V. Marion, O. Lecoindre, C. Lewandowski, J. G. Morillon, R. Aufrere, B. Marcotegui, R. Chapuis, S. Beucher, Robust perception algorithms for road and track autonomous following, in : Proc. SPIE, Vol. 5422, 2004, pp. 55–66. doi :10.1117/12.542998.  
URL <http://dx.doi.org/10.1117/12.542998>
- [32] F. Meyer, Contrast feature extraction, Special Issues of Practical Metallography 8 (1978) 374–380.
- [33] S. Beucher, Numerical residues, in : C. Ronse, L. Najman, E. Decencière (Eds.), Mathematical Morphology : 40 Years On, Vol. 30 of Computational Imaging and Vision, Springer-Verlag, Dordrecht, 2005, pp. 23–32.
- [34] S. Beucher, Numerical residues, Image and Vision Computing 25 (4) (2007) 405–415.
- [35] R. Vogt, A spatially variant, locally adaptive, background normalization operator, in : J. Serra, P. Soille (Eds.), Mathematical morphology and its applications to image processing, Kluwer Academic Publishers, 1994, pp. 45–52.
- [36] M. Pesaresi, J. Benediktsson, Image segmentation based on the derivative of the morphological profile, in : J. Goutsias, L. Vincent, D. S. Bloomberg (Eds.), Mathematical Morphology and its Applications to Image and Signal Processing, Vol. 18, Springer US, 2000, pp. 179–188.
- [37] M. Pesaresi, J. Benediktsson, A new approach for the morphological segmentation of high-resolution satellite imagery, Geoscience and Remote Sensing, IEEE Transactions on 39 (2) (2001) 309–320.
- [38] J. Fabrizio, B. Marcotegui, Fast implementation of the ultimate opening, in : Proc. of the 9th Intern. Symposium on Mathematical Morphology and Its Application to Signal and Image Processing, ISMM '09, Springer-Verlag, 2009, pp. 272–281.
- [39] P. Salembier, A. Oliveras, J. L. Garrido, Anti-extensive connected operators for image and sequence processing, IEEE Trans. on Image Processing 7 (1998) 555–570.
- [40] J. Serra, A lattice approach to image segmentation, J. Math. Imaging Vis. 24 (2006) 83–130.

- [41] C. Ronse, Partial partitions, partial connections and connective segmentation, *J. Math. Imaging Vis.* 32 (2008) 97–125.
- [42] T. Retornaz, B. Marcotegui, Scene text localization based on the ultimate opening, in : *Workshop ImagEval*, Amsterdam, The Netherlands, 2007.
- [43] S. Outal, S. Beucher, Controlling the ultimate openings residues for a robust delineation of fragmented rocks, in : *The 10th European Congress of Stereology and Image Analysis*, 2009.
- [44] T. Retornaz, B. Marcotegui, Scene text localization based on the ultimate opening, in : *International Symposium of Mathematical Morphology 8th ISMM*, 2007, pp. 177–188.
- [45] J. Hernández, B. Marcotegui, Ultimate attribute opening segmentation with shape information, in : *Proc. of the 9th Intern. Symposium on Mathematical Morphology and Its Application to Signal and Image Processing, ISMM '09*, Springer-Verlag, 2009, pp. 205–214.
- [46] X. Zhang, G. Thibault, E. Decenci re, B. Marcotegui, B. Lay, R. Danno, G. Cazuguel, G. Quellec, M. Lamard, P. Massin, A. Chabouis, Z. Victor, A. Erginay, Exudate detection in color retinal images for mass screening of diabetic retinopathy, *Medical Image Analysis* 18 (7) (2014) 1026 – 1043. doi :<http://dx.doi.org/10.1016/j.media.2014.05.004>.  
URL <http://www.sciencedirect.com/science/article/pii/S1361841514000693>
- [47] A. Serna, B. Marcotegui, Attribute controlled reconstruction and adaptive mathematical morphology, in : C. Hendriks, G. Borgefors, R. Strand (Eds.), *Mathematical Morphology and Its Applications to Signal and Image Processing*, Vol. 7883 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2013, pp. 207–218.  
URL [http://dx.doi.org/10.1007/978-3-642-38294-9\\_18](http://dx.doi.org/10.1007/978-3-642-38294-9_18)
- [48] A. Serna, B. Marcotegui, E. Decenci re, T. Baldeweck, A.-M. Pena, S. Brizion, Segmentation of elongated objects using attribute profiles and area stability : Application to melanocyte segmentation in engineered skin, *Pattern Recognition Letters* 47 (0) (2014) 172 – 182, *advances in Mathematical Morphology*. doi :<http://dx.doi.org/10.1016/j.patrec.2014.03.014>.  
URL <http://www.sciencedirect.com/science/article/pii/S0167865514000944>
- [49] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in : *Proc. 8th Int'l Conf. Computer Vision*, Vol. 2, 2001, pp. 416–423.
- [50] S. Alpert, M. Galun, R. Basri, A. Brandt", "image segmentation by probabilistic bottom-up aggregation and cue integration.", in : "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", "2007", pp. 1–8.

- [51] D. Comaniciu, P. Meer, Mean shift : A robust approach toward feature space analysis, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (5) (2002) 603–619.
- [52] D. Comaniciu, P. Meer, S. Member, Mean shift : A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002) 603–619.
- [53] F. Meyer, From connected operators to levelings, in : H. Heijmans, J. Roerdink (Eds.), *Mathematical Morphology and its Applications to Image and Signal Processing*, Vol. 12 of *Computational Imaging and Vision*, Kluwer Academic Publishers, Dordrecht, 1998, pp. 191–198.
- [54] F. Meyer, Levelings, image simplification filters for segmentation, *Journal of Mathematical Imaging and Vision* 20 (1-2) (2004) 59–72.
- [55] S. Loncaric, A survey of shape analysis techniques, *Pattern Recognition* 31 (1998) 983–1001.
- [56] R. C. Veltkamp, M. Hagedoorn, State-of-the-art in shape matching, Tech. rep., *Principles of Visual Information Retrieval* (1999).
- [57] G. Charpiat, O. Faugeras, R. Keriven, Approximations of shape metrics and application to shape warping and empirical shape statistics, *Foundations of Computational Mathematics* 5 (1) (2005) 1–58.
- [58] R. C. Veltkamp, M. Hagedoorn, Shape similarity measures, properties, and constructions, in : *In Advances in Visual Information Systems, 4th Intern. Conf., VISUAL 2000*, Springer, 2000, pp. 467–476.
- [59] J. Benner, A. Geiger, K. Leinemann, Flexible generation of semantic 3d buildings models, in : *First Intern. Workshop on Next Generation 3D City Models, 2005*, pp. 17–22.
- [60] P. Müller, G. Zeng, P. Wonka, L. V. Gool, Image-based procedural modeling of facades, *ACM Trans. on Graphics* 26 (3) (2007) 85–93.
- [61] H. Mayer, S. Reznik, Building façade interpretation from image sequences, in : *CMRT05, Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms, and Evaluation*, Vol. XXXVI, 2005, pp. 55 – 60.
- [62] F. Korč, W. Förstner, eTRIMS Image Database for interpreting images of man-made scenes, Tech. Rep. TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn (April 2009).  
URL [http://www.ipb.uni-bonn.de/projects/etrimis\\_db/](http://www.ipb.uni-bonn.de/projects/etrimis_db/)
- [63] T. Retornaz, Détection de textes enfouis dans des bases d’images généralistes. un descripteur sémantique pour l’indexation., Ph.D. thesis, Ecole des Mines de Paris (2007).

- [64] Z. Yin, X. Zhou, C. Bakal, F. Li, Y. Sun, N. Perrimon, S. T. C. Wong, Using iterative cluster merging with improved gap statistics to perform online phenotype discovery in the context of high-throughput rna screens., *BMC Bioinformatics* 9 (2008) 264.
- [65] B. Marcotegui, J. Hernández, T. Retornaz, Ultimate opening and gradual transitions, in : *Mathematical Morphology and Its Applications to Image and Signal Processing*, Springer, 2011, pp. 166–177.
- [66] T. Retornaz, B. Marcotegui, Ultimate opening implementation based on a flooding process, in : *The 12th Intern. Congress for Stereology, ICS XII*,, 2007.
- [67] J. Fabrizio, M. Cord, B. Marcotegui, Text extraction from street level images, in : *ISPRS Workshop, CMRT City Models Roads and Traffic*, 2009, pp. 199–204.
- [68] J. Fabrizio, B. Marcotegui, M. Cord, Text segmentation in natural scenes using toggle-mapping, in : *IEEE International Conference on Image Processing, ICIP'09*, 2009, pp. 2373–2376.
- [69] R. Minetto, N. Thome, M. Cord, J. Fabrizio, B. Marcotegui, Snoopertext : A multiresolution system for text detection in complex visual scenes, in : *Image Processing (ICIP)*, 2010 17th IEEE International Conference on, IEEE, 2010, pp. 3861–3864.
- [70] J. Fabrizio, B. Marcotegui, M. Cord, Text detection in street level images, *Pattern Analysis and Applications* 16 (4) (2013) 519–533. doi :10.1007/s10044-013-0329-7.  
URL <http://dx.doi.org/10.1007/s10044-013-0329-7>
- [71] E. Decenci ere, G. Cazuguel, X. Zhang, G. Thibault, J.-C. Klein, F. Meyer, B. Marcotegui, G. Quellec, M. Lamard, R. Danno, et al., Teleophta : Machine learning and image processing methods for teleophthalmology, *IRBM* 34 (2) (2013) 196–203.
- [72] E. J. Breen, R. Jones, Attribute openings, thinnings, and granulometries, *Computer Vision and Image Understanding* 64 (3) (1996) 377–389.
- [73] R. Jones, Connected Filtering and Segmentation Using Component Trees, *Computer Vision and Image Understanding* 75 (3) (1999) 215–228.
- [74] C. Lantu ejoul, S. Beucher, On the use of the geodesic metric in image analysis, *Journal of Microscopy* 121 (1) (1981) 39–49. doi :10.1111/j.1365-2818.1981.tb01197.x.
- [75] V. Morard, E. Decenci ere, P. Dokl adal, Efficient geodesic attribute thinnings based on the barycentric diameter, *Journal of Mathematical Imaging and Vision* 46 (1) (2013) 128–142. doi :10.1007/s10851-012-0374-7.
- [76] S. C. Lee, R. Nevatia, Extraction and integration of window in a 3d building model from ground view images, *IEEE Computer Vision and Pattern Recognition* 02 (2004) 113–120. doi :<http://doi.ieeecomputersociety.org/10.1109/CVPR.2004.99>.



- [77] X. Wang, S. Totaro, F. Taillandier, A. Hanson, S. Teller, Recovering façade texture and microstructure from real-world images, in : ECCV Texture 2002 Workshop, Copenhagen, Denmark, 2002.
- [78] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, ISBN : 0521540518, 2004.
- [79] A. Serna, J. Hernández, B. Marcotegui, Adaptive parameter tuning for morphological segmentation of building facade images, in : Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, IEEE, 2012, pp. 2268–2272.
- [80] O. Teboul, L. Simon, P. Koutsourakis, N. Paragios, Segmentation of building facades using procedural shape priors, in : CVPR, 2010, pp. 3105–3112.  
URL <http://www.mas.ecp.fr/vision/Personnel/teboul/data.html>
- [81] J. Hernández, B. Marcotegui, Ultimate opening segmentation with shape constraints, in : Proceedings of the Eighth IASTED International Conference, Vol. 630, 2008, p. 333.
- [82] J. Hernández, B. Marcotegui, Point cloud segmentation towards urban ground modeling, in : 5th GRSS/ISPRS Joint workshop on remote sensing and data fusion over urban areas, Shanghai, China, 2009.
- [83] J. Hernández, B. Marcotegui, Morphological segmentation of building façade images, in : IEEE International Conference on Image Processing, ICIP'09, 2009, pp. 4029–4032.
- [84] J. Hernández, B. Marcotegui, Shape ultimate attribute opening, Image and Vision Computing 29 (8) (2011) 533–545.
- [85] D. J. R. Meagher, Octree encoding : a new technique for the representation, manipulation and display of arbitrary 3-D objects by computer, Tech. rep., Electrical and Systems Engineering Department Rensselaer Polytechnic Institute Image Processing Laboratory (1980).
- [86] J. L. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM 18 (9) (1975) 509–517. doi :10.1145/361002.361007.
- [87] J. Demantké, C. Mallet, N. David, B. Vallet, Dimensionality based scale selection in 3d lidar point clouds, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5) (2011) W12.
- [88] S. Calderon, T. Boubekeur, Point morphology, ACM Transactions on Graphics (TOG) 33 (4) (2014) 45.
- [89] A. Hoover, G. Jean-baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, R. B. Fisher, An experimental comparison of range image segmentation algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (7) (1996) 673–689.

- [90] B. Gorte, Planar feature extraction in terrestrial laser scans using gradient based range image segmentation, in : The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVI-3/W52, 2007, pp. 173–177.
- [91] P. J. Soille, M. M. Ansoult, Automated basin delineation from digital elevation models using mathematical morphology, *Signal Processing* 20 (2) (1990) 171 – 182. doi :[http://dx.doi.org/10.1016/0165-1684\(90\)90127-K](http://dx.doi.org/10.1016/0165-1684(90)90127-K).  
URL <http://www.sciencedirect.com/science/article/pii/016516849090127K>
- [92] J. Hernández, Analyse morphologique d’images pour la modélisation d’environnements urbains, Ph.D. thesis, Mines ParisTech - C.M.M., Fontainebleau - France (december 2009).
- [93] J. Hernández, B. Marcotegui, Segmentation et interprétation de nuages de points pour la modélisation d’environnements urbains, *Revue Francaise de Photogrammetrie et de Teledetection* (191) (2008) 28.
- [94] C. Dold, C. Brenner, Registration of terrestrial laser scanning data using planar patches and image data, in : Intern. Archives of Photogrammetry and Remote Sensing, 2006, pp. 25–27.
- [95] I. Stamos, G. Yu, G. Wolberg, S. Zokai, 3d modeling using planar segments and mesh elements, in : Intern. Symposium on 3D Data Processing, Visualization and Transmission, IEEE Computer Society, Washington, DC, USA, 2006, pp. 599–606. doi :<http://dx.doi.org/10.1109/3DPVT.2006.5>.
- [96] S. Becker, N. Haala, Combined feature extraction for façade reconstruction, in : ISPRS Workshop on Laser Scanning and SilviLaser, Espoo, Finland, 2007, pp. 44–50.
- [97] H. Boulaassal, T. Landes, P. Grussenmeyer, F. Tarsha-Kurdi, Automatic segmentation of building facades using terrestrial laser data, *ISPRS Workshop on Laser Scanning and SilviLaser, Espoo, Finland (2007)* 65–70.  
URL <http://www.scientificcommons.org/34412932>
- [98] G. Yu, M. Grossberg, G. Wolberg, I. Stamos, Think globally, cluster locally : A unified framework for range segmentation, in : Intern. Symposium on 3D Data Processing, Visualization and Transmission, 2008.
- [99] F. Goulette, F. Nashashibi, S. Ammoun, C. Lourceau, An Integrated on-Board Laser Range Sensing System for On-the-Way City and Road Modelling, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34 (A) (2006) 3–5.
- [100] R. Schnabel, R. Wessel, R. Wahl, R. Klein, Shape recognition in 3D point clouds, in : The 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Union Agency-Science Press, 2008, pp. 1–8.

- [101] X. Zhu, H. Zhao, Y. Liu, Y. Zhao, H. Zha, Segmentation and classification of range image from an intelligent vehicle in urban environment, in : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), 2010, pp. 1457–1462. doi :10.1109/IROS.2010.5652703.
- [102] S. Pu, M. Rutzinger, G. Vosselman, S. O. Elberink, Recognizing basic structures from mobile laser scanning data for road inventory studies, ISPRS Journal of Photogrammetry and Remote Sensing 66 (6) (2011) 28–39. doi :10.1016/j.isprsjprs.2011.08.006.
- [103] M. Poreba, F. Goulette, RANSAC algorithm and elements of graph theory for automatic plane detection in 3D point clouds, Archives of Photogrammetry, Cartography and Remote Sensing 24 (2012) 301–310.
- [104] A. K. Aijazi, P. Checchin, L. Trassoudaine, Segmentation Based Classification of 3D Urban Point Clouds : A Super-Voxel Based Approach with Evaluation, Remote Sensing 5 (4) (2013) 1624–1650.
- [105] A. Hervieu, B. Soheilian, Semi-automatic road/pavement modeling using mobile laser scanning, in : City Models, Roads and Traffic. CMRT13, 2013.
- [106] M. Schmitt, F. Prêteux, Un nouvel algorithme en morphologie mathématique : les rh maxima et rh minima, Proc. 2ieme Semaine Internationale de l’Image Electronique (1986) 469–475.
- [107] C. Mallet, F. Bretar, U. Soergel, Analysis of full-waveform lidar data for classification of urban areas, Photogrammetrie Fernerkundung GeoInformation (PFG) 5 (2008) 337–349.
- [108] C. Alexander, K. Tansey, J. Kaduk, D. Holland, N. J. Tate, Backscatter coefficient as an attribute for the classification of full-waveform airborne laser scanning data in urban areas, ISPRS Journal of Photogrammetry and Remote Sensing 65 (5) (2010) 423–432.
- [109] G. Mountrakis, J. Im, C. Ogole, Support vector machines in remote sensing : A review, ISPRS Journal of Photogrammetry and Remote Sensing 66 (3) (2011) 247–259.
- [110] M. Brédif, B. Vallet, A. Serna, B. Marcotegui, N. Paparoditis, TerraMobilita/iQmulus urban point cloud classification benchmark, in : IQmulus workshop on Processing Large Geospatial Data, iQmulus/TerraMobilita contest, 2014, pp. 1–6, <http://data.ign.fr/benchmarks/UrbanAnalysis/> (Last accessed : June 26, 2014).
- [111] UN, United Nations Convention on the Rights of Persons with disabilities, <http://www.un.org/disabilities/convention/conventionfull.shtml> (Last accessed : September 9, 2014) (2007).
- [112] G. Vosselman, L. Zhou, Detection of curbstones in airborne laser scanning data, in : The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-3/W8, 2009, pp. 11–116.

- [113] J. Siegemund, D. Pfeiffer, U. Franke, W. Förstner, Curb Reconstruction using Conditional Random Fields, in : IEEE Intelligent Vehicles Symposium (IV), IEEE Computer Society, 2010, pp. 203–210.
- [114] J. Hernández, B. Marcotegui, Filtering of artifacts and pavement segmentation from mobile lidar data, in : ISPRS Workshop Laserscanning 2009, 2009, pp. 329–333.
- [115] A. Serna, B. Marcotegui, Urban accessibility diagnosis from mobile laser scanning data, ISPRS Journal of Photogrammetry and Remote Sensing 84 (2013) 23–32.
- [116] C. Lantuejoul, F. Maisonneuve, Geodesic methods in quantitative image analysis, Pattern Recognition 17 (2) (1984) 177 – 187.  
doi :[http://dx.doi.org/10.1016/0031-3203\(84\)90057-8](http://dx.doi.org/10.1016/0031-3203(84)90057-8).  
URL <http://www.sciencedirect.com/science/article/pii/S0031320384900578>
- [117] V. Morard, E. Decencière, P. Dokládál, Geodesic attributes thinnings and thickenings, in : Proceedings of the 10th International Symposium on Mathematical Morphology (ISMM 2011), Springer-Verlag, 2011, pp. 200–211.
- [118] L. Zhou, G. Vosselman, Mapping curbstones in airborne and mobile laser scanning data, International Journal of Applied Earth Observation and Geoinformation 18 (2012) 293–304.
- [119] A. Serna, B. Marcotegui, F. Goulette, J.-E. Deschaud, Paris-rue-Madame database : a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods, in : 3rd International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014, 2014, pp. 1–4.
- [120] A. Serna, B. Marcotegui, Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning, {ISPRS} Journal of Photogrammetry and Remote Sensing 93 (0) (2014) 243 – 255.  
doi :<http://dx.doi.org/10.1016/j.isprsjprs.2014.03.015>.  
URL <http://www.sciencedirect.com/science/article/pii/S0924271614000872>
- [121] D. Craciun, A. Serna-Morales, J.-E. Deschaud, B. Marcotegui, F. Goulette, Scalable and detail-preserving ground surface reconstruction from large 3d point clouds acquired by mobile mapping systems, in : PCV (Photogrammetric Computer Vision), Vol. 3, 2014, pp. 73–80.
- [122] M. Brédif, B. Vallet, A. Serna, B. Marcotegui, N. Paparoditis, Terramobilita/iq-mulus urban point cloud classification benchmark, in : Workshop on Processing Large Geospatial Data, 2014.
- [123] A. K. Aijazi, A. Serna, B. Marcotegui, P. Checchin, L. Trassoudaine, Segmentation and Classification of 3D Urban Point Clouds : Comparison and Combination of Two Approaches., in : 10th Field and Service Robotics (FSR), Toronto, Canada, 2015.

- [124] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, N. Paparoditis, Terramobilita/i-qmulus urban point cloud analysis benchmark, *Computers & Graphics* 49 (0) (2015) 126 – 133. doi :<http://dx.doi.org/10.1016/j.cag.2015.03.004>.  
URL <http://www.sciencedirect.com/science/article/pii/S009784931500028X>
- [125] E. Carlinet, T. Géraud, A comparison of many max-tree computation algorithms, in : *Mathematical Morphology and Its Applications to Signal and Image Processing*, Springer, 2013, pp. 73–85.
- [126] E. Carlinet, T. Géraud., A color tree of shapes with illustrations on filtering, simplification, and segmentation, in : *Proc. of the 12th Intern. Symposium on Mathematical Morphology and its Applications to Image and Signal Processing, Computational Imaging and Vision*, Springer-Verlag, 2015, pp. 363–374.

# A Toolbox for Interactive Segmentation Based on Nested Partitions

F. Zanoguera, B. Marcotegui and F. Meyer  
*Centre de Morphologie Mathématique*  
*Ecole Nationale Supérieure des Mines de Paris*  
*77305 Fontainebleau Cedex, France*  
{zanoguera,marcotegui,meyer}@cmm.ensmp.fr

## Abstract

*This paper presents a toolbox for interactive image segmentation based on a series of nested partitions of increasing coarseness. The user can navigate among the different resolution levels and select regions with simple mouse clicks. The whole family of partitions can be created during a single morphological flooding of the image, leading to a very fast algorithm. The information is stored in a minimum spanning tree.*

## 1. Introduction

Automatic segmentation has proved to be a difficult issue in applications where generic types of images must be treated. In multimedia applications, where no *a priori* knowledge of the image exists, interactive segmentation presents an attractive solution. The goal here is to minimise the effort required from the user to obtain the desired output.

Most existing interactive segmentation tools suffer from limitations. Some lack flexibility in the interaction process [6] [7] [2] while others rely on expert knowledge from the user side [7] or make assumptions on the characteristics of the images to be segmented [3].

A system for interactive segmentation of generic images is presented. No expert knowledge from the user is required, and the interaction mechanisms are flexible and appear as natural.

We show in the first place how a multiscale segmentation can be defined as the set of balls of increasing radius for an ultrametric distance. We describe afterwards how this multiscale segmentation can be created during a single morphological flooding and stored in the form of a minimum spanning tree. Finally, we show how the interaction with the user is carried out.

## 2. Multiscale segmentation

Our system is based on the generation of a set of nested partitions  $P_1, P_2, \dots, P_k$  of the image. These partitions are

organised as a multiscale segmentation, where the lowest level  $P_1$  gives the finest resolution with  $k$  regions. The highest level  $P_k$  corresponds to a partition with the whole image as only one region. A partition at a higher level is obtained by merging regions of the partition below. For this reason each contour present in a given partition is also present in all partitions of lower level.

### 2.1. Neighbourhood graph and minimum spanning tree

**2.1.1. The watershed transform.** The watershed transform [1] constitutes the paradigm of morphological segmentation. The morphological (colour) gradient of the image to segment is first constructed. This image, considered as a topographic surface, is steadily flooded. Starting from the minima, lakes are formed which eventually meet with other lakes. When this happens, they are prevented from merging by erecting a zero-thickness dam between them. The water level increases until the whole image has been flooded. At the end of this process, a partition of the image is obtained, which contains one region for each catchment basin of the gradient image. In the following, the watershed transform will be referred to as uniform flooding, as the water level is kept constant during the flooding.

The process described produces a very strong over-segmentation of the image. The solution to this problem has traditionally been the use of markers. In the marker approach, the topographic surface is not flooded from sources placed at all the image minima, but from selected sources, called markers, placed at different points of the image. The segmentation displaces the over-segmentation problem into finding good markers.

**2.1.2. The neighbourhood graph.** The gradient image (interpreted as a topographic surface) can be represented by a neighbourhood graph, which contains a node for each catchment basin of the gradient image. Two nodes are linked by an edge if their corresponding catchment basins are neighbours. The edges are weighted with a dissimilarity measure between regions.

The flooding from markers may be carried out on the neighbourhood graph. Sources are placed on some nodes, and the edge weights represent the height of the borders that the water must cross. If the dissimilarity measure chosen to weigh the graph edges is the lowest pass point along the border separating the two catchment basins on the gradient image, then flooding the neighbourhood graph is equivalent to flooding the image. However, the graph representation is more general and can be used in any context where a fine partition and a dissimilarity measure between neighbouring regions are defined.

Flooding the neighbourhood graph is much faster than flooding the topographic surface, as the number of nodes on the graph is much smaller than the number of pixels on the image.

**2.1.3. The minimum spanning tree.** A close observation of the flooding process reveals that the water always follows the path of lowest sup-section (sup-section: highest value along the path). This is as well the path chosen by the minimum spanning tree of the neighbourhood graph [5]. Therefore, the minimum spanning tree of the neighbourhood graph contains all the necessary information for flooding purposes.

On the minimum spanning tree, every pair of nodes is linked by a unique path, which corresponds to the one of lowest sup-section among all the existing paths between the two nodes on the neighbourhood graph. Because on the minimum spanning tree every pair of nodes is linked by a unique path, eliminating  $n - 1$  edges of the tree yields  $n$  connected components, and therefore a segmentation of the image into  $n$  regions. For different values of  $n$  we obtain partitions containing a different number of regions, and therefore a multiscale segmentation.

The minimum spanning tree can be created from the neighbourhood graph by using any of the algorithms found in the literature. However, a more direct algorithm may be derived for our application by taking advantage of the fact that among all possible paths, the minimum spanning tree always chooses the one with lowest sup-section. This is also the path chosen by the water during the flooding. The minimum spanning tree can therefore be created simultaneously to the flooding process.

Initially, a node of the minimum spanning tree is created for each minimum of the gradient image. The gradient image is then flooded as described. Every time that two lakes meet for the first time, an edge is added between their corresponding nodes of the tree, weighted with the current water level. At the end of the process, the minimum spanning tree has been created.

## 2.2. Flooding ultrametrics

An ultrametric distance is a metric that satisfies the following conditions:

1.  $d(x, x) = 0 \quad \forall x$
2.  $d(x, y) = d(y, x) \quad \forall x, y$
3.  $d(x, y) \leq \text{Max}(d(x, z), d(z, y)) \quad \forall x, y, z$

The water level for which each pair of catchment basins meet for the first time defines an ultrametric distance.

The first two conditions are obviously satisfied by this distance. The third condition is also satisfied, as the right hand part of the inequality represents the water level for which the catchment basins  $x$ ,  $y$  and  $z$  meet for the first time while the left hand part represents the water level for which only two of these lakes meet for the first time.

This ultrametric distance may alternatively be defined on the minimum spanning tree between any pair of nodes as the sup-section of the unique path linking the two nodes.

The ultrametric distance has a very interesting property:

**Proposition 1** *Two open (resp. closed) balls  $B(x, r)$  and  $B(y, r)$  of same radius and different centre are either disjoint or identical.*

**Interpretation:** A ball of centre  $C$  and radius  $R$  is the set of points that are at a distance smaller or equal than  $R$  from the centre  $C$ . On the minimum spanning tree, the distance between two nodes has been defined as the sup-section of the path linking the two nodes. Cutting the edges of the tree with value higher than  $R$  and taking the connected component that contains  $C$  yields the ball of centre  $C$  and radius  $R$ . If another node of the same subtree is taken as centre, the same subtree is obtained. If an element of another subtree is taken as centre, the subtree obtained is disjoint with the previous one.

The set of maximal balls with radius smaller than a certain value  $R$  produce a multiscale segmentation for increasing values of  $R$  [4]. These balls can be obtained from the minimum spanning tree by cutting the edges with value higher than  $R$ .

However, the partitions obtained using the uniform flooding ultrametrics are not perceptually satisfactory. Other ultrametrics must be defined in order to obtain meaningful results.

## 2.3. New ultrametrics defined through synchronous flooding

In the synchronous flooding [4], the topographic surface is not flooded at a constant water level, but in such a way that all the lakes keep constant a certain measure (depth,

surface, volume) at any given time. When a catchment basin is about to overflow into a neighbouring catchment basin, the first is already full and stops growing until its neighbour has reached the same water level. At this moment, the full catchment basin is assigned to the same class as its neighbour, i.e. it is absorbed by it. As the flooding continues, successive absorptions will take place, until the whole image has been flooded and all the lakes belong to the same class.

New ultrametries can be defined through synchronous flooding, by associating to each pair of catchment basins the measure (the same one chosen for flooding the lakes: depth, surface or volume) of the lake that was full when they were joined into the same class for the first time.

The edge weights of the minimum spanning tree may be modified so that the distances calculated from it correspond to the new ultrametric distances. This can again be done simultaneously to the synchronous flooding. Initially, a node of the tree is created for each catchment basin of the gradient image. Every time that a lake is about to overflow into another lake, an edge is added between their corresponding nodes on the tree, weighted with the measure of the full lake. At the end of the process, the tree has been created, its edge weights representing the chosen ultrametries.

The ultrametries chosen will in general have a strong repercussion on the quality of the results obtained. Figure 1 shows the original image and the 15 largest balls for three different ultrametries. The depth ultrametries (Figure 1-b) considers as largest balls the most contrasted regions, regardless of their size. The surface (Figure 1-c) rates large regions as important, regardless of their contrast. Finally, the volume (Figure 1-d) balances well size and contrast of the regions and is the distance we have chosen to create our multiscale segmentation.

The synchronous flooding is however difficult to program. An equivalent algorithm is proposed in the following section, based on the uniform flooding.

## 2.4. Proposed algorithm

The following algorithm is based on the fact that for both synchronous and uniform flooding processes, the relevant events happen at the same meeting points and for the same measures of the smaller lakes. For the uniform flooding, the relevant event is the meeting of two lakes having the same water level. When this happens both lakes are full. For the synchronous flooding, the relevant event is that only one of the lakes is full, the one with smaller measure.

The proposed algorithm is very similar to the one described in Section 2.1.3 for the creation of the minimum spanning tree during uniform flooding. Initially, every

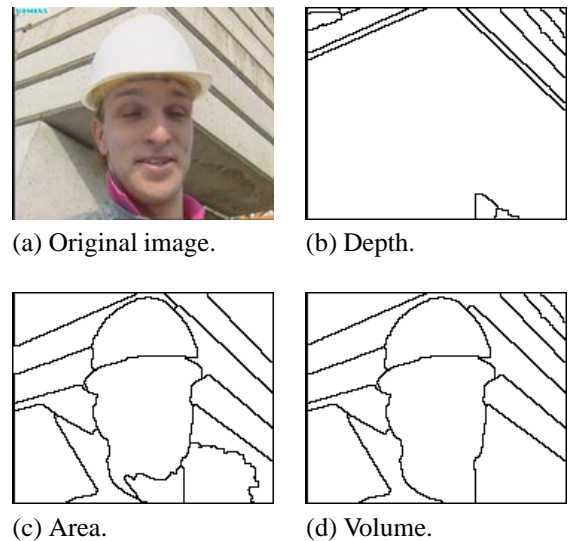


Figure 1: Comparison between the depth, area and volume ultrametries.

catchment basin belongs to a different class. Every time that two lakes that belong to different classes meet during the flooding process: (a) an edge is added between the nodes corresponding to their catchment basins weighted with the measure of the smaller lake (depth, surface, volume) and (b) both classes are joined together.

Figure 2 illustrates this algorithm. For each catchment basin there is a node on the minimum spanning tree. As the water level increases, the first pair of lakes to meet are 2 and 3. At this moment the volumes of the two lakes are compared. Lake 3 has a volume of 6 while the volume of lake 2 is only 4. Lake 3 and lake 2 are grouped together into the same class and an edge is introduced on the tree between nodes 2 and 3, of value 4 (smallest volume). The volume of the class is now updated to the sum of both, i.e. 10. The next pair of lakes to meet are 3 (which belongs to the same class than lake 2) and 4. Since lakes 3 and 4 belong to different classes, an edge will be introduced between them, weighted with the smallest class volume. Lake 3 has a class volume of 18 while the volume of lake 4 is 7. therefore, the new edge is weighted with the class volume of lake 4, i.e. 7. The last meeting is between lake 2 and lake 1. Again, class volumes are compared and the last edge is added to the tree.

## 3. Interaction with the user

The “good” segmentation does not exist in general, as it depends on both the image and the application. Often se-



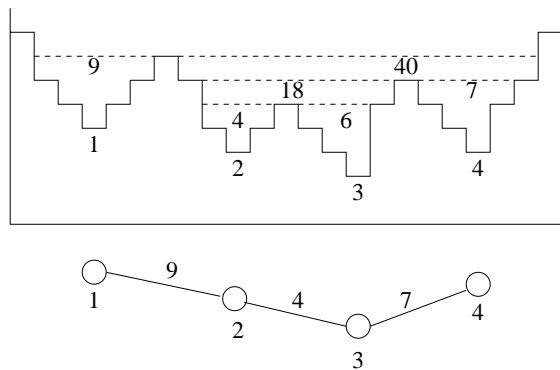


Figure 2: Flooding process with creation of a minimum spanning tree.

mantics plays an important role and this information must be introduced in the segmentation process. The injection of information by the user must be as easy and intuitive as possible. The nested segmentations produced so far provide a solid and natural support for human interaction, as will be shown in the next section.

### 3.1. Selection of the total number of regions

The most general and simple way of user interaction consists of the selection of the total number of regions in which the image must be segmented. The user indicates a total number  $n$  of regions. This corresponds to a request for an automatic segmentation of the image into  $n$  regions. The  $n$  largest balls of the ultrametrics must be presented to the user. To do this, the  $n - 1$  edges of the tree with highest weight are suppressed. This type of interaction is a starting point in the segmentation process that provides the user with a first proposal to work on. It can be presented in the form of a sliding bar that can be moved up and down to adapt the size of the regions to the specific application. This can be done in real user time because all the calculations are carried out on the minimum spanning tree rather than on the image.

Figure 3 shows three different levels of the multiscale segmentation.

### 3.2. Local actions on the image

The interaction type described in the previous section treats the image as a whole, finding the  $n$  largest balls from the point of view of the volume ultrametrics. However, the user may be interested in having some regions/objects segmented with more detail than others. In this case, the user must be offered the possibility to refine a certain area or to coarsen it by merging it with neighbouring regions.

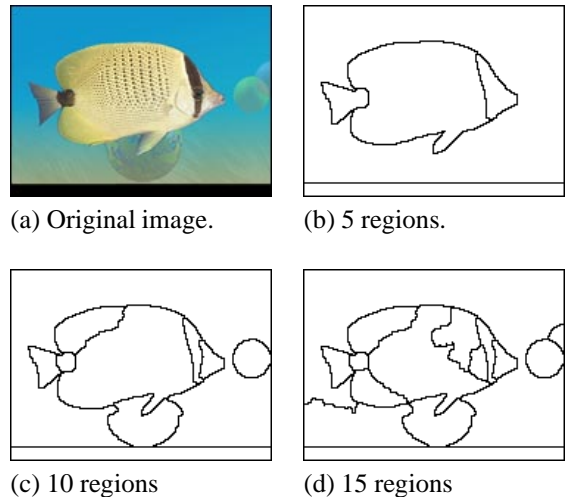


Figure 3: Original image and three different levels of the multiscale segmentation.

This is done by locally decreasing/increasing the radius of a selected ball. Two operations allow the user to locally navigate up and down the scale of segmentations.

In the *refine* operation, the user clicks on a certain area of the image with the mouse. At the same time, the number of regions in which the selected region must be subdivided may be specified. If it is not, a default value is used. The radius of the selected ball/region is decreased until it produces a re-segmentation into the requested number of regions. From the point of view of the implementation, the  $n - 1$  ( $n$  being the parameter specified by the user) edges of highest weight are searched for and eliminated, but this time only the edges inside the region selected by the user are considered.

In the *coarsen* operation, the user selects a region with a mouse click, and again a parameter  $n$  may be specified. The radius of the selected ball is increased by merging neighbouring regions. The  $n - 1$  most similar neighbouring regions are merged to the one selected. Among the previously eliminated edges, the  $n - 1$  of lowest weight that link a node belonging to the selected region with an external one are re-inserted.

Figure 4 shows an example of local operations. Starting from an automatic segmentation into 10 regions, the user has clicked on the large region inside the woman contour in Figure 4-b to request a re-segmentation of that region into 6 new regions. Figure 4-c shows the result. Next, the user has clicked on the left part of the background and requested a merging of the most similar 4 regions into one. The result is shown in Figure 4-d.

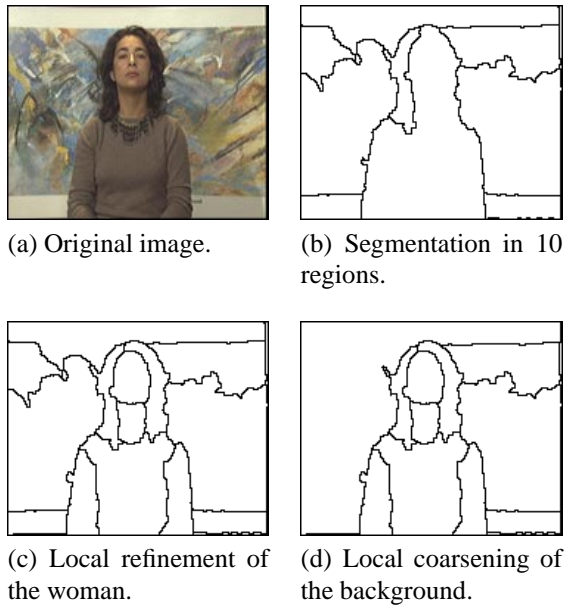


Figure 4: Example of local actions on the image.

### 3.3. Marker drawing

This type of interaction is an alternative to the selection of the total number of regions. The user draws a marker on every object of interest, and also a marker for the background. The algorithm will then again consider the edges for suppression in decreasing order, but validate a suppression only if each produced subtree contains at least one marker. Alternative algorithms can be found in [5]. Figure 5 shows the markers that have been drawn by the user to select the objects of interest and the results obtained. Further interaction of the types described in Section 3.2 is possible from this point.

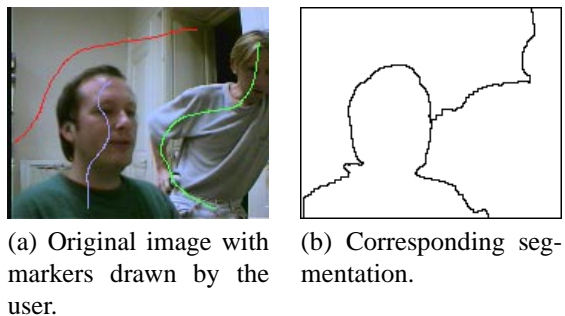


Figure 5: Example of marker drawing and the corresponding results.

## 4. Conclusion

A toolbox for the interactive segmentation of generic images has been presented. It is based on a multiscale segmentation calculated during a single flooding of the image, operation for which there exist fast implementations based on hierarchical queues. The results are stored in the form of a minimum spanning tree. Several tools have been developed, allowing an easy interactive navigation between the different levels of the scale of segmentations. Because the interaction with the user is based on simple operations on the minimum spanning tree, the computing time is fast enough for the user to perceive the results as immediate.

## References

- [1] S. Beucher and F. Meyer. "The morphological approach to segmentation: the watershed transformation", *Mathematical morphology in image processing*, E. Dougherty editor, M. Dekker, 1993.
- [2] E. Chalom and V.M. Bove. "Segmentation of an image sequence using multi-dimensional image attributes", *Proceedings of the IEEE International Conference on Image Processing, ICIP*, Lausanne, September 1996.
- [3] S. Delgado Olabarriaga, D. Koelma and A.W.M. Smeulders. "A Simple Application Framework for Interactive Segmentation Systems", *ASCI*, 1997.
- [4] F. Meyer "Morphological multiscale and interactive segmentation", *IEEE-EURASIP Workshop on Non-Linear Signal and Image Processing (NSIP'99)*, Antalya, Turkey, June 1999.
- [5] F. Meyer "Minimum spanning forests for morphological segmentation" *Mathematical Morphology and its Applications to Image Processing, ISMM'94*, J. Serra and P.Soleille Eds., Kluwer Academic Publishers, 1994.
- [6] N.E. O'Connor and S. Marlow. "Supervised semantic object segmentation and tracking via EM-based estimation of mixture density parameters", *Noblesse Workshop on Non-Linear Model Based Image Analysis*, Glasgow, July 1998.
- [7] K.R. Subramanian, D.M. Lawrence and M.T. Mostafavi. "Interactive segmentation and analysis of fetal ultrasound images", *8th EG Workshop on ViSC*, Boulogne sur Mer, April 1997.

# FAST IMPLEMENTATION OF WATERFALL BASED ON GRAPHS

B. Marcotegui and S. Beucher  
Centre de Morphologie Mathématique.  
Ecole des Mines de Paris  
{marcotegui,beucher}@cmm.ensmp.fr

**Abstract** The waterfall algorithm is a contrast-based hierarchical segmentation approach. In this paper we propose an efficient implementation based on the minimum spanning tree of the neighborhood graph. Furthermore, other hierarchies are proposed and compared to the original version of the algorithm.

**Keywords:** Hierarchical image segmentation, watershed, waterfall, minimum spanning tree, graphs.

## Introduction

Segmentation, together with filtering is often the first step of image analysis or image interpretation. The success of the whole chain of treatment relies on the accuracy of the segmentation results. Important efforts have been devoted to segmentation during the last years and it still remains a key topic of research.

The watershed transformation [3, 1] is the paradigm of segmentation of Mathematical Morphology. It has proved to be a powerful tool used in the solution of multiple applications. Its main drawback is the over-segmentation produced. Two approaches are proposed in the literature to overcome this drawback:

- the selection of markers [9], which supposes that the characteristics of the interesting objects are known;
- hierarchical approaches, that are able to rank the importance of each region.

We focus on hierarchical approaches because of their ability to segment generic images.

Several hierarchical approaches can be found in the literature. Grimaud [4] introduced the dynamics of minima that assign to each minimum a measure of its contrast. By thresholding this measure with increasing values, a hierarchy is obtained. Najman and Schmitt [10] showed that the same measure of dynamics may be assigned to a contour and introduced the geodesic saliency of watershed contours. Vachier and Meyer [11] generalized the concept of dynamics with the extinction values and proposed to assign to a minimum other measure than contrast such as area or volume. Volume extinction values result in a well adapted criterion for evaluating the visual significance of regions.

Meyer proposed a graph-based implementation of these hierarchies [7],[8]. Nodes correspond to the catchment basins of the topographic surface. If two catchment basins are neighbors, their corresponding nodes are linked by an edge. The valuation of this edge is the minimum pass point of the gradient along their common frontier. In the following we will refer to this graph as the neighborhood graph. Meyer found that all the information of a hierarchy may be stored in a very condensed structure: the minimum spanning tree (MST). This is due to the fact that the flooding always follows the path of minimum height, the same that chooses the MST of the neighborhood graph. This consideration leads to a very efficient algorithm of hierarchical segmentation [7] and has also been used for interactive segmentation [12].

In [1, 2] Beucher proposed a very interesting hierarchical segmentation approach: the waterfall. Starting from the watershed result, it consists in an iterative algorithm that at each step removes all the watershed contours completely surrounded by higher ones. Typically, less than 10 hierarchical levels are produced by iterating the waterfall algorithm. In [2] each step is implemented by a reconstruction process followed by a new flooding of the resulting image. Another implementation based on graphs is also proposed in [1, 2].

The hierarchies based on extinction values produce a different level for each merging of two regions. This is useful for interactive segmentation approaches because it offers flexibility. The waterfall generates several steps of the hierarchy with an autocalibrated number of regions. This autocalibration may be interesting for segmenting generic images without imposing a given number of regions, which can be a tricky parameter to fix.

In [5] an automatic track detection application is developed using the waterfall algorithm in the initialization step. Several waterfall iterations may be necessary until a region compatible with the track geometry is found. The existing implementation of the waterfall does not allow this application in real time.

In this paper we propose an implementation of the waterfall algorithm based on the MST. It allows to access to different levels of the hierarchy in a very efficient way. Furthermore it allows the possibility to obtain different hierarchies

based on other criteria than the frontier height, used in the original version of the waterfall algorithm.

Section 1 describes the waterfall algorithm introduced by Beucher. The proposed efficient implementation based on graphs is presented in section 2. Other hierarchies are easily introduced in the new framework, as shown in section 3. Finally section 4 concludes.

## 1. Waterfall

Let's consider a partition  $P$ . It can be the outcome of the watershed of the gradient image (as proposed in [2]) or any other partition. The frontiers are valued with a distance between regions (e.g. the minimum pass point of the gradient along the frontier, see figure 1(a)).

The waterfall algorithm removes from a partition all the frontiers completely surrounded by higher frontiers (see figure 1(b)). Thus, the importance of a frontier is measured with respect to its neighborhood. This process can be seen as a watershed applied not to the pixels of an image but to the frontiers of a partition. The iteration of the waterfall algorithm finishes with a partition of only one region.

Figure 2 illustrates the result of the waterfall process applied to a real image. Figure (a) shows the original image, (b) its gradient and (c) the watershed of the gradient. Figure (d), (e) and (f) are the different hierarchy levels produced by the iteration of the waterfall algorithm.

A first implementation of this algorithm based on graphs is proposed in [2]. The proposed graph contains a node for each arc frontier of the input partition and an edge between every pair of arcs (frontiers) delimiting the same catchment basin. (Note that this graph is different from the one we propose in the next section). The algorithm is considered as complex by the author and an image-based algorithm is proposed instead. The image-based algorithm consists in a reconstruction of the image from the watershed lines followed by the watershed transformation. Thus, a reconstruction and a new watershed computation are required to obtain a new level of the hierarchy.

In this paper, we propose an implementation of the waterfall algorithm that only requires a flooding of the image to obtain all levels of the hierarchy. The rest of the process is performed on the MST, that is much more efficient.

## 2. Waterfall based on the Minimum Spanning Tree

Given that the flooding always follows the path of minimum height, the MST obtained from the neighborhood graph, contains all the information required for the flooding process [6]. The MST is a very condensed way to store the information. Thus, it leads to very efficient implementation of hierarchical segmentation approaches.

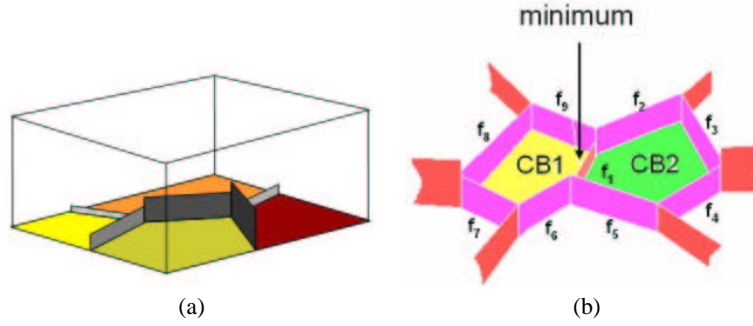


Figure 1. (a) Partition with valuated frontiers and (b) example of frontier : as the value of frontier  $f_1$  is smaller than the values of its neighboring frontiers ( $f_2$  to  $f_9$ ) it will be removed by the waterfall algorithm.

We propose an implementation of the waterfall algorithm based on the MST. The algorithm is performed in two steps:

- Minimum spanning tree (MST) generation
- Waterfall from the MST

### MST generation

We consider as input partition of the waterfall algorithm the result of the watershed. Thus, the gradient image is flooded to obtain the initial partition. The MST is obtained simultaneously to the flooding process [7, 12]. The graph is initialized with a node corresponding to each minimum of the image and without any edge. A lake is associated to each minimum. During the flooding each time that two regions of different lakes meet, an edge is added to the graph, linking both regions and the corresponding lakes are merged. Its valuation is the height of water when regions meet. At the end of the flooding process the graph has become the MST because:

- an edge is added only if regions that meet belong to different lakes (so the graph does not contain cycles),
- at the end of the flooding all the image belong to the same lake (it is an spanning tree)
- the flooding follows the path of minimum height (it is a minimum tree).

Note that the edges of the MST are valuated (not the nodes). We can define a regional minimum of the MST as a connected component of the graph, such as all the values of its edges are equal and it is surrounded by edges with strictly higher values. This definition will be used in the following subsection.

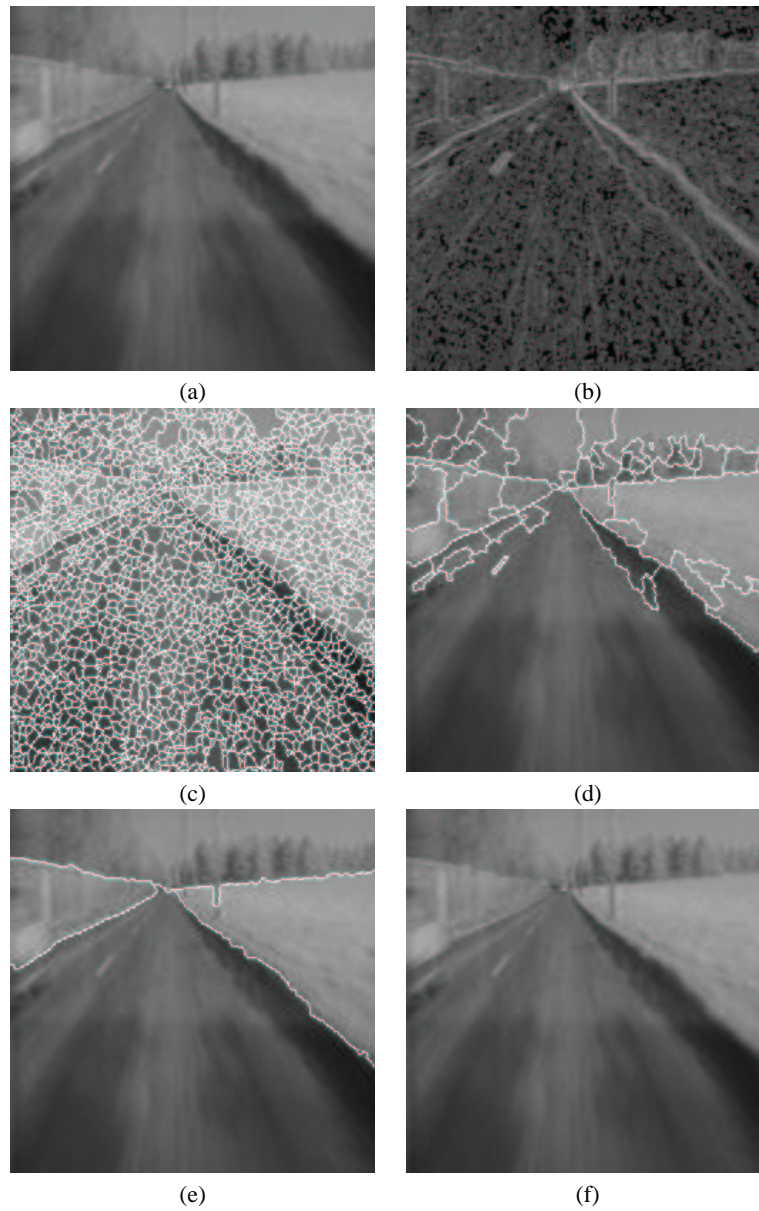


Figure 2. Waterfall iteration; First row: (a) original image and its (b) gradient. Second row: (c) Watershed segmentation and (d) first waterfall result Third row: (e),(f) Two more iterations of the waterfall algorithm

Figure 3 shows an example of partition (a) and its corresponding Minimum Spanning Tree (b).

In the following step we address the waterfall algorithm based on the MST.

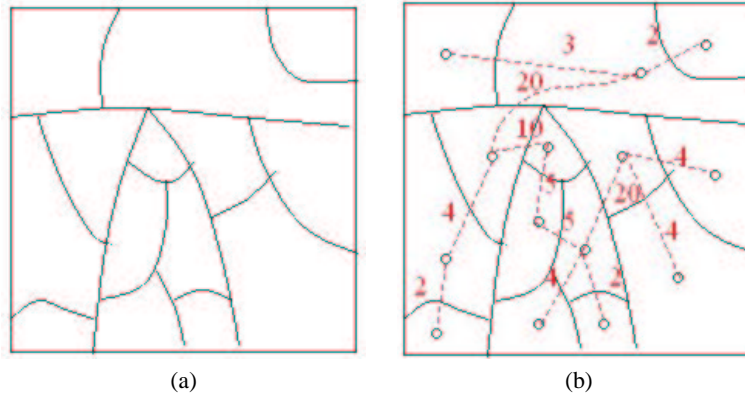


Figure 3. Example of partition (a) and its associated Minimum Spanning Tree (b).

### Waterfall from the MST

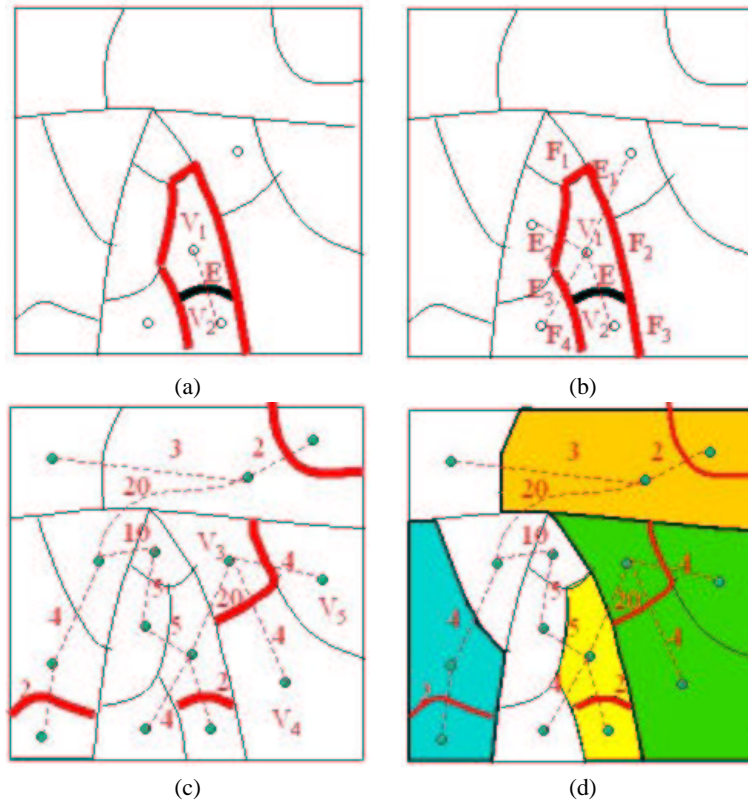
The waterfall algorithm removes from a partition the frontiers that are completely surrounded by higher frontiers, leading to a coarser partition. The waterfall can be implemented on the MST. The first step consists in identifying regional minima frontiers.

Let's consider the partition of figure 3(a) and its corresponding MST 3(b). If we take for example edge E linking regions  $V_1$  and  $V_2$  of figure 4(a) we should compare its valuation with the valuation of frontiers surrounding catchment basins  $V_1 \cup V_2$ . These frontiers are drawn in bold line in figure 4(b) (E1, E2, E3, F1, F2, F3 and F4). Edges named with an F do not belong to the MST, so by construction their valuations are higher than valuations of at least one edge named with an E. Therefore, in order to know if E is a regional minimum, it is enough to compare it with E1, E2 and E3. Thus, the MST has all the information required to identify regional minimum edges. In practice, we will compare the valuation of an edge E between  $V_1$  and  $V_2$  with edges of the MST having as one extremity  $V_1$  or  $V_2$ . More generally, we are looking for all edges that belong to a regional minimum of the MST (defined in the previous subsection).

If E is a regional minimum edge, it corresponds to a frontier that should be removed by the waterfall. This is implemented by assigning the same label to both extremities (nodes) of the minimum edge. These labels identify regions in the output partition. Thus, a different label is assigned to each minimum edge. See figure 4(d). Regions in white (those that are not neighbors of a



minimum edge) are not granted any label at this stage. Regions with a label are considered as markers. In order to obtain the final partition, markers are propagated following edges of the MST in increasing order of their valuation (fig 5). This process is a segmentation from markers on the MST [6].



*Figure 4.* Waterfall on the MST. First row : (a) Edge  $E(V_1, V_2)$  and (b) comparison of  $E$  with its neighboring edges. Second row: (c) regional minimum edges. (d) Vertex labelled from minimal edges

If edges of the MST are valued with the lowest pass point along frontiers, this algorithm is equivalent to the algorithm presented in [2].

### 3. Hierarchies with other criteria

The original waterfall algorithm removes edges according to their height. Thus it produces contrast-based hierarchies. The graph implementation of the algorithm, presented in the previous section, allows to easily produce other hierarchies, changing the edge valuation of the Minimum Spanning Tree. For

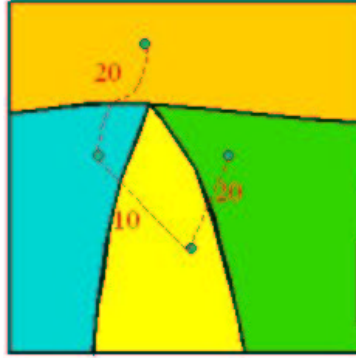


Figure 5. Waterfall result (label propagation from 4(d)).

example, we can value the edges with the volume extinction values [11], a trade-off between the contrast and the size of a region. This is equivalent to applying the waterfall algorithm to an image of saliency of watershed contours, valued with volume extinction values.

Figure 6(a) shows a level of the contrast-based hierarchy with 36 regions and 6(b) shows a partition with 24 regions obtained with the waterfall based on volume. We can see that the volume produces a more significant partition with less regions. For example, we can see that the waterfall does not get the hat, because it has a low contrast. However, the volume preserves it because even if the contrast is low it is big enough to be seen. The volume combines size and contrast trying to obtain good perceptual results. This combination is not optimal yet, because it segments regions in the background that are big but not visible at all.

The volume criterion usually over-segments big homogeneous regions. A last step may reduce this problem, just removing frontiers with contrast under a given (and small) threshold. Doing that at the end of the process is much more reliable because frontiers are longer and their contrast may be better estimated, reducing the effect of noise in a small frontier. Figure shows 7 the result of removing frontiers with contrast under 5 gray levels. The partition preserves the important regions (18) removing the over-segmentation of the background.

#### 4. Conclusion

In this paper we propose an efficient implementation of the waterfall algorithm. It consists in obtaining a Minimum Spanning Tree simultaneously to the flooding of the image. Then, all the process is performed on the graph. All the information required to manipulate the hierarchy is stored in a very condensed structure, leading to very efficient algorithms. Real-time applications such as



Figure 6. Comparison between different hierarchies. (a) Contrast-based Waterfall, 36 regions. (b) Volume based Waterfall, 24 regions.



Figure 7. Elimination of low contrasted frontiers. (a) volume based waterfall (24 regions). (b) From (a) low contrasted frontiers (contrast lower than 5) are removed (18 regions remain)

the one described in [5] may be addressed thanks to this new implementation. This implementation is based on the same data structure as the algorithms of volume extinction values, the MST. Thus, it opens the door to a combination of both approaches. For example, we have presented a waterfall based on volume extinction values and compared the results with the original version of the algorithm.

## References

- [1] S. Beucher. *Segmentation d'Images et Morphologie Mathématique*. PhD thesis, E.N.S. des Mines de Paris, 1990.
- [2] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical Morphology and its Applications to Image Processing, Proc. ISMM'94*, pages 69–76, Fontainebleau, France, 1994. Kluwer Ac. Publ.
- [3] S. Beucher and C. Lantuejoul. Use of watersheds in contour detection. In *Proc. Int. Workshop Image Processing, Real-Time Edge and Motion Detection/Estimation*, 1979.
- [4] M. Grimaud. New measure of contrast : dynamics. *Image Algebra and Morphological Processing III, San Diego CA, Proc. SPIE*, 1992.
- [5] V. Marion, O. Lecoine, C. Lewandowski, J.G. Morillon, R. Aufrère, B. Marcotegui, R. Chapuis, and S. Beucher. Robust perception algorithm for road and track autonomous following. In *SPIE Defense & Security*, Orlando, USA, April 2004.
- [6] F. Meyer. Minimum spanning forests for morphological segmentation. In *Mathematical Morphology and its Applications to Image Processing, Proc. ISMM'94*, pages 77–84, Fontainebleau, France, 1994. Kluwer Ac. Publ.
- [7] F. Meyer. Graph based morphological segmentation. In *IAPR-TC-15 Workshop on Graph-Based Representation*, pages 51–61, Vienna, Austria, May 1999.
- [8] F. Meyer. An overview of morphological segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(7):1089–1118, 2001.
- [9] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, September 1990.
- [10] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12), 1996.
- [11] C. Vachier and F. Meyer. Extinction values: A new measurement of persistence. *IEEE Workshop on Non Linear Signal/Image Processing*, pages 254–257, June 1995.
- [12] F. Zanoguera, B. Marcotegui, and F. Meyer. A tool-box for interactive image segmentation based on nested partions. In *IEEE International Conference on Image Processing*, Kobe, Japan, October 1999.



## Morphological segmentation on learned boundaries

Allan Hanbury<sup>a,\*</sup>, Beatriz Marcotegui<sup>b</sup>

<sup>a</sup> Pattern Recognition and Image Processing Group (PRIP), Institute of Computer-Aided Automation, Favoritenstraße 9/1832, A-1040 Vienna, Austria

<sup>b</sup> Centre de Morphologie Mathématique, Ecole des Mines de Paris, 35 rue Saint-Honoré, 77305 Fontainebleau cedex, France

### ARTICLE INFO

#### Article history:

Received 13 November 2006

Received in revised form 16 June 2008

Accepted 30 June 2008

#### Keywords:

Image segmentation  
Watershed  
Waterfall  
Normalised cuts  
Segmentation evaluation  
Volume extinction values

### ABSTRACT

Colour information is usually not enough to segment natural complex scenes. Texture contains relevant information that segmentation approaches should consider. Martin et al. [Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 530–549] proposed a particularly interesting colour-texture gradient. This gradient is not suitable for Watershed-based approaches because it contains gaps. In this paper, we propose a method based on the distance function to fill these gaps. Then, two hierarchical Watershed-based approaches, the Watershed using volume extinction values and the Waterfall, are used to segment natural complex scenes.

Resulting segmentations are thoroughly evaluated and compared to segmentations produced by the Normalised Cuts algorithm using the Berkeley segmentation dataset and benchmark. Evaluations based on both the area overlap and boundary agreement with manual segmentations are performed.

© 2008 Elsevier B.V. All rights reserved.

### 1. Introduction

Image segmentation is often used as a first step in general object recognition in complex natural scenes, for example in [1–3]. The object recognition is simplified if the regions produced by the segmentation algorithm already correspond to “meaningful” objects. Nevertheless, unless it is made clear what the objects of interest in a scene are, even humans may not agree on the best segmentation of such a scene [4]. If a number of people are instructed to segment an image of an arbitrary scene, then each person will most likely produce a different segmentation of the image. This could be due to different interpretations of the scene or considering the scene at different scales.

Many algorithms for image segmentation are available, two of the most popular being the Normalised Cuts (NCuts) [5] and the Watershed [6]. Both of them require a way of measuring the similarity (or difference) between pixels in an image. The Watershed, for example, is usually applied to the gradient of an image. A particularly promising algorithm was presented in [7] for detecting the boundaries in an image based on brightness, colour and texture cues learned from human segmentations of a set of images. It calculates for every pixel in an image the probability that it is part of a boundary. Unfortunately, these boundaries are not suitable to be used as a gradient for a Watershed algorithm due to gaps in the

boundary lines. In this paper, we present a solution to this problem. This solution consists in filling the small gaps by applying a distance transform to the boundary image. We make use of the recently introduced distance function for greyscale images, the *quasi-distance function* [8]. This has the advantage that it can be applied directly to the greyscale boundary probability image and that no parameters need to be set. This is in contrast to the classic distance function, for which the boundary probability image would first have to be thresholded, and to smoothing filters, for which the size of the filter must be chosen.

Two different hierarchical segmentation approaches based on the Watershed are studied: the *hierarchy based on the volume extinction values* of the Watershed catchment basins that produces a partition with a specified number of regions, and the *Waterfall* that is iterated a given number of times producing a variable number of regions according to the image complexity. The segmentations produced are thoroughly evaluated and compared to segmentations produced by the NCuts algorithm using area and boundary-based segmentation evaluation measures. The manual segmentations from the Berkeley Segmentation Dataset and Benchmark are used as ground truth.

The paper is structured as follows. Section 2 is devoted to an overview of boundaries based on learning [7] and a presentation of the technique we propose to close the gaps in order to make them suitable for the Watershed approach. Section 3 summarises the two hierarchical Watershed algorithms. Area- and boundary-based segmentation evaluation is discussed in Section 4 and applied to evaluate the segmentations in Section 5. Section 6 concludes.

\* Corresponding author. Tel.: +43 1 58801 18359; fax: +43 1 58801 18392.

E-mail addresses: [hanbury@rip.tuwien.ac.at](mailto:hanbury@rip.tuwien.ac.at) (A. Hanbury), [marcotegui@cmm.ensmp.fr](mailto:marcotegui@cmm.ensmp.fr) (B. Marcotegui).

URL: <http://www.prip.tuwien.ac.at/people/hanbury> (A. Hanbury).

## 2. Gradients based on learning

The literature abounds with algorithms for computing gradients of both colour and greyscale images. While most of them may be used in conjunction with a Watershed algorithm to segment an image, almost all gradients tend to suffer from many strong responses in highly textured image regions, which prevents them from clearly delimiting textured areas. To solve this problem, Martin et al. introduced the boundaries based on learning [7], which we briefly review in the first part of this section. While these boundaries are better at delimiting highly textured areas, they cannot be directly used with a Watershed segmentation technique as the boundaries are usually not closed. We solve this problem by applying a distance function to close the gaps in the boundary image, as described in the second part of this section.

### 2.1. Boundaries based on learning

The boundaries based on learning approach introduced by Martin et al. [7] make use of brightness, colour and texture gradients to compute the boundaries. To calculate the gradients, a circular region is moved over the image. At each pixel, for a number of orientations of a line dividing the circle into two halves, the  $\chi^2$  histogram difference is evaluated for histograms of the features in the two halves. For colour, three 32-bin histograms of the values of  $L^*$ ,  $a^*$  and  $b^*$  in the CIELAB space (taken separately) are used; for texture, one 64-bin histogram of the textons used in [7] is used. For each feature, the gradient is taken to be the maximum value obtained over all the orientations of the line dividing the circle. The result of this algorithm is therefore a vector of four gradient values at every pixel (three colour and one texture).

These four gradients are combined to form a boundary probability. The weight for each gradient is obtained by logistic regression. As ground truth, human segmentations of the 200 images in the training group of the Berkeley segmentation dataset were used. Every pixel marked as a boundary by at least one person was considered as part of the ground-truth boundaries. We made use of the weights provided by the authors of [7] in their software.<sup>1</sup> The resultant boundary probabilities are in the range [0,1]. As an example, the boundaries detected in Fig. 1(a) are shown in Fig. 1(b).

### 2.2. Distance functions applied to boundary images

The boundary image produced by the algorithm outlined in the previous section (see Fig. 1(b)) seems to be a good gradient approximation: the values in the koala fur are low, while its body is well delimited. But, if we look more closely (see Fig. 1(c)), we can clearly see gaps in the boundary lines. This results in very few local minima in the boundary image (often only one), which makes applying Watershed-based segmentation difficult. Our solution to the problem is to attempt to close the gaps by calculating a *distance function* of the boundary image.

The classic distance function takes as input a binary image. It associates with each foreground pixel the distance to the closest background pixel (see Fig. 2 for an example). Overlapping binary objects may be segmented using the well-known approach [9,10] that combines the distance function and the Watershed. If a connected component contains several particles, its distance function will have a maximum in each particle. Thus, maxima of the distance function (pixels represented with a hatched pattern in Fig. 2(b)) mark the different particles contained in the connected component. The Watershed applied to the complement of the dis-

tance function (grey pixels in Fig. 2(b)) correctly separates the different particles of the connected component.

As the classic distance function must be applied to a binary image, applying it to the boundary image would require that the boundary image first be thresholded. To avoid the necessity of choosing this threshold, we make use of the quasi-distance introduced by Beucher [8]. The quasi-distance, qd, of a greyscale image  $I$  is defined as

$$\text{qd}(x, y) = \arg \max_i (\epsilon_{i-1}(x, y) - \epsilon_i(x, y)), \quad (1)$$

where  $\epsilon_i$  is the morphological erosion of size  $i$ , and  $(x, y)$  is a given pixel of the image  $I$ . In other words, the quasi-distance associates with each pixel  $(x, y)$  the size  $i$  of the erosion that produces the biggest change in greylevel, among all possible sizes of erosions. Thus the quasi-distance is able to characterize the size of objects in a greylevel image without first applying a threshold.

If we take the boundaries detected by the Martin et al. algorithm as the background, the distance function encodes the shortest distance to each of the detected boundary lines. The value of the distance function on the detected boundaries will be zero. Within small gaps in the detected boundaries the value of the distance function will be small. As we want the Watershed to take these boundaries as the edges of regions, we use the complement of this distance function, in which the detected boundaries will have the maximum possible value. The lower values of the distance function in small gaps lead to higher values in the complement, effectively closing the gaps in the topographical representation of the image used by the Watershed. The complement of the quasi-distance function applied to the boundary image in Fig. 1(b) is shown in Fig. 3(a), with a zoomed in area shown in Fig. 3(b).

## 3. Waterfall and volume extinction value hierarchies

The Watershed algorithm usually leads to a strong over-segmentation of the image. Several hierarchical approaches have been proposed to overcome this problem. In this paper, we will study two of these methods: the hierarchy based on the volume extinction values [11,12] and the Waterfall [13].

### 3.1. Watershed based on volume extinction values

During the flooding process of the Watershed a measure is associated with each merging. This measure, called the *extinction value*, corresponds to a geometric measure of the smallest lake involved in the merging and is used to evaluate the relevance of the merging. After the flooding process is completed, the “extinction” of small lakes is allowed (the merging is performed), whereas the biggest lakes (according to the measure) are preserved (the merging is not performed). In order to obtain a partition with  $N$  regions, the  $N - 1$  fusions with highest extinction values are avoided. Several measures have been proposed in the literature: the area of a lake that tries to obtain big regions regardless of their contrast, the depth of a lake that privileges contrasted regions regardless of their size and the volume of a lake that combines size and contrast. The use of volume provides a good approximation of human perceptual importance of a region and leads to the most useful segmentations. Fig. 4(d) shows the segmentation of Fig. 1(a) into 18 regions by this algorithm. We abbreviate this segmentation method as *Volume Watershed*.

### 3.2. Waterfall

The Waterfall [13] is a Watershed-based hierarchical segmentation approach. It consists in two steps:

<sup>1</sup> Downloadable on the Berkeley Segmentation Benchmark page: <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>.



Fig. 1. (a) An image and (b) its boundary probabilities (darker pixels indicate higher probability). (c) Detail of (b) showing the gaps in the contour.

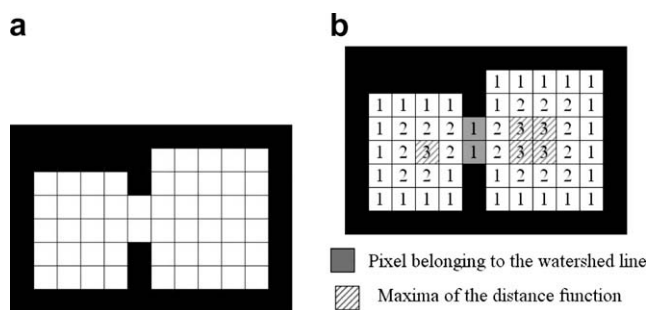


Fig. 2. (a) Binary image. (b) Associated distance function.

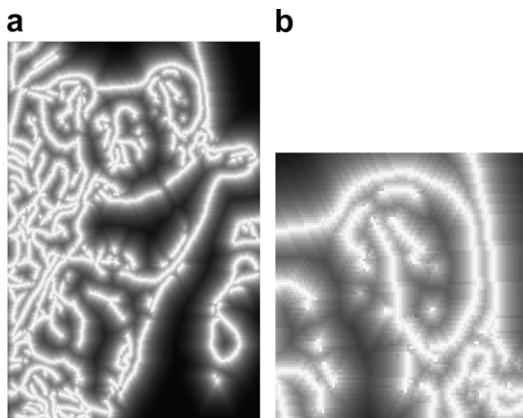


Fig. 3. (a) Complement of the quasi-distance on the boundary image. (b) Detail of (a).

- First, each region is filled with the value of the lowest pass point of its frontier. The pass point is the pixel where, during the flooding process associated with the Watershed, neighbouring “lakes” (regions) meet for the first time. A morphological reconstruction may be used for this purpose.
- Second, the Watershed of the resulting image is computed.

In the example of Fig. 5, the Watershed lines are indicated by arrows and only solid line arrows will be preserved by the Waterfall.

The process may be iterated until a single region covers the whole image, establishing a hierarchy among the frontiers produced by the Watershed. An efficient graph-based Waterfall algorithm is presented in [14].

An example of the Waterfall algorithm applied to the complement of the qd function of the detected boundary image is shown in Fig. 4. Image (a) shows the result of applying the Watershed algorithm to the complement of the quasi-distance function, image (b) is the result of applying the Waterfall algorithm once (referred to as level 1 of the hierarchy) and image (c) is the result of two iterations of the Waterfall (level 2).

### 3.3. Complete segmentation algorithm

We summarise here the algorithm used to perform the segmentation:

- (1) Calculate the learning-based boundaries (we use the combined colour and texture gradients [7]).
- (2) Calculate the complement of the quasi-distance function on the inverse boundary image.
- (3) Calculate the final partition using the Waterfall or the volume extinction value hierarchy on the complement of the distance function.

## 4. Segmentation evaluation

A number of methods for evaluating segmentations when ground truth is available have been proposed. They measure the similarity between a segmentation and a ground-truth segmentation by considering either the amount of region overlap [4], the proximity of the region boundaries to each other [7] or measurements of cluster goodness [15].

We evaluate our algorithms by using the area-based method from [4] and a newly introduced boundary-based method making use of the distance function. As ground truth we use the 300 colour images and their human segmentations from the Berkeley Segmentation Dataset and Benchmark. For each image, at least five segmentations produced by different people are available.

### 4.1. Area-based error measure

Two measures of the difference between two segmentations based on the overlapping areas of the segmentation regions are introduced in [4]: the global and local consistency errors (GCE and LCE). As the GCE is a tougher measure, we only use this measure.

Let  $S_1$  and  $S_2$  be the two segmentations of an image. The region  $R(S, p_i)$  is the set of pixels corresponding to the region in segmentation  $S$  that contains pixel  $p_i$ . A segmentation  $S_1$  is a *simple refine-*

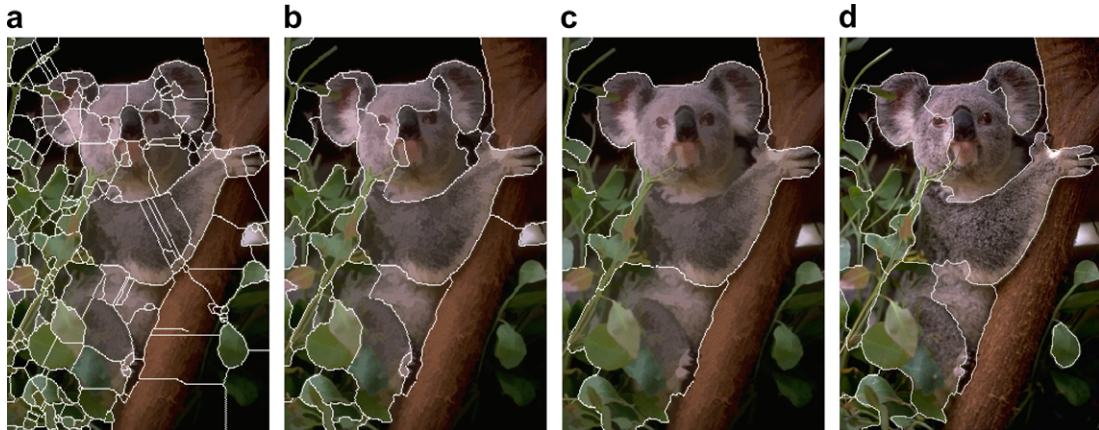


Fig. 4. (a) Watershed of the complement of the quasi-distance on the boundary probability image (level 0). (b) Waterfall level 1. (c) Waterfall level 2. (d) Watershed using volume extinction values (18 regions).

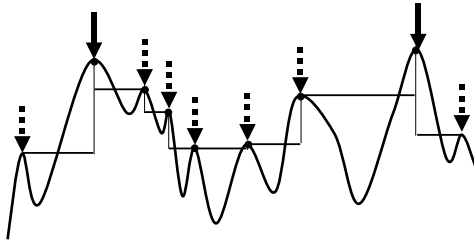


Fig. 5. Waterfall principle.

ment of  $S_2$  if at every pixel  $p_i$ ,  $R(S_1, p_i) \subseteq R(S_2, p_i)$ . The GCE is defined in terms of the local refinement error

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \setminus R(S_2, p_i)|}{|R(S_1, p_i)|}, \quad (2)$$

where  $\setminus$  denotes the set difference and  $|x|$  is the cardinality of set  $x$ . As can be seen, this error measure is not symmetric. If, at pixel  $p_i$ ,  $R(S_1, p_i) \subseteq R(S_2, p_i)$ , then  $E(S_1, S_2, p_i) = 0$ , but  $E(S_2, S_1, p_i) > 0$ . The GCE of segmentations  $S_1$  and  $S_2$  is defined as

$$\text{GCE}(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\}, \quad (3)$$

where  $n$  is the number of pixels and the sums are over all pixels. If  $S_1$  (resp.  $S_2$ ) is a simple refinement of  $S_2$  (resp.  $S_1$ ), then  $\text{GCE}(S_1, S_2) = 0$ . As the local refinement error is not symmetrical, the minimum of the local refinement error sums calculated in both directions is taken. Note that this measure is zero if one of the segmentations is only a single region covering the whole image, or if each pixel of one of the images is taken to be a region. This measure is therefore only useful if segmentations with a similar number of regions are compared.

#### 4.2. Boundary based error measure

Martin et al. [7] introduced a boundary-based error measure. They first compute the correspondence between machine boundary and human labelled boundary maps. This correspondence is performed by minimizing the distance in the image plane of pairs of matched pixels. If this distance is beyond a given threshold  $d_{\max}$ , they declare boundary pixels to be non-hits. As this boundary pixel matching procedure is time consuming, the authors propose strategies to speed up the process through the use of a bipartite graph matching algorithm.

We propose a simpler strategy based on the distance function. It allows the evaluation of the quality of a boundary map without a previous bipartite graph matching. Fig. 6 illustrates the proposed evaluation algorithm. Let us evaluate the quality of an automatic segmentation (Fig. 6(d)) with respect to a human made partition (Fig. 6(a)). Fig. 6(b) presents the distance of each image pixel to its closest human labelled boundary pixel. This operation has a complexity of  $O(n)$ . For each machine contour, we take the value of the computed distance (Fig. 6(b)). Pixels with a distance value lower than  $d_{\max}$  are considered as matched (i.e. a human boundary pixel is close enough) and pixels above  $d_{\max}$  are considered as false positives. Fig. 6(c) shows the pixels that have been “matched” (close enough to a manual labelled boundary). The parameter  $d_{\max}$  allows one to vary the maximum deviation accepted to match a contour point. We define the precision (measure currently used in the indexing context) as the ratio of matched machine contour pixels with respect to the number of contour pixels detected by the automatic algorithm:

$$\text{precision} = \frac{\text{Number of machine contour pixels matched}}{\text{Number of machine contour pixels}}. \quad (4)$$

We can repeat the process and compute the distance of each image pixel to the closest machine contour (Fig. 6(e)) and consider how many human labelled boundary pixels are close enough ( $< d_{\max}$ , see Fig. 6(f)) to machine boundary pixels. We define the recall as the ratio between manual contour pixels matched with respect with the number of manual contour pixels:

$$\text{recall} = \frac{\text{Number of manual contour pixels matched}}{\text{Number of manual contour pixels}}. \quad (5)$$

A high recall is obtained if most human labelled boundary pixels are closer than  $d_{\max}$  to a machine boundary pixel.

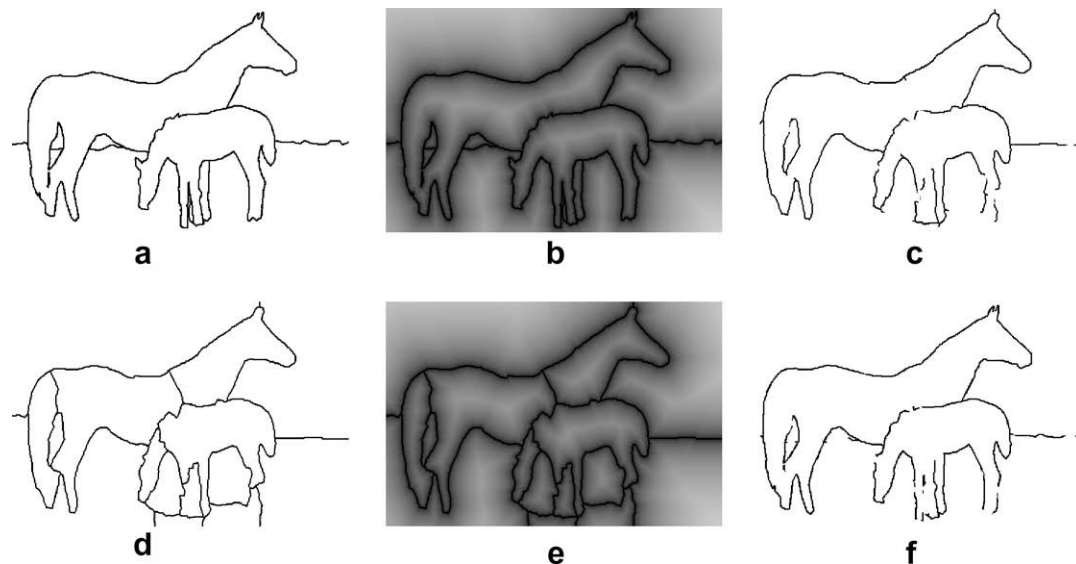
Each human segmentation therefore gives rise to a pair of precision–recall values  $(P, R)$ . To summarise these values in a single figure, the  $F$ -measure, defined as  $F = 2PR/(R + P)$  is used.

The proposed measures are similar to those proposed by Martin et al. [7] because they consider that a simple binary counting after matching is sufficient. The advantage of our method is that we avoid the matching procedure which is complex and time consuming.

## 5. Results

We compare the segmentations of the two hierarchical Watershed-based approaches operating on the complement of the quasi-distance of the colour and texture boundary image to the segmentations produced by the NCuts algorithm. For NCuts, we





**Fig. 6.** Computation of the boundary error measure. (a) Manual segmentation. (b) Distance to the closest manual contour. (c) Machine contours matched (considered for precision calculation). (d) Machine segmentation. (e) Distance to the closest machine contour. (f) Manual contours matched (considered for recall calculation).

use the implementation by Shi<sup>2</sup> [5], which requires that one specifies in advance the number of regions required. We applied the NCuts algorithm to two types of weighting function. The first is calculated from the multiscale greyscale gradient [16] using the intervening contour method originally introduced in [17] and included in the NCuts implementation used. The second is calculated from the learning-based colour and texture boundary probability image using a simplified intervening contour method that does not include orientation energy information (which is not available in the boundary probability images). The weighting based on the former was found to lead to better segmentations, so we present the results using this weighting function in the following analysis.

For the Waterfall algorithm, as level 1 of the hierarchy is almost always over-segmented, we evaluate level 2. The mean of the number of regions obtained over all 300 images by this segmentation algorithm is 5.8, so we compare them to the NCuts algorithm producing six regions.

For the Watershed using volume extinction values, the number of regions should be specified, as is the case for the NCuts. We chose 18 regions for the comparison, the mean number of regions over all the human segmentations. Segmentation results produced by these algorithms applied to the 300 images of the Berkeley segmentation dataset are available on the web.<sup>3</sup> Some example segmentations are shown in Fig. 7.

To evaluate a segmentation algorithm, it was first applied to each of the 300 images. Then, for each image, the GCE (area based measure), precision, recall and *F*-measure (boundary based measures) of the segmentation produced by the algorithm with respect to each of the available human segmentations for that image were calculated. The mean values of these measures were calculated as the mean over the measure for each manual segmentation. One of the disadvantages of the boundaries based on learning is their long computation time. For the test images used (of size  $321 \times 481$  pixels), the mean computation time for the boundaries was 1.9 min on a Pentium 4 computer. The Waterfall and Watershed segmentations require on average 0.05 s irrespective of the number of regions produced. In order for the NCuts segmentation to be computation-

ally tractable, the size of the image is reduced to  $160 \times 240$  pixels. The gradient used for the NCuts requires on average 9 s computation time on such a reduced size image. For comparison, detecting the learning based boundaries on a reduced size image requires on average 21 s. The NCuts segmentation requires an average of 23 s to segment the image into six regions, and 35 s to segment it into 18 regions. The region labelled images produced by the NCuts algorithms were enlarged to the original image size by pixel replication, leading to ragged region boundaries (see Fig. 7).

### 5.1. Area-based comparison

The mean GCE values for all segmentation algorithms evaluated are shown in the left column of Table 1. Histograms showing the distributions of the GCE values of each of the manual segmentations are shown in Fig. 8(a)–(d). Cumulative histograms are shown in Fig. 8(e)–(f). These curves indicate the fraction of GCE values that are below the GCE value on the *x*-axis. Algorithms with lower GCE values will produce curves that climb faster and hence lie more to the left. Note that some of the segmentations at level 2 of the Waterfall hierarchy consist of only one region. As the GCE for such a segmentation is zero, we chose to use level 1 of the Waterfall hierarchy if level 2 contained only a single region.

For the segmentations into 18 regions, the GCE values produced by both the Volume Watershed and the NCuts are almost identical. On the other hand, when segmenting the image into a small number of regions, the mean GCE for the Waterfall level 2 is much smaller than for the NCuts with six regions. This suggests that the regions produced by the Waterfall method are a better match to the human segmentations, although this is discussed further after considering the boundary based evaluation.

### 5.2. Boundary-based comparison

Fig. 9 presents the comparison of segmentation results using the NCut approach and the hierarchy based on the volume extinction values, both with 18 regions. The evaluation method used is the one presented in Section 4.2 with  $d_{\max} = 4$ , which represents 0.70% of the image diagonal. In Fig. 9(a), we have represented the histogram of the precision (contours of the automatic segmentation that are closer than  $d_{\max}$  to a manually drawn boundary)

<sup>2</sup> <http://www.cis.upenn.edu/jshi/software/>.

<sup>3</sup> [http://muscle.prip.tuwien.ac.at/IVC\\_segresult/](http://muscle.prip.tuwien.ac.at/IVC_segresult/).



(a) WF2 GCE:0.25 (b) NC6 GCE:0.37 (c) V18 GCE:0.24 (d) NC18 GCE:0.27  
P:0.55 R:0.39 F:0.42 P:0.34 R:0.22 F:0.25 P:0.50 R:0.65 F:0.52 P:0.40 R:0.56 F:0.44



(e) WF2 GCE:0.14 (f) NC6 GCE:0.26 (g)V18 GCE:0.20 (h) NC18 GCE:0.17  
P:0.79 R:0.45 F:0.57 P:0.77 R:0.48 F:0.59 P:0.75 R:0.67 F:0.71 P:0.61 R:0.67 F:0.64



(i) WF2 GCE:0.38 (j) NC6 GCE:0.33 (k) V18 GCE:0.26 (l) NC18 GCE:0.17  
P:0.52 R:0.27 F:0.35 P:0.63 R:0.55 F:0.59 P:0.47 R:0.56 F:0.51 P:0.61 R:0.67 F:0.64

**Fig. 7.** Examples of segmentations produced by the four methods tested on three images. The leftmost column shows the waterfall level 2 (WF2), the column second from left shows the NCuts with six regions (NC6), the third column shows the volume Watershed with 18 regions (V18) and the rightmost column shows the NCuts with 18 regions (NC18). Below each image are the values of each of the segmentation comparison measures: global consistency error (GCE), precision (P), recall (R) and  $F$ -measure (F).

and in Fig. 9(b) the histogram of the recall (contours of the manual segmentation that are closer than  $d_{\max}$  to an automatic contour). We see that the hierarchy based on volume extinction values generally outperforms the NCuts, because its histogram lies further to the right than the NCuts histogram, meaning that the precision and recall are concentrated at higher values.

Fig. 10 presents the comparison of the NCut with six regions and the second level of the hierarchy based on Waterfalls. As stated be-

fore, the first level of the Waterfall hierarchy is kept if the second level contains only one region. Again, the histograms of the precision and recall for the Waterfall lie further to the right than the NCut histograms. The mean precision, recall and  $F$ -measure over all the human segmentations are shown in the rightmost columns of Table 1. The mean  $F$ -measures resulting from the segmentations are low, demonstrating that the segmentations produced by all methods are not at all close to human segmentations.

**Table 1**

The mean values over all manual segmentations for GCE, precision, recall and  $F$ -measure for various segmentation algorithms

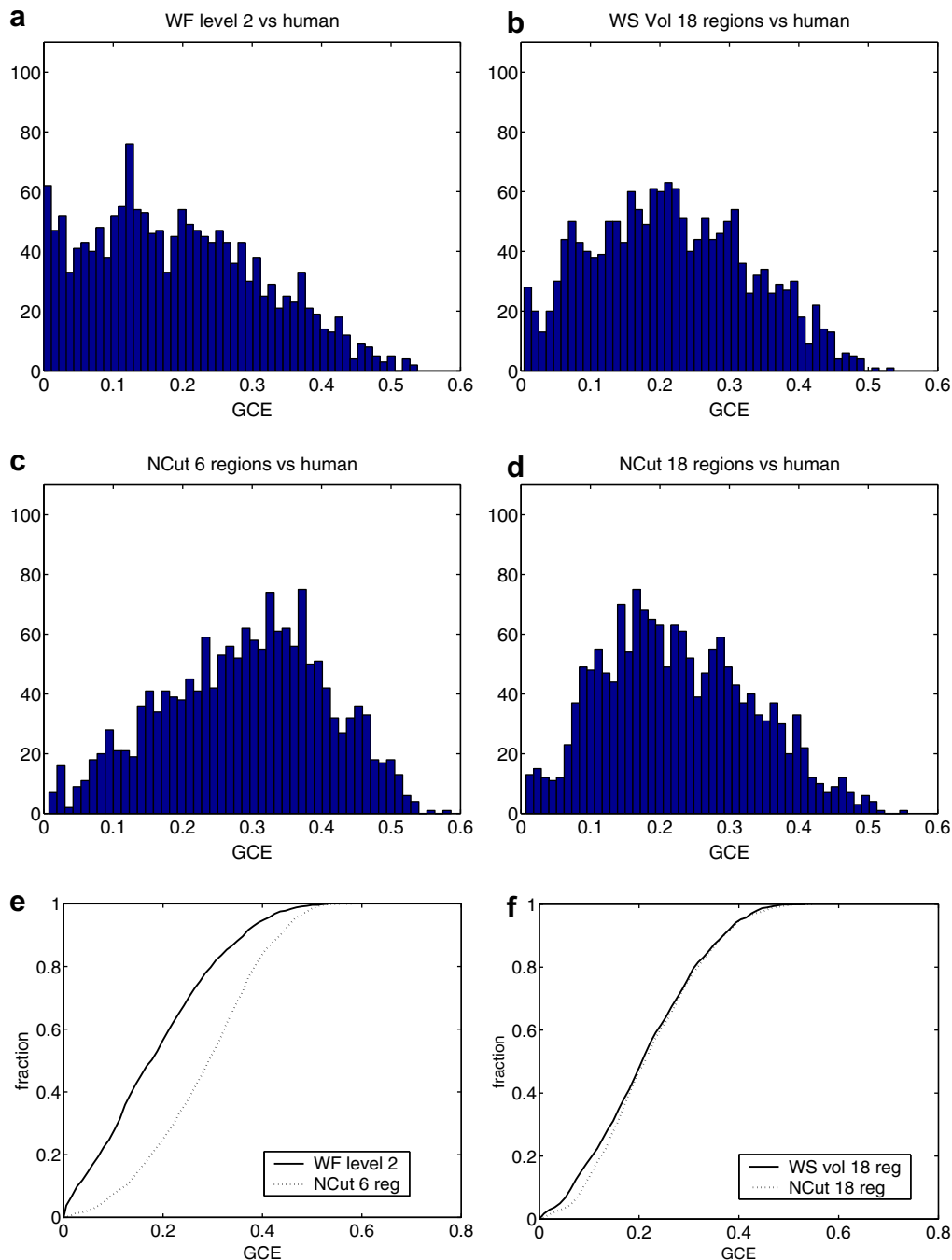
Method	GCE	Precision	Recall	$F$ -measure
WF level 2	0.19	0.64	0.37	0.44
NCuts (6 regions)	0.29	0.52	0.38	0.42
WS Vol (18 regions)	0.22	0.54	0.60	0.55
NCuts (18 regions)	0.23	0.44	0.58	0.48

These are the Waterfall algorithm (WF) for level 2 of the hierarchy, the Watershed using volume extinction values (WS Vol) and the NCuts algorithm. Note that better agreement with the ground truth is indicated by smaller GCE values, but by larger precision, recall and  $F$ -measure values.

### 5.3. Discussion

In this section, we analyse the global trends as well as relating them to segmentations of specific images. For the segmentations in Fig. 7, the mean values of GCE, precision, recall and  $F$ -measure calculated over the manual segmentations corresponding to each image are shown for each segmentation.

We begin by considering the evaluation based on boundaries. In Table 1, the mean recall values for the two methods producing a low number of regions are similar to each other, as are the recall values for the two methods producing 18 regions. Larger differences are



**Fig. 8.** Histograms of the distribution of the GCE for each of the human segmentations for: (a) level 2 of the Waterfall algorithm, (b) the watershed with volume extinction values for 18 regions, (c) the NCuts algorithm with six regions, and (d) the NCuts algorithm with 18 regions. (e) Cumulative histogram of (a) and (c). (f) Cumulative histogram of (b) and (d).

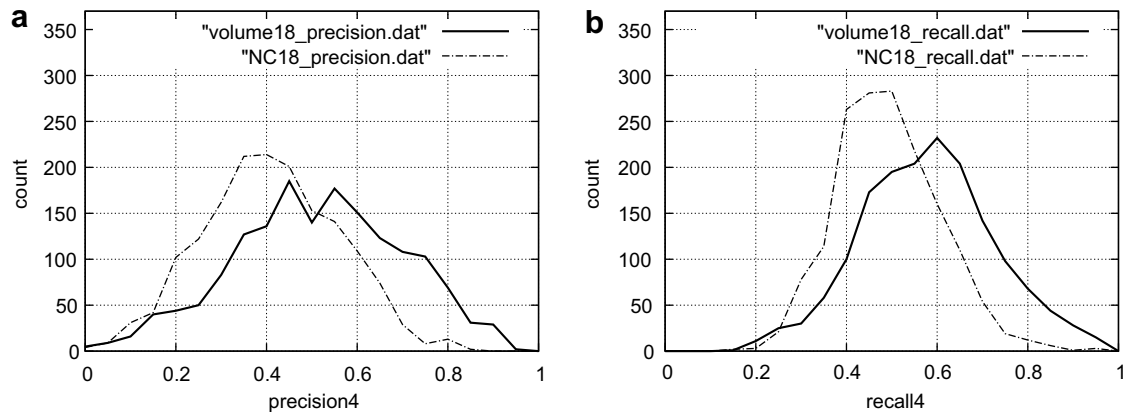


Fig. 9. Comparison of hierarchy based on volume extinction values and NCuts with 18 regions. (a) Precision histogram. (b) Recall histogram.

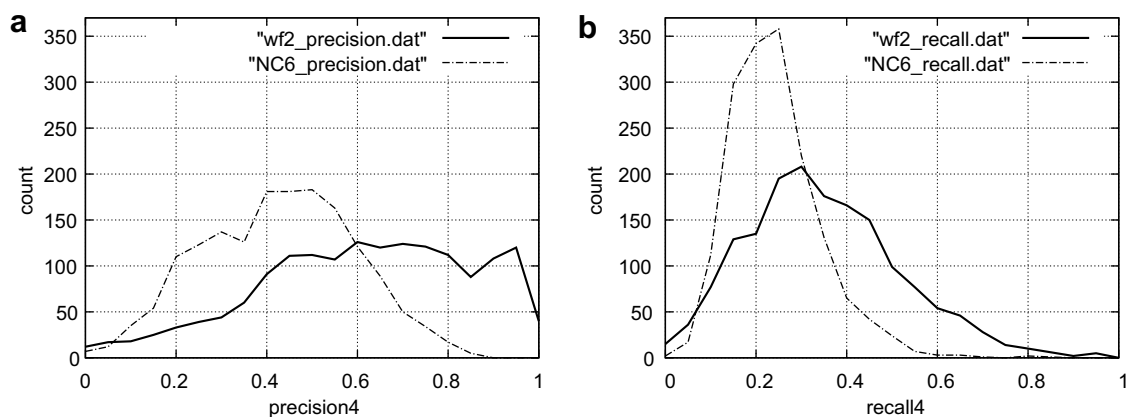


Fig. 10. Comparison of waterfall level 2 and NCuts with six regions. (a) Precision histogram. (b) Recall histogram.

visible in the mean precision values, where the morphological methods have larger values. This demonstrates that in general, while both segmentation methods find a similar proportion of the segment boundaries corresponding to the ground truth, the morphological methods find fewer false boundaries. A possible explanation for this is that the NCuts has a tendency to produce regions of similar size, often leading to an over-segmentation of homogeneous regions. These spurious region boundaries lead to a lower precision as they are in general not close to any lines in the manual segmentations. A good example can be seen in Fig. 7, where images (g) and (h) have identical recall values, but the precision for the Watershed approach is much larger than for the NCuts approach. For the methods producing 18 segments, the mean GCE values differ only by 0.1. This is most likely because the GCE is designed so as to ignore over-segmentation. The over-segmented regions produced by the NCuts therefore do not affect this error much. This is also well demonstrated by images (g) and (h) of Fig. 7. For this image, the NCuts segmentation has a lower GCE value than the volume Watershed, even though the background is over-segmented.

For the two methods producing a small number of regions, the Waterfall algorithm has a lower mean GCE than the NCuts (also visible in Fig. 8(e)). Due to the design of the GCE, this could indicate two possibilities. The first is that the segmentations produced are closer to the manual segmentations, as illustrated by segmentations (a) and (b) in Fig. 7, where the GCE for the Waterfall segmentation is much smaller than for the NCut with six regions, agreeing with a visual evaluation of the segmentations. The second is that the number of regions in the Waterfall segmentation is less than 6, which also often leads to a smaller GCE. This can be seen in seg-

mentations (e) and (f) of Fig. 7, where segmentation (e) is visually worse than segmentation (f), but has a lower GCE. The visual judgement for these two segmentations is better represented by the boundary measures.

There are also images for which the NCuts segmentations are better than the Watershed approaches, as can be seen in the bottom row of Fig. 7. Here, all measures indicate that both NCuts segmentations perform better, which can be confirmed by visual evaluation.

Over all images, for the segmentations into a small number of regions, 60% of the  $F$ -measures are larger for the Waterfall level 2 than for the NCuts (six regions). For the segmentations into 18 regions, 85% of the  $F$ -measures are larger for the Volume Watershed than for the NCuts.

## 6. Conclusion

In this paper, we combine the colour and texture boundaries based on learning introduced by Martin et al. [7] with hierarchical Watershed-based segmentation. These boundaries are not directly suitable for Watershed-based algorithms due to gaps in the boundary lines. We have solved the problem by calculating the complement of the quasi-distance function applied to the boundary image.

Two different hierarchical segmentation approaches based on the Watershed have been studied: volume extinction values and Waterfalls. The segmentations obtained compare favorably with NCuts results. We have used the Berkeley Segmentation Dataset for comparison. For evaluation purposes, we have used the area-based method proposed in [4] and a newly introduced boundary-

based evaluation method. The proposed method makes use of the distance function between manual and machine contours.

In general, the Watershed approaches produced boundaries matching ground-truth segmentations with higher precision. The recall of both the Watershed and NCuts methods is similar. The Waterfall-based approach has the advantage that the number of regions does not need to be specified in advance. It nevertheless has the disadvantage that it tends to produce too many regions at the first level of its hierarchy and too few at the second level [18]. It should be possible to change the region merging criteria to improve on this. There is a version of the NCuts which determines the number of regions automatically [16], but we currently have no implementation of it.

As further work, we intend to investigate other region merging criterion for the Waterfall algorithm, to choose a level between the over-segmentation of level 1 and the under-segmentation of level 2. We plan to compare this with the version of the NCuts which includes criterion on when to stop splitting regions. The calculation time of the boundaries based on learning is unacceptably high. We plan to either accelerate it in some way or find a good approximation with a lower computation time.

### Acknowledgements

This work was supported by the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04) and the European Union Network of Excellence MUSCLE (FP6-507752). The area-based error measure code was written by Adrian Ion and Branislav Mičušík.

### References

- [1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, M.I. Jordan, Matching words and pictures, *Journal of Machine Learning Research* 3 (2003) 1107–1135.
- [2] P. Carbonetto, N. de Freitas, K. Barnard, A statistical model for general contextual object recognition, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004, pp. 1:350–1:362.
- [3] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, *Journal of Machine Learning Research* 5 (2004) 913–939.
- [4] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: *Proceedings of the 8th International Conference on Computer Vision*, 2001, pp. II:416–II:423.
- [5] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8) (2000) 888–905.
- [6] S. Beucher, F. Meyer, The morphological approach to segmentation: the watershed transformation, in: E. Dougherty (Ed.), *Mathematical Morphology in Image Processing*, Marcel Dekker, New York, 1993, pp. 433–481 (Chapter 12).
- [7] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (5) (2004) 530–549.
- [8] S. Beucher, Numerical residues, *Image and Vision Computing* 25 (4) (2007) 405–415.
- [9] C. Lantuejoul, S. Beucher, On the use of geodesic metric in image analysis, *Journal of Microscopy* 121 (1981) 39–49.
- [10] P. Soille, *Morphological Image Analysis*, 2nd ed., Springer, Berlin, 2002.
- [11] C. Vachier, F. Meyer, Extinction values: a new measurement of persistence, in: *Proceedings of the IEEE Workshop on Non Linear Signal/Image Processing*, 1995, pp. 254–257.
- [12] F. Meyer, An overview of morphological segmentation, *International Journal of Pattern Recognition and Artificial Intelligence* 15 (7) (2001) 1089–1118.
- [13] S. Beucher, Watershed, hierarchical segmentation and waterfall algorithm, in: *Mathematical Morphology and its Applications to Image Processing*, *Proceedings of the ISMM'94*, 1994, pp. 69–76.
- [14] B. Marcotegui, S. Beucher, Fast implementation of waterfall based on graphs, in: *Mathematical Morphology and its Applications to Image Processing*, *Proceedings of the ISMM'05*, 2005, pp. 177–186.
- [15] X. Jiang, C. Marti, C. Irniger, H. Bunke, Distance measures for image segmentation evaluation, *EURASIP Journal on Applied Signal Processing* (2006), Article ID 35909, 10 pp.
- [16] J. Malik, S. Belongie, T. Leung, J. Shi, Contour and texture analysis for image segmentation, *International Journal of Computer Vision* 43 (1) (2001) 7–27.
- [17] T. Leung, J. Malik, Contour continuity in region-based image segmentation, in: *Proceedings of the European Conference on Computer Vision*, 1998, pp. 544–559.
- [18] A. Hanbury, B. Marcotegui, Waterfall segmentation of complex scenes, in: *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2006, pp. I:888–I:897.

Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



## Shape ultimate attribute opening<sup>☆</sup>

Jorge Hernández, Beatriz Marcotegui<sup>\*</sup>

Mines-ParisTech, CMM, Centre de morphologie mathématique, Mathématiques et Systèmes, 35 rue St Honoré 77305-Fontainebleau-Cedex, France

### ARTICLE INFO

#### Article history:

Received 29 September 2010  
Received in revised form 8 March 2011  
Accepted 17 May 2011

#### Keywords:

Ultimate opening  
Attribute opening  
Mathematical morphology  
Shape information  
Image segmentation  
Facade analysis  
Scene-text detection  
Cell segmentation

### ABSTRACT

The ultimate opening (*UO*) is a powerful segmentation operator recently introduced by Beucher [1]. It automatically selects the most contrasted regions of an image. However, in the presence of nested structures (e.g. text in a signboard or windows in a contrasted facade), interesting structures may be masked by the containing region. In this paper we focus on ultimate attribute openings and we propose a method that improves the results by favoring regions with a predefined shape via a similarity function. An efficient implementation using a max-tree representation of the image is proposed. The method is validated in the framework of three applications: facade analysis, scene-text detection and cell segmentation. Experimental results show that the proposed method yields better segmentation results than *UO*.

© 2011 Elsevier B.V. All rights reserved.

### 1. Introduction

Segmentation is a fundamental problem in image analysis to distinguish between objects of interest and “the rest”. It creates a partition of the image into disjoint and uniform regions, according to some features such as gray value, color, or texture [2]. In literature, deformable models [3] and level sets [4] have become the most studied techniques for shape segmentation, due to their ability to adapt to the specific shape of the object of interest. These methods generally are too sensitive to the initial contour position and to the stop criterion. More robust algorithms have been proposed [5,6] in order to deal with those drawbacks, at the expense of a higher computational complexity.

In this paper, an extended version of [7], we describe a fast morphological segmentation approach that combines gray-scale information and prior shape knowledge. The method is easily parameterizable because it uses simple shape features. An overview of morphological segmentation is presented by Meyer in [8] where a unified framework for supervised or unsupervised, multi-scale or single scale, color or gray-scale and 2D or 3D images is introduced. Shape information has been used in mathematical morphology as attribute operators (connected filters [9]). Attribute operators compute criteria describing the shape or size of each connected

component and then decide which components are preserved or discarded. Firstly, Vincent [10] presented area openings and then Breen and Jones [11] extended them to attribute openings (*AO*) and thinnings. Several applications have been developed using attribute filters such as: enhanced filtering [12], image compression [13] and image retrieval [14]. Maragos [15] proposed a multi-scale shape-size description using morphological opening and closing filters, named pattern spectrum, which is related to the notion of granulometries [16]. Urbach [17] extended them to shape pattern spectrum (shape granulometry), by using an attribute thinning as a shape operator. A geometric spectrum is presented in [18] for quantifying the geometric features on multidimensional binary images, which extends the work of Goutsias and Schonfeld [19]. They presented a morphological shape decomposition that uses a sequence of structuring elements and a sequence of set transformations. The obtained decomposition is called a generalized morphological segmentation.

Furthermore, a large number of segmentation methods exist to compute prior shape knowledge based on watershed [20–22]. They impose a priori shape knowledge, usually smoothness of the region contour, on the segmentation result. In [21], the snake energy criterion is used to merge over-segmented regions produced by the watershed while in [20], watershed segmentation is defined as an energy minimization. Hamarneh and Li [22] used k-means to incorporate shape knowledge before the segmentation stage. Besides, a new morphological operator, named ultimate opening (*UO*) [1], has been increasingly used as a powerful segmentation method due to its various advantages: non-parametric operator, segmentation of contrasted structures, intrinsically multi-scale, etc.

Our approach integrates the shape information into ultimate attribute opening *UAO*. In contrast to using only the difference of

<sup>☆</sup> This paper has been recommended for acceptance by Sinisa Todorovic.

<sup>\*</sup> Corresponding author. Tel.: +33 1 64694706.

E-mail addresses: [jorge.hernandez@mines-paristech.fr](mailto:jorge.hernandez@mines-paristech.fr) (J. Hernández), [beatriz.marcotegui@mines-paristech.fr](mailto:beatriz.marcotegui@mines-paristech.fr) (B. Marcotegui).

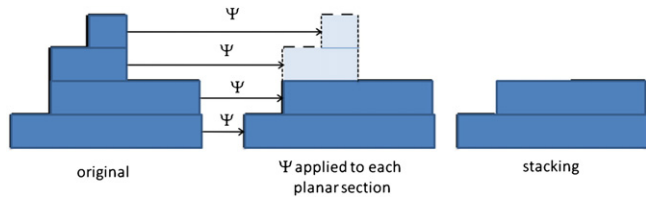


Fig. 1. Connected operator  $\Psi$  extended to functions: it applies to each planar section and stacks the results.

openings to locate regions, as done in the *UO*, the proposed method uses a similarity function defined through a prior knowledge of the shapes. This similarity function is based on the characteristics of the connected components *CCs* in images (shapes). Urbach et al. defined in [23] vector-attribute filters based on shape descriptors. They filter out *CCs* different from the prior one, based on a threshold on the vector distance. Our method combines the contrast with the attribute distance. Thus, contrasted shapes are extracted but favoring those which are shaped like the prior one. Experimental results are shown on real applications: facade analysis, scene-text detection and cell segmentation. In scene-text segmentation and cell segmentation, attribute histograms are analyzed in order to determine the main shape attributes. The method improves the result with respect to the classical *UAO*.

The paper is organized as follows. In Section 2, the theoretical background of connected operators, the *AO*, Salembier's max-tree and the *UO* are presented. Qualities and drawbacks of the *UO* are illustrated. Section 4 describes our method and introduces shape information into the *UO* Definition. In Section 5, experimental results are shown and the advantages of our method are illustrated. Finally, conclusions and future work are drawn in Section 6.

## 2. Theoretical background

In this section we introduce the ultimate attribute opening (*UAO*) operator, as well as some related aspects such as flat zones or the max-tree structure, used to implement the *UAO* efficiently. Let us consider  $X$ , a binary image on a space  $E \subseteq Z^n$ , and  $I$  a gray-scale image:  $E \rightarrow Z$ .

### 2.1. Flat zones and connected operators

Connected operators have been introduced by Serra and Salembier in [9]. These operators do not modify individual pixels, but act on connected components (*CC*), for binary images, or on flat zones (for gray level images). According to [9], a *CC* of a set, is a set of points that are connected by a path included in the set, while flat zones (*FZ*) of an image  $I$  are the largest *CCs* where the image is constant. We denote  $FZ(I)$  the set of flat zones of image  $I$ .

Connected operators ( $\Psi$ ) can only merge *FZ* but not break them into pieces. As a consequence, connected operators cannot add new

boundaries, nor shift existing contours, but only simplify the image by removing contours. In other words, each *FZ* of the original image should be included in a single *FZ* of the resulting image:  $\forall B \in FZ(I), \exists C \in FZ(\Psi(I)) / B \subset C$ .

### 2.2. Attribute opening

Attribute openings (*AO*) are connected operators. For binary images, each *CC* of the original image is analyzed and an attribute is computed on it. Examples of attributes are the *CC* area, height or width of the bounding box or any other measure we can define on a given *CC* such as circularity (other examples can be found in [11]). An increasing criterion  $T$  is defined based on this attribute, usually in the form of a threshold on the attribute. A criterion  $T$  is said to be increasing if the fact that a set  $C$  satisfies  $T$ , implies that any other set  $B$  containing  $C$  satisfies  $T$  too. Then the *AO* of  $I$  is equal to the union of *CC* of  $I$  verifying the criterion  $T$ . For example, an area opening of size  $\lambda$  ( $\gamma_\lambda^{area}(I)$ ) is equal to the union of *CCs* of  $I$  which area is greater than or equal to  $\lambda$ .

Attribute openings can be extended to gray level images by applying the transform to each planar section of the image  $X_t(I)$ , where  $X_t(I) = \{x \in Z | I(x) \geq t\}$ , and "stacking" the results ( $X_t(\Psi(I)) = \Psi(X_t(I))$ ) [9]. Fig. 1 illustrates this process. Thus, an attribute opening filters out peaks (bright regions), until the regional maximum verifies the criterion.

Fig. 2 shows a gray-scale image and two area openings, of sizes 100 and 500 pixels respectively. The regional maxima of these images are represented in blue. We can see that while the original image has many small maxima, the filtered images have larger and fewer maxima. The remaining maxima have an area of at least 100 and 500 pixels respectively, according to the corresponding opening size.

Efficient algorithms based on hierarchical queues [10], max-trees [24] and union find [25] have largely contributed to the diffusion of these operators. A comparison of these three methods has been presented in [26].

### 2.3. Ultimate opening

Ultimate opening (*UO*), has been introduced by Beucher in [1]. This is a non-parametric method and a non-linear scale-space based on morphological numerical residues to extract *CCs*. It analyzes the evolution of an image with a series of openings of increasing sizes:  $\gamma_\lambda(I)$ ,  $\lambda = \{0, 1, \dots, N-1\}$  and  $N$  the maximum opening size considered. *UO* studies the series of residues between consecutive openings  $r_\lambda(I) = \gamma_\lambda(I) - \gamma_{\lambda+1}(I)$  and keeps two significant outputs:

- $R_\theta(I)$ , the maximal residue  $r_\lambda(I)$ . It is the strongest change generated by an opening. An important structure is supposed to be filtered out by this opening and the corresponding residue estimates its contrast.
- $q_\theta(I)$ , the size of the opening leading to the maximal residue. If several openings lead to the same maximum  $r_\lambda(I) = R_\theta(I)$ , the

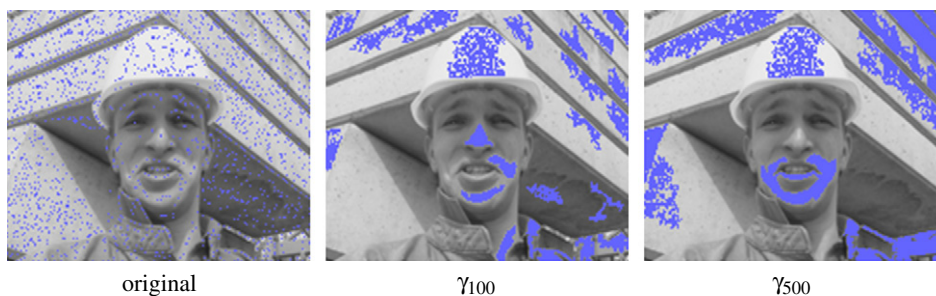


Fig. 2. Area openings (regional maxima in blue).



largest  $\lambda$  among them is chosen.  $q_\theta(I)$  is set to 0 for pixels where all residues are null. This happens for the minimum of the image or for objects larger than  $N$ , the largest opening size considered.

According to Beucher [1], the definition describing the  $UO$  is written as:

**Definition 1.** The ultimate opening operator,  $\theta$ , of an image  $I$  is given by:

$$\theta(I) : I \xrightarrow{\theta} (R_\theta(I), q_\theta(I)) \quad (1)$$

$$R_\theta(I) = \max_\lambda (r_\lambda(I)) = \max_\lambda (\gamma_\lambda(I) - \gamma_{\lambda+1}(I)) \quad (2)$$

$$q_\theta(I) = \begin{cases} \max\{\lambda + 1 | r_\lambda(I) = R_\theta(I)\} & R_\theta(I) > 0 \\ 0 & R_\theta(I) = 0 \end{cases}$$

where,  $\gamma_\lambda$  is an opening of size  $\lambda$ .

Beucher introduced  $UO$  with morphological openings. In this paper attribute openings (see Section 2.2) will be used instead, leading to ultimate attribute opening ( $UAO$ ).  $UAO$  is a connected operator, that

can be implemented efficiently on a max-tree structure (see Section 2.4). Fig. 3 shows three examples on real applications by using an ultimate height opening. The height is defined as the y-extent of a CC. The same non-parametric operator is applied on these images. The first example illustrates how the  $UO$  extracts internal structures (windows, doors, etc.) from facade images. In Fig. 3(b), we can appreciate the scene-text segmentation, by showing the operator powerfulness. Finally, a cell segmentation example is shown in Fig. 3(c). These applications will be further studied in Section 5.

Various applications have been developed using the  $UO$ : image analysis to measure the granulometry of rocks [27], automatic localization of text [28] and document image binarization [29].  $UAO$  can be computationally expensive, if applied naively from its definition. Fabrizio in [30], proposes an efficient implementation based on the max-tree representation [24].

#### 2.4. Max-tree

The max-tree, by duality min-tree, has been introduced by Salembier [24] as a structure to compute connected operators. Here

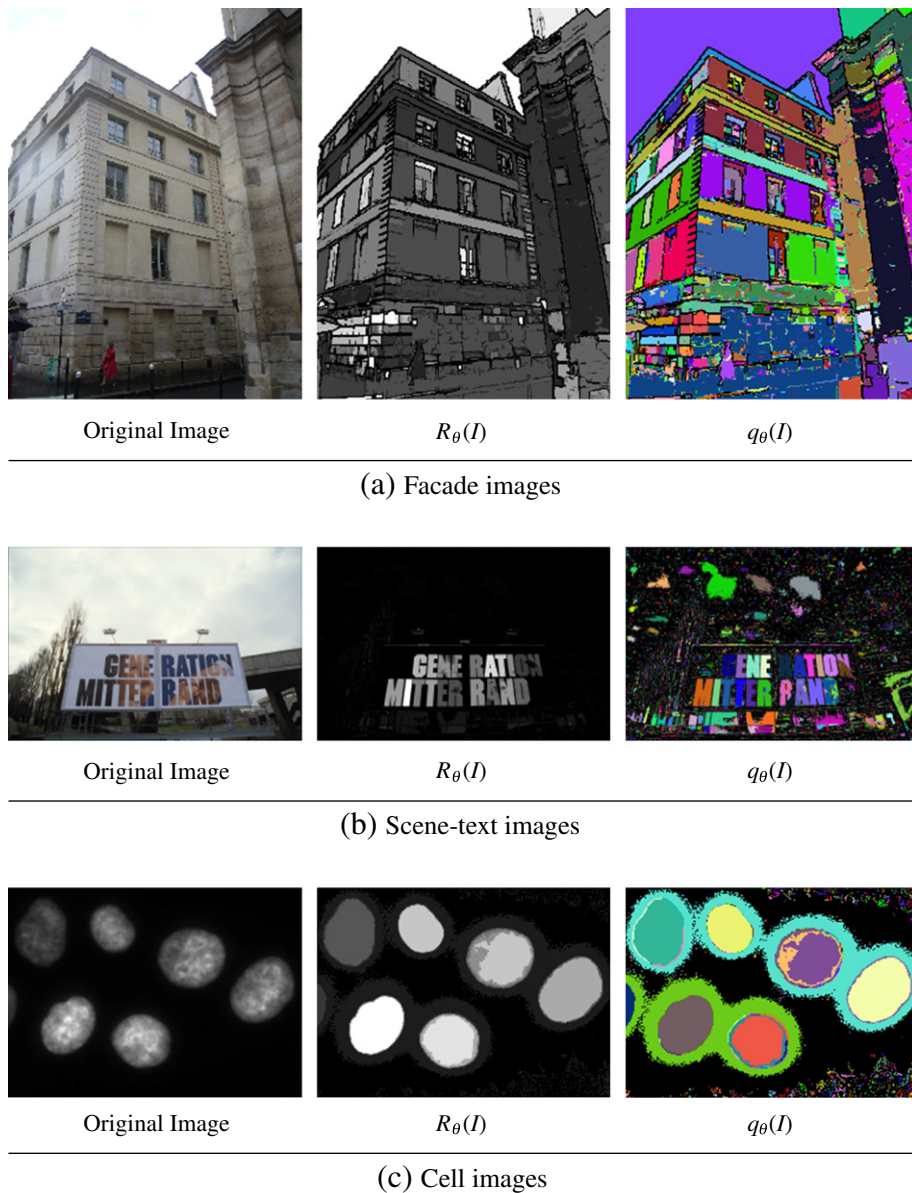


Fig. 3. Ultimate height opening on real images.  $N$  is equal to the vertical size of the image.  $q_\theta(I)$  is represented in false colors in order to see the segmented regions.

we illustrate the concept of max-tree. For an exhaustive definition please refer to [24]. The max-tree is a tree, in the sense of graph theory, which gives a multi-scale representation of an image. Nodes represent the binary CCs of  $X_t(I)$ , the thresholded versions of image  $I$  at level  $t$ . Parent-child links describe inclusion relationships between the corresponding CCs. Thus, the root node corresponds to the whole image while leaf nodes correspond to the image maxima. One or several attributes, computed on CCs, can be associated with nodes, allowing an efficient computation of attribute openings. An example is presented in Fig. 4. The attribute  $\kappa$  used is the height (y-extent) of the associated CC. Root node,  $A_0^4$ , of level 0 and of height attribute  $\kappa=4$  has two children  $B_1^3$  at level 1 and  $C_2^3$  at level 2, both with an attribute  $\kappa=3$ . Connected component  $C$  remains unchanged from  $X_1(I)$  to  $X_2(I)$ . In that case, the maximum threshold for which the connected component is the same, is assigned to the node (level 2 in our example). Each branch leads to a leaf-node ( $E_5^1$  and  $F_6^2$ ) corresponding to the regional maxima of the image, at levels 5 and 6 respectively.

An attribute opening  $\gamma_\lambda$  removes nodes  $n$  of attribute  $\kappa(n)$  smaller than  $\lambda$ . Fig. 5 shows the attribute opening results, with increasing  $\lambda$  values. These openings are obtained by pruning the tree at the corresponding dot-lines.

Fabrizio in [30] implemented a fast UAO on max-tree. The definition of the ultimate attribute opening involves a series of attribute openings (connected operators). Therefore, the max-tree is a suitable structure for its implementation [24]. Once the max-tree is computed, the different openings are obtained by simply pruning the tree nodes, with a simple threshold of their attribute. The residue between two consecutive openings is computed as the gray level difference of a node and the gray level of its first ancestor with a different attribute, as follows:

$$r_\lambda(\text{child}) = \begin{cases} t(\text{child}) - t(\text{parent}) & \kappa(\text{parent}) \neq \kappa(\text{child}) \\ t(\text{child}) - t(\text{parent}) + r_\lambda(\text{parent}) & \text{otherwise} \end{cases} \quad (3)$$

where *child* and *parent* are two linked nodes of the max-tree,  $t(n)$  the gray level associated to node  $n$  and  $\kappa(n)$  its corresponding attribute. Once these residues are computed, it is easy to compute the UAO, in a single upward tree traversal. Appendix A explains this process, as well as the step by step computation of UAO for Fig. 5.

### 3. Ultimate opening drawbacks

The UO presents several advantages as a segmentation method of homogeneous regions, such as non-parametric, intrinsically multi-scale and extraction of the most contrasted structures. However, the

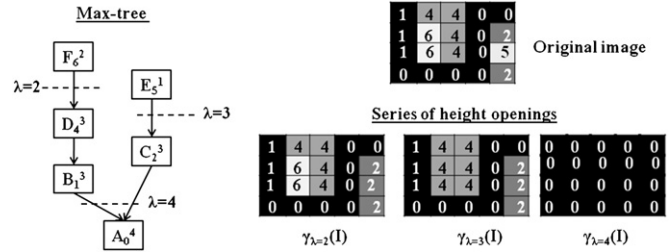


Fig. 5. Max-tree and height openings  $\gamma_T(I)$ , where  $T : \kappa \geq \lambda$ .

last advantage may be its biggest drawback. The UO is an operator without memory and it only saves the last maximum residue. It does not keep intermediate images, which could be also important. Retornaz describes this problem as an operator myopia [28]. UO problems are mainly produced by two configurations: nested structures and gradual transitions.

#### 3.1. Nested structures

This problem appears when the interest structures are nested in other structures with a bigger contrast. Therefore, when a structure is contained in another, it may be masked by a bigger residue. For instance, a pyramidal configuration in the image is a typical case of the nested structures.

For example, in Fig. 6, a masking problem is illustrated by applying the ultimate attribute (height) opening. The synthetic image in Fig. 6(a) has three nested shapes: a rectangle (dimensions  $120 \times 40$ ,  $t=250$ ), a square (dimensions  $30 \times 30$ ,  $t=200$ ) and a circle (diameter 90,  $t=175$ ). These structures are on a plateau of level  $t=130$ . Using Definition 1 with a height opening, step by step, we obtain the following results: the square is the first shape found ( $\lambda=31$ ), and then this shape is masked by a circle filtered by opening  $\lambda=91$ . Before the height opening  $\lambda=161$  is applied, two shapes have been detected; nevertheless, in this opening, an important residue masks relevant information.

#### 3.2. Gradual transitions

Gradual transitions in the image produce small residues when an UO is carried out. For this reason, structures with gradual transitions and also nested in contrasted areas are likely to be masked. An example of this problem is shown in Fig. 7. Fig. 7(a) shows a synthetic image with a rectangle (dimensions  $120 \times 40$ ,  $t=250$ ) on a plateau of level  $t=110$ . The rectangle has a transition in the bottom (dimensions  $2 \times 40$ ,  $t=170$ ). This transition can be observed in Fig. 7(b). By using an ultimate height opening, the rectangle is masked by the plateau, because it has a residue of  $250 - 170 = 80$  smaller than the plateau residue of 110.

In order to solve UO problems, we propose to incorporate shape information, exploiting a prior knowledge of the image and preserving specific shapes before being masked or/and merged.

#### 3.3. Ultimate opening drawbacks on real images

Fig. 8 provides various examples of UO problems on real images. In Fig. 8(a), a lot of internal facade structures are masked. It is a problem of nested structures because the contrast between the wall and the windows is smaller than the contrast between the sky and the building facade. An example of gradual transitions is presented in Fig. 8(b). As it can be seen, all letters are masked because they have blurred boundaries, caused by the digitization process. Fig. 8(c) shows a gradual transition problem on cell images. Two cells are visibly separable but the operator merges both. An extended analysis of those problems on real images will be presented in Section 5.

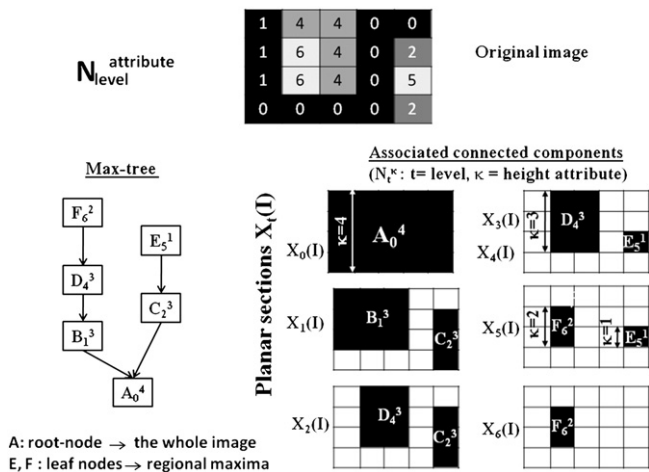


Fig. 4. Max-tree representation.

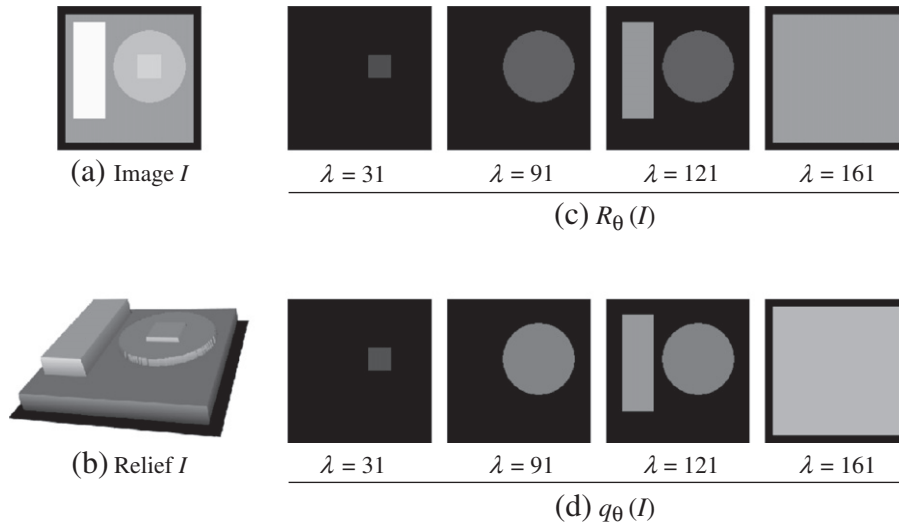


Fig. 6. Example: Masking problem produced by nested structures (a) Synthetic Image, (b) relief  $I$  representation, (c) and (d) intermediate images of  $R_\theta(I)$  and  $q_\theta(I)$ , respectively.

#### 4. Shape ultimate attribute opening

Shape ultimate attribute opening (SUAO) combines shape and gray-scale information in order to extract the most contrasted structures similar to a specific prior shape.

##### 4.1. Shape information

In this section, we define our shape similarity measure in order to combine prior shape knowledge with the  $UO$  operator. The shape definition has been widely studied in the literature. Surveys of shape analysis methods can be found in [31–33].

In our context, we are interested in comparing two shapes ( $\Omega_i, \Omega_j$ ) via a similarity function  $\psi(\cdot)$ .

Most of the prior shape knowledge on segmentation approaches work on contours and a deformable model framework. In contrast, we propose to define a similarity function via shape attributes  $\kappa_\Omega$  of regions, in order to favor specific shapes. In our method, the considered regions  $\Omega$ , on which shape descriptors are computed, are the CCs associated with the max-tree nodes. We use the simplest shape descriptors: geometric features (height, width, etc.) and their relations (fill ratio, circularity, moments, etc.). Furthermore, more complex shape descriptors, such as

Fourier moments or those based on multiple characteristics of a polygon form, can be used to describe shape information.

**Definition 2.** A similarity measure between two shapes,  $\Omega_i$  and  $\Omega_j$ , based on attribute  $\kappa$ , is defined as a function  $\psi_\kappa : \Omega \times \Omega \rightarrow [0, 1] \in \mathbb{R}$  verifying the following conditions[34]:

- Identity:  $\psi_\kappa(\Omega_i, \Omega_i) = 1$ .
- Uniqueness:  $\psi_\kappa(\Omega_i, \Omega_j) = 1$  implies  $\kappa_{\Omega_i} = \kappa_{\Omega_j}$
- Symmetry:  $\psi_\kappa(\Omega_i, \Omega_j) = \psi_\kappa(\Omega_j, \Omega_i)$

After having established these conditions, an example of similarity function  $\psi_\kappa$  is presented in Eq. 4:

$$\psi_\kappa(\Omega_i, \Omega_j) = \begin{cases} \left(1 - \frac{|\kappa_{\Omega_i} - \kappa_{\Omega_j}|}{\tau}\right)^\varsigma & |\kappa_{\Omega_i} - \kappa_{\Omega_j}| \leq \tau \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where  $\tau$  is the similarity threshold of attribute  $\kappa$ . The function reaches its maximum value when the difference between two shape attributes is zero ( $\kappa_{\Omega_i} - \kappa_{\Omega_j} = 0$ ), the min value when the attribute difference is over a given threshold  $\tau$  and a value in the interval  $[0, 1]$  when  $|\kappa_{\Omega_i} - \kappa_{\Omega_j}| < \tau$ .  $\varsigma$  determines if the dynamic range is wide ( $\varsigma < 1$ ), linear ( $\varsigma = 1$ ) or narrow (selective) ( $\varsigma > 1$ ).

We have defined a similarity function with only one attribute  $\kappa$ . In practice, several measures are used to describe a prior shape in a same application. In that case a (simple) multiplication of similarity functions would be applied:

$$\psi = \prod_{\forall \kappa} \psi_\kappa(\Omega_i, \Omega_j). \quad (5)$$

All these possible shape attributes and similarity functions can be utilized to give an advantage to specific shapes in a segmentation process. Nevertheless, we must be careful with the selection of measures, because of the following reasons:

- *Computing time:* Measures are computed for each tree node. To keep a reasonable computing time, we have used the simplest shape attributes, because they can straightforwardly be estimated during the max-tree construction.
- *Invariance with respect to position, scale, and rotation:* The invariance of the proposed method depends on the chosen attributes. For example, the  $UO$  with a fill ratio attribute is not

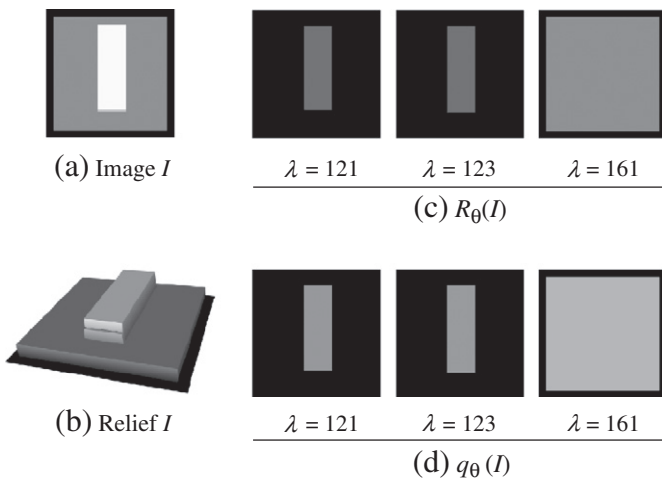


Fig. 7. Example: Masking problem produced by gradual transitions (a) synthetic image, (b) relief  $I$  representation, (c) and (d) intermediate images of  $R_\theta(I)$  and  $q_\theta(I)$ , respectively.

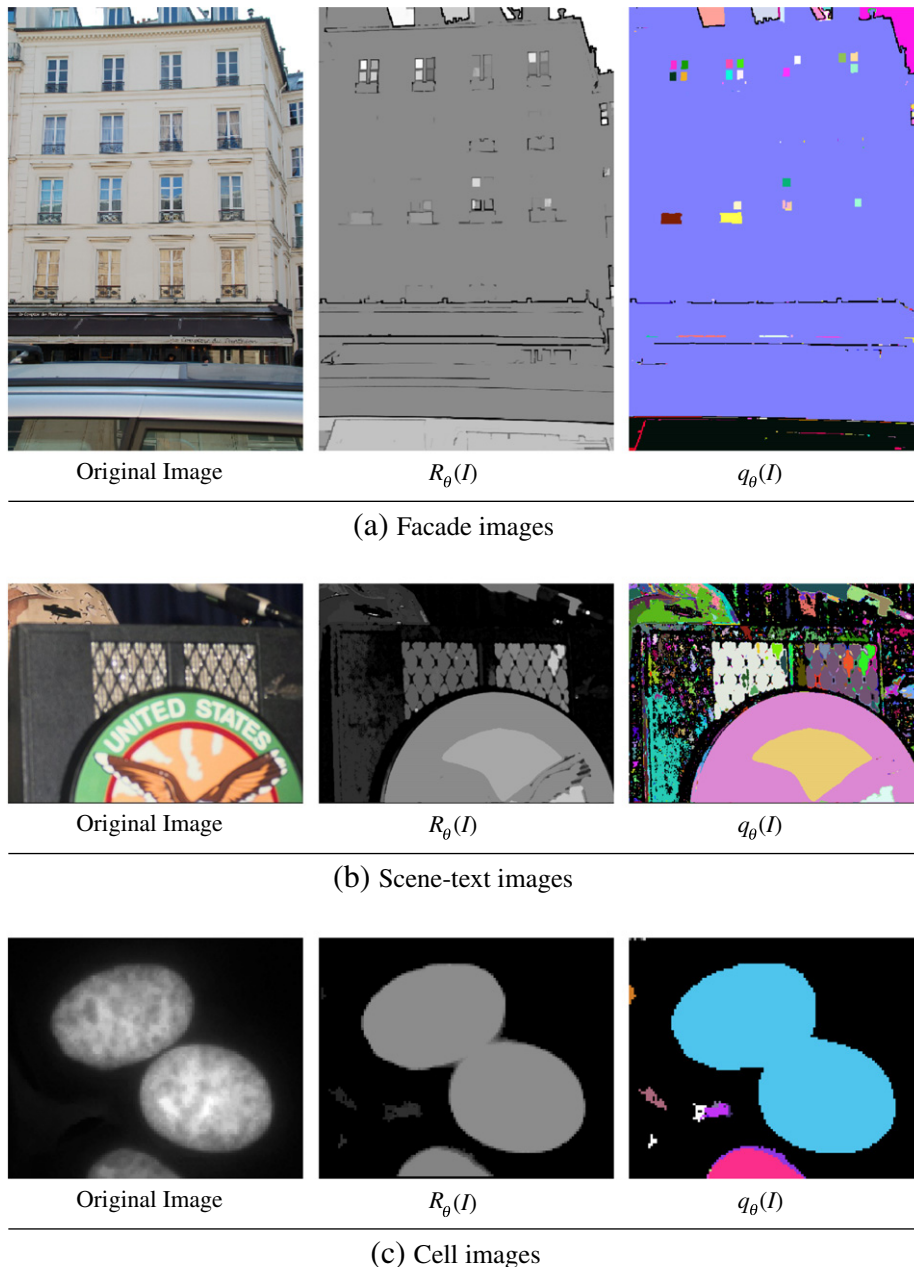


Fig. 8. Problems on real images.

rotation invariant, while if we choose the compactness descriptor, it will be fully invariant.

#### 4.2. Shape ultimate opening

In this paper, we focus on attribute openings, that are connected operators. As explained in Section 2.4, each tree node corresponds to a CC of the image (see Fig. 4): the closer the node is to the root, the larger the CC is (according to the inclusion principle of the tree). We propose to rely on the CC associated to the nodes, named  $\Omega$  in the following, in order to estimate the shape of the segmented regions.

We propose to consider a shape factor function  $f(\Omega, \Omega_{ref})$  to a reference shape  $\Omega_{ref}$  within the residue computation (Eq. 6). As  $\Omega_{ref}$  is fixed,  $f(\Omega, \Omega_{ref})$  is denoted by  $f(\Omega)$ .

$$r_\lambda^\Omega \leftarrow f(\Omega)r_\lambda \quad (6)$$

In that way, the residue of a  $\Omega$  similar to  $\Omega_{ref}$  is artificially increased, favoring its presence in the segmentation result, even if it is nested in a more contrasted shape different from the prior.

The shape factor function  $f(\Omega)$  is related to the similarity function  $\psi_{\mathcal{V}\kappa}$  of  $\Omega$  as follows:  $1 + \alpha/\psi_{\mathcal{V}\kappa}$ . An offset of 1 is added in order to switch to the standard  $UO$  when the similarity function is equal to 0 ( $r_\lambda^\Omega$  becomes  $r_\lambda$ ). As well, a multiplicative factor  $\alpha$  is used to control the influence of the shape factor with respect to the gray level. Hence,  $1 + \alpha$  represents the maximum value that the function may reach. Finally, function  $f_\Omega$  is stored on an image  $F_\theta^\Omega(I)$  when the maximal residue ( $R_\theta(I)$ ) is generated. With this information, we modify the original expression of the  $UO$  definition 1 by Definition 3.

**Definition 3.** We define a shape ultimate opening  $\theta^\Omega$  of an image  $I$  by:

$$\theta^\Omega(I) : I \xrightarrow{\theta^\Omega} \left( R_\theta^\Omega(I), q_\theta^\Omega(I), F_\theta^\Omega(I) \right) \quad (7)$$

$$R_{\theta}^{\Omega}(I) = \max_{\lambda} (r_{\lambda}^{\Omega}(I)) = \max_{\lambda} (f(\Omega) \times r_{\lambda}(I)) \quad (8)$$

$$q_{\theta}^{\Omega}(I) = \begin{cases} \max\{\lambda + 1 | r_{\lambda}^{\Omega}(I) = R_{\theta}^{\Omega}(I)\} & R_{\theta}^{\Omega}(I) > 0 \\ 0 & R_{\theta}^{\Omega}(I) = 0 \end{cases} \quad (9)$$

$$F_{\theta}^{\Omega}(I) = \begin{cases} f(\Omega) | r_{\lambda}^{\Omega}(I) = R_{\theta}^{\Omega}(I) & R_{\theta}^{\Omega}(I) > 0 \\ 0 & R_{\theta}^{\Omega}(I) = 0 \end{cases} \quad (10)$$

where,  $\Omega$  is a CC associated with a tree node (see Fig. 4) and  $f(\Omega)$  is a shape factor function.  $F_{\theta}^{\Omega}(I)$  gives the similarity of segmented regions with the reference shape.

As it can be seen, the shape factor is directly applied on the residue computation, artificially increasing the residue of regions similar to the prior one.

### 4.3. Example on synthetic image

We test our approach on a synthetic image (Fig. 6(a)). First, we try to favor rectangular shapes. For this purpose, we use as a shape metric the fill ratio  $\Upsilon_{\Omega} = \frac{A_{\Omega}}{Abbox_{\Omega}}$ , where ( $A_{\Omega}$ ) is the shape area and ( $Abbox_{\Omega}$ ) is the bounding box area. The ratio lies in the range [0,1]; where, if the value is close to 1, it means that the shape corresponds to a rectangular polygon without rotation, i.e.  $\Upsilon_{\Omega_{ref}} = 1$ . Then, we have imposed an area limit (90% of image area  $A_I$ ) to validate the shape in order to reject the largest regions. This factor function is translated into Eq. 11.

$$\begin{aligned} f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \\ \text{where, } \alpha &= \max(I)/2, \max = 255 \\ \psi_{\kappa_1}(\Omega) &= \Upsilon_{\Omega} \\ \psi_{\kappa_2}(\Omega) &= \begin{cases} 1 & A_{\Omega} < 90\%A_I \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (11)$$

Fig. 9 presents the result on a synthetic image. In this case, the masking problem is solved and the three shapes are segmented. The importance of limit  $\psi_{\kappa_2}(\Omega)$  is remarkable in this example, because the masking shape has a high fill ratio  $\Upsilon_{\Omega} \approx 1$ . Using the limit, the shape has a factor equal to 1, thus  $r_{\lambda}^{\Omega} = r_{\lambda}$ .

Even though  $\Upsilon$  mainly favors rectangular structures, the circular shape factor is high enough to unmask it. If we want to be more selective, we modify  $\psi_{\kappa_1}(\Omega)$  by a narrower function. For example, we change  $\Upsilon_{\Omega}$  in Eq. 11 by  $\Upsilon_{\Omega}^3$  (Eq. 12).

$$\begin{aligned} f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \\ \text{where, } \alpha &= \max(I)/2, \max = 255 \\ \psi_{\kappa_1}(\Omega) &= (\Upsilon_{\Omega})^3 \\ \psi_{\kappa_2}(\Omega) &= \begin{cases} 1 & A_{\Omega} < 90\%A_I \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (12)$$

Another shape factor example is implemented to favor circular shapes using as a shape metric the circularity measure  $\frac{4\pi A_{\Omega}}{(L_{\Omega})^2}$ , with  $A$  the area and  $L$  the perimeter of  $\Omega$ . Eq. 13 shows a factor function by

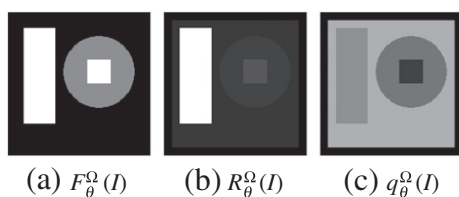


Fig. 9. SUAO of Fig. 6(a)) with Eq. 11 as factor function.

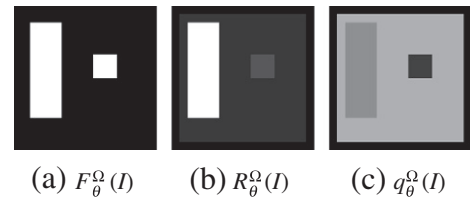


Fig. 10. SUAO of Fig. 6(a) with Eq. 12 as factor function.

using the circularity expression. Figs. 10(c) and 11(c) confirm that rectangles and circles are segmented, respectively.

$$\begin{aligned} f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \\ \text{where, } \alpha &= \max(I)/2 \\ \psi_{\kappa_1}(\Omega) &= \frac{4\pi A_{\Omega}}{(L_{\Omega})^2} \end{aligned} \quad (13)$$

## 5. Applications

In order to demonstrate the performance of our method, we illustrate three segmentation applications: facade image analysis, scene-text detection and cell segmentation. All databases and result tests are available on the following web site: [http://cmm.ensmp.fr/~hernandez/uo\\_en.html](http://cmm.ensmp.fr/~hernandez/uo_en.html).

### 5.1. Facade image analysis

We focus on a segmentation procedure behind the facade modeling [35,36] to detect/extract structural objects, mainly windows. This facade analysis has been developed in the framework of Cap Digital Business Cluster TerraNumerica project (<http://cmm.ensmp.fr/TerraNumerica>) which aims at increasing the productivity and the realism of urban modeling.

Initially, we employed an ultimate attribute opening to segment facade images. For the facade structure detection, a height attribute of the CCs is used. Fig. 12 shows an example of UAO and SUAO using HSL color gradient [37]. In the example, most internal structures are masked in the segmentation process due to the fact that the contrast between the sky and the facade is bigger than the contrast between the wall and the windows (Fig. 12(c)). Most urban images contain sky information and high contrast roof-wall; for those reasons the UAO segmentation is highly affected by the masking problem.

In facade images, windows and doors are at least partially rectangular. Mayer and Reznik [38] propose fill ratio ( $\Upsilon_{\Omega}$ ) and aspect ratio ( $\varkappa_{\Omega} = \frac{\text{height}}{\text{width}}$ ) as features to discriminate them. They state that the aspect ratio of a window generally lies between 0.2 and 5.0. With these features, we define the similarity function:

$$\begin{aligned} f(\Omega) &= 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \\ \psi_{\kappa_1}(\Omega) &= (\Upsilon_{\Omega})^2 \\ \psi_{\kappa_2}(\Omega) &= \begin{cases} (\varkappa_{\Omega})^{0.5} & \varkappa_{\Omega} > 0.2 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (14)$$

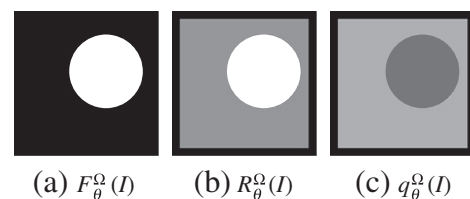


Fig. 11. SUAO of Fig. 6(a) with Eq. 13 as factor function.

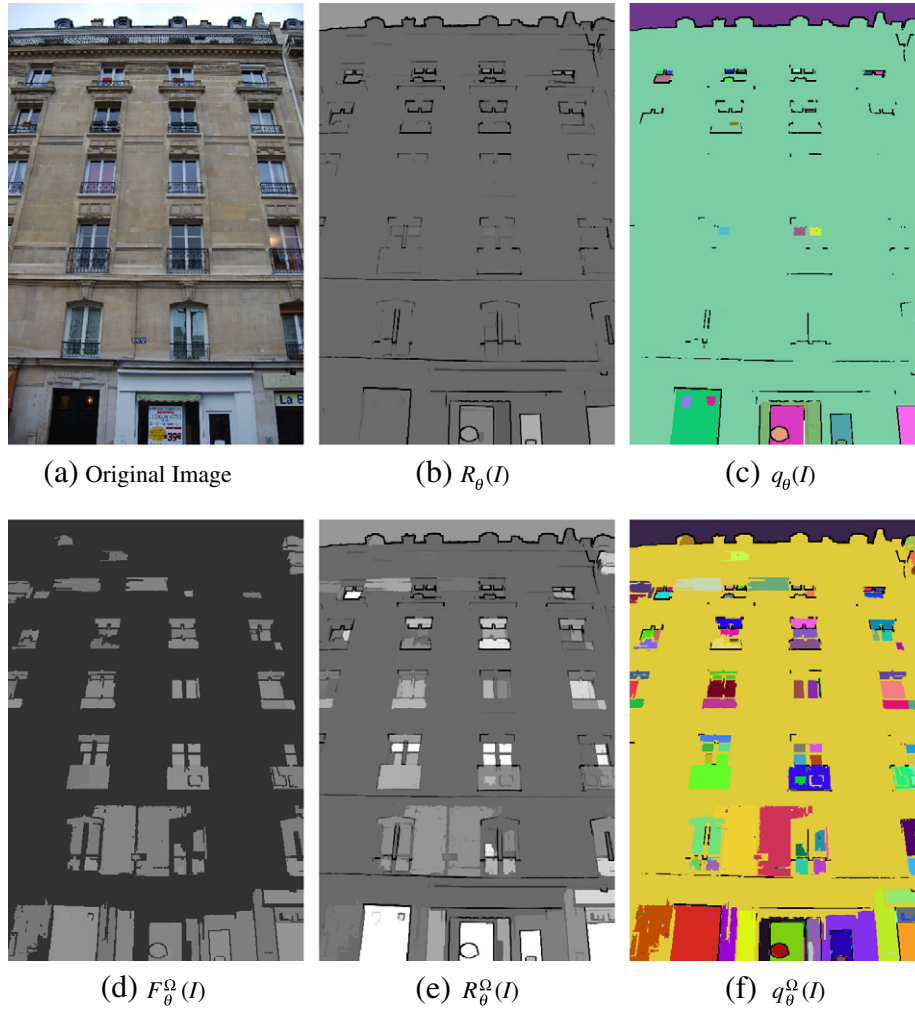


Fig. 12. (a) Original image, (b)–(c)  $R_\theta(I)$  and  $q_\theta(I)$  of UAO, (d)–(f)  $F_\theta^\Omega(I)$ ,  $R_\theta^\Omega(I)$  and  $q_\theta^\Omega(I)$  of SUAO.

The first shape attribute is fill ratio ( $\kappa_1 = \Upsilon_\Omega$ ). We have used a square value  $\varsigma = 2$  as narrow dynamic function to be more selective. The second shape attribute  $\kappa_2$  is an aspect ratio  $\kappa$ . The standard aspect ratio ( $h/w$ ) is too sensitive to small width variations, moreover it does not validate shape similarity conditions (see Section 4.1). Thus, we have computed  $\aleph = \frac{\min(h,w)}{\max(h,w)}$ . We have used a wide dynamic function  $\varsigma = 0.5$  and truncated on  $\kappa > 0.2$ . After several tests with

facade images, the biggest gradient in a facade is about ten times bigger than the one between the wall and the windows. For this reason, we have chosen  $\alpha = 9$ , i.e.  $1 + \alpha = 10$ .

The segmentation result on the facade image is illustrated in Fig. 12(f). Our approach shows a better segmentation because most structures of interest are correctly segmented.

To illustrate the robustness of the shape factor to perspective effects, Fig. 13 shows the shape factor for a window turning around its

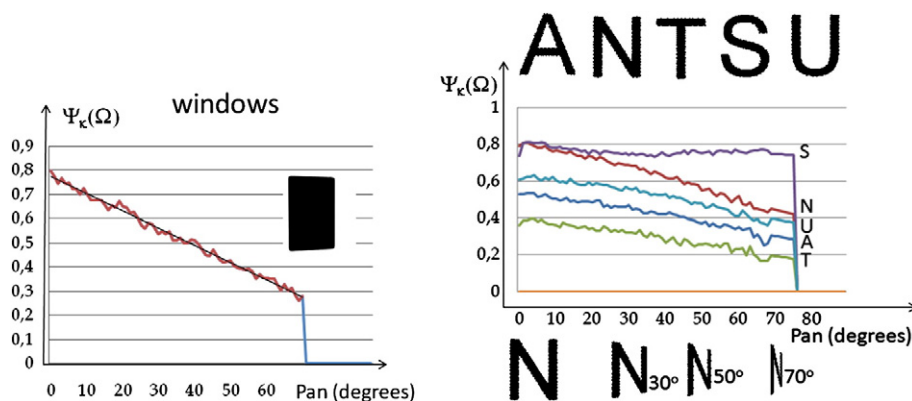


Fig. 13. Shape factor evolution with pan angle (rotation around the Z-axis), for windows (left) and letters (right).

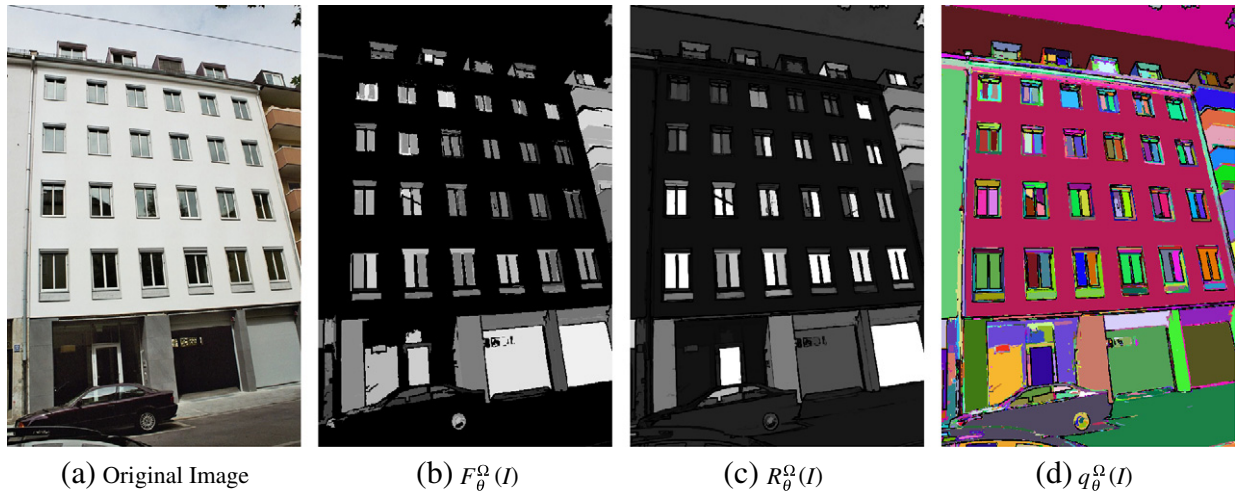


Fig. 14. (a) Original image [39], (b)–(d)  $F_{\theta}^{\Omega}(I)$ ,  $R_{\theta}^{\Omega}(I)$  and  $q_{\theta}^{\Omega}(I)$  of SUAO.

Z-axis in a view frustum (pan angle from  $0^{\circ}$  to  $80^{\circ}$ ). The shape factor decreases when the perspective increases. However, it is still of about 0.5 for  $40^{\circ}$  of panning. Fig. 14 shows an example on a real image, illustrating the good detection of windows even with non fronto-parallel facades.

### 5.2. Scene-text detection

The text in a scene is linked to the semantic context of the image and constitutes a relevant descriptor for content-based image indexing [28]. This research has been developed as part of iTOWNS (Image-based Town On-line Web navigation and Search Engine, <http://www.itowns.fr/>) project. The goal of this project is to develop a new generation of multimedia web tools that combine a 3D geographic browser with an image-based search engine based on a data indexing images and visual context.

In several cases, the text on images is at least partially placed on a surface of different colors, such as placards, posters, etc.; and favors the visibility of letters. But this surface is also contrasted in comparison to its surroundings (Figs. 15(a) and 16(a)). When we utilize the UAO, characters may be masked by the contrast of the signboard with its surroundings. In the same way, we propose some text features to define shape information. Based on histograms of approximately 5000 analyzed characters, letter features are described as follows: 1 – 97% of letters have an aspect ratio bigger than 0.4, 2 – 84% of letters fall approximately between 0.2 and 0.9 fill ratio and 3 –

the biggest height (resp. width) of a character is  $1/3$  of the image height  $H_i$  (resp. width  $W_i$ ). We have used a similarity function analogous to facade application. For  $\kappa$  and  $\Upsilon$  metrics, wide dynamic functions have been used. For  $\Upsilon$  attribute, we have used a centred function on  $\Upsilon_{ref}=0.55$  and a threshold  $\tau=\pm 0.35$  leading to the required limits  $\Upsilon_{\Omega}\in(0.2, 0.9)$  (Eq.15). Fig. 13 shows the evolution of this factor with a pan angle, for different letters. Some letters, such as S or N, have a high shape factor, that decreases slowly with a pan angle. Thus, the residue of these letters is boosted to be present in the result. Other letters, those with smaller fill ratio such as T, have a smaller shape factor, specially with large pan angles. In that case, the combination of contrast and shape factor is required to select significant CCs.

$$f(\Omega) = 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \psi_{\kappa_3}(\Omega)$$

where,  $\alpha = 9$

$$\psi_{\kappa_1}(\Omega) = \begin{cases} (\kappa_{\Omega})^{0.5} & \kappa_{\Omega} > 0.4 \\ 0 & \text{otherwise} \end{cases}$$

$$\psi_{\kappa_2}(\Omega) = \begin{cases} \left(1 - \left(\frac{\Upsilon_{\Omega} - 0.55}{0.35}\right)\right)^{0.5} & 0.2 < \Upsilon_{\Omega} < 0.9 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$\psi_{\kappa_3}(\Omega) = \begin{cases} 1 & h_{\Omega} < \frac{H_i}{3} \wedge w_{\Omega} < \frac{W_i}{3} \\ 0 & \text{otherwise} \end{cases}$$

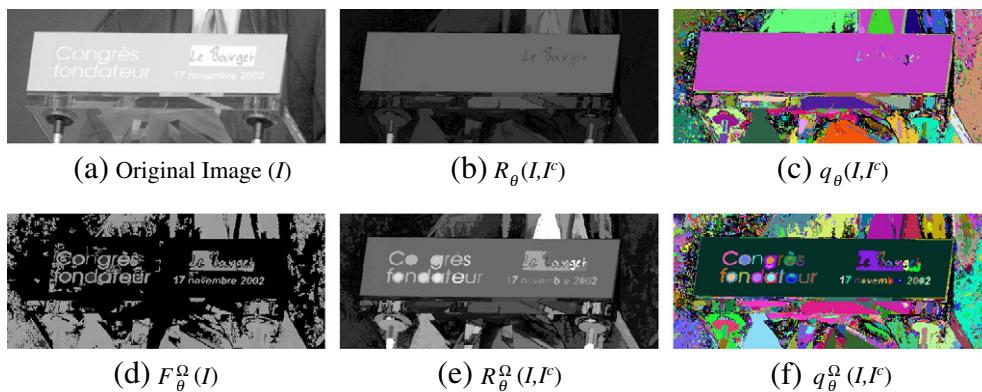


Fig. 15. (a) Original image, (b)–(c)  $R_{\theta}(I, F)$  and  $q_{\theta}(I, F)$  of UAO, (d)–(f)  $F_{\theta}^{\Omega}(I, F)$ ,  $R_{\theta}^{\Omega}(I, F)$  and  $q_{\theta}^{\Omega}(I, F)$  of SUAO.

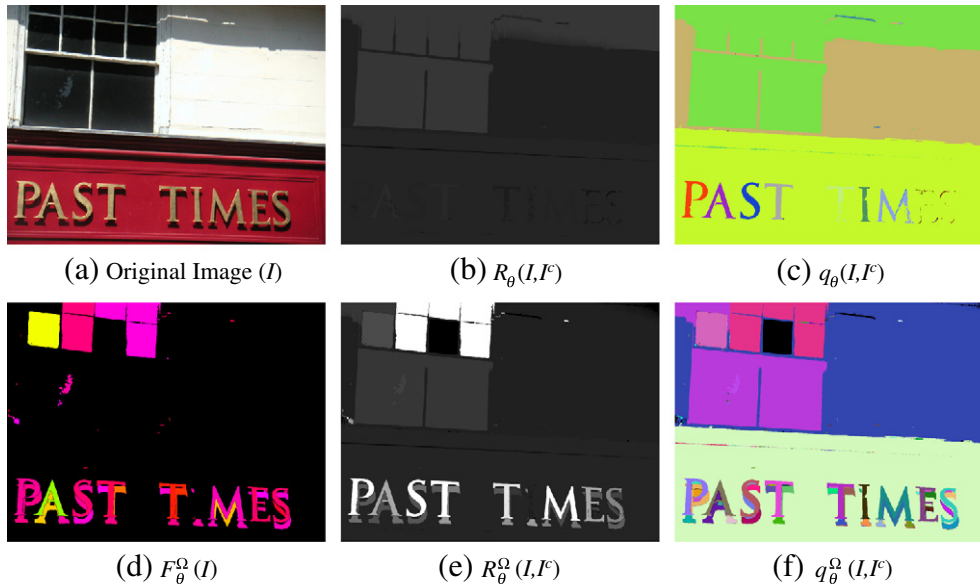


Fig. 16. (a) Original image, (b)–(c)  $R_\theta(I, I^c)$  and  $q_\theta(I, I^c)$  of UAO, (d)–(f)  $F_\theta^\Omega(I)$ ,  $R_\theta^\Omega(I, I^c)$  and  $q_\theta^\Omega(I, I^c)$  of SUAQ.

Besides, scene text may be light or/and dark. In order to deal with both polarities the *UO* is applied to the image and to its inverse. For each pixel, the polarity leading to a bigger residue is kept (see Eq. 16).

$$R_\theta(I, I^c) = \max(R_\theta(I), R_\theta(I^c))$$

$$q_\theta(I, I^c) = \begin{cases} q_\theta(I) & R_\theta(I) > R_\theta(I^c) \\ q_\theta(I^c) & \text{otherwise} \end{cases} \quad (16)$$

Figs. 15 and 16 show the results of the text detection using the *UAO* and the *SUAQ* of a placard image. Using *UAO*, most letters have been masked. With our method, the masking problem is avoided and all letters become visible (Figs. 15(f) and 16(f)). As well, we obtain a homogenous value on the whole placard surface. The results from this study indicate that the proposed method is superior to the classical *UAO* segmentation.

### 5.3. Segmentation of cell images

The recent emergence of high-throughput automated image acquisition technologies has changed forever the way cell biologists collect and analyze data [40]. The interpretation of cellular phenotypes is now a major investigation subject in biology. Thus, the development of automated methods which aim at identifying novel and biological relevant phenotypes is a major challenge. For instance, blocking a gene with a RNA interference (RNAi) may highlight this gene effect over the cell normal functioning. The observation of a toxic molecule may show new action modes and put on the right track to develop new medicines. This study is carried out within the framework of RAMIS (Rock Analysis Module and Interface for Screening) project of Cancer-Bio-Health Cluster of Toulouse-France. An image collection acquired by multi-parametric cell labeling and automated cell image methods is analyzed.

In image processing, the morphological (size, shape, texture, etc.) properties of cells provide valuable information worth exploring by starting with a segmentation step. We propose as a segmentation approach the *UAO* operator. However, the use of this operator presents several drawbacks, such as a masking problem produced by gradual transition associated with blur and cell proximity and undesirable nuclei over-segmentation. By analyzing histograms of approximately 1200 cells, the cell features are described as follows: 1 – circularity is the principal attribute, 2 – area ( $A$ ) falls approximately

between 500 and 3000 pixels and 3 – cells height ( $h$ ) and width ( $w$ ) lie between 15 and 110 pixels. Eq. 17 shows the resulting shape factor function:

$$f(\Omega) = 1 + \alpha \psi_{\kappa_1}(\Omega) \psi_{\kappa_2}(\Omega) \psi_{\kappa_3}(\Omega)$$

where  $\alpha = 10$ ,  $\psi_{\kappa_1}(\Omega) = \frac{4\pi A_\Omega}{(L_\Omega)^2}$

$$\psi_{\kappa_2}(\Omega) = \begin{cases} 1 & 500 \leq A_\Omega \leq 3000 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

$$\psi_{\kappa_3}(\Omega) = \begin{cases} 1 & h_\Omega \text{ and } w_\Omega \in (15, 110) \\ 0 & \text{otherwise} \end{cases}$$

Fig. 17 shows the comparison between the *UAO* and the *SUAQ*. Note that  $q_\theta(I)$  and  $q_\theta^\Omega(I)$  present a lot of noisy CCs. Thus, once the *SUAQ* is performed, a coarse elimination of CCs is carried out. A global threshold ( $t=2$ ) of average  $R_\theta(I)$  on  $q_\theta(I)$  is applied in order to remove low contrasted structures. With the coarse filter most noisy CCs are eliminated. Both segmentations are similar (Fig. 17(d) and (h)). However, the *SUAQ* has unmasked several cells on the right-down region and has separated some linked cells. Our approach cannot avoid a leakage problem when leak between two cells happens before CCs have a high factor. Besides, it reduces nuclei segmentation by favoring the whole cell structure. Nevertheless, when a nuclei is slightly circular, it is favored too. Experimental results indicate that the proposed approach can be used to segment cells for further analysis by reducing the problems of the classical *UAO*.

### 6. Conclusion and future work

A novel segmentation method based on the ultimate attribute opening (*UAO*), combined with shape information, is introduced in this paper. It consists in reinforcing the contrast of regions which are shaped like the prior one. An efficient implementation based on the max-tree representation is proposed. In order to estimate the shape of segmented regions, we rely on the CCs associated to tree nodes. The independence of the residue and the shape descriptor computation allow the use of any shape distance in the proposed framework. Our approach focuses on particle segmentation [41] and it is not suitable for highly textured regions.



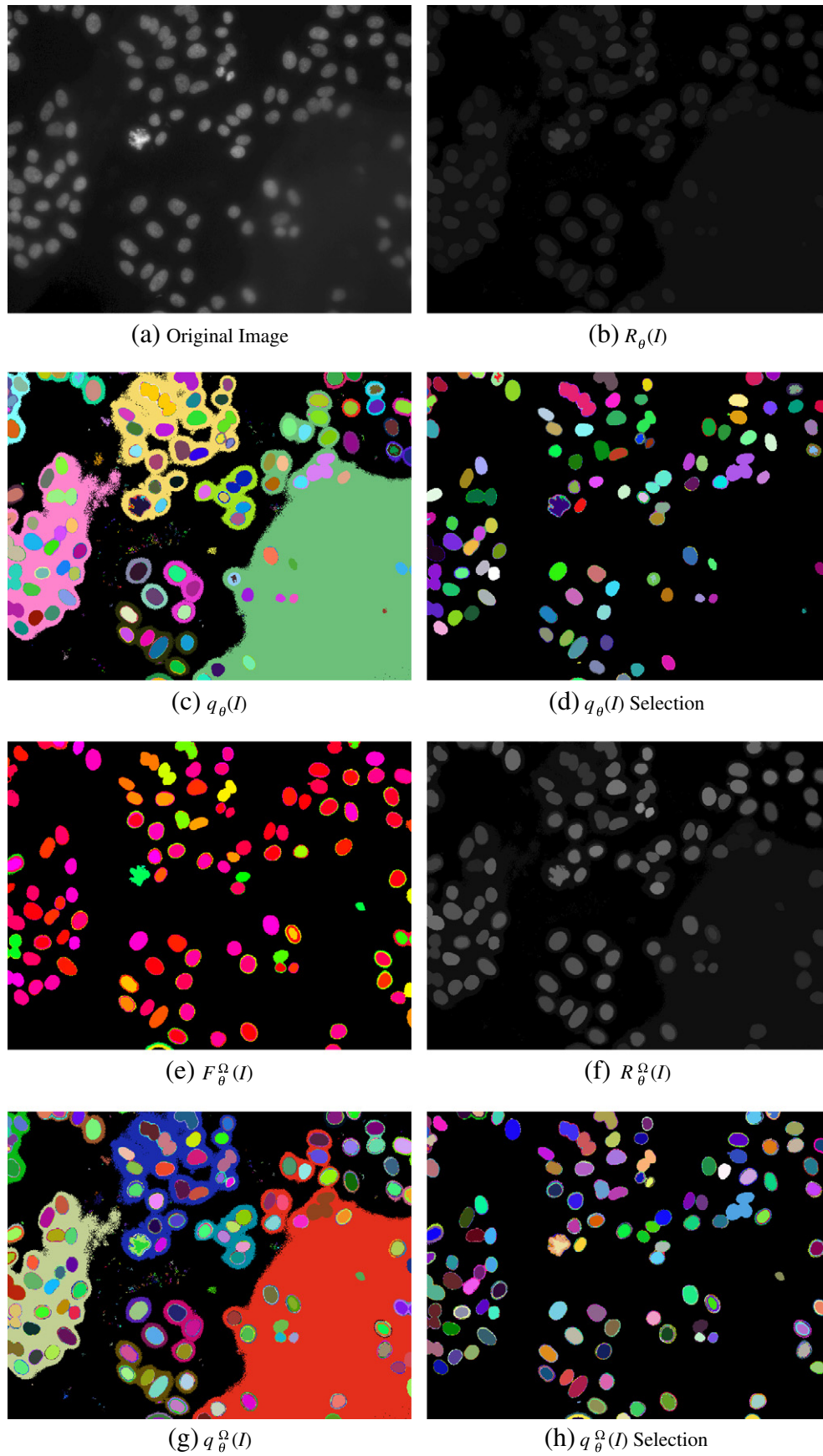


Fig. 17. (a) Original image: sample RNAi fluorescent images, (b)–(d)  $R_\theta(I)$ ,  $q_\theta(I)$  and  $q_\theta(I)$  + coarse elimination of  $UAO$ , (e)–(h)  $F_\theta^\Omega(I)$ ,  $R_\theta^\Omega(I)$ ,  $q_\theta^\Omega(I)$  and  $q_\theta^\Omega(I)$  of  $SUAO$ .

Much better segmentation results, compared to the standard UAO, are reported and many other examples can be seen in our website <sup>1</sup>. The use of very simple shape descriptors, such as fill ratio or aspect ratio, significantly improves the results for a given application, with a marginal increase in the computation time. The approach has been validated in three applications: 1. facade, 2. text and 3. cell segmentation. A benchmark with other state of the art methods would be interesting, but as far as we know there is no annotated database with objects of a given shape. The creation of such a database is a tedious work, out of the scope of this paper.

The UAO provides two pieces of information, contrast ( $R_\theta^O(I)$ ) and size ( $q_\theta^O(I)$ ). The proposed method provides a third interesting piece of information: the shape factor image ( $F_\theta^O(I)$ ), that conveys a shape similarity measure with a reference shape.

In the future, we will analyze in detail the shape factor image. The detection process is the first step in computer vision problems. We intend to apply a machine learning process using shape features (together with color descriptors) to classify regions in the three applications. Furthermore, machine learning techniques could be introduced as a factor function into the presented method. For example, machine learning techniques can be used to estimate the probability of regions to belong to a given class. This probability can be used to weigh the original residues, favoring the segmentation of regions of the given class.

### Acknowledgments

The work reported in this paper has been performed as part of Cap Digital Business Cluster Terra Numerica project, iTOWNSs project and RAMIS project.

### Appendix A. UAO computation on max-tree

Fig. A.18 shows max-tree computation of the UAO. When pruning the tree, all regions corresponding to the nodes of the branch are merged in a single region. This region becomes indivisible for larger openings. Hence, removing a node produces a residue for all pixels of its descendant nodes. For this reason, an upward tree analysis, from the root to the leaves, allows to efficiently propagate the residual information, as it has been described in [30]. Instead of computing for each node the maximum (required for the UAO computation) between its residue and the residue of all its ancestors, an ancestor transmits its residue to its children, and the maximum residue ( $R_\theta$ ) of all nodes is computed in a single tree traversal.

The process starts at the root node:  $R_\theta(\text{root})$  and  $q_\theta(\text{root})$  are initialized to zero, and propagated to root children. Every child computes its own residue, according to Eq. 3 and compares it with  $R_\theta(\text{parent})$ .  $R_\theta(\text{child})$  keeps the maximum value between them.  $q_\theta$  is defined as the size of the opening producing the maximum residue. If  $r(\text{child}) > R_\theta(\text{parent})$ ,  $q_\theta$  keeps the child attribute ( $\kappa(\text{child}) + 1$ ). Otherwise,  $R_\theta$  and  $q_\theta$  are propagated ( $R_\theta(\text{child}) = R_\theta(\text{parent})$  and  $q_\theta(\text{child}) = q_\theta(\text{parent})$ ). Each child becomes parent and repeats the process. Let us see step by step this process on a simple tree, the one of Fig. 4.

- $B_1^3$ ,  $D_4^3$  and  $C_2^3$  are CCs with an attribute  $\kappa=3$ . Then they are modified by  $\gamma_4$  opening ( $\lambda = \kappa + 1 = 4$ ), generating the following residues:  $r_4(C_2^3) = 2 - 0 = 2$ ,  $r_4(B_1^3) = 1 - 0 = 1$  and  $r_4(D_4^3) = 4 - 0 = 4$ . These residues are the first ones computed, so all of them are maximum and  $R_\theta(C_i^3)$  is set to  $r_4(C_i^3)$  and  $q_\theta(C_i^3)$  is set to  $4 = \kappa(C_i^3) + 1$  for the three nodes ( $B_1^3$ ,  $D_4^3$  and  $C_2^3$ ).

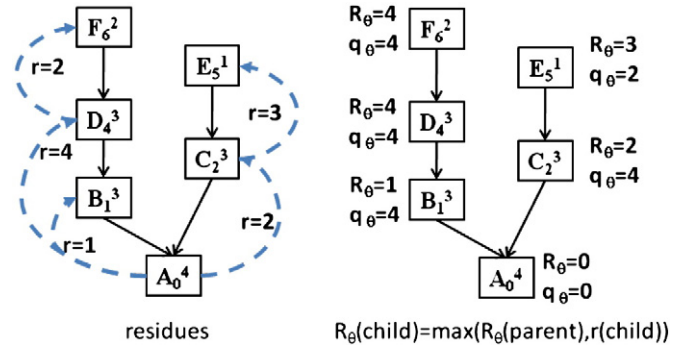


Fig. A.18. UAO computation on max-tree.

- Opening  $\gamma_3$  cuts  $F_6^2 \rightarrow D_4^3$  branch, producing a residue  $r_3(F_6^2) = 6 - 4 = 2$ , which is smaller than the parent residue. Thus, the parent residue is propagated:  $R_\theta(F_6^2) = \max(R_\theta(D_4^3), r_3(F_6^2)) = \max(4, 2) = 4$  and  $q_\theta(F_6^2) = 4$ .
- Opening  $\gamma_2$  cuts  $E_5^1 \rightarrow C_2^3$  branch, producing a residue  $5 - 2 = 3$ . This residue is bigger than the parent residue, updating  $R_\theta(E_5^1) = \max(R_\theta(C_2^3), r_2(E_5^1)) = \max(2, 3) = 3$  and  $q_\theta(E_5^1) = \kappa(E_5^1) + 1 = 2$ .

As mentioned before, a node residue is computed as the difference of its gray level and the gray level of its first ancestor with a different attribute (and not with the gray level of its direct parent). An example of this situation can be seen on the left branch of the tree. The residue of node  $D_4^3$  is not 3, ( $D_4^3 - B_1^3 = 4 - 1 = 3$ ) but 4, ( $D_4^3 - A_0^4$ ). This residue can be computed on the tree in a recursive procedure, following Eq. 3. In this case,  $r_4(D_4^3) = (4 - 1) + (1 - 0) = 3 + 1 = 4$ .

### References

- [1] S. Beucher, Numerical residues, Image Vision Comput. 25 (4) (2007) 405–415.
- [2] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd Ed. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [3] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vision 1 (4) (1988) 321–331.
- [4] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.
- [5] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (5) (1989) 577–685, doi:10.1002/cpa.3160420503.
- [6] T.F. Chan, L.A. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277, doi:10.1109/83.902291.
- [7] J. Hernández, B. Marcotegui, Ultimate attribute opening segmentation with shape information, Proc. of the 9th Intern. Symposium on Mathematical Morphology and Its Application to Signal and Image Processing, ISMM '09, Springer-Verlag, 2009, pp. 205–214.
- [8] F. Meyer, An overview of morphological segmentation, Int. J. Pattern Recognit. Artif. Intell. 15 (7) (2001) 1089–1118.
- [9] P. Salembier, J. Serra, Flat zones filtering, connected operators, and filters by reconstruction, IEEE Trans. Image Process. 4 (1995) 1153–1160.
- [10] L. Vincent, Y.-L. O, A. Toet, D. Foster, H.J.A.M. Heijmans, P. Meer, Morphological area openings and closings for grey-scale images, Proc. of the Workshop “Shape in Picture”, 7–11 September 1992, Driebergen, Springer, Berlin, The Netherlands, 1994, pp. 197–208.
- [11] E.J. Breen, R. Jones, Attribute openings, thinnings, and granulometries, Comput. Vision Image Understanding 64 (3) (1996) 377–389.
- [12] M.H.F. Wilkinson, M.A. Westenberg, Shape preserving filament enhancement filtering, Medical Image Computing and Computer-Assisted Intervention, volume 2208 of Lecture Notes in Computer Science, 2001, pp. 770–777.
- [13] F. Tushabe, M.H.F. Wilkinson, Image preprocessing for compression: attribute filtering, inproceedings World Congress on Engineering & Computer Science 2007, 2007, pp. 999–1005.
- [14] E.R. Urbach, J.B.T.M. Roerdink, M.H.F. Wilkinson, Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 272–285.
- [15] P. Maragos, Pattern spectrum and multiscale shape representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 701–716.
- [16] G. Matheron, Random Sets and Integral Geometry, John Wiley & Sons, New York, 1975.

<sup>1</sup> [http://cmm.ensmp.fr/~hernandez/uo\\_en.html](http://cmm.ensmp.fr/~hernandez/uo_en.html)

- [17] E. Urbach, M. Wilkinson, Shape-only granulometries and gray-scale shape filters, Proc. of the 2nd Proc. of the eight Intern. Symposium on Mathematical Morphology, ISMM '02, 2002, pp. 305–314.
- [18] F.Y. Shih, C.C. Pu, Morphological shape description using geometric spectrum on multidimensional binary images, Pattern Recognit. 25 (9) (1992) 921–927.
- [19] D. Schonfeld, J. Goutsias, Robust morphological representation of binary images, Proc. of the "Intern. Conf. on Acoustics, Speech, and Signal Processing", 3–6 April 1990, Albuquerque, New Mexico, IEEE, New York, NY, U.S.A, 1990, pp. 2065–2068.
- [20] J. Park, J.M. Keller, Snakes on the watershed, IEEE Trans. Pattern Anal. Mach. Intell. 23 (10) (2001) 1201–1205.
- [21] H. Nguyen, M. Worring, R. van den Boomgaard, Watersnakes: energy-driven watershed segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 25 (3) (2003) 330–342.
- [22] G. Hamarneh, X. Li, Watershed segmentation using prior shape and appearance knowledge, Image Vision Comput. 27 (1–2) (2009) 59–68.
- [23] E.R. Urbach, N.J. Boersma, M.H. Wilkinson, Vector-attribute filters, Proc. of the 7th Intern. Symposium on Mathematical Morphology, ISMM '05, 2005, pp. 95–104.
- [24] P. Salembier, A. Oliveras, J.L. Garrido, Anti-extensive connected operators for image and sequence processing, IEEE Trans. Image Process. 7 (1998) 555–570.
- [25] M.H.F. Wilkinson, J.B.T.M. Roerdink, Fast morphological attribute operations using Tarjan's union-find algorithm, Proc. Intern. Symposium on Mathematical Morphology, ISMM '00, Kluwer, 2000, pp. 311–320.
- [26] A. Meijster, M. Wilkinson, A comparison of algorithms for connected set openings and closings, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 484–494.
- [27] S. Oual, Quantification par analyse d'images de la granulométrie des roches fragmentées: amélioration de l'extraction morphologique des surfaces, amélioration de la reconstruction stéréologique, Ph.D. thesis, CMM/GEOSCIENCES - École Mines Paris (June 2006).
- [28] T. Retornaz, Détection de textes enfouis dans des bases d'images généralistes. un descripteur sémantique pour l'indexation., Ph.D. thesis, École des Mines de Paris (2007).
- [29] J. Hernández, B. Marcotegui, Document image binarization using ultimate attribute opening, Tech. rep., CMM-Mines ParisTech, 2009.
- [30] J. Fabrizio, B. Marcotegui, Fast implementation of the ultimate opening, Proc. of the 9th Intern. Symposium on Mathematical Morphology and Its Application to Signal and Image Processing, ISMM '09, Springer-Verlag, 2009, pp. 272–281.
- [31] S. Loncaric, A survey of shape analysis techniques, Pattern Recognit. 31 (1998) 983–1001.
- [32] R.C. Veltkamp, M. Hagedoorn, State-of-the-art in shape matching, Tech. rep., Principles of Visual Information Retrieval, 1999.
- [33] G. Charpiat, O. Faugeras, R. Keriven, Approximations of shape metrics and application to shape warping and empirical shape statistics, Found. Comput. Math. 5 (1) (2005) 1–58.
- [34] R.C. Veltkamp, M. Hagedoorn, Shape similarity measures, properties, and constructions, In Advances in Visual Information Systems, 4th Intern. Conf., VISUAL 2000, Springer, 2000, pp. 467–476.
- [35] J. Benner, A. Geiger, K. Leinemann, Flexible generation of semantic 3d buildings models, First Intern. Workshop on Next Generation 3D City Models, 2005.
- [36] P. Müller, G. Zeng, P. Wonka, L.V. Gool, Image-based procedural modeling of facades, ACM Trans. Graphics 26 (3) (2007) 85–93.
- [37] A. Hanbury, J. Serra, Mathematical morphology in the hls colour space, Proc. of the British Machine Vision Conf. 2001, BMVA, 2001, pp. 451–460.
- [38] H. Mayer, S. Reznik, Building facade interpretation from image sequences, CMRT05, Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms, and Evaluation, Vol. XXXVI, 2005, pp. 55–60.
- [39] F. Korč, W. Förstner, eTRIMS Image Database for interpreting images of man-made scenes, Tech. Rep. TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn URL, [http://www.ipb.uni-bonn.de/projects/etrimis\\_db/](http://www.ipb.uni-bonn.de/projects/etrimis_db/), April 2009.
- [40] Z. Yin, X. Zhou, C. Bakal, F. Li, Y. Sun, N. Perrimon, S.T.C. Wong, Using iterative cluster merging with improved gap statistics to perform online phenotype discovery in the context of high-throughput rna screens, BMC Bioinformatics 9 (2008) 264.
- [41] L. Vincent, E.R. Dougherty, Morphological segmentation for textures and particles, in: E.R. Dougherty (Ed.), Digital Image Processing Methods, Marcel Dekker, New York, 1994, pp. 43–102.

# ADAPTIVE PARAMETER TUNING FOR MORPHOLOGICAL SEGMENTATION OF BUILDING FAÇADE IMAGES

*Andrés Serna, Jorge Hernández and Beatriz Marcotegui*

Mines ParisTech

CMM - Centre de morphologie mathématique

Mathématiques et Systèmes

35 rue St Honoré 77305-Fontainebleau-Cedex, France

## ABSTRACT

In this paper, we describe an automatic method to segment street level building façade images. Our approach assumes that images are rectified, cropped and their elements are aligned in a pseudo-regular structure. It is based on the accumulation of directional color gradients, combined with morphological filters in order to deal with textured façades. We propose an automatic parametrization of three filters included in the process: opening filter of size  $n_{op}$ , alternate sequential filter (ASF) of size  $n$ , and H-minima filter with contrast threshold  $h$ . This automatic selection offers robustness to noise, image resolution changes, shadows and textures. Quantitative and qualitative results are reported on a public annotated database, validating the good performances of our approach.

*Index Terms*— Mathematical morphology, façade segmentation, window detection, urban modeling

## 1. INTRODUCTION

3D urban modeling is a very active research area. An increasing number of geographic applications, such as Google Earth, Microsoft Virtual Earth and Geoportail, are flourishing nowadays. These applications are not limited to 3D navigation, but also are considered as invaluable tools for environmental studies, urban planning, making accessibility diagnosis, etc. Some of these applications do not only require to look realistic, but also have to be faithful to reality.

Initially, virtual scenarios were created by infographic approaches, leading to time-consuming procedures, unsuitable for large-scale urban modeling. Procedural modeling allows to speed up the 3D virtual environment creation [1]. It is based on a set of rules, a grammar, defining a given architectural style. Procedural modeling approaches create realistic models in an efficient way, but a

precise parametrization is required if the model has to be faithful to reality. Automatic analysis of façade images, combined with procedural modeling, allows to increase the productivity while remaining faithful to reality.

Many algorithms focusing on automatic façade analysis have been designed in the recent years. In general, existing methods use rectified and cropped images containing a single building. Usually, individual buildings are manually extracted. In [2], Lee and Nevatia develop a method based on thresholding of directional gradient projections. In [1], Müller et al. find repetitive architectural structures using mutual information to describe a single façade image. These methods are very sensitive to noise and fail if the building contains textured walls or balconies with different wrought iron designs, very common elements in Parisian Hausmannian architecture. In [3], Hernández et al. describe a method that automatically extracts an isolated building from a city block street level image. Besides, they extend Lee and Nevatia method introducing morphological filters in the directional gradient projections. These filters improve the robustness to textured façades. In [4], Teboul et al. learn a shape dictionary using random forest technique and publish an annotated database with 100 building images. In [5], Hammoudi extracts façade structures from 3D point clouds data using Hough transform. Finally, in [6], Pinte et al. combine color information and 3D point cloud data to improve the method robustness.

In this paper we focus on directional gradient projection techniques combined with morphological filters. We study the adaptive parameter tuning of these filters and evaluate the proposed algorithm on the cited public database. The paper is organized as follows. Section 2 describes the directional gradient projection technique, illustrated on a vertical splitting example. Section 3 describes the filter parametrization technique. Section 4 shows the performance of our method on Teboul's annotated database. Finally, Section 5 is devoted to conclude this work.

---

{andres.serna\_morales, jorge.hernandez, beatriz.marcotegui}  
@mines-paristech.fr

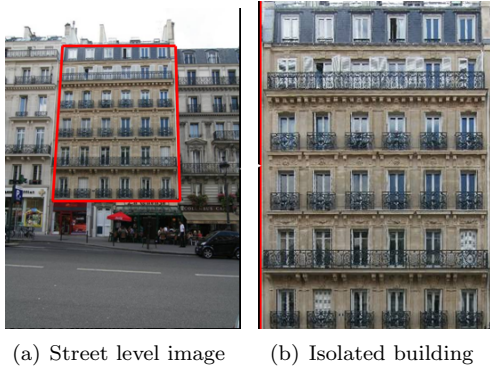


Fig. 1. Automatic extraction of an isolated building [3].

## 2. SEGMENTATION OF BUILDING FAÇADES

The starting point for our approach is the method developed by Hernández et al. [3]. Therefore, it is assumed that input images are rectified and cropped, as shown in Fig. 1. Fig. 2 shows the diagram of the whole process and Fig. 3 illustrates intermediate images. First, a vertical gradient  $G_y(x, y)$  detects horizontal contours (Fig. 3(a)). Then, a horizontal opening filter of size  $n_{op}$  is applied in order to eliminate the undesirable details. Fig. 3(c) shows the accumulation, column by column, of the vertical gradient. This 1D projected gradient contains peaks at window locations and valleys between them.

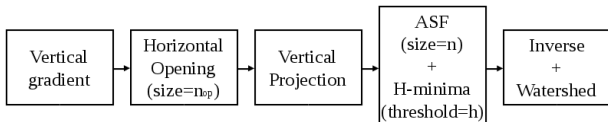
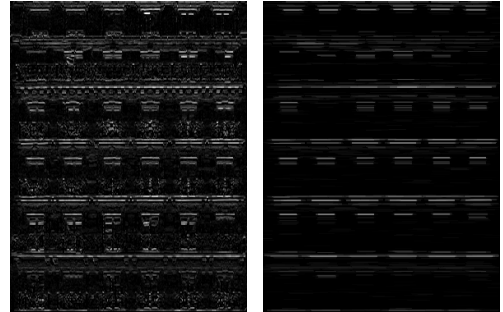


Fig. 2. Process scheme to compute vertical divisions.

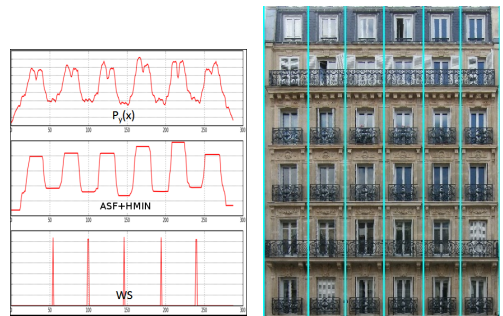
Afterwards, this projection is filtered in order to get a single maximum for each window. An Alternate Sequential Filter (ASF) of size  $n$  and a H-minima filter are used for this purpose. Finally, this profile is inverted and a watershed process computes the façade division. Fig. 3(d) shows the final result superimposed on the original image. Although a vertical splitting is shown, the same technique applies to horizontal splitting, just changing vertical by horizontal and vice versa.

## 3. FILTERING PARAMETRIZATION

The method introduced in the previous section leads to interesting results but relies on a good filter parametrization. Specifically, three parameters require tuning: the size  $n_{op}$  for the horizontal opening, the size  $n$  for the



(a) Vertical gradient:  $G_y(x, y)$  (b) Horizontal Opening



(c) Vertical projection, filtering and watershed (d) Vertical splitting

Fig. 3. Vertical splitting

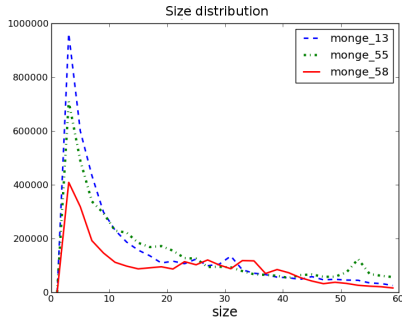
ASF, and the contrast threshold  $h$  for the H-minima filter. If these parameters are too small, the result will be over-segmented (Figures 5(a) and 5(b)). On the other hand, if they are too big, the result will be under-segmented (Fig. 5(d)). The aim of this section is to tune in an adaptive way these filter parameters, according to intrinsic image information. The parameter tuning of each step is explained below.

### 3.1. Opening filter parametrization

Windows are the image largest structures. Small details such as façade ornaments, wrought iron balconies or other noisy structures can produce fake divisions on projected profile  $P_y(G_y)$ . A morphological opening with a horizontal structuring element of size  $n_{op}$  is used in order to get rid of these details from gradient images. The selection of  $n_{op}$  is based on the pattern spectrum. Pattern spectrum plots the quantity of information filtered out by each opening  $\gamma_i$ : ( $PS_i = \sum_{\forall \text{pixel}} (\gamma_{i-1} - \gamma_i)$ ). The resulting curve is also called *size distribution* because its peaks correspond to the prevailing sizes of the image structures [7].

Fig. 4 shows size distributions for different  $G_y(x, y)$  images. These curves present an important peak for small size openings. This peak corresponds to noisy de-

tails. Note that this peak exists for the three images in spite of shadows (Fig. 11(b)), balconies (Fig. 10(b)), and vegetation (Fig. 11(a)). The opening size is chosen as the value  $i$  for which the pattern spectrum falls down under 25% of its maximum. This pattern spectrum analysis offers robustness to image resolution changes.



**Fig. 4.** Size distribution of  $G_y(x, y)$  with a horizontal structuring element. Test images correspond to Figures 10 and 11

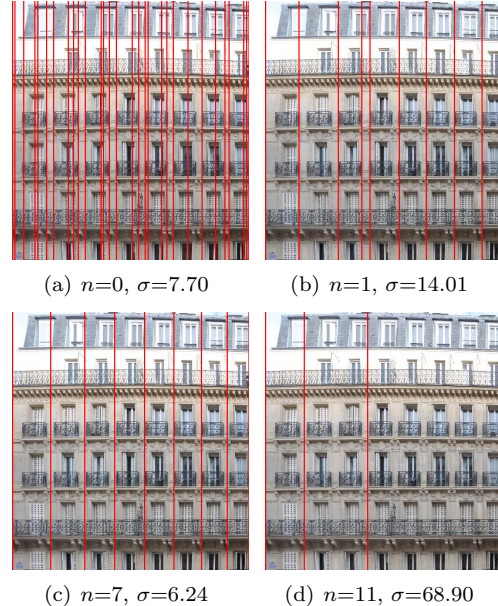
### 3.2. ASF filter parametrization

An ASF consists in a sequence of openings ( $\gamma$ ) and closings ( $\varphi$ ) of increasing sizes. The sequence starts with the filter of size 1 and ends with the filter of size  $n$ :  $ASF_n(P_y(G_y)) = \gamma_n \varphi_n \dots \gamma_2 \varphi_2 \gamma_1 \varphi_1 (P_y(G_y))$ . This filter is particularly appropriated when the noise is present over a wide range of scales [8]. The filter size is chosen based on the façade regularity. Several filters of different sizes are applied, and the one leading to the most regular result is chosen. The regularity is estimated by the standard deviation  $\sigma$  of the segmented façade division sizes. This filter applies to 1D profiles. Then, evaluating different sizes is not a time-consuming task. A frequency domain analysis of this profile would also be possible, but our approach is more robust to pseudo-periodic structures.

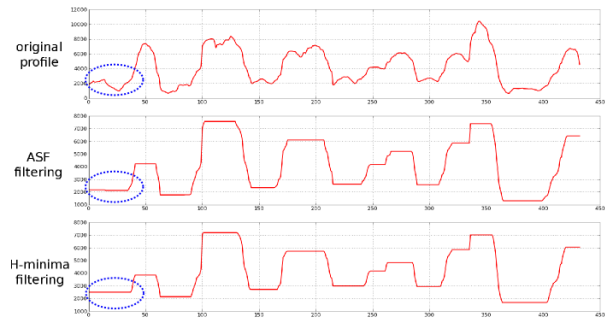
Fig. 5 shows the resulting vertical divisions for different filter sizes. Note that the filter size that minimizes the standard deviation,  $n=7$ , leads to a correct façade division.

### 3.3. H-minima filter parametrization

H-minima filter is a filtering tool based on a contrast criterion. More precisely, this transformation suppresses all minima whose contrast is lower than a given threshold  $h$  [9]. The contrast threshold  $h$  is chosen as a percentage of the dynamic of the extrema in the profile, that is  $h \propto \max(f) - \min(f)$ , where  $f = ASF_n(P_y(G_y))$ . This adaptive selection provides independence with respect to



**Fig. 5.** Façade divisions for different ASF sizes



**Fig. 6.** ASF and H-minima filtering. The original profile corresponds to Fig. 10(a)

image resolution. Fig. 6 illustrates the effect of ASF and H-minima filtering. Note that the strongest filtering is carried out by the ASF, while the H-minima removes still remaining possible low contrasted extrema, as shown in the left side of Fig. 6.

### 3.4. Window detection

We assume that there is only one column of windows per vertical division. Analyzing the extrema of the filtered profile  $\tilde{P}_y(G_y)$ , we found that minima pass through the wall while maxima pass through the windows. Using this information, we apply a constrained watershed on the projected horizontal gradient  $P_y(G_x)$ , taking the extrema of  $\tilde{P}_y(G_y)$  as markers. Fig. 7 illustrates the process of window detection. Although we describe only the

vertical case, the same approach applies to the horizontal detection.

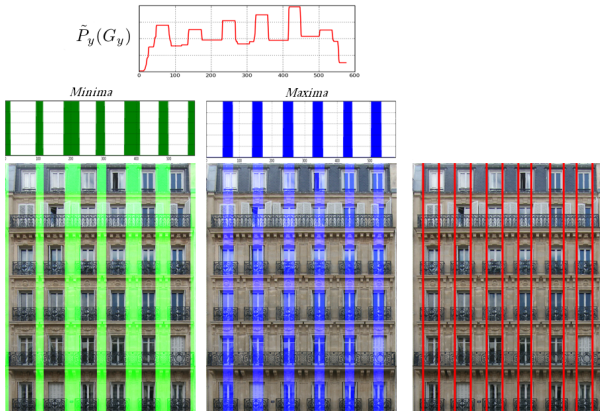


Fig. 7. Location of the vertical edge of the windows

#### 4. EXPERIMENTS

Our method is tested on the public database [4] that contains 100 annotated images. Images are rectified and various semantic elements are manually annotated. An example of this database is shown in Fig. 9. We evaluate our system on window localization with the classic *precision – recall –  $f_{mean}$*  criteria. Note that our procedure detects windows including their corresponding balconies. In order to evaluate correctly the window detection performance, we remove from our detection the ground truth balconies regions

Fig. 8(a) shows the evaluation scores obtained with increasing ASF sizes. We can observe that the maximum  $f_{mean}=0.79$  corresponds to filters of size between 7 and 10. If we use the ASF fitting method proposed in Section 3.2, we get the same score, the maximum in the figure, which proves the efficiency of the proposed tuning.

Once the parameters  $n_{op}$  and  $n$  are chosen according to the procedure aforementioned, we need to choose the best  $h$  threshold for the H-minima filter. Fig. 8(b) shows an exhaustive test varying  $h$  from 1% to 30% of the dynamic in the profiles. The best values found in the test correspond to  $h_v=14\%$  and  $h_h=2.5\%$  of the dynamic for the vertical and horizontal filter thresholds, respectively. Note that this parameter is not so critical since the lowest and highest  $f_{mean}$  correspond to 0.78 and 0.80, respectively. However, it improves the global performance up to 1% with respect to Fig. 8(a), where H-minima filter is not applied.

0

The results reported by Teboul et al. are  $P = 0.58$ ,  $R = 0.81$  and  $f_{mean} = 0.68$  [4]. However, they only

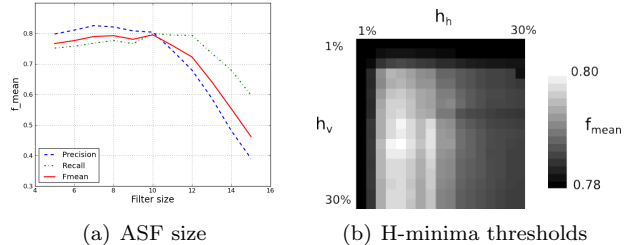


Fig. 8.  $f_{mean}$  sensitivity to parameters  $n$ ,  $h_h$  and  $h_v$

test 10 images of the database, while we have run our experiments on the whole dataset.

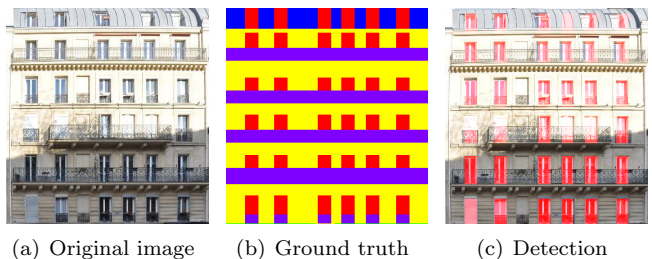


Fig. 9. Example of annotated image: (a) original image, (b) ground truth, and (c) our detection result

Using our proposed adaptive parameter tuning, the results are  $P = 0.82$ ,  $R = 0.79$  and  $f_{mean} = 0.80$ , which is much better than other results reported in the literature on the public Teboul’s database. Qualitative results are shown in Figures 10 and 11. Fig. 10 shows examples in which the proposed method fails. Those images do not respect the regularity hypothesis on which our system is based: some window columns are almost adjacent. The standard deviation of division width is smaller when those columns are merged than when they are separated. Fig. 11 illustrates the robustness of our system to shadows, textures and images on which the distance between windows are pseudo-regular.

#### 5. CONCLUSIONS

We propose an automatic parameter tuning of the three filters in the process: i) size  $n_{op}$  of the opening filter is deduced from the pattern spectrum analysis of gradient images. This filter removes texture details on the façade in order to avoid fake divisions. Moreover, its adaptive tuning offers robustness to image resolution changes. ii) Size  $n$  of the ASF is chosen as the value that minimizes the standard deviation  $\sigma$  of the segmented region sizes. This filter size has a strong influence on the result, as shown in Fig. 5. Its adaptive tuning according

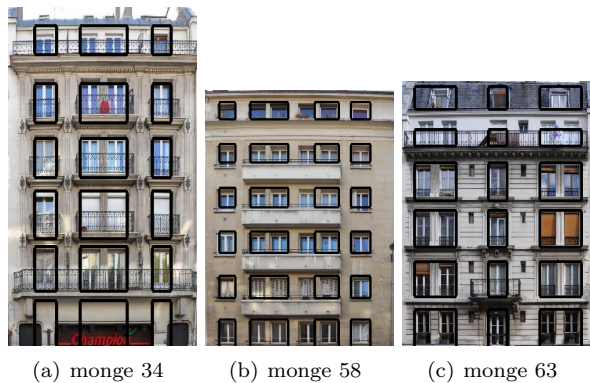


Fig. 10. Examples of problematic results.



Fig. 11. Examples of correct results.

to the standard deviation of division sizes leads to the best result among all filter sizes. The adaptive result is just as good as the best result obtained with any filter size. This is because the whole dataset has the same resolution. If the database resolution were heterogeneous, the results would have been even better than the score with any filter parameter, because our parameter tuning would have been adapted to each image size. And, iii) contrast threshold  $h$  in the H-minima filter is chosen as a percentage of the dynamic of the extrema in the profile. This filter improves  $f_{mean}$  criterion of about 1% (from 79% to 80%). Its sensitivity in its whole dynamic range is very low. That means that the spurious maxima remaining after the ASF are very low contrasted, as shown in Fig. 6. In the horizontal direction, a very wide range leads to almost the same performances. On the other hand, in the vertical direction, filter thresholds between 5% and 15% of the dynamic score best.

The adaptive parameter tuning offers robustness to noise, image resolution changes, shadows and textures. These adaptive filters lead to the best performance score compared to any filter parameters tested in an exhaustive way. Thus, our approach is validated. Qualitative and quantitative results are reported. Our performances are much better than others reported in the literature on

Teboul’s public database.

In the future, the use of an adaptive opening operator, called ultimate opening, will be studied. This operator automatically adapts its size to the image structures, based on a contrast criterion.

## 6. ACKNOWLEDGEMENTS

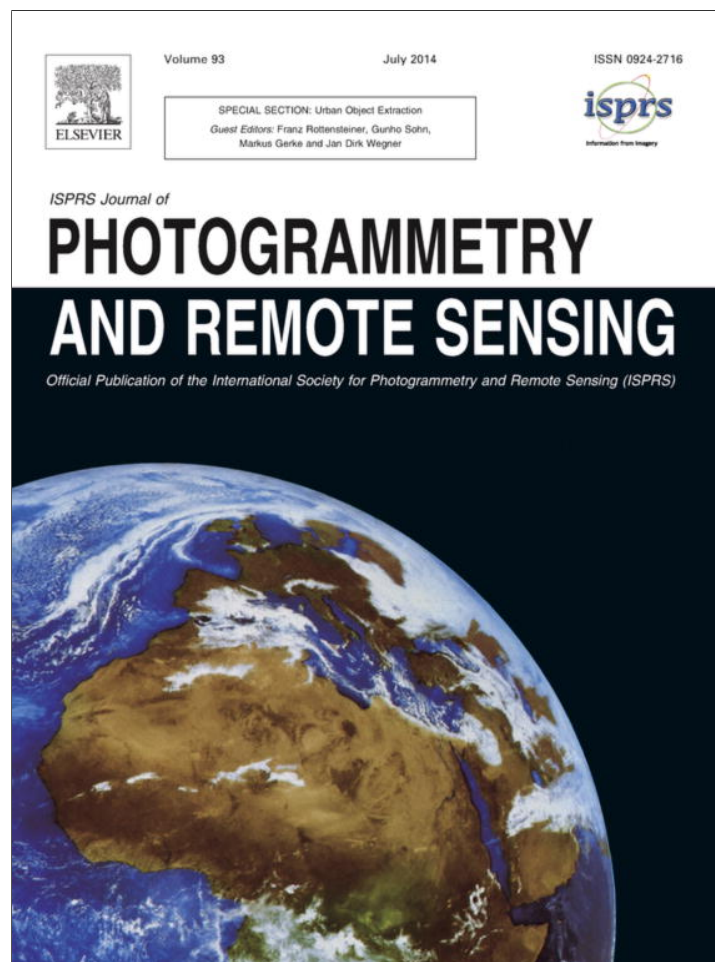
The work reported in this paper has been performed as part of Cap Digital Business Cluster TerraNumerica project.

## 7. REFERENCES

- [1] Pascal Müller, Gang Zeng, Peter Wonka, and Luc Van Gool, “Image-based procedural modeling of facades,” *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 85–93, 2007.
- [2] Sung Chun Lee and Ram Nevatia, “Extraction and integration of window in a 3d building model from ground view images,” *IEEE Computer Vision and Pattern Recognition*, vol. 02, pp. 113–120, 2004.
- [3] Jorge Hernández and Beatriz Marcotegui, “Morphological segmentation of building façade images,” in *IEEE International Conference on Image Processing, ICIP’09*, 2009, pp. 4029–4032.
- [4] Olivier Teboul, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios, “Segmentation of building facades using procedural shape priors,” in *CVPR*, 2010, pp. 3105–3112.
- [5] Karim Hammoudi, *Contributions to the 3D city modeling*, Ph.D. thesis, Université Paris-Est, 2011.
- [6] Antoine Pinte, Caroline Baillard, and Eloïse Denis, “Modélisation de façades par analyse conjointe d’images terrestres et de données laser,” *Revue française de photogrammétrie et de télédétection*, vol. 194, pp. 53–65, 2011.
- [7] Georges Matheron, *Random Sets and Integral Geometry*, John Wiley & Sons, New York, 1975.
- [8] Jean Serra, *Image Analysis and Mathematical Morphology*, vol. 2, Academic Press, London, 1988.
- [9] Pierre Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.



Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

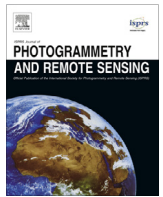
Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/authorsrights>

Contents lists available at [ScienceDirect](http://www.sciencedirect.com)

## ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: [www.elsevier.com/locate/isprsjprs](http://www.elsevier.com/locate/isprsjprs)

# Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning



Andrés Serna\*, Beatriz Marcotegui

MINES ParisTech, CMM – Centre de Morphologie Mathématique, 35 rue St. Honoré, 77305 Fontainebleau-CEDEX, France

## ARTICLE INFO

## Article history:

Received 31 May 2013

Received in revised form 18 March 2014

Accepted 27 March 2014

Available online 24 April 2014

## Keywords:

3D urban analysis

Laser scanning

Detection

Segmentation

Classification

Mathematical morphology

Support vector machine (SVM)

## ABSTRACT

We propose an automatic and robust approach to detect, segment and classify urban objects from 3D point clouds. Processing is carried out using elevation images and the result is reprojected onto the 3D point cloud. First, the ground is segmented and objects are detected as discontinuities on the ground. Then, connected objects are segmented using a watershed approach. Finally, objects are classified using SVM with geometrical and contextual features. Our methodology is evaluated on databases from Ohio (USA) and Paris (France). In the former, our method detects 98% of the objects, 78% of them are correctly segmented and 82% of the well-segmented objects are correctly classified. In the latter, our method leads to an improvement of about 15% on the classification step with respect to previous works. Quantitative results prove that our method not only provides a good performance but is also faster than other works reported in the literature.

© 2014 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.

## 1. Introduction

Thanks to new 3D data availability, an increasing number of geographic applications such as Google Earth, Geoportail, iTowns and Elyx-3D is flourishing nowadays. Most of them have been recently enhanced with pedestrian navigation options and realistic 3D models. In general, 3D city models are useful for many applications: urban planning, emergency response simulation, cultural heritage documentation, virtual tourism, itinerary planning, accessibility analysis for different types of mobility, among others. Some of these applications not only require to look realistic but have also to be faithful to reality. Thus, semantic analysis from real data (images and 3D point clouds) are required in order to give faithfulness to 3D city models. These analyses are usually carried out by manual assisted approaches, leading to time consuming procedures, unsuitable for large scale applications. In that sense, automatic methods for urban semantic analysis are required.

Our work is part of TerraMobilita project<sup>1</sup>—“3D mapping of roads and urban public space, accessibility and soft-mobility”. The project is built around two main topics: (i) to develop new methods and tools to create and update 3D urban maps using laser scanning

and digital imagery; (ii) to develop innovative applications for soft-mobility itinerary planning. The focus of this work is automatic detection, segmentation and classification of urban objects from laser scanning data. Our method is based on elevation images, mathematical morphology and supervised learning. It is validated on three databases in order to get comparative results with the state of the art: two Mobile Laser Scanning (MLS) datasets from Paris (France) and an Aerial/Terrestrial Laser Scanning (ALS/TLS) dataset from Ohio (USA).

This paper is organized as follows. Section 2 reviews related work in the state of the art. Section 3 describes our method to process 3D point clouds using elevation images. Section 4 presents experiments and comparative results with the state of the art. Finally, Section 5 concludes the work.

## 2. Related work

Even though 3D acquisition systems have a high maturity level, 3D automatic analysis of urban areas is still an active research area. In the last years, several automatic solutions have been developed with different aims.

Table 1 summarizes representative papers related to our work. The detection–segmentation method, the classification strategy, the data structure and the accuracy reported on each paper is summed up in the table. Performance ranges from 58% to 95% but results are not comparable because they use different

\* Corresponding author. Tel.: +33 777491896.

E-mail addresses: [andres.erna\\_morales@mines-paristech.fr](mailto:andres.erna_morales@mines-paristech.fr) (A. Serna), [beatriz.marcotegui@mines-paristech.fr](mailto:beatriz.marcotegui@mines-paristech.fr) (B. Marcotegui).

<sup>1</sup> <http://cmm.ensmp.fr/TerraMobilita/>.

databases and different object classes, have different aims, use different data structures and process data in different ways. This table only offers an idea on each method performance. As a general observation, several authors use elevation images, clustering methods for detection–segmentation, and supervised techniques for classification. Further details are given below.

Several methods project 3D information onto a 2D grid in order to reduce the problem complexity and to speed up the computational processing. As each pixel of the projected grid contains elevation information, it is called elevation image or digital elevation model. This kind of 2.5D images has a long tradition in the scientific community (Hoover et al., 1996) and it is of great interest nowadays due to technological developments in remote sensing equipments such as Riegl, Velodyne and Kinect sensors. Gorte (2007) presents a method to segment planes on TLS data using range images. The 3D point cloud is projected from the sensor point of view. As a result, a ‘panoramic’ range image is obtained and plane estimations are done for each pixel on the image. Then, a region growing approach is performed in order to segment pixels belonging to the same plane. In a similar way, Zhu et al. (2010) project MLS data to a ‘panoramic’ range image in which rows represent the acquisition time of each laser scan-line, columns represent the sequential order of measurement and pixel values code the distance from the sensor to the point. They propose a segmentation–classification pipeline using graphs, SVM and decision trees. Hernández and Marcotegui (2009b) propose a method projecting MLS data to elevation images, i.e. a nadir view of the scene. Ground and objects are segmented using morphological transformations and objects are classified in four categories (cars, lamp-posts, pedestrians, and others) using SVM.

Since processing based on elevation images is both precise and fast, real-time applications such as guiding autonomous vehicles have been addressed. Kammel et al. (2008) and Ferguson et al. (2008) have developed autonomous vehicles, for the DARPA Challenge 2007, able to drive through urban scenarios. They use off-line processed aerial images and 2D maps in order to determine road structure. Then, on-board laser scanners are used to build elevation images in order to detect static and mobile obstacles. Munoz et al. (2009), extending the work by Anguelov et al. (2005), propose High Order Markov Random Fields for on-board contextual classification. In general, approaches for autonomous vehicles do not require high (centimetre) accuracy but high speed in order to detect and predict obstacles in real time. More accurate but slower

methods process the 3D point cloud directly. These approaches are suitable for applications with high accuracy requirements but no strict time constraints. One of the major problems is the 3D neighborhood definition, which is not as trivial as it is in the 2D case using elevation images. Demantke et al. (2010) propose a method to adapt 3D neighborhood radius based on local features. Radius selection is carried out optimizing local entropy. Then, dimensionality features are calculated on spherical neighborhoods in order to characterize lines (1D), planes (2D) and volumes (3D). Douillard et al. (2011) present a set of 3D segmentation methods based on voxelisation and meshing. Their algorithms are evaluated on manually labeled datasets and the best performance is achieved using clustering approaches.

Several general segmentation and classification frameworks can be also found in the literature. Golovinskiy et al. (2009) develop a set of algorithms to detect, segment, characterize and classify urban objects. Their method is evaluated on an ALS/TLS database from Ohio (USA). Their pipeline is as follows: (i) ground segmentation using graph cuts, (ii) object detection and segmentation using hierarchical clustering, (iii) object characterization using geometrical and contextual descriptors, and (iv) object classification using SVM. Recently, Velizhev et al. (2012) have improved this workflow including spin images and implicit shape models. The major problems of these approaches are noise, sparse sampling and proximity between objects. Moreover, some prior knowledge about the object scale is required to set up thresholds. Schnabel et al. (2008) present a semantic system for 3D shape detection. Their algorithm consists in two main steps: (i) a topology graph is built with primitive shapes extracted from the data; (ii) a search is carried out in order to detect characteristic subgraphs of semantic entities. The main drawback is the graph complexity when dealing with non-trivial objects. Pu et al. (2011) propose a framework for segmenting and classifying urban objects from MLS data. This work starts with a rough classification into three large categories: ground, on-ground objects and off-ground objects. Then, based on geometrical attributes and topological relations, more detailed classes such as traffic signs, trees, building walls and barriers are recognized. Owechko et al. (2010) describe a similar pipeline: first, a spatial cueing is applied in order to identify potential objects; then, statistical classifiers based on decision trees are trained with geometrical and contextual features. Using these methods, occlusions and point density distribution are critical. Additionally, there is barely any problem recognizing large flat features such as

**Table 1**  
Comparison of the state of the art (P: Precision, R: Recall). Colors indicate similar methods used by different authors.

Authors	Detection and segmentation	Classification	Number of classes	Accuracy
Mallet et al. (2008)	Full-waveform analysis, mathematical morphology	SVM	3 (Buildings, ground, vegetation)	$P = 95.0\%$
Golovinskiy et al. (2009)	Elevation images, graphs, contextual analysis	Hierarchical clustering, SVM	16 (Cars, pole-like objects, trash cans, parking meters, ...)	$P = 58\%$ , $R = 65\%$
Hernández and Marcotegui (2009b)	Elevation images, mathematical morphology	SVM, Linear Discriminant Analysis	4 (Cars, lampposts, pedestrians, others)	$P = 86.21\%$
Munoz et al. (2009)	Contextual analysis, clustering	High-order Markov models	5 (Vegetation, wires, poles/trunks, load bearing, facades)	$P = 87.1\%$
Owechko et al. (2010)	3D strip by strip processing	Decision trees	17 (Buildings, ground, cars, bollards, lampposts, trees, ...)	$P = 70.0\%$
Zhu et al. (2010)	Elevation images, graph-cuts	SVM, decision trees	7 (Buildings, bushes, cars, trees, pedestrians, bicycles, others)	$P = 89.6\%$
Demantke et al. (2010)	3D adaptive neighborhood, Principal Component Analysis	Decision trees, dimensionality features	4 (Lines, planes, volumes, noise)	$P = 69.3\%$
Douillard et al. (2011)	Voxelisation, hierarchical clustering	Decision trees, RANSAC, clustering	16 (Ground and several urban objects)	$P = 89.0\%$
Rutzinger et al. (2011)	3D Hough transform, region growing	Shape models, 3D alpha shapes	2 (Trees, non-tree)	$P = 93\%$ , $R = 86\%$
Pu et al. (2011)	Geometrical and topological analysis	Decision trees	3 (Poles, trees, others)	$P = 73.5\%$
Velizhev et al. (2012)	RANSAC, hierarchical clustering, spin images	Implicit shape models	2 (Cars, light poles)	$P = 69\%$ , $R = 80\%$

ground, barriers and walls. However, there are some problems classifying pole-like objects such as trees, bollards and lampposts.

In order to solve these problems, several specific approaches have been proposed. For instance, Mallet et al. (2011) investigate the potential of full-waveform LiDAR data for urban areas classification. In that work, waveform features are used as input for a SVM classifier. Their results show that echo amplitude and radiometric features are suitable to classify buildings, ground and vegetation. Rutzinger et al. (2011) describe an automated workflow to segment and to model trees from MLS data. First, the input point cloud is segmented into planar regions using the 3D Hough Transform and surface growing algorithms. Then, the remaining small segments are merged applying a connectivity analysis. Next, non-tree objects are removed from the analysis using statistical measures. Finally, trees are thinned using 3D alpha shapes (Edelsbrunner and Mücke, 1994) and realistic 3D models are generated. Zhou and Vosselman (2012) segment and model curbstones from ALS/MLS data. Their process is performed directly on the 3D point cloud, on a strip by strip basis, so intrinsic information between the neighboring strips is missing. Recently, Serna and Marcotegui (2013b) solved this problem by processing all strips at the same time using elevation images.

In the present work, we aim at developing a method to detect, segment and classify urban objects, suitable for large scale applications. We adopt a method based on elevation images because of their demonstrated efficiency in terms of result quality and computational time. Our method is fully-automatic using few a priori information, it is based on robust morphological operators and supervised classification. It can manage partial occlusions, it is robust to noise, and re-segmentation process is carried out in order to separate connected objects. Simple geometrical and contextual features lead to better results than other works reported in the literature. Additionally, computation is faster than other works because image elevation reduces the amount of data to be processed.

This work provides an incremental contribution over Hernández and Marcotegui (2009b) work. The main contributions of this paper are the improvements in the detection and classification steps: (i) an improved object detection is provided dealing with objects located at the border of the scene and also thin vertical objects, such as poles (Section 3.3); (ii) classification is carried out in an effective way using simple geometrical and contextual features and a hierarchical classification is proposed (Section 3.5); (iii) finally, this work presents quantitative results on Paris and Ohio databases, leading to comparisons with the state of the art (Section 4).

### 3. Proposed methodology

Our general workflow is shown in Fig. 1. First, the 3D point cloud is projected to elevation images. At that point, a digital terrain model (DTM) is automatically created and object hypotheses are generated as discontinuities on the ground. Facades are automatically segmented as the highest vertical structures in the elevation image. Then, small and isolated regions are eliminated and

connected objects are segmented. As a result of the segmentation process, a label image is created containing a unique identifier for each segmented object. Next, several geometrical and contextual features are computed for each object and classification is carried out. As a result of the classification process, a class image is created containing a category for each segmented object. Having labels and classes in two different images is useful in the case of connected objects belonging to the same class, e.g. alignments of parked cars. Finally, the label and class images are reprojected to the 3D point cloud in order to get the final result. This reprojection step transforms the 2D resulting images into a 3D point cloud. For this purpose, all 3D points projected on a given pixel take the label and the class from that pixel. This step is required only if the result have to be displayed in 3D. Detailed descriptions are presented in following subsections.

#### 3.1. Elevation images

Elevation images are 2.5D structures that contain altitude information at each pixel. 3D point clouds are projected to elevation images because they are convenient structures to visualize and to process data. One can utilize all the large collection of existing image processing tools, in particular mathematical morphology (Matheron, 1975; Serra, 1988; Soille, 2003). Additionally, images can be processed quickly, implicitly define neighborhood relationships and require less memory than 3D data.

Elevation images are generated by an orthographic projection of the 3D point cloud using a virtual camera. This projection is a transformation from  $\mathbf{R}^3 \rightarrow \mathbf{N}^2$ . The virtual camera is located on the horizontal plane with normal vector  $\vec{n} = (0, 0, 1)$  and crossing the lowest point in the point cloud  $(0, 0, z_{min})$ . Thus, each pixel on the elevation image contains the elevation of the grid cell above  $z_{min}$ . The only free parameter of this projection is the spatial pixel size ( $pw$ ), which has to be carefully chosen. On one hand, if  $pw$  is too large, too many points would be projected on the same pixel losing fine details. On the other hand, too small  $pw$  implies connectivity problems and large image sizes, which would no longer justify the use of elevation images instead of 3D point clouds. To avoid connectivity problems and loss of information,  $pw$  is chosen according to the point cloud resolution, as explained in Section 4.

In general, several points are projected on the same pixel. Thus, four images are defined: (i) *maximal elevation image*, or simply elevation image, stores the maximal elevation among all projected points on the same pixel; (ii) *minimal elevation image*, which stores the minimal elevation among all projected points on the same pixel; (iii) *height difference image*, which contains the difference between maximal and minimal elevation images; and, (iv) *accumulation image*, which stores the number of points projected on each pixel. In general, processing steps are performed on the elevation image. The other images are used to support some decisions during the analysis or to compute object features.

After image creation, a morphological interpolation is performed in order to fill holes caused by occlusions and missing scan lines. An interpolation technique based on the morphological fill

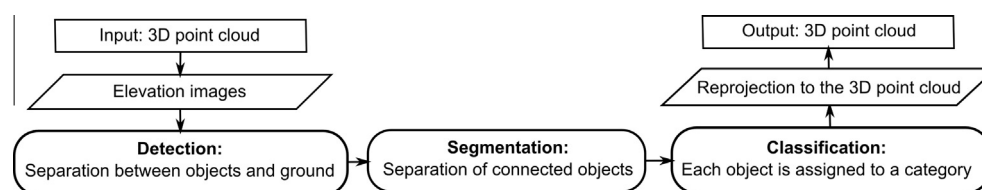


Fig. 1. Detection, segmentation and classification of urban objects from 3D point clouds.

holes operation ( $Fill(f)$ ) is preferred since this transformation does not create new regional maxima in the image. In the most simple sense, a hole is a dark region (i.e. surrounded by brighter pixels), not connected to the image border. This interpolation strategy has been proposed by Hernández and Marcotegui (2009a) and a detailed explanation can be found in (Serna and Marcotegui, 2013b).

When detection, segmentation and classification have been carried out, images are reprojected to the 3D point cloud. Fig. 2 describes the 3D point cloud processing using elevation images. A detailed explanation is presented in the following subsections.

### 3.2. Ground segmentation

Ground segmentation is a critical step since urban objects are assumed to be located on it. When objects are filtered from the ground mask, the DTM can be generated. With the aim of segmenting ground, we use the approach proposed by Hernández and Marcotegui (2009a). It is based on the  $\lambda$ -flat zones labeling algorithm, firstly introduced in image processing by Nagao et al. (1979), defined by Meyer (1998) as:

**Definition 1.** Let  $f$  be a digital gray-scale image  $f : D \rightarrow V$ , with  $D \subset Z^2$  the image domain and  $V = [0, \dots, R]$  the set of gray levels. Two neighboring pixels  $p, q$  belong to the same  $\lambda$ -flat zone of  $f$ , if their difference  $|f_p - f_q|$  is smaller than or equal to a given  $\lambda$  value. For all  $x \in D$ , let  $A_x(\lambda)$  be the  $\lambda$ -flat zone of image  $f$  containing pixel  $x$ .

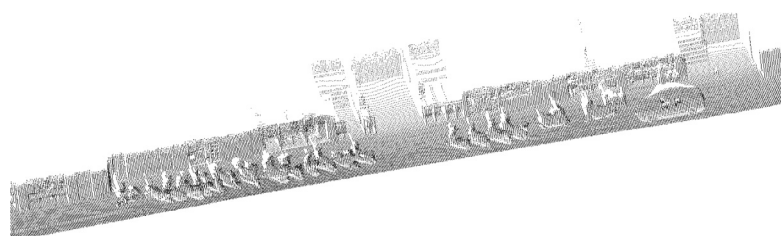
$$A_x(\lambda) = \{x\} \cup \{q | \exists \varphi = (p_1 = x, \dots, p_n = q) \text{ such that } |f_{p_j} - f_{p_{j+1}}| \leq \lambda\} \quad (1)$$

The  $\lambda$ -flat zone labeling leads to a segmentation of the image, “that is, a partition into disjoint connected subsets (called segments) such that there exists a logical predicate returning true on each segment but false on any union of adjacent segments” (Horowitz and Pavlidis, 1974). With this definition, we want to obtain the ground mask  $g_m(f) = \max \arg\{|A_x(\lambda)|\}$  as the largest  $\lambda$ -flat zone in the elevation image. We set  $\lambda = 20$  cm because it is usually high enough to merge road and sidewalk without merging other objects, even if there is no ramp access for the sidewalk.

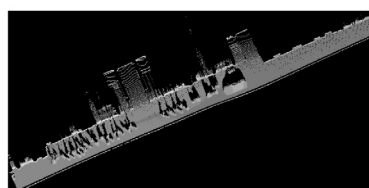
### 3.3. Object detection

Our object detection method is based on mathematical morphology, inspired by Hernández and Marcotegui (2009a). They propose to detect urban objects using the top-hat by filling holes (THFH) followed by an area opening. In the first step, THFH is an effective and parameterless way to extract objects that appear as bumps on the elevation image. However, it fails extracting objects touching the image border because they are not considered as bumps. In the second step, an area opening  $\gamma_{A_{min}}$  (Vincent, 1992) is performed in order to filter out small and noisy structures. Area opening is a morphological filter that removes objects with an area smaller than a given threshold  $A_{min}$ . This procedure is effective to get rid of noisy and isolated regions. However, it also removes thin objects such as bollards. In general, pole-like objects have a small area when they are seen from a nadir view, so they are suppressed by this filter. In this section, we propose an object detection framework that solves these two problems.

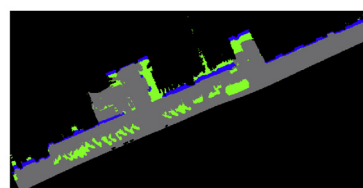
In order to solve the drawbacks of THFH step, a twofold strategy is proposed. A structure is considered to be object candidate if at least one of the two following conditions are fulfilled: (i) it has



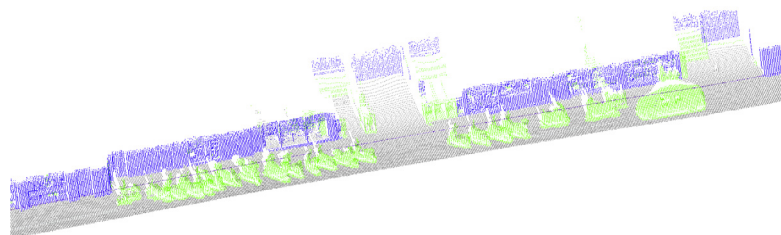
(a) Input point cloud



(b) Elevation image



(c) Segmented image



(d) Reprojection to the 3D point cloud

**Fig. 2.** 3D point cloud processing using elevation images. Segmentation results: ground (gray), facade (blue), objects (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

not been reached by the  $\lambda$ -flat zones algorithm, i.e. it does not belong to the ground mask; (ii) it appears as a bump on the elevation image. Therefore, the first set of object candidates is the ground residue, which is computed by the arithmetic difference between the elevation image and the ground mask ( $f - g_m(f)$ ). The second set of object candidates is extracted using the THFH( $f$ ), as proposed originally by Hernández and Marcotegui (2009a). Then, the union of these sets is performed in order to get all object candidates. In order to solve the  $\gamma_{A_{min}}$  drawbacks, the accumulation image is used. In general, vertical structures have high accumulation values. Thus, pole-like objects can be easily reinserted since their accumulation is higher than the accumulation for noisy objects.

Let us explain our detection method with an example. Fig. 3 illustrates a typical acquisition profile. The urban profile contains the following urban objects enumerated from 1 to 7: (1) car, (2) pedestrian, (3) noisy structure, (4) dog, (5) pedestrian, (6) house facade, and (7) chimney. Note that this is only an illustrative example on a 1D profile. The processing is performed on the entire 2.5D elevation image.

The first step consists in interpolating occluded zones using a fill holes transformation, as explained in Subsection 3.1. Fig. 3(a) presents the interpolated profile  $f$ . Using this transformation, each hole is filled with the minimal value surrounding the hole. For example, consider the hole in the left part, between objects 2 and 4. This hole is filled at the ground level because in 2.5D it is connected to ground pixels. Additionally, consider the hole in the right part, between objects 5 and 6. This hole is also filled at the ground level even if the ground is not the minimal value surrounding that hole in this 1D profile. We assume that this hole can be filled at that level because the ground is not occluded by the pedestrian (object 5) in the previous or following profiles.

Fig. 3(b) presents the first set of object candidates obtained as the ground residue. Note that almost all objects are retrieved. However, the dog in the middle of the sidewalk (object 4) is not detected because it is too low, thus it has been reached by the  $\lambda$ -flat zones propagation.

In order to obtain the second set of object candidates, the profile is inverted and holes are filled using the morphological fill holes transformation, as shown in Fig. 3(c). Then, the transformation  $\text{THFH}(f) = \text{Fill}(\hat{f}) - \hat{f}$  consists in subtracting the inverted image  $\hat{f}$  from the inverted filled image  $\text{Fill}(\hat{f})$ , as shown in Fig. 3(d). Note that this transformation correctly detects the dog in the middle of the sidewalk (object 4). However, the car in the left part (object 1) and the house in the right part (objects 6 and 7) are not retrieved because they are touching the border and then they do not become holes in the inverted profile. Fig. 3(e) presents the complete set of object candidates, computed as the supremum between the two aforementioned sets of candidates  $(f - g_m(f)) \vee \text{THFH}(f)$ .

Fig. 3(f) illustrates the effect of  $\gamma_{A_{min}}$  in order to eliminate small and noisy structures. Note that the noisy structure in the middle of the sidewalk (object 3) has been correctly eliminated. However, the chimney (object 7) has also been suppressed. Finally, Fig. 3(g) shows the result of the detection process, where the chimney has been reinserted because it has an important accumulation value.

Fig. 4 illustrates the detection process on real data. Note that all objects are detected by our method. For a better understanding, facades are marked in a different color. In our experiments, facades are automatically segmented as the highest vertical structures in the elevation image using a controlled reconstruction from markers, as explained in our previous work (Serna and Marcotegui, 2013a). Facade segmentation is out of scope of this work, but if one is interested in detecting facades independently, several other works are available in the literature (Boulaassal et al., 2007; Hammoudi, 2011; Rutzinger et al., 2011; Poreba and Goulette, 2012).

Fig. 5a illustrates the pole-like object reinsertion. Note that several pole-like objects are removed by an area opening filter with  $A_{min} = 0.1 \text{ m}^2$ . In Fig. 5(b), objects with an accumulation higher than 10 are reinserted (in red). Note that a tilted bollard (black) is not recovered because it has not enough accumulation. A lower threshold can be used in order to retrieve this tilted bollard but at the risk of preserving other noisy structures.

### 3.4. Object segmentation

Using our detection approach, it is possible to get several objects, close to each other, merged into a single connected component (CC). For example, in the left part of Fig. 3, a pedestrian (object 2) and a car (object 1) are detected in the same CC. Another example is shown in Fig. 6(a), where several cars are merged into a single CC. In order to solve this problem, we apply the solution proposed by Hernández and Marcotegui (2009b): “the number of connected objects in the same CC is equal to the number of significant maxima on it”. With the aim of preserving only the most significant maxima, i.e. to get rid of maxima due to texture and noise on the upper part of the objects, a morphological  $h$ -Maxima filter is used (Soille, 2003). The  $h$ -Maxima filter eliminates maxima with a low local contrast whose relative height is less than or equal to a given threshold  $h$ . Using filtered maxima as markers, a constrained watershed on the elevation image is applied in order to segment connected objects. Fig. 6 illustrates the performance of this resegmentation.

The main disadvantage appears when segmenting objects such as bikes, fences, trees or lampposts with several arms. They could be over-segmented because they have more than one significant maximum in the elevation image.

### 3.5. Object classification

Several classification methods have already been applied to 3D data in urban areas. In general, supervised classifiers are preferred since they offer a higher performance. In addition to the feature vector, a set of labels associated to each training sample is required. This set is called the training dataset, which is used to estimate the parameters of the classifier. An important underlying assumption is that the whole dataset has similar feature distribution with respect to the training dataset. This means that test and training datasets must have similar features in order to achieve a good performance. To prevent over-fitting, bootstrapping or cross-validation techniques can be used.

In our work, SVM is chosen because it has remarkable abilities to deal with both high-dimensional data and limited training sets, it is easy to implement, a simple set of features is used as input and a good performance is reported in the literature for similar applications (Mallet et al., 2008; Hernández and Marcotegui, 2009b; Alexander et al., 2010; Mountrakis et al., 2011). Other methods, such as random forests and high order Markov models could also be suitable and they are known for providing similar performance (Anguelov et al., 2005; Mallet et al., 2008; Munoz et al., 2009).

In order to build the feature vector, three set of features are used:

- *Geometrical features*: object area and perimeter; bounding box area; maximum, mean, standard deviation and mode (the most frequent value) of the object height; object volume, computed as the integral of the elevation image over each object.
- *Contextual features*: Neighboring objects  $N_{neigh}$ , defined as the number of regions touching the object, using 8-connectivity on the elevation image. This feature is very discriminative in the case of group of trees and cars parked next to each other; confidence index  $C_{ind} = n_{real} / (n_{real} + n_{interp})$ , where  $n_{real}$  and

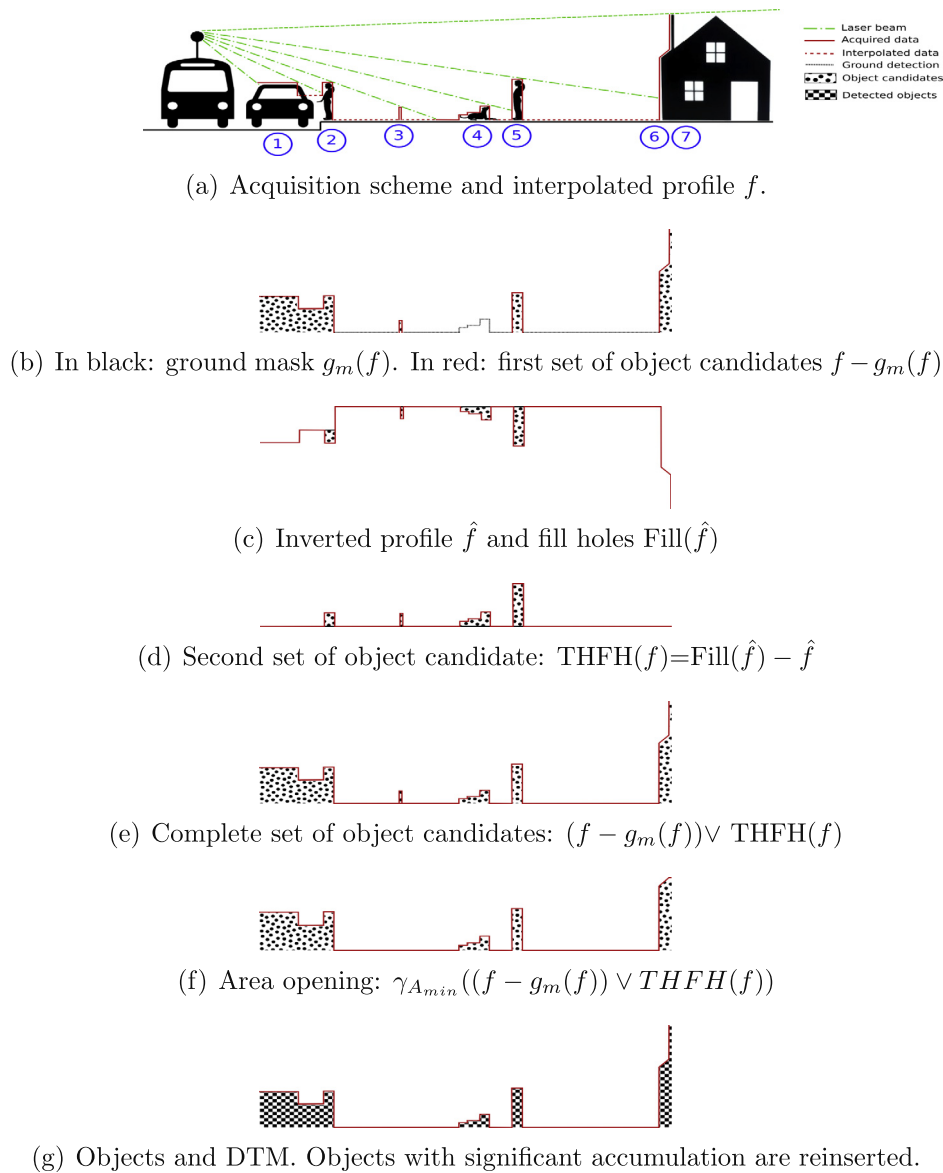


Fig. 3. Detection method on a 1D profile.

$n_{interp}$  are the number of non-empty object pixels before and after elevation image interpolation, respectively. In general, occluded and far objects have a low confidence index.

- **Color features:** Average red, green and blue over the object. These features are used if available.

The reliability of these features depends on the acquisition system. Accurate and calibrated sensors contribute to compute accurate features and to get a good classification performance. Note that geometrical features can be adapted to any XYZ point cloud, taking into account the acquisition system resolution. In our experiments, geometrical features are computed in the international unit system (SI units).

### 3.5.1. Hierarchical classification

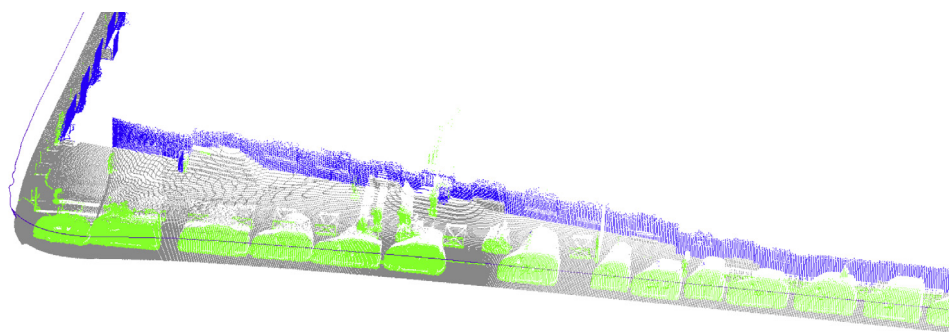
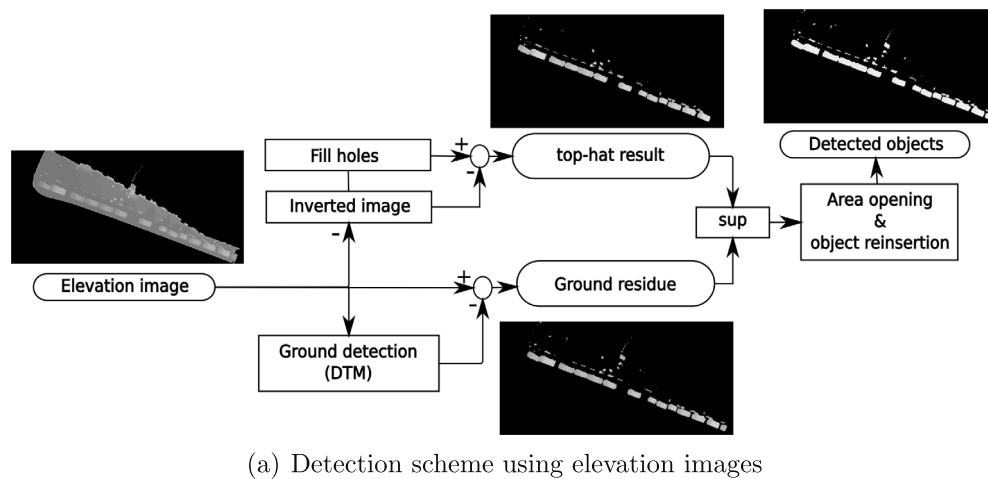
With the aim of reducing confusion between classes with similar features and few examples in the database, we propose a hierarchical classification approach. The idea of hierarchical classification comes directly from the study of biological perceptual systems (Hubel and Wiesel, 1962; Poggio and Shelton,

1999), and it has been also applied in the remote sensing community (Avci and Akyurek, 2004; Pu et al., 2011).

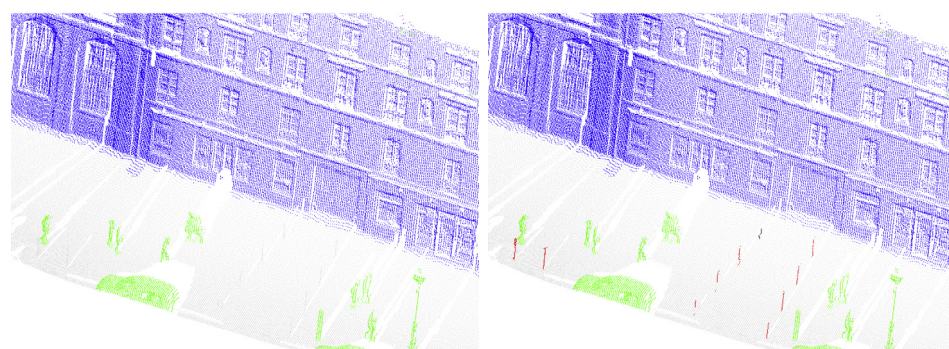
First, data are separated into two parts: training and test sets. The definition of the hierarchical steps is entirely carried out on the training dataset.

Our hierarchical classification begins using general classes, then it continues in a top-down approach until obtaining more detailed classes.

This approach can be implemented as follows: (i) an analysis is carried out on the training dataset applying a global classification taking all available classes into account; (ii) training errors are computed using a  $k$ -fold cross-validation approach. In  $k$ -fold cross-validation, we first divide the training set into  $k$  subsets of equal size. In our experiments, we have used  $k = 10$ . Sequentially one subset is tested using the classifier trained on the remaining  $k - 1$  subsets. Thus, each instance of the whole training set is predicted once. (iii) classical Precision  $P(\text{train})$ , Recall  $R(\text{train})$  and  $f_{\text{mean}}(\text{train}) = (2 \times P(\text{train}) \times R(\text{train})) / (P(\text{train}) + R(\text{train}))$  statistics are computed in order to evaluate our training results. Classes with high confusion rates ( $f_{\text{mean}}(\text{train})$  lower than 80%) are identified. In



**Fig. 4.** Object detection using the top-hat by filling holes and the ground residue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 5.** Pole reinsertion using accumulation. In the real scene there are 10 bollards. However, one of them is not reinserted because it is tilted, thus it has not enough accumulated points on the same pixel. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

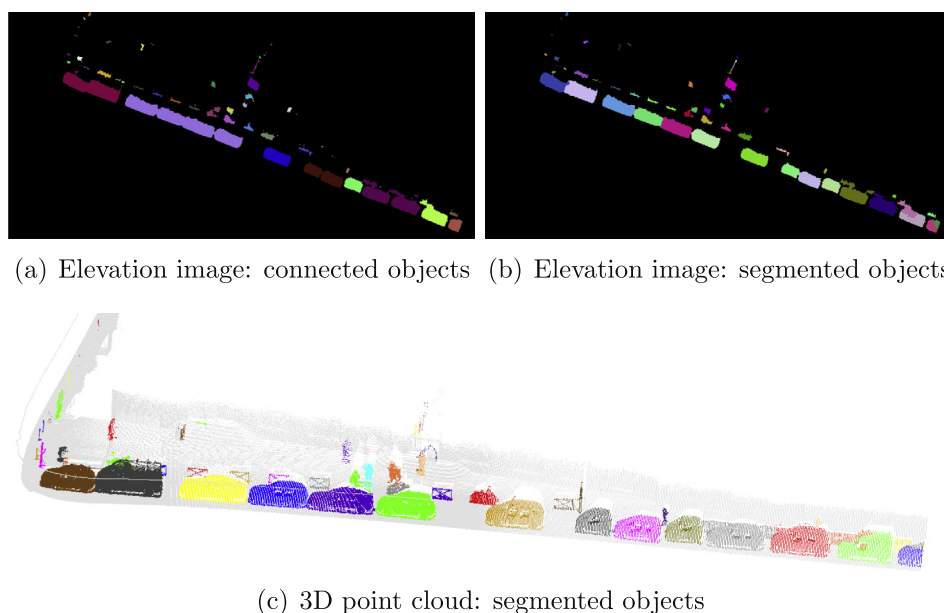
general, these classes correspond to heterogeneous objects with few examples. These classes are gathered in more general new classes; (iv) using the whole training dataset, two kind of classifiers are trained: the first one is a classifier trained with the well-distinguished original classes and the new general ones; the second one is a more specific classifier used for each new general class aiming at obtaining more detailed classes; and (v) the process can be iterated. In our experiments, only two levels of hierarchy have been used.

Then, the resulting classifier is used to predict the test dataset. Precision  $P(test)$ , Recall  $R(test)$  and  $f_{mean}(test)$  results reported in Section 4 have been computed on the test dataset and reflect the performances of our system on real operation conditions.

#### 4. Results

Our methodology is evaluated on three databases: rue Soufflot (Paris), Ohio (USA) and rues Vaugirard-Madame (Paris). As a





(a) Elevation image: connected objects (b) Elevation image: segmented objects

(c) 3D point cloud: segmented objects

**Fig. 6.** Object segmentation using a constraint watershed from object maxima. Each color represents a different object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

general remark, our experiments demonstrate that almost all objects are retrieved by our detection approach. Then, segmentation is useful to separate connected objects such as pedestrians and cars. However, bikes and trees can be over-segmented. Finally, classification is carried out in an effective way using simple geometrical and contextual features. In our experiments, spatial pixel size is set to  $0.04 \text{ m}^2$  ( $pw = 20 \text{ cm}$  width) and  $0.01 \text{ m}^2$  ( $pw = 10 \text{ cm}$  width) for Ohio and Paris datasets, respectively.

It is noteworthy that our algorithms were initially developed to process Paris databases in the framework of TerraMobilita project. One of the main advantages of our method is that it can be easily generalized to other datasets without any major modification. This is underlined by the good results obtained on the Ohio database. Detailed results are presented below.

#### 4.1. Rue Soufflot, Paris

For this experiment, we use a manual labeled dataset from rue Soufflot, a street approximately 500 m long in the 5th Parisian district. Acquisition was done by the Stereopolis MLS system from the French National Mapping Agency (IGN) (Paparoditis et al., 2012). A typical scene is shown in Fig. 7. It contains pedestrians, cars, lampposts, motorcycles, among others. This database was firstly used by Hernández and Marcotegui (2009b) to classify objects in four categories: cars, lampposts, pedestrians and others. However, his original annotation is no longer available. For the sake of comparison, we have manually annotated the database again and managed to reproduce results consistent with those reported by the author (shown in brackets in Table 2).

First, data are separated into two parts, training and test sets. This separation has been randomly done keeping 50% of the objects of each class in the training set and the rest in the test set.

Color is not available in this database, thus only geometrical and contextual features have been used. In a first attempt, a single SVM classifier has been trained for all available categories. Training errors have been computed using 10-fold cross validation and high confusion rates were found between heterogeneous classes and classes with few examples, as shown in Fig. 8(a). To solve these

problems, the hierarchical classification proposed in Section 3.5 is applied, as shown in Fig. 8(b). The first SVM classifies well-discriminated objects ( $f_{mean}(train)$  greater than 80%), while the second one is exclusively dedicated to classes with higher confusion rates ( $f_{mean}(train)$  lower than 80%). Table 2 presents our classification results on the test set.

Our main contribution in the classification step is the use of contextual features and hierarchical SVM. With respect to Hernández and Marcotegui (2009b) work, classification results have been improved. On the one hand, cars and lamppost classification have the same maximal accuracy (100%) while the performance on the pedestrian class has been improved by about 15%. On the other hand, we use all available categories preserving the performance on cars and lampposts categories. The main problems appear with classes *furniture* and *others* because they are very heterogeneous. The same problem appears for *traffic lights* and *trash cans* classes because there are not enough samples in the database (4 and 5 samples, respectively).

#### 4.2. Ohio database

The Ohio database has also been used by Golovinskiy et al. (2009) and Velizhev et al. (2012) in order to evaluate their detection, segmentation and classification methods. This dataset is a combination of ALS and TLS data scanned in Ottawa city (Ohio, USA). It contains 26 tiles,  $100 \times 100 \text{ m}$  (approximately  $4 \times 10^6$  points) each, as shown in Fig. 9. A typical scene contains trees, cars, lampposts, among others. The ground-truth (GT) consists in a labeled point marking the center of each object and its class.

Since our method is sequential, i.e. the input of each processing step is the output of the previous one, its evaluation is carried out in the same way. First, the detection process is applied to the entire database; second, detected objects are used as input for the segmentation step; and third, correctly segmented objects are separated in two subsets (train and test) in order to perform the classification. Let us to explain each processing step and its evaluation.

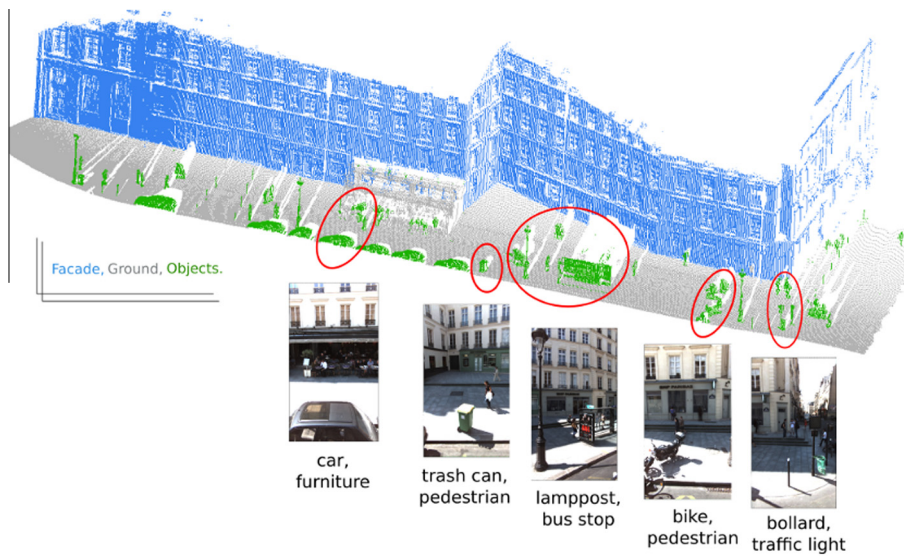


Fig. 7. Example of urban objects manually annotated in the rue Soufflot dataset (Paris). Acquired by IGN-Stereopolis system.

**Table 2**  
Classification results on rue Soufflot test set. In brackets results from Hernández and Marcotegui (2009b).

Class	Samples	Precision (%)	Recall (%)	$f_{mean}$ (%)
Cars	27	100 (100)	100 (100)	100 (100)
Lampposts	12	100 (100)	100 (100)	100 (100)
Bollards	39	89	100	94
Walls	12	100	100	100
Fences	5	100	100	100
Pedestrians	101	86 (70)	84 (71)	85 (71)
Bikes	14	100	54	70
Furniture	30	67	67	67
Others	23	50	100	66.6
Traffic lights	4	0	0	0
Panels	7	100	100	100
Trash cans	5	0	0	0

#### 4.2.1. Evaluation: Detection

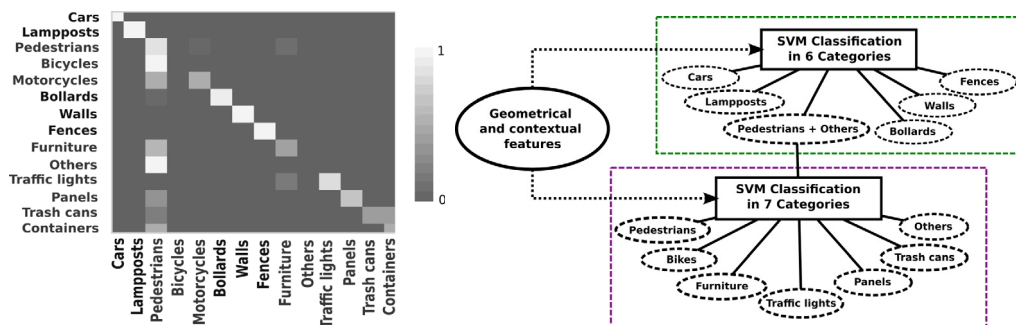
In order to evaluate our detection approach, an object is considered to be correctly detected if its GT center is included in the object hypotheses mask (Subsection 3.3), i.e. it has not been suppressed by any noise filter and it has not been wrongly merged with the ground. Note that an object hypothesis may contain several connected objects or only a part of an object. In the detection step, we are only interested in keeping all possible objects. This is

important because non detected objects cannot be recovered in the subsequent steps. Table 3 presents the percentage of retrieved objects in this database. Our detection method retrieves 98% of the objects, which outperforms other methods reported in the literature (92% by Golovinskiy et al. (2009) and 96% by Velizhev et al. (2012)). The number of false alarms cannot be estimated because many objects located on building roofs and in the forest are detected by our method (since they are real objects), but they have not been annotated in the database. Fig. 10 shows the detection results on the 3D point cloud.

#### 4.2.2. Evaluation: Segmentation

In order to evaluate our segmentation approach, an object is considered to be correctly segmented if it is isolated as a single object, i.e. connected objects are correctly separated (there is no under-segmentation) and each individual object is inside one and only one connected component (there is no over-segmentation). However, an estimation of under-segmentation and over-segmentation errors cannot be done on the Ohio database because it only contains a GT point for each object. In that sense, an object is considered to be correctly segmented if it is marked with one and only one GT point.

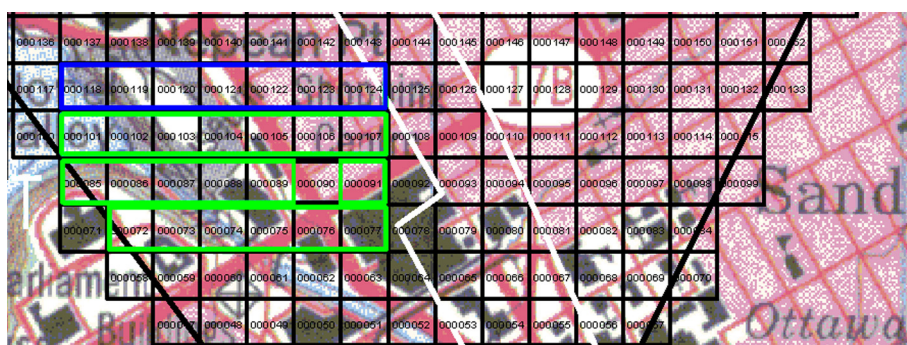
As shown in Table 3, our method segments correctly 76% of the detected objects. Objects such as cars, lampposts, parking meters and signs are correctly segmented (Recall greater than 80%). The



(a) Confusion matrix using all available categories on the training set.

(b) Hierarchical classification

Fig. 8. Hierarchical SVM classification on rue Soufflot dataset.



**Fig. 9.** Ottawa city, Ohio (USA). The database contains 26 annotated tiles 100 × 100 m each. Blue: train set; green: test set; black: non-annotated data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 3**  
Detection and segmentation results on the Ohio dataset.

Class	Name	GT	Detection		Segmentation	
			Detected	Recall (%)	Segmented	Recall (%)
1	Ad cylinder	6	6	100	5	83
2	Bush	29	28	97	23	82
3	Car	240	237	99	195	82
4	Dumpster	1	1	100	1	100
5	Fire hydrant	19	16	84	13	81
6	Flagpole	2	2	100	2	100
7	Lamppost	146	143	98	117	82
8	Light pole	62	60	97	46	77
9	Mailing box	4	4	100	1	25
10	Newspaper box	42	35	83	5	14
11	Parking meter	10	10	100	10	100
12	Post	377	376	100	208	55
13	Recycle bin	6	6	100	3	50
14	Sign	96	92	96	79	86
15	Telephone booth	4	4	100	2	50
16	Traffic ctrl. box	8	5	63	2	40
17	Traffic light	42	42	100	34	81
18	Trash can	19	19	100	8	42
19	Tree	552	543	98	490	90
20	Box transformer	2	2	100	0	0
Total		1667	1631	98	1244	76

main problem comes from under-segmentation of connected objects such as light poles, posts and trees. Since this kind of clusters has only one maximum on the elevation image (the highest object), they are not correctly segmented by our method. Note that trees recall is 90%, which means a good segmentation. However, trees represent approximately 34% of the objects in the database, which implies that under-segmented trees affect seriously the recall of other classes, in particular for classes with few objects.

#### 4.2.3. Evaluation: Classification

For the classification experiments, segmented objects in the north quarter of the city (7 tiles, 458 objects) are used for training and the rest (19 tiles, 677 objects) for testing. Training and testing tiles are the same as in (Golovinskiy et al., 2009), for comparison purposes. The number of objects per class on both training and test sets are detailed in Table 5.

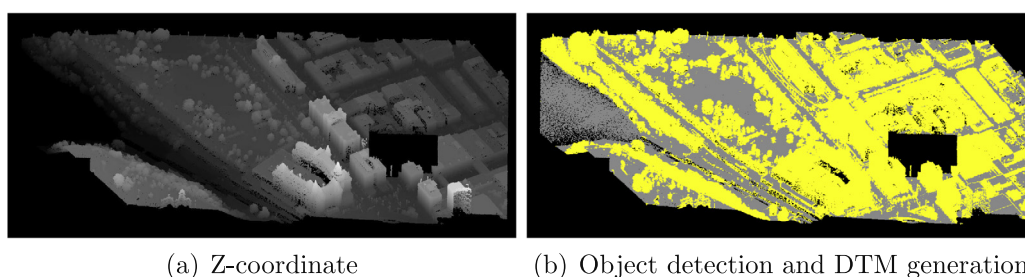
Geometrical, contextual and color features (Subsection 3.5) are combined in this experiment in order to define the best classification features. Classification performance obtained using different combinations of them is given in Table 4. The best overall accuracy (82%), defined as the ratio between the number of correctly classified objects and the total number of objects, is obtained combining geometrical and contextual features. Detailed results are presented in Table 5.

It is noteworthy that including color information degrades the classification accuracy. The reason is that in this database, color information is the result of overlapping several aerial and terrestrial scans. During acquisitions, color sensors were not calibrated, thus their superposition is not perceptually coherent, as shown in Fig. 11.

Table 5 shows detailed classification results. Precision, Recall and  $f_{mean}$  for each class are presented. In this experiment, classes with less than 5 objects, either in the training set or in the testing set, are not considered in the classification process. Therefore, only 6 categories have been used. It is noteworthy that cars, trees and posts are correctly classified. However, lampposts, lights and signs have lower accuracies.

For a better understanding, Table 6 shows the confusion matrix. Note that cars are correctly classified while lampposts, lights, posts, signs and trees are mixed up, which is comprehensible because they are pole-like objects.

In an attempt to solve these confusion problems, the hierarchical classification approach (proposed in Section 3.5) has been studied. Lampposts, lights, posts and signs have been put together in a



(a) Z-coordinate

(b) Object detection and DTM generation

**Fig. 10.** The Ohio database: object detection (yellow) and DTM generation (gray). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 4**  
Classification accuracy using different features combination.

Features	Overall accuracy (%)
Geometrical	75
Geometrical + $C_{ind}$	77
Geometrical + $C_{ind}$ + $N_{neigh}$	<b>82</b>
Geometrical + $C_{ind}$ + $N_{neigh}$ + color	72

The bold value indicates the best overall accuracy obtained for the combination of geometrical and contextual features.

**Table 5**  
Classification results on the Ohio dataset. Classes with less than 5 objects, either in the training set or in the testing set, are not considered. Pred: predicted, TP: true positives, P: Precision, and R: Recall.

Class	Name	Correctly segmented		Classification				
		Train	Test	Pred	TP	P (%)	R (%)	$f_{mean}$ (%)
1	Ad cylinder	2	3					
2	Bush	1	22					
3	Car	108	87	85	75	88	86	87
4	Dumpster	1	0					
5	Fire hydrant	3	10					
6	Flagpole	1	1					
7	Lamppost	33	84	78	51	65	61	63
8	Light pole	14	32	22	16	73	50	59
9	Mailing box	0	1					
10	Newspaper box	0	5					
11	Parking meter	10	0					
12	post	132	76	85	66	78	87	82
13	Recycle bin	1	2					
14	sign	34	45	44	33	75	73	74
15	Telephone booth	1	1					
16	Traffic ctrl. box	1	1					
17	Traffic light	4	30					
18	Trash can	0	8					
19	Tree	137	353	363	317	87	90	89
20	Box transformer	0	0					
Total (used classes)		458	677	677	558	82	82	82
Total (all objects)		483	761					

new class, while cars and trees are preserved in their original classes. A first classifier is applied to separate correctly discriminated objects, and a second one is exclusively dedicated to classes with higher confusion rates. After our experiments, we have noted that this approach does not provide any global improvement in this database since  $f_{mean}$  increases by 16% for lampposts and lights, but it decreases by 15% for posts and signs. The conclusion here is that a hierarchical approach is not enough to solve confusion

problems since objects are too similar. A possible solution is the use of other features which allow a clearer separation between classes.

Table 7 presents results gathering lampposts, lights, posts, and signs in a more general category called pole-like objects. With 3 classes, the overall accuracy rises up to 88%.

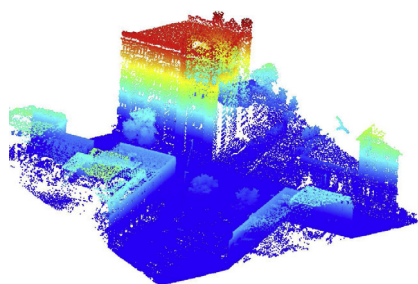
#### 4.2.4. Comparison with the state of the art

The Ohio database has been chosen because it contains many different objects, it is large enough to exemplify a large-scale application, and comparison with the state of the art is possible since it has been used in other works (Golovinskiy et al., 2009; Velizhev et al., 2012).

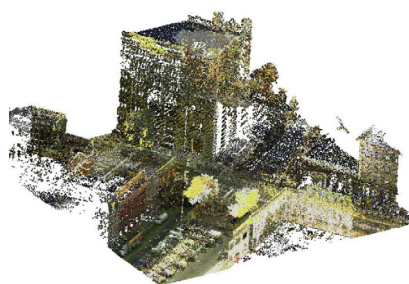
We present our results on 26 tiles. However, in the original publication by Golovinskiy et al. (2009) (the website containing the dataset is not longer available), they report 27 tiles. Therefore, the number of objects is not the same due to this missing tile. Additionally, some important differences have been noticed with respect to the aforementioned authors: on the one hand, with respect to Velizhev et al. (2012), they have only used 2 classes (cars and light poles), thus only a partial comparison can be done; on the other hand, with respect to Golovinskiy et al. (2009), the main difference comes from the fact that they do not consider trees nor bushes in their analysis.

Table 8 presents a quantitative comparison with the state of the art. Taking into account only 6 categories, the ones used during classification, our detection method (accuracy equal to 99%) performs better than the other two reported in the literature; our classification accuracy is equal to 82%, whereas Golovinskiy et al. (2009) correctly classify 65% of the objects considered by their method; with respect to the segmentation method, results from Velizhev are not available and our accuracy (78%) is 8% lower than that reported by Golovinskiy et al. (2009). On the one hand, our major under-segmentation problem is due to clusters formed by trees and pole-like objects, where the highest object is the only significant maximum. On the other hand, our major over-segmentation problem is when segmenting objects with several regional maxima such as trees. To summarize, our sequential method correctly detects, segments and classifies  $99\% \times 78\% \times 82\% = 64\%$  of the annotated objects.

With respect to computational time (last row in Table 8), our method is up to 10 times faster than the other two works. In spite of hardware differences, these three works use general-purpose machines and they are not specially optimized nor parallelized. The aim of this comparison is to give an idea to the reader about the computational time and the potential to large-scale or other time-constrained applications. One of the reasons of our faster processing is due to the use of elevation images and image processing algorithms since their computational cost is lower than that on the 3D case.



(a) Tile #72: Height information



(b) Tile #72: Color information

Fig. 11. The Ohio database.

**Table 6**  
Confusion matrix for classification in 6 classes on the Ohio database.

GT \ predict.	Cars	Lampposts	Light	Post	Sign	Tree	Total
Car	<b>75</b>	0	0	0	1	11	87
Lamppost	1	<b>51</b>	1	11	1	19	84
Light	0	6	<b>16</b>	0	0	10	32
Post	0	3	1	<b>66</b>	2	4	76
Sign	0	3	0	7	<b>33</b>	2	45
Tree	9	15	4	1	7	<b>317</b>	353
Total	85	78	22	85	44	363	

The bold values indicate the number of correctly classified objects per class.

**Table 7**  
Confusion matrix gathering lampposts, lights, posts, and signs in the same category. Results on the Ohio dataset.

GT \ predict.	Cars	Pole-like	Trees	Total	Precision (%)	Recall (%)	$f_{mean}$ (%)
Car	<b>75</b>	1	11	87	88	86	87
Pole-like	1	<b>201</b>	35	237	88	85	86
Trees	9	27	<b>317</b>	353	87	90	89
Total	85	229	363				

The bold values indicate the number of correctly classified objects per class.

**Table 8**  
Summarized comparison with other methods reported in the literature. The percent values indicate the accuracy in each stage of the workflow.

	Golovinskiy et al. (2009)	Velizhev et al. (2012)	Our method (2013)
Detection	92%	96%	99%
Segmentation	86%	N/A	78%
Classification	65%	67%	82%
Overall accuracy			<b>64%</b>
Computational time	7.3 min/tile (3 GHz PC)	5–10 min/tile (4 × 2.4 GHz PC)	1 min/tile (4 × 2.4 GHz PC)

The bold value indicates the overall accuracy of our sequential method.

Note that the typical speed of a MLS system is 30 km/h, which corresponds approximatively to a covered area of 10,000 m<sup>2</sup>/min on a 20 m wide street without considering stops nor traffic lights. In this database, our processing speed is 10,000 m<sup>2</sup>/min. This is a very fast off-line processing since acquisition and processing times are equal.

#### 4.3. Paris database: Rues Vaugirard-Madame

Dealing with cars has a particular interest in the framework of the TerraMobilita project since one of the applications consists in computing automatic parking statistics. In order to evaluate the potential of an automatic method, several 3D point clouds of the same street in Paris (Rues Vaugirard-Madame, approximatively a 500 m long section) have been acquired at different hours. Acquisition was done by the Stereopolis MLS system from the French National Mapping Agency (IGN) (Paparoditis et al., 2012). Then, we apply our automatic methodology in order to detect, segment and classify cars. For the classification step, urban objects were manually labeled. We use 2307 objects (129 cars and 2178 others) as training set, and 970 objects (53 cars and 917 others) as testing set. Note that a hierarchical classification is not applied since we are only interested in cars.

Color information is not available. Therefore, only geometrical and contextual features have been used. Table 9 presents our classification results using a binary SVM. The performance of our

**Table 9**  
Results: car classification.

Class	Precision (%)	Recall (%)	$f_{mean}$ (%)
Cars	100.0	94.6	97.2
Others	99.7	100.0	99.9

method is proved since 99.7% of the objects are correctly classified. Note that 5.4% of the cars have not been properly identified due to occlusion and over-segmentation problems.

In order to demonstrate that our method can be easily generalized, we have used a classifier trained on the Ohio dataset in order to classify rue Vaugirard-Madame cars. A  $f_{mean}$  equal to 90.0% has been obtained. This result is slightly lower than that reported in Table 9 (97.2%). However, the great advantage is that a new annotation may not be required when working with a new database.

At this point, our system is able to correctly extract cars and present some additional information such as the geographic position, geometric features and GPS time at the acquisition moment. However, a comparison between cars parked in the same place at different moments is required to compute parking duration statistics. In order to avoid confusions between those cars, geometrical and color features should be used. Additionally, relative sensor precision between different acquisitions becomes a critical issue. In efficiency terms, an automatic method seems to be suitable for this problem since the acquisition vehicle can go up to 20 times faster than a person. Additionally, the automatic processing takes only a few minutes and it is comparable to the acquisition time.

## 5. Conclusions

We propose an automatic and robust approach to detect, segment and classify urban objects from 3D point clouds. Processing is carried out using elevation images and the final result is presented reprojecting the image onto the 3D point cloud.

First, the ground is segmented using a lambda-flat zones propagation. Next, objects are detected using a two-fold strategy considering both structures connected to the boundary of the scene as well as ground discontinuities. Then, a filtering step is performed in order to reduce noise but preserving thin vertical structures. Subsequent, connected objects are segmented assuming that the number of significant maxima is equal to the number of connected objects. Finally, objects are classified in several categories using a SVM approach with geometrical and contextual features. Our geometrical features have can be adapted to any XYZ point cloud. Thus, the classification can be easily generalized, i.e. training on a database and testing on another one, as shown in rues Vaugirard-Madame dataset. This is a significant advantage because the model learned for a database can be applied to another one, even acquired by a different acquisition system, without the tedious manual annotation.

Our methodology is qualitatively and quantitatively evaluated on MLS and ALS/TLS databases from Paris (France) and Ohio (USA). Our results on the Ohio dataset show that our method retrieves 99% of the objects in the detection step, 78% of connected objects are correctly segmented, and 82% of correctly segmented ones are correctly classified using geometrical and contextual features. On Paris dataset, our proposed hierarchical classification leads to an improvement of about 15% on the pedestrian class with respect to previous works while preserving a good performance in other classes. Moreover, new classes (not considered in previous works) have been taken into account.

Our method is robust to noise since small and isolated structures are eliminated using morphological filters. Additionally, it is fast because we project 3D points onto an elevation image and

we process them as a complete set using digital image processing techniques.

Even if our method presents good results, it is noteworthy that several improvements should be done before developing a mature application. Our main problem, common to all methods in the literature, is due to large occluded regions. Several scans of the same zone could reduce this problem. Some under-segmentation and over-segmentation problems have been also pointed out. A possible solution can include shape/texture analysis to help deciding whether or not an object should be re-segmented.

Up to now, we have only used the spatial information available in the point cloud. However, additional features such as laser intensity and texture could improve our performance. Additionally, in the future we are planning to use Velodyne<sup>2</sup> data in order to distinguish static from mobile obstacles and to reduce occlusion problems.

### Acknowledgements

The work reported in this paper has been performed as part of Cap Digital Business Cluster TerraMobilita project.

We want to thank Alexander Velizhev for providing us with his results on the Ohio database and the dataset itself.

### References

- Alexander, C., Tansey, K., Kaduk, J., Holland, D., Tate, N.J., 2010. Backscatter coefficient as an attribute for the classification of full-waveform airborne laser scanning data in urban areas. *ISPRS J. Photogramm. Rem. Sens.* 65 (5), 423–432.
- Anguelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A., 2005. Discriminative learning of Markov random fields for segmentation of 3D scan data. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 2, pp. 169–176.
- Avcı, M., Akyurek, Z., 2004. A hierarchical classification of landsat TM imagery for landcover mapping. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXV-B4, pp. 511–516.
- Boulaassal, H., Grussenmeyer, P., Tarsha-kurdi, F., 2007. Automatic segmentation of building facades using terrestrial laser data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVI-3/W52, pp. 65–70.
- Demantke, J., Mallet, C., David, N., Vallet, B., 2010. Dimensionality based scale selection in 3D LiDAR point clouds. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-5/W12, pp. 97–102.
- Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A., 2011. On the segmentation of 3D LiDAR point clouds. In: *IEEE International Conference on Robotics and Automation, ICRA'11*, pp. 2798–2805.
- Edelsbrunner, H., Mücke, E.P., 1994. Three-dimensional alpha shapes. *ACM Trans. Graph.* 13, 43–72.
- Ferguson, D., Darms, M., Urmsen, C., Kolski, S., 2008. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In: *IEEE Intelligent Vehicles Symposium*, pp. 1149–1154.
- Golovinskiy, A., Kim, V.G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments. In: *12th IEEE International Conference on Computer Vision*, pp. 2154–2161.
- Gorte, B., 2007. Planar feature extraction in terrestrial laser scans using gradient based range image segmentation. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVI-3/W52, pp. 173–177.
- Hammoudi, K., 2011. Contributions to the 3D City Modeling. Ph.D. Thesis, Université Paris-Est.
- Hernández, J., Marcotegui, B., 2009a. Filtering of artifacts and pavement segmentation from mobile LiDAR data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-3/W8, pp. 329–333.
- Hernández, J., Marcotegui, B., 2009b. Point cloud segmentation towards urban ground modeling. In: *The 5th GRSS/ISPRS Joint Urban Remote Sensing Event (URBAN2009)*. Shanghai, China, pp. 1–5.
- Hoover, A., Jean-baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B., 1996. An experimental comparison of range image segmentation algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (7), 673–689.
- Horowitz, S.L., Pavlidis, T., 1974. Picture segmentation by a directed split-and-merge procedure. In: *Proceedings of the 2nd International Joint Conference on Pattern Recognition*, pp. 424–433.
- Hubel, D.H., Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160 (1), 106–154.
- Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagzent, D., Schröder, J., Thuy, M., Goebel, M., Hundelshausen, F.v., Pink, O., Frese, C., Stiller, C., 2008. Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge. *J. Field Robot.* 25 (9), 615–639.
- Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform LiDAR data for urban area classification. *ISPRS J. Photogramm. Rem. Sens.* 66 (6), 71–84.
- Mallet, C., Bretar, F., Soergel, U., 2008. Analysis of full-waveform LiDAR data for classification of urban areas. *Photogramm. Fernerkundung Geoinf. (PFG)* 5, 337–349.
- Matheron, G., 1975. *Random Sets and Integral Geometry*. John Wiley & Sons, New York.
- Meyer, F., 1998. From connected operators to levelings. *Mathematical Morphology and its Applications to Image and Signal Processing of Computational Imaging and Vision*, vol. 12. Kluwer Academic Publishers, pp. 191–198.
- Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: a review. *ISPRS J. Photogramm. Rem. Sens.* 66 (3), 247–259.
- Munoz, D., Vandapel, N.D., Hebert, M., 2009. Onboard contextual classification of 3-D point clouds with learned high-order Markov Random Fields. In: *IEEE International Conference on Robotics and Automation, ICRA '09*, pp. 2009–2016.
- Nagao, M., Matsuyama, T., Ikeda, Y., 1979. Region extraction and shape analysis in aerial photographs. *Comput. Graph. Image Process.* 10 (3), 195–223.
- Owechko, Y., Medasani, S., Korah, T., 2010. Automatic recognition of diverse 3-D objects and analysis of large urban scenes using ground and aerial LiDAR sensors. In: *Conference on Lasers and Electro-Optics (CLEO) and Quantum Electronics and Laser Science Conference (QELS)*, pp. 16–21.
- Papadimitris, N., Papellard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N., Houzay, E., 2012. Stereopolis II: a multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. *Rev. Fr. Photogramm. Télédétection* 200 (1), 69–79.
- Poggio, T., Shelton, C.R., 1999. Machine learning, machine vision, and the brain. *AI Mag.* 20 (3), 37–56.
- Poreba, M., Goulette, F., 2012. RANSAC algorithm and elements of graph theory for automatic plane detection in 3D point clouds. *Arch. Photogramm., Cartogr. Rem. Sens.* 24, 301–310.
- Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Rem. Sens.* 66 (6), 28–39.
- Rutzinger, M., Pratihast, A.K., Oude Elberink, S.J., Vosselman, G., 2011. Tree modelling from mobile laser scanning data-sets. *Photogramm. Record* 26 (135), 361–372.
- Schnabel, R., Wessel, R., Wahl, R., Klein, R., 2008. Shape recognition in 3D point clouds. In: Skala, V. (Ed.), *The 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision. Union Agency-Science Press*, pp. 1–8.
- Serna, A., Marcotegui, B., 2013a. Attribute controlled reconstruction and adaptive mathematical morphology. In: *11th International Symposium on Mathematical Morphology*. Uppsala, Sweden, pp. 205–216.
- Serna, A., Marcotegui, B., 2013b. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS J. Photogramm. Rem. Sens.* 84, 23–32.
- Serra, J., 1988. *Image Analysis and Mathematical Morphology*, vol. 2. Academic Press, London.
- Soille, P., 2003. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, Secaucus, NJ, USA.
- Velizhev, A., Shapovalov, R., Schindler, K., 2012. Implicit shape model for object detection in 3D point clouds. In: *The ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-3, pp. 179–184.
- Vincent, L., 1992. Morphological area openings and closings for grey-scale images. In: *Proceedings of the Workshop: Shape in Picture*. Springer, Driebergen, The Netherlands, pp. 197–208.
- Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. *Int. J. Appl. Earth Obs. Geoinf.* 18, 293–304.
- Zhu, X., Zhao, H., Liu, Y., Zhao, Y., Zha, H., 2010. Segmentation and classification of range image from an intelligent vehicle in urban environment. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010)*, pp. 1457–1462.

<sup>2</sup> Velodyne LiDAR: several lasers are mounted on upper and lower blocks of 32/64 lasers each and the entire unit spins. Taken from <http://velodynelidar.com/> [last accessed: 16.09.13].



# Urban accessibility diagnosis from mobile laser scanning data

Andrés Serna, Beatriz Marcotegui

*MINES ParisTech, CMM-Centre de Morphologie Mathématique, Mathématiques et Systèmes, 35 rue St Honoré 77305-Fontainebleau-CEDEX, France*

---

## Abstract

In this paper we present an approach for automatic analysis of urban accessibility using 3D point clouds. Our approach is based on range images and it consists in two main steps: urban objects segmentation and curbs detection. Both of them are required for accessibility diagnosis and itinerary planning.

Our method automatically segments facades and urban objects using two hypothesis: facades are the highest vertical structures in the scene and objects are bumps on the ground on the range image. The segmentation result is used to build an urban obstacle map. After that, the gradient is computed on the ground range image. Curb candidates are selected using height and geodesic features. Then, nearby curbs are reconnected using Bézier curves. Finally, accessibility is defined based on geometrical features and accessibility standards.

Our methodology is tested on two MLS databases from Paris (France) and Enschede (The Netherlands). Our experiments show that our method has good detection rates, is fast and presents few false alarms. Our method overcomes other works reported in the literature on the same databases.

*Keywords:* Accessibility, soft-mobility, mathematical morphology, curbs, urban modeling, mobile laser scanning

---

*Email addresses:* [andres.serna\\_morales@mines-paristech.fr](mailto:andres.serna_morales@mines-paristech.fr) (Andrés Serna),  
[beatriz.marcotegui@mines-paristech.fr](mailto:beatriz.marcotegui@mines-paristech.fr) (Beatriz Marcotegui)

*Preprint submitted to ISPRS Journal of Photogrammetry and Remote Sensing April 26, 2013*

## 1. Introduction

Digital 3D city models are useful for many applications: urban planning, emergency response simulation, cultural heritage documentation, virtual tourism, itinerary planning, accessibility analysis for different types of mobility, among others. Thanks to new 3D data availability, an increasing number of geographic applications such as Google Earth, Microsoft Virtual Earth, Open Street Maps and Geoportail are flourishing nowadays. Most of these applications are recently enhanced with pedestrian navigation options. Some of these applications do not only require to look realistic, but also have to be faithful to reality. Thus, semantic analysis from real data (images, 3D point clouds, GPS measurements, etc.) are performed in order to give faithfulness to 3D city models. This analysis is usually made by manual assisted approaches, leading to time consuming procedures, unsuitable for large scale process. In that sense, automatic methods for semantic analysis of urban structures are required.

In general, large cities are built based on demographical and geographical constraints, architectural preferences and governmental budgets. Many places are not accessible for wheelchairs, skaters, segways, baby buggies, among others. Taking a short trip through the street can reveal an environment plagued with physical barriers, where wheelchair users bear the brunt.

About 80 million people living in European Union (EU) have a mild to severe disability. Physical obstacles make them vulnerable to social exclusion, low employment and limited education level. In fact, mean poverty rate for those with disabilities is 70% higher than the average. In 2007, EU signed the *United Nations convention on the rights of persons with disabilities* (UN, 2007). The aim is allowing people with disabilities to go on their daily lives like everyone else and enjoy their rights as EU citizens. One of the strategies consists in ensuring physical access to buildings, roads, transportation, schools, housing, medical centers and workplaces.

In France according to Law 2005-102<sup>1</sup>, local authorities are required to make accessibility diagnoses and to take corrective actions in public spaces. An available manual-assisted solution is Wheelmap<sup>2</sup>. It is an on-line service, based on Open Street Maps, aiming to tag wheelchair-accessible places. It is

---

<sup>1</sup>Loi 2005-102 du 11 février 2005: “Pour l’égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées”.

<sup>2</sup><http://wheelmap.org/en/>



a crowd-sourcing project where everyone can collaborate by tagging public places according to accessibility for wheelchair users. Other solutions could include automatic 3D urban modeling techniques (Golovinskiy et al., 2009; Hernández and Marcotegui, 2009b; Pu et al., 2011; Douillard et al., 2011; Rutzinger et al., 2011). However, automatic accessibility analysis is still an open problem.

In general, urban accessibility information can be integrated into navigation services as on-line maps, support systems using cell phones (Rashid et al., 2010), collaborative social networks (Menkens et al., 2011), or even in automated wheelchairs and segways (García et al., 2010). Including accessibility parameters in city maps, one can define adaptive itineraries according to detailed and accurate information about barriers and obstacles in the public infrastructure.

Our work is part of Cap Digital Business Cluster TerraMobilita project: “3D mapping of roads and urban public space, accessibility and soft-mobility”. This project responds precisely to requests about 3D urban maps and soft-mobility applications. Contribution is twofold: i) To develop new methods and tools in order to create and update urban maps using laser scanning and digital imagery. ii) To develop innovative applications for soft-mobility itinerary planning.

The focus of this work is automatic obstacle map creation and analysis of curbs accessibility from Mobile Laser Scanning (MLS) data. Our method begins filtering facades and urban objects not only to center our attention on ground information, but also to build 3D obstacle maps useful for itinerary planning. Then, automatic methods for detection, reconnection and characterization of curbs are developed. For our experiments, MLS data are acquired by Stereopolis II (Paparoditis et al., 2012) and LARA-3D2 (Goulette et al., 2006), two MLS systems from IGN France and MINES ParisTech, respectively. Additionally, a public database from Enschede (The Netherlands) is used to get quantitative results and to compare with the state of the art.

The paper is organized as follows. Section 2 reviews some related works in the state of the art and establishes their differences with respect to our work. Section 3 describes our method using range images, explains the accessibility analysis and presents an illustrative itinerary planning application. Section 4 presents comparative results with other database available in the literature. Finally, Section 5 is devoted to conclude this work.

## 2. Related work

Ayres and Kelkar (2006) present a method to characterize sidewalks based on ground elevation profiles. The aim is identifying unusual elevation changes and obstacles. The authors highlight the impact of elevation changes on pedestrians and soft-mobility users safety. They work with very precise data from streets in California. Unfortunately, this is not suitable for large scale applications since data are acquired manually.

Hernández and Marcotegui (2009a) use range images from 3D point clouds in order to extract quasi-flat zones on the ground and use them as markers for a constrained watershed (Beucher and Meyer, 1993). Then, a region adjacency graph is used to determine the border between road and sidewalk. This procedure fails in presence of access ramps because quasi-flat zones merge road and sidewalk and it is no longer possible to detect curbs.

Vosselman and Zhou (2009) use airborne laser scanning data to detect curbstones. First, small height jumps at ground level are detected. Second, they fit a smooth curve to generate the separation between sidewalk and road. Finally, small gaps between nearby and collinear segments are closed. In unoccluded regions, this approach is robust to traffic signs, cars and multi-line roads. Good results are presented with respect to ground truth measurements. However, airborne laser resolution ( $20 \text{ points}/m^2$ ) is not precise enough to detect access ramps and low curbs. In fact, only curbstones of at least 10cm height can be clearly detected. More recently, the same authors (Zhou and Vosselman, 2012) extended their methodology to denser ( $1000 \text{ points}/m^2$ ) MLS datasets. They improve the planimetric accuracy of curbstone locations fitting a sigmoidal function to detected points, in a similar way to Siegemund et al. (2010). In these methods, although detection is performed directly on the 3D point cloud, it is made on a strip by strip basis, so intrinsic information between neighboring strips is missing.

Valero et al. (2010) present an automatic road extraction methodology for high resolution imagery. This procedure can be extrapolated to curb detection since both problems have similar assumptions: they are thin and elongated paths, but not necessary straight, and present color differences with respect to their neighborhood. Their experimental results show accurate road extractions in terms of completeness and correctness. Some post processing has to be proposed in order to reconnect isolated segments due to occlusions and shadows.

Gang and Guangshun (2010) propose an approach to model urban road

networks based on manual markers. They use an interactive interface to mark sidewalks and roads on aerial images. Then, Bézier curves and polygons are used to model the road. This method is realistic and very fast to render, however manual marking is a time-consuming task. Automatic detections and/or road network databases are needed for large scale modeling.

With respect to other works reviewed in the state of the art, we aim at developing a method of urban accessibility diagnosis suitable for large scale applications. Our method is automatic using few constraints on MLS data, creates an obstacle map segmenting objects on ground, defines the accessibility for each curb point, and can manage reconnection problems due to access ramps. A detailed description is presented below.

### 3. Methodology

Figure 1 shows our proposed methodology. First, input point cloud is mapped to range images. Second, the image is interpolated in order to avoid connectivity problems and the quasi-flat zones algorithm is used to segment the ground (road+sidewalk). Third, facades and objects are extracted using morphological transformations and the obstacle map is defined. Then, curb candidates are selected using height and elongation criteria, and close curbs are reconnected using Bézier curves. Curbs accessibility is defined according to international standards. Finally, the obstacle map is used to define adaptive itineraries taking into account the path width such that a wheelchair can pass.

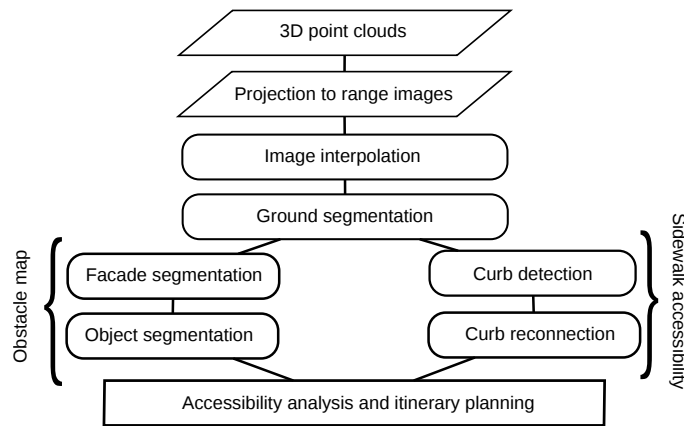


Figure 1: Workflow of our proposed accessibility analysis from MLS data

### 3.1. Projection to range images

Our method projects 3D point clouds to range images because they are convenient structures to visualize and process data. One can utilize all the large collection of existing image processing tools, in particular mathematical morphology (Serra, 1988; Soille, 2003). An image can be processed quickly, implicitly defines neighborhood relationships and requires little memory.

A range image is a 2.5D structure that contains height information at each pixel. Range images are generated projecting 3D points to a  $XY$ -plane using a virtual camera located on the horizontal plane with normal vector  $\vec{n} = (0, 0, 1)$ , and crossing the lowest point in the  $Z$ -coordinate axis  $(0, 0, z_{min})$ . In general, several points are projected on the same pixel. Thus, we define two images:

- Maximal range image (or simply range image): stores the maximal depth among all projected points on the same pixel.
- Minimal range image: stores the minimal depth among all projected points on the same pixel.

These range images are used to segment facades, objects and curbs. When this process is completed, images are reprojected to the 3D point cloud in order to get the final result.

Spatial pixel size is the only projection parameter and it has to be carefully chosen. On one hand, if the pixel size is too large, fine details are not preserved because too many points would be projected on the same pixel. On the other hand, too small pixel size implies connectivity problems and larger image sizes, which would no longer justify the use of range images. To avoid connectivity problems and loss of information, spatial pixel size is chosen according to point cloud resolution. In our experiments, a pixel corresponds to a square of 10 cm side ( $0.01 m^2$  area). Figure 2 describes the process using range images. A detailed explanation is presented in the following subsections.

### 3.2. Image interpolation

On range images, an interpolation is required in order to fill holes caused by occlusions and missing scan lines. A morphological interpolation based on filling holes is preferred since this transformation does not create new regional maxima, it can fill holes of any size and no parameters are required. This

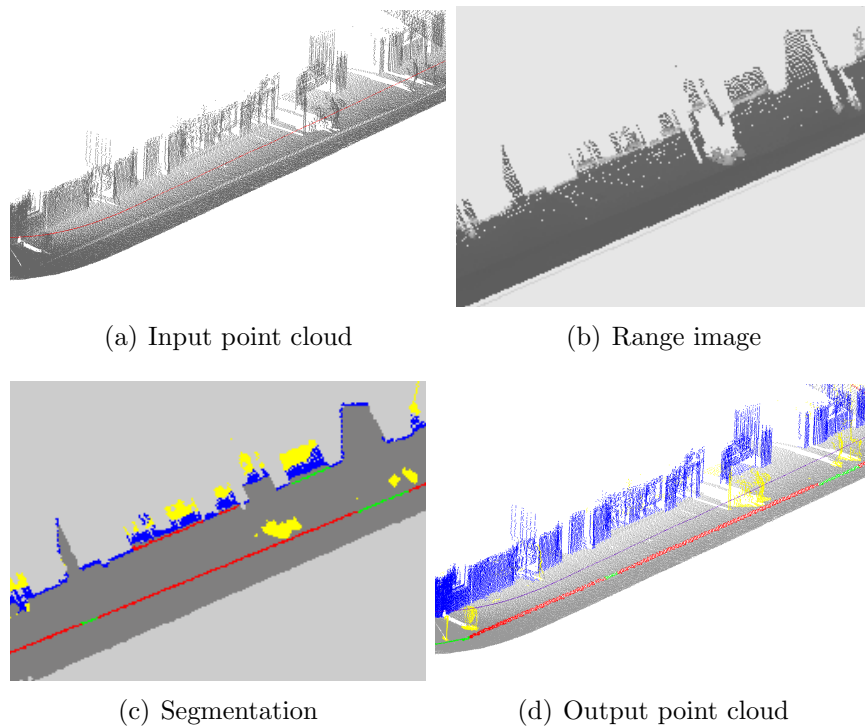


Figure 2: 3D point cloud processing using range images. Segmentation results: ground (gray), curbs (red and green), facade (blue), objects (yellow). Note that some objects are detected behind facade because they have been seen through windows and doors.

is important in order to avoid false alarms in the object detection approach (explained in Section 3.5).

In the most simple sense, a hole is a dark region which is not connected to the image border and is surrounded by brighter pixels. The fill-holes transformation is implemented as the reconstruction by erosion of the image  $f$  from the marker  $f_{marker}$ , as shown in Figure 3. The marker  $f_{marker}$  is set to the maximum image value except along the image border where the original image value is kept. Figure 3(b) illustrates this definition on 1D. Note that a minimum in the left part of the signal is not a hole because it touches the border. In order to preserve original data, only pixels with no data are modified, while other pixels keep their original value. For further details on grayscale reconstruction operators, the reader must refer to (Soille and Ansault, 1990; Vincent, 1993)

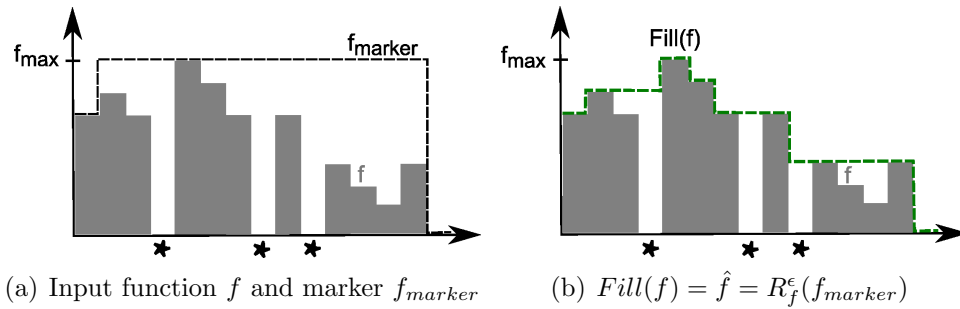


Figure 3: Fill-holes transformation: Morphological reconstruction by erosion of  $f$  from  $f_{marker}$ .  $\star$  indicates no-data points

Figure 4(a) exhibits an experimental scenario in Saint Sulpice Place at Paris. Figure 4(b) presents the range image, where black pixels indicate no data. In the middle of the place, sparse points are obtained as the result of occlusions and faraway objects. This image has to be interpolated before processing. Note that almost all dark regions are touching the image border, so they would not be filled by a classical fill-holes transformation. To solve this problem, each isolated region is connected to its closest neighbor by the shortest path. This artificially defines the boundaries of Figure 4(c). Then, the fill-holes transformation is applied in order to interpolate the image (Figure 4(d)). Note that our methodology performs well on near objects. However, several false artifacts can appear when interpolating far objects because there are not enough points. This can be easily corrected in the segmentation step eliminating objects for which the number of interpolated points ( $N_{interp}$ ) is much higher than the number of real points ( $N_{real}$ ). That leads us to define a confidence index  $C = N_{real}/(N_{real} + N_{interp})$  for each segmented object. The distance of an object to the acquisition system could also be considered.

### 3.3. Ground segmentation

Ground segmentation is a critical step for access diagnosis since curbs and obstacles are located on it. With the aim of segmenting the ground, we use the  $\lambda$ -flat zones labeling algorithm, firstly introduced in image processing by Nagao et al. (1979), defined by Meyer (1998) as:

**Definition 1.** *Two neighboring pixels  $p, q$  belong to the same quasi-flat zone*

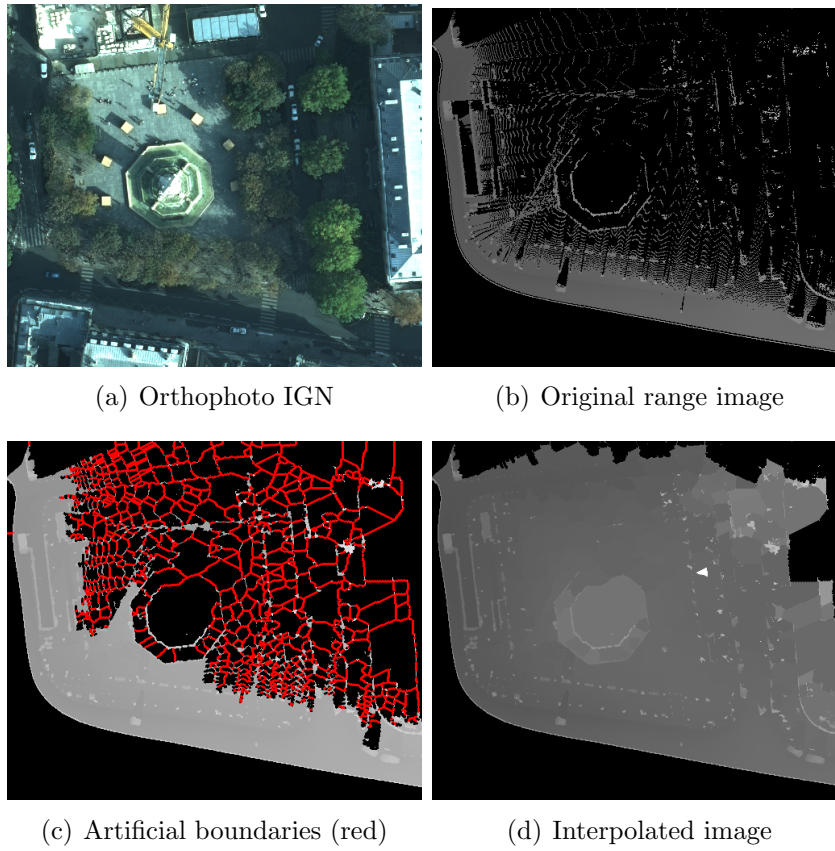


Figure 4: Image interpolation. Artificial boundaries are created for regions touching the border and then the fill-holes transformation is applied.

of a function  $f$ , if their difference  $|f_p - f_q|$  is smaller than or equal to a given  $\lambda$  value.

$$\forall (p, q) \text{ neighbors} : |f_p - f_q| \leq \lambda \quad (1)$$

With this definition, we want to obtain the ground as the largest  $\lambda$ -flat zone on the range image. We set  $\lambda = 20\text{cm}$  because it is usually high enough to merge road and sidewalk (even if there are no access ramps) without merging objects. Note that this approach is strongly inspired by the work of Hernández and Marcotegui (2009a). In the context of that work, their procedure fails in presence of access ramps because road and sidewalk are

merged. However, in our work, we take advantage of this to extract the complete ground mask filtering out facades and objects.

#### 3.4. Facade segmentation

Once the ground is extracted, all remaining structures are considered as facades and objects. Discrimination between them is important because facades delimit the end of public space, and urban objects define the obstacle map required for itinerary planning.

Due to specific requirements in our Paris database, the MLS system was oriented to the ground. Therefore, structures higher than 2.5m are out of the laser field of view. This is a challenge for methods using height constraints (Hernández and Marcotegui, 2009a), since high wall parts are not visible. To solve this problem, we propose a solution extracting facade markers. For that, we take advantage of the acquisition cycle of the MLS sensor. In our configuration, the sensor scans vertical lines starting from the top. Thus, the first point corresponds to the highest point of that cycle. Each first point is labeled as facade marker candidate. Isolated and faraway points are not considered. Then, points forming lines are taken as facade markers.

In order to reconstruct all the facade, an attribute controlled reconstruction is performed from facade markers (Serna and Marcotegui, 2013). It consists in appending nearby points with similar height until the maximization of an attribute on the segmented region. In our case, facades are the longest and most elongated structures in the range image. Then, the maximal geodesic elongation is used.

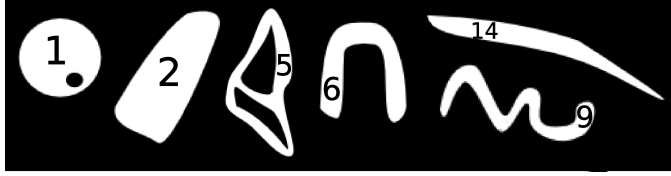
The geodesic elongation  $E(X)$ , introduced by Lantuéjoul and Maisonneuve (1984), is computed to only select thin and elongated structures. Equation 2 defines the geodesic elongation of an object  $X$ , where  $S(X)$  is the area and  $L(X)$  is the geodesic diameter (Lantuéjoul and Beucher, 1981). A toy example of geodesic elongation for different binary objects is shown in Figure 5(a). The number on each object corresponds to its elongation. A fast computation of this geodesic attribute can be found in (Morard et al., 2011).

Using this controlled reconstruction maximizing the geodesic elongation, it is possible to reconstruct the facade without merging adjacent objects.

#### 3.5. Object segmentation

To segment urban objects, the methodology proposed by Hernández and Marcotegui (2009a) is used. We assume urban objects appear as bumps on the ground in the range image. Then, they are extracted using the top-hat





$$E(X) = \frac{\pi L^2(X)}{4S(X)} \quad (2)$$

Figure 5: Geodesic elongation for different binary objects.

transformation by filling holes:  $FTH(\hat{f}) = Fill(\hat{f}) - \hat{f}$ . First, the range image is inverted and its holes are filled. Then, the top-hat transformation consists in subtracting the original image from the filled image (see Soille, 2003, Chapter 4.5).

Figure 6 presents the segmentation of facade and objects in a 3D point cloud. Note that pedestrians touching the facade are correctly segmented. By the moment, we assume all objects static. However, classification techniques can be used in order to distinguish mobile objects such as pedestrians from static objects. Additionally, using velodyne data and color images can help to improve this process.

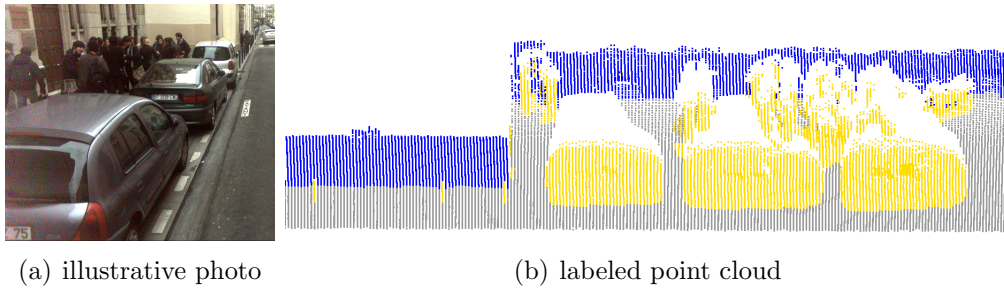


Figure 6: Segmentation of facade (blue) and objects (yellow). Note that pedestrians touching the facade are well discriminated.

### 3.6. Curb detection

In order to detect curbs, we use the interpolated minimal range image because it contains the lowest projected point on each pixel. Next, the morphological external gradient is computed as the arithmetic difference between the dilation and the original image  $\rho_{sup} = \delta_B(f) - f$ , where  $B$  is a hexagonal structuring element of size 0.1m (1 pixel).

Regions with range changes between 3cm and 20cm are considered as curb candidates. Then, a minimum elongation  $E_{min}=10$  is experimentally defined in order to accept curbs. This threshold corresponds to the geodesic elongation of a curb of approximately 1m long and 0.08m wide. We prefer geodesic measurements because, in general, curbs are not straight structures, so the Euclidean distance could sub-estimate their real length.

In Figure 7, we illustrate the effect of several  $E_{min}$  thresholds for curbs detection. Note that  $E_{min}=0$  preserves all structures between 3 and 20cm height,  $E_{min}=5$  does not take noise away and  $E_{min}=20$  removes some real curbs, those that are short due to occlusions. Note that steps at building entrances are considered as curbs and its detection can be used to define building accessibility. In the case that they should not be considered, a constraint of minimal distance from the facade can be imposed.

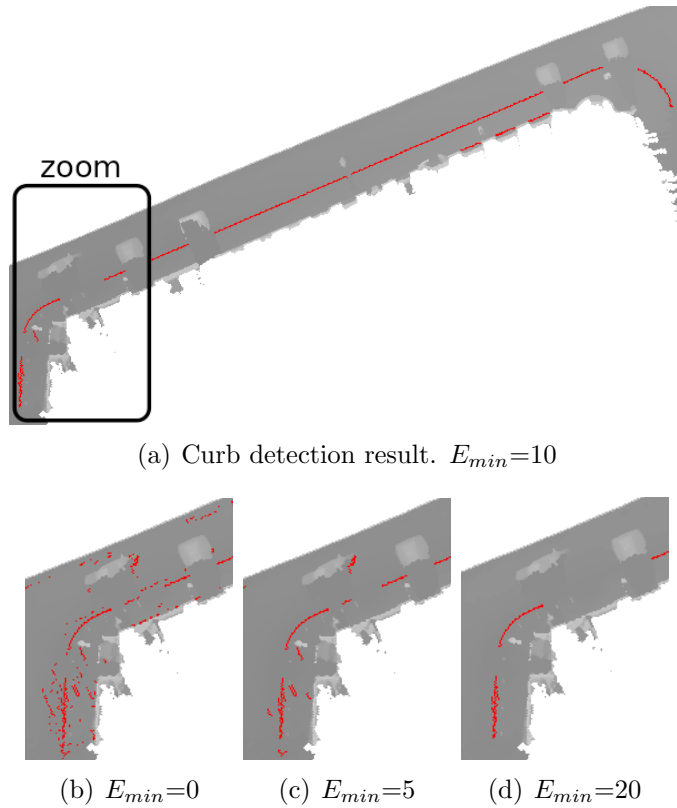


Figure 7: Curb detection at different thresholds on the geodesic elongation.

### 3.7. Curb reconnection

The main drawback of the previous process is the lack of connectivity between curbs due to access ramps, occlusions, missing scan lines and acquisition problems. Some solutions can be found in the literature: Zhou and Vosselman (2012) close gaps between adjacent and co-linear curbs using lines, Shih and Cheng (2004) present an approach based on adaptive mathematical morphology for linking broken edges, and Talbot and Appleton (2007) propose a sophisticated solution incorporating incomplete path openings. Unfortunately, these solutions are not suitable since reconnection through access ramps are not always straight (Figure 9(a)) and height discontinuities of access ramps are close to the noise level (Figure 9(c)).

We propose a reconnection strategy based on quadratic Bézier curves. Two curbs closer than a threshold  $d_{min}$  are reconnected tracing a Bézier curve between their geodesic extremities. Curb orientations are used in order to define the Bézier parameters, as explained below. In our experiments,  $d_{min} = 5\text{m}$  for Paris database.

A Bézier curve is a parametric path traced by the function  $B(t)$ , given points  $P_0$ ,  $P_1$ , and  $P_2$ , as shown in Equation 3. It departs from  $P_0$  towards  $P_1$ , then bends to arrive to  $P_2$ . As a consequence, tangent lines in  $P_0$  and  $P_2$  both pass through  $P_1$ . Thus, the user can control input and output angles of the curve. This is an important smooth constraint, because in our application, initial and final angles of the reconnection should not change abruptly.

$$B(t) = (1 - t)^2 P_0 + 2(1 - t)t P_1 + t^2 P_2, \forall t \in [0, 1] \quad (3)$$

Conveniently, the reconnection process can be written as the problem to find the three control points for a Bézier curve. Points  $P_0$  and  $P_2$  correspond to the geodesic extrema of curbs  $C_0$  and  $C_2$ . Thus, the problem is reduced to find  $P_1$ . In general, there are two reconnection types:

- If the curbs to be reconnected are co-linear,  $P_1$  is put in the middle of them. Therefore, the three control points are co-linear and the resulting reconnection is a straight line, as described in Figure 8(a).
- If the curbs to be reconnected are not co-linear,  $P_1$  is put in the intersection of the two projection lines from  $C_0$  and  $C_2$ . Therefore, the resulting reconnection is a parabolic segment, as described in Figure 8(b).

Figure 9 illustrates this procedure in a real case. Note that the resulting curve is smooth and faithful to reality.

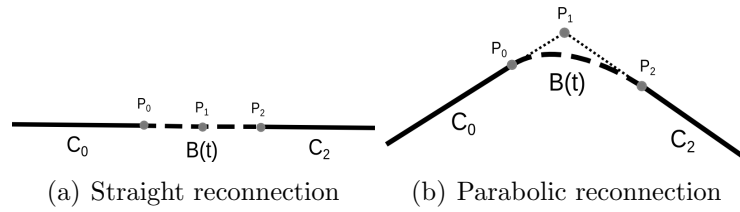


Figure 8: Quadratic Bézier reconnection in straight and bent cases

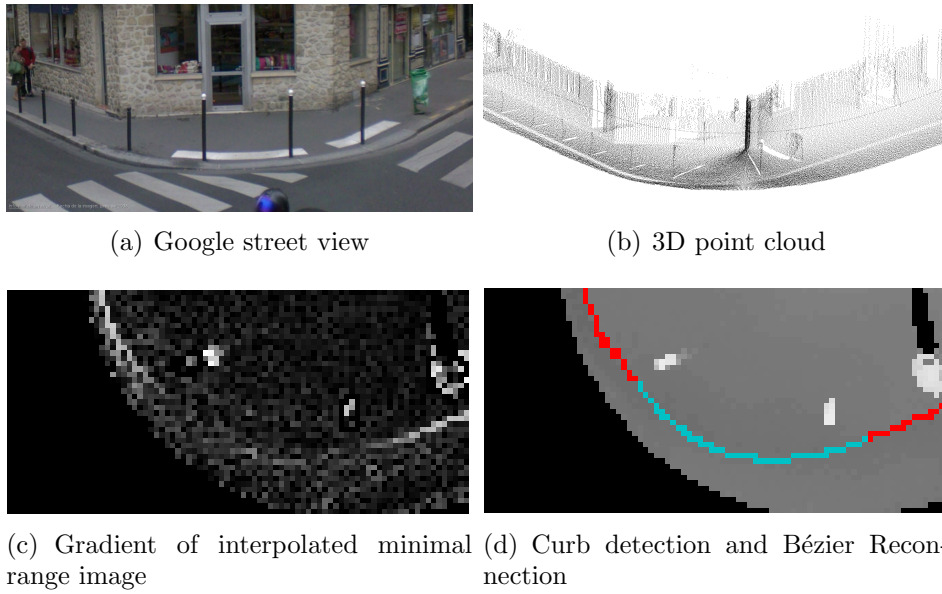


Figure 9: Reconnection of detected curbs (red) with Bézier curves (cyan).

### 3.8. Accessibility analysis and itinerary planning

One of the aims of TerraMobilita project is planning itineraries for different types of mobility, including soft-mobility. Therefore, curb characterization is a very important task because it determines the suitability of a path. For example, sidewalks without access ramps may be appropriate for rollers but not for wheelchairs. Additionally, obstacles on sidewalk represent physical barriers to free mobility. In our work, we define the accessibility according to curbs geometry and obstacles on the street. In our opinion, the most critical case is the accessibility for wheelchair users, so our experiments are conducted in that sense. However, we can define the accessibility depend-

ing on any other type of soft-mobility since our method provides geometrical information of curbs and obstacles for each ground point.

A standard wheelchair is between 60 and 69cm wide, therefore the minimum clear width of an access ramp is 91.5cm between railings (ISO, 2008; ADA, 2010). Thus, curb accessibility is defined taking the following criteria into consideration: i) **wheelchair-accessible (green)**: Sidewalk access with one step maximum, wider than 1 m and not higher than 7cm. ii) **wheelchair-inaccessible (red)**: Otherwise.

This simple traffic-light color code is strongly inspired by international standards and it is compatible with on-line maps such as Wheelmap. Figure 10 illustrates two labeled 3D point clouds.

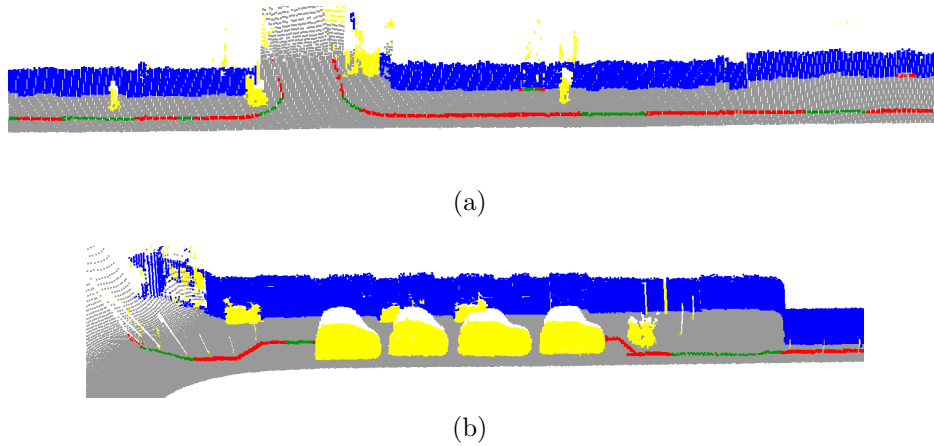


Figure 10: Labeled point clouds: Facades (blue), urban objects (yellow), ground (gray), inaccessible curb (red), accessible curb (green).

A direct application consists in planing adaptive itineraries for different types of mobility. For example, defining the start and final points of a journey, it is possible to suggest an adaptive itinerary according to obstacles on the ground and curb accessibility. Thus, the problem consists in finding a path that optimizes certain criteria (i.e., the shortest path). Figure 11 presents an example of an adaptive itinerary for a person using a wheelchair going from A to B. In this case, we assume a minimum passing space of 1m, which is large enough for a standard wheelchair. Note that this example is only illustrative, real applications for itinerary planing will be developed in the framework of TerraMobilita project.

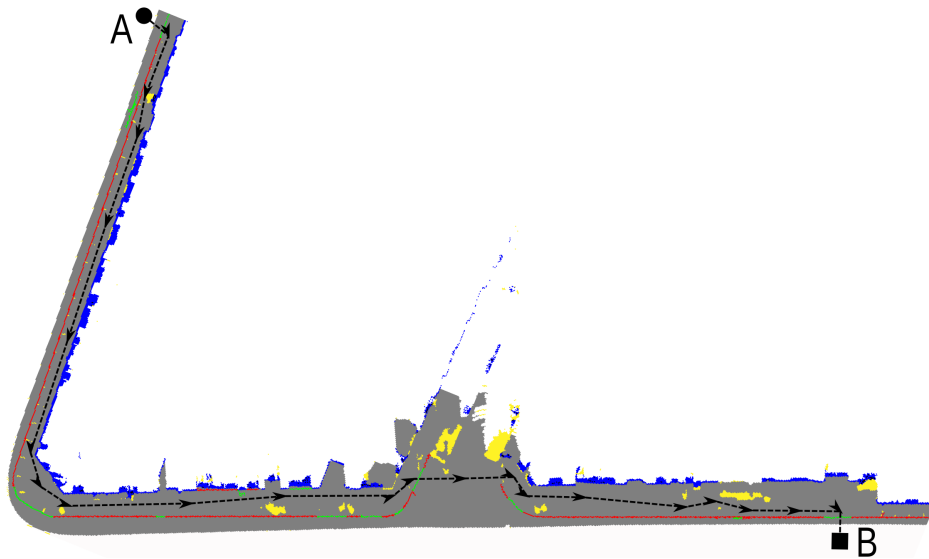


Figure 11: Adaptive itinerary (black) for a person using a wheelchair going from A to B.

#### 4. Results

Our methodology was tested on our own database in order to get qualitative results. It contains approximately 2km ( $\sim 10$  million points) of MLS data from the 6<sup>th</sup> district of Paris, France. All examples aforementioned correspond to this database. By the moment, we do not have ground truth annotations, so quantitative results are not available yet. However, our experiments demonstrate that facades are correctly segmented even when their high part is out of the laser field of view. Objects are segmented as bumps on the ground and assumed static. Our method presents correct curb detections, few false alarms and appropriate accessibility analysis. The main problem is due to occluded regions, which can be managed if the reconnection distance does not exceed 5m.

In order to get quantitative results, we use another database containing three test sites at Enschede, The Netherlands. The database contains approximately 1km ( $\sim 12.5$  million points) of MLS data with ground truth annotations. Two manual data are collected: i) roadside lines, corresponding to inaccessible curbs higher than 7cm. ii) gap lines, corresponding to access ramps lower than 7cm. We use the same evaluation strategy as Vosselman and Zhou (2009; 2012). Quantitative analysis is performed by comparison between automatic and manual extracted lines. As the amount of false alarms

near real road lines is very low in this database, a buffer around ground truth lines is taken. Automatic lines are labeled as true positives or false positives if they are located inside or outside the buffer, respectively. A buffer width of 50cm was used. Two classical statistics are computed. The completeness (or recall) is defined as the length of the extracted lines inside the buffer divided by the length of the reference lines. The correctness (or precision) is defined as the length of the extracted lines inside the buffer divided by the length of all extracted lines.

Figure 12 illustrates automatic curb detection in the three test sites. Results of completeness, correctness and processing times are given in Table 1. In order to simplify lines geometry, the well-known Douglas and Peucker (1973) algorithm was used with 20cm distance threshold.

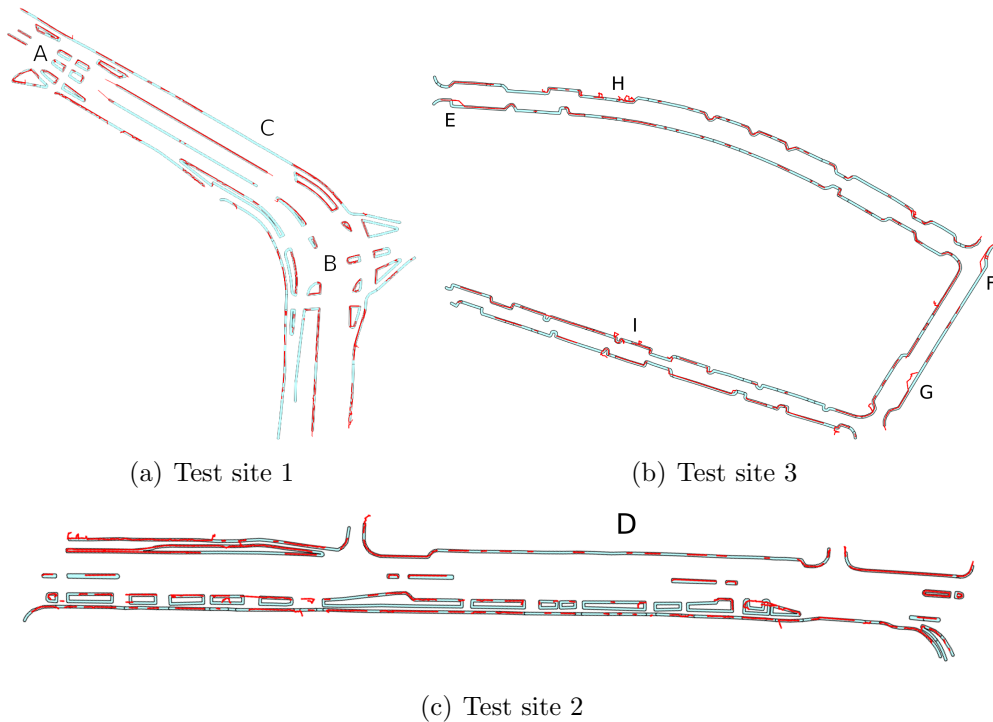


Figure 12: Test sites at Enschede. Our detection (red) and ground truth (cyan).

Our results show that our method has good detection rates, is fast and presents few false alarms. On one hand, completeness in sites 2 and 3 is better than other works reported in the literature on the same database. On

the other hand, correctness is greater than 90% for all sites, which indicates that our method produces few false alarms, which are mainly due to low vegetation (Zones H and I in Figure 12(b)).

Table 1: Completeness, correctness and processing time for three test sites at Enschede, The Netherlands. Between brackets the results obtained by Zhou and Vosselman (2012)

	Site 1	Site 2	Site 3
Completeness	65% (83%)	54% (53%)	60% (54%)
Correctness	95% (91%)	94% (92%)	91% (84%)
Time	5.5 min (1 hour)		

First test site leads to low performances because of polygonal curbstones in the middle of the road (Zones A and B in Figure 12(a)). Since MLS data was acquired only from one side of the street, only one side of the polygons is visible. As our original goal consists in detecting curbs limiting the sidewalk in order to perform accessibility diagnostics, our method does not process polygonal curbstones in any special way, then the invisible part is not detected. The scores published by Zhou and Vosselman (2012) take these polygons into account. Fitting polygons can be a suitable solution and it will be evaluated in our future work.

Another problem in this site is due to long access ramps that cannot be reconnected by our method. For example, zone C in Figure 12(a) shows an access ramp lower than 3cm and 45m long. Therefore, it is neither detected nor reconnected.

The presence of cars and other obstacles is the main problem in the detection procedure. In fact, several curbs are not detected due to large occluded areas. For example, zone D in Figure 12(c) shows a large occluded area due to cars on both sides of the street. Only short curb parts are detected between parked cars and they are not reconnected because the distance exceeds our reconnection threshold. For this database, reconnection threshold has been reduced to 2m due to wrong reconnections in polygonal curbs in the middle of the road. Therefore, curb reconnections longer than 2m are not allowed and it is the reason of low recall in curbs detection.

Inspecting test sites, we found several inconsistent ground truth lines since they do not correspond to real curbs. For example, Figure 13(b) shows a straight detected curb (red) on the right side, while the ground truth (cyan)



marks it as an extrusion. A photo from the scene (Figure 13(a)) demonstrates that automatic detection is correct in this case. Note that this is a Google Street View photo, taken a different day, so parked cars are not the same. Other ground truth problems can be found in zones E, F and G in Figure 12(b).

Note that our method is designed to detect height jumps on the ground, not only curbstones. Therefore, stairs and steps at building entrances are detected as well (Figure 13). These structures are not errors, but they are not marked in ground truth data. Thus, they were not taken into account in the quantitative results in order to do a fair comparison. To automatically filter out these structures, a constraint of minimal distance (50 cm) from the facade was imposed.

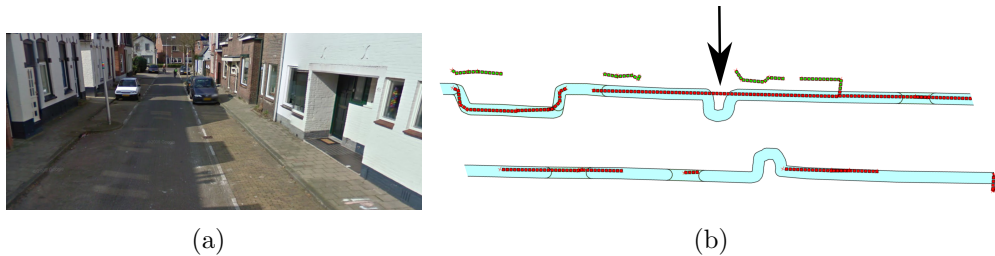


Figure 13: Inconsistent ground truth annotation indicated with an arrow. Ground truth (cyan) presents an extrusion that does not correspond to the real curb. Detected curbs (red) and building entrances (green). Photo taken from Google Street View.

Processing was carried out on an Intel Core i7 CPU @2.93 GHz with 8 GB RAM. Note that our method takes less than 9 minutes to process the three test sites, which is quite faster than any other method running in the same database. For example, Vosselman and Zhou (2009) reported 1 hour for the processing time. Although our machine is faster, the conceptual difference consists in the fact that they process the 3D point cloud on a strip by strip basis, while we project all 3D points to a range image and we process them as a complete set using digital image processing techniques.

For further analysis, Table 2 presents the individual completeness results for each curb type. In general, occlusion affects all detection types. The best recall occurs for roadside lines, which are curbs higher than 7cm. Long curb reconnections are not allowed and it is the reason of low recall in curbs detection. The lowest completeness is due to polygonal curbstones. As

Table 2: Individual completeness for each curb type.

	Site 1	Site 2	Site 3
Roadside (Inaccessible)	82%	67%	64%
Gaps (Accessible)	55%	48%	46%
Polygonal curbstones	46%	45%	N/A

aforementioned, we only detect one side, then an special processing should be performed for this type of curbs.

## 5. Conclusions

We propose an automatic and robust approach to segment facades, urban objects and curbs in 3D point clouds. Processing is carried out using range images and the final result is reprojected onto the 3D point cloud. Facades are used to delimit the public space, while urban objects define the obstacle map required for itinerary planning. To detect curb candidates, gradient information is used. Then, thin and elongated structures are selected using geodesic elongation. Our method is robust to noise since small, isolated and not elongated structures are removed from the range image using morphological filters. Finally, close curbs are reconnected using Bézier curves and characterization is done based on geometrical features and accessibility standards.

Our methodology was qualitatively and quantitatively tested on two MLS databases from Paris (France) and Enschede (The Netherlands). Our results show that our method has good detection rates, is fast and presents few false alarms. In fact, correctness and completeness results are higher than those for other works reported in the literature on the same databases.

Occlusions, long access ramps and polygonal curbstones are the main problems in the detection procedure. In the first case, several scans of the same zone can reduce the occlusion. Using color gradients can be a suitable solution in order to detect low and long access ramps. And, as suggested by other works in the literature, a special strategy for the occluded side of polygonal curbstones should be developed. Other problems were due to inconsistent ground truth annotations.

Our approach is still a research prototype and efforts to be optimized have not been carried out. However, our process is faster than other methods

reported in the literature. In our experiments, we estimate processing speed of 5 min/km. In a city like Paris, with 1700 km of streets, only 2.4 days would be required in order to automatically analyze the accessibility of the whole city. It is reasonable for large scale applications. If the analysis would be implemented in a real time platform, for instance to guide a robot, important optimization and parallelization are possible.

In the future, we are planning to use velodyne and color sensors to distinguish static from mobile obstacles and to help to solve occlusions problems. We expect to include quantitative results from Paris database in future works.

### **Acknowledgements**

The work reported in this paper has been performed as part of Cap Digital Business Cluster TerraMobilita project.

We want to thank Liang Zhou and Prof. Georges Vosselman for providing us with the Enschede database.

### **References**

- ADA, 2010. ADA Standards for Accessible Design. [http://www.ada.gov/2010ADASTandards\\_index.htm](http://www.ada.gov/2010ADASTandards_index.htm).
- Ayres, T., Kelkar, R., 2006. Sidewalk potential trip points: A method for characterizing walkways. *International Journal of Industrial Ergonomics* 36, 1031–1035.
- Beucher, S., Meyer, F., 1993. The morphological approach to segmentation: the watershed transformation, in: Dougherty, E.R. (Ed.), *Mathematical Morphology in Image Processing*. Marcel Dekker, New York. chapter 12, pp. 433–481.
- Douglas, D., Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer* 10, 112–122.
- Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A., 2011. On the segmentation of 3D LIDAR point clouds, in: *IEEE International Conference on Robotics and Automation, ICRA'11*, pp. 2798–2805.

- Gang, L., Guangshun, S., 2010. Procedural Modeling of Urban Road Network, in: International Forum on Information Technology and Applications, IFITA'10, pp. 75–79.
- García, J., Amaral, P., Marrón, M., Mazo, M., Bastos Filho, T., 2010. Proposal for an Ambient Assisted Wheelchair (A2W), in: IEEE International Symposium on Industrial Electronics (ISIE'10), pp. 2325–2330.
- Golovinskiy, A., Kim, V.G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments, in: 12th IEEE International Conference on Computer Vision, pp. 2154–2161.
- Goulette, F., Nashashibi, F., Ammoun, S., Lourceau, C., 2006. An Integrated on-Board Laser Range Sensing System for On-the-Way City and Road Modelling. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34, 3–5.
- Hernández, J., Marcotegui, B., 2009a. Filtering of artifacts and pavement segmentation from mobile LiDAR data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38, 329–333.
- Hernández, J., Marcotegui, B., 2009b. Point Cloud Segmentation towards Urban Ground Modeling, in: Joint Urban Remote Sensing Event, Urban'09, pp. 1–5.
- ISO, 2008. ISO 7176-5: Wheelchairs–Part 5: Determination of dimensions, mass and manoeuvring space. ISO–International Organization for Standardization.
- Lantuéjoul, C., Beucher, S., 1981. On the use of the geodesic metric in image analysis. Journal of Microscopy 121, 39–49.
- Lantuéjoul, C., Maisonneuve, F., 1984. Geodesic methods in quantitative image analysis. Pattern Recognition 17, 177–187.
- Menkens, C., Sussmann, J., Al-Ali, M., Breitsameter, E., Frtunik, J., Nendel, T., Schneiderbauer, T., 2011. EasyWheel - A Mobile Social Navigation and Support System for Wheelchair Users, in: 8th International Conference on Information Technology: New Generations ITNG'11, pp. 859–866.

- Meyer, F., 1998. From connected operators to levelings, in: Heijmans, H., Roerdink, J. (Eds.), *Mathematical Morphology and its Applications to Image and Signal Processing*, Kluwer Academic Publishers. pp. 191–198.
- Morard, V., Decencière, E., Dokladal, P., 2011. Geodesic attributes thinnings and thickenings, in: *10th International Conference on Mathematical Morphology and its Applications to Image and Signal Processing ISMM'11*, pp. 200–211.
- Nagao, M., Matsuyama, T., Ikeda, Y., 1979. Region extraction and shape analysis in aerial photographs. *Computer Graphics and Image Processing* 10, 195–223.
- Paparoditis, N., Papelard, J.P., Cannelle, B., Devaux, A., Soheilian, B., David, N., Houzay, E., 2012. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. *Revue Française de Photogrammétrie et de Télédétection* 200, 69–79.
- Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing* 66, S28–S39.
- Rashid, O., Dunabr, A., Fisher, S., Rutherford, J., 2010. Users Helping Users: User Generated Content to Assist Wheelchair Users in an Urban Environment, in: *9th International Conference on Mobile Business and Global Mobility Roundtable ICMB-GMR'10* , pp. 213–219.
- Rutzinger, M., Pratihast, A.K., Oude Elberink, S.J., Vosselman, G., 2011. Tree modelling from mobile laser scanning data-sets. *The Photogrammetric Record* 26, 361–372.
- Serna, A., Marcotegui, B., 2013. Attribute controlled reconstruction and adaptive mathematical morphology, in: *11th International Symposium on Mathematical Morphology ISMM'13*, pp. 207–218.
- Serra, J., 1988. *Image Analysis and Mathematical Morphology*. volume 2. Academic Press, London.
- Shih, F.Y., Cheng, S., 2004. Adaptive mathematical morphology for edge linking. *Information Sciences* 167, 9–21.

- Siegemund, J., Pfeiffer, D., Franke, U., Förstner, W., 2010. Curb Reconstruction using Conditional Random Fields, in: IEEE Intelligent Vehicles Symposium, IEEE Computer Society. pp. 203–210.
- Soille, P., 2003. Morphological Image Analysis: Principles and Applications. Springer-Verlag, Secaucus, NJ, USA.
- Soille, P.J., Ansoult, M.M., 1990. Automated basin delineation from digital elevation models using mathematical morphology. *Signal Processing* 20, 171–182.
- Talbot, H., Appleton, B., 2007. Efficient complete and incomplete path openings and closings. *Image and Vision Computing* 25, 416–425.
- UN, 2007. United Nations Convention on the Rights of Persons with disabilities. <http://www.un.org/disabilities/convention/conventionfull.shtml>.
- Valero, S., Chanussot, J., Benediktsson, J.A., Talbot, H., Waske, B., 2010. Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Letters* 31, 1120–1127.
- Vincent, L., 1993. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Transactions on Image Processing* 2, 176–201.
- Vosselman, G., Zhou, L., 2009. Detection of curbstones in airborne laser scanning data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38, 111–116.
- Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation* 18, 293–304.