



HAL
open science

Outils pour la reconstruction automatique de bâtiments à partir d'imagerie aérienne

Christophe Vestri

► **To cite this version:**

Christophe Vestri. Outils pour la reconstruction automatique de bâtiments à partir d'imagerie aérienne. Autre. Université Nice Sophia Antipolis, 2000. Français. NNT: . tel-00346358

HAL Id: tel-00346358

<https://theses.hal.science/tel-00346358>

Submitted on 11 Dec 2008

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

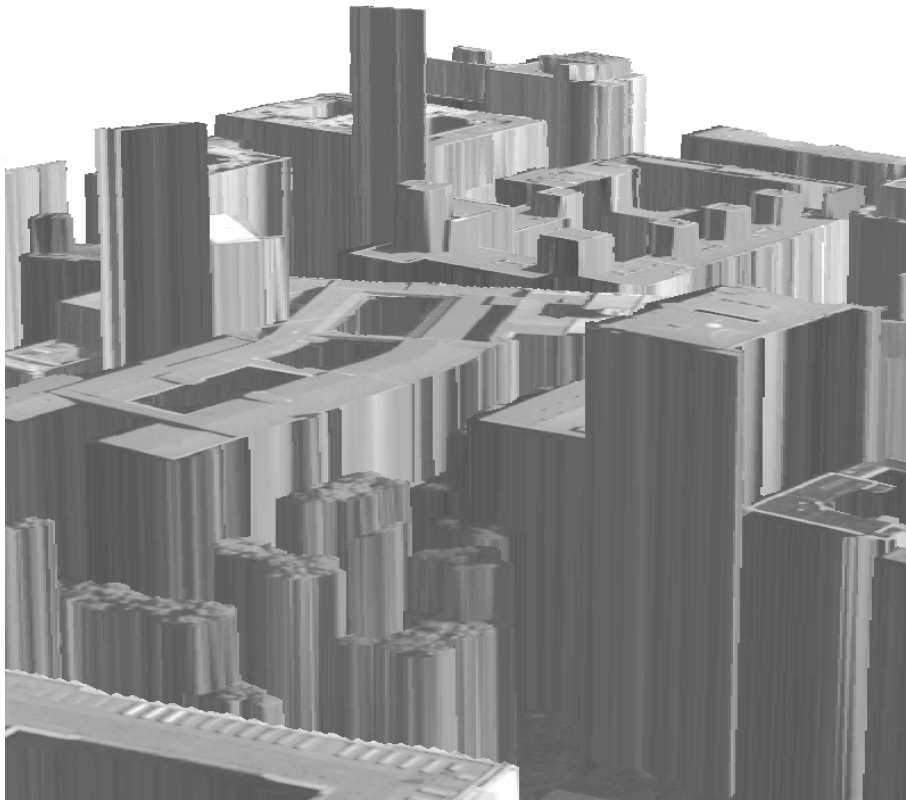
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse présentée pour obtenir le grade de
DOCTEUR EN SCIENCE DE L'INGENIEUR

spécialité
INFORMATIQUE

par
Christophe VESTRI

Outils pour la reconstruction automatique de bâtiments à partir d'imagerie aérienne



Soutenue le 15 septembre 2000 devant le jury composé de :

M. Rachid DERICHE	Président
M. Olivier FAUGERAS	Directeur de thèse
M. Henri MAÎTRE	Rapporteurs
M. Hervé LE MEN	
M. Philippe NONIN	Examineurs
M. Pascal FUA	
M. Frédéric DEVERNAY	Membre invité

A celle qui m'a tout donné,
ma femme

Je remercie Olivier Faugeras qui a dirigé cette thèse, m'a accueilli dans son laboratoire et m'a fait bénéficier de sa confiance et de ses conseils.

Je remercie Laurent Renouard et Philippe Nonin qui m'ont accueilli au sein de l'équipe de Recherche d'ISTAR.

Je remercie Henri Maître et Hervé Le Men qui ont eu la gentillesse d'accepter d'être rapporteurs de cette thèse, avec le travail et le temps que cela représente.

Je remercie Rachid Deriche de m'avoir fait l'honneur de présider mon jury de thèse, ainsi que Pascal Fua et Frédéric Devernay pour y avoir participé.

Je tiens à exprimer un remerciement spécial à Frédéric Devernay pour son encadrement, ses conseils, son soutien, son aide, mais aussi et surtout pour son amitié. Je remercie également ma femme, Virginie, qui m'a accompagné et encouragé par tous les moyens possibles du début jusqu'à la fin de ma thèse. Un grand merci à mes correcteurs : Marie-joëlle et Virginie principalement, mais aussi Frédéric (il est partout celui là), Stéphane, Philippe et Christophe. Je remercie aussi mes parents, ma soeur, Laurent et toute ma famille. Je remercie mes amis, Laurent, Fred et Catherine, Eric et Anne, Emmanuel, la famille Guillermin-colin, et la famille Bachellier. Ensuite dans le désordre, je remercie Marion, Stéphane, Bruno, Marie-Jo, TOFF, Franck, Gilles, Phil, Benoît, l'équipe système, le dev, Christian, Seb, et les autres, l'équipe de foot d'ISTAR et ses supporters, mais aussi ceux de la prod, de la HR, le mkt, et tous les autres d'ISTAR, les robotvissiens avec toujours dans le désordre, Jose, Fred A., Sylvain, Théo, Pierre, Barbare, Jack, et les autres et toutes les personnes d'ISTAR et de l'INRIA.

Table des matières

1	Introduction	1
2	Des images à la troisième dimension	5
2.1	Retrouver la troisième dimension	5
2.1.1	Modélisation de la caméra	6
2.1.2	Système de caméras	11
2.1.3	Calibrage du système	12
2.1.4	Rectification de deux images	13
2.1.5	Disparité et espaces de disparité	14
2.1.6	Reconstruction de la scène	16
2.2	Stéréoscopie	16
2.2.1	Un exemple, le système binoculaire humain	16
2.2.2	Complications pour résoudre la mise en correspondance	17
2.2.3	Contraintes générales	19
2.3	Les principales approches de la stéréoscopie	20
2.3.1	Les méthodes surfaciques	21
	Méthodes surfaciques locales	21
	Autres approches par surfaces	23
2.3.2	Les méthodes à base de primitives	24
2.3.3	Les approches énergétiques	26
2.3.4	Quelques concepts généraux	28
2.4	Conclusion	30
3	Reconstruction de bâtiments	31
3.1	Contexte	31
3.2	De l'acquisition aux modèles numériques	32
3.3	La reconstruction des bâtiments, un problème complexe	34
3.4	Un état de l'art	38
3.4.1	Projets et universités américains	39
	Les différents projets de recherche américains	39
	Les universités ayant participé aux projets	40
3.4.2	Une revue des travaux sur les bâtiments, triée par laboratoire	43
3.5	Bilan de l'état de l'art	51
3.6	Stratégie générale proposée	52
3.7	Liens	53

4	Reconstruction de façades	55
4.1	Problèmes liés aux façades en stéréoscopie aérienne	55
4.2	Présentation géométrique du problème	58
4.3	Étude du problème de transformation des images	62
4.3.1	Étude du capteur linéaire	62
	Mise en équation	63
	Étude de la transformation à appliquer aux images	63
	Problème de rotation des bâtiments	66
4.3.2	Étude générale d'un capteur projectif 3D	66
	Projection des façades 3D dans les images	67
	Transformation des façades projetées	69
	Solutions pour un cas particulier du capteur sténopé	70
	Linéarisation du capteur sténopé	72
4.4	Reconstruction des points des façades	75
4.4.1	Résultats de la mise en correspondance	76
4.4.2	Cartes d'accumulation	77
4.4.3	Étude de la localisation	77
4.4.4	Sensibilité de la reconstruction	77
4.5	Un algorithme de reconstruction automatique de façades	78
4.5.1	Présentation générale de l'algorithme	79
4.5.2	Détection des bâtiments	80
4.5.3	Choix des images pour une orientation donnée	80
4.5.4	Construction des nouvelles images et des façades	81
4.5.5	Reconstruction des points des façades	81
4.5.6	Filtrage des points 3D	81
	Orientation de la surface	82
	Utilisation d'une carte d'accumulation	82
	Filtrage 3D	82
	Correction du MNE	83
4.5.7	Résultats de reconstruction automatique	83
4.5.8	Résultats de la correction des MNE	85
4.6	Conclusion	89
5	Segmentation d'un Modèle Numérique de Terrain	91
5.1	Estimation de modèles de surface	91
5.1.1	Estimation des paramètres d'un modèle de surface	92
5.1.2	Les estimateurs robustes	93
5.2	Reconstruction de surfaces à partir de données 3D	94
5.2.1	Choix de la stratégie	95
5.3	Les estimateurs robustes	96
5.3.1	Les M-Estimateurs	96
5.3.2	Case Deletion Diagnostic	98
5.3.3	Échantillonnage aléatoire	99
	Méthode de la moindre médiane des carrés	99
	MUSE: Minimum Unbiased Scale Estimator	101
	Le RANSAC: RANdom SAmple Consensus	102
	Le MINPRAN: MINimize the Probability of RANdomness	103
5.3.4	La transformée de <i>Hough</i>	104

5.4	Sélection de modèles	104
5.4.1	Sélection de modèles pour la segmentation de données tridimensionnelles	105
5.4.2	Les différents critères et mesures de qualité	105
5.5	L'algorithme de segmentation de MNE	106
5.5.1	Stratégie générale de la segmentation	106
5.5.2	Exploration des données	108
5.5.3	Fusion des hypothèses	109
5.5.4	Sélection des hypothèses	110
	Les fonctions de longueur de description	110
	Le problème d'optimisation booléenne	111
	Optimisation par recherche Tabou	112
5.5.5	Résultats de segmentation	116
5.6	Conclusion	117
6	Vectorisation	119
6.1	Modélisation du bâtiment	119
6.1.1	La construction des chaînes polygonales du modèle	120
	Approximation polygonale d'une chaîne de points	120
	Les critères d'une bonne approximation polygonale	120
6.1.2	L'algorithme de construction du modèle polygonal du bâtiment	121
	Pré-traitement	121
	Approximation polygonale des chaînes	122
	Traitement des jonctions	124
	Résultat de la polygonalisation	127
6.2	Les méthodes de correction des contours	127
6.2.1	Les modèles déformables	128
6.2.2	Les contours actifs	129
	Snakes polygonaux	130
	Pondération multi-image	132
	Résultats des snakes polygonaux multi-image	132
6.2.3	Méthodes exploratoires	135
	Recalage par programmation dynamique	135
	La fonction de coût	138
	Résultats de recalage par programmation dynamique	139
6.2.4	Corrections géométriques et topologiques	141
	Réajustement du modèle avec contraintes angulaires	143
6.2.5	Résultats	144
6.3	Conclusion	146
7	Résultats	147
7.1	Le cadre	147
7.2	Système de modélisation proposé	152
7.2.1	Segmentation du MNE	152
7.2.2	Modèle polygonal du bâtiment	153
	Construction du modèle	153
	Modèle avec contraintes angulaires	153
7.2.3	Temps de calcul	157

7.2.4	Validation des modèles reconstruits	157
7.3	Autres études et résultats	160
7.3.1	Reconstruction de façades	161
	Reconstruction des points sur les façades	161
	Correction du MNE	161
	Initialisation à l'aide des vecteurs automatiques	161
7.3.2	Système de modélisation appliqué sur un MNE corrigé	167
	Segmentation d'un MNE corrigé	167
	Modèle polygonal du bâtiment	167
7.3.3	Orthogonalisation des modèles recalés par contours actifs	167
7.3.4	Snakes guidés par les façades	171
7.3.5	Temps de calcul	174
7.4	Conclusion	174
8	Conclusion	177
8.1	Contributions	177
8.2	Perspectives	180
	Bibliographie	182

Chapitre 1

Introduction

Disposer de descriptions tridimensionnelles d'une scène urbaine intéresse de plus en plus d'applications industrielles. De nombreuses applications comme l'urbanisme ou la cartographie poussent les recherches liées à ce domaine. Mais c'est surtout l'augmentation du nombre d'utilisateurs de téléphones mobiles qui a accru l'intérêt pour ces bases de données géographiques numériques. Les fournisseurs de services en télécommunications ont besoin de Modèles Numériques Urbains (MNU) précis pour alimenter les outils de planification qui leur permettent de déployer et de densifier leurs réseaux cellulaires. Ces modèles urbains permettent d'améliorer et de faciliter le développement de leurs réseaux de télécommunications.

Avec comme objectif la fourniture d'un produit destiné aux applications des télécommunications, cette thèse traite de la modélisation de bâtiments. Elle présente plusieurs outils en vue d'une automatisation de ce processus complexe. Les objectifs que nous nous fixons sont la détection de tous les bâtiments, le traitement des bâtiments de forme quelconque, obtenir un modèle représentatif de la forme générale du bâtiment et avoir un contour du bâtiment de bonne qualité. Tous les outils que nous utilisons ont été développés pour répondre au mieux à ces objectifs.

Cette thèse s'est déroulée à la fois dans un milieu industriel et universitaire. Elle a été financée par un contrat CIFRE entre le projet Robotvis de l'INRIA (Institut National de Recherche en Informatique et Automatique) de Sophia-Antipolis et la société ISTAR (Imagerie STéréo Appliquée au Relief). Le projet Robotvis est spécialisé depuis de nombreuses années dans plusieurs domaines de la vision par ordinateur et ISTAR est une société de hautes technologies produisant des bases de données cartographiques numériques à partir d'images ou de cartes, principalement pour le domaine des télécommunications.

Pour la reconstruction des bâtiments, nous disposons de plusieurs sources d'information. Le chantier de test utilisé est Berlin, il comprend : 24 images d'une résolution de $37.5cm$, tous les paramètres du système de caméras défini par ces images, un Modèle Numérique d'Élévation (MNE) de la scène et l'ortho-image correspondante, ces derniers ayant une résolution de $50cm$. Nous avons choisi une stratégie de reconstruction en trois étapes : (1) la construction d'un MNE (Modèle Numérique d'Élévation) dense et fiable, (2) la segmentation des bâtiments en régions planes et (3) la vectorisation des contours des plans pour obtenir la description finale du bâtiment. Cette thèse se compose de six chapitres que nous résumons maintenant.

Plan de la thèse

Le chapitre 2 est une introduction à la reconstruction de scène à partir d'images. Nous décrivons tout d'abord les étapes fondamentales d'un système de vision : la modélisation du système, le calibrage, la mise en correspondance des images et la reconstruction. Nous présentons ensuite une revue des principales méthodes de mise en correspondance des images. L'objectif est de comprendre à la fois le processus qui permet de former les images, la géométrie d'un système de caméras, mais aussi les problèmes liés à la mise en correspondance des images qui permet la reconstruction de la scène observée.

Le chapitre 3 traite de la reconstruction de bâtiments à partir d'images. Nous commençons par rappeler les étapes de ce système de vision dans le contexte du traitement d'un chantier aérien. Nous exposons le problème lié à la modélisation des bâtiments et sa complexité. De nombreux instituts de recherche ou entreprises ont focalisé leur recherche sur la réalisation de ces modèles avec des objectifs, des applications, des stratégies ou des images différents. Certains utilisent des prises de vues du sol pour obtenir la texture des bâtiments tandis que d'autres utilisent des données issues de l'imagerie radar. Nous présentons ensuite un état de l'art des méthodes de reconstruction de bâtiments qui utilisent l'imagerie optique comme nous le faisons.

Tout algorithme de mise en correspondance appliqué à des images aériennes se trouve confronté aux difficultés liées à l'appariement des points sur les façades. Les images des façades représentant une quantité d'information importante en imagerie aérienne, nous proposons dans le chapitre 4 une méthodologie permettant la mise en correspondance de ces images et la reconstruction des points de ces surfaces tridimensionnelles. Le principe est de "rectifier" les images par rapport à un plan vertical représentant chaque façade. Un simple algorithme de mise en correspondance par corrélation permet ensuite de reconstruire les points 3D de la surface verticale correspondante. L'algorithme que nous proposons n'utilise aucune connaissance a priori sur la forme des bâtiments. Nous proposons ensuite d'utiliser les points reconstruits pour corriger un MNE.

Le chapitre 5 traite de la segmentation d'un MNE en plans. L'objectif est de retrouver les différentes parties des toits qui composent chaque bâtiment. Nous proposons tout d'abord une revue des principales méthodes d'estimation robuste. Nous avons développé ensuite un algorithme de segmentation de MNE en plans, qui comprend trois phases. Une phase d'exploration basée sur la technique robuste du RANSAC (**R**ANdom **S**Ample **C**onsensus) permet de construire un jeu d'hypothèses de modèles de plans. Une phase de fusion supprime ensuite la redondance dans la liste de modèles. Enfin, une phase de sélection basée sur le MDL (**M**inimum **D**escription **L**ength) et la recherche par tabou permet de ne conserver que les meilleures hypothèses. L'algorithme que nous avons développé permet de rechercher des plans d'orientation quelconque ou bien des plans horizontaux. Du fait de la faible résolution du MNE (50cm) et de son aspect lisse, nous utilisons uniquement des plans horizontaux dans les chapitres suivants de la thèse pour décrire la structure principale des bâtiments.

Le chapitre 6 traite de la vectorisation. L'objectif est de construire le modèle polygonal du bâtiment. Nous proposons tout d'abord un algorithme d'approximation polygonale de chaînes qui permet de construire un modèle polygonal initial du bâtiment à partir des plans sélectionnés par l'algorithme de segmentation. Nous étudions ensuite trois techniques différentes pour corriger le modèle construit : une technique de recalage à l'aide de snakes polygonaux multi-images, une technique de recalage par programmation dynamique et une

technique de correction qui impose des contraintes angulaires d'orthogonalité au modèle polygonal.

Le dernier chapitre présente les résultats obtenus avec les algorithmes que nous avons développés. Nous utilisons pour cela une zone d'intérêt de taille importante ($1km^2$) sur une zone urbaine assez complexe. Nous proposons un système de modélisation et présentons ses résultats sur la zone d'intérêt définie. Avec le but de valider les modèles obtenus, nous présentons ensuite les résultats d'une première étude qualitative basée sur des outils télécoms. Nous présentons aussi d'autres résultats, obtenus en utilisant et en combinant au mieux les outils que nous avons développés au cours de cette thèse.

Chapitre 2

Des images à la troisième dimension

Ce premier chapitre présente de manière rapide les notions de base d'un système de vision stéréoscopique. Il est important de comprendre à la fois le processus qui permet de former les images, la géométrie qui relie les caméras, ainsi que tous les problèmes de mise en correspondance des images pour permettre la reconstruction de la scène observée. Ce chapitre est composé de deux sections : une première présentant toutes les étapes fondamentales d'un système de vision stéréoscopique et une seconde partie traitant plus particulièrement de la mise en correspondance des images.

2.1 Retrouver la troisième dimension

Pour accéder à l'information de profondeur de la scène observée, il est généralement nécessaire d'utiliser au minimum deux vues dont le point de vue diffère. Le processus qui permet de combiner les images pour en extraire les informations géométriques tridimensionnelles est nommé *stéréoscopie* ou *stéréo-vision*. Un exemple simple et couramment cité est le système stéréoscopique binoculaire humain. Ce système biologique complexe est vraisemblablement l'un des plus performants. Pour cette raison, de nombreux algorithmes de stéréoscopie présentés dans la littérature font référence à certaines de ses propriétés [Bel96, MF81].

Le problème que l'on souhaite résoudre est de reconstruire l'objet ou la scène visualisée à partir de plusieurs images de cette scène. La toute première étape consiste à déterminer l'ensemble des lois et des paramètres qui régissent le système de caméras utilisé. On commence le plus souvent par choisir le modèle de caméra le mieux adapté à l'application traitée. Il faut ensuite définir tous les paramètres de ce système : les paramètres intrinsèques des caméras (focale, ...), leurs positions dans l'espace et leurs orientations, c'est l'étape dite de *calibrage*. Cette étape est fondamentale pour déterminer les fonctions qui permettent de passer d'un point de l'espace 3D à ses projections dans les images et inversement. Connaissant la géométrie du système, il devient alors possible de l'étudier et de mettre en œuvre une procédure pour reconstruire la scène 3D à partir des images.

Lorsque le calibrage est effectué, le problème que l'on cherche à résoudre est de trouver dans les différentes images les points ou les objets qui se correspondent. C'est le problème dit de *mise en correspondance des images* que nous détaillons dans la section 2.2. La scène est ensuite reconstruite en utilisant le principe de *triangulation* (figure 2.1). On reconstruit un point matériel de la scène en calculant l'intersection des rayons optiques (aussi appelés

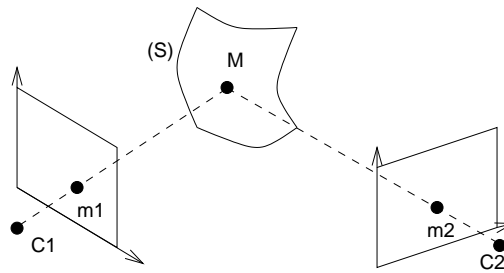


FIG. 2.1 – *Principe de la triangulation : le point M de la scène est reconstruit à partir du couple de points homologues $m1$ et $m2$. Il se trouve à l'intersection des rayons optiques, les droites passant par chacun de ces deux points et les centres optiques des caméras $C1$ et $C2$.*

lignes de vues) passant par ses projections. L'étape de mise en correspondance des images joue donc un rôle important puisque la qualité finale de la scène reconstruite est directement liée à la capacité de l'algorithme d'appariement à retrouver les points qui se correspondent. Nous verrons par la suite qu'il est possible de reconstruire la scène observée en utilisant et mettant en correspondance d'autres primitives telles que les segments ou les contours.

Le processus de reconstruction de scène tridimensionnelle qui vient d'être présenté peut être décomposé en plusieurs modules distincts constituant encore aujourd'hui des axes de recherche importants pour de nombreux laboratoires de recherche. Plusieurs ouvrages [Fau93, XZ96, Aya89] ont été publiés sur le domaine et sont vivement conseillés pour approfondir certains des sujets que nous abordons dans le chapitre. Nous commençons par présenter les modèles de caméra, la géométrie d'un système de caméras, les principales méthodes de calibrage et la rectification des images qui simplifie considérablement la mise en correspondance des images.

2.1.1 Modélisation de la caméra

Nous allons étudier et décrire maintenant le processus de formation des images à partir de la scène tridimensionnelle. Il s'agit d'un problème important de la vision par ordinateur puisque les imperfections du modèle de caméra choisi vont se répercuter sur tous les processus qui vont utiliser la géométrie choisie du capteur. Il est nécessaire que le modèle choisi corresponde le plus précisément possible au système d'acquisition utilisé.

Modèles de caméras Différents modèles géométriques ont été proposés pour représenter mathématiquement le système d'acquisition utilisé. Le choix du modèle de caméra que l'on utilise dépend à la fois du modèle de projection (perspectif, parallèle, sphérique), du modèle de distorsion due à l'optique et du degré de simplification que l'on choisit selon l'application. La figure 2.2 présente les trois modèles les plus couramment utilisés en vision par ordinateur.

Le modèle orthographique Le modèle de projection orthographique est le plus simple. C'est un modèle perspectif dont le centre optique est à l'infini. Tous les rayons optiques sont parallèles entre eux et orthogonaux au plan rétinien (plan de formation de l'image) comme le montre la figure 2.2. On appelle aussi ce modèle : **modèle perspectif faible**. Il représente très grossièrement le processus d'acquisition d'image. On peut utiliser un facteur d'échelle de l'image pour améliorer ce modèle.

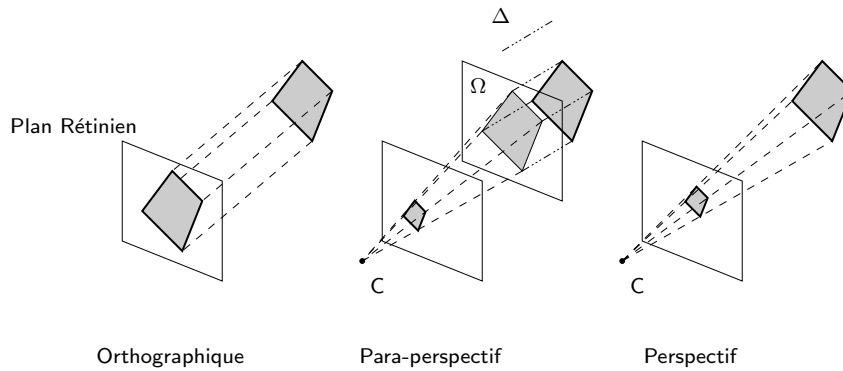


FIG. 2.2 – Trois modèles de caméras : orthographique, para-perspectif et perspectif. Le même objet est projeté de trois manières différentes. Les rayons de projection sont indiqués en pointillés.

Le modèle para-perspectif Le modèle para-perspectif est plus apprécié car il est plus complet que le modèle orthographique. Il tient compte du facteur d'échelle. Ce modèle se compose d'une projection parallèle de direction Δ sur un plan Ω parallèle au plan rétinien suivie d'une projection perspective de centre \mathbf{C} (représentant le centre optique de la caméra). Le plan Ω est choisi de manière à ce qu'il représente au mieux la scène. On choisit généralement le plan médian. On considère dans ce modèle que tous les points de la scène ont une même profondeur, la distance qui sépare le plan rétinien du plan Ω .

Le modèle perspectif Le modèle perspectif (appelé aussi sténopé) est le plus couramment utilisé et aussi le plus complet. Il permet de représenter de manière plus précise le processus de formation de l'image en schématisant la caméra par un plan rétinien et son centre optique (figure 2.2). Ce modèle est une projection perspective parfaite de l'espace 3D observé vers le plan 2D correspondant à l'image. Il possède l'avantage de modéliser fidèlement la plupart des capteurs projectifs en simplifiant les mathématiques mises en jeu pour l'estimation des paramètres du modèle.

Description algébrique du modèle sténopé La figure 2.3 présente le modèle sténopé en détail. La caméra est représentée par un plan rétinien Ω^r et un centre optique \mathbf{C} n'appartenant pas à ce plan. On définit le plan focal comme étant le plan parallèle au plan rétinien Ω^r passant par le centre optique \mathbf{C} , on le note Ω^f . La distance qui sépare ces deux plans est la distance focale \mathbf{f} . Le point \mathbf{c} est la projection orthogonale de \mathbf{C} sur le plan Ω^r . On appelle \mathbf{c} le point principal et la droite \mathbf{Cc} l'axe optique de la caméra. L'image (aussi appelée projection ou projeté) d'un point \mathbf{M} de la scène 3D est le point d'intersection de la droite (appelée rayon optique ou ligne de vue) (\mathbf{CM}) avec le plan rétinien Ω^r .

Modéliser le système d'acquisition consiste à définir la fonction de projection permettant de passer du point \mathbf{M} défini dans le repère orthonormé $\mathbf{R}(O, R_x, R_y, R_z)$ de l'espace euclidien tridimensionnel \mathcal{R}^3 au point \mathbf{m} défini dans le repère affine $(\mathbf{o}, \mathbf{u}, \mathbf{v})$ du plan rétinien Ω^r . θ désigne l'angle formé par les vecteurs \mathbf{u} et \mathbf{v} . On démontre qu'il est possible de décomposer cette fonction de projection \mathbf{P} en trois étapes distinctes : un changement de repère dans l'espace \mathbf{T} , une projection \mathbf{P}_0 , puis un changement de repère dans le plan \mathbf{A} .

On utilise généralement la géométrie projective pour obtenir une forme simplifiée de la

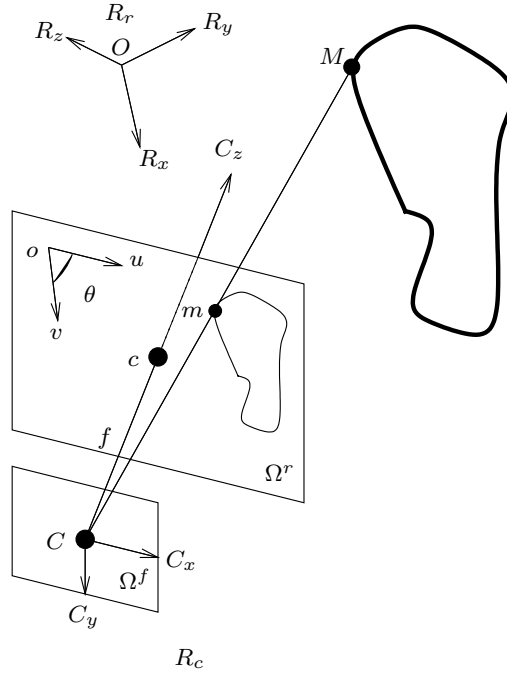


FIG. 2.3 – Le modèle de caméra sténopé (perspectif ou projectif linéaire), projection d'un objet de l'espace 3D sur le plan rétinien.

fonction de projection. Soit $\mathbf{M} \cong (x, y, z, 1) \in \mathcal{P}^3$ la représentation en coordonnées homogènes du point $\mathbf{M} \notin \Omega^f$ et $\mathbf{m} \cong (u, v, 1) \in \mathcal{P}^2$ celle du point image associé. Alors l'expression de la fonction de projection est simplement une multiplication du vecteur représentant les coordonnées du point \mathbf{M} par la matrice 3×4 \mathbf{P} :

$$\mathbf{m} \cong \mathbf{P}\mathbf{M} \quad (2.1)$$

En utilisant les trois étapes successives pour définir la fonction de projection, on peut écrire la matrice de projection \mathbf{P} comme suit :

$$\mathbf{P} \cong \mathbf{A}\mathbf{P}_0\mathbf{T} \quad (2.2)$$

Changement de repère dans l'espace \mathbf{T} Le changement de repère \mathbf{T} permet d'exprimer les coordonnées du point \mathbf{M} de l'espace, défini dans le repère R_r et représenté par le vecteur $\mathbf{M}_r(X_r, Y_r, Z_r)$, dans le repère $R_c(C, C_x, C_y, C_z)$. On définit \mathbf{T} comme une transformation rigide : une rotation \mathbf{R} suivie d'une translation \mathbf{t} . On obtient le vecteur $\mathbf{M}_c(X_c, Y_c, Z_c)$ tel que $\mathbf{M}_c(X_c, Y_c, Z_c) = \mathbf{R}\mathbf{M}_r + \mathbf{t}$ (avec \mathbf{R} matrice 3×3 et \mathbf{t} vecteur de dimension 3). La matrice \mathbf{T} s'écrit :

$$\mathbf{T} \cong \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \quad (2.3)$$

Projection simple dans le plan \mathbf{P}_0 La projection sur le plan rétinien Ω^r s'écrit :

$$\mathbf{P}_0 \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.4)$$

Le vecteur \mathbf{m} , représentant les coordonnées de la projection dans le plan rétinien exprimées dans le repère $\mathbf{R}_c(C, C_x, C_y, C_z)$, doit avoir pour dernière coordonnée 1. Cette transformation équivaut à diviser les coordonnées X_c et Y_c par Z_c/f .

Changement de repère dans le plan A Le passage en coordonnées pixels se fait alors par le changement de repère affine \mathbf{A} :

$$\mathbf{A} \cong \begin{bmatrix} \alpha_u & -\alpha_u \cdot \cot \theta & u_0 \\ 0 & \alpha_v \cdot \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

α_u et α_v sont les facteurs multiplicatifs selon chaque axe. Ils représentent la distance focale exprimée en unités pixels horizontalement et verticalement. u_0 et v_0 sont les coordonnées du point \mathbf{c} dans le repère (o, u, v) . En pratique, θ est souvent proche de $\pi/2$, les valeurs de α_u et de α_v sont en général données avec les dimensions des pixels sur le capteur (fournies par le constructeur de la caméra) et la distance focale (qui dépend de l'objectif).

La matrice \mathbf{A} ne dépend que des paramètres propres à chaque caméra, elle est appelée matrice des paramètres intrinsèques. \mathbf{R} et \mathbf{t} sont les paramètres extrinsèques de la caméra, ils ne dépendent que de la position et de l'attitude de la caméra dans l'espace. \mathbf{R} et \mathbf{t} dépendent de 3 paramètres chacun. \mathbf{P} est de rang 3, elle est définie à une constante multiplicative près et dépend de 11 paramètres.

La distorsion Le modèle de projection sténopé développé ci-dessus correspond à une projection perspective parfaite. Les caméras, et les systèmes optiques qu'elles comprennent, présentent des imperfections. La fonction de projection doit prendre en compte ces imperfections pour limiter les erreurs de projection en donnant plus de souplesse au modèle. Nous avons formulé dans le modèle sténopé trois hypothèses implicites. Tout d'abord, nous avons supposé que l'image se forme sur une surface plane et le centre de projection ponctuel. Ensuite, que les pixels sont régulièrement espacés dans le plan image. Enfin, la dernière hypothèse suppose que la caméra ne possède aucune distorsion géométrique (il peut aussi y avoir des distorsions chromatiques ou radiométriques). Les deux premières hypothèses sont généralement vérifiées pour les caméras CCD. Pour obtenir une meilleure précision de la fonction de projection, il faut tenir compte des distorsions non linéaires engendrées par les imperfections de l'optique dans le modèle de caméra : distorsion radiale, distorsion de décentrage et distorsion prismatique (voir [Sla80, Dev97]).

L'expression de ces distorsions peut se faire en ajoutant un terme correctif à chacune des coordonnées dans l'image \mathbf{x} et \mathbf{y} correspondant au modèle sténopé. Les coordonnées \mathbf{x}_d et \mathbf{y}_d mesurées sont de la forme :

$$\begin{aligned} x_d &= x + \delta_x(x, y) \\ y_d &= y + \delta_y(x, y) \end{aligned}$$

Dans ces équations, $\delta_x(\mathbf{x}, \mathbf{y})$ et $\delta_y(\mathbf{x}, \mathbf{y})$ sont eux-mêmes la somme de plusieurs termes correctifs correspondant aux différentes formes de distorsion, ainsi qu'au degré d'approximation du modèle de distorsion. L'estimation de ces paramètres supplémentaires du modèle de projection peut être réalisée soit directement à partir des images pour nous ramener au modèle projectif simple, soit grâce au calibrage des paramètres du système.

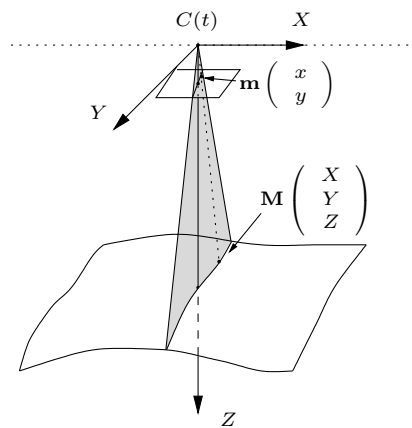


FIG. 2.4 – Projection dans le cas d'une caméra linéaire push-broom.

Le capteur linéaire (push-broom) Le principe des capteurs push-broom est présenté dans la figure 2.4. Le centre de projection $C(t)$ se déplace le long de la trajectoire du satellite ou de l'avion. Cette trajectoire (direction \mathbf{X} dans la figure) correspond globalement à un mouvement linéaire ou elliptique. Lorsque l'avion survole la scène à photographier, on lance les acquisitions des lignes CCD (Charge Coupled Device) avec une période δ_t . La direction du capteur (direction \mathbf{Y} dans la figure) doit être perpendiculaire à la trajectoire. L'image finale est construite par la juxtaposition de ces lignes. A chaque acquisition, on réalise une projection perspective simple bidimensionnelle dans le plan formé par la position du centre instantané de projection $C(t)$ et le capteur linéaire. δ_t est choisi généralement pour que la dimension des pixels selon \mathbf{X} soit identique à la dimension des pixels selon \mathbf{Y} . Pour plus d'informations sur ce type de capteur, il est conseillé de lire l'article de *R. Hartley* et *R. Gupta* [HG94].

Le capteur linéaire est très utilisé en imagerie aérienne et spatiale. Des exemples connus sont le capteur optique SPOT4 et certains capteurs laser. ISTAR a développé avec le DLR (équivalent du CNES en Allemagne) un capteur aérien linéaire, le HRSC (High-Resolution Stereo Camera), pour pouvoir lancer des campagnes d'acquisition de haute résolution (25cm) sur des zones de très grande taille (plusieurs milliers de km^2). Le principal avantage de ce capteur est de pouvoir reconstruire de très grandes surfaces 3D à partir d'un jeu de données de taille plus faible (en comparaison à la taille des données que nécessiterait un capteur de type sténopé). De plus, le capteur a une capacité d'acquisition illimitée, en théorie, le long d'une trace. Il permet ainsi de reconstruire, à partir de deux traces parallèles se recouvrant, de très grandes bandes de la surface 3D.

La modélisation et l'analyse de ce capteur restent toutefois difficiles. Le mode d'acquisition est différent entre un capteur sténopé et un capteur linéaire. Dans le premier cas, la prise de vue est instantanée et on peut ne pas prendre en compte la cinématique de l'avion. On peut modéliser le capteur par un modèle sténopé. Dans le second cas, l'image est constituée d'une succession de prises de vue linéaires et il faut alors tenir compte de tous les mouvements possibles de l'avion (tangage, roulis, variation de vitesse, ...) pour modéliser l'image. On utilise pour cela une centrale inertielle couplée à un système GPS (Global Positioning System).

2.1.2 Système de caméras

Nous avons examiné le formalisme pour passer de l'espace 3D à une image. Nous nous intéressons maintenant à celui qui permet de passer d'une image à une autre. La figure 2.5 nous présente un système de deux caméras représentées par les centres optiques $C1$ et $C2$. L'ensemble des propriétés géométriques liées à ce système est souvent appelé géométrie épipolaire.

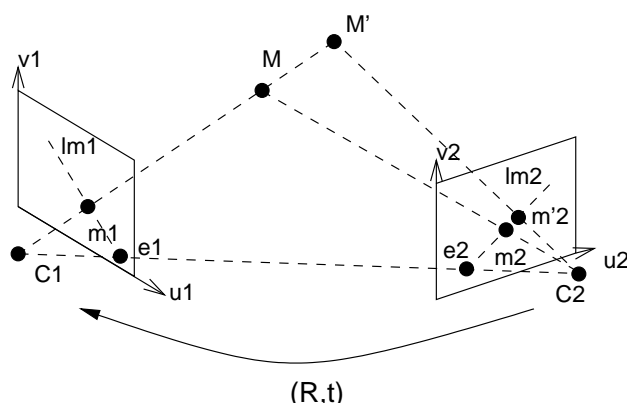


FIG. 2.5 – Un système de deux caméras : La géométrie épipolaire permet d'exprimer l'ensemble des propriétés géométriques de ce système.

Nous cherchons à exprimer la relation permettant de lier les différentes projections d'un même point M de l'espace. Soit $m1$ un point de la première image. Le point de l'espace M qui lui correspond est nécessairement sur la droite $(C1m1)$. La projection $m2$ du point M dans la seconde image est obligatoirement sur la projection de cette droite $(C1m1)$ dans la deuxième image. Cette droite, notée $lm2$, est appelée droite épipolaire.

Comme le présente la figure 2.5, tous les points de la seconde image appartenant à la droite épipolaire $lm2$ définissent un point de l'espace différent lorsqu'on les associe au point $m1$ de la première image. Ainsi, le point de l'espace M correspond au couple de points homologues $(m1, m2)$ et le point M' correspond au couple $(m1, m'2)$.

On remarque que toutes les droites épipolaires de la seconde image passent par le point $e2$, appelé épipole, qui est l'intersection de la droite $(C1C2)$ avec le plan rétinien de la seconde caméra. Réciproquement, on peut déterminer l'épipole dans la première image que l'on nomme $e1$. Les points $m1$, $m2$, $C1$ et $C2$ appartiennent à un même plan. Ce plan est utilisé comme contrainte pour définir la géométrie épipolaire reliant les deux caméras.

La matrice essentielle E Lorsque les paramètres intrinsèques des caméras sont connus, la géométrie épipolaire est décrite par la matrice essentielle E introduite par *Longuet-Higgins* dans [LH81]. Cette matrice E , en choisissant un repère judicieux de l'espace, permet le passage d'un point d'une image à la droite épipolaire qui lui correspond dans la seconde image. Pour plus d'informations sur le calcul de cette matrice et ses propriétés, on peut se reporter par exemple à [Fau93].

La matrice fondamentale F Lorsque les paramètres intrinsèques des caméras sont inconnus, on utilise alors une autre entité qui est la matrice fondamentale F . Cette matrice, définie à facteur d'échelle près, est reliée à la matrice essentielle E par une simple relation

linéaire et permet d'agir sur les coordonnées pixels. Cette matrice reflète la contrainte de coplanarité des points $(\mathbf{C}_1, \mathbf{C}_2, \mathbf{m}_1, \mathbf{m}_2)$. L'équation 2.6 nous présente la relation utilisée :

$$\mathbf{m}_1^T \mathbf{F}_{12} \mathbf{m}_2 = 0 \quad \text{avec} \quad \mathbf{F}_{12} = \mathbf{A}_1^T [\mathbf{t}]_{\times} \mathbf{R} \mathbf{A}_2^{-1} \quad (2.6)$$

où $[\mathbf{t}]_{\times}$ est la matrice antisymétrique définie par $[\mathbf{t}]_{\times} \mathbf{x} = \mathbf{t} \wedge \mathbf{x}$ quel que soit le vecteur \mathbf{x} . \mathbf{A}_1 et \mathbf{A}_2 sont les matrices des paramètres intrinsèques de chaque caméra. Réciproquement, on peut montrer en échangeant le rôle des caméras que l'équation 2.6 devient $\mathbf{m}_2^T \mathbf{F}_{21} \mathbf{m}_1 = 0$ et que $\mathbf{F}_{21} \cong \mathbf{F}_{12}^T$.

2.1.3 Calibrage du système

Cette opération est importante puisqu'elle va servir à définir les relations entre la scène et les caméras. Nous avons vu dans les sections précédentes différents modèles de caméra ainsi que les outils permettant le passage d'une image à l'autre. Calibrer le capteur ou le système de capteurs consiste à déterminer tous les paramètres de ce système. Nous pourrons ensuite associer les points de l'espace 3D à leurs projections dans les images et les points des images entre eux. Dans l'application que nous traiterons dans les chapitres suivants, nous sommes dans le cas du calibrage d'un système multi-caméras observant la même scène.

Il existe différents niveaux de calibrages, dépendant principalement des informations dont on dispose sur la scène et le système de caméras lui-même. Chacun de ces niveaux est directement lié à la structure géométrique dans laquelle nous allons reconstruire la scène (voir 2.1.6 pour une description plus détaillée). *O. Faugeras* présente dans [Fau95] ces différents niveaux de calibrages sous forme de strates et *C. Zeller* présente dans sa thèse [Zel96] les calibrages correspondant et leur mise en œuvre. Les deux principaux types de calibrage couramment utilisés sont le calibrage fort et le calibrage faible.

Calibrage fort On parle de calibrage fort, ou euclidien, lorsque tous les paramètres intrinsèques et extrinsèques du système sont connus. La plupart des méthodes de calibrage se servent de points d'appui pris dans l'espace et retrouvés dans les images. Chacun des points utilisés génère une équation. On résout ensuite le système d'équations obtenu pour déterminer tous les paramètres du système et la géométrie épipolaire des différents couples d'images. Les points utilisés sont soit des points 3D connus dans un référentiel absolu de l'espace (extraits d'une carte de la scène par exemple), soit des points d'un objet de calibrage (ou d'une mire) de géométrie connue [FT86, LT88, Sla80]. Il existe aussi d'autres méthodes permettant d'obtenir un calibrage euclidien, basées sur l'ajout d'information ou de contraintes à un calibrage de niveau inférieur (projectif ou affine).

Calibrage faible Dans le cas où nous n'avons aucune information 3D sur la scène, il devient nécessaire d'utiliser une méthode qui tienne compte du fait que les caméras qui constituent notre système observent la même scène. Dans ce cas, on cherche à déterminer la géométrie épipolaire du système de caméra, la matrice fondamentale \mathbf{F} , à partir de couples de points ou de primitives extraits des deux images. On parle alors de calibrage faible ou de calibrage projectif [LH81]. Ce type de calibrage permet d'obtenir une bonne précision de la géométrie épipolaire (élément essentiel pour les algorithmes de mise en correspondance qui utilisent cette géométrie pour apparier les primitives). Une revue des méthodes de calibrage faible est présentée dans [Zha96].

Autres types et méthodes de calibrage Le but final est de déterminer la profondeur dans l'espace et d'obtenir ainsi une représentation euclidienne de la scène étudiée. Il est donc important d'obtenir un calibrage précis du système. On préfère souvent combiner le calibrage faible, qui nous donne une meilleure précision sur la géométrie épipolaire (en utilisant \mathbf{F}), et le calibrage fort qui permet quant à lui la reconstruction euclidienne de la scène.

2.1.4 Rectification de deux images

Une grande partie des algorithmes de stéréo-vision utilise la géométrie épipolaire pour réaliser la mise en correspondance des primitives¹. Pour une primitive extraite de la première image, on va rechercher dans l'espace restreint par la contrainte épipolaire (une droite pour un point) de la seconde image la primitive qui lui est associée. La rectification des images, un exemple est présenté en figure 2.6, est une étape intermédiaire entre le calibrage et la mise en correspondance, elle permet de faciliter cette recherche en simplifiant la géométrie du système.

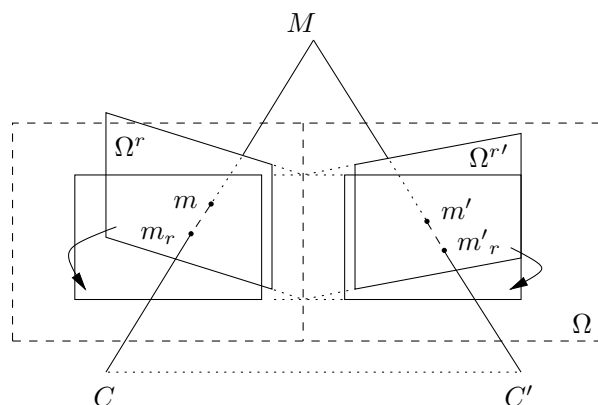


FIG. 2.6 – *Un exemple de rectification possible : Cette rectification consiste à reprojeter les images des plans rétinaux Ω^r et $\Omega^{r'}$ sur un même plan Ω , le plan de rectification, passant par la droite d'intersection de Ω^r et $\Omega^{r'}$ et parallèle à la droite (CC') .*

La rectification permet d'obtenir une géométrie épipolaire simple dans laquelle toutes les droites épipolaires sont parallèles et horizontales, elles correspondent alors aux lignes des images. Les points (ou pixels) qui se correspondent possèdent alors obligatoirement la même ordonnée et la recherche d'appariements est mono-dimensionnelle, le long des lignes de chaque image. La rectification ne demande aucune connaissance a priori sur la géométrie de la scène. La figure 2.6 nous présente une des rectifications possibles avec un modèle de caméra sténopé. En réalité, il existe toute une famille de rectifications possibles. *F. Devernay* présente dans [Dev97] des méthodes pour choisir une paire de matrices appelées **matrices de rectification**, lorsque l'on connaît uniquement la géométrie épipolaire du système stéréoscopique. La rectification peut aussi être faite par rapport à un plan. Cette méthode est employée lorsque la surface moyenne de la scène est inclinée comme le sol en imagerie aérienne car la contrainte fronto-parallèle n'est alors plus valable. Plusieurs rectifications sont présentées dans le livre de *O. Faugeras* [Fau93]

1. notamment les techniques dites de corrélation, section 2.3.1

2.1.5 Disparité et espaces de disparité

La disparité On définit la *disparité* \mathbf{dr} comme le déplacement relatif mesuré entre les deux projections $\mathbf{m1}$ et $\mathbf{m2}$ d'un point \mathbf{M} de la scène 3D :

$$\mathbf{dr} : m1 \rightarrow \mathbf{dr}(m1) \quad \text{tel que} \quad m2 = m1 + \mathbf{dr}(m1) \quad (2.7)$$

Dans une géométrie que nous qualifions de *standard* (rectifiée parallèlement aux centres optiques), on utilise sa composante \mathbf{d} le long des lignes épipolaires comme le présente la figure 2.7.

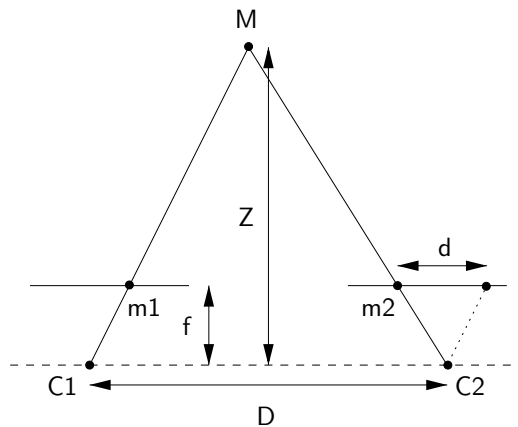


FIG. 2.7 – La disparité en géométrie épipolaire.

Toujours dans cette géométrie, la disparité est directement liée à la profondeur du point 3D :

$$d = -f \frac{D}{Z} \quad (2.8)$$

Lorsque la disparité est nulle, cela implique que le point \mathbf{M} est à une distance \mathbf{Z} infinie. La disparité est donc une grandeur comprise entre 0 et $+\infty$. Elle varie de manière discontinue et elle peut être non définie lorsqu'un point 3D de la scène n'est vu que dans une seule image au plus.

L'espace de mise en correspondance Pour simplifier le problème de mise en correspondance, on utilise la géométrie épipolaire. Considérons alors la mise en correspondance de deux droites épipolaires homologues (même ligne dans chacune des images). On va rechercher pour chaque point de la première droite son correspondant dans la seconde et réciproquement. Pour représenter les appariements réalisés entre tous les points des deux droites étudiées, on construit un graphe comme le présente la figure 2.8 C. On appelle *espace de mise en correspondance* l'espace 2D dont les variables sont les positions des pixels $\mathbf{x_d}$ et $\mathbf{x_g}$ dans chaque droite épipolaire. Le résultat de la mise en correspondance est une courbe, tracée dans cet espace discret, correspondant aux appariements trouvés entre les points des deux lignes épipolaires.

On remarque que la courbe ainsi définie présente certaines propriétés. Comme le présente la figure 2.8 C, la première propriété que l'on peut démontrer est que la courbe est monotone lorsqu'on utilise la contrainte d'ordre. La contrainte d'ordre peut être imposée sur la totalité

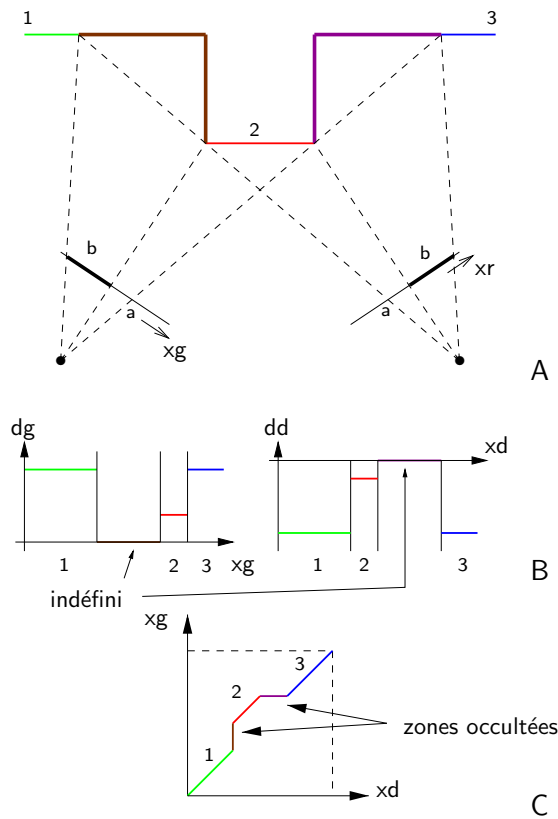


FIG. 2.8 – Les différents espaces de disparité : A présente une scène et le système d’acquisition. Les régions de la scène qui n’apparaissent que dans l’une des images (en trait épais) engendrent des zones où la disparité n’est pas définie (a) et des contours d’occultation (b). B présente les courbes correspondant aux images de l’espace de disparité. Les sauts verticaux de chacune des courbes correspondent aux contours d’occultation. Les sauts horizontaux correspondent aux régions occultées. C présente l’espace de mise en correspondance. La courbe représente aussi les appariements réalisés pour une droite épipolaire. Les occultations (zones en trait épais) présentes dans les images se traduisent par des discontinuités (horizontales ou verticales) de cette courbe.

des scènes aériennes étudiées (cette contrainte, voir la section 2.2.3, n’est pas toujours vérifiée dans les applications de robotique). La seconde propriété concerne les occultations. A un contour d’occultation présent dans une des images correspond une zone d’occultation dans l’autre (voir la section 2.2.2 pour ces deux propriétés). Ainsi, comme le présente la figure 2.8 C, le contour d’occultation (a) présent dans l’image de gauche (entre les zones 2 et 3) se caractérise par une discontinuité horizontale. Cette discontinuité correspond à une zone de pixels demi-occultés (b) dans l’image de droite qui ne possèdent aucun correspondant. Réciproquement, un contour d’occultation présent dans l’image de droite se caractérise par une discontinuité verticale qui engendre une zone de pixels demi-occultés dans l’image de gauche. L’utilisation de cet espace possède l’avantage de pouvoir introduire directement dans la procédure qui construit la courbe d’appariement des contraintes sur la forme des objets de l’espace.

2.1.6 Reconstruction de la scène

L'étape de reconstruction de la scène intervient lorsqu'on a mis en correspondance les différentes images du système (voir 2.2). Cette reconstruction se fait par triangulation. Il existe différentes familles de reconstruction de la scène, qui dépendent des informations dont on dispose sur la scène (position des caméras dans un référentiel absolu, mesures métriques d'objets de la scène ...). Ainsi, la structure géométrique tridimensionnelle de la scène dépend essentiellement du calibrage du système de caméras.

O. Faugeras présente dans [Fau95] une hiérarchie naturelle de ces différentes structures géométriques sous la forme d'un jeu de trois strates : la structure projective, la structure affine et la structure euclidienne. Connaissant uniquement la géométrie épipolaire du système de caméras (calibrage faible), on réalise une reconstruction projective. La scène est reconstruite à une collinéation ou homographie près, elle préserve uniquement le birapport de quatre points. La seconde strate est la structure affine. Pour passer de la structure projective à la structure affine, il suffit d'ajouter la connaissance du plan à l'infini. La scène est maintenant construite à une transformation affine près, elle préserve le rapport des longueurs le long d'une droite et le parallélisme. Enfin, la dernière strate est la structure euclidienne, c'est celle qui nous intéresse pour la reconstruction de bâtiments. Pour passer de la structure affine à la structure euclidienne, il suffit d'ajouter la connaissance de la conique absolue. On peut aussi obtenir cette reconstruction en imposant des contraintes euclidiennes à une reconstruction affine, par auto-calibrage ou par calibrage fort. La scène est construite à un facteur d'échelle global près, elle permet de mesurer des angles et le rapport des longueurs de segments non-parallèles. Dans le cas du calibrage fort ou lorsqu'on connaît une mesure de distance, la reconstruction est obtenue dans le référentiel absolu utilisé.

2.2 Stéréoscopie

L'étape de mise en correspondance des images, que l'on nomme aussi stéréoscopie est l'un des points clefs de la modélisation à partir d'images. C'est au cours de cette étape que l'on met véritablement en forme la scène. On utilise des algorithmes de stéréoscopie pour extraire et mettre en correspondance les projections des objets qui forment cette scène, tout en appliquant des contraintes. Ces contraintes vont empêcher d'effectuer certaines erreurs, mais elles vont surtout guider la reconstruction de la scène. Il existe un grand nombre d'approches pour résoudre le problème de mise en correspondance des images, chacune d'entre elles se focalisant sur une caractéristique de la scène à reconstruire ou de l'application. Il n'est pas possible de définir une solution qui puisse prendre en compte complètement tous les phénomènes liés au système. C'est un problème d'autant plus difficile que l'on utilise des images de scène réelles sans aucun contrôle sur les conditions d'acquisition.

2.2.1 Un exemple, le système binoculaire humain

Certaines publications ont cherché à mettre en évidence une partie des stratégies du système de vision humain pour résoudre le problème de mise en correspondance. D. Marr, T. Poggio [MP77], J. Mayhew et P. Frisby [MF81] ont étudié le comportement de ce système de vision naturel et déterminé des contraintes (voir 2.2.3) permettant de réduire l'espace des solutions et d'éviter certaines configurations impossibles ou peu probables de scène.

Même si le processus humain est présenté comme une référence dans la majorité des publications, ce système de vision peut être trompé comme le présentent les exemples de la

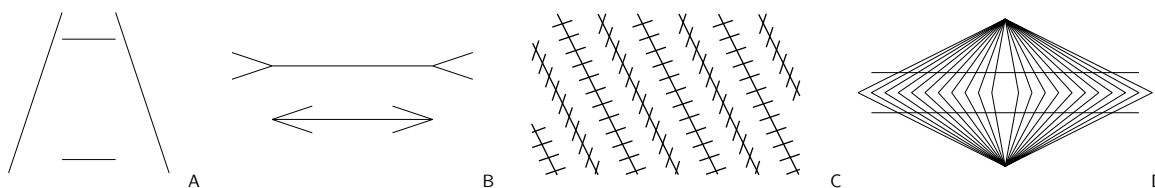


FIG. 2.9 – Exemple de figures géométriques qui mettent le système humain en erreur: *A* Illusion de Ponzo, le segment horizontal du haut paraît plus long que celui du bas. *B* Illusion de Muller-Lyer, le segment en haut paraît plus long que celui du bas. *C* Illusion de Zöllner, les lignes obliques ne semblent pas parallèles. *D* Illusion de Hering, les lignes horizontales semblent incurvées.

figure 2.9. Ces figures géométriques trompent notre perception de la longueur, du parallélisme ou de la rectitude. Ces erreurs sont dues à la capacité du système humain à intégrer l'environnement d'un objet pour le décrire et le comprendre. On trouve un autre exemple simple dans [Aya89] où l'auteur présente une pièce vide. On intègre immédiatement la profondeur et la géométrie de cette pièce au moyen d'une seule image. L'auteur nous présente ensuite la même scène, mais cette fois avec une personne au milieu de la pièce. On s'aperçoit alors que cette pièce est de taille réduite, l'idée que l'on s'en était faite était erronée. Deux notions importantes peuvent être extraites de ce dernier exemple. La première est que le système de vision humain utilise en plus d'une stratégie de mise en correspondance (dans le système binoculaire) une méthode de perception basée sur un apprentissage ou une expérience. Il associe la scène observée dans une image à ce que l'on voit et décode tous les jours. Le système utilise cet a priori pour associer une information tridimensionnelle à l'image, qui malheureusement est erronée dans cet exemple. La seconde notion est que le système humain utilise aussi des objets de la scène comme référence pour retrouver une information de proportion entre tous ceux qui la composent. Dans la première image, tous les objets qui composent la scène (porte, horloge et proportions de la pièce) conduisent à déduire une certaine dimension de la pièce. Dans la seconde image, un homme apparaît et contrarie cette information. Le système lui attribue un niveau de confiance supérieur et réduit alors la dimension de cette pièce en conséquence.

Les exemples sur lesquels nous nous sommes attardés permettent de mettre en évidence l'utilisation par le système de vision humain de connaissances pour extraire la troisième dimension des images. De la même manière, la majeure partie des algorithmes de stéréo-vision sont aussi guidés par les connaissances sur la scène ou l'application (choix de primitives à extraire ou du critère de corrélation par exemple). Nous allons examiner dans la suite de cette section, les phénomènes qui vont gêner voire empêcher l'extraction de l'information tridimensionnelle.

2.2.2 Complications pour résoudre la mise en correspondance

La stéréoscopie consiste à reconstruire la scène à partir des images. Malheureusement, plusieurs éléments perturbent la formation de ces images. Le problème de mise en correspondance devient alors beaucoup plus complexe et certaines informations ou caractéristiques de la scène peuvent être perdues. La suite présente les éléments perturbateurs les plus importants.

Le bruit : Le premier élément perturbateur est le bruit. Celui-ci provient des éléments optiques utilisés lors de l'acquisition mais aussi de l'erreur de quantification due à la discrétisation du problème. Certaines conditions d'acquisition, mouvement de la caméra ou mauvais réglage de celle-ci, peuvent aussi contribuer à rajouter un effet de flou.

Luminance et Réflexion : Une seconde catégorie d'éléments pouvant nuire à la mise en correspondance est liée à l'éclairage. Tout d'abord la luminance des deux prises peut être différente (surtout si les deux images ne sont pas prises au même moment). La source de lumière qui éclaire la scène peut ne pas être diffuse. Enfin, les surfaces éclairées des objets ne réfléchissent pas uniformément la lumière reçue lorsqu'elles ne sont pas lambertiennes. Un autre facteur important est l'ombre des objets qui dépend directement de l'éclairage de la scène. La dynamique de la lumière reçue par le capteur pour ces régions est généralement très faible.

Le temps : Les deux images qui constituent notre paire stéréoscopique peuvent ne pas être prises au même instant. Lorsque l'on ne dispose que d'une caméra, que ce soit pour utiliser des paramètres intrinsèques fixes ou parce que l'on ne peut disposer que d'un seul système d'acquisition (imagerie aérienne ou satellitaire), les objets de la scène peuvent avoir été modifiés ou s'être déplacés. Les conditions d'éclairage ainsi que l'orientation des ombres peuvent aussi varier lorsque le laps de temps entre deux acquisitions est très important. En imagerie aérienne, c'est un paramètre très important. La plupart des erreurs liées au temps se retrouvent au niveau des véhicules qui se déplacent entre deux prises de vues successives, provoquant des erreurs d'appariement.

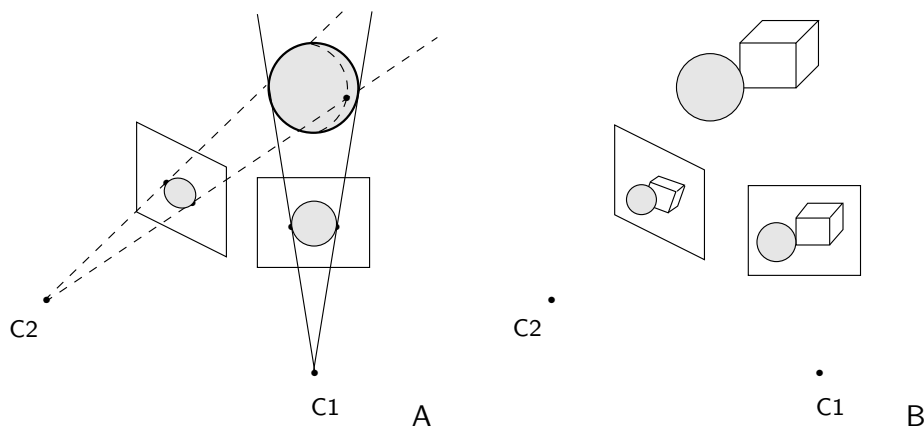


FIG. 2.10 – *Les occultations : A Problème des contours d'occultation avec un objet sphérique. Les contours des images ne correspondent aux mêmes points 3D de la surface de l'objet observé. B Problème des régions d'occultation, une partie du cube n'est pas visible dans l'une des images.*

Les occultations : Les principaux problèmes de la mise en correspondance proviennent des occultations, c'est-à-dire des zones que l'on aperçoit dans une image mais qui sont cachées dans la seconde. On rencontre deux sortes d'occultations, les contours d'occultation et les régions d'occultation qui sont présentés dans la figure 2.10.

Les contours d’occultation : Les contours d’occultation correspondent aux limites des objets de la scène. Des problèmes apparaissent avec les objets de forme sphérique. Comme le présente la figure 2.10 A, les contours d’occultation dans les deux images ne correspondent pas aux mêmes points 3D de la surface de l’objet. La mise en correspondance de ces contours va engendrer une erreur de reconstruction.

Les régions d’occultation : Les régions d’occultation correspondent aux zones de la scène que l’on observe dans l’une des images mais qui sont cachées dans la seconde. Les points qui appartiennent à ces régions de demi-occultation ne peuvent être appariés dans la seconde image car ils n’ont pas de correspondant. On ne peut donc leur attribuer aucune profondeur. Dans la figure 2.10 B, une partie de la face de devant du cube est cachée dans l’une des images.

2.2.3 Contraintes générales

Pour contrer ou compenser les effets perturbateurs que nous venons d’évoquer, la plupart des algorithmes de stéréo-vision utilisent des contraintes fortes sur la structure de la scène. Ces contraintes correspondent à des hypothèses réalisées sur la nature ou la forme des surfaces de la scène. Pour la majeure partie des scènes, les hypothèses ou contraintes qui vont être présentées sont nécessaires pour pouvoir reconstruire la scène. On suppose généralement que la radiométrie, l’échelle et l’orientation de ces surfaces sont comparables dans les différentes images.

Contrainte lambertienne : Les variations de l’intensité lumineuse reçue par le capteur permettent de percevoir les objets de la scène. La puissance de cette intensité reçue est directement proportionnelle à la puissance émise par la scène à l’endroit où l’objet observé se projette [Hor86]. Pour la mise en correspondance, il est important que les intensités reçues pour un point 3D de la scène soient identiques dans toutes les images. Pour que les images d’un point 3D aient la même intensité, la surface de l’objet étudié doit satisfaire la contrainte lambertienne (réflecteur diffus). Cette contrainte suppose que les surfaces doivent en chaque point être lambertiennes, c’est-à-dire que la luminosité réfléchie ne dépende pas de la position d’observation. Ceci est rarement vrai, il faudrait donc pouvoir tenir compte de ces phénomènes de réflexion pour reconstruire correctement la scène. Il est toutefois difficile de représenter correctement la diversité des surfaces présentes dans la scène sans l’avoir modélisée auparavant : béton, verre, arbre, ... Les propriétés de ces objets sont totalement différentes. Lors de l’étape de mise en correspondance, les algorithmes de stéréoscopie essaient de compenser l’erreur réalisée en extrayant des informations peu sensibles à cette perturbation (utilisation des contours, critères de corrélation, ...).

Contrainte de continuité et contrainte figurale : La contrainte de continuité suppose que les surfaces des objets observés soient toutes continues (C_0). Certains algorithmes supposent aussi que ces surfaces soient lisses (C_1). Malheureusement, les objets peuvent s’occulter et l’orientation de ces surfaces peut être discontinue. La disparité, qui permet de déterminer la profondeur d’un point 3D par rapport à l’une de ses images, est donc continue par morceaux. La disparité va être continue le long des surfaces visibles et discontinue aux lieux correspondant aux zones d’occultation comme nous l’avons présenté dans 2.1.5. La contrainte de continuité consiste à imposer aux surfaces une continuité locale. Un autre moyen d’imposer la continuité des objets est d’utiliser la contrainte figurale. Considérant

que les contours des images correspondent à des courbes physiques 3D, l'idée est d'imposer la continuité le long des contours des images.

Unicité : L'unicité est l'une des contraintes les plus utilisées dans les algorithmes de stéréo-vision. Elle consiste à imposer un seul appariement par primitive, un pixel de l'image de gauche ne peut avoir au plus qu'un seul correspondant dans l'image de droite. Cette contrainte résulte des deux hypothèses suivantes :

- les objets de la scène sont opaques,
- deux primitives distinctes ne peuvent être alignées avec le centre optique de la caméra.

Contrainte d'ordre : Certaines scènes tridimensionnelles étudiées obéissent à la contrainte d'ordre (ou contrainte de monotonie) : si un objet **a** est à droite d'un objet **b** dans l'image de gauche, alors l'objet **a** sera à la droite de l'objet **b** dans l'image de droite. Quelques configurations peuvent violer cette règle, notamment en robotique.

Contrainte fronto-parallèle : La plupart des algorithmes de stéréoscopie supposent que la surface observée est fronto-parallèle (parallèle aux plans rétiniens des deux caméras). Cette contrainte permet d'assurer que les voisinages autour des projections d'un point 3D sont identiques dans les deux images. Cette contrainte n'a pas besoin d'être strictement vérifiée. Toutefois, lorsque l'inclinaison de la surface observée devient trop importante, il est nécessaire d'en tenir compte lors de la mise en correspondance en déterminant les déformations locales de la surface lors de sa projection dans les deux images comme le fait la corrélation fine présentée par *F. Devernay* et *O. Faugeras* dans [DF00].

Domaine de disparité : Dans la plupart des applications on peut estimer les profondeurs maximales et minimales de la scène. Cet espace autorisé peut correspondre à un domaine de variation de disparité. L'utilisation de cette contrainte va restreindre l'espace de recherche des appariements, ce qui permet à la fois de limiter les risques d'erreurs et de diminuer le temps de calcul.

Contrainte du gradient de disparité : Cette contrainte est tirée d'expériences psychophysiques réalisées avec le système humain. Le principe est de limiter la norme du gradient de disparité. Cette contrainte est équivalente à éliminer les surfaces de forte pente de la scène lors de l'appariement.

Contrainte épipolaire : La contrainte épipolaire est directement issue de la géométrie épipolaire de notre système de caméras (voir 2.1.2 et figure 2.5). Cette contrainte géométrique permet de réduire l'espace de recherche à une droite au lieu de la totalité du plan image lors de la mise en correspondance de points. Elle nécessite toutefois le calibrage du système. Certains algorithmes utilisent aussi une bande épipolaire au lieu d'une droite, pour compenser certaines imprécisions du calibrage.

2.3 Les principales approches de la stéréoscopie

Nous allons étudier les principales approches permettant de résoudre le problème de la mise en correspondance des images. Ce problème a été l'objet de nombreuses recherches.

Les différentes approches se distinguent essentiellement par les contraintes utilisées et la formulation du problème. Nous avons présenté plusieurs contraintes liées à la scène et à la stéréoscopie (contrainte épipolaire, unicité, continuité . . .). D'autres contraintes sont aussi introduites par le choix de l'algorithme (approche surfacique ou énergétique) ou des éléments des images que l'on cherche à mettre en correspondance (points, segments, courbes, . . .).

Tous ces choix dépendent essentiellement du contexte : de la connaissance sur la scène étudiée et de l'application proposée (robotique mobile, photogrammétrie ou réalité virtuelle). Il est difficile de comparer des stratégies stéréoscopiques sans tenir compte du contexte. Toutefois, on peut regrouper presque toutes les stratégies de mise en correspondance dans trois classes distinctes : les méthodes de stéréoscopie à base de primitives (*feature-based*, section 2.3.2), les méthodes surfaciques (*area-based*, section 2.3.1) et les méthodes énergétiques (section 2.3.3). Il existe quelques articles présentant et comparant les principales méthodes [DA89, Kos93, Zha93].

2.3.1 Les méthodes surfaciques

Pour retrouver les images d'un point de l'espace 3D, les méthodes surfaciques utilisent le fait que leurs voisinages sont similaires. L'idée générale consiste à calculer une mesure de ressemblance entre les voisinages des points à apparier. Les méthodes surfaciques se proposent généralement de fournir une solution pour chaque point d'intérêt. Elles ont l'avantage sur les méthodes à base de primitives de fournir une représentation dense de la scène 3D observée. On note toutefois que ces méthodes rencontrent des difficultés avec les discontinuités de profondeur. Pour que les voisinages se correspondent, la scène doit respecter plusieurs contraintes : la contrainte lambertienne pour que les intensités soient comparables, la contrainte fronto-parallèle pour que les voisinages soient peu déformés et la contrainte de continuité pour que les voisinages correspondent à la même surface. La plupart des méthodes travaillent en géométrie standard (épipolaire). Dans la pratique, ces contraintes qui dépendent principalement de la nature de la scène n'ont pas besoin d'être strictement respectées. Dans la plupart des cas, les méthodes surfaciques sont peu sensibles au non-respect de ces contraintes (tant que l'on reste dans des configurations non critiques).

Méthodes surfaciques locales

Les méthodes surfaciques locales recherchent les couples de points homologues incrémentalement le long des droites épipolaires. Le principe général est le suivant : on se fixe un point d'intérêt dans l'une des images, à l'aide d'un critère de ressemblance à maximiser, on recherche le(s) point(s) homologue(s) lui correspondant dans le(s) autre(s) image(s) de telle manière que les contraintes locales imposées soient vérifiées.

Utilisation d'un critère de corrélation Une des méthodes classiques de mise en correspondance de points est l'utilisation d'un critère de corrélation entre des vecteurs d'attributs représentant les images au voisinage de ces points. Ces vecteurs d'attributs peuvent correspondre aux intensités des pixels ou à toute autre mesure discriminatoire permettant d'obtenir les bonnes correspondances quelle que soit la situation. La plupart des critères de corrélation (étudiés dans [AG92]) dérivent de deux mesures de base : la distance euclidienne entre les deux vecteurs (notée SSD, comme *Sum of Squared Differences*), et le produit scalaire de ces deux vecteurs (noté CC, comme *Cross-Correlation* ou corrélation croisée).

Le principe de l'algorithme par corrélation est de calculer pour chaque valeur de la disparité fixée le critère de corrélation. On décale les deux images de la valeur de la disparité, puis on calcule le critère. On ne garde ensuite que les couples de points ayant un score suffisant. Les principales difficultés surviennent avec les surfaces de forte pente, les surfaces non texturées ou avec des textures répétitives et les régions occultées et de forte pente.

Pour préserver les discontinuités de profondeur et les occultations, *P. Fua* propose dans [Fua91] d'ajouter une phase de validation croisée entre les cartes de disparité droite et gauche. Sous l'hypothèse que les discontinuités d'intensité correspondent soit à de fortes textures, soit à des discontinuités altimétriques ou d'orientation de surface, certaines méthodes vont essayer de s'appuyer sur les contours des images pour préserver les discontinuités de la scène. *C. Baillard* propose une stratégie en deux phases, une première commençant par mettre en correspondance les points de contours, puis une seconde qui met en correspondance les points compris entre les différents couples obtenus lors de la première phase [Bai97].

Un autre moyen de gérer les discontinuités de surface est de réajuster le support pour stabiliser l'appariement. *T. Kanade* et *M. Okutomi* proposent d'utiliser des fenêtres adaptatives dans [KO94]. Ils améliorent une carte de disparité initiale en utilisant une fenêtre dont la taille est définie par une analyse locale de la disparité courante et des intensités images. Dans [Lot96], *J.L. Lotti* utilise ces fenêtres dans le processus de mise en correspondance pyramidal. Considérant que les contours correspondent aux discontinuités de profondeur de la scène, l'expansion des fenêtres est stoppée par une carte de contours radiométriques. Exploitant aussi une carte de contours, d'autres approches utilisent une taille fixe de fenêtre munie d'un masque adaptatif défini par diffusion [Pap98], ou d'un masque adaptatif pondéré pour privilégier le voisinage central qui est moins sensible aux effets de distorsions dus aux surfaces 3D de forte pente [Cor98]. *D. Scharstein* et *R. Szeliski* réajustent le support dans une phase d'agrégation en utilisant une procédure de diffusion non uniforme dans [SS88]. Ils utilisent un algorithme simple de diffusion, équation de la membrane, associé à un critère local d'arrêt. La diffusion non linéaire résultante leur permet d'avoir un support adaptatif.

La plupart des algorithmes cités précédemment rencontrent des difficultés avec les zones fortement texturées puisqu'ils considèrent souvent toutes les discontinuités d'intensité comme des discontinuités de profondeur. De plus, toutes les approches utilisent la contrainte fronto-parallèle de la surface. Elles permettent donc de ne gérer qu'une partie des discontinuités de profondeur de la scène. *F. Devernay* propose dans [DF00, Dev97] la méthode de corrélation fine qui corrige une carte de disparité initiale en tenant compte de l'orientation de la surface 3D en chaque point. La précision de reconstruction est considérablement améliorée pour les surfaces de forte pente.

L'algorithme de Programmation Dynamique L'algorithme de programmation dynamique est un outil très utilisé pour résoudre le problème de mise en correspondance des droites épipolaires : [GLY95, SB94, OK85, Bai97]. Il offre une solution bien meilleure que celle obtenue avec un appariement primitive par primitive. La programmation dynamique est un algorithme de recherche permettant d'obtenir une solution optimale au sens du critère défini. Il est particulièrement adapté pour résoudre le problème de mise en correspondance tout en préservant un temps de calcul raisonnable. Il calcule un coût global des appariements le long de la courbe des appariements dans l'espace de mise en correspondance. Cette courbe permet de prendre en compte les contraintes d'unicité et d'ordre, tout en imposant une cohérence globale et en tenant compte des discontinuités et des occultations.

Lorsque la fonction de coût global est définie comme l'intégrale d'un coût local le long de la courbe, cet algorithme garantit l'optimalité de la solution calculée dans le domaine exploré. Le coût local défini pour chaque paire de pixels doit permettre de discriminer les faux appariements, pour lesquels le coût sera fort, des vrais appariements pour lesquels le coût sera faible. On peut donc définir par exemple un coût inverse au score de corrélation. Il faut toutefois faire bien attention en définissant le coût global de ne pas dépendre du nombre d'appariements composant le graphe; le chemin le plus court n'est pas toujours le meilleur. L'optimisation globale de l'algorithme permet d'éviter les problèmes de propagation des erreurs et de choix entre deux primitives possédant la même confiance.

Dans l'algorithme de base, on réalise une optimisation le long de chacune des deux droites épipolaires. *H. Ishikawa* et *D. Geiger* utilisent dans [IG98] une fonction de coût incorporant la plupart des contraintes existantes. Cette fonction modélise aussi les occultations et discontinuités, permettant d'en tenir compte lors de l'optimisation globale. Pour tenir compte du voisinage extérieur à la droite épipolaire, *Y. Ohta* et *T. Kanade* [OK85] proposent de combiner deux processus de recherche réalisés par l'algorithme de programmation dynamique : une recherche intra-scanline qui recherche les points homologues le long de droites épipolaires, et la recherche inter-scanline qui met en correspondance les contours connexes verticaux extraits de chacune des images. Ils construisent pour cela un espace 3D constitué par l'empilement des espaces 2D de mise en correspondance des droites épipolaires. Les auteurs alternent les deux recherches pour obtenir une solution globale plus consistante. *B. Serra* et *M. Berthod* ont proposé une version continue de l'algorithme de programmation dynamique dans [SB94] qui permet au graphe construit de posséder une précision sous-pixélique. L'algorithme proposé traite la mise en correspondance de contours et permet dans un même temps d'obtenir une mesure de déformation des deux contours appariés. *S. Intille* et *A. Bobick* proposent dans [IB95] une méthode de mise en correspondance utilisant la programmation dynamique dans les images des espaces de disparité (DSI, Disparity Space Image). L'utilisation des DSI leur permet de définir des points de contrôle qui limitent l'influence du bruit, les points de contours. Ces points de contrôle correspondent à des points de haute confiance et forcent le graphe à les utiliser.

Autres approches par surfaces

Les méthodes que nous venons de voir permettent de construire un ensemble de données tridimensionnelles non structurées. Leur interprétation directe est difficile. Il est souvent nécessaire de transformer ces données en ensembles plus compacts et structurés que l'on peut alors analyser et utiliser plus aisément. Il existe un grand nombre d'approches permettant de reconstruire les surfaces de la scène à partir de points 3D comme le présente la revue de *R. Bolle* et *B. Vemuri* [BV91]. Nous nous intéressons maintenant aux méthodes **stéréoscopiques** qui reconstruisent la scène 3D avec des surfaces. La section 5.1 s'intéressera à l'estimation des modèles de surface et la section 5.2 à leur reconstruction à partir de données tridimensionnelles. On peut considérer qu'il y a deux stratégies de reconstruction de la surface : celles qui réunissent les points ou primitives appartenant à une même surface puis estiment cette surface, et celles qui quadrillent l'espace de manière régulière, estiment les surfaces locales puis les regroupent.

W. Hoff et *N. Ahuja* proposent dans [HA89] une approche multi-résolution pour reconstruire la scène par un ensemble de surfaces quadratiques. Ils supposent que la scène est constituée de surfaces lisses et continues par morceaux. Leur approche intègre les étapes de mise en correspondance de points et d'approximation de surface (des points 3D extraits)

sur trois niveaux de résolution. A chaque niveau de résolution, ils appliquent un algorithme qui va reconstruire une carte de disparité correspondant aux surfaces lisses par morceaux calculées. La mise en correspondance dans une résolution plus fine est guidée par le résultat obtenu à la résolution précédente pour rendre le résultat plus cohérent. Plus récemment, *P. Fua* présente dans [Fua97] une méthode de reconstruction de surface 3D dans un système multi-vues. Il combine une représentation basée sur un système de particules (morceaux de surface 3D) à de l'estimation robuste et de l'optimisation. La méthode se décompose en trois étapes : après obtention d'une carte de disparité par corrélation, il initialise le jeu de particules dans l'espace 3D. Il affine ensuite la position de chacune des particules en minimisant une fonction d'objectif basée sur des mesures de corrélation. Enfin, il élimine les particules isolées et regroupe celles qui appartiennent à la même surface pour obtenir une représentation de la scène à partir de morceaux de surface.

2.3.2 Les méthodes à base de primitives

L'utilisation de primitives est très fréquente dans le domaine de la vision par ordinateur. Le choix ou la définition de ces primitives est généralement arbitraire, puisque l'utilisation de ces primitives de haut niveau permet la généralisation du problème. Toutefois, ces primitives doivent posséder quelques propriétés comme l'unicité, la répétabilité et la signification physique. L'utilisation de primitives comporte deux principaux avantages : la diminution du volume de données à traiter et l'apport d'informations supplémentaires. Cet apport est lié à la particularité de la primitive extraite qui peut permettre d'utiliser des contraintes supplémentaires de plus haut niveau (notamment sur la structure de la scène).

Étant donné que le but de la stéréoscopie est de reconstruire une scène en trois dimensions, les primitives utiles en stéréo sont celles comportant une information sur la structure 3-D de la scène. En général, et surtout dans les environnements construits par l'homme, la structure 3-D peut être décrite par des contours et des intersections de contours (coins et jonctions). Les contours peuvent être obtenus par un détecteur comme ceux de *Canny* [Can86] ou de *Deriche* [Der87]. Ce type d'information peut ensuite être utilisé pour générer des primitives de plus haut niveau comme les segments, les polygones ou les courbes splines. L'utilisation de ces primitives va permettre plus de flexibilité dans le choix de la méthode de mise en correspondance. La précision obtenue dépend de celle des primitives extraites de l'image. Un autre avantage des primitives basées sur les contours est leur insensibilité aux variations d'éclairage et aux changements de point de vue. Des difficultés vont être rencontrées lorsque les primitives extraites sont en nombre insuffisant pour représenter la scène ou lorsqu'elles ne correspondent à aucune information 3D. Le choix des primitives à utiliser est primordial pour une bonne reconstruction.

Une des méthodes permettant la mise en correspondance de primitives est la programmation dynamique qui a été présentée auparavant. Les algorithmes proposés dans [OK85, SB94, IB95] utilisaient des points de contours. Nous allons maintenant voir plusieurs méthodes de mise en correspondance utilisées pour l'appariement de primitives.

L'algorithme de relaxation L'algorithme de relaxation est un algorithme de recherche itératif. Le problème de mise en correspondance est alors interprété comme un problème d'étiquetage : à une primitive d'une image on associe au plus une primitive de la seconde image de telle manière que la paire formée vérifie certaines contraintes (notamment avec son voisinage). La vraisemblance de chaque appariement est représentée par une probabilité. A l'initialisation, on détermine les probabilités à l'aide de critères de ressemblance.

L'algorithme de relaxation va faire évoluer ces probabilités (que l'on associe aux nœuds d'un graphe) de manière itérative par rapport aux appariements voisins et aux contraintes de cohérence avec le voisinage. La différence entre les divers algorithmes de relaxation proposés repose essentiellement sur la règle de mise à jour des probabilités.

D. Marr et *T. Poggio* proposent dans [MP76] un algorithme basé sur les contraintes d'unicité et de continuité. L'algorithme est initialisé par tous les appariements qui satisfont la contrainte épipolaire. L'évolution des probabilités (poids dans leur cas) se caractérise par deux processus distincts : (1) un processus inhibiteur qui va réduire la probabilité de tous les appariements conflictuels (qui violent la contrainte d'unicité) et (2) un processus exciteur qui augmente celles dont la disparité est voisine. Les appariements de probabilité faible sont éliminés et l'algorithme converge en quelques itérations. *W. Grimson* a développé et testé cet algorithme dans [Gri81] en utilisant une approche multi-résolution. Il utilise dans son algorithme des points de contours extraits des images. Ses expérimentations l'ont amené à écrire une version modifiée de cet algorithme dans [Gri85] où la contrainte de continuité est remplacée par la contrainte figurale. *S. Pollard*, *J. Mayhew* et *J. Frisby* ont aussi utilisé l'algorithme de relaxation dans [PMF91]. Leur algorithme impose initialement la contrainte épipolaire et une contrainte d'orientation du contour. Ils imposent ensuite progressivement les contraintes d'unicité et de continuité figurale lors de l'évolution des probabilités. Leur algorithme fait aussi intervenir une contrainte de gradient de disparité.

Utilisation de graphes Dans les sections précédentes, nous avons vu des méthodes qui appliquent des contraintes de similarité ou de continuité au voisinage des primitives. Les relations entre les primitives d'une même image (proximité, parallélisme, ...) peuvent constituer un autre moyen de discrimination et d'appariement de primitives. On construit des graphes pour chaque image à partir de ces relations puis on résout le problème d'appariement des primitives par mise en correspondance de sous-graphes, tout en tenant compte des contraintes citées qui simplifient la résolution de ce nouveau problème. Nous allons maintenant voir deux approches qui permettent de résoudre ce problème.

Recherche de la clique maximale *R. Horaud* et *T. Skordas* proposent dans [HS89] de mettre en correspondance des segments de contours par la recherche de la clique maximale d'un graphe relationnel. Les segments extraits des images sont caractérisés par leur position et leur orientation. Ils construisent deux graphes relationnels, un par image, à partir des relations de voisinage des primitives segments. A partir de ces relations, ils construisent ensuite le graphe de correspondance dont les nœuds représentent tous les couples de segments valides. La mise en correspondance est réalisée en recherchant les cliques maximales de ce graphe, une clique maximale correspondant au jeu maximum de nœuds que l'on peut construire à partir d'une hypothèse choisie, c'est-à-dire un nœud de référence. Avec ce type d'approche structurelle une difficulté majeure apparaît lorsque des relations sont manquantes (notamment dû au processus d'extraction de segment). Il n'est plus garanti alors de trouver des jeux de nœuds totalement connectés. Pour pallier ce problème, les auteurs proposent alors d'associer une fonction de bénéfice (basés sur des calculs de ressemblance des segments) à chaque clique trouvée puis de choisir la ou les meilleure(s) clique(s) qui correspondra(ont) à celle(s) qui possède(nt) la fonction de bénéfice la plus élevée.

L’algorithme de prédiction-vérification *N. Ayache* et *B. Faverjon* ont proposé dans [AF87] un algorithme de mise en correspondance par prédiction et vérification. Le principe est un peu similaire à celui que nous venons de voir. Après avoir construit les deux graphes relationnels des segments, un par image, la phase de prédiction va tenter de générer au minimum une hypothèse d’appariement pour chaque primitive à partir de mesures de similarité. On va ensuite étendre le nombre des appariements lors de la phase de propagation. La propagation des hypothèses consiste à trouver récursivement un appariement pour chaque voisin de l’appariement initial. La phase de validation va comparer tous les appariements réalisés en appliquant les contraintes de continuité et d’unicité et en utilisant un algorithme de relaxation discrète. On sélectionne les appariements qui constituent la plus grande composante connexe.

O. Faugeras et *L. Robert* étudient dans [FR96] la prédiction des éléments d’une image à partir des primitives de deux autres images et de la géométrie épipolaire entre les trois images. Ils traitent les cas de prédiction de points, de lignes et de courbures. Plus récemment, *C. Baillard*, *A. Zisserman* et *A. Fitzgibbon* proposent dans [BZF99] une méthode de mise en correspondance de segments multi-images basée sur ce principe.

Autres approches par modèle Il existe d’autres approches par modèle telles que celle proposée par *L. Chen* et *W. Lin* dans [CL97]. Ils proposent une méthode de reconstruction de surfaces fermées obtenues en combinant des techniques de stéréoscopie par régions et par primitives. Une technique d’accroissement de régions segmente la scène tandis que les limites de ces régions sont déterminées à partir des primitives de contour mises en correspondance.

2.3.3 Les approches énergétiques

Les approches énergétiques constituent une troisième classe de méthodes permettant de résoudre le problème de mise en correspondance. La mise en correspondance est exprimée comme un problème de minimisation d’une énergie globale. Il existe deux principales techniques pour résoudre le problème : l’approche Bayésienne qui cherche à maximiser la probabilité d’appariement et l’approche variationnelle qui cherche à minimiser le bruit avec des contraintes pour régulariser le problème.

Les approches Bayésiennes Les approches Bayésiennes modélisent des grandeurs physiques par des lois de probabilité. Elles recherchent ensuite la solution de probabilité maximale. Le problème est généralement formulé de manière probabiliste en modélisant le processus de formation des images et la scène 3D. Les algorithmes bayésiens se distinguent principalement par le choix des modèles utilisés et le choix de la méthode d’estimation.

Considérons la géométrie S d’une scène 3D qui génère un couple d’images stéréoscopiques I_d et I_g . Le problème à résoudre est de chercher la géométrie de la scène la plus probable connaissant les deux images. On définit $P(S|I_d, I_g)$ la probabilité a posteriori de l’état du monde donné par les mesures comme suit :

$$P(S|I_d, I_g) = \frac{P(I_d, I_g|S)P(S)}{P(I_d, I_g)} \quad (2.9)$$

$P(S|I_d, I_g)$ représente la probabilité de l’espace 3D S reconstruit à partir des images stéréoscopiques droite et gauche : I_d et I_g . Le premier terme du numérateur, $P(I_d, I_g|S)$, est le modèle de formation des images, aussi appelé modèle des données (construction de I_d et

I_g à partir d'une scène 3D S). Le second terme du numérateur, $P(S)$, représente le modèle a priori de la scène S . Enfin, le dénominateur représente le modèle a priori des images. Ce dernier terme ne dépend pas de la géométrie de la scène et la probabilité qui lui est associée est donc considérée comme constante ou indépendante.

L'énergie fonctionnelle du modèle des données représente le processus de formation de l'image. Pour que le processus soit correctement modélisé, cette énergie doit prendre en compte la possibilité que le point de la scène puisse être non visible dans l'une des images du système de caméra. *P. Belhumeur* propose dans [Bel96] un modèle de formation des images dérivé de la fonction de disparité du modèle cyclopéen qui prend en compte les occultations. *D. Geiger, B. Ladendorf* et *A. Yuille* proposent dans [GLY95] une autre méthodologie qui modélise le processus de mise en correspondance au lieu du processus de formation des images. Ils utilisent aussi un processus d'occultation pour chaque vue. Même s'ils dérivent d'une approche différente, les deux modèles présentés ont la même forme. L'énergie fonctionnelle du modèle a priori représente les connaissances dont on dispose sur la scène 3D. Dans [GLY95], les auteurs imposent que les cartes de disparité, droite et gauche, soient des fonctions continues par morceaux. *P. Belhumeur* présente une approche intéressante pour définir un modèle des connaissances dont on dispose sur l'image. Il représente le modèle sous la forme d'un processus aléatoire et définit progressivement trois mondes. Le premier monde qu'il présente est celui des surfaces lisses. Le second monde va prendre en compte la possibilité qu'il y ait plus d'un objet dans la scène (ajout des discontinuités de profondeurs). Le troisième monde va prendre en compte le fait que certaines surfaces d'objets peuvent être linéaires par morceaux le long d'une ligne épipolaire (ajout des discontinuités d'orientation). Les auteurs de ces différentes approches procèdent ensuite à la minimisation de l'énergie totale composée par les deux modèles, le modèle de formation des images et le modèle à priori. Dans [GLY95, Bel96], les auteurs utilisent des versions modifiées de la programmation dynamique tandis que dans [IG98, RC98] ils préfèrent résoudre le problème sous la forme d'une recherche de flot maximum pour contrôler le niveau de lissage de la surface.

C. Steward, R. Flatland et *K. Bubna* présentent dans [SFB96] deux algorithmes de reconstruction de surface à partir d'images stéréoscopiques. Sous l'hypothèse d'équiprobabilité directionnelle des vecteurs normaux aux surfaces locales de la scène, les auteurs déterminent la densité de probabilité de l'orientation de la surface de disparité. Ils introduisent ensuite trois autres densités qu'ils utilisent pour comparer et analyser les techniques de stéréoscopie courantes. Ils présentent ensuite leurs deux algorithmes fondés sur une approche Bayésienne.

Les approches variationnelles Les approches variationnelles transforment le problème de mise en correspondance en un problème d'optimisation d'une énergie. Ce type de problème est mal posé. L'une des premières approches de régularisation classiques est celle présentée par *Tikhonov* [TA77] qui consiste à lisser la solution, donc les discontinuités du problème autrement dit les discontinuités de la scène.

J. Shah propose dans [Sha93] une approche variationnelle de la stéréoscopie gérant à la fois les discontinuités de disparité et les régions occultées. Sachant qu'à une discontinuité de disparité dans une image correspond une zone d'occultation dans la seconde, *J. Shah* propose de minimiser deux fonctionnelles couplées en alternance, une pour chaque image. Il fixe les lieux de discontinuité de l'image de droite (et donc les régions d'occultation de l'image de gauche), et minimise la fonctionnelle de gauche pour trouver les lieux des discontinuités dans cette image et vice-versa. Les fonctionnelles sont couplées puisque les discontinuités

trouvées dans une image déterminent les régions d'occultations pour la seconde.

L. Robert et *R. Deriche* définissent dans [RD95] une approche très différente qui dérive des études variationnelles consacrées habituellement à la restauration d'image. L'amélioration et la restauration d'images furent les premiers problèmes en traitement d'image à recueillir une grande attention de la part des communautés scientifiques. Depuis quelques années, beaucoup se sont intéressés à préserver les discontinuités présentes dans les images. De nombreuses fonctions de régularisation ont été introduites pour répondre à ce problème. C'est ainsi que l'on a défini et étudié différents opérateurs de diffusion anisotrope² pour préserver les contours. L'idée de *L. Robert* et *R. Deriche* dans [RD95] a été d'utiliser les travaux consacrés à ce domaine, notamment [DF95], dans d'autres applications telles que la stéréoscopie qui peuvent se modéliser par une **Équation à Dérivées Partielles (EDP)**.

R. Keriven, *O. Faugeras* proposent dans [FK98] une nouvelle méthode de résolution du problème de stéréoscopie à partir d'un nombre quelconque d'images. Utilisant le principe variationnel que doivent satisfaire les surfaces de la scène et leurs images, les auteurs définissent un ensemble d'EDP's servant à faire évoluer une surface 3D initiale vers les objets de la scène. La résolution du système est réalisée à l'aide des surfaces de niveaux (Level Sets) permettant de prendre en compte les changements de topologie de celles-ci. Les auteurs présentent une implémentation 2D de la méthode tandis que *J. Gomes* et *O. Faugeras* proposent dans [GF99] une nouvelle EDP et son implémentation tridimensionnelle.

2.3.4 Quelques concepts généraux

Nous avons vu jusqu'à présent plusieurs approches permettant de résoudre le problème de mise en correspondance. Nous allons voir maintenant deux concepts plus généraux qui permettent aussi de résoudre ce problème : les différentes formes des approches hiérarchiques et les cas de stéréoscopie avec trois caméras ou plus. La plupart des approches que nous avons vues peuvent être facilement modifiées pour utiliser ces concepts.

Les approches hiérarchiques Les approches hiérarchiques sont les approches de mises en correspondance qui utilisent une structure de calcul ou de résolution hiérarchique. Ces algorithmes utilisent cette structure hiérarchique pour garder une consistance globale sur la carte de disparité construite. Il existe trois types d'approches hiérarchiques. La première approche, qui est la plus courante, est celle qui consiste à résoudre le problème étape par étape ou niveau par niveau, on parle d'**optimisation hiérarchique**. On commence à résoudre le problème à un niveau de résolution bas puis on utilise le résultat obtenu pour initialiser la résolution du problème de niveau supérieur. Ces méthodes sont appelées **multi-résolution**. La seconde approche diffère légèrement de la première puisque l'on modélise le problème à plusieurs niveaux de résolution mais on procède à une optimisation simultanée pour résoudre le problème. Ces méthodes sont dites **multi-échelle** mais on parle aussi de **modélisation hiérarchique**. La dernière approche consiste à résoudre le problème en imposant une hiérarchie sur les composantes ou primitives à mettre en correspondance. On parle de **hiérarchie de primitives**.

Optimisation hiérarchique L'idée conductrice de cette classe de méthodes est de résoudre petit à petit le problème de mise en correspondance ou de construction de la carte

2. Forte diffusion dans les zones à faible gradient et faible diffusion dans les zones à fort gradient qui correspondent aux discontinuités.

de disparité. Le problème de mise en correspondance correspond, dans les cas où l'on opte pour une modélisation fidèle du processus, à une optimisation d'un critère non convexe. La solution que l'on obtient va donc fortement dépendre de la solution initiale que l'on utilise. Pour pallier ce problème, on génère des images de plus basse résolution que l'on va utiliser pour approcher plus facilement la solution optimale du problème. Commencant au niveau de résolution le plus bas, on obtient une solution qui va servir à initialiser l'optimisation au niveau de résolution supérieur. On recommence jusqu'à finalement résoudre le problème au niveau de résolution des images initiales avec une solution initiale proche de la solution optimale souhaitée du problème.

Les premières approches d'optimisation hiérarchique, comme celle proposée par *S. Barnard* [Bar89], utilisent une technique d'optimisation stochastique hiérarchique qui résout le problème à chaque niveau de résolution et propage la carte de disparité solution. On appelle ce type de résolution *Coarse to fine resolution*. Il permet de résoudre le problème d'optimisation plus rapidement et avec une plus grande cohérence spatiale. Les approches plus récentes utilisent des stratégies plus complexes permettant de remettre en cause les valeurs obtenues à un niveau supérieur. *W. Hoff* et *N. Ahuja* proposent dans [HA89] une approche multi-résolution intégrant les étapes de mise en correspondance et d'interpolation (des points 3D extraits) sur trois niveaux de résolution. La mise en correspondance dans une résolution plus fine est guidée par la carte de disparité composée de surfaces lisses par morceaux pour rendre le résultat plus consistant.

Modélisation hiérarchique L'idée conductrice de cette classe de méthodes est de modéliser le problème sur plusieurs résolutions. On obtient ensuite la solution optimale en une seule optimisation. Pour cela, il faut pouvoir définir le signal dans un espace multi-échelle. Les premières approches utilisent une fonction de coût qui dépend de toutes les résolutions utilisées [Ter86]. Des travaux récents se sont portés sur cet espace [Lin94, DF96] et notamment l'équipe du CEREMADE [AGLM92, ALM92] surtout dans le cadre de la restauration d'images.

Hiérarchie de primitives Cette dernière catégorie de méthodes est basée sur une résolution du problème de mise en correspondance en imposant une hiérarchie des primitives (ou modèles) à apparier. On commence par apparier les primitives de plus haut niveau, puis celles du niveau inférieur et ainsi de suite. Ces méthodes sont très employées pour l'extraction de bâtiments, comme nous le verrons dans le chapitre 3.

Nous avons vu plusieurs approches hiérarchiques pour la mise en correspondance d'images. Il existe encore d'autres stratégies que l'on pourrait classer comme hiérarchiques. Pour les approches énergétiques, les optimisations des énergies peuvent être résolues progressivement. L'approche de *P. Belhumeur* et ses différents mondes est aussi une forme de hiérarchie.

Système trinoculaire Deux images d'une même scène permettent de reconstruire une scène. Il suffit de retrouver dans chacune des images les points qui se correspondent. L'utilisation de prises de vues supplémentaires de la scène permet de rajouter de l'information 3D et donc d'aider la mise en correspondance inter-images. Cet apport d'information permet d'obtenir une redondance des informations. Il sera plus facile de régler les problèmes de vision binoculaire : occultation (en général l'ajout d'une caméra peut améliorer les résultats suivant les points de vue utilisés), ambiguïtés d'appariements, réflexions spéculaires (si en une région l'une des deux premières images est 'faussée', la troisième peut remplacer la

source d'information) ... Cet apport d'information peut contribuer énormément au processus de mise en correspondance. Son importance va dépendre du point de vue de chaque caméra qui n'est pas obligatoirement choisie par l'utilisateur.

La plupart des algorithmes de vision trinoculaire ou multi-images se servent de la ou des images supplémentaires comme moyen de vérification ou de sélection lors de la mise en correspondance des deux premières images (comme le présentent *O. Faugeras* et *L. Robert* dans [FR96]). Dans le cas de la vérification (voir [DA89]), la contrainte trinoculaire ne servira qu'à départager les ambiguïtés d'appariement. Dans le cas de la sélection, la contrainte trinoculaire est utilisée directement pour l'appariement [Aya89]. On privilégie ainsi les contraintes géométriques aux contraintes heuristiques telles que la continuité. De cette manière, on peut simplifier la recherche de points homologues. On obtient dans un même temps une précision meilleure puisque l'on utilise trois mesures au lieu de deux.

Plus de trois caméras Avec plus de trois caméras, il est préférable d'utiliser une approche qui ne privilégie aucune caméra. Une première approche est d'apparier tous les couples d'images puis de fusionner les résultats. *L. Gabet* présente dans [GGR94] une méthode de reconstruction de MNT dont la fusion est basée sur un vote majoritaire. Une seconde approche proposée par *M. Okutomi* et *T. Kanade* dans [OK93] permet de faire de la corrélation multi-images. Avec un système de plusieurs caméras alignées, les auteurs présentent une stratégie permettant de fusionner les courbes de corrélation calculées pour différents couples. On commence par rectifier les images deux à deux, on calcule ensuite les courbes de corrélation de chaque couple, on exprime enfin les courbes dans le même référentiel pour en extraire le maximum de la courbe somme. *D. Canu* propose dans [Can97] une version généralisée permettant d'utiliser un système de caméras avec des positions quelconques. Nous verrons dans le chapitre 3 d'autres approches qui ont largement traité ce problème dans la cadre de l'imagerie aérienne.

2.4 Conclusion

Nous avons présenté dans ce chapitre les notions de base d'un système de vision stéréoscopique. Nous avons étudié chacune des étapes qui composent ce système en présentant des références : modélisation de la caméra, modélisation d'un système de caméras, calibrage du système, rectification des images et reconstruction de la scène. Nous avons ensuite traité plus en détail les algorithmes de mise en correspondance des images. Nous nous sommes particulièrement intéressés à ces méthodes afin de développer une stratégie qui puisse permettre la mise en correspondance des façades des bâtiments de la scène. Cette étude est présentée dans le chapitre 4.

Chapitre 3

Reconstruction de bâtiments

Disposer de descriptions 3D d'une scène urbaine devient un besoin sans cesse croissant. De nombreuses applications comme la planification des villes ou la cartographie poussent les recherches liées à ce domaine. Mais c'est surtout l'augmentation du nombre d'utilisateurs de téléphones mobiles qui a accru l'intérêt de ces bases de données géographiques numériques. Les fournisseurs de service en télécommunications ont besoin de Modèles Numériques Urbains (MNU) précis pour étudier la propagation des ondes transmises et reçues par leurs cellules (antennes). Ces modèles de villes permettent d'améliorer et de faciliter le développement de leurs réseaux de télécommunications. Certaines applications des MNU (en particulier la téléphonie) nécessitent de connaître en plus de la forme des bâtiments, leurs matériaux (verre, acier, pierre, béton), information qui peut être obtenue à partir d'images des façades prises au sol. Dans ce contexte, de nombreux instituts de recherche ou entreprises ont focalisé leur recherche sur la réalisation de ces modèles. Le chapitre présent traite de l'imagerie aérienne et des modèles numériques obtenus à partir de ces images. Nous présentons ensuite un état de l'art des principales recherches traitant la reconstruction des modèles de bâtiments.

3.1 Contexte

Cette thèse s'est déroulée avec un contrat CIFRE entre le projet Robotvis de l'INRIA (Institut National de Recherche en Informatique et Automatique) de Sophia-Antipolis et la société ISTAR (Imagerie STéréo Appliquée au Relief). Le projet Robotvis s'est spécialisé depuis de nombreuses dans le domaine de la vision, notamment sur les problèmes liés à la modélisation de scènes tridimensionnelles. ISTAR est une société de hautes technologies produisant des bases de données cartographiques numériques, principalement pour le domaine des télécommunications. Son département de recherche a développé des outils et des chaînes de traitement visant à reconstruire presque automatiquement des modèles numériques de terrain (MNT), des modèles numériques d'élévation (MNE) et des ortho-images géo-référencées. Ces modèles numériques sont reconstruits à partir de cartes scannées ou de capteurs aériens et spatiaux : SPOT, LANDSAT, imagerie analogique aérienne, capteur HRSC d'ISTAR, ...

Depuis plusieurs années, la reconstruction automatique des bâtiments est un sujet de recherche complexe et non résolu qui intéresse non seulement les industriels comme ISTAR mais aussi les laboratoires universitaires. Actuellement, le modèle vectoriel de ville réalisé par ISTAR est construit par des opérateurs photo-interprètes qui détournent les bâtiments

manuellement dans les ortho-images et le MNE brut. ISTAR s'intéresse depuis longtemps à la reconstruction de modèles de bâtiments comme beaucoup d'autres industriels. Le but de cette thèse CIFRE est à la fois l'étude des recherches déjà effectuées dans le domaine et le développement d'outils visant à automatiser le processus de reconstruction des bâtiments. L'application principalement visée pour ces travaux est le domaine des télécommunications. Ce chapitre présente un état du problème posé et des principales publications dans le domaine de la reconstruction des bâtiments. Les chapitres suivants traiteront des recherches et études réalisées au cours de cette thèse.

3.2 De l'acquisition aux modèles numériques

L'acquisition : Le processus de modélisation des bâtiments n'est que l'étape finale d'un système plus complet tel que nous l'avons vu dans le chapitre 2 : acquisition, calibrage, mise en correspondance et une dernière étape nommée reconstruction. La première étape de ce système est l'acquisition des données. On dispose généralement de trois types de données : les cartes ou les SIG (Système d'Information Géographique), les données satellitaires et les données aériennes.

La méthode de modélisation peut s'appuyer sur des cartes numérisées de villes (comme [L99, RM97]). Les problèmes liés à cette source d'information sont la précision des bâtiments reconstruits, leur pérennité liée à la date de mise à jour des cartes et surtout le manque d'information altimétrique pour ces bâtiments, qui est pourtant un paramètre essentiel. Pour pallier à ces inconvénients, on préfère utiliser des systèmes de capteurs d'image. Ces capteurs peuvent avoir des origines diverses : radar, optique, infrarouge ou laser, nous permettant d'obtenir des informations de natures différentes. Les capteurs optiques que nous avons utilisés pour ces travaux sont les plus employés pour la reconstruction des bâtiments (suivis des capteurs laser). Il en existe deux catégories : les capteurs analogiques et les capteurs numériques. Les capteurs satellitaires sont le plus souvent numériques tandis que la majeure partie des capteurs optiques aériens sont analogiques. Le résultat des acquisitions analogiques est un film que l'on numérise à un pas choisi, directement lié à la résolution planimétrique des images que l'on souhaite obtenir.

L'acquisition d'un chantier aérien est un problème complexe. La qualité du modèle final dépend considérablement des conditions de cette acquisition. Il faut à la fois satisfaire les conditions climatiques et les contraintes administratives. Les clichés doivent être obtenus en une seule fois pour éviter les problèmes dus au changement de luminosité, à la variation des ombres et aux modifications de la scène (construction ou destruction de bâtiments). En imagerie aérienne, on possède toutefois l'avantage de pouvoir choisir la disposition des caméras. La configuration des caméras est choisie pour minimiser au maximum les imperfections dues aux multiples échantillonnages (numérisation et mise en géométrie épipolaire) et utiliser au mieux la surface des films. L'acquisition des photos se fait à une fréquence régulière le long de traces linéaires et parallèles de manière à disposer d'un système de caméras correspondant à une grille carrée régulière (figure 3.1). La visée est verticale. Les clichés sont orientés le long de la trace pour exploiter au mieux la forme carrée de l'image fournie par le capteur analogique. Le recouvrement des images est de 80% le long des lignes et colonnes des images (le long de la trace et perpendiculairement à cette direction). Après l'étape d'acquisition, il faut numériser les films. C'est une opération délicate qui est source de problèmes supplémentaires. Le choix de la disposition des caméras nous permet théoriquement d'être directement en géométrie épipolaire après numérisation. Les couples d'images

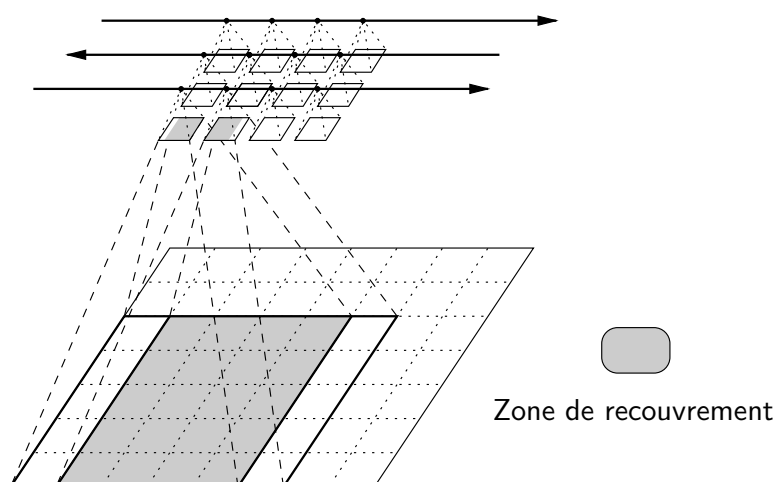


FIG. 3.1 – L'acquisition d'un chantier aérien : la disposition des photographies est choisie pour avoir une redondance d'informations altimétriques. Les recouvrements choisis à ISTAR sont de l'ordre de 80% le long de la trace et perpendiculairement à cette direction.

rectifiées pourront être obtenus à partir des images initiales avec peu de déformations.

Le calibrage : Le but du calibrage est de déterminer tous les paramètres du système d'acquisition (section 2.1.3) mais aussi de déterminer les positions géographiques des capteurs du système d'acquisition dans un référentiel absolu connu. Le but reste d'obtenir une cartographie numérique de la zone étudiée. On utilise généralement des points d'appui (ou GCP : Ground Control Points) qui sont des points remarquables sur le sol. On s'aide de cartes topographiques pour avoir une estimée initiale du terrain et choisir ces points caractéristiques. On peut aussi utiliser des points GPS (Global Positioning System) pris par des opérateurs dans des lieux remarquables. Pour renforcer la géométrie des couples d'images, on utilise aussi des points homologues (points appariés entre les images). Ces points peuvent être extraits automatiquement ou manuellement. On calibre tout le système d'acquisition pour le géo-référencer et connaître sa géométrie. L'étape de calibrage permet d'estimer et de corriger dans un même temps toutes les imperfections du système d'acquisition (erreurs de position et d'attitude des caméras, déformation due à la numérisation . . .) pour obtenir des couples d'images en géométrie épipolaire. Pour que le système soit bien modélisé, on doit s'assurer de deux conditions. Tout d'abord le modèle choisi doit être suffisamment souple pour pouvoir représenter le système et s'adapter aux déformations engendrées par les imperfections de l'acquisition. Ensuite, les points utilisés doivent être suffisamment dispersés dans la scène pour que le support soit représentatif.

La mise en correspondance : L'étape suivante est celle de la stéréoscopie, comme nous l'avons vu dans 2.2 et 2.3. Cette étape est l'une des plus importantes puisqu'elle permet une première construction de la scène que nous nommons communément MNE brut (Modèle Numérique d'Élévation Brut). Avec des images de basse résolution on obtient les caractéristiques du sol (montagnes, lac . . .), on reconstruit un MNT (Modèle Numérique de Terrain). On filtre généralement le MNE pour supprimer les erreurs importantes dues à de faux appariements. Le système d'appariement d'ISTAR permet de construire ce MNE brut à partir

d'un grand nombre d'images. Toutes les vues étant calibrées, le MNE brut est obtenu par la fusion robuste de plusieurs MNE locaux, chacun d'eux correspondant au résultat de l'appariement de deux images voisines (voisinage direct le long de la trace et perpendiculairement à cette direction). Pour l'appariement de chaque paire d'images, on utilise un algorithme de corrélation multi-résolutions. La redondance des informations permet d'éliminer les erreurs d'appariement ou le manque d'informations présents dans les résultats de certains couples.

Reconstruction d'un modèle urbain exploitable : La dernière étape est la reconstruction. Le but est de construire le modèle numérique exploitable par différentes applications. Lorsqu'on s'intéresse aux villes ou aux bâtiments, on utilise des images avec des résolutions plus élevées. Dans ce cas, on reconstruit un MNU (Modèle Numérique Urbain) représentant le plus fidèlement possible les objets de cet environnement. Certaines applications comme la liaison point à point haut débit, LMDS en télécommunications (Local Multipoint Distribution Service), demandent l'enveloppe tridimensionnelles de la scène (points les plus hauts). On peut alors utiliser un MNE brut ou un MNE brut filtré pour générer le MNU exploitable par cette classe d'applications. Pour des applications comme l'étude de la propagation des ondes dans les villes, il est nécessaire d'avoir des données beaucoup plus propres et surtout plus représentatives de la structure de la scène. Tout comme pour les applications de réalité virtuelle, on souhaite connaître la forme exacte des bâtiments et surtout la position et l'orientation des façades des bâtiments. Le modèle de chaque bâtiment ou son détournement exact est nécessaire pour reconstruire un MNU exploitable par ce type d'application.

3.3 La reconstruction des bâtiments, un problème complexe

Les problèmes de la stéréoscopie : Dans le chapitre précédent, la section 2.2.2 présentait les multiples facteurs de perturbation d'un système stéréoscopique. En imagerie aérienne, on retrouve la plupart de ces problèmes. Le bruit est un élément perturbateur important. Il provient de l'optique, du système de numérisation et des conditions de vol lors de la prise de vue dans l'avion. Le bruit peut aussi provenir des conditions atmosphériques, notamment lorsqu'il y a du brouillard ou un nuage. Les conditions d'éclairage et le laps de temps entre deux prises de vues sont aussi très importants. Il faut minimiser le temps d'acquisition total des prises de vues pour que la luminosité ambiante et les ombres des bâtiments restent identiques dans les différentes images. Seuls les véhicules qui se déplacent peuvent poser des problèmes, ils n'ont pas souvent la même position dans les différentes images de la scène.

Occultations, surfaces de forte pente et discontinuités d'orientation des objets de la scène : Comme le présente la figure 3.2, une scène urbaine est principalement composée d'un sol et d'une multitude d'objets correspondant au sur-sol (arbres et bâtiments). Trois difficultés liées aux systèmes stéréoscopiques vont prendre une très grande importance avec ce type de scène. Tout d'abord, tous les objets du sur-sol vont engendrer des régions d'occultation sur le sol d'une part, mais aussi entre les objets lorsque ceux-ci sont très proches (arbres avec bâtiments ou bâtiments entre eux dans les zones très denses). Toutes ces régions d'occultation vont compliquer le problème. La deuxième difficulté se trouve au niveau des façades des bâtiments. Toutes les visées des acquisitions sont presque verticales. Les façades sont donc des surfaces de très forte pente qui ne pourront pas être mises en cor-



FIG. 3.2 – *Exemple de scène aérienne.*

respondance directement à cause de la contrainte fronto-parallèle. Elles vont correspondre à des régions non appariées car trop déformées. La dernière des difficultés se trouve au niveau des discontinuités d'orientation des surfaces des objets, autrement dit les arêtes des bâtiments. Ces discontinuités vont se répercuter dans les cartes de disparité et vont aussi compliquer le problème.

Complexité des données pour une scène urbaine : Du fait de la haute résolution nécessaire pour reconstruire les bâtiments, de nombreux détails sont présents dans les images (cheminées, voitures ...), la scène urbaine étudiée est complexe. La complexité des données pour une scène urbaine dépend à la fois :

- du nombre d'objets qui composent la scène : arbres, voitures, bâtiments ... Le nombre d'objets est directement lié au contenu de la scène (zone rurale, urbaine ou militaire), et aux conditions d'observation (résolution, saison, qualité ...). Dans un environnement urbain, le nombre d'objets est considérable.
- de la diversité des objets qui composent la scène : arbres, arbustes, haies, voitures, maisons, immeubles, feux, routes, trottoirs, parcs ... Tous ces objets ont des caractéristiques radiométriques, des formes et des tailles différentes. On ne peut pas les traiter de façon identique, à la fois pour leur extraction et leur reconstruction. Cette

- diversité dépend aussi du contenu et des conditions d'observation de la scène.
- de la **complexité des objets** qui composent la scène : la végétation comme les bâtiments peuvent avoir des formes très variées. Pour les bâtiments, un autre problème s'impose : à quel degré de détail doit-on restituer les bâtiments? Doit-on retrouver les cheminées, les cages d'escaliers et les lucarnes? Ce problème dépend d'une part des conditions d'observation (on ne peut pas aller plus loin que ce que la résolution nous permet de voir) et d'autre part de l'application.
 - du **nombre d'images** qui est important pour assurer à la fois la couverture totale et la reconstruction d'une ville.
 - du **type d'images** : images N/B, images couleurs, images multi-spectrales, capteur push-broom, capteur laser ...

Les modèles de bâtiments : Toutes les méthodes de reconstruction de bâtiments commencent par le choix d'un modèle. Contrairement aux informations données par un MNE sur le bâtiment, le modèle du bâtiment permet de décrire la structure de celui-ci par des modèles mathématiques. Utiliser un modèle va permettre de structurer la forme du bâtiment, de la simplifier, de lui imposer des contraintes de régularité ou de répondre à des règles de cohérence. Cela permet aussi de diminuer considérablement la taille mémoire nécessaire pour la description du bâtiment. Le modèle représente les limites extérieures des bâtiments. On se retrouve face au problème du niveau de détail. Jusqu'à quel degré doit-on décrire le bâtiment ou le simplifier? *H. Mayer* présente dans [May98] un formalisme permettant d'intégrer le niveau de détail (LOD) à la description des bâtiments.

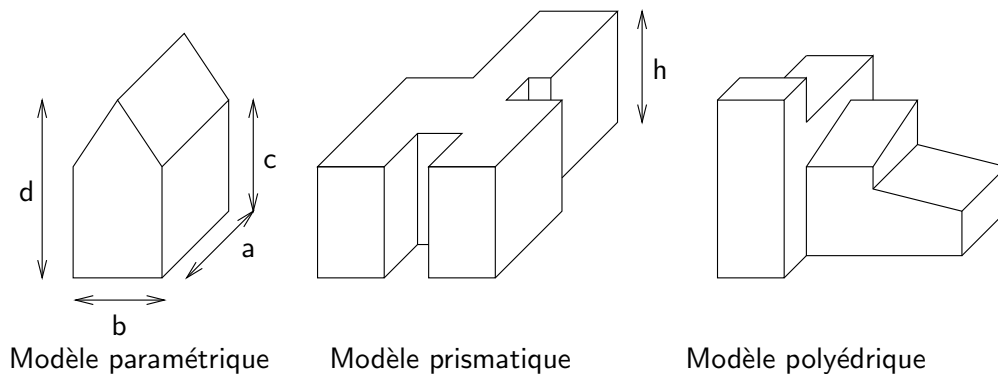


FIG. 3.3 – Plusieurs modèles de bâtiments.

Les bâtiments ont des formes aussi diverses que compliquées. On distingue deux classes de modèles : les **modèles paramétriques** et les **modèles génériques**. Différents modèles sont présentés dans la figure 3.3.

Les modèles paramétriques décrivent la forme des objets avec un jeu minimum de variables. Pour ces modèles, la topologie est fixée tandis que la géométrie, la localisation et l'orientation de l'objet sont variables. Ils décrivent des primitives simples : bâtiments rectangulaires à toit plat, bâtiments non orthogonaux à toit plat, bâtiments rectangulaires avec toit en pente ... L'avantage de ces modèles est de fournir une modélisation complète du bâtiment à partir de la primitive volumétrique paramétrique. Par contre, on ne peut pas modéliser les bâtiments aux formes trop complexes.

Les modèles génériques vont quant à eux permettre de modéliser toutes les formes de bâtiments. Ces modèles résultent de l'assemblage de plusieurs primitives. On distingue trois différentes classes pour ces modèles : les **modèles prismatiques**, les **modèles polyédriques** et les **modèles de CSG** (*Constructive Solid Geometry*). Les modèles prismatiques supposent que le bâtiment peut être décrit par un polygone dans le plan du sol, défini à une altitude fixe. Le modèle est généralement déduit d'un assemblage de segments. Le toit est horizontal et les façades verticales. Le nombre de paramètres du modèle est défini par rapport au nombre de sommets du polygone. Les modèles polyédriques permettent de décrire des formes plus complexes et surtout de pouvoir modéliser les toits en pente. Les modèles polyédriques décrivent les bâtiments par l'assemblage de morceaux de surface plans. La description du modèle est un peu plus complexe mais permet de décrire la quasi-totalité des bâtiments. On peut aussi introduire des contraintes sur le modèle : symétries ou verticalité des façades. Le dernier modèle de bâtiment est le modèle CSG. Le modèle du bâtiment est décomposé en primitives simples que l'on assemble par union, soustraction, intersection . . .

Stratégies de reconstruction : Le choix du modèle est généralement effectué en même temps que le choix de la stratégie de reconstruction des bâtiments. C'est même cette stratégie qui va induire le modèle à employer. Avec un modèle paramétrique, la stratégie est souvent similaire. On commence par choisir le modèle du bâtiment. Pour cela, on peut s'aider de primitives extraites des images telles que les segments. Pour l'estimation des paramètres du modèle, on définit une position et orientation initiale, on projette le modèle dans les images et on ré-estime les paramètres du modèle jusqu'à ce que l'enveloppe projetée du bâtiment corresponde aux contours des images.

La construction d'un modèle générique se fait par assemblage. Il existe dans ce cas une multitude de stratégies que l'on choisit selon le type de données initiales (MNE, contours, images . . .) et de l'application. La stratégie est généralement définie selon les connaissances disponibles sur les bâtiments et selon les caractéristiques que l'on souhaite extraire. Les différentes approches peuvent être basées sur : la fusion de primitives 2D ou de primitives 3D, la génération d'hypothèses puis vérification, approche par contours actifs, segmentation en région, segmentation en altitude . . . Il existe une très grande variété de méthodes pour construire un modèle de bâtiment. Nous proposons dans la section suivante (3.4) un état de l'art sur les principales approches de reconstruction.

Complexité du problème : Nous venons d'étudier les modèles de bâtiments et les différentes stratégies de reconstruction de ces modèles. Il est nécessaire de bien adapter le système de reconstruction à l'application. Lorsque la stratégie ou un modèle est simple, la solution obtenue est grossière. Le système ne peut évidemment pas construire un modèle détaillé lorsque le bâtiment est trop compliqué. Il peut toutefois apporter rapidement une solution représentative. Lorsque le modèle ou la stratégie devient plus complexe, on pourra modéliser plus fidèlement la structure du bâtiment. Dans le même temps, le problème devient beaucoup plus complexe et la solution obtenue pourra très vite devenir inefficace et non représentative. La modélisation des bâtiments est un problème complexe. Résoudre ce problème consiste dans un premier temps à analyser l'objectif final pour définir les caractéristiques des bâtiments à extraire et permettre de se limiter en complexité.

3.4 Un état de l'art

Cette section présente un état de l'art des publications traitant de la reconstruction de bâtiments. Les premières images utilisées étaient issues de capteurs optiques. D'autres types de capteurs permettent maintenant d'utiliser des informations de sources et de natures différentes. Ainsi, on peut obtenir des MNE à partir de données altimétriques obtenues par interférométrie radar ou par des techniques laser (travaux de l'université de Stuttgart). L'acquisition laser permet d'obtenir des résultats précis sur les zones étudiées mais reste toutefois peu exploitable dans un contexte industriel. Le coût de production et le temps d'acquisition sont encore très élevés. L'interférométrie radar est également difficilement exploitable pour la reconstruction des bâtiments en zone urbaine, de nombreux artéfacts diminuent considérablement la qualité de reconstruction et le fond des rues n'est pas visible.

Chaque type d'imagerie possède des atouts différents, mais aucune n'est indispensable. Il est possible de combiner ces informations dans le système de reconstruction [HKN90]. Cet état de l'art s'intéressera principalement aux techniques de reconstruction basées sur le traitement d'images optiques qui ont été utilisées pour les travaux de cette thèse. Les principales stratégies qui ont été proposées jusqu'à présent sont présentées dans la suite, classées par laboratoire.

Les objectifs fixés : Avant de commencer cet état de l'art, nous allons nous fixer plusieurs objectifs afin de pouvoir qualifier si chaque méthode présentée répond à ces objectifs et peut nous convenir, ou si elle ne permet pas de répondre à nos exigences. Le premier objectif que nous nous fixons est d'utiliser une stratégie qui permet de traiter *toutes les formes de bâtiment*. Pour cette raison, nous n'utiliserons pas une approche basée sur les modèles paramétriques. Le deuxième objectif est de traiter *tous les bâtiments* de la scène, on ne veut manquer aucun bâtiment ou partie de bâtiment. Nous avons écarté l'utilisation des approches par modèles qui regroupent des primitives extraites des images. Certaines parties du modèle d'un bâtiment peuvent être incomplètes lorsque ces primitives sont en nombre insuffisant. Notre troisième objectif est d'obtenir un *modèle représentatif* de la forme générale du bâtiment. Nous ne sommes pas intéressés par les détails du bâtiment, la stratégie doit être simple et efficace. Le dernier objectif fixé concerne la *qualité du détournement du bâtiment*, les applications de propagation des ondes utilisent principalement la diffraction sur les arêtes des bâtiments et la réflexion sur les faces du modèle. Le détournement des bâtiments doit être de bonne qualité, bonne localisation et orientation des faces du modèle (notamment les façades).

Nous avons développé dans cette thèse une stratégie basée sur la segmentation en plans d'un MNE dense (chapitre 5). L'utilisation d'un MNE dense permet de ne rater aucun bâtiment de la scène et de traiter toutes les formes de bâtiments. Nous utilisons ensuite les plans de la segmentation pour construire le modèle du bâtiment. Nous proposons enfin plusieurs algorithmes pour corriger le modèle (recalage du modèle sur les contours des images et orthogonalisation) et répondre ainsi aux deux derniers objectifs : représentativité et qualité du détournement (chapitre 6). Indépendamment de cette stratégie, nous avons aussi développé un système de reconstruction automatique des façades (chapitre 4). Nous pouvons ensuite utiliser ces points pour corriger un MNE, guider un snake ou reconstruire le modèle. Notre stratégie générale est présentée plus en détails dans la section 3.6. Nous allons maintenant présenter les principales approches de la reconstruction des bâtiments et essayer autant que possible de les qualifier par rapport à nos objectifs et de les comparer à notre stratégie.

3.4.1 Projets et universités américains

Les laboratoires américains ont participé et participent encore à plusieurs projets de recherche militaires liés à l'utilisation d'imagerie aérienne ou satellitaire pour la reconstruction de bâtiments. Leur recherches ont porté à la fois sur les problèmes de détection de bâtiments, d'extraction de bâtiments, de leur modélisation, de la mise à jour de modèles de ville, de l'utilisation de cartes, . . . Il y a eu quatre principaux projets américains : RADIUS, IUE, APGD et MURI. Beaucoup de laboratoires se sont investis dans ces multiples programmes de recherche.

Les différents projets de recherche américains

Le projet RADIUS Les États-Unis ont lancé par le biais de l'agence militaire **DARPA**^[1] (Defense Advanced Research Projects Agency) un vaste programme de recherche, **RADIUS**^[2] (Research And Development for Image Understanding Systems). Ce projet visait à tester les laboratoires de recherche américains en leur fournissant des outils d'exploitation automatiques ou semi-automatiques. Le projet est focalisé sur deux problèmes principaux : la construction automatique et semi-automatique de modèles de sites, ainsi que l'exploitation de ces modèles et leur mise à jour. Dans ce but, ils ont proposé en 1991 l'utilisation d'un environnement de développement commun, le **RCDE** (RADIUS Common Development Environment), dérivé du **CME** (Cartographic Modeling Environment) du département d'intelligence artificielle du SRI. L'historique ainsi que le système lui-même sont présentés dans [HQ97]. Le projet a été réalisé en deux phases : la première fût le développement du concept, des différentes opérations à effectuer et des exigences demandées. La seconde phase a été l'implémentation des résultats de la première phase dans un système complet, le **RADIUS Testbed System**. La composante correspondant à l'acquisition d'un modèle de site est décrite dans [HFCS96]. Ce projet est maintenant fini.

Le projet IUE l'Image Understanding Environment^[3] est une plate-forme logicielle orientée-objet dont le but est de développer des algorithmes d'analyse d'image et des applications. Le but était de faciliter l'échange des algorithmes et des logiciels de groupes de recherche. Cette plate-forme a servi de support de développement d'intégration et d'échange à plusieurs projets de recherche du DARPA : RADIUS, MSTAR et IUBA. Actuellement, cette plate-forme est en cours d'intégration dans l'environnement de développement plus général de General Electric, **TargetJr**^[4].

Automatic Population of Geospatial Databases : APGD C'est un projet du programme IU du DARPA auquel ont participé plusieurs laboratoires de recherche : Carnegie Mellon University, Colorado State University, MIT Artificial Intelligence Lab, SRI International, University of Massachusetts et University of Southern California. Le but de ce projet était de développer, d'évaluer et de valider la technologie permettant la production automatique de bases de données géospatiales à partir de sources d'images variées pour des applications d'environnements synthétiques et d'exploitation d'images. Chaque laboratoire travaillait sur un sujet défini. Le projet APGD^[5] a été abandonné en 1999.

Le projet MURI Multidisciplinary University Research Initiative regroupe des projets de recherche financés par plusieurs départements de recherche militaires : U.S. Army Research Office (ARO), Office of Naval Research (ONR) . . . Le but de ces projets multidisciplinaires est

de faire avancer la recherche dans des domaines d'intérêt militaire. De nombreuses études sont supportées par ce projet : modélisation de terrain et de ville [USC, Purdue et GDE Systems], visualisation directe 3D [Berkeley, MIT et UCSF], ... Le projet de modélisation des villes s'intéresse au développement de méthodes pour un système d'extraction urbain rapide et abordable. Le but est d'augmenter les niveaux d'automatisation de la détection et de la description d'objets à partir de toutes sortes d'images.

Les universités ayant participé aux projets

Perception Program, SRI Une grande partie des travaux du SRI traite de la réalisation du RADIUS Testbed System. L'article de A. Heller [HFCS96] présente le système choisi. La composante centrale du projet RADIUS était la réalisation d'un système fiable de création de modèle 3D du site examiné. Pour cette raison, le système choisi fut une solution interactive et semi-automatique. Aucune des méthodes automatiques réalisées dans ce programme de recherche n'était assez robuste pour reconstruire complètement une ville.

Le choix adopté pour la stratégie générale est basé sur l'optimisation de modèles : MBO Model-Based Optimisation, présenté par P. Fua dans [Fua95, Fua96b]. Cette technique est basée sur les snakes, ou contours actifs, introduits dans le traitement d'image par M. Kass, S. Witkins et D. Terzopoulos [KWT87]. Le principe des contours actifs est de déplacer progressivement une courbe vers un contour d'image. La procédure originale proposée dans [KWT87] utilise des équations dynamiques d'évolution de la courbe pour arriver à la solution. La méthode du MBO optimise ces équations par une descente de gradient ou un gradient conjugué. L'avantage est de pouvoir réaliser une optimisation sous contrainte : contrainte sur les plans, sur les angles ... Nous avons développé aussi dans cette thèse un algorithme de snake polygonaux multi-images pour recalculer le modèle sur les images. Nous aborderons ce sujet plus précisément dans le chapitre 6.

L'intégralité de la reconstruction est réalisée de manière itérative et interactive. L'opérateur initialise un modèle (dans une vue orthogonale), puis une procédure d'affinement basée sur le MBO est lancée. La procédure générale de reconstruction est la suivante :

- **Segmentation de la zone à analyser en régions** (forêts, rivières, ...). L'opérateur commence par tracer manuellement ces régions. Chacune de ces régions est définie comme une courbe 3D fermée. L'affinement des modèles est réalisé par la procédure interactive d'optimisation de modèle (l'opérateur effectue des rectifications a posteriori).
- **Extraction des lignes de communication.** On identifie ensuite les réseaux routiers, chemins de fer, ... Les routes sont modélisées par des primitives de rubans. Ces primitives correspondent à des courbes 3D où en chaque point on ajoute un paramètre supplémentaire correspondant à l'épaisseur de la courbe. L'opérateur définit grossièrement le réseau puis lance le système d'optimisation (*Ziplock snakes* [NFI⁺94]).
- **Extraction des bâtiments.** L'opérateur peut choisir un grand nombre de primitives pour représenter un bâtiment. La primitive est placée sur le bâtiment à extraire et de la même manière que précédemment, une procédure d'affinage automatique va faire correspondre le modèle aux images.

IRIS Group, University of Southern California (USC) Le laboratoire de recherche en Computer Vision de USC basé à Los Angeles est actif dans plusieurs domaines de l'imagerie aérienne. Les sujets spécifiques sur lesquels il travaille incluent la segmentation de scène et d'image, la stéréo, le groupement perceptuel, l'analyse de formes et la reconnaissance d'objets.

S. Noronha et *R. Nevatia* présentent dans [NN97] une méthode pour la détection et la description de bâtiments rectangulaires à partir de deux ou plusieurs images. Ils extraient des primitives (segments et coins) qu'ils mettent en correspondance puis utilisent une technique de groupement perceptuel hiérarchique pour organiser des structures cohérentes. Cette procédure de groupement permet de générer des hypothèses de toits de bâtiments qui sont ensuite examinées par vérification des murs et des ombres. La sortie du système est un modèle 3D de chaque bâtiment avec une mesure de confiance. En ne disposant que d'une seule image de la scène, la stratégie est alors sensiblement différente [LHN94]. Les auteurs extraient les primitives linéaires et les regroupent pour générer des hypothèses de parallélogrammes. Ils sélectionnent ensuite les bonnes hypothèses en utilisant des évidences extraites des images. Ils valident ensuite ces hypothèses en vérifiant simultanément s'il existe une ombre et un mur pouvant correspondre à cette situation (connaissant la direction d'illumination).

J. Li, *R. Nevatia* et *S. Noronha* proposent aussi dans [LNN99] un système de reconstruction interactif basé sur leurs précédents travaux. Le système interactif est composé de trois modules. Le premier module est le MABS (Monocular Automated Building System). Il permet de construire automatiquement et complètement un modèle 3D de bâtiment à partir d'une vue. Le deuxième module est le module d'interaction initiale qui permet à l'opérateur de sélectionner des hypothèses supplémentaires pour le MABS. Le dernier module est un module d'interaction corrective qui va permettre de corriger les modèles 3D ou les hypothèses en vue d'obtenir une modélisation finale correcte. Leur système est implémenté de manière à fonctionner dans l'environnement RCDE. Les derniers travaux du laboratoire sont présentés dans [NHK99, HKN90]. Ces travaux portent sur l'utilisation d'autres sources de données et l'extraction des réseaux linéaires. Les travaux ont porté principalement sur des bâtiments aux formes simples, leur stratégie ne traite pas les bâtiments aux formes plus complexes. Certains parties des bâtiments sont parfois manquantes.

Computer Vision, University of Massachusetts (UMASS) Le laboratoire de recherche en Computer Vision^[10] de l'université du Massachusetts (UMASS) a développé un système de reconstruction automatique de bâtiments lors du projet RADIUS. Ce système, ASCENDER (Automated Site Construction, Extension, Detection and Refinement) est construit avec RCDE. La procédure d'extraction des bâtiments est étudiée dans [CHR⁺96] et décrite en détail dans [CJS⁺95]. La détection des bâtiments commence avec l'extraction des segments présents dans les images. Les points de contours extraits des images sont groupés récursivement selon un critère d'organisation perceptuel. Les segments servent ensuite à construire des hypothèses de contours de bâtiments en utilisant une technique basée sur les graphes (recherche de cycles non contradictoires) et le groupement perceptuel. Ils calculent ensuite les informations tridimensionnelles de chaque polygone 2D via une mise en correspondance multi-images des segments et détectent les toits. Une procédure de triangulation permet ensuite de déterminer tous les paramètres des objets de la scène : longueurs, orientations, positions dans l'espace objet avant de reconstruire la scène tridimensionnelle. Les textures des toits sont prises dans les images. *X. Wang* et *A. Hanson* présentent dans [WH97] une méthode pour extraire les fenêtres des façades et affiner le modèle du bâtiment. *C. Jaynes et al.* présentent dans [JHR97] une méthode de reconstruction plus fine des bâtiments. Ils utilisent les images de la scène et un MNE initial du bâtiment. Pour chaque région classée avec le label `bâtiment`, ils calculent les vecteurs normaux à la surface en chaque point et génèrent un histogramme tridimensionnel des directions des vecteurs normaux. Les

auteurs définissent une base de donnée de 8 primitives de surface permettant de composer 51 modèles simples de bâtiments. L'histogramme leur permet de choisir le modèle approprié et le jeu de paramètres nécessaires pour estimer la surface. Ils calculent un score de corrélation entre les histogrammes des vecteurs et des différents modèles pour tous les alignements possibles de ces histogrammes tridimensionnels, sélectionnent le modèle ayant le meilleur score puis estiment ses paramètres par la méthode de la moindre médiane des carrés (LMS). *R. Collins* présente dans [Col96] une technique nommée **space-sweep** permettant la mise en correspondance de modèles dans un système multi-images. Il définit sa technique comme méthode de mise en correspondance multi-images vraie puisqu'elle est indépendante vis-à-vis du nombre d'images, d'une complexité linéaire par rapport au nombre d'images et ne privilégie aucune des images. Le principe est le suivant : un plan balaye l'espace 3D le long de l'axe des Z, les primitives extraites dans toutes les images (des points de contours par exemple) sont reprojctées dans ce plan, les cellules du plan ayant un nombre suffisant de votes correspondent aux primitives 3D cherchées. Ces approches sont basées sur les modèles, les 51 modèles bâtiments ne peuvent représenter correctement toutes les formes des bâtiments.

Digital Mapping Laboratory (MAPSLab), Carnegie Mellon University Les premiers travaux du laboratoire^[11] sur la génération d'hypothèses de bâtiments à partir de vues monoculaires ont abouti au système BABE [RD89]. Celui-ci extrait des lignes de contours, les regroupe pour former des coins, construit ensuite des chaînes qui relient ces coins, génère des hypothèses de parallélogrammes puis vérifie ces hypothèses de bâtiment à l'aide de contraintes d'intensité, de taille et d'ombre. *J. McGlone* et *J. Shufelt* proposent dans [JJ94] une méthode pour retrouver les segments horizontaux et verticaux dans une image. Ils utilisent la propriété de géométrie projective selon laquelle toutes les droites parallèles dans une image s'intersectent en un même point. Comme les bâtiments sont formés de segments parallèles et perpendiculaires dans le plan horizontal, ils proposent de faire voter chacun de ces segments (parallèles ou perpendiculaires à une direction choisie) pour un azimut. L'azimut ayant le nombre de votes maximum correspond à celui du plan horizontal. Les auteurs proposent d'utiliser cette information directionnelle dans le processus de reconstruction des bâtiments. *M. Roux* et *D. McKeown* proposent dans [MD94] le système MULTIVIEW. Ce système est une extension tridimensionnelle du système précédent. Il intègre aussi les informations des nouvelles images au fur et à mesure, en mettant à jour les coins 3D, les segments 3D et leur graphe relationnel. *Y. Hsieh* présente dans [Hsi96] le système d'extraction de bâtiments semi-automatique SiteCity. Disposant de trois modèles de bâtiments, un opérateur place le modèle choisi sur une des images qui sert à vérifier l'hypothèse à partir de segments extraits de cette image. Il calcule ensuite le modèle final à partir de toutes les autres images. Tout comme les deux laboratoires précédent, USC et UMASS, les travaux du MAPSLab sont basés sur des modèles, leur approche regroupe des primitives extraites des images.

RADIUS: Les autres travaux D'autres universités ont aussi participé aux projets du DARPA. *X. Liu*, *R. Haralick* et *K. Thornton* de l'université de Washington^[12] proposent dans [LHT96] une méthode d'optimisation permettant d'estimer les paramètres de plusieurs modèles de bâtiments à partir de points 3D et de contraintes. Le laboratoire propose aussi dans [HNT⁺96] un CD-ROM avec des scènes et des modèles pouvant servir de référence absolue. Le Computer Vision Laboratory^[13] de l'université du Maryland (CVL) a travaillé sur

la mise à jour de sites à partir de nouvelles images. En se basant sur l'analyse d'images et leur interprétation, ils ont développé une série d'outils visant à détecter les changements des bâtiments ou autres objets de la scène [CZD⁺93, CBL⁺99].

3.4.2 Une revue des travaux sur les bâtiments, triée par laboratoire

Graphics Group du Massachusetts Institute of Technology (MIT) [CT98] Le Computer Graphics Group^[14] du MIT dirigé par *S. Teller* s'intéresse à la réalité virtuelle. L'objectif du MIT City Scanning Project est de modéliser complètement (extérieur et intérieur) et automatiquement une scène de bâtiments. L'acquisition des structures urbaines est réalisée automatiquement par l'intermédiaire d'un robot photographe mobile. Celui-ci est muni d'un système GPS embarqué et de capteurs inertiels. Le robot se déplace entre les bâtiments et tout au long de son parcours il acquiert des photos terrestres. En chaque point d'acquisition, le robot construit des mosaïques sphériques d'images géo-référencées qui vont servir à reconstruire la scène 3D. La thèse de *S. Coorg* présente une description précise des principales étapes de ce système [Coo98]. La reconstruction et l'extraction des façades est décrite dans [CT98]. Une première étape extrait l'azimut à partir des lignes horizontales extraites des images. Les auteurs utilisent ensuite le *space-sweep algorithm* pour apparier les segments horizontaux de la scène et reconstruire les bâtiments. Disposant d'un nombre important d'images des façades, ils proposent d'extraire la texture médiane des images pour obtenir le modèle final. Il est à noter que ces travaux sont très actifs et que de nombreuses autres publications sont à venir. L'algorithme de reconstruction des façades que nous proposons dans le chapitre 4 est aussi basé sur une technique du type *space-sweep algorithm* proposée initialement par *R. Collins* dans [Col96].

Computer Vision Group, Université de Californie à Berkeley Toujours à partir de photos prises au sol, *P. Debevec* de Berkeley^[15] présente dans [DTM96, Deb96] le système de reconstruction de bâtiments semi-automatique FACADE^[16]. L'idée de ce système est de combiner des approches à base de géométrie et à base d'images pour reconstruire des bâtiments pour une application architecturale. Dans une première phase, le système FACADE permet à un opérateur de construire un modèle du bâtiment à partir des images. Cette approche interactive permet de définir un modèle grossier du bâtiment représentant les principales facettes du modèle. Souhaitant un modèle plus détaillé des façades (colonnes, statues ...), *P. Debevec* propose une méthode de raffinement du modèle à base de mise en correspondance. Le principe est de rectifier les images par rapport aux faces du modèle grossier de bâtiment, puis de mettre en correspondance ces nouvelles images pour retrouver toutes les petites structures présentes sur les façades. Pour que le modèle soit photo-réaliste et puisse servir à générer des films virtuels, l'auteur propose une méthode nommée *view-dependent texture mapping (VDTM)* permettant de générer des photographies correspondant à des caméras virtuelles [DBY98]. L'idée est de pondérer l'influence des caméras ayant les points de vue les plus proches. Ces travaux ont été utilisés pour certains effets spéciaux du film *The Matrix*^[17]. Nous utilisons dans notre algorithme de reconstruction des points des façades une technique semblable à celle utilisée par FACADE, rectification des images par rapport au modèle initial. *P. Debevec* utilise cette technique pour raffiner le modèle initial tandis que l'algorithme que nous proposons rectifie les images pour pouvoir mettre en correspondance des surfaces de très forte pente qui ne peuvent être appariées dans les images initiales (ceci parce que nous utilisons aussi des images aériennes).

Département TSI, École Nationale Supérieure des Télécommunications (ENST) Dans [ML92], *H. Maître* et *W. Luo* présentent l'approche générale de l'ENST_{TS} pour la génération de MNE denses. Cette approche fait intervenir et coopérer à la fois des informations photométriques et stéréoscopiques. La philosophie générale consiste à utiliser le résultat d'une segmentation radiométrique réalisée sur une des images pour guider la densification d'une carte 3D brute. La procédure de densification utilise toutes les informations 3D contenues dans une région pour estimer un modèle 3D représentatif. Dans cet article, les auteurs s'intéressent à la dernière étape de ce processus, la densification. Ils définissent une stratégie utilisant deux modèles de surface : le modèle surface plan et le modèle surface quadratique. Pour chacune des régions de l'image, on teste le modèle plan sur le champ de disparité restreint à la région homogène. Si le modèle est accepté, chaque pixel de la région prend la valeur du modèle (même les points non définis), dans le cas contraire on lance une seconde procédure d'estimation robuste basée sur l'algorithme du RANSAC ou on relance le processus de segmentation dans cette région avec un critère plus strict pour tenter de retrouver les différents modèles qui composent la région. En cas de succès de la re-segmentation, on recommence le même processus avec le test de planarité, ou alors on teste le modèle quadratique. Dans [RMG97], *M. Roux*, *H. Maître* et *S. Girard* proposent et étudient plusieurs étapes de l'approche générale de l'ENST. Une étape de filtrage anisotrope des images est introduite dans la procédure générale pour éliminer l'aspect granulaire introduit par la digitalisation des images. Ils présentent ensuite un processus de segmentation pour les images de couleur. Ils utilisent l'algorithme itératif de *Suk* adapté à un critère basé sur une mesure colorimétrique. Les auteurs étudient différentes stratégies basées sur des mesures dans les deux espaces : RVB et Lab. Enfin, ils finissent leur article par l'étude de trois méthodes de densification de carte. La première est basée sur la recherche du plan d'inertie, la deuxième est une technique de moindres carrés récursifs et la dernière est la méthode du LMedS récursif. Ils proposent un algorithme combinant les trois méthodes. Dans [FRM99], les auteurs proposent de projeter sur le sol les points appariés pour détecter les façades. Ces façades sont ensuite utilisées pour détecter les bâtiments et délimiter les régions. Cette approche permet d'obtenir une reconstruction dense de la scène. Toutefois, la qualité de la modélisation dépend de celle de la segmentation.

Laboratoire MATIS, Institut Géographique National (IGN) A l'IGN_{IG}, *O. Dissard* et *O. Jamet* présentent dans [DJ95] une méthode de reconstruction de bâtiments rectangulaires. Leur technique combine la détection de bâtiments à partir d'analyses monoculaires et les résultats de stéréoscopie par corrélation. Dans sa thèse [Bai97], *C. Baillard* présente deux algorithmes dont le but est de faciliter la reconstruction et l'interprétation des images aériennes. La première partie présente un algorithme de stéréoscopie pour la construction de cartes 3D denses. La méthode utilise deux phases successives de mise en correspondance par programmation dynamique (section 2.3.1). La première de ces étapes est réalisée sur les points de contour pour pouvoir délimiter des zones de points à appairer. Après un post-traitement appliqué au résultat pour éliminer les incohérences, *C. Baillard* réalise un appariement photométrique de ces zones en deux passes. La première passe va privilégier la contrainte radiométrique en utilisant un seuil de corrélation strict et un coût d'occultation faible. Dans la seconde passe, elle utilise le même processus mais privilégie cette fois la contrainte géométrique (coût d'occultation plus fort et seuil de corrélation plus lâche) pour obtenir la carte finale. Il est à noter que plusieurs post-traitements sont effectués à la fin de chacune de ces étapes. La seconde partie de sa thèse présente une approche

pour extraire les objets du sur-sol et les classer en deux classes : les bâtiments et la végétation. Elle procède pour cela en deux étapes distinctes. La première est la détection des objets du sur-sol. *C. Baillard* segmente tout d'abord le MNE obtenu en différents niveaux de hauteurs. Elle le découpe ensuite en deux classes (sol et sur-sol) et construit un MNT par triangulation de Delaunay dans un processus itératif. La seconde étape est la classification des objets du sur-sol en Bâtiment ou Végétation. La procédure suppose que les zones correspondant à la végétation possèdent une structure radiométrique non structurée. La stratégie développée dans cette thèse peut être utilisée pour extraire des zones d'intérêt sur les bâtiments que nous souhaitons reconstruire. Dans [JMM99], *O. Jamet*, *H. Maître* et *H. Le Men* présentent un modèle d'incertitude permettant la détection de changements planimétriques des bâtiments.

Groupe ETIS de l'Université de Cergy-Pontoise *M. Jordan* et *J. Cocquerez* du groupe ETIS^[20] présentent dans [JC95] un système complet d'analyse de scène aérienne pour obtenir une description tridimensionnelle. Le système est composé de trois parties : la modélisation géométrique et le calibrage des paramètres du système, la mise en correspondance des images et la description de la scène. La mise en correspondance est réalisée en deux étapes : appariement des points de contours puis appariement des régions délimitées par ces points de contours. Pour obtenir une description de la scène, les auteurs définissent une stratégie permettant la segmentation d'une image prise comme référence en régions planes (horizontale, oblique ou verticale) ou inconnues. *M. Cord*, *N. Paparoditis* et *M. Jordan* présentent dans [CJCP99] un algorithme de mise en correspondance par corrélation avec des voisinages non rectangulaires. L'idée est d'utiliser une carte des contours extraite de l'image pour délimiter le voisinage physique du point 3D à reconstruire. Contrairement aux fenêtres adaptatives, la fenêtre est de taille fixe, c'est le support utilisé qui est adaptatif. Les auteurs masquent les régions ne correspondant pas à la même surface et associent une pondération géodésique pour limiter la diffusion au travers des contours non fermés. L'algorithme est inséré dans une procédure multi-résolution pour restreindre l'espace de recherche et diminuer les temps de calcul. Les auteurs présentent ensuite un schéma de classification de la scène. Après l'extraction du sur-sol, ils classifient les différentes régions de ce sur-sol en bâtiment ou végétation selon une étude locale des normales à la surface. La reconstruction du modèle final de bâtiment est basée sur la combinaison de segments extraits des images et la limite des régions Bâtiment extraites. La thèse de *M. Cord* présente ces algorithmes en détail [Cor98]. Leur approche permet d'obtenir un MNE de bonne qualité, les limites des bâtiments sont beaucoup mieux respectées.

ISTAR et le Projet PASTIS de l'INRIA *M. Berthod*, *L. Gabet*, *G. Giraudon* et *J. Lotti* (ISTAR^[21] et PASTIS^[22]) présentent dans [BGGL95] deux méthodes d'appariement d'images aériennes permettant d'obtenir des cartes denses. La première est une technique multi-résolution utilisant des fenêtres de tailles adaptatives. Ils appliquent le CBDL (Contour Based Disparity Limits) à chaque image pour définir quatre fenêtres excentrées pour chaque pixel (extension d'une fenêtre selon quatre directions). Ces fenêtres sont utilisées pour calculer les scores de corrélation. Le CBPR (Contour Based Pyramidal Resolution) consiste à utiliser ces fenêtres à chaque résolution pour apparier les images. Si aucune des fenêtres n'est valide, ils utilisent une simple fenêtre carrée. Une description précise de cette méthode peut être trouvée dans [Lot96]. La seconde méthode a été définie dans un cadre applicatif par la société ISTAR. Le principe est d'utiliser la redondance d'information pour obtenir des

cartes denses et fiables. Utilisant des couples d'images avec d'importants recouvrements, on génère plusieurs MNE d'une même zone cartographique. Les cartes sont ensuite fusionnées par une technique de vote majoritaire. *L. Gabet, G. Giraudon et L. Renouard* présentent dans [GGR97] une étude détaillée de l'algorithme. Nous utilisons dans cette thèse des MNE construits selon la technique présentée.

Groupes BIWI et Chair of Photogrammetry and Remote Sensing, The Swiss Federal Institute of Technology (ETH Zürich) Les groupes d'image BIWI^[23] et de photogrammétrie^[24] de l'ETH Zürich se sont beaucoup intéressés à la construction de modèles de bâtiments au sein du projet AMOBE. Leurs recherches se composent de deux approches aux conceptions différentes pour la reconstruction de bâtiments : TOBAGO (TOPOlogy Builder for the Automated Generation of 3D Objects from point clouds) et ARUBA (Automated Reconstruction of subUrban Buildings from Aerial images). Les deux approches sont présentées dans [Hen96]. La première est semi-automatique dans le sens où elle permet de générer des modèles CAO complets à partir de nuages de points 3D sélectionnés manuellement par un opérateur externe. Dans la seconde approche, la procédure est composée de deux phases complètement automatiques, une phase de détection et une phase de reconstruction de bâtiments. La détection des bâtiments est basée sur la détection de Blobs d'élévation dans le MNE combinée avec une analyse colorimétrique. La seconde étape est la reconstruction du bâtiment à partir des petites images extraites. La stratégie adoptée et développée au sein du projet est décrite dans [BHFS96], elle utilise quatre prises de vues chromatiques. Cette stratégie combine des contours 2D, des attributs photométriques et chromatiques et des informations 3D pour générer des toits complexes. Après détection des contours présents dans la première image, les auteurs procèdent à une phase d'agrégation et leur associent des attributs 2D photométriques et chromatiques. Ils construisent ensuite des primitives géométriques selon les relations détectées (perpendicularité, parallélisme, intersection, ...). En parallèle de ce processus, un algorithme tridimensionnel met en correspondance les segments extraits des trois autres images initiales, reconstruit une scène de segments 3D puis regroupe les segments 3D coplanaires. Le toit du bâtiment est reconstruit en combinant les informations 3D et 2D réunies selon un critère de cohérence. L'approche automatique ARUBA est beaucoup trop complexe pour pouvoir réussir à reconstruire tous les bâtiments d'un environnement urbain dense. La stratégie de reconstruction est principalement basée sur les segments extraits des images. Sur la base de ces travaux, *A. Gruen et X. Wang* présentent dans [GW98] un système commercial complet de reconstruction semi-automatique de bâtiments : CC-Modeler (CyberCity Modeler). Comme dans TOBAGO, un opérateur définit manuellement un ensemble de points et le système extrait la surface automatiquement. L'opérateur peut ensuite corriger les erreurs et visualiser ou sauvegarder la scène.

PSI-VISICS, Université Catholique de Louvain (K. U. Leuven) *M. Hendrickx, J. Vandekerckhove, D. Frere, T. Moons et L. Van Gool* du groupe VISICS^[25] présentent dans [HVF⁺97] une méthode de reconstruction des toits de bâtiments basée sur les modèles. La modélisation est réalisée en quatre étapes. La première étape est l'extraction de segments suivie d'une sélection basée sur l'homogénéité radiométrique. Cette sélection permet l'extraction des bâtiments pour les traiter indépendamment. La deuxième étape est la mise en correspondance des segments extraits et leur reconstruction 3D. La troisième étape est le groupement des segments 3D pour former des morceaux de surface plan et générer des hypothèses de polygones. La dernière étape est la génération du modèle de bâtiment par fu-

sion des polygones en utilisant des labels. *T. Moons, D. Frère, J. Vandekerckhove* et *L. Van Gool* ajoutent à cette méthode une phase d'ajustement du modèle final aux images dans [MFVG98]. Ils proposent de maximiser une fonctionnelle multi-images permettant de recaler les segments du modèle sur les contours images. La construction du modèle est obtenue avec une approche par modèle. La méthode d'ajustement du modèle proposée est semblable à celle du MBO présentée par *P. Fua* dans [Fua95]. Nous avons aussi développé une technique de recalage par des snakes polygonaux similaire à ces deux méthodes. Contrairement à ces approches, nous contraignons le snake à se déplacer dans un plan horizontal (section 6.2.1).

Institute of Photogrammetry de l'université de Bonn (IPB) Plusieurs approches ont été développées à l'Université de Bonn^[26]. Deux approches sont automatiques, celle de *U. Weidner* basée sur la détection et la reconstruction de bâtiments à partir d'un MNE et celle de *A. Fischer* basée sur la reconstruction du bâtiment à partir de coins. Une autre approche est semi-automatique, le système **ObEx**. *U. Weidner* présente dans [Wei97] une approche de reconstruction basé sur l'utilisation d'un MNE. La première phase est l'extraction du sol et la détection des bâtiments. La seconde phase est la reconstruction des bâtiments qui est celle présentée dans [WF95], utilisant des modèles paramétriques et prismatiques. Les contours des bâtiments sont extraits puis vectorisés. L'auteur utilise ensuite une procédure de simplification des polygones basée sur le critère du MDL. Si le polygone est simple, la reconstruction est faite en utilisant le modèle paramétrique approprié. Sinon, une procédure d'extraction des toits est lancée sur le modèle prismatique. *F. Fisher, T. Kolbe, F. Lang, A. Cremers, W. Förstner, L. Plümer* et *V. Steinhage* présentent dans [FKL⁺98] un processus de reconstruction de bâtiments à partir de plusieurs images. Leur approche se base sur l'utilisation des coins pour reconstruire et vérifier le modèle. Les coins 2D servent à construire des coins 3D qui permettent de générer les hypothèses de bâtiments. La phase de vérification se fait ensuite en reprojétant le modèle dans les images. La procédure est construite sous la forme d'une boucle fermée pour permettre de retrouver des coins qui n'étaient pas détectés. **ObEx** est un système semi-automatique de reconstruction de bâtiments présenté par *E. Gülch, H. Müller* et *T. Labe* dans [GML99]. Le système utilise des modèles de type CSG. L'opérateur construit le modèle dans une image prise comme référence, il dispose ensuite de trois modes différents pour le recaler dans une autre image : une méthode manuelle, une méthode guidée ou une méthode automatique basée sur l'utilisation de méthodes robustes (RANSAC) présentées dans [LG98]. Le système sélectionne aléatoirement des segments valides pour générer des hypothèses de modèles. La résolution et la qualité du MNE ne peuvent permettre d'obtenir un modèle précis. La stratégie de *U. Weidner* reste donc très simple.

Institut für Photogrammetrie de l'Université de Stuttgart (IFP) A l'IFP^[27], *N. Haala* et *M. Hahn* présentent dans [HH95] une première approche pour reconstruire des bâtiments à partir d'un MNH (Modèle Numérique des Hauteurs) et de plusieurs images. Une détection des bâtiments par extraction des maxima locaux du MNH permet de focaliser l'attention sur les bâtiments. Ils se servent ensuite de segments extraits des images pour construire des lignes 3D. Ces lignes sont groupées pour former des hypothèses de bâtiments. Un modèle de bâtiment est ensuite ajusté à chacune des hypothèses et le modèle final choisi correspond à celui qui possède la somme des résidus minimale. Dans [HB97], *N. Haala* et *C. Brenner* proposent une nouvelle méthode utilisant un plan du sol et un MNE laser. Ils segmentent le MNE en plans, le plan du sol servant à délimiter les contours des

bâtiments. Une première reconstruction du bâtiments est réalisée avec les plans extraits du bâtiment en s'appuyant sur le détournage défini par le plan du sol. Tous les plans sont ensuite validés ou supprimés itérativement jusqu'à obtenir le modèle final. Un système complet de reconstruction semi-automatique est présenté dans [BH98]. Les auteurs ont ajouté plusieurs éléments : la construction du plan du sol à partir de cartes ou d'un GIS, l'extraction automatique de la végétation à l'aide de données multispectrales, une interface permettant l'interactivité avec les modèles reconstruits et l'ajout de textures sur les façades à partir de photos terrestres.

B. Ameri et *D. Fritsch* présentent dans [AF00] un système complet de reconstruction automatique intéressant. Le système est divisé en trois parties : une première partie traitant de la détection et de la segmentation des bâtiments, une deuxième partie s'occupant de la reconstruction du modèle du bâtiment et une dernière partie permettant le raffinement du modèle. La détection d'un bâtiment est basée sur la morphologie mathématique et la segmentation en plan utilise un algorithme de croissance par région et d'estimation par moindres carrés des surfaces. Les auteurs utilisent ensuite des techniques robustes pour générer des polygones plans 3D. Ils construisent un graphe d'adjacence (PAR : Polygon Adjacency Relationships) et fusionnent les polygones compatibles. *B. Ameri* et *D. Fritsch* proposent ensuite la reconstruction d'un modèle polyédrique grossier grâce à leur outil POLY-MODELER (voir [AF99] pour une description détaillée). POLY-MODELER permet de construire un modèle polyédrique 3D générique à partir des polygones 3D et de leurs droites d'intersection dans l'espace. Une procédure de raffinement, le FBMV (Feature Based Model Verification) permet ensuite de réajuster ce modèle à des points de contour extraits des images.

Les deux premières approches ne nous conviennent pas car nous ne souhaitons pas utiliser de primitives extraites des images pour construire le modèle et nous ne disposons pas de cartes digitalisées. L'algorithme de segmentation en plans utilisé par *B. Ameri* et *D. Fritsch* est basé sur l'algorithme du RANSAC, tout comme l'algorithme que nous proposons dans le chapitre 5. Les auteurs proposent ensuite un système de reconstruction de modèle polygonal qui semble être assez robuste, les auteurs présentent de très bons résultats.

FGAN FIM Dans [SML96], *U. Stilla*, *E. Michaelsen* et *K. Lütjen* du département FIM^[28] proposent une méthode d'extraction de bâtiments basée sur la construction d'un réseau de fabrication (comparable à un réseau de Pétri). A partir des images, ils extraient les contours puis génèrent des segments. Ils combinent ensuite ces primitives pour obtenir dans un premier temps des coins, des rectangles puis des toits. Les auteurs construisent leur réseau de fabrication de modèle à partir de toutes ces primitives. Les relations qui les lient sont des liens géométriques et plusieurs modèles génériques sont générés automatiquement. Pour vérifier la validité de chacun de ces modèles construits, les auteurs proposent une procédure de test. Utilisant un modèle paramétrique, les auteurs retrouvent le modèle final du bâtiment en comparant ce modèle à différents objets de la scène étudiée. Cette comparaison est faite à plusieurs niveaux : comparaison directe (3D) du modèle aux modèles construits par le réseau de fabrication, comparaison des primitives extraites de ce modèle à celles extraites des images et comparaison des images initiales à une image virtuelle générée avec le modèle. *U. Stilla*, *R. Geibel* et *K. Jurkiewicz* proposent dans [SGJ97] de construire leur réseau de fabrication à partir d'une carte. Dans ce cas, les modèles sont prismatiques et tous de même hauteur. Le système présenté utilise des modèles et semble difficile à réaliser.

Chair for Photogrammetry and Remote Sensing de l'Université de Munich *H. Mayer* de l'université de Munich^[29] présente dans [May98] un concept de LOD (Level Of Detail) pour les modèles de bâtiment, basé sur la théorie des espaces d'échelle. Il propose une approche qui permet de simplifier les modèles de bâtiments 2D et 3D et donc de définir un modèle avec différents niveaux de complexité. Cette méthode est basée sur les opérations d'ouverture et de fermeture de la morphologie mathématique et les événements liés aux espaces d'échelles, les chocs. Il définit deux types de chocs topologiques, les événements internes et externes. Il définit ensuite l'espace des courbures discrètes et la rectification des structures non orthogonales pour simplifier les modèles. Les résultats sont présentés pour des modèles 2D et 3D. *H. Mayer* présente dans [May99] une revue des principales approches pour l'extraction automatique d'objets à partir d'images aériennes en se focalisant sur l'extraction des bâtiments. Il commence par s'intéresser aux complexités des données et des approches pour définir des critères d'évaluation pour les images, les modèles et les stratégies. Il étudie ensuite en détail un panel de huit approches avec les critères qu'il a définis. Il présente enfin une revue des principales stratégies et modèles définis dans la littérature existante.

Groupe TNT de l'Université de Hannover Les travaux du laboratoire^[30] utilisent le système AIDA^[31] Automatic Image Data Analyser. Dans [WG96], *S. Weik* et *O. Grau* présentent une approche pour reconstruire des bâtiments à partir de plusieurs prises de vues terrestres. Le système décrit utilise une connaissance a priori de la scène pour compenser le bruit. Cette connaissance est construite sous la forme d'un réseau sémantique reliant les différentes parties du modèle de bâtiment défini, les caméras, les images, les régions ou primitives extraites de ces caméras . . . Un opérateur peut ajouter des contraintes définies entre les différentes parties du modèle et les primitives extraites des images. Le système utilise une optimisation numérique qui permet de déterminer le modèle qui correspond le mieux aux contraintes.

Chair of Photogrammetry and Remote Sensing de l'Université Technologique de Delft A l'université de Delft^[32], deux approches traitent de la reconstruction de bâtiments : celle de *F. van den Heuvel* qui utilise une seule image terrestre et celle de *H. Maas* qui utilise des données altimétriques aériennes. *F. van den Heuvel* présente dans [vdH98] une approche automatique dont le but est de construire des modèles pour la réalité virtuelle. Son approche nécessite que les objets de l'image utilisée possèdent des contours parallèles et perpendiculaires. L'idée générale de l'algorithme est de reconstruire le modèle en appliquant des contraintes géométriques déterminées automatiquement aux lignes extraites des images. Une détection des points de fuite permet de déterminer les contraintes à imposer aux lignes parallèles des images. On impose au maximum trois contraintes de perpendicularité entre les trois principales directions de droites, définissant un repère orthogonal (les trois directions principales du bâtiment). La redondance des informations permet de vérifier les contraintes imposées. L'auteur présente la formulation mathématique de ces contraintes de parallélisme et perpendicularité. Il présente aussi les contraintes de symétrie et de coplanarité qui peuvent être appliquées à des lignes polygonales. L'estimation des paramètres est réalisée en deux étapes par une technique de moindres carrés, application des contraintes de parallélisme et application des contraintes de perpendicularité. Dans [vdH99], *F. van den Heuvel* ajoute la distorsion dans son modèle pour les caméras grand angle. Il propose dans cet article de séparer l'estimation de cette distorsion de l'estimation des paramètres des lignes avec les contraintes.

H. Maas et *G. Vosselman* présentent dans [MV99] deux techniques de reconstruction de bâtiments à partir de données laser haute résolution. Les données utilisées sont sous la forme d'un nuage de points $2D\frac{1}{2}$. La détection des bâtiments est réalisée à partir de seuillage et de morphologie mathématique. Les auteurs présentent une première méthode permettant de retrouver les paramètres d'un modèle paramétrique de bâtiment avec toit à deux pans en utilisant les moments du nuage de points correspondant au bâtiment. L'idée est d'associer les invariances des moments du premier et du second ordre à la forme du bâtiment pour retrouver une solution proche des paramètres du modèle. La seconde méthode permet de reconstruire des modèles polyédriques. Les auteurs proposent une technique de vote dans l'espace des paramètres pour retrouver les différents plans. Ils proposent ensuite une méthode pour extraire le détournage du bâtiment en privilégiant les droites parallèles et perpendiculaires. Le modèle est ensuite retrouvé en calculant les intersections des plans.

Department of Geomatic Engineering, University College London *T. Kim* et *J. Muller* du Geomatic Engineering^[33] présentent dans [KM95] une approche combinant une information stéréoscopique à une information monoscopique. Ils construisent un MNE à l'aide d'une méthode de corrélation pyramidale. En parallèle, ils proposent une technique monoscopique de détection de bâtiments basée sur l'extraction des lignes, la construction d'un graphe relationnel pour générer des hypothèses de bâtiment et la vérification par analyse des ombres. La fusion de ces deux informations sert à générer un MNE dense possédant des limites de bâtiment franches. Ils interpolent les surfaces délimitées par les bâtiments extraits.

Visual Geometry Group, Université d'Oxford *A Oxford*^[34], *C. Baillard*, *A. Zisserman* et *A. Fitzgibbon* proposent dans [BZF99] de reconstruire les bâtiments à partir de segments 3D. Dans une première partie, ils présentent une méthode de mise en correspondance de segments multi-images. La méthode propose en réalité d'apparier les segments extraits dans trois images, tous les résultats étant ensuite fusionnés. Une première mise en correspondance est réalisée entre les segments des deux premières images. La troisième image permet de lever l'ambiguïté entre plusieurs solutions en vérifiant la présence des segments dans cette image (par accumulation des scores de corrélation). Une technique de croissance et fusion permet ensuite de compléter les segments 3D pour définir des droites. La seconde partie de l'article présente une méthode pour reconstruire les bâtiments à partir des droites 3D définies. L'idée est d'associer à chaque droite 3D au maximum deux demi-plans correspondant aux surfaces délimitées par cette droite. Ils proposent de retrouver l'orientation de ces demi-plans en parcourant l'espace des orientations et en mesurant un score de corrélation pour chacune des orientations possibles. L'orientation choisie est celle qui possède le meilleur score. Ils regroupent ensuite les droites colinéaires et complètent le groupe de droites 3D par les intersections de plan avant de reconstruire le modèle du bâtiment. Cette approche combine les modèles et les surfaces. Les bâtiments reconstruits sont d'excellente qualité, les auteurs utilisent des images de très haute résolution (8cm).

Computer Graphics and Vision (ICG) de l'Université de Technologie de Graz *M. Maresh* et *D. Scheiblhofer* de l'ICG^[35] présentent dans [MS96a] un système d'enregistrement de façade. Le système est composé de trois lignes CCD pour l'enregistrement des façades, d'un capteur laser pour mesurer la distance entre le véhicule et les bâtiments et d'une caméra matricielle de moyenne résolution. Une étude théorique des capteurs CCD li-

néaires est présentée dans [MD96] pour justifier leurs placements et orientations. La caméra matricielle couvre une petite partie centrale du bâtiment et permet d'évaluer le mouvement du véhicule. Les auteurs reconstruisent ensuite les lignes enregistrées dans un plan virtuel. Cette projection est déterminée grâce au mouvement estimé et au mouvement latéral enregistré par le capteur laser. Leur objectif est d'utiliser les différentes images pour faire de la stéréo et éliminer les obstacles des images de façade.

3.5 Bilan de l'état de l'art

La reconstruction des bâtiments a été beaucoup étudiée. Les premières approches du problème étaient principalement basées sur les modèles. Les études portaient alors sur des zones industrielles ou militaires. Les bâtiments étaient de très faible complexité. La recherche étant encouragée par des applications de plus en plus nombreuses et exigeantes, les dernières approches tentent maintenant de résoudre le problème le plus rapidement possible pour des bâtiments plus variés et complexes. Les laboratoires se sont alors intéressés à la segmentation des MNE en plans. Le MNE permet d'avoir une quantité constante d'informations tridimensionnelles sur la forme du bâtiment. Les derniers développements tendent à combiner à la fois les approches par modèle et par surface tout en préservant une cohérence $2D\frac{1}{2}$: segmentation du MNE en morceaux de surfaces plans en s'aidant des modèles extraits des images. La plupart des laboratoires ont ensuite développé des outils semi-automatiques pour apporter des solutions rapides aux diverses applications. Ces outils permettent d'obtenir des modèles directement applicables, focalisant les recherches sur l'automatisation des différentes étapes et la convivialité des systèmes.

Les approches qui nous semblent être les plus intéressantes actuellement sont celle développée par C. Baillard, A. Zisserman et A. Fitzgibbon de l'Université d'Oxford et celle de B. Ameri et D. Fritsch de l'Université de Stuttgart. L'approche d'Oxford combine de manière très intéressante une reconstruction par modèle et une reconstruction à base de surfaces. Ils commencent par la mise en correspondance de segments, ce qui peut poser problème dans le cas où des contours n'ont pu être extraits. Ils proposent ensuite de retrouver les surfaces délimitées par ces segments en faisant varier l'orientation d'un demi-plan autour de l'axe du segment et en calculant le score de corrélation de la surface ainsi orientée. L'idée est un peu similaire à l'approche que nous avons développée pour reconstruire les points des façades (idée du *space-sweep algorithm*). On discrétise l'espace des orientations pour retrouver la solution obtenant le meilleur score dans leur cas et tous les points 3D appartenant aux façades dans notre cas. L'approche est originale et donne de bons résultats, mais nous pensons toutefois qu'il serait préférable de résoudre le problème dans le sens contraire, à savoir utiliser un modèle basé sur les surfaces pour construire un modèle grossier puis d'utiliser les segments 3D construits pour retrouver les limites réelles de l'objet. C'est l'approche que nous avons choisie tout comme B. Ameri et D. Fritsch de l'Université de Stuttgart [AF00]. Nous utilisons la même stratégie générale : segmentation d'un MNE en plans, construction du modèle polygonal puis correction ou recalage des contours sur les images. Le point intéressant de leur approche est l'outil POLY-MODELER qu'ils utilisent pour construire un modèle polyédrique 3D générique. Dans notre cas, nous avons préféré construire un modèle polygonal à partir de plans horizontaux pour assurer un résultat de reconstruction de qualité constante (du fait de la résolution et de la qualité du MNE initial).

Pour conclure, nous avons présenté un état de l'art important présentant les principales approches pour la reconstruction des bâtiments. Nous avons présenté notre approche et

positionné celle-ci par rapport aux autres approches, notamment les travaux de l'Université d'Oxford et ceux de l'Université de Stuttgart qui nous semblent être les deux approches les plus intéressantes actuellement.

3.6 Stratégie générale proposée

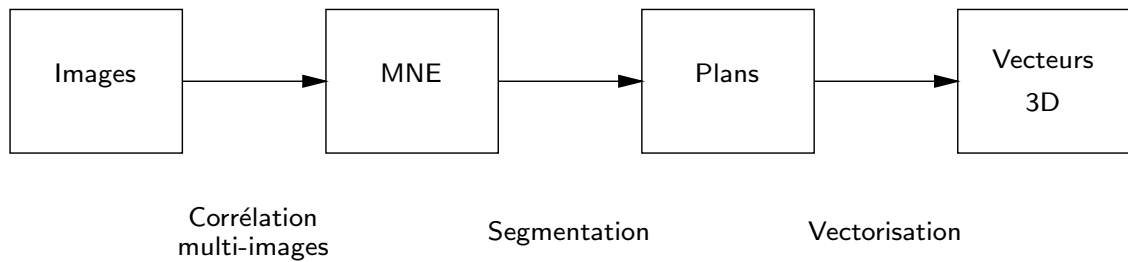


FIG. 3.4 – La stratégie générale pour reconstruire les bâtiments. La première étape permet la construction d'un MNE dense et fiable. La deuxième étape est la segmentation des bâtiments en morceaux de plans. La dernière étape est la vectorisation des contours des morceaux de plans pour obtenir la description finale du bâtiment.

Les objectifs que nous visons sont les suivants : détecter tous les bâtiments, traiter toutes les formes de bâtiment, obtenir un modèle représentatif de la forme générale du bâtiment et avoir un détournage du bâtiment de bonne qualité. Tous les outils proposés dans cette thèse ont été développés pour répondre au mieux à ces objectifs.

La stratégie générale pour reconstruire un bâtiment ou un ensemble de bâtiments qui est proposée dans cette thèse se compose de trois étapes principales, elle est présentée en figure 3.4. La première étape permet la construction d'un MNE dense et fiable. La deuxième étape est la segmentation des bâtiments en morceaux de plans. La dernière étape est la vectorisation des contours des morceaux de plans pour obtenir la description finale du bâtiment. Nous avons choisi d'utiliser une approche qui permette de préserver le maximum d'indépendance vis-à-vis de la forme des bâtiments (nous traitons toutes les formes sauf les surfaces du second ordre : toits arrondis...).

La première étape est la construction d'un Modèle Numérique d'Élévation dense et fiable (section 3.2, paragraphe *la mise en correspondance*). La construction de ce MNE a déjà été présentée précédemment [GGR97]. Ce MNE est obtenu par fusion robuste des MNE intermédiaires construits pour chaque couple d'images valide. La fiabilité de la méthode est assurée par le fort recouvrement entre les vues pour assurer une redondance importante de l'information. On extrait ensuite les bâtiments de ce MNE pour pouvoir les traiter un après l'autre, cette étape est présentée dans la section 4.5.2.

La deuxième étape est la segmentation du MNE en surfaces locales plans. L'objectif de cette étape est de décrire la scène urbaine au moyen de morceaux de surface qui vont correspondre aux différentes faces des bâtiments. L'approche proposée dans cette thèse est décrite dans le chapitre 5. Nous utilisons tout d'abord un algorithme d'exploration robuste (le RANSAC) qui construit une liste d'hypothèses de modèles à partir des données. Nous utilisons ensuite un algorithme de sélection qui permet de trouver la plus petite liste des meilleurs modèles permettant de décrire le bâtiment. La totalité des faces du modèle du bâtiment est décrite par des plans.

La troisième et dernière étape est la vectorisation. On souhaite construire un modèle polygonal pour décrire le plus correctement possible le bâtiment. Nous proposons tout d'abord une méthode d'approximation polygonale des limites de chaque morceau de surface pour obtenir un premier modèle polygonal du bâtiment. Nous proposons ensuite plusieurs méthodes de correction du modèle pour qu'il corresponde mieux au bâtiment. La vectorisation est décrite dans le chapitre 6.

Les erreurs présentes dans le MNE ont un impact important sur les résultats des étapes de segmentation et de vectorisation. Le chapitre 4 présente un outil permettant de reconstruire les façades de la scène. Nous verrons dans ce chapitre comment utiliser ces données reconstruites pour corriger le MNE initial. Cette correction facilite les tâches des deux dernières étapes de la stratégie générale. Nous verrons ensuite dans le chapitre 7 plusieurs autres utilisations possibles de ces points 3D, nous les faisons intervenir dans différentes étapes de la stratégie générale proposée.

3.7 Liens

Projets et universités américains

- ① **DARPA**, Defense Advanced Research Projects Agency,
<http://www.darpa.mil>
- ② **Le projet RADIUS**, Research And Development for Image Understanding Systems,
<http://www.ai.sri.com/~radius/>
- ③ **Le projet IUE**, Image Understanding Environment,
<http://www.aai.com/AAI/IUE/IUE.html>
- ④ **TargetJr**, autre environnement pour le traitement d'images,
<http://www.targetjr.org/>
- ⑤ **APGD**, Automatic Population of Geospatial Databases,
<http://www.ai.sri.com/~apgd/>
- ⑥ **Projet MURI**, Rapid and Affordable Generation of Terrain and Detailed Urban Feature Data,
<http://iris.usc.edu/Outlines/muri-project.html>
- ⑦ **Projet MURI**, 3D Direct Interfaces (3DDI),
<http://www.cs.berkeley.edu:80/~jfc/MURI/>
- ⑧ **Perception Program**, SRI Artificial Intelligence Center,
<http://www.ai.sri.com/perception/>
- ⑨ **IRIS Computer Vision Lab**, University of Southern California (USC),
<http://iris.usc.edu/USC-Computer-Vision.html>
- ⑩ **UMASS Computer Vision Research Laboratory**, University of Massachusetts,
<http://vis-www.cs.umass.edu/>
- ⑪ **Digital Mapping Laboratory (MAPSLab)**, Carnegie Mellon University,
<http://www.maps.cs.cmu.edu/>
- ⑫ **Intelligent Systems Laboratory**, University of Washington,
<http://george.ee.washington.edu/>
- ⑬ **Computer Vision Laboratory (CVL)**, University of Maryland,
<http://www.cfar.umd.edu/cvl/>

Une revue des travaux sur les bâtiments, triée par laboratoires

- [14] **MIT Computer Graphics Group**, Massachusetts Institute of Technology,
<http://graphics.lcs.mit.edu/index.html>
- [15] **Computer Vision Group**, UC Berkeley,
http://http.cs.berkeley.edu/projects/vision/vision_group.html
- [16] **FACADE**, Modeling and Rendering Architecture from Photographs,
<http://www.cs.berkeley.edu/~debevec/Research/>
- [17] **The Campanile Movie**, University of California at Berkeley,
<http://www.cs.berkeley.edu/~debevec/Campanile/>
- [18] **Département TSI**, Ecole Nationale Supérieure des Télécommunications (ENST),
<http://www-ima.enst.fr/>
- [19] **MATIS**, Institut Géographique National (IGN),
<http://www.ign.fr/fr/PI/recherche/matis/>
- [20] **Groupe ETIS**, Université de Cergy-Pontoise,
<http://www-etis.ensea.fr/>
- [21] **ISTAR**, Imagerie STéréo Appliquée au Relief,
<http://www.istar.com>
- [22] **PASTIS ARIANA**, INRIA Sophia Antipolis,
<http://www.inria.fr/Equipes/PASTIS-fra.html>
- [23] **BIWI**, The Swiss Federal Institute of Technology (ETH Zürich),
<http://www.vision.ee.ethz.ch/>
- [24] **Chair of Photogrammetry and Remote Sensing**, The Swiss Federal Institute of Technology (ETH Zürich),
<http://www.photogrammetry.ethz.ch/>
- [25] **PSI-VISICS**, Université Catholique de Louvain (K. U. Leuven),
<http://www.esat.kuleuven.ac.be/psi/visics/>
- [26] **IPB**, Institute of Photogrammetry de l'université de Bonn,
<http://www.ipb.uni-bonn.de/>
- [27] **IFP**, Institut für Photogrammetrie de l'université de Stuttgart,
<http://www.ifp.uni-stuttgart.de/>
- [28] **FGAN FIM**, Research Establishment for Applied Science,
<http://www.fgan.de/>
- [29] **Chair for Photogrammetry and Remote Sensing**, université de Munich,
<http://www.photo.verm.tu-muenchen.de/>
- [30] **TNT**, université de Hannover,
<http://www.tnt.uni-hannover.de/>
- [31] **AIDA**, Automatic Image Data Analyser,
<http://www.tnt.uni-hannover.de/soft/tnt/aida/>
- [32] **Chair of Photogrammetry and Remote Sensing**, université technologique de Delft,
<http://www.geo.tudelft.nl/>
- [33] **Department of Geomatic Engineering**, University College London (UCL),
<http://www.ge.ucl.ac.uk/>
- [34] **Visual Geometry Group**, Université d'Oxford,
<http://www.robots.ox.ac.uk:80/~vgg/>
- [35] **Computer Graphics and Vision**, Université de Technologie de Graz (ICG),
<http://www.icg.tu-graz.ac.at/>

Chapitre 4

Reconstruction de façades

Ce chapitre présente un outil pour reconstruire les façades d'un bâtiment. Les façades des bâtiments sont des surfaces tridimensionnelles de très forte pente dans le système stéréoscopique d'acquisition. Ces surfaces apparaissent très déformées dans les images aériennes et ne sont généralement pas utilisées. Ces déformations sont différentes entre deux images, on ne peut les apparier pour reconstruire les points 3D qui leur correspondent. Ces surfaces étant de tailles comparables à celles des toits dans les images, nous souhaitons extraire la quantité importante d'information qu'elles représentent. Nous proposons dans ce chapitre une stratégie pour utiliser ces surfaces en reconstruisant les points 3D qui leur correspondent.

Dans un premier temps, nous présenterons en détail les difficultés de la mise en correspondance de ces surfaces verticales. Nous étudierons ensuite un moyen de construire des images dans lesquelles une façade de la scène ne sera plus déformée, leur reconstruction sera alors possible. Après la mise en correspondance de ces nouvelles images, nous étudierons la localisation des points reconstruits et la sensibilité de la reconstruction. Nous présenterons enfin une stratégie permettant de reconstruire automatiquement toutes les façades visibles de la scène, sans aucune connaissance sur l'orientation des façades et la forme du bâtiment traité. Comme première application directe, nous proposons dans ce chapitre d'utiliser les points 3D reconstruits pour corriger les bords des bâtiments dans un MNE. Cette méthode s'inscrit dans la première étape de notre stratégie, obtenir un MNE dense et fiable. Nous verrons dans le chapitre 7 que nous pouvons aussi utiliser les points reconstruits sur les façades dans d'autres étapes de notre stratégie générale de reconstruction des bâtiments.

4.1 Problèmes liés aux façades en stéréoscopie aérienne

La mise en correspondance des images dans un environnement urbain Pour obtenir une information dense sur la scène étudiée, on utilise des méthodes de mise en correspondance surfaciques, qui se proposent généralement de fournir un appariement pour chaque point de l'image de référence. En imagerie aérienne ou spatiale, la méthode la plus répandue est basée sur le calcul d'un score de corrélation. Pour retrouver les points qui se correspondent, la scène doit respecter un certain nombre d'hypothèses ou de contraintes (contrainte lambertienne, surfaces fronto-parallèles et continues, continuité figurale, ...). Dans la pratique, ces contraintes sont plutôt lâches car elles dépendent principalement de la nature de la scène. Elles ne peuvent pas être strictement satisfaites, mais on les utilise pour obtenir une solution plus ou moins conforme à la réalité.

La grande caractéristique de la mise en correspondance par corrélation est de retrouver principalement les surfaces dites "fronto-parallèles" de la scène. Dans certaines applications, la reconstruction de ces surfaces est largement suffisante pour décrire et comprendre la scène. L'utilisation de photographies aériennes d'une scène urbaine nous place dans une configuration particulière du système de caméras. Les prises de vues sont réalisées le long de lignes de vol horizontales effectuées à une altitude élevée et la visée de la caméra est verticale. Les surfaces fronto-parallèles au plan image du capteur sont les surfaces horizontales de la scène. Dans un environnement urbain, elles correspondent principalement aux toits des bâtiments, aux rues ou plus généralement au sol. On ne peut pas reconstruire les surfaces verticales car ce sont des surfaces de très forte pente. La scène reconstruite est uniquement décrite par les surfaces proches de l'horizontale, les toits et le sol. La plupart des approches de reconstruction de bâtiments utilisent uniquement ces points caractéristiques. Dans certains cas, ces données ne sont pas suffisantes pour permettre de reconstruire un modèle représentatif du bâtiment.

Puisqu'aucune surface verticale ne peut être reconstruite, on représente couramment les données reconstruites dans un espace $2D\frac{1}{2}$ défini par le plan horizontal, c'est une représentation explicite de la surface 3D (section 5.1.1). A chaque couple de coordonnées (X,Y) correspond au maximum une altitude Z . La scène 3D est donc complètement décrite par une image dont la valeur associée à chaque pixel correspond à l'altitude de l'unique point de la scène qui possède ces coordonnées. On appelle ces images ou cartes 3D des Modèles Numériques, comme le *Modèle Numérique d'Élévation* (MNE) présenté en figure 4.1. Ceci permet de simplifier la représentation de l'espace 3D et de structurer la scène reconstruite.

Les problèmes aux limites des bâtiments La mise en correspondance par corrélation est un outil simple et puissant pour obtenir une reconstruction dense et fiable. Malheureusement, il arrive souvent que l'information disponible sur les toits et le sol ne suffise pas à la reconstruction des bâtiments. Le manque de texture, le manque de dynamique dans les ombres ou les textures répétitives peuvent poser problème, mais les principales difficultés apparaissent aux bords des bâtiments. Elles sont dues tout d'abord à la présence de discontinuités de surfaces (les bords des bâtiments). Le score de corrélation se calculant sur un voisinage, les discontinuités de profondeur peuvent tromper la mesure. En regardant un MNE (figure 4.1), on s'aperçoit que la surface reconstruite déborde des limites réelles du bâtiment, surtout dans les ombres. De plus, les bords prennent une forme arrondie, les limites ne sont plus franches et linéaires.

Un premier moyen de contrer le problème présent aux limites des bâtiments est d'utiliser les contours des images. On suppose que les discontinuités physiques de la scène, comme les limites des bâtiments, donnent lieu à des discontinuités d'intensité. Les approches de mise en correspondance basées sur les modèles extraient les contours pour reconstruire des coins, des segments ou des droites. Elles se servent ensuite de ces contours pour reconstruire les bâtiments. On peut coupler ces informations à celle du MNE pour obtenir une meilleure reconstruction. D'autres méthodes comme celles présentées dans la section 2.3.1 permettent de prendre en compte l'information des contours dans le calcul des scores de corrélation. On peut utiliser des fenêtres de taille adaptative s'arrêtant aux contours, des masques pondérés, ... Toutes ces méthodes permettent d'améliorer la reconstruction mais vont rencontrer des difficultés lorsqu'il n'y a pas de contour, s'il y en a plusieurs ou si les bords du bâtiment sont trop texturés. De plus, elles ne prennent pas en compte les déformations des surfaces de forte pente. Le système proposé dans ce chapitre est un complément

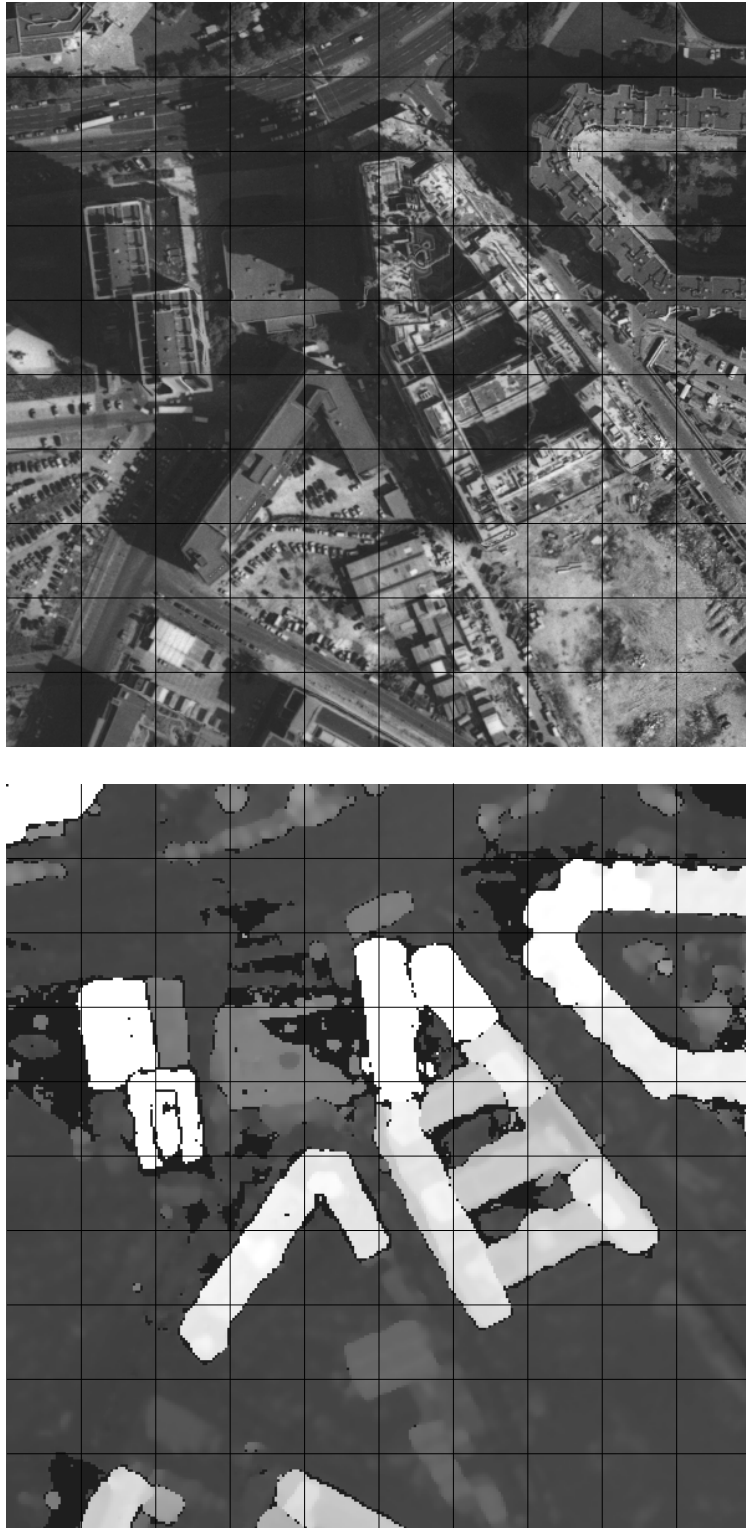


FIG. 4.1 – Un exemple d'ortho-image (A) et de MNE, Modèle Numérique d'Élévation, (B) reconstruits automatiquement par la chaîne de production d'ISTAR. Ces deux images correspondent à la même zone géographique, les grilles permettent de visualiser la correspondance entre ces deux images. Dans le MNE, la couleur du pixel est liée à l'altitude du point de la scène. Plus la couleur est claire et plus l'élévation est importante.

des approches de mise en correspondance par corrélation. Il permet à ces techniques d'appariement de reconstruire les surfaces verticales pour délimiter et reconstruire le bâtiment.

Les façades des bâtiments en imagerie aérienne Les façades des bâtiments dans les images sont sujettes à deux types de problèmes. Tout d'abord, elles sont invisibles dans la région centrale des images puisqu'elles sont perpendiculaires à l'axe optique. Elles ne peuvent dès lors nous apporter aucune information supplémentaire pour la reconstruction du bâtiment. Dans les parties de l'image éloignées du point principal, les façades apparaissent progressivement et leurs surfaces deviennent comparables à celles des toits dans le bord des images. Elles peuvent alors nous apporter une information complémentaire ou supplémentaire. Leur reconstruction n'est malheureusement pas facile. Ces surfaces étant des surfaces de forte pente, les différentes images d'une façade de la scène sont fortement déformées, et dans des proportions différentes comme le présente la figure 4.2. Lors de la mise en correspondance de deux images, les façades ne peuvent donc pas être appariées directement. Nous présentons dans ce chapitre un moyen de mesurer et de compenser les déformations qui gênent la mise en correspondance. Nous construisons de nouvelles images de telle manière que la façade soit perçue de manière identique dans chacune des vues. Nous allons voir dans la suite qu'il est possible de reconstruire les points 3D des façades de la scène à partir de l'unique hypothèse de verticalité des façades.

4.2 Présentation géométrique du problème

Considérons un système de projection simple qui correspond au modèle perspectif linéaire de la figure 4.3 [Alo90]. Nous supposons que la scène observée est un plan à une distance fixe. Cette simplification nous apporte des propriétés géométriques qui vont nous permettre de mieux comprendre le problème que nous voulons résoudre. La scène observée par le capteur est donc considérée ici comme étant une droite parallèle à la droite image. La distance séparant la scène du centre optique est fixe et notée Z_c . La focale du capteur est notée f . Dans la suite et conformément à la figure 4.3, x représente l'abscisse image, X l'abscisse du point 2D dans la scène et α l'angle défini par le rayon optique du capteur et l'axe optique. Nous définissons le cône optique par les deux rayons optiques passant par les point $x - \delta x$ et x .

Nous nous intéressons aux surfaces de la scène projetées dans l'image dans ce cône optique. La variation δx de l'image représente une surface de la scène de taille δX si celle-ci est horizontale et de taille δZ lorsque celle-ci est verticale comme le montre la figure 4.3. La scène et la droite image étant parallèles, on en déduit que *les variations $\delta \mathbf{x}$ et $\delta \mathbf{X}$ sont proportionnelles*.

En imagerie aérienne, toutes les prises de vues sont effectuées approximativement à la même altitude et observent la même scène. Les surfaces fronto-parallèles aux plans images sont les surfaces horizontales de la scène. Lorsque tous les plans images des différentes prises de vues sont échantillonnés avec un pas régulier, les surfaces horizontales de la scène auront la même taille dans toutes les images. Ceci se vérifie dans un couple d'images comme celui de la figure 4.2 et c'est la raison pour laquelle nous pouvons mettre en correspondance ces surfaces. On peut alors se poser la question suivante : existe-t-il un échantillonnage particulier des images qui permette de préserver les surfaces verticales ? La scène et l'image



FIG. 4.2 – Cette paire d'images présente les problèmes que nous rencontrons avec les façades. Les surfaces verticales sont déformées et n'ont pas la même taille et orientation dans les différentes vues, ce qui rend leur appariement difficile.

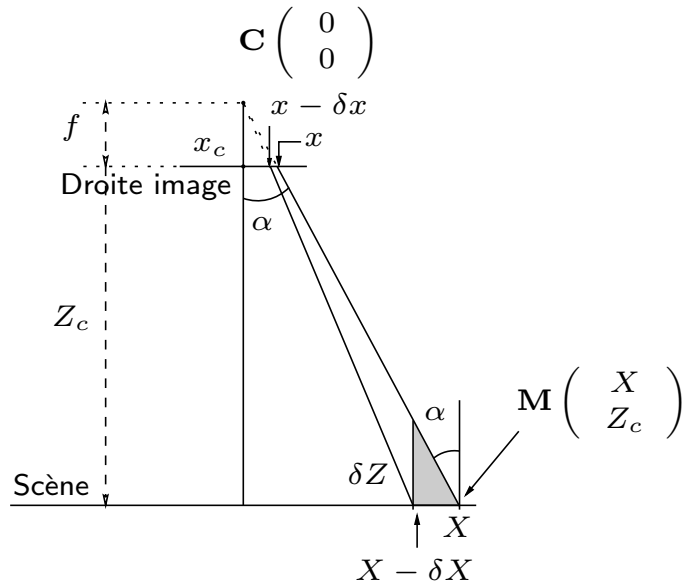


FIG. 4.3 – Présentation de la projection utilisée. On utilise un modèle de projection perspectif linéaire. La scène que nous considérons est une droite parallèle à la droite image. On s'intéresse à la surface de la scène correspondant à un pixel de l'image (dans le rayon optique).

étant parallèles, on peut écrire :

$$\frac{\delta X}{\delta x} = \frac{Z_c}{f} = \text{constante} \quad (4.1)$$

L'axe optique étant vertical, les façades sont toutes parallèles à celui-ci. L'angle α que nous avons défini précédemment correspond aussi à l'angle formé entre la droite de projection (**CM**) et la droite verticale passant par le point M de la scène (figure 4.3). Nous pouvons alors écrire :

$$\tan \alpha = \frac{x - x_c}{f} = \frac{X - X_c}{Z_c} = \frac{\delta X}{\delta Z} \quad (4.2)$$

Ces deux équations vont nous permettre de répondre à la question posée. L'équation 4.1 exprime que les variations δx et δX sont proportionnellement dépendantes. L'équation 4.2 permet de relier les tailles des surfaces horizontales (δX) et verticales (δZ) représentées dans le cône optique, toutes deux étant liées au même point M . Cette équation dépend de la position du point M de la scène par rapport à l'axe optique.

Deux choix sont possibles : on veut préserver (ou privilégier) les surfaces horizontales ou alors les surfaces verticales. Le premier choix est présenté dans la figure 4.4 A. On veut que l'image preserve les éléments horizontaux, c'est-à-dire qu'à un pixel de l'image $\delta x = 1$ corresponde une surface horizontale de taille fixe. En imposant la contrainte $\delta X = K$, on déduit de manière triviale que l'image doit être échantillonnée elle aussi de manière constante ($\delta x = \frac{f\delta X}{Z_c} = \frac{fK}{Z_c}$). C'est l'échantillonnage couramment réalisé du plan image (ou d'une ligne dans le cas présent) pour obtenir les images initiales de la scène. Les surfaces qui sont échantillonnées de manière constante sont les toits et le sol de la scène, elles pourront alors être facilement appariées. Lorsque les pixels représentent des surfaces verticales, ces

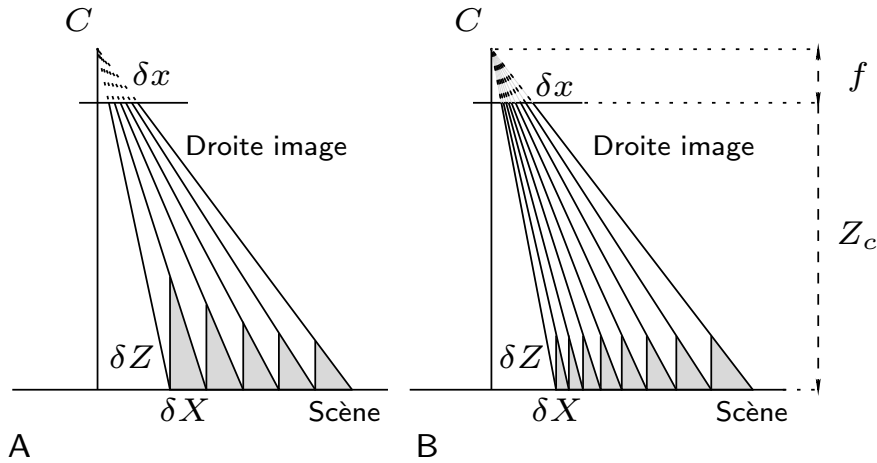


FIG. 4.4 – *Choix du critère d'échantillonnage* : *A* représente un échantillonnage avec δX constant, *B* représente un échantillonnage avec δZ constant. Dans le premier cas, la taille des éléments verticaux perçus diminue en s'éloignant du centre optique. Dans le second cas, la taille des éléments horizontaux perçus augmente en s'éloignant du centre optique.

surfaces auront des tailles irrégulières ($\delta Z = \frac{fK}{(x-x_c)}$). Plus on est proche du point principal et plus l'aire de surface verticale représentée par un pixel est grande. D'un pixel à l'autre, la surface verticale observée est complètement différente. Il est donc impossible de mettre en correspondance les façades en échantillonnant les images de manière régulière, ce que nous avons précédemment constaté.

Si on décide de préserver les surfaces verticales (figure 4.4 B), on impose cette fois $\delta Z = K$ (où K est une constante choisie). Sous cette contrainte, on peut calculer les tailles des éléments horizontaux δX et δx correspondant à chaque élément vertical : $\delta X = \frac{K(x-x_c)}{f}$ et $\delta x = \frac{K(x-x_c)}{Z}$ qui dépendent l'un et l'autre de leur position par rapport à l'axe optique. Pour échantillonner l'image de manière à ce que chaque pixel représente une surface verticale unitaire (ou constante), il faut utiliser une grille irrégulière. Le pas d'échantillonnage doit être variable comme le montre la figure 4.4 B. Cette fois-ci, ce sont les surfaces horizontales qui auront des tailles irrégulières, et qui ne pourront donc pas être appariées.

Surfaces d'orientation quelconque Dans un contexte plus général on peut à présent considérer des éléments $\delta\theta$ d'orientation quelconque. Nous cherchons dans ce cas à échantillonner l'image de manière à obtenir des pixels images représentant des éléments de taille constante d'une surface d'inclinaison fixe par rapport à l'axe optique. Cette inclinaison est définie par rapport à la scène par l'angle θ comme le présente la figure 4.5. L'équation 4.2 peut être réécrite comme suit :

$$\tan \alpha = \frac{x - x_c}{f} = \frac{\delta X - \delta S \cos \theta}{\delta S \sin \theta} \quad (4.3)$$

De la même manière que précédemment, nous choisissons de rééchantillonner l'image en imposant la contrainte $\delta S = K$ (où K est une constante choisie). On obtient alors $\delta X = \frac{K}{f}((x - x_c) \sin \theta + f \cos \theta)$ et $\delta x = \frac{K}{Z_c}((x - x_c) \sin \theta + f \cos \theta)$. Une fois encore, pour que chaque pixel de l'image corresponde à une surface non horizontale de la scène fixe, il faut

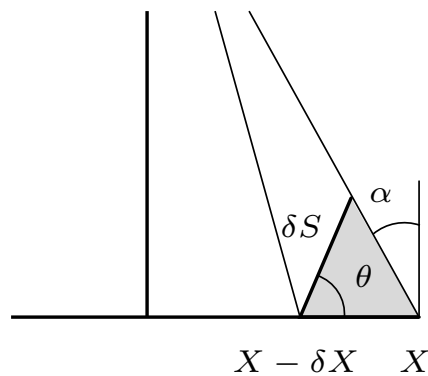


FIG. 4.5 – Étude des surfaces de la scène dont l'inclinaison par rapport à l'axe optique est constante.

que le pas d'échantillonnage tienne compte de la position de la surface par rapport au centre optique.

Cette présentation géométrique d'un modèle de caméra simple permet d'interpréter l'acquisition des surfaces verticales et de comprendre pourquoi il est difficile, voire même impossible de mettre en correspondance les façades. Le phénomène de déformation provient de la méthode d'échantillonnage du plan image, ou du film. Il est bien entendu impossible techniquement de réaliser un système d'échantillonnage permettant de préserver les façades. Par contre, nous allons poursuivre cette étude et montrer comment rééchantillonner les images initiales pour obtenir des images dans lesquelles les façades ne sont plus déformées.

4.3 Étude du problème de transformation des images

Cette section présente le moyen de construire de nouvelles images des façades. Dans ces nouvelles images, les projections d'une façade doivent avoir le même aspect pour pouvoir être appariées. Dans un premier temps, nous poursuivons l'étude du problème avec le capteur linéaire présenté dans la section précédente 4.2 et la scène particulière. Nous étudierons ensuite le cas général d'un modèle de projection tridimensionnel, et proposerons une solution pour le cas du modèle sténopé.

4.3.1 Étude du capteur linéaire

Le problème traité maintenant est le suivant : comment rééchantillonner les deux lignes images pour que les façades aient la même taille? La seule hypothèse que nous formulons est la verticalité des façades. Pour que deux images de façades aient la même taille dans les deux lignes, il faut que la variation en altitude (le long des façades) soit constante et la même le long de chacune des lignes, $\delta Z = \text{cte}$ pour $\delta x = \text{cte}$ dans les deux lignes. Le principe est de rééchantillonner les images sous cette contrainte, la variation d'altitude δZ doit être constante et la même dans les deux nouvelles lignes images. Nous appellerons par la suite cette transformation : la transformation des façades.

Mise en équation

Considérons les lignes images \mathbf{I}_1 et \mathbf{I}_2 extraites de deux photos \mathbf{Photo}_1 et \mathbf{Photo}_2 d'une même scène \mathbf{S} dont les points de vues sont distincts. On recherche les deux fonctions \mathbf{G}_i , pour $i = \{1,2\}$, appliquées respectivement aux images \mathbf{I}_1 et \mathbf{I}_2 telles que :

$$\mathbf{G}_i : \begin{array}{l} I_i \longmapsto I'_i \\ x_i \longrightarrow x'_i = \mathbf{g}_i(x_i) \end{array}$$

avec les deux contraintes suivantes :

$$\frac{\delta x'_i}{\delta Z} = K \quad \text{pour } i = \{1,2\} \quad (4.4)$$

en conservant les notations de la figure 4.4, x est la coordonnée de l'image initiale I , x' la coordonnée dans la nouvelle image I' et K est une constante choisie. Nous exprimons ensuite ces contraintes sous la forme des équations à dérivées partielles suivantes :

$$\frac{\partial \mathbf{g}_i(x_i)}{\partial Z} = K \quad \text{pour } i = \{1,2\} \quad (4.5)$$

Le problème présenté consiste maintenant à déterminer les transformations $\mathbf{g}_i(x_i)$ nous permettant de passer d'une image correspondant à la figure 4.4 A à une image correspondant à la figure 4.4 B.

Étude de la transformation à appliquer aux images

L'équation de projection d'un point $P(X,Z)$ de la scène bidimensionnelle \mathbf{S} dans une ligne de l'image \mathbf{I}_i est la suivante :

$$x_i = f_{x_i} \frac{X}{Z} + x_{ic}$$

f_{x_i} correspond au facteur multiplicatif selon l'axe des abscisses et x_{ic} est l'abscisse du point principal de l'image \mathbf{I}_i . On recherche à présent l'application $\mathbf{g}_i(x_i)$ vérifiant la contrainte (4.5). Réécrivant cette contrainte sous forme décomposée et dérivant l'équation de projection, on obtient facilement l'équation suivante :

$$\begin{aligned} \frac{\partial \mathbf{g}_i(x_i)}{\partial Z} &= \frac{\partial x_i}{\partial Z} \times \frac{\partial \mathbf{g}_i(x_i)}{\partial x_i} \\ &= -f_{x_i} \frac{X}{Z^2} \times \frac{\partial \mathbf{g}_i(x_i)}{\partial x_i} \\ &= K \end{aligned}$$

qui nous permet maintenant de redéfinir une nouvelle contrainte exprimée cette fois-ci par rapport à l'abscisse x_i de l'image \mathbf{I}_i .

$$\frac{\partial \mathbf{g}_i(x_i)}{\partial x_i} = -\frac{K \times Z^2}{f_{x_i} \times X} \quad (4.6)$$

Reprenant l'équation de projection, on exprime l'abscisse X de l'espace bidimensionnel en fonction de l'altitude Z et de l'abscisse image x_i :

$$X = \frac{(x_i - x_{ic})Z}{f_{x_i}}$$

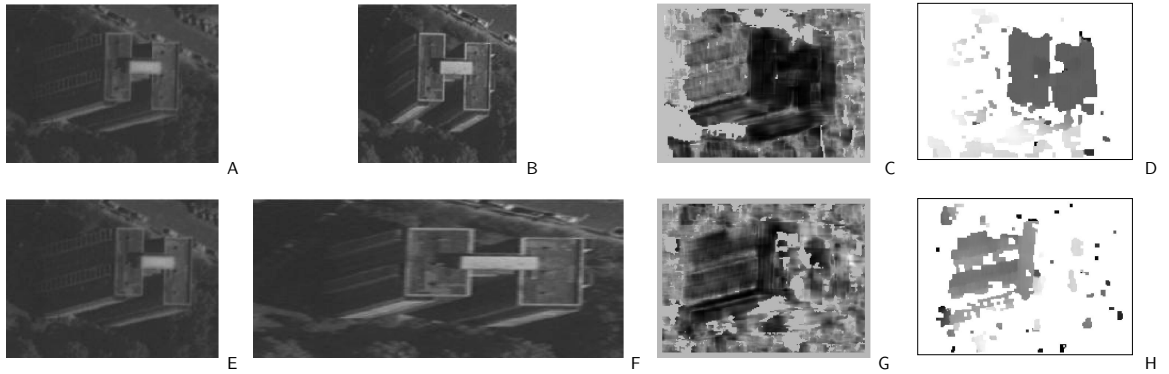


FIG. 4.6 – Résultats obtenus avec la transformation calculée le long d’une courbe épipolaire. La première ligne (images A, B, C et D) présente les résultats d’appariement des images initiales et la seconde (E, F, G et H) présente les résultats après rééchantillonnage des images. A et B sont les images originales. C représente la carte des scores de corrélation pour ces deux images et D la carte de disparité correspondante après seuillage. On voit sur cette carte que seul le toit a ”corrélé”. E et F sont les images obtenues après transformation. On remarque que les façades sont de même taille. G et H sont respectivement la carte des scores de corrélation et la carte de disparité correspondante obtenues pour le couple E F. On voit que c’est la façade qui est reconstruite.

pour l’injecter dans la contrainte. On obtient enfin l’équation générale suivante permettant de définir la dérivée selon x_i de la fonction de passage $\mathbf{g}_i(x_i)$ recherchée :

$$\frac{\partial \mathbf{g}_i(x_i)}{\partial x_i} = -\frac{K Z(x_i)}{(x_i - x_{ic})} \quad (4.7)$$

Nous pouvons donc maintenant estimer cette fonction $\mathbf{g}_i(x_i)$ en intégrant l’équation 4.7. La fonction $Z(x_i)$ correspond à la distance du point x_i au centre optique de notre caméra. On peut connaître ou approximer cette fonction par reconstruction de la scène à partir des surfaces horizontales. Pour se faire une idée sur la forme de la fonction $\mathbf{g}_i(x_i)$, on se place dans le cas d’une scène simple composée d’une droite d’équation $Z = Z_c$ avec Z_c constant. La projection de la scène dans le plan image est alors de type perspective faible. Dans ces conditions, l’équation de projection devient simplement :

$$x = f_{x_i} \frac{X}{Z_c} + x_{ic}$$

avec Z_c constant, on obtient la transformation $\mathbf{g}_i(x_i)$ suivante :

$$\mathbf{g}_i(x_i) = -K Z_c \ln(x_i - x_{ic}) \quad (4.8)$$

L’équation 4.8 correspondant au traitement à appliquer pour redresser les façades revient à rééchantillonner les images de manière logarithmique. Le résultat de cette transformation logarithmique est de dilater les zones de l’image en fonction de leur position par rapport au centre optique : les parties peu visibles (celles qui sont proches du centre optique) seront plus dilatées que les zones les plus visibles (celles qui sont les plus éloignées du point principal : les bords des images). Les figures 4.6 et 4.7 nous présentent les résultats obtenus sur une paire d’images. Dans la figure 4.6, on utilise les images d’un bâtiment. Les deux

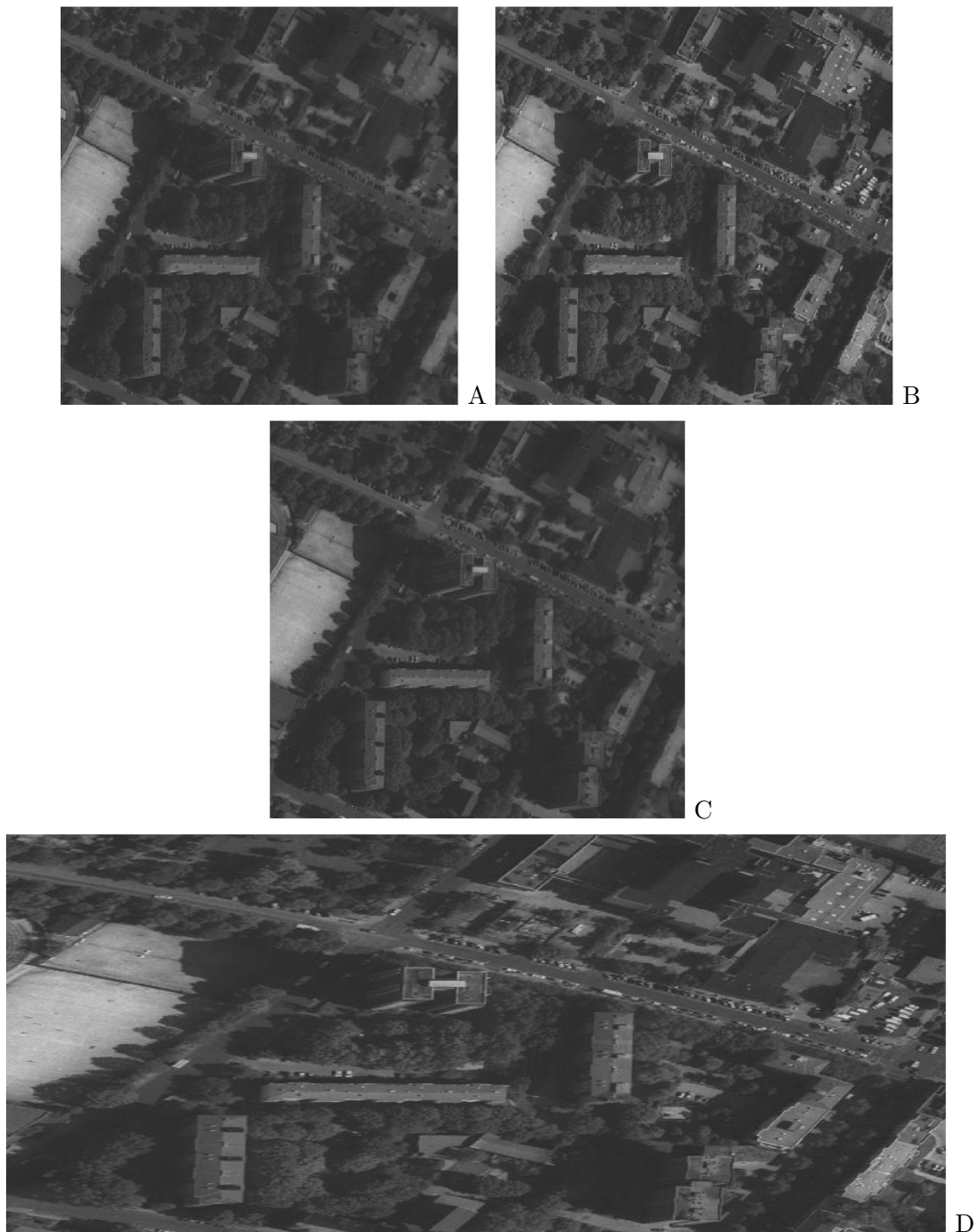


FIG. 4.7 – Résultats obtenus en appliquant la transformation aux courbes épipolaires. A et B sont les deux images initiales et C et D les nouvelles images. Les façades des bâtiments n'ont pas la même taille dans les images initiales. Dans les nouvelles images, nous voyons que les façades des bâtiments dont l'orientation est perpendiculaire aux lignes sont à présent de surface équivalente. Les toits (correspondant à des surfaces horizontales) sont "étirés" et leurs surfaces sont perçues différemment.

images initiales sont rééchantillonnées selon l'équation 4.8 avec la même constante K . On applique l'algorithme de mise en correspondance par corrélation aux deux couples d'images. En regardant les scores de corrélation et les cartes de disparité des deux couples d'images, on voit que la corrélation qui appariait le toit pour les images initiales apparie maintenant la façade de ce même bâtiment. La figure 4.7 présente les résultats de la transformation appliquée à une zone plus importante.

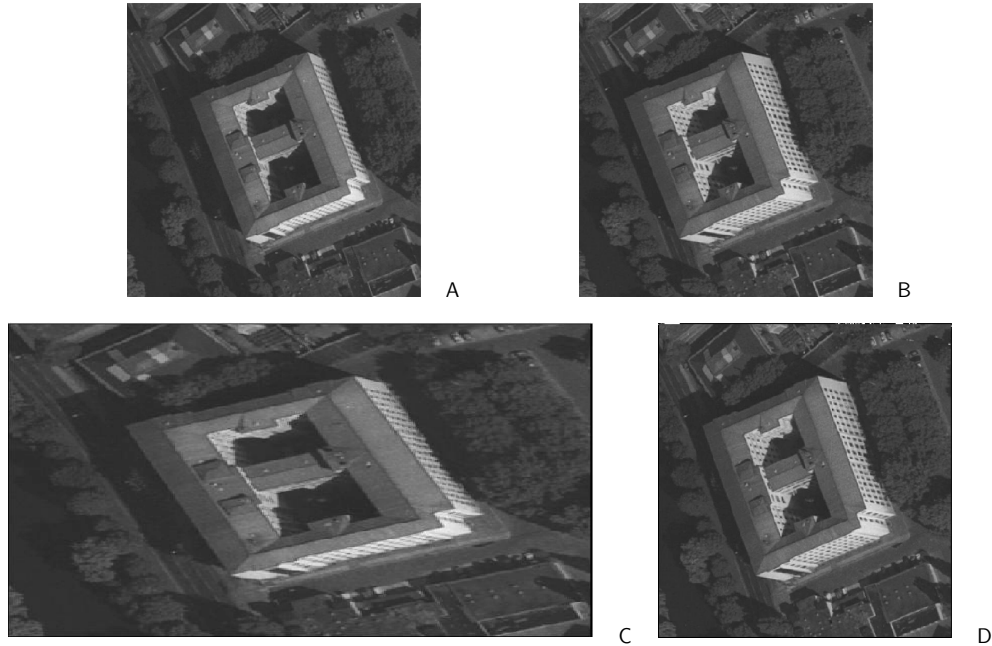


FIG. 4.8 – *Effet de rotation pour les façades dont l'orientation n'est pas perpendiculaire aux lignes images. A et B correspondent aux images initiales. Les images C et D correspondent à leurs transformées. On voit que les façades de droite ont la même taille dans les nouvelles images, mais elles sont sujettes à un effet de rotation.*

Problème de rotation des bâtiments

Le cas du modèle linéaire présenté précédemment peut correspondre à l'étude d'une acquisition du capteur 2D linéaire (ou push-broom, cf. [HG94]). Il peut aussi correspondre à la mise en correspondance de deux droites épipolaires acquises par un capteur sténopé. Nous avons supposé jusqu'à présent que le problème ne comportait qu'une seule variable, la coordonnée x_i . Ceci nous amène à supposer que toutes les façades sont perpendiculaires aux lignes des images. Lorsque les bâtiments ne vérifient pas cette condition, il apparaît un effet de rotation comme le présente la figure 4.8. Cet effet sera expliqué dans l'étude du capteur sténopé, en section 4.3.2. Nous allons maintenant tenir compte de l'orientation des bâtiments dans la mise en équation du problème.

4.3.2 Étude générale d'un capteur projectif 3D

Nous considérons maintenant nos façades comme des surfaces verticales orientées dans l'espace tridimensionnel. Les surfaces de la scène 3D correspondant aux façades (surfaces

verticales orientées) doivent être perçues identiquement dans les images. La stratégie opère en deux étapes. La première étape consiste à quantifier les déformations des façades dues à la projection. On reconstruit ensuite dans la seconde étape de nouvelles images de la scène dans lesquelles les déformations sont compensées.

Projection des façades 3D dans les images

Nous commençons par considérer notre scène comme une surface continue \mathbf{G} représentant le sol. En tout point $M(X,Y,Z)$ de la surface \mathbf{G} on définit une surface unitaire verticale orientée, qui représente un morceau de façade unitaire de la scène. Cette surface unitaire est définie par deux vecteurs : un vecteur $\vec{e} = (0,0,1)^T$ représentant l'élévation (orientation de la verticale) et un vecteur $\vec{h} = (X_h, Y_h, 0)^T$ (avec $X_h = \cos \theta$, $Y_h = \sin \theta$ et $X_h^2 + Y_h^2 = 1$) représentant l'orientation θ de la façade dans le plan horizontal.

On se fixe une orientation de façade et on définit maintenant sur la surface du sol \mathbf{G} deux champs de vecteurs 3D suivants : le champ d'élévation représentant l'orientation de la verticale et le champ d'orientation représentant une orientation de façade dans le plan horizontal. Ces deux champs de vecteurs sont constants. Le champ d'élévation 3D $\vec{\mathbf{E}}$ est représenté en tout point par le vecteur \vec{e} et le champ 3D d'orientation $\vec{\mathbf{H}}$ est représenté en tout point par le vecteur \vec{h} . Ces deux champs de vecteurs sont orthogonaux et générateurs de "façades" pour l'orientation fixée (figure 4.9).

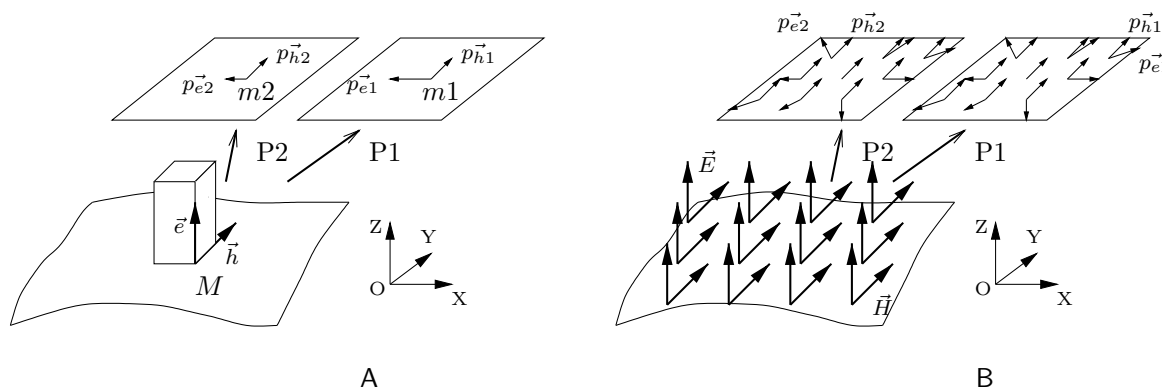


FIG. 4.9 – Définition de la surface unitaire et des différents champs de vecteurs 2D et 3D : A présente la projection d'une surface unitaire de la scène dans les différentes images. Cette surface est définie par les couples de vecteurs (\vec{e}, \vec{h}) . B présente les champs de vecteurs. Le champ de vecteurs 3D $\vec{\mathbf{E}}$ est le champ d'élévation, il est représenté en tout point du sol \mathbf{G} par le vecteur \vec{e} . Le champ de vecteurs 3D d'orientation $\vec{\mathbf{H}}$ est représenté en tout point du sol \mathbf{G} par le vecteur \vec{h} . Ces deux champs 3D définissent les façades dans la scène. Les projections de ces deux champs 3D dans l'image nous donnent deux champs de vecteurs 2D dans chaque image \mathbf{I}_i : $\vec{\mathbf{p}}_{e_i}$ et $\vec{\mathbf{p}}_{h_i}$ qui nous permettent de quantifier les déformations dues à la projection.

On s'intéresse maintenant à la projection de ces deux champs dans les images. On définit deux champs de vecteurs 2D pour chaque image \mathbf{I}_i (avec $i = \{1,2\}$) que nous nommerons $\vec{\mathbf{p}}_{e_i}$ et $\vec{\mathbf{p}}_{h_i}$ correspondant respectivement aux projetés des champs 3D $\vec{\mathbf{E}}$ et $\vec{\mathbf{H}}$ par la projection \mathbf{Pr}_i . Ces deux champs 2D permettent de quantifier les déformations des façades orientées en tout point de chaque image \mathbf{I}_i .

La projection \mathbf{Pr}_i d'un point $M(X,Y,Z)$ de \mathcal{R}^3 au point $m_i(x,y)$ de \mathcal{R}^2 dans l'image \mathbf{I}_i est définie comme suit :

\mathbf{Pr}_i :

$$\begin{aligned} M &\longrightarrow m_i \\ (X,Y,Z) &\longrightarrow (x_i,y_i) = \mathbf{Pr}_i(X,Y,Z) \end{aligned}$$

telle que

$$\mathbf{Pr}_i(X,Y,Z) = (\mathbf{Px}_i(X,Y,Z), \mathbf{Py}_i(X,Y,Z)) \quad (4.9)$$

On peut estimer les champs 2D $\vec{\mathbf{p}}_{e_i}$ et $\vec{\mathbf{p}}_{h_i}$, projections des champs 3D $\vec{\mathbf{E}}$ et $\vec{\mathbf{H}}$ dans l'image \mathbf{I}_i par \mathbf{Pr}_i , de la manière suivante (figure 4.9) :

$$\vec{\mathbf{p}}_{e_i} = \begin{pmatrix} \mathbf{x}_{e_i}(x,y) \\ \mathbf{y}_{e_i}(x,y) \end{pmatrix} = \text{Jac}(\mathbf{Pr}_i) \vec{\mathbf{E}} \quad (4.10)$$

$$\vec{\mathbf{p}}_{h_i} = \begin{pmatrix} \mathbf{x}_{h_i}(x,y) \\ \mathbf{y}_{h_i}(x,y) \end{pmatrix} = \text{Jac}(\mathbf{Pr}_i) \vec{\mathbf{H}} \quad (4.11)$$

où $\text{Jac}(\mathbf{Pr}_i)$ est le jacobien de la projection \mathbf{Pr}_i et est défini comme suit :

$$\text{Jac}(\mathbf{Pr}_i) = \begin{pmatrix} \frac{\partial \mathbf{Px}_i}{\partial X}(X,Y,Z) & \frac{\partial \mathbf{Px}_i}{\partial Y}(X,Y,Z) & \frac{\partial \mathbf{Px}_i}{\partial Z}(X,Y,Z) \\ \frac{\partial \mathbf{Py}_i}{\partial X}(X,Y,Z) & \frac{\partial \mathbf{Py}_i}{\partial Y}(X,Y,Z) & \frac{\partial \mathbf{Py}_i}{\partial Z}(X,Y,Z) \end{pmatrix} \quad (4.12)$$

Les champs 3D $\vec{\mathbf{E}}$ et $\vec{\mathbf{H}}$ sont constants et connus. Les composantes des deux champs 2D $\vec{\mathbf{p}}_{e_i}$ et $\vec{\mathbf{p}}_{h_i}$ définis dans l'image \mathbf{I}_i dépendent donc uniquement du modèle de projection \mathbf{Pr}_i utilisé.

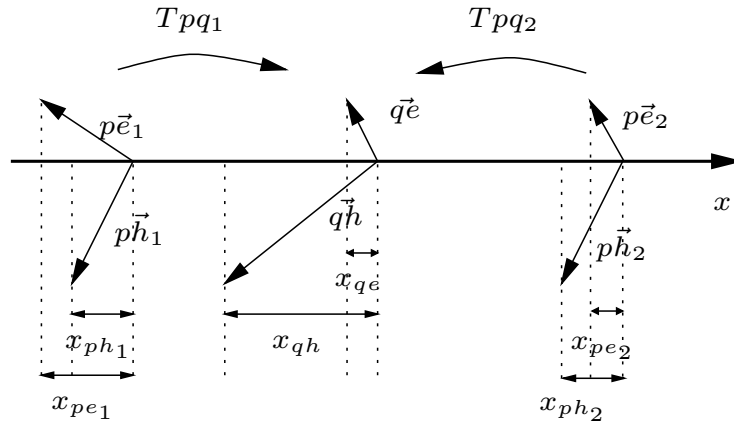


FIG. 4.10 – Transformation des façades, les façades des différentes images rectifiées sont mises dans un même référentiel en préservant la géométrie épipolaire. Les transformations \mathbf{T}_{pqi} permettent de passer du référentiel défini par les champs de vecteurs 2D $(\vec{\mathbf{p}}_e, \vec{\mathbf{p}}_h)$ au référentiel $(\vec{\mathbf{q}}_e, \vec{\mathbf{q}}_h)$.

Transformation des façades projetées

On souhaite maintenant appliquer une transformation à chacune des deux images du couple initial \mathbf{I}_1 et \mathbf{I}_2 et obtenir deux nouvelles images \mathbf{I}'_1 et \mathbf{I}'_2 dans lesquelles les projections de la façade orientée choisie sont similaires. La figure 4.10 nous présente le Redressement des façades, qui nous permet de construire ces images. Nous redressons chaque image I_i par la transformation affine $\mathbf{T}_{\mathbf{p}\mathbf{q}_i}$, qui permet de passer du référentiel $(\vec{\mathbf{p}}_{\mathbf{e}_i}, \vec{\mathbf{p}}_{\mathbf{h}_i})$ au référentiel fixe $(\vec{\mathbf{q}}_{\mathbf{e}}, \vec{\mathbf{q}}_{\mathbf{h}})$ que nous imposons pour chacune des images. Ce référentiel fixe peut correspondre à celui de l'une des images ou au référentiel de la vue cyclopéenne. Pour simplifier les notations, nous considérons maintenant le "redressement" de l'une des images.

Calcul des composantes de la transformation $\mathbf{T}_{\mathbf{p}\mathbf{q}}$: Calculons donc la transformation affine $\mathbf{T}_{\mathbf{p}\mathbf{q}}$ permettant de passer du référentiel $(\vec{\mathbf{p}}_{\mathbf{e}}, \vec{\mathbf{p}}_{\mathbf{h}})$ au référentiel fixe $(\vec{\mathbf{q}}_{\mathbf{e}}, \vec{\mathbf{q}}_{\mathbf{h}})$. Le référentiel $(\vec{\mathbf{p}}_{\mathbf{e}}, \vec{\mathbf{p}}_{\mathbf{h}})$ est défini au point (x, y) de l'image \mathbf{I} et le référentiel $(\vec{\mathbf{q}}_{\mathbf{e}}, \vec{\mathbf{q}}_{\mathbf{h}})$ est défini au point (u, v) lui correspondant dans la nouvelle image \mathbf{I}' . Nous définissons cette transformation comme suit :

$$\begin{aligned} \mathbf{T}_{\mathbf{p}\mathbf{q}} & : \quad \mathbf{I} \longrightarrow \mathbf{I}' \\ \vec{\mathbf{p}}_{\mathbf{e}} & \longrightarrow \vec{\mathbf{q}}_{\mathbf{e}} = \mathbf{T}_{\mathbf{p}\mathbf{q}}(\vec{\mathbf{p}}_{\mathbf{e}}) \\ \vec{\mathbf{p}}_{\mathbf{h}} & \longrightarrow \vec{\mathbf{q}}_{\mathbf{h}} = \mathbf{T}_{\mathbf{p}\mathbf{q}}(\vec{\mathbf{p}}_{\mathbf{h}}) \end{aligned}$$

que l'on peut réécrire sous la forme développée suivante :

$$\begin{aligned} \begin{pmatrix} x_{\mathbf{q}_e}(u, v) \\ y_{\mathbf{q}_e}(u, v) \end{pmatrix} &= \begin{pmatrix} \mathbf{T}\mathbf{a}(x, y) & \mathbf{T}\mathbf{b}(x, y) \\ \mathbf{T}\mathbf{c}(x, y) & \mathbf{T}\mathbf{d}(x, y) \end{pmatrix} \begin{pmatrix} x_{\mathbf{p}_e}(x, y) \\ y_{\mathbf{p}_e}(x, y) \end{pmatrix} \\ \begin{pmatrix} x_{\mathbf{q}_h}(u, v) \\ y_{\mathbf{q}_h}(u, v) \end{pmatrix} &= \begin{pmatrix} \mathbf{T}\mathbf{a}(x, y) & \mathbf{T}\mathbf{b}(x, y) \\ \mathbf{T}\mathbf{c}(x, y) & \mathbf{T}\mathbf{d}(x, y) \end{pmatrix} \begin{pmatrix} x_{\mathbf{p}_h}(x, y) \\ y_{\mathbf{p}_h}(x, y) \end{pmatrix} \end{aligned}$$

La transformation $\mathbf{T}_{\mathbf{p}\mathbf{q}}$ cherchée a quatre paramètres. Si on connaît les valeurs des quatre vecteurs $\vec{\mathbf{p}}_{\mathbf{e}}$, $\vec{\mathbf{p}}_{\mathbf{h}}$, $\vec{\mathbf{q}}_{\mathbf{e}}$ et $\vec{\mathbf{q}}_{\mathbf{h}}$ en tout point, on dispose alors d'un système de quatre équations permettant de déterminer toutes les composantes de la transformation $\mathbf{T}_{\mathbf{p}\mathbf{q}}$ à appliquer au point (x, y) . Connaissant les fonctions des composantes des champs 2D $\vec{\mathbf{p}}_{\mathbf{e}}$, $\vec{\mathbf{p}}_{\mathbf{h}}$, $\vec{\mathbf{q}}_{\mathbf{e}}$ et $\vec{\mathbf{q}}_{\mathbf{h}}$ en tout point (x, y) de l'image I , on peut alors déterminer les fonctions des composantes de la transformation étudiée $\mathbf{T}_{\mathbf{p}\mathbf{q}}$:

$$\mathbf{T}\mathbf{a} = \frac{y_{\mathbf{p}_h} x_{\mathbf{q}_e} - y_{\mathbf{p}_e} x_{\mathbf{q}_h}}{x_{\mathbf{p}_e} y_{\mathbf{p}_h} - y_{\mathbf{p}_e} x_{\mathbf{p}_h}} \quad \mathbf{T}\mathbf{b} = \frac{x_{\mathbf{p}_e} x_{\mathbf{q}_h} - x_{\mathbf{p}_h} x_{\mathbf{q}_e}}{x_{\mathbf{p}_e} y_{\mathbf{p}_h} - y_{\mathbf{p}_e} x_{\mathbf{p}_h}} \quad (4.13)$$

$$\mathbf{T}\mathbf{c} = \frac{y_{\mathbf{p}_h} y_{\mathbf{q}_e} - y_{\mathbf{p}_e} y_{\mathbf{q}_h}}{x_{\mathbf{p}_e} y_{\mathbf{p}_h} - y_{\mathbf{p}_e} x_{\mathbf{p}_h}} \quad \mathbf{T}\mathbf{d} = \frac{x_{\mathbf{p}_e} y_{\mathbf{q}_h} - x_{\mathbf{p}_h} y_{\mathbf{q}_e}}{x_{\mathbf{p}_e} y_{\mathbf{p}_h} - y_{\mathbf{p}_e} x_{\mathbf{p}_h}} \quad (4.14)$$

Chaque composante des champs 2D $\vec{\mathbf{p}}_{\mathbf{e}}$ et $\vec{\mathbf{p}}_{\mathbf{h}}$ et chaque composante de la transformation $\mathbf{T}_{\mathbf{p}\mathbf{q}}$ sont des fonctions dépendant de x et de y .

Recherche d'une fonction solution $U(x, y)$: La transformation $\mathbf{T}_{\mathbf{p}\mathbf{q}}$ associe aussi les surfaces de côtés $dx dy$ de l'image de départ \mathbf{I} aux surfaces $du dv$ qui lui correspondent dans la nouvelle image \mathbf{I}' . Nous pouvons donc écrire l'équation suivante :

$$\begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} \mathbf{T}\mathbf{a}(x, y) & \mathbf{T}\mathbf{b}(x, y) \\ \mathbf{T}\mathbf{c}(x, y) & \mathbf{T}\mathbf{d}(x, y) \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} \quad (4.15)$$

où $dx dy$ représente la projection d'un élément de surface 3D dans l'image \mathbf{I} et $du dv$ cette même surface projetée dans l'image \mathbf{I}' . Si on connaît les fonctions $\mathbf{T}\mathbf{a}$, $\mathbf{T}\mathbf{b}$, $\mathbf{T}\mathbf{c}$ et $\mathbf{T}\mathbf{d}$, on peut réécrire la transformation $\mathbf{T}_{\mathbf{pq}}$ par le système différentiel suivant :

$$\begin{aligned} du &= \mathbf{T}\mathbf{a}(x,y) dx + \mathbf{T}\mathbf{b}(x,y) dy \\ dv &= \mathbf{T}\mathbf{c}(x,y) dx + \mathbf{T}\mathbf{d}(x,y) dy \end{aligned} \quad (4.16)$$

On souhaite aussi que la transformation préserve la géométrie épipolaire dans les nouvelles images. On recherche donc une transformation $\mathbf{T}_{\mathbf{pq}}$ qui opère le long des lignes images comme dans la figure 4.10. On choisit d'imposer la contrainte $dv = dy$. Cette contrainte se répercute uniquement dans la seconde équation et nous donne : $\mathbf{T}\mathbf{c}(x,y) = 0$ et $\mathbf{T}\mathbf{d}(x,y) = 1$. On remarque que l'on peut obtenir le même résultat en imposant $\mathbf{y}_{\mathbf{qe}} = \mathbf{y}_{\mathbf{pe}}$ et $\mathbf{y}_{\mathbf{qh}} = \mathbf{y}_{\mathbf{ph}}$ dans les équations 4.14.

On cherche maintenant une fonction $\mathbf{U}(x,y)$, solution du système différentiel 4.16. Puisque $\mathbf{T}\mathbf{c}(x,y) = 0$ et $\mathbf{T}\mathbf{d}(x,y) = 1$, ce problème équivaut à résoudre la première forme différentielle du système. On cherche une fonction $\mathbf{U}(x,y)$ telle que :

$$\begin{aligned} \frac{\partial \mathbf{U}(x,y)}{\partial x} &= \mathbf{T}\mathbf{a}(x,y) \\ \frac{\partial \mathbf{U}(x,y)}{\partial y} &= \mathbf{T}\mathbf{b}(x,y) \end{aligned}$$

avec

$$\frac{\partial \mathbf{T}\mathbf{a}(x,y)}{\partial y} - \frac{\partial \mathbf{T}\mathbf{b}(x,y)}{\partial x} = 0 \quad (4.17)$$

Si la condition 4.17 est vérifiée, alors il existe une fonction $\mathbf{U}(x,y)$ solution qui sera une intégrale première de la forme différentielle 4.16. Si l'égalité de Schwarz 4.17 n'est pas vérifiée, la forme différentielle n'est pas intégrable et il n'existe pas de fonction $\mathbf{U}(x,y)$ solution. Dans le premier cas, la solution finale prendra la forme :

$$\begin{cases} u = \mathbf{U}(x,y) \\ v = y \end{cases} \quad (4.18)$$

Solutions pour un cas particulier du capteur sténopé

On cherche une fonction solution pour le modèle de projection sténopé. Nous considérons le cas où tous les points de la scène sont sur un plan à une distance constante Z_c . L'étude de ce cas particulier permet de faire le lien avec les effets de rotation obtenus dans l'étude du capteur linéaire en section 4.3.1. Les champs \vec{q}_e et \vec{q}_h sont constants en tout point des images pour l'orientation de façade θ . On cherche les composantes des deux champs de vecteurs 2D \vec{p}_e et \vec{p}_h . La scène 3D étant plane avec ce modèle de capteur, nous obtenons les champs \vec{p}_e et \vec{p}_h qui sont déterminés comme suit :

$$\begin{cases} \vec{\mathbf{p}}_e = \begin{pmatrix} \mathbf{x}_{\mathbf{pe}}(x,y) \\ \mathbf{y}_{\mathbf{pe}}(x,y) \end{pmatrix} = \begin{pmatrix} \frac{x}{Z_c} \\ \frac{y}{Z_c} \end{pmatrix} \\ \vec{\mathbf{p}}_h = \begin{pmatrix} \mathbf{x}_{\mathbf{ph}}(x,y) \\ \mathbf{y}_{\mathbf{ph}}(x,y) \end{pmatrix} = \begin{pmatrix} \frac{X_h}{Z_c} \\ \frac{Y_h}{Z_c} \end{pmatrix} = \begin{pmatrix} x_{h1} \\ y_{h1} \end{pmatrix} \end{cases} \quad (4.19)$$

avec $X_h = \cos \theta$ et $Y_h = \sin \theta$.

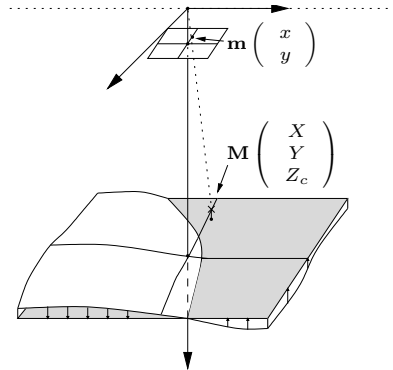


FIG. 4.11 – Présentation du cas particulier étudié. On utilise un modèle de projection sténopé, la scène est approximée par un plan.

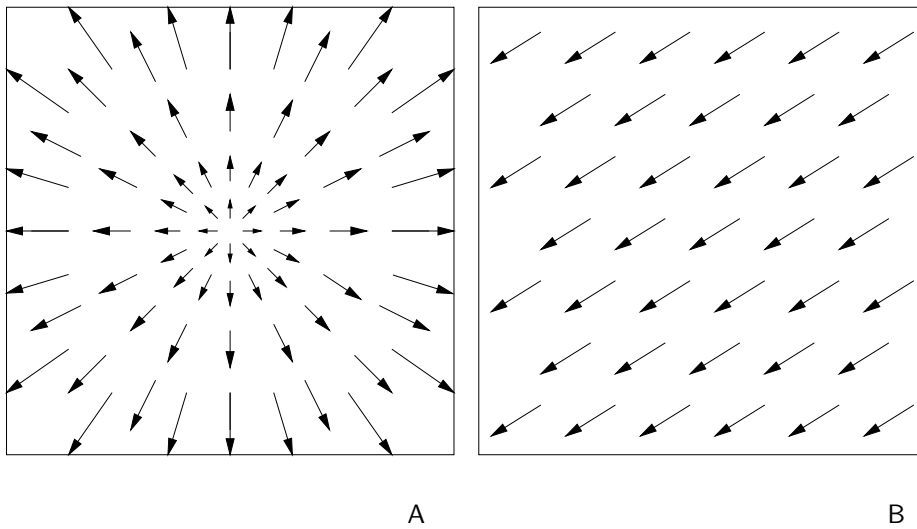


FIG. 4.12 – Champs de vecteurs 2D représentant les orientations des façades obtenues dans les images avec un capteur sténopé et une scène à Z constant. La figure A présente le champ de vecteurs 2D d'élévation \vec{p}_e (projection du champ 3d \vec{E}). La figure B présente le champ de vecteurs 2D d'orientation \vec{p}_h (projection du champ 3d \vec{H}).

Ces deux champs 2D sont présentés dans la figure 4.12. Les vecteurs du premier champ, $\vec{\mathbf{p}}_e$ dépendent des coordonnées images x et y tandis que les vecteurs du second champ $\vec{\mathbf{p}}_h$ sont constants. On obtient les coefficients de la forme différentielle du système que nous voulons résoudre comme suit :

$$\mathbf{Ta}(x,y) = \frac{Z_c y_{h1} x_{q_e} - y \times x_{q_h}}{x \times y_{h1} - y \times x_{h1}} \quad (4.20)$$

$$\mathbf{Tb}(x,y) = \frac{x \times x_{q_h} - Z_c y_{h1} x_{q_e}}{x \times y_{h1} - y \times x_{h1}} \quad (4.21)$$

Cas où $x_{q_h} = 0$: On choisit de traiter tout d'abord le cas où $\mathbf{x}_{q_h} = 0$ et $\mathbf{x}_{q_e} \neq 0$. Ce cas correspond à construire une image dans laquelle seule l'élévation varie le long des lignes épipolaires. Les façades deviennent donc perpendiculaires aux lignes des images. Les variations le long de l'orientation des façades 3D n'entraîneront pas de déplacement le long des lignes (puisque $\mathbf{x}_{q_h} = 0$). La forme différentielle étant exacte, nous l'intégrons directement pour obtenir la solution suivante :

$$\mathbf{U}(x,y) = x_{q_e} Z_c \ln(-x \times y_{h1} + y \times x_{h1}) \quad (4.22)$$

On remarque que la solution que nous obtenons dans 4.22 reprend la forme de la solution obtenue avec le capteur linéaire (équation 4.8). Nous avons choisi une orientation initiale des façades rendant le vecteur d'orientation $\vec{p}_h = (x_{h1}, y_{h1})$ perpendiculaire aux lignes épipolaires, ce qui est équivalent à $x_{h1} = 0$ et $y_{h1} \neq 0$. La figure 4.13 présente les résultats obtenus en utilisant cette solution.

Dans cette figure 4.13, nous avons choisi de traiter les surfaces verticales dont l'orientation est de 125° dans le cercle trigonométrique. Les images A et B présentent les images initiales d'un bâtiment et les images C et D présentent les images reconstruites en appliquant la transformation 4.22. Dans les nouvelles images, les façades traitées sont perpendiculaires aux lignes images et ont la même surface. Elles peuvent être appariées simplement. Les importantes déformations des images entraînées par la transformation rendent toutefois ces images peu exploitables.

Autres cas : Dans tous les autres cas, l'égalité de Schwarz 4.17 n'est pas vérifiée. La forme différentielle n'est pas intégrable, il n'existe aucune fonction solution pour les cas où $x_{q_h} \neq 0$ et $x_{q_e} \neq 0$.

Remarque : On peut remarquer dans la figure 4.12 A que le champ de vecteur 2D \vec{p}_e représentant l'élévation possède un point de fuite au point principal de l'image. L'étude de cette région entraînerait des problèmes numériques. Les façades n'étant pas visibles dans cette région, nous ne nous y intéressons pas.

Linéarisation du capteur sténopé

La fonction $\mathbf{Z}(x,y)$ est maintenant inconnue. On traite le capteur sténopé en linéarisant la projection. La surface du sol variant continûment et dans des proportions infimes, on suppose que les champs d'élévation et d'orientation sont constants dans le voisinage d'un bâtiment. Pour projeter la surface unitaire définie au point (x_0, y_0) de l'image (projeté du point 3D (X_0, Y_0, Z_0)) il nous faut connaître le jacobien de la projection au point (X_0, Y_0, Z_0) de la scène.

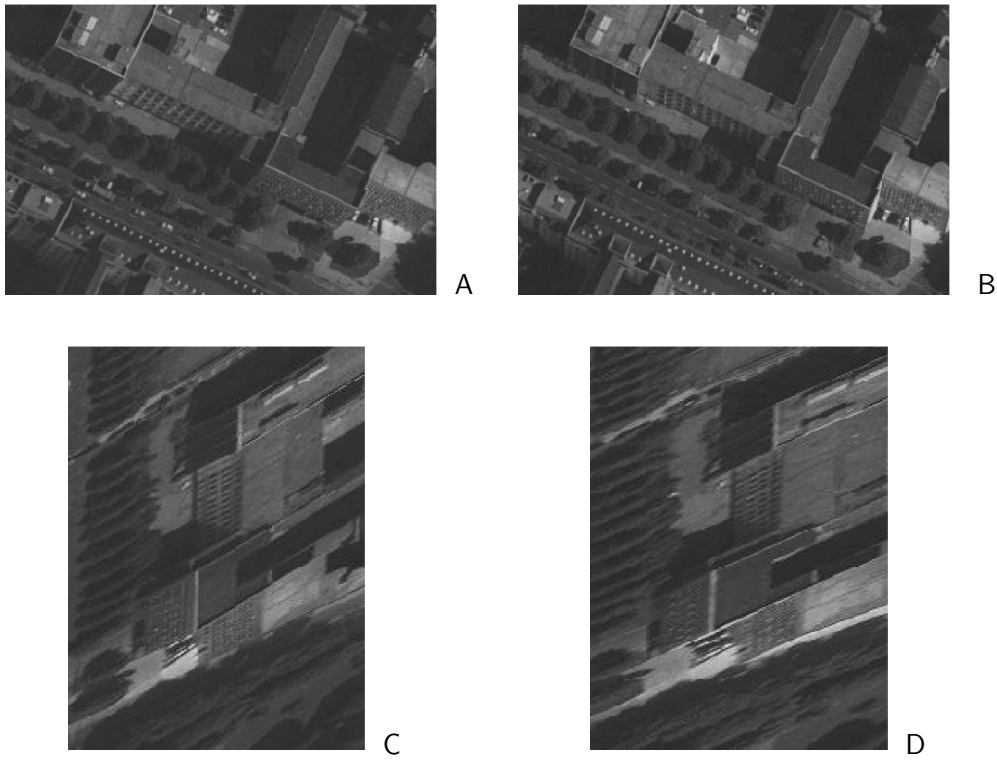


FIG. 4.13 – Résultats obtenus dans le cas où $x_{qh} = 0$ et $x_{qe} \neq 0$. A et B sont les images originales et C et D sont les images obtenues après transformation. On peut voir que les façades sont "réorientées" perpendiculairement aux droites épipolaires.

Considérons la projection \mathbf{Pr} définie comme suit :

$$(x,y) = \mathbf{Pr}(X,Y,Z) = (\mathbf{P}\mathbf{x}(X,Y,Z), \mathbf{P}\mathbf{y}(X,Y,Z)) \quad (4.23)$$

En choisissant un voisinage suffisamment petit (le bâtiment), on linéarise la projection \mathbf{Pr} au point (X_0, Y_0, Z_0) :

$$\begin{aligned} (x,y) = & P(X_0, Y_0, Z_0) + (X - X_0) \times \frac{\partial P}{\partial X} \Big|_{X_0, Y_0, Z_0} \\ & + (Y - Y_0) \times \frac{\partial P}{\partial Y} \Big|_{X_0, Y_0, Z_0} \\ & + (Z - Z_0) \times \frac{\partial P}{\partial Z} \Big|_{X_0, Y_0, Z_0} \end{aligned}$$

ce qui est équivalent au système suivant:

$$(x,y) = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} + (x_0, y_0) \quad (4.24)$$

avec :

$$(x_0, y_0) = Pr(X_0, Y_0, Z_0) \quad (4.25)$$

$$\begin{aligned}
a &= \left. \frac{\partial Px}{\partial X} \right|_{X_0, Y_0, Z_0} & b &= \left. \frac{\partial Px}{\partial Y} \right|_{X_0, Y_0, Z_0} \\
c &= \left. \frac{\partial Px}{\partial Z} \right|_{X_0, Y_0, Z_0} & d &= \left. \frac{\partial Py}{\partial X} \right|_{X_0, Y_0, Z_0} \\
e &= \left. \frac{\partial Py}{\partial Y} \right|_{X_0, Y_0, Z_0} & f &= \left. \frac{\partial Py}{\partial Z} \right|_{X_0, Y_0, Z_0}
\end{aligned}$$

Nous allons utiliser ces équations pour estimer la fonction de projection au point (x_0, y_0) . Dans notre cas, nous nous intéressons à la projection de la surface unitaire verticale orientée définie par les deux vecteurs 3D d'élévation \vec{e} et d'orientation \vec{h} , seul le jacobien de la transformation nous est utile. Les valeurs des champs 2D d'élévation \vec{p}_e et d'orientation \vec{p}_h constant sur le voisinage obtenus sont les suivants :

$$\vec{p}_e(x_0, y_0) = \begin{pmatrix} x_{p_e}(x_0, y_0) \\ y_{p_e}(x_0, y_0) \end{pmatrix} = \begin{pmatrix} c \\ f \end{pmatrix} \quad (4.26)$$

$$\begin{aligned}
\vec{p}_h(x_0, y_0) &= \begin{pmatrix} x_{p_h}(x_0, y_0) \\ y_{p_h}(x_0, y_0) \end{pmatrix} \\
&= \begin{pmatrix} a & b \\ e & f \end{pmatrix} \begin{pmatrix} X_h \\ Y_h \end{pmatrix}
\end{aligned} \quad (4.27)$$

Nous pouvons maintenant calculer les valeurs des coefficients de la forme différentielle Ta et Tb au point (x_0, y_0) et obtenir la solution $\mathbf{U}(x, y)$ équivalente à la transformation $T_{pq}(x_0, y_0)$ valable dans le voisinage local :

$$\mathbf{U}(x, y) = Ta(x_0, y_0)x + Tb(x_0, y_0)y \quad (4.28)$$

avec

$$\begin{aligned}
Ta(x_0, y_0) &= \frac{(dX_h + eY_h)x_{q_e} - f x_{q_h}}{(cd - fa)X_h + (ce - fb)Y_h} \\
Tb(x_0, y_0) &= \frac{c x_{q_h} - (aX_h + bY_h)x_{q_e}}{(cd - fa)X_h + (ce - fb)Y_h}
\end{aligned} \quad (4.29)$$

La figure 4.14 présente les résultats obtenus avec la transformation 4.29. Le choix des paramètres joue un rôle important dans la qualité des résultats. On rappelle que X_h et Y_h définissent l'orientation des façades étudiées. Nous les choisissons pour représenter les façades avec une orientation de 155° . Les composantes x_{q_e} et x_{q_h} sont choisies de façon à limiter les importantes déformations des images. Dans les nouvelles images, les façades traitées ont la même structure, et peuvent être appariées simplement.

Remarque Le fait de linéariser le problème rend notre méthode de transformation des façades analogue à la rectification des images par rapport à un plan 3D vertical, lorsque ce plan est orienté comme la façade et passe par le $M(X_0, Y_0, Z_0)$ de la scène. C'est ce que propose *P. Debevec* dans FACADE pour affiner la surface de ses bâtiments [DTM96, Deb96]. Il rééchantillonne les images des façades du bâtiment par rapport aux facettes du modèle grossier pour pouvoir mieux appairer ces surfaces. Cependant, les façades ne sont pas autant déformées que dans notre cas, puisqu'il utilise des photographies terrestres.

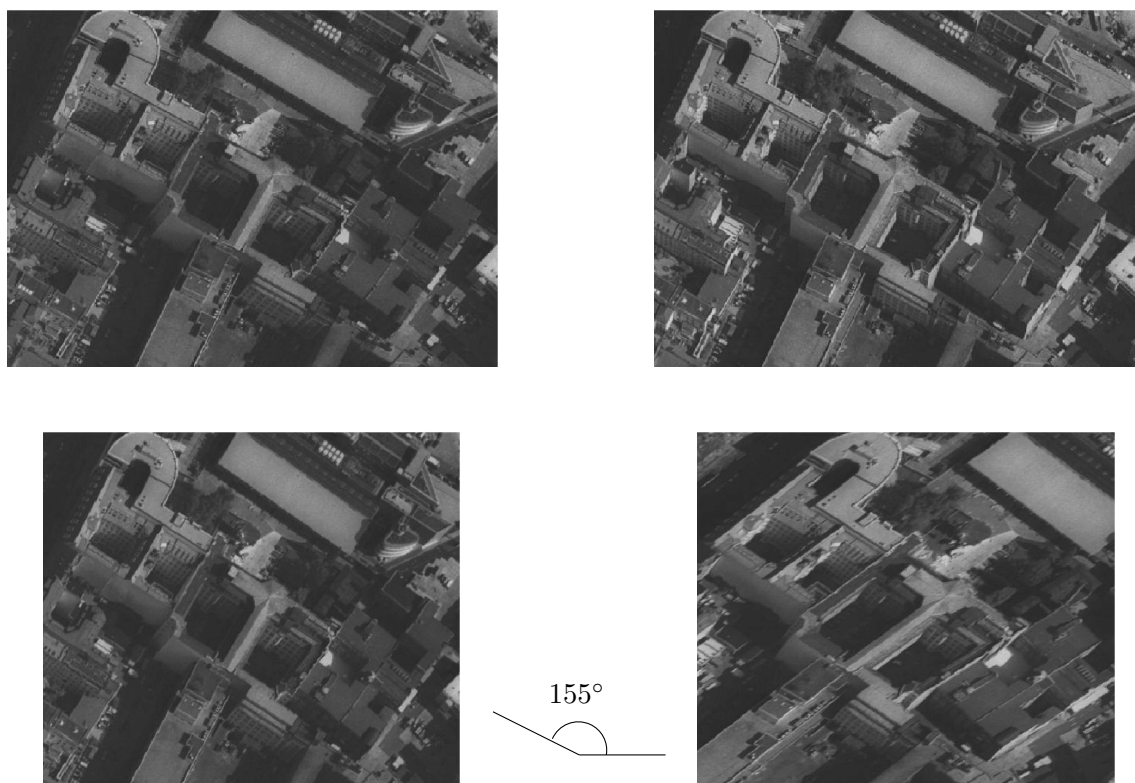


FIG. 4.14 – Résultats de la transformation avec le capteur sténopé par linéarisation de la fonction de projection. A et B sont les deux images initiales. C et D sont les images reconstruites en imposant $x_{q_e} = 1$ et $x_{q_h} = 2$. Ce choix des coefficients permet de limiter les déformations introduites par le rééchantillonnage des images. Les façades étudiées sont celles dont l'orientation est de 155° par rapport aux lignes des images. Dans les images initiales, ces façades sont perçues différemment et ne peuvent être appariées. Dans les images reconstruites, elles peuvent être appariées plus facilement.

4.4 Reconstruction des points des façades

La mise en correspondance des façades est réalisée par un algorithme de corrélation multi-résolutions. Nous utilisons deux prises de vues rectifiées sur un bâtiment. L'orientation des façades à traiter est mesurée manuellement. Nous verrons par la suite comment se passer de cette connaissance. Nous nous intéressons dans cette section aux points reconstruits sur les façades. Nous étudions l'amélioration apportée ainsi que la sensibilité de la reconstruction vis-à-vis du seul paramètre nécessaire à la construction des images : l'orientation des façades.

Choix du référentiel (\vec{q}_e, \vec{q}_h) Il est nécessaire d'éviter les multiples rééchantillonnages des images pour ne pas perdre trop d'informations. Nous avons donc choisi de construire les nouvelles images non pas avec les images rectifiées, mais avec les images initiales. Nous appliquons successivement les transformations de rectification et de redressement des façades avant la construction des images. De même, pour limiter des déformations trop importantes, le référentiel final (\vec{q}_e, \vec{q}_h) choisi est celui de l'une des images rectifiées. L'image choisie pour référence est celle qui possède la plus grande surface de façade. On peut estimer la taille de

cette surface directement à partir des vecteurs (\vec{p}_e, \vec{p}_h) déterminés pour chacune des images.

4.4.1 Résultats de la mise en correspondance

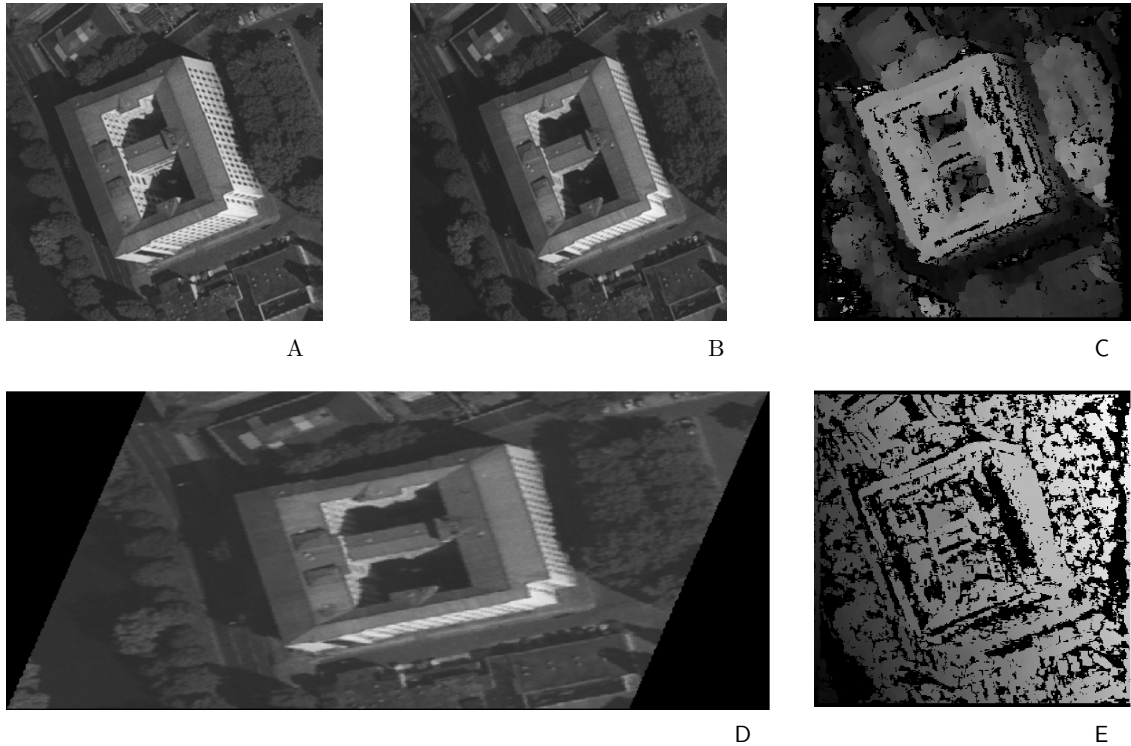


FIG. 4.15 – *Corrélation des façades : A et B sont les images rectifiées du bâtiment étudié. C est la carte de disparité, relative à l'image A obtenue par l'algorithme de mise en correspondance. Les toits et le sol sont correctement retrouvés, tandis que les disparités de la façade de droite du bâtiment ont une forme en "escalier" (au lieu d'une forme plane). D correspond à B traitée par la transformation des façades. La façade de droite dans D possède le même aspect visuel que dans l'image A. Dans la carte de disparité correspondante (E), la disparité au niveau de la façade est maintenant constante.*

Ces travaux ont fait l'objet de la communication [VD99]. La figure 4.15 nous présente les résultats de la mise en correspondance. Les images A, B et C présentent le processus de mise en correspondance avec les images rectifiées initiales. Les images A, D et E présentent les résultats du même processus après la transformation des façades. Dans le premier cas (C), le sol et le toit sont bien appariés mais pas la façade de droite. Les disparités correspondant à cette région de l'image ont une forme en "escalier" car les voisinages utilisés lors de la mise en correspondance sont trop différents. Dans le second cas (E), la surface de disparité qui correspond à la façade est constante. Après la transformation, la façade de la scène possède le même aspect visuel dans les deux images. Les voisinages utilisés dans le calcul des scores de corrélation se correspondent et permettent donc la mise en correspondance et la reconstruction des façades.

4.4.2 Cartes d'accumulation

Pour visualiser les points reconstruits, nous utilisons des cartes d'accumulation dans lesquelles nous projetons et comptabilisons tous les points reconstruits. La couleur affectée aux pixels dépend du nombre de points comptabilisé : la couleur est blanche dans le cas où aucun point n'est comptabilisé et la couleur devient de plus en plus sombre lorsque le nombre de points projetés augmente. Cette carte est référencée en coordonnées terrain. Le choix de la résolution des cellules est important. Elle est de l'ordre de la résolution des images. Les façades étant verticales, les points reconstruits sur ces surfaces vont être projetés dans le même accumulateur. Ils correspondent aux pics d'accumulation des points reconstruits.

4.4.3 Étude de la localisation

Nous utilisons les cartes d'accumulation pour visualiser l'amélioration de la localisation obtenue grâce à la transformation des façades. Nous utilisons deux cartes pour permettre la comparaison. Dans la première carte, nous projetons les points issus de la mise en correspondance des images rectifiées. Nous nommons cette carte la **carte d'accumulation horizontale** puisque les points reconstruits résultent d'un processus d'appariement privilégiant les surfaces horizontales. Dans la seconde carte, la **carte d'accumulation d'orientation -60°** , nous projetons les points issus de la mise en correspondance des nouvelles images obtenues après la transformation des façades pour une orientation de -60° . Les résultats sont présentés dans la figure 4.16. Nous reconstruisons la totalité des points appariés, sans aucun filtrage, les projetons dans des cartes d'accumulation et observons la dispersion de ces points dans la direction perpendiculaire à l'orientation des façades. Les pics d'accumulation correspondent aux façades. La dispersion des points en ces lieux est le critère de qualité choisi. Plus la dispersion est importante, plus la localisation est mauvaise et inversement. Pour visualiser cette dispersion, nous utilisons une carte d'accumulation de 10 centimètres de résolution (les images ont une résolution de $37,5cm$), projetons la totalité des points reconstruits et lisons l'image avec une gaussienne de sigma 20 centimètres pour visualiser la forme des accumulations de points. La dispersion des points reconstruits autour des façades est nettement moins importante en utilisant les images transformées. On voit dans l'image B un grand nombre de points qui s'accumulent le long de la façade de droite du bâtiment. En comparant les deux régions zoomées, on voit que la localisation des points de la façade étudiée est bien meilleure si l'on prend en compte les déformations des surfaces verticales dues à la projection.

4.4.4 Sensibilité de la reconstruction

Nous nous intéressons maintenant à la sensibilité des données reconstruites par rapport à l'orientation. Pour cela, nous allons reconstruire ces mêmes cartes d'accumulation en faisant varier l'orientation θ (X_h et Y_h). Les résultats sont présentés dans la figure 4.17. La façade réelle que nous essayons de reconstruire possède une orientation de -60° . Nous faisons varier l'orientation de -30° à -90° , construisons les nouvelles images et appliquons le processus de mise en correspondance avant de produire les cartes d'accumulation correspondantes. La carte localisant le mieux la façade du bâtiment est bien celle correspondant à l'orientation de -60° . Toutes les cartes dont l'orientation est comprise entre -90° et -40° localisent mieux cette façade que la carte d'accumulation horizontale. On aperçoit un bruit régulier dans la carte dont l'orientation est de -30° . Ceci est dû au fait que le vecteur d'orientation $\vec{\mathbf{p}}_{h(x_0,y_0)}$ utilisé est proche du vecteur d'élévation $\vec{\mathbf{p}}_{e(x_0,y_0)}$ qualifiant la direction verticale.

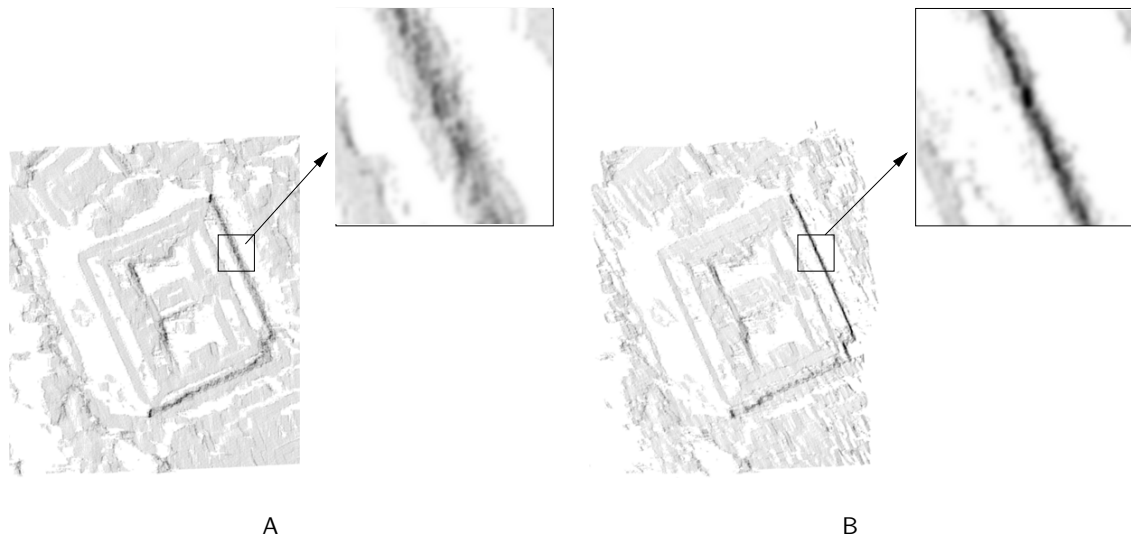


FIG. 4.16 – Localisation des points reconstruits : *A* est la carte d'accumulation dans laquelle sont projetés tous les points reconstruits par le processus de mise en correspondance utilisant les images rectifiées. *B* est la carte d'accumulation d'orientation -60° . Les points projetés dans cette carte proviennent du processus de mise en correspondance des images après transformation des façades. En observant les zones extraites de ces deux cartes, on constate que la dispersion des points reconstruits est moins importante dans la direction perpendiculaire à l'orientation de la façade. De plus, le nombre de points reconstruits sur les façades augmente.

Nous n'appliquons pas la transformation des façades dans ce cas dégénéré où le plan vertical correspondant à la façade étudiée passe par le centre optique. Les façades ne sont pas visibles. Pour conclure, la sensibilité de la reconstruction est faible vis-à-vis de l'orientation utilisée pour les façades tant que $\vec{\mathbf{p}}_{\mathbf{h}(x_0, y_0)} \neq \vec{\mathbf{p}}_{\mathbf{e}(x_0, y_0)}$.

4.5 Un algorithme de reconstruction automatique de façades

Nous allons maintenant présenter une stratégie permettant la reconstruction automatique des façades. Ces travaux ont fait l'objet de la publication [VD00]. Nous avons vu comment construire de nouvelles images permettant d'apparier les façades. Il est nécessaire de connaître l'orientation de la façade de la scène pour pouvoir la reconstruire. Puisque la reconstruction des points des façades est très peu sensible vis-à-vis de l'orientation utilisée, la stratégie que nous proposons consiste à parcourir l'espace discret des orientations possibles et de ne garder que les points 3D reconstruits qui appartiennent aux façades. Cette technique peut être comparée à des techniques telles que le *space-sweep algorithm* de *R. Collins* [Col96] ou de *S. Teller* [CT98]. Le *space-sweep algorithm* discrétise l'espace de recherche (un plan balayant l'espace 3D) et retrouve les primitives 3D cherchées par leurs projections qui s'accumulent dans certains lieux de la scène. Nous verrons dans la suite que l'approche présentée ici utilise aussi une technique de filtrage basée sur le vote pour retrouver les points des façades. Ce filtrage utilise également le contexte pour extraire l'information utile.

Associé à cette stratégie de reconstruction automatique des façades, nous proposons un outil permettant de corriger les limites des bâtiments dans un MNE. Nous avons vu dans la

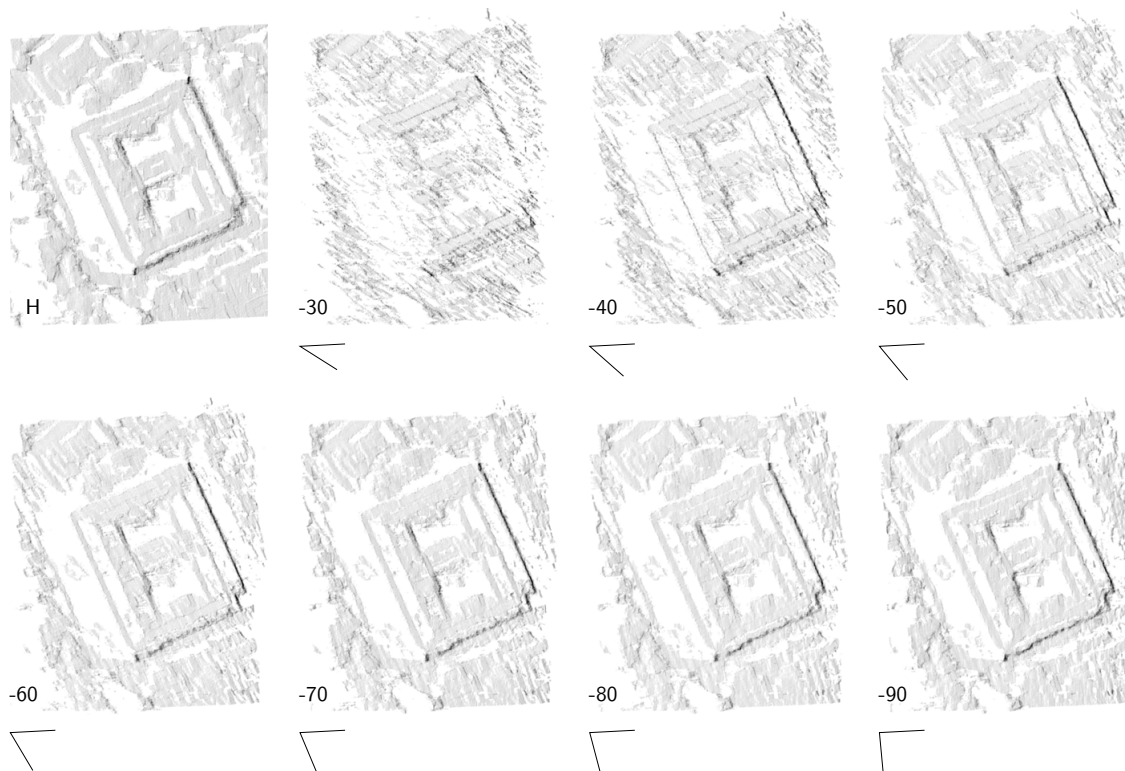


FIG. 4.17 – *Sensibilité des points reconstruits. L'orientation réelle de la façade étudiée est de -60° . Nous faisons varier l'orientation de -30° à -90° et construisons les cartes d'accumulation correspondantes. Toutes les cartes dont l'orientation est comprise entre -40° et -90° possèdent une bonne localisation de cette façade. La carte dont l'orientation est de -30° présente des défauts liés à un cas dégénéré de la transformation des façades.*

section 4.1 que les bords des bâtiments dans un MNE obtenu par corrélation sont arrondis et que les limites ne sont pas franches et linéaires. La surface reconstruite déborde des limites réelles du bâtiment, surtout dans les ombres. Ce phénomène est lié directement à l'algorithme que nous utilisons pour construire le MNE, la corrélation, qui ne peut reconstruire que les surfaces horizontales. L'idée est d'utiliser les points reconstruits sur les façades pour mieux délimiter le bâtiment.

Nous avons vu dans 3.6 la stratégie générale adoptée pour la reconstruction de bâtiments. L'algorithme proposé ici participe à la première étape de cette stratégie générale, la construction d'un MNE dense et fiable. Les erreurs présentes dans ce MNE auront un impact important sur les résultats des étapes de segmentation et de vectorisation. L'algorithme que nous proposons intervient avant ces deux étapes, son rôle est la correction des limites des bâtiments.

4.5.1 Présentation générale de l'algorithme

Nous supposons tout d'abord connue l'orientation de la façade de la scène que nous souhaitons reconstruire (figure 4.9). Les différentes étapes de l'algorithme proposé sont les

suivantes :

1. Choisir deux images de la façade: Parce que les façades sont perçues différemment dans toutes les images, nous avons le besoin de choisir deux images selon nos critères (section 4.5.3).
2. Construire les images des façades: Nous avons déjà vu l'utilité de cette étape. Ces nouvelles images sont obtenues par application de la transformation des façades. Les façades ont ensuite le même aspect visuel dans ces images (section 4.5.4).
3. Mise en correspondance par corrélation et reconstruction : Nous appliquons un processus de mise en correspondance des images basé sur la corrélation pour apparier les points des images et reconstruire les points 3D correspondant aux façades (section 4.5.5).
4. Filtrer les points 3D incorrects:(section 4.5.6).
5. Corriger le MNE : Nous utilisons les points 3D reconstruits (pour l'orientation définie de la façade) pour corriger le MNE. Nous creusons simplement dans le MNE aux lieux où nous avons reconstruit les points des façades (section 4.5.6).

Pour pouvoir traiter toutes les façades des bâtiments, nous appliquons cet algorithme pour toutes les orientations possibles de façades de la scène. Ceci a l'avantage de ne nécessiter aucune connaissance sur la forme des bâtiments. La reconstruction étant peu sensible, un sous-échantillonnage de l'espace des orientations des façades de 20° est suffisant pour retrouver toutes les façades (section 4.4.4). Cet algorithme ne dépend pas du nombre d'images disponibles, de l'orientation des façades, ou de la forme des bâtiments de la scène.

4.5.2 Détection des bâtiments

Pour garder un temps de calcul bas et satisfaire les conditions d'application locale de la transformation (due à la linéarisation de la projection), le processus entier (traitement de toutes les orientations) est seulement appliqué au voisinage de chaque bâtiment ou groupe de bâtiments adjacents extrait automatiquement du MNE, sur la région d'intérêt $ROI_{\text{bât}}^{2.5D}$. Pour cette détection, nous soustrayons le Modèle Numérique de Terrain (MNT qui correspond au sol) de la scène du MNE pour obtenir une carte des hauteurs, le Modèle Numérique des Hauteurs (MNH). Nous extrayons le MNE local correspondant à chaque blob (sur-sol) de la carte des hauteurs en utilisant un seuil de 3m. Nous conservons uniquement les bâtiments qui ont une hauteur et une taille suffisamment importantes. Nous appliquons ensuite l'algorithme de segmentation sur chaque zone d'intérêt correspondant à un bâtiment (la $ROI_{\text{bât}}^{2.5D}$).

Nous allons voir maintenant les détails de chacune des étapes de cet algorithme. Chaque étape travaille pour une orientation de façade donnée. S'il existe une vraie façade possédant cette orientation dans la zone étudiée, l'algorithme extrait les points des façades, filtre les données incorrectes et corrige le MNE. Dans le cas contraire, tous les points reconstruits sont incorrects et éliminés par le filtre. L'algorithme utilise cette méthode pour toutes les orientations de façade possibles, échantillonnées tous les 20° , et corrige le MNE incrémentalement.

4.5.3 Choix des images pour une orientation donnée

Dans un premier temps, nous devons sélectionner deux images, dans la liste des couples disponibles (section 3.2), dans lesquelles les façades sont le mieux visualisées. Nous avons testé deux critères: (1) les images dans lesquelles les surfaces projetées des façades ont la

taille maximum, (2) l'angle entre la direction normale à la surface du bâtiment et la direction du nadir (projection verticale du centre optique sur le sol) est minimum (la vue doit être dans l'axe de la façade).

Nous avons défini dans la section 4.3.2 deux vecteurs 3D unitaires, le vecteur d'élévation \vec{e} (orientation de la verticale) et le vecteur \vec{h} (orientation le long de la façade dans le plan horizontal), comme le présente la figure 4.9. Ces deux vecteurs sont définis au point 3D $M(X0, Y0, Z0)$ de la scène pris sur le sol et au centre de la $ROI_{\text{bât}}^{2.5D}$ correspondant au bâtiment étudié. Ces deux vecteurs correspondent à une façade unitaire de la scène. Nous calculons les deux vecteurs 2D \vec{p}_{ei} et \vec{p}_{hi} dans chaque image I_i qui correspondent respectivement aux projections de \vec{e} et \vec{h} . $\vec{p}_{ei} = (x_{ei}, y_{ei})$ représente l'orientation de la verticale au point m_i de l'image I_i et $\vec{p}_{hi} = (x_{hi}, y_{hi})$ représente l'orientation le long de la façade au même point m_i . Le calcul de ces vecteurs est effectué en linéarisant la projection comme dans la section 4.3.2.

Les deux critères présentés sont définis comme suit :

$$A_i = \|\vec{p}_{ei} \times \vec{p}_{hi}\|, \quad C_i = \frac{\vec{p}_{ei} \cdot \vec{p}_{hi}}{\|\vec{p}_{ei}\| \|\vec{p}_{hi}\|},$$

où \times est le produit vectoriel de deux vecteurs. La mesure A_i doit être maximum et C_i , qui correspond au cosinus de l'angle formé par les deux vecteurs \vec{p}_{ei} et \vec{p}_{hi} , doit être minimum (ces deux vecteurs sont orthogonaux quand le nadir est dans l'axe de la façade). Nous avons testé ces deux critères et leur combinaison pour la sélection du couple d'images. Les meilleurs résultats ont été obtenus en utilisant uniquement le critère A_i . Nous commençons notre sélection en choisissant l'image I_i parmi toutes les images disponibles qui possède la plus grande valeur A_i . Parmi tous les couples d'images qui contiennent cette image, nous sélectionnons ensuite l'image I_j qui possède la plus grande valeur A_j . Cette méthode nous permet d'obtenir un couple d'images adjacentes sur lequel nous pouvons appliquer la transformation des façades, il n'y a pas d'orientation qui entraîne de cas dégénéré comme nous avons pu le voir dans la section 4.4.4.

4.5.4 Construction des nouvelles images et des façades

Maintenant, nous voulons construire une paire d'images rectifiées dans lesquelles les façades correspondant à l'orientation donnée aient le même aspect visuel, ceci pour simplifier leur mise en correspondance. Nous ne disposons à ce stade que du MNE du bâtiment et des images initiales de la scène. La région d'intérêt $ROI_{\text{bât}}^{3D}$ dans chaque image initiale est définie comme la projection de l'enveloppe grossière tridimensionnelle du bâtiment prise dans le MNE. Nous appliquons ensuite successivement les transformations de rectification et de transformation des façades pour construire ces nouvelles images. Comme l'explique la section 4.4, le référentiel final choisi est celui de l'image qui possède la plus grande surface de façade.

4.5.5 Reconstruction des points des façades

Pour reconstruire les points des façades, nous utilisons le même algorithme de corrélation qui a permis de construire le MNE. Cette fois-ci, il permet d'obtenir une reconstruction dense des surfaces verticales comme le présentait la figure 4.15.

4.5.6 Filtrage des points 3D

Nous avons vu dans la figure 4.15 que nous sommes capables de reconstruire les façades, mais beaucoup de couples de points homologues dans la carte de disparité correspondent à des données incorrectes. Ils doivent donc être rejetés. L'étape de filtrage se décompose en trois filtres appliqués successivement : un filtre appliqué à chaque carte de disparité qui utilise le vecteur normal à la surface 3D, un filtre 2D utilisant une carte d'accumulation, et un filtre 3D qui prend en compte la forme de la surface.

Orientation de la surface

Nous commençons par calculer, pour chaque carte de disparité, le vecteur normal à la surface pour chaque point reconstruit, par l'estimation d'un plan dans le voisinage de ce point (le voisinage est considéré dans l'espace de la disparité). Lorsque l'orientation du vecteur normal diffère trop de l'orientation de la façade étudiée, le point 3D est rejeté (le seuil de ce filtre est de 15°). Le vecteur normal est ensuite reprojété dans le plan horizontal de l'espace 3D pour que la surface soit verticale. Dans la suite, on conserve pour chaque point reconstruit le vecteur normal à la surface locale qui lui est associé.

Utilisation d'une carte d'accumulation

Le second filtre utilise une carte d'accumulation qui est référencée dans les mêmes coordonnées cartographiques que celles du MNE (section 4.4.3). Les points reconstruits sont tous projetés dans cette carte, et les points 3D qui correspondent à la même façade doivent s'accumuler dans les mêmes cellules. Les points projetés dans des cellules dont le score d'accumulation est inférieur à 2 sont écartés.

Filtrage 3D

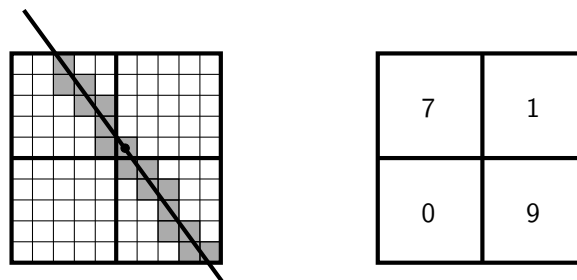


FIG. 4.18 – Filtrage 3D des points : Le tableau de gauche présente les voxels (en traits épais) et les sous-cellules (en traits fins). Le segment représente une hypothèse de façade définie par un point 3D reconstruit. Les cellules grises sont les cellules sélectionnées par l'algorithme de Bresenham. Le tableau de droite présente les valeurs attribuées à chaque voxel.

Le dernier filtre est dérivé des idées liées au groupement perceptuel. Une approche intéressante pour la reconstruction de surface à partir de données 3D éparses est décrite dans [TM98]. Notre filtre 3D est inspiré par cette approche. Il s'agit d'un système simple basé sur un filtrage par vote qui privilégie les surfaces verticales. Nous construisons un

tableau tridimensionnel en coordonnées cartographiques. La taille des voxels est de 1 mètre en X et Y et de 3 mètres en Z .

Chaque point 3D est associé au voxel qui lui correspond. Tous les points reconstruits vont voter dans leur voisinage des façades. Le voisinage des façades est défini par un morceau de plan rectangulaire, centré sur le point 3D et orienté par le vecteur normal associé au point (ce morceau rectangulaire est toujours vertical). Chaque point reconstruit va voter dans les voxels intersectés par ce voisinage. Ce voisinage 3D permet de voter pour la façade qui est définie par le point reconstruit. Le système de vote utilise un sous-échantillonnage de l'espace des voxels de pas 20cm , car la résolution du tableau 3D n'est pas suffisante pour calculer un vote représentatif : nous voulons qu'une façade qui intersecte un voxel en son coin possède un vote moins important qu'une façade qui passe au milieu de ce même voxel.

Pour voter le long de la façade, nous créons un segment horizontal d'une longueur de 6 mètres (la résolution du MNE est de 0.5 mètres), centré sur le point 3D et orienté par le vecteur associé, votant pour tous les voxels dans lesquelles il passe. Nous utilisons l'algorithme de *Bresenham* pour trouver toutes les sous-cellules des voxels qu'il intersecte. Le calcul du vote associé à ce segment est présenté dans la figure 4.18, le poids attribué au voxel est simplement le nombre des sous-cellules du voxel qui ont été marquées par l'algorithme de *Bresenham*. Nous répliquons ensuite le vote du segment aux voxels supérieurs et inférieurs du tableau 3D qui correspondent au voisinage des façades que nous avons défini (les deux voxels supérieurs et les deux voxels inférieurs). Nous conservons les points 3D pour lesquels le voxel correspondant possède au moins 3 points et un vote de poids supérieur à 300.

La figure 4.19 présente le résultat du filtrage des points 3D, après application de chacun des trois filtres présentés. La première colonne présente la carte d'accumulation obtenue après projection des points 3D filtrés selon l'orientation de la surface locale. La deuxième colonne présente la carte d'accumulation obtenue après projection des points 3D filtrés en utilisant ensuite le filtre d'accumulation dans une carte $2D\frac{1}{2}$. La dernière colonne présente la carte d'accumulation obtenue après projection des points restants après l'application du filtre 3D. Pour les orientations de 0° et de 220° (les deux premières lignes), le filtrage permet de ne récupérer que les points 3D appartenant aux façades. Il n'existe aucune façade dans la scène ayant pour orientation 220° , le filtre a supprimé presque tous les mauvais points pour cette orientation (troisième ligne). Il fonctionne donc correctement.

Correction du MNE

Pour corriger le MNE du bâtiment, nous utilisons le fait qu'aucun point 3D ne peut physiquement se situer entre un point reconstruit sur la façade et les centres optiques des vues utilisées pour reconstruire ce point. Nous calculons les rayons optiques correspondant à chaque point et *creusons* le MNE entre le point et les centres optiques comme le présente la figure 4.20. Cette méthode offre l'avantage de conserver une indépendance vis-à-vis d'un modèle de surface ou d'un modèle de bâtiment. Ce système peut utiliser plusieurs outils pour creuser dans le MNE : nous utilisons un prisme de diamètre 1,5 mètre avec une section verticale. La correction est faite incrémentalement après chaque orientation de façade traitée.

4.5.7 Résultats de reconstruction automatique

La figure 4.21 présente les cartes d'accumulation obtenues après avoir lancé le processus pour toutes les orientations. On peut voir que les façades du bâtiment sont extraites dans les cartes d'accumulation de 0° , 100° , 180° et 280° . En regardant les cartes des orientations

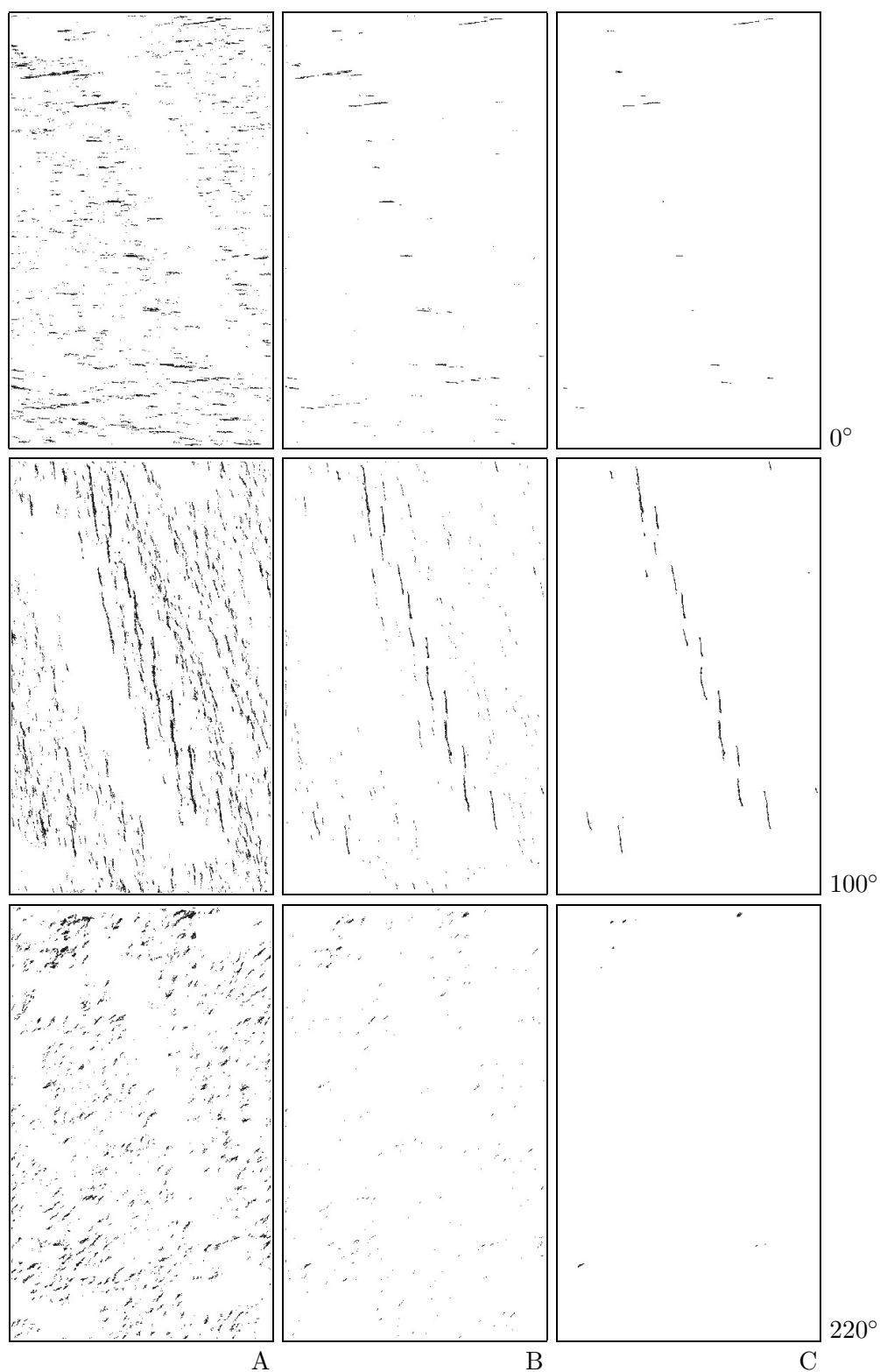


FIG. 4.19 – Résultat des trois filtres : Chaque ligne présente le filtrage pour une orientation donnée. Chaque colonne présente les points restants après chacun des trois filtres : A correspond au filtre sur l'orientation des vecteurs, B le filtre utilisant la carte d'accumulation et C le filtre 3D. Pour les orientations de 0° et de 100° , les façades sont retrouvées. Il n'existe aucune façade dans la scène ayant pour orientation 220° , le filtre a réussi à éliminer presque tous les mauvais points.

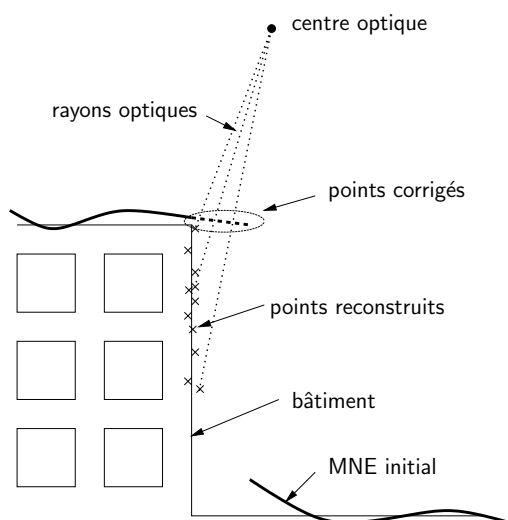


FIG. 4.20 – *Correction du MNE : on creuse dans le MNE pour chaque point reconstruit sur la façade. La correction est réalisée en supprimant tous les points du MNE qui se trouvent entre les points et les centres optiques des images qui ont permis de les reconstruire.*

de 80° , 100° et 120° , on voit que les façades réelles sont extraites dans la carte de 100° . Dans les deux autres cartes, on trouve de petites surfaces discontinues le long des façades. Cette forme en escalier provient du fait que les voisinages utilisés lors de la mise en correspondance sont trop différents, comme pour les façades lorsque on utilise l'algorithme de mise en correspondance par corrélation des images initiales (figure 4.15). En observant la somme des cartes d'accumulation obtenues pour toutes les orientations et l'ortho-image du bâtiment traité, on voit que l'algorithme a permis de retrouver et de reconstruire des points sur toutes les façades du bâtiment.

4.5.8 Résultats de la correction des MNE

Pour corriger les limites du bâtiment nous utilisons les points 3D obtenus pour chaque orientation traitée. Tout d'abord, nous présentons les résultats avec des images de synthèse dans la figure 4.22. L'image A est l'une des neuf vues d'une scène simple texturée : un plan horizontal représente le sol, un parallélépipède représente un bâtiment et une source lumineuse représente le soleil. L'image B est le MNE extrait automatiquement par la chaîne de production de MNE d'ISTAR. Nous voyons dans cette image que le bâtiment reconstruit est en dehors de ses limites réelles (trait fin). La correction du MNE est présentée dans l'image C. A cause des ombres, aucun point n'a pu être apparié sur la façade de gauche. L'algorithme de corrélation a besoin d'une surface suffisamment texturée pour pouvoir la retrouver. Les trois autres façades ont été retrouvées et corrigées.

Les résultats avec un cas réel sont présentés dans la figure 4.23. A cause du contraste entre le toit et le sol (surtout dans les ombres) nous voyons dans le MNE (image A) que le bâtiment reconstruit est en dehors de ses limites réelles (trait fin). La correction réalisée par l'algorithme est présentée dans l'image C. L'algorithme a pu retrouver et corriger une grande partie des limites du bâtiment. Malgré le bruit important qui est présent avec les points 3D reconstruits (faux appariements), le filtrage a été efficace et a permis de conserver uniquement les points des façades. On voit tout de même que quelques données aberrantes

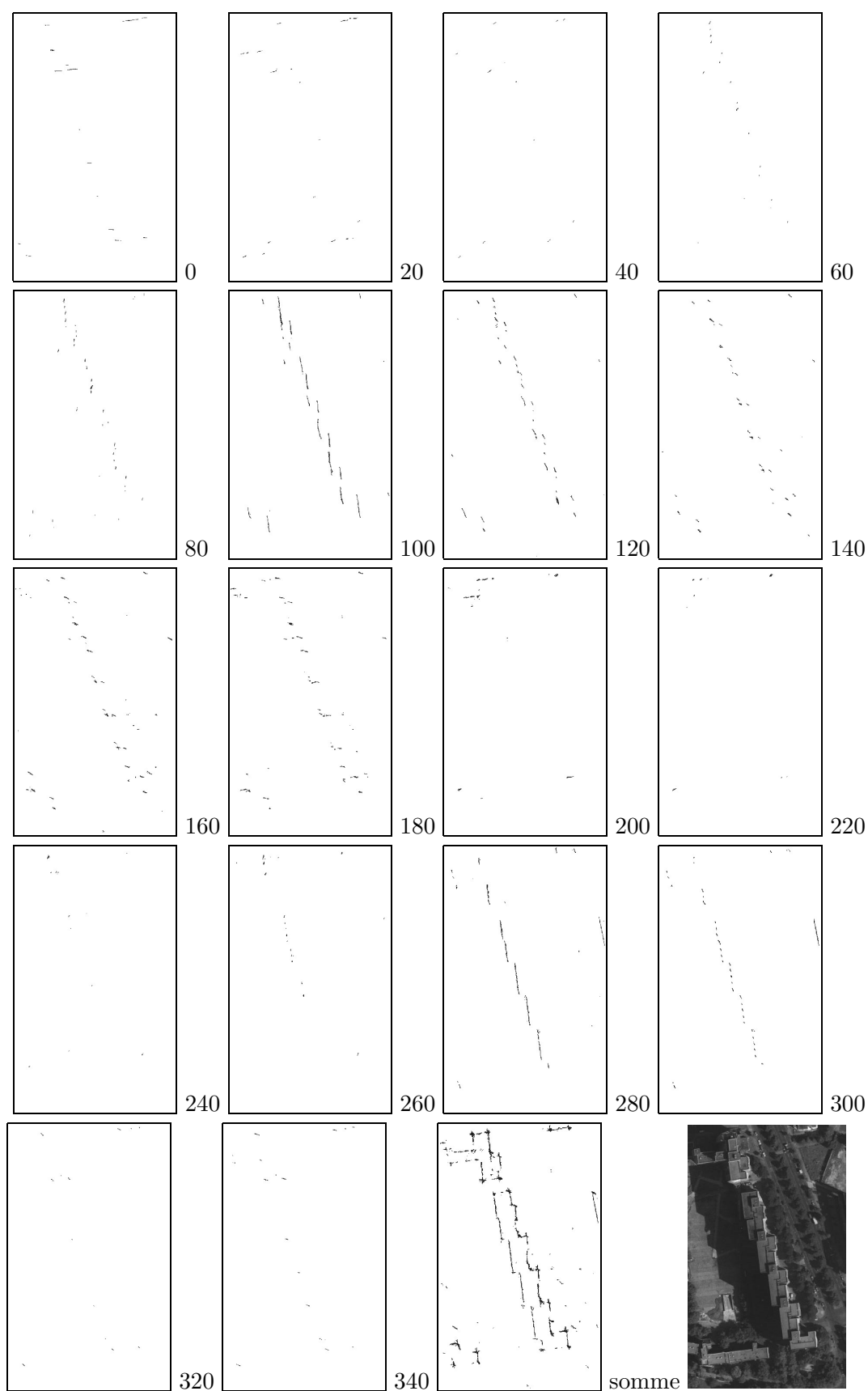


FIG. 4.21 – Présentation des cartes d'accumulation obtenues pour chaque orientation traitée. Nous présentons aussi la carte somme des cartes d'accumulation pour toutes les orientations.

n'ont pu être supprimées dans le bâtiment du haut. Dans les deux cas, avec des images synthétiques ou des images réelles, l'algorithme a été capable de retrouver les limites du bâtiment et de corriger la forme de celui-ci. La figure 4.24 présente six vues de l'ensemble des points reconstruits avec le bâtiment réel traité. On peut voir que ces points correspondent bien aux façades de ce bâtiment, et qu'il sont répartis sur toute sa hauteur.

Programmation et paramètres importants L'algorithme de reconstruction automatique des façades a été développé en C++ et PERL. Nous utilisons plusieurs outils développés à ISTAR : l'algorithme de mise en correspondance par corrélation (*match1d*), la librairie de format d'images et la librairie capteur (transformations géométriques). Tous les autres outils ont été développés au cours de cette thèse : extraction du sol, extraction des bâtiments, choix des images initiales, construction des images de façades, les différents filtres, construction des cartes d'accumulation, correction de MNE, ... Les points reconstruits peuvent être sauvegardés en VRML et OOGL (pour Geomview¹). Plusieurs scripts PERL permettent de faciliter l'utilisation de l'ensemble des programmes développés. Une seule commande permet actuellement de lancer l'algorithme automatique.

Très peu de paramètres doivent être choisis. L'extraction automatique des bâtiments permet de définir une boîte englobante de la zone d'intérêt à traiter. Pour aider la mise en correspondance des images des façades, nous utilisons une image de contraintes estimée à partir du MNE initial. Les espaces de disparité utilisés par l'algorithme de corrélation sont déterminés automatiquement. Concernant le paramètre de l'orientation, nous échantillons l'espace des orientations des façades avec un pas de 20°.

Les paramètres importants sont ceux des filtres. Pour le premier filtre, la normale à la surface reconstruite doit être proche de l'orientation de la façade étudiée (15°). Pour le deuxième filtre, nous avons choisi d'utiliser une carte d'accumulation deux fois plus résolue que le MNE. Les points projetés dans des cellules dont le score d'accumulation est inférieur à 2 sont écartés. Pour le dernier filtre, nous avons défini la taille des voxels de 1 mètre en X et Y et de 3 mètres en Z . Après le vote des points dans un voisinage représentant une façade, nous conservons les points 3D pour lesquels le voxel correspondant possède au moins 3 points et un vote de poids supérieur à 300.

En changeant la résolution des images ou celle du MNE, les paramètres des deux premiers filtres n'ont pas à être modifiés. Pour le filtre 3D, il sera sûrement nécessaire de réadapter la taille des voxels de la grille ainsi que le seuil limite des votes. Celui-ci est estimé actuellement de manière empirique. Il dépend de la densité de points reconstruits au niveau d'une façade.

4.6 Conclusion

Nous avons présenté dans ce chapitre plusieurs outils permettant la reconstruction et l'utilisation des façades de la scène pour corriger un MNE. Nous avons tout d'abord montré comment construire des images dans lesquelles les façades d'une orientation donnée ont le même aspect visuel. Nous utilisons alors un algorithme de corrélation standard pour appairer les façades et reconstruire les points 3D qui appartiennent à ces surfaces verticales. Nous avons montré que la localisation de ces points est bonne et que la reconstruction est peu sensible vis à vis de l'orientation.

1. <http://www.geom.umn.edu/software/geomview/>

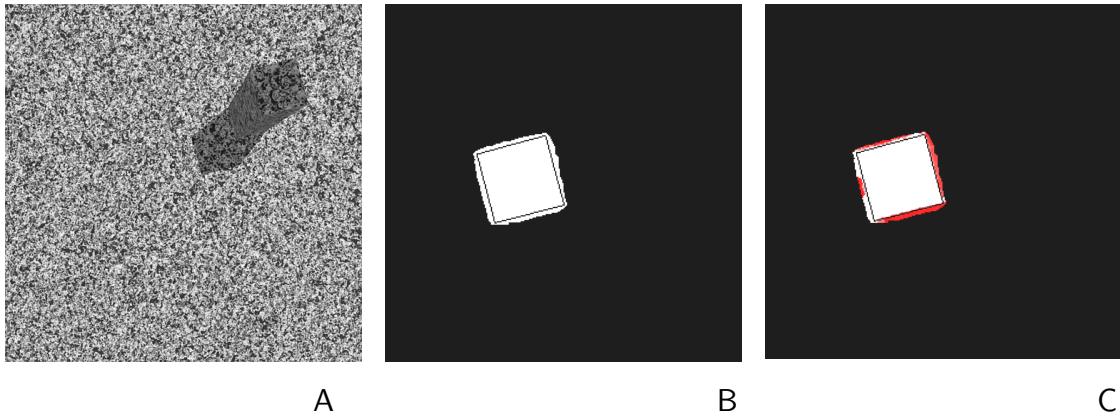


FIG. 4.22 – Résultats sur un bâtiment synthétique : A est l'une des neuf vues synthétiques de la scène. B est le MNE reconstruit automatiquement par la chaîne de production d'ISTAR. Le toit est en dehors des limites réelles (trait fin). Les résultats de la correction de MNE sont présentés dans la figure C, les zones rouges ont été supprimées du MNE. Trois façades ont été retrouvées et corrigées.

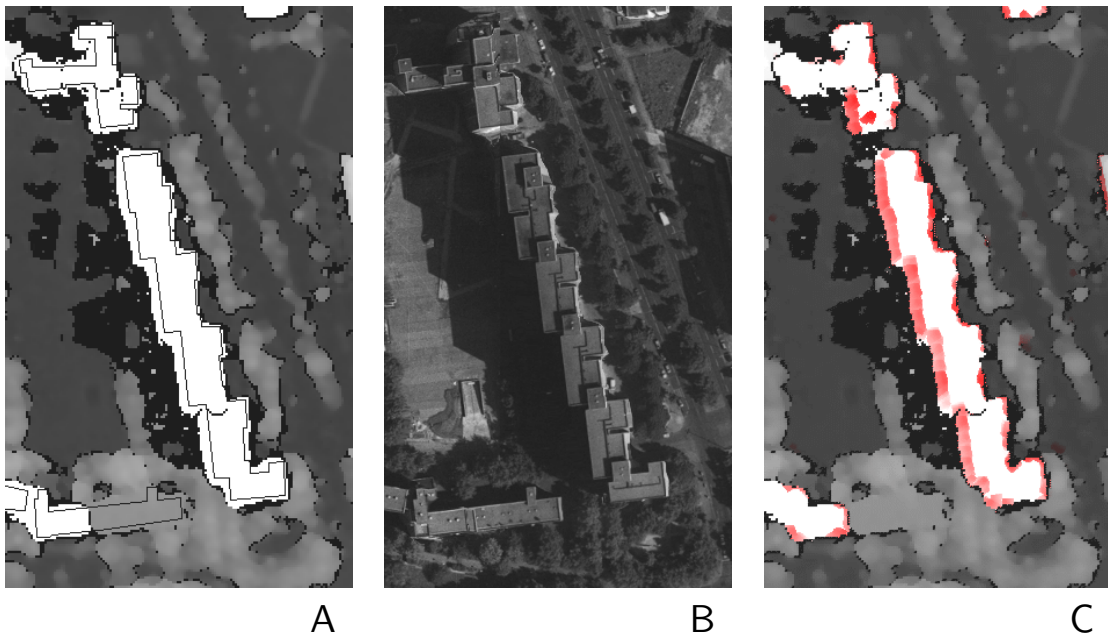


FIG. 4.23 – Résultats sur un bâtiment réel : A est le MNE initial du bâtiment traité. B est l'ortho-image correspondante. Dans A, le toit est en dehors des limites réelles du bâtiment (trait fin). C est le MNE résultat après la correction, les régions rouges ont été supprimées du MNE, et les limites réelles du bâtiment ont été retrouvées.

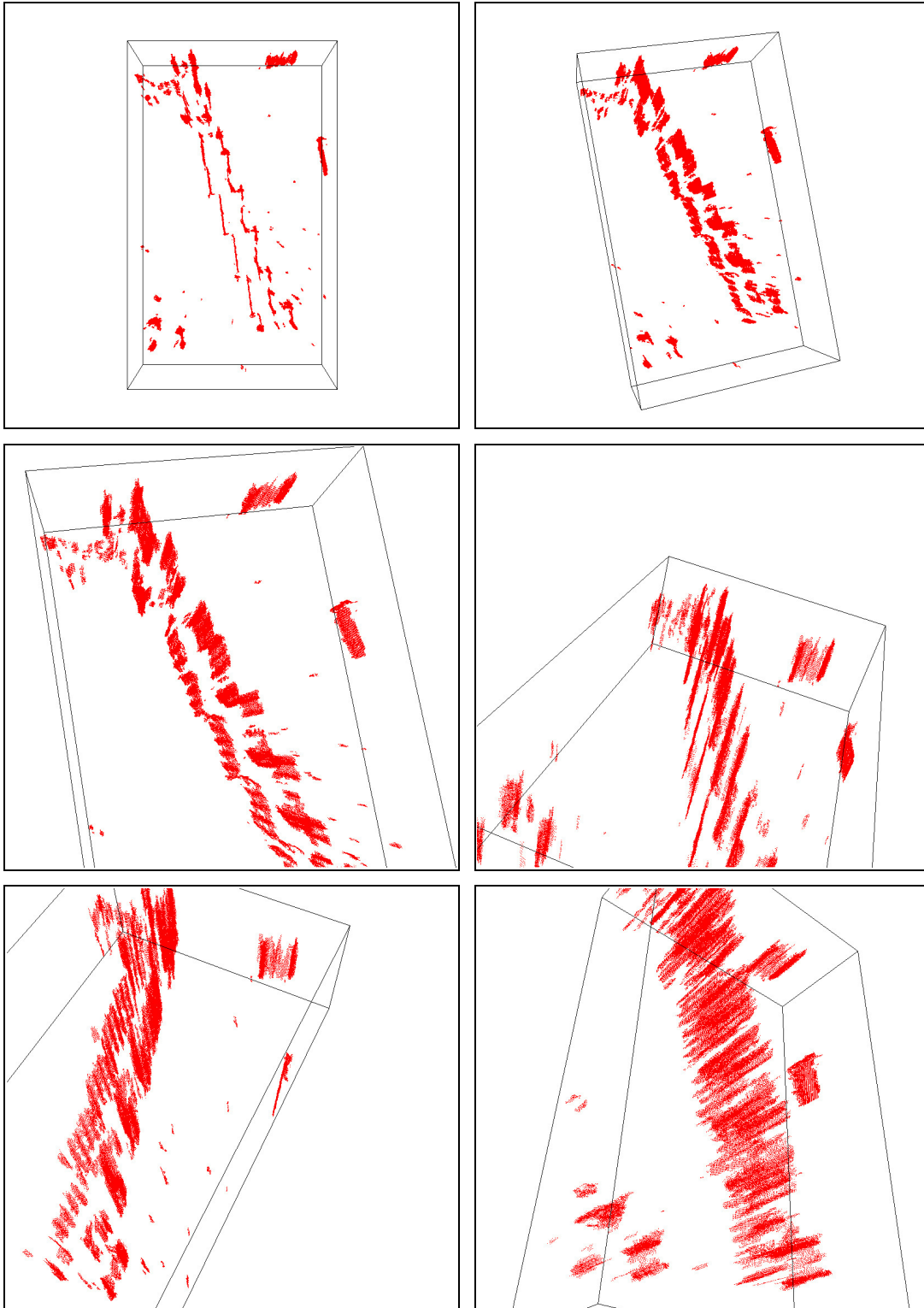


FIG. 4.24 – *Visualisation des points 3D reconstruits. Les images correspondent à six vues des points 3D reconstruits sur les façades du bâtiment traité. Nous avons utilisé Geomview pour cette visualisation.*

Nous avons ensuite présenté un algorithme permettant la reconstruction de toutes les façades d'un bâtiment sans aucun a priori sur la forme de ce bâtiment ni l'orientation de ses façades. Il ne dépend pas du nombre d'images disponibles. L'algorithme traite indépendamment toutes les orientations possibles des façades pour reconstruire les points 3D. Un filtrage en trois passes permet de supprimer l'importante quantité de bruit présent dans les données. Nous avons montré que l'algorithme est capable de reconstruire ces points avec précision dans des cas réels et synthétiques. Nous avons enfin proposé une technique qui utilise les points reconstruits pour corriger les limites des bâtiments dans le MNE. Ces points peuvent aussi être directement utilisés pour reconstruire le modèle 3D du bâtiment. Nous verrons dans le chapitre 7 plusieurs autres utilisations des points reconstruits sur les façades.

Chapitre 5

Segmentation d'un Modèle Numérique de Terrain

Ce chapitre traite de la segmentation d'un Modèle Numérique de Terrain (MNE). La segmentation de la scène que nous souhaitons obtenir consiste à extraire les régions planes des toits pour construire un modèle polyédrique de chaque bâtiment. L'objectif est d'extraire du MNE une description simple et représentative de chaque bâtiment de la scène sans aucune connaissance a priori sur sa forme, sinon qu'il peut être approximé par un polyèdre. Parce que nous utilisons un MNE comme donnée initiale, ce problème de segmentation peut être considéré comme la modélisation d'un nuage de points 3D en présence de bruit.

Ce chapitre est composé de cinq parties. Tout d'abord nous présentons les techniques d'estimation de modèle. Nous abordons ensuite le problème de segmentation de données en plans, la reconstruction des surfaces à partir de points 3D. Après avoir défini l'approche choisie, les troisième et quatrième parties présenteront les deux concepts importants de notre approche, l'estimation robuste et la sélection de modèles. Des revues récentes de ces deux domaines sont fournies par le rapport technique de *P. Torr* [Tor99] pour la sélection de modèle et l'article de *C. Stewart* [Ste99] pour l'estimation robuste. La dernière partie présente l'algorithme de segmentation de MNE que nous proposons pour résoudre le problème de segmentation qui combine les méthodes vues au cours de ce chapitre.

5.1 Estimation de modèles de surface

Dans les chapitres 2 et 4 nous avons vu plusieurs systèmes permettant la mise en correspondance des images. Le résultat de ces systèmes est un ensemble de points 3D peu structuré soumis à un important niveau de bruit. Il est souvent nécessaire de transformer cet ensemble de données tridimensionnelles en ensembles plus compacts et structurés pour faciliter l'analyse et l'utilisation de la structure de la scène. C'est le cas pour certaines des applications télécoms qui nous intéressent (propagation des ondes électromagnétiques en milieu urbain). L'interprétation directe d'un ensemble de points 3D est difficile notamment en présence de bruit. Modéliser ces données va permettre de structurer la forme des données et de les simplifier. Il sera également possible d'imposer des contraintes de régularité afin que le modèle soit plus cohérent. La modélisation de données permet enfin de diminuer considérablement la taille nécessaire pour stocker la description de ces données.

Pour que la description soit simple et représentative, il est nécessaire de choisir un modèle d'objet adapté à la structure des données. Dans la plupart des applications, des modèles d'objets simples comme une droite ou un plan permettent de représenter fidèlement l'ensemble des données. Le cas que nous proposons d'étudier dans cette section est celui de la reconstruction d'un modèle de bâtiment polyédrique. Le bâtiment est composé uniquement de morceaux de plans 3D. Le modèle de surface tridimensionnelle que nous utiliserons est la **facette plane**. Nous supposons que l'on dispose de données en nombre suffisant pour reconstruire chaque facette du modèle de bâtiment. La segmentation consiste à trouver le bon jeu de facettes pour décrire le modèle de bâtiment, trouver les points 3D qui supportent chaque modèle de surface et déterminer les paramètres de chaque facette en tenant compte de la présence du bruit.

Remarque : Nous allons souvent utiliser le mot **modèle** dans ce chapitre. Il convient de distinguer les différentes significations que nous donnons à ce mot. Nous différencions deux sens distincts. Les **modèles d'objets** (ou **modèles de surfaces** dans notre cas) sont les modèles mathématiques génériques utilisés pour décrire la surface, les facettes planes dans notre approche. Dans un sens plus général, nous appelons les **modèles** les entités de la scène résultant de la modélisation : modèles de surface munies de leur géométrie (les paramètres).

5.1.1 Estimation des paramètres d'un modèle de surface

Nous cherchons à modéliser une surface tridimensionnelle \mathbf{S} définie par un jeu \mathbf{Y}_i de n points $m_i(x_i, y_i, z_i)$, avec $i = 1, \dots, n$ dans un espace euclidien tridimensionnel \mathcal{R}^3 muni du repère orthonormé $(O, \vec{u}, \vec{v}, \vec{w})$. Les surfaces tridimensionnelles peuvent être décrites sous une forme explicite ou implicite. La forme explicite d'une surface est le graphe d'une fonction de deux variables. La forme explicite généralement employée est la suivante :

$$z = f(x, y)$$

avec (x, y) coordonnées dans le plan image et z la profondeur dans le cas d'une application de vision, ou avec (x, y, z) coordonnées cartographiques dans le cas de modélisation d'une scène 3D urbaine. La forme implicite d'une surface s'écrit comme suit :

$$f : \mathcal{R}^3 \rightarrow \mathcal{R} \mid f(x, y, z) = 0.$$

On peut citer aussi les surfaces paramétriques, qui font partie des surfaces explicites. Les surfaces écrites sous une forme explicite peuvent toujours être réécrites sous une forme implicite mais l'inverse n'est pas vrai. *R. Bolle* et *B. Vemuri* présentent dans [BV91] une revue des méthodes d'estimation de surface. *G. Taubin* traite aussi dans [Tau91] l'estimation de plusieurs modèles d'objets par des équations implicites.

Quels que soient la forme et le modèle de surface choisis, l'estimation des paramètres du modèle de surface peut être considéré comme un problème d'optimisation, nous présentons ici le cas d'une surface polynomiale. Soit $f(x)$ l'équation d'un modèle de surface telle que :

$$\begin{aligned} f(x, y, z) &= 0 \\ \Leftrightarrow ax + by + cz + \dots &= 0 \\ \Leftrightarrow \mathbf{F}\mathbf{X} &= 0 \end{aligned}$$

où $\mathbf{F} = (a, b, c, \dots)$ correspond aux paramètres du modèle et $\mathbf{X} = (x, y, z, x^2, y^2, \dots)$ est le vecteur des variables. Tous les points $m = (x_m, y_m, z_m)$ d'une même surface vérifient la même

équation. Disposant d'un ensemble de n données $\mathbf{Y}_i = \{m_i(x_i, y_i, z_i)\}$ avec $(i = 1, \dots, n)$, le but est de déterminer le jeu de paramètres \mathbf{F} qui représente au mieux ces données. Dans le cas de données non bruitées, la solution de ce problème d'estimation revient à trouver le jeu de paramètres $\tilde{\mathbf{F}}$ qui vérifie le système d'équations suivant :

$$f(x_i, y_i, z_i) = 0 \quad \forall i$$

Dans la majorité des applications, les données sont bruitées. On définit une mesure $r_i = d(m_i, \mathbf{F})$, le résidu du point m_i , qui représente la distance d'un élément $m_i = (x_i, y_i, z_i)$ au modèle \mathbf{F} . La fonction distance $d(m_i, \mathbf{F})$ peut être la distance algébrique, la distance euclidienne ou toute autre distance. Le but n'est plus de trouver le modèle qui passe par les données, mais de trouver le modèle qui approxime au mieux ces données. On cherche donc intuitivement à minimiser l'écart des données mesurées au modèle. Les paramètres estimés du modèle sont obtenus par :

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F}} \sum_{i=1}^n \rho(r_i) \quad (5.1)$$

où $\rho(r_i)$ est une fonction non-négative, monotone dans sa partie positive et paire.

Les techniques employées pour l'optimisation de ce problème peuvent être multiples : moindres carrés linéaires, moindres carrés pondérés itératifs (IRLS, Iterated Re-weighted Least Squares), descente par gradient conjugué, itérations de Newton, méthode de Levenberg-Marquardt ... Ces méthodes s'appliquent à tous types d'optimisation de fonctions supposées *lisses*. La fonction ρ généralement utilisée est la fonction quadratique $\rho(x) = x^2$ et peut donc être résolue par des techniques dites de *moindres carrés*. K. Kanatani propose dans [Kan96] et [Kan98] une technique générale d'estimation optimale de paramètres de modèles pour différents problèmes de vision par ordinateur. Cette technique est basée sur l'estimateur du Maximum de Vraisemblance (MV) muni d'une opération de *renormalisation*. Il propose alors pour différents problèmes, dont ceux qui nous concernent, des estimateurs optimaux dans les conditions de bruit isotropique gaussien, de bruit anisotropique et de bruit anisotropique et inhomogène.

5.1.2 Les estimateurs robustes

Les techniques d'estimation présentées précédemment permettent d'approximer un ensemble de données ou de points par un modèle choisi. Leur utilisation apportera une solution efficace dans le cas où seul un bruit connu, de type gaussien généralement, est présent dans les données. Dans le cas où des erreurs d'autres types telles que celles dues aux mauvais appariements pour notre application sont présentes, le résultat obtenu sera altéré voire complètement perturbé et inutilisable. On utilise alors les techniques dites d'**estimation robuste de paramètres** pour limiter l'effet néfaste des données hautement altérées. Les techniques robustes sont composées dans la majorité des cas de deux processus différents : un premier qui recherche les données aberrantes (**outliers**) qui perturbent l'estimation et un second processus qui estime les paramètres du modèle en limitant leur effet (qui peut être une des techniques d'estimation traditionnelle évoquées précédemment). Nous présentons l'estimation robuste dans la section 5.3.

5.2 Reconstruction de surfaces à partir de données 3D

La reconstruction de surfaces à partir de données consiste à combiner des objets locaux (points, surfaces locales . . .) pour reconstruire des primitives globales représentant les différentes surfaces de la scène. Il existe deux approches pour la segmentation de données 3D (Segmentation of range images) : les approches de segmentation basées sur les régions et les approches de segmentation basées sur les contours. Le premier type d'approches construit les différentes surfaces de la scène en regroupant les points qui ont des propriétés similaires tandis que dans le second, on extrait les discontinuités de la scène pour guider le groupement des primitives locales.

Segmentation de données basée sur les régions : Parmi les méthodes de segmentation basées sur les régions, on peut considérer plusieurs catégories de méthodes. Une première catégorie utilise les propriétés différentielles locales du voisinage des points pour regrouper ceux-ci. En chaque point 3D de la scène, on procède à une analyse de la surface (courbure, normale à la surface . . .) puis on regroupe les points voisins qui possèdent des propriétés différentielles concordantes. *P. Fua* et *P. Sanders* utilisent dans [FS92] une procédure itérative pour obtenir les propriétés différentielles de la surface en chaque point de l'ensemble des données. Le voisinage d'un point est défini par une sphère centrée sur ce point. Une première approximation de la surface par un plan est réalisée dans chaque voisinage, elle permet de définir un référentiel local pour chaque donnée. Les auteurs approximent ensuite la surface locale par une quadrique en utilisant une estimation robuste. Les données aberrantes sont éliminées et les données justes sont reprojctées sur la surface calculée. Cette procédure est répétée plusieurs fois pour lisser les données. Les auteurs rééchantillonnent ensuite les surfaces sur une grille régulière pour obtenir une description dense de l'objet. Ils regroupent enfin les points qui appartiennent aux mêmes surfaces et construisent une triangulation de Delaunay pour obtenir le modèle final. *C. Tang* et *G. Medioni* reconstruisent dans [TM98] les surfaces en proposant une méthodologie de groupement perceptuel basée sur le Tensor voting. Utilisant un tableau tridimensionnel régulier, les points 3D munis de la normale à la surface locale votent dans les cellules voisines. Le vote est réalisé par un tenseur qui représente la surface la plus probable passant par le point 3D votant et le centre de cette cellule. Tous les votes des cellules sont ensuite interprétés pour extraire la surface. D'autres approches comme celle de l'ENST [RMG97] guident la reconstruction de surface par le résultat d'une segmentation radiométrique (regroupement selon les propriétés radiométriques des données au lieu des propriétés différentielles).

Une seconde catégorie de méthodes de segmentation basées sur les régions utilise les techniques d'échantillonnage aléatoire pour retrouver les différentes surfaces de la scène. Ces méthodes ont l'avantage d'être beaucoup plus rapides et surtout d'être robustes vis-à-vis des données aberrantes présentes dans l'ensemble des données. Les premières techniques proposées pour la segmentation sont la procédure du RANSAC [FB81] et la moindre médiane des carrés [RL87] (section 5.3.3). Plusieurs autres méthodes basées sur l'une de ces deux techniques ont ensuite été proposées pour résoudre le problème de segmentation : RESC [YBK94], ALKS [LMP98], MUSE [MS96b] . . . Nous avons choisi une technique d'échantillonnage aléatoire pour résoudre le problème de segmentation de données d'un MNE. Une description détaillée de plusieurs de ces méthodes est présentée dans la section 5.3 et la stratégie que nous avons développée est présentée dans la section 5.5.

D'autres méthodes considèrent le problème de segmentation sous la forme d'un problème d'optimisation globale. *P. Fua* utilise un système de particules dans [Fua97], une particule

représentant la surface locale (un plan ou une quadrique) définie pour un point 3D donné. Dans un tableau tridimensionnel, l'auteur initialise le système de particules à partir d'une carte de disparité et d'estimation robuste. Il regroupe les particules pour obtenir une première estimation des surfaces puis propose ensuite une technique d'optimisation pour affiner le modèle global. Il définit pour cela une énergie qui prend en compte à la fois des scores de corrélation et de l'interaction entre les particules.

Segmentation basée sur les contours Les approches de segmentation basées sur les contours extraient des discontinuités de la scène pour guider le groupement des primitives locales. *W. Hoff* et *N. Ahuja* présentent dans [HA89] une approche multi-résolutions. Pour chaque point, les auteurs approximent la surface locale (par moindres carrés) par un plan, ou par une surface quadratique dans les cas où un plan n'est pas assez représentatif. Ils extraient ensuite les discontinuités de surfaces, puis interpolent les surfaces délimitées par ces discontinuités. Leur procédure est utilisée sur une pyramide de quatre résolutions d'images. Plus récemment, dans [CL97], *L. Chen* et *W. Lin* proposent d'utiliser la dualité des approches basées sur les régions et les contours. Une technique basée sur la croissance de région permet de regrouper les surfaces locales en surfaces globales. Les limites de ces régions sont obtenues par une stratégie utilisant à la fois les contours extraits des images et les contours provenant des discontinuités déduites des intersections des surfaces globales. Ils définissent une étape de complétion permettant d'obtenir des contours fermés.

5.2.1 Choix de la stratégie

A. Hoover et al. présentent dans [HBJJ⁺96] une étude comparative expérimentale de plusieurs algorithmes de segmentation. Quatre d'entre eux sont évalués et aucun ne se montre supérieur sur la totalité des critères qu'ils définissent. Les discontinuités de surface et d'orientation ne sont pas assez francs dans l'ensemble des données pour utiliser une approche par contour. L'approche que nous avons décidé d'utiliser pour résoudre le problème de segmentation de données d'un MNE est basée sur les régions et les techniques robustes. La principale raison de ce choix est la présence importante de données aberrantes dans le MNE. Les données sont extrêmement perturbées par ces erreurs ainsi que par toutes les petites structures présentes sur les toits. Les autres approches basées sur les régions ne permettent pas de gérer ces perturbations aussi facilement, les résultats sont considérablement altérés. De plus, ces autres approches utilisent des critères ou des techniques qui lissent la surface extraite.

L'approche que nous avons développée est présentée dans la section 5.5. Elle est basée sur l'algorithme ExSel++ présenté par *M. Stricker* et *A. Leonardis* dans [SL95a]. Nous avons choisi cette approche car elle permet d'utiliser modèles de surfaces et d'extraire toutes les hypothèses de modèles de la scène. Le principe est de construire dans une première étape une liste d'hypothèses de modèles à partir des données de la scène. Nous utilisons pour cela une technique robuste d'échantillonnage aléatoire permettant de retrouver les différentes surfaces des bâtiments. Dans un second temps, nous trions les modèles de la liste et sélectionnons le jeu minimal de modèles représentant au mieux la totalité de la scène. Avant de présenter cette approche dans les détails, nous présentons dans les sections 5.3 et 5.4 les estimateurs robustes et la sélection de modèles qui forment les bases de cette approche.

5.3 Les estimateurs robustes

Les techniques robustes se caractérisent par leur capacité à retrouver l'information à partir d'un ensemble de données composé de données justes (information perturbée par un faible bruit de type gaussien) et des données aberrantes (valeurs ne correspondant à aucune information). Les différentes méthodes sont principalement évaluées sur leur capacité à retrouver l'information essentielle d'un ensemble de données malgré un taux de données aberrantes important. La limite de ce taux est appelée le point de rupture de l'estimateur. Il représente le taux limite de données aberrantes à partir duquel l'estimateur ne pourra plus retrouver le modèle. Une technique basée sur le critère des moindres carrés possédera un point de rupture égal à 0. Une seule donnée aberrante présente dans l'ensemble des données pourra perturber totalement l'estimateur. *P. Rousseeuw* et *A. Leroy* affirment dans [RL87] que le point de rupture maximum théorique est de 50% car si plus de la moitié des points est erronée, elle *conspirera* vers un modèle qui paraîtra meilleur que le modèle correct. Si au contraire, nous supposons que cette conspiration n'est pas à sens unique (vers un autre modèle unique), nous serons alors capable de tolérer un point de rupture bien supérieur. Certains algorithmes comme le RANSAC (RANdom SAmple Consensus) [FB81] ou le LMS (Least Median of Square) [RL87] peuvent alors atteindre des points de rupture théoriques de 90%. Plusieurs articles comme [MMRK91, Ste99, TM97, Zha97] présentent et comparent différents estimateurs robustes.

5.3.1 Les M-Estimateurs

On considère qu'une approche standard par moindres carrés suffit en cas de présence de données aberrantes mais leur présence va perturber le processus estimateur et fausser la solution du problème. De telles données vont générer, dans le cas d'une technique de minimisation par moindres carrés, des résidus élevés qui vont contribuer de manière plus importante que les données correctes au résultat final. L'idée des M-estimateurs [Hub81] est de suivre la formulation du Maximum de Vraisemblance (MV) en cas de bruit non gaussien, en diminuant l'influence (plutôt que de la supprimer) des données aberrantes lors de l'estimation des paramètres du modèle. On va attribuer une pondération de confiance à chaque élément i de l'ensemble de données \mathbf{Y}_i en fonction de la distance r_i qui le sépare du modèle.

Les M-estimateurs peuvent se présenter selon plusieurs formes différentes, nous les introduisons dans ce chapitre sous la forme d'une fonction $\rho(r_i)$, r_i étant le résidu du point, telle que définie dans l'équation 5.1 pour rester cohérent avec les notations. Nous appelons la fonction $\rho(r_i)$ la fonction d'objectif. De nombreuses fonctions $\rho(r_i)$ ont été proposées. Elles permettent d'obtenir des comportements différents et adaptés à des applications qui nécessitent des méthodes d'estimation robuste. Le choix d'une fonction peut être justifié par la connaissance de la forme des perturbations ou, lorsque nous ne disposons d'aucune information sur l'origine de la perturbation, à partir d'une étude comparative des diverses fonctions proposées.

Écart type de l'erreur σ Certains critères, comme ceux présentés dans le tableau de la figure 5.1, sont définis à partir de la connaissance de l'écart type de l'erreur σ des données justes. Nous pouvons estimer sa valeur $\hat{\sigma}$ avec l'estimateur du Maximum de Vraisemblance. Nous utilisons la médiane des résidus d'une distribution gaussienne pour éliminer les effets

due à la présence de données aberrantes dans l'ensemble des données :

$$\hat{\sigma} = \frac{\text{med}_i r_i}{0.6745} \quad (5.2)$$

Cet estimateur est asymptotiquement consistant. Nous pouvons l'utiliser lorsque la taille du support du modèle est importante. Lorsque le nombre n d'échantillons est proche du nombre de paramètres p nécessaires pour définir un modèle, nous devons utiliser l'estimateur suivant :

$$\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n-p} \right) \sqrt{\text{med}_i r_i^2} \quad (5.3)$$

Estimateur	Fonction $\rho(r_i)$	Échelle des résidus
L_2	$\frac{r_i^2}{2}$	
L_1	$ r_i $	
Fonction de Huber	$\frac{r_i^2}{2}$ $a(r_i - \frac{a}{2})$	$r_i \leq a$ $r_i > a$
Bi-pondération de Tuckey	$\frac{a^2}{6}(1 - [1 - (\frac{x}{a})^2]^3)$ $\frac{a^2}{6}$	$r_i \leq a$ $r_i > a$
Fonction de Welsch	$\frac{a^2}{2}[1 - \exp(-\frac{r_i}{a})^2]$	

FIG. 5.1 – Quelques fonctions pour les M-estimateurs. La valeur de a est une constante de réglage dépendant de l'écart type $\hat{\sigma}$ et de la proportion de données aberrantes ($a \approx 1,5 \cdot \hat{\sigma}$ pour Huber et $a \approx 6 \cdot \hat{\sigma}$ pour Tuckey par exemple).

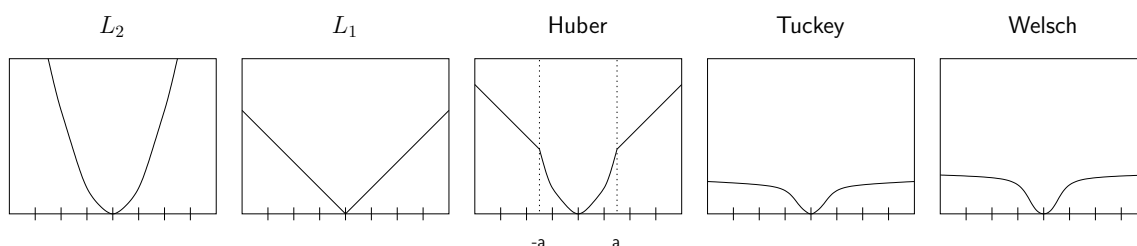


FIG. 5.2 – Représentation graphique de plusieurs M-estimateurs.

Le tableau de la figure 5.1 nous présente plusieurs M-estimateurs. Leur représentation graphique est présentée dans la figure 5.2. Toutes les fonctions $\rho(r_i)$ ont approximativement le même comportement général : pour les points qui ont de faibles résidus, que l'on suppose affectés par un bruit gaussien, la fonction de pondération $\rho(r_i)$ doit avoir un comportement quadratique comme celle proposée par P. Huber. La confiance en ces points est totale, l'influence de ces points sera importante dans l'estimation des paramètres du modèle. Les points dont la valeur des résidus est plus forte conservent une confiance certaine mais possèdent une influence moindre, la fonction de pondération $\rho(r_i)$ va avoir un comportement plutôt linéaire. Les derniers points qui possèdent des résidus forts sont des données aberrantes et leur influence dans l'estimation doit rester très faible, même pondération pour tous comme

pour la fonction de *Welsch*. On remarque que toutes les fonctions ne suivent pas ce comportement général, tout dépend de l'application et du comportement souhaité. On peut noter toutefois que les données aberrantes ne sont pas éliminées de l'ensemble de données initiales.

L'estimation des paramètres du modèle s'effectue dans un schéma à deux temps, calcul des pondérations et estimation, comme celui de l'estimateur itératif des moindres carrés pondérés (IRLS). La pondération est la fonction $\rho(r_i)$. Pour chaque estimée \mathbf{F} des paramètres, on calcule tous les résidus des données de l'ensemble initial. Nous estimons alors l'écart type des résidus $\hat{\sigma}$ par rapport au modèle fixé, puis calculons toutes les pondérations de confiance des éléments de l'ensemble de données. Nous pouvons ensuite estimer les paramètres du modèle par minimisation de la somme des résidus pondérés. Le processus est itératif et demande généralement 3 à 5 itérations avant de converger vers la solution lorsque l'estimée initiale des paramètres est proche de la solution. Il est nécessaire d'avoir une bonne estimée initiale pour que l'algorithme converge vers la solution optimale. La plupart des M-estimateurs possèdent un point de rupture de 20% à 30% selon le choix des pondérations et des applications.

Les R-estimateurs et les L-estimateurs Il existe deux autres types d'estimateurs dont le principe est de filtrer les données les plus critiques par pondération. La catégorie des R-estimateurs est basée sur l'arrangement du jeu de résidus. L'idée est de classer les données selon les valeurs de leurs résidus, puis d'affecter une pondération à chaque donnée qui dépende du classement plutôt que de la valeur du résidu. On réécrit l'équation 5.1 en utilisant une pondération multiplicative, la fonction de score $a_n(R_i)$:

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F}} \sum_i a_n(R_i) r_i$$

où n est le nombre de données et R_i la position du résidu r_i dans la liste ordonnée des résidus. Un exemple de fonction de score est celle de *Wilcoxon*: $a_n(R_i) = R_i - (n + 1)/2$. L'avantage de ce type d'estimateur est son indépendance vis-à-vis de l'écart type $\hat{\sigma}$. La méthode est d'échelle invariante.

Les L-estimateurs sont basés sur les fonctions de puissance α . La minimisation est la même que celle des M-estimateurs mais utilise une fonction ρ telle que $\rho(r_i) = |r_i|^\alpha$. Dans le cas où $\alpha = 1$ on utilise comme fonction d'objectif la fonction: $\rho(r_i) = |r_i|$. Dans le cas où $\alpha = 2$, on se ramène à une minimisation des moindres carrés. Cette classe d'estimateurs donne de moins bons résultats que les deux autres catégories présentées auparavant.

5.3.2 Case Deletion Diagnostic

Cette catégorie d'estimateurs robustes est basée sur l'étude du comportement du problème lorsque nous supprimons une donnée de l'ensemble initial. Les données justes *votent* toutes pour le même modèle que nous cherchons. Les données aberrantes conspirent individuellement vers un autre modèle, leur influence sur les paramètres est d'autant plus importante qu'elles sont aberrantes. Pour retrouver les données aberrantes, on supprime individuellement chaque donnée de l'ensemble initial, puis on examine la réaction du système. Les éléments les plus perturbateurs (qui sont supposés être en minorité) correspondent aux données aberrantes. La difficulté de ces méthodes est de définir la procédure d'analyse. Des mesures d'influence existent dans la littérature statistique et dans le domaine de vision par ordinateur. *P. Torr* et *D. Murray* étudient dans [TM97] cette technique pour estimer la matrice fondamentale. Le résultat est très peu dépendant de la solution initiale, mais nécessite

toutefois un temps de calcul important ainsi qu'une très bonne estimée de l'écart type $\hat{\sigma}$ du bruit.

5.3.3 Échantillonnage aléatoire

Cette catégorie d'estimateurs robustes s'oppose aux techniques conventionnelles de filtrage des données qui demandent une bonne solution initiale pour assurer une convergence vers le modèle correct. Plutôt que d'utiliser toutes les données en même temps, les techniques d'échantillonnage aléatoire vont retrouver le modèle correct à partir de petits échantillons de données tirés aléatoirement. Le principe de base des méthodes aléatoires est de déduire à partir d'un nombre suffisant d'hypothèses la meilleure d'entre toutes, ou celle qui possède la plus grande probabilité de contenir uniquement des données correctes. Ces techniques restreignent l'espace des solutions à celui des données en supposant que la solution correcte (ou une très bonne estimée) peut être retrouvée dans l'espace des solutions définies par les données. Ces techniques sont particulièrement adaptées pour l'estimation de modèles à partir d'un ensemble de données initial de taille importante.

Méthode de la moindre médiane des carrés

La méthode de la moindre médiane des carrés (LMS : Least Median of Squares) est présentée comme la résolution du problème de minimisation non linéaire suivant, pour un ensemble de données de n points :

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F}} \text{med}_i r_i^2 \quad \text{pour } i = 0, 1, \dots, n \quad (5.4)$$

La référence principale des techniques basées sur la moindre médiane est le livre de *P. Rousseeuw* et *A. Leroy* [RL87]. La résolution du problème présenté ci-dessus ne peut se réduire à une solution basée sur des techniques de résolution par moindres carrés comme dans le cas des M-estimateurs. La minimisation de la moindre médiane des carrés doit être résolue par une recherche de la solution du problème dans l'espace des solutions définies par l'ensemble de données. Le principe consiste à construire des modèles à partir de sous-ensembles de données, puis à évaluer leur qualité de représentativité pour sélectionner le meilleur. L'espace des solutions définies par l'ensemble de données étant beaucoup trop grand pour un traitement exhaustif, il est nécessaire de construire une base de modèles plus petite mais représentative de l'ensemble des données. On utilise un processus de recherche aléatoire qui extrait plusieurs jeux de données servant chacun à construire une hypothèse de modèle. La sélection du modèle solution est réalisée par la recherche du mode de la probabilité de distribution des résidus.

La résolution du système se fait de la manière suivante : (1) On effectue un tirage aléatoire d'un jeu de p points, p étant le nombre minimal de points pour définir un modèle. (2) On estime les paramètres \mathbf{F}_j du modèle défini par ces points. (3) On calcule tous les résidus r_i , par rapport à ce modèle \mathbf{F}_j , des $n - p$ points restants de l'ensemble de données. (4) On trie les résidus en ordre croissant et conserve la valeur du résidu médian M_j (ou la position du mode de distribution des erreurs, équation 5.6). (5) On réitère les 4 premières étapes jusqu'à avoir réalisé un nombre suffisant de tirages. (6) On extrait le minimum des valeurs M_j conservées pour obtenir la solution du problème $\tilde{\mathbf{F}}$ qui correspond au modèle associé à cette valeur.

Les estimateurs basés sur la moindre médiane des carrés sont biaisés. On utilise donc souvent un estimateur non biaisé pour estimer les valeurs finales des paramètres du modèle.

Une estimée robuste de l'écart type $\hat{\sigma}$ peut être directement obtenue à partir du résultat précédent par les équations 5.2 ou 5.3, on utilise directement la valeur M_j minimum trouvée. L'estimation des paramètres du modèle est alors réalisée par la minimisation des moindres carrés pondérés que nous nommons par la suite RLS pour *Re-weighted Least Square* suivante :

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F}} \sum_i w_i r_i^2$$

où les pondérations sont définies par :

$$w_i = \begin{cases} 1 & \frac{r_i}{\hat{\sigma}} \leq 2.5 \\ 0 & \frac{r_i}{\hat{\sigma}} > 2.5 \end{cases}$$

Procédure de Monte-Carlo Pour déterminer le nombre suffisant de modèles à calculer, on utilise la technique de Monte-Carlo. Considérons la fraction ξ de données altérées dans notre ensemble initial. La probabilité P qu'au moins un p -uplet de points parmi m p -uplets tirés aléatoirement dans notre ensemble de données ne possède aucune donnée aberrante est définie par :

$$P = 1 - [1 - (1 - \xi)^p]^m$$

On peut alors déterminer le nombre de tirages nécessaires m pour obtenir au moins un échantillon correct avec la probabilité P choisie :

$$m = \frac{\ln(1 - P)}{\ln(1 - (1 - \xi)^p)} \quad (5.5)$$

Méthode des moindres carrés tronqués (LTS : Least Trimmed Squares) L'algorithme du LTS est une technique semblable à celle du LMS. La différence réside uniquement dans le problème de minimisation :

$$\tilde{\mathbf{F}} = \arg \min_{\mathbf{F}} \text{moy}_i d_\theta^2(r_i) \quad \text{avec } i = 1, 2, \dots, (n \times \theta + 1)$$

Le principe est de trier les résidus selon un ordre croissant pour obtenir la liste des résidus classés $d_\theta(r_i)$. On utilise alors uniquement la θ -partie des résidus classés (les $\theta\%$ plus faibles valeurs, on prend généralement $\theta = 0.5\%$) et on estime la moyenne de ces résidus. On réitère l'opération pour tous les modèles obtenus par tirage aléatoire et on conserve ensuite le modèle qui possède la plus faible déviation moyenne d_θ .

Les méthodes LQS, LKS et ALKS Si on connaît le taux de contamination ξ de l'ensemble des données par des données aberrantes, il est possible de généraliser les deux algorithmes précédents, le LMS et le LTS, qui possèdent tous deux un point de rupture théorique de 50% (fenêtre de taille égale à 50% des points). Minimiser la valeur médiane des résidus ou la valeur moyenne sur la moitié des résidus signifie que l'algorithme fera confiance uniquement à la moitié des points. Bien évidemment, certains problèmes possèdent un plus grand nombre de données correctes (ou beaucoup moins). Connaître le nombre de données non contaminées $k = (1 - \xi)n$ va permettre de rechercher à travers les résidus la fraction de k points qui correspond au modèle (les k résidus les plus proches). L'algorithme du LQS

(Least Quantile of Squares) est similaire au LMS mais utilise les résidus d'un sous-ensemble de points de taille k .

L'algorithme LKS (Least k th order Square) utilise aussi le nombre de données non contaminées k , avec $p < k < n$, dans une estimation robuste de type LMS (p étant le nombre de points nécessaires pour définir un modèle). Prenant en compte à la fois les définitions d'implosion et d'explosion du point de rupture, cet estimateur possède un point de rupture théorique de $\min(\frac{k}{n}, 1 - \frac{k}{n})$. On tire aléatoirement un modèle, calcule les résidus r_i correspondant aux données restantes et les classe dans un ordre croissant. On recherche ensuite le mode de distribution qui correspond à trouver la plus petite différence de résidus $\hat{\gamma}_k$ pour k résidus consécutifs :

$$\hat{\gamma}_k = \min_l \gamma_{l,k} \quad \text{pour } l = 1, 2, \dots, (n - p)$$

avec

$$\gamma_{l,k} = \frac{d_{(l+k-1)} - d_l}{2} \quad (5.6)$$

où d_l signifie que l'on utilise le l -ième résidu classé parmi $(n - p)$. La valeur minimum de cette mesure correspond à la solution du LKS.

On peut obtenir maintenant une estimée robuste de l'écart type \hat{s}_k du bruit perturbant les données correctes d'ordre k correspondant par :

$$\hat{s}_k = \frac{\hat{\gamma}_k}{\Phi^{-1}[0.5(1 + k/n + 1)]}$$

où $\Phi^{-1}[\cdot]$ est la fonction de densité cumulative normale des résidus. Cette estimée robuste de l'écart type \hat{s}_k permet d'extraire les points aberrants des données correctes (résidus tels que $d_{i,k} \leq 2.5\hat{s}_k$). On peut estimer ensuite l'écart type robuste non biaisé à partir des données correctes $\hat{\sigma}_k$ et utiliser le RLS pour ré-estimer les paramètres du modèle. Nous renommons les q_k données correctes extraites par cette méthode : $r_{i,k}$. Il est alors possible de déterminer la variance de l'erreur normalisée ϵ^2 évaluée pour les k données :

$$\epsilon^2 = \frac{1}{q_k - p} \sum_{i=1}^{q_k} \left(\frac{r_{i,k}}{\hat{s}_k} \right)^2 = \frac{1}{\hat{s}_k^2} \frac{\sum_{i=1}^{q_k} r_{i,k}^2}{q_k - p} = \frac{\hat{\sigma}_k^2}{\hat{s}_k^2} \quad (5.7)$$

Récemment, dans [LMP98], *K. Lee, P. Meer* et *R. Park* proposent d'estimer la valeur de k , qui était choisie arbitrairement auparavant, à partir de cette mesure. Les auteurs proposent pour cela de minimiser l'erreur normalisée ϵ^2 comme suit :

$$\hat{k} = \min_k \epsilon_k^2$$

Partant d'une valeur de k faible, les auteurs proposent une procédure itérative qu'ils nomment ALKS (Adaptive Least k th order Square) permettant de déterminer \hat{k} .

MUSE : Minimum Unbiased Scale Estimator

Les techniques robustes que nous venons de voir sont toutes capables d'extraire un modèle et ses paramètres à partir d'un ensemble de données contaminées. Ces techniques possèdent toutefois l'inconvénient de ne pas gérer la présence de plusieurs modèles dans leur ensemble de données. Le modèle le plus important sera extrait tandis que tous les points

appartenant aux modèles restants seront considérés comme des données aberrantes. Les algorithmes que nous venons de voir peuvent être améliorés en recherchant un à un tous les modèles présents dans le domaine. *J. Miller* et *C. Stewart* proposent dans [MS96b] un algorithme, MUSE (Minimum Unbiased Scale Estimator), fondé sur le LMS et inspiré des travaux [LMP98] permettant de gérer plusieurs modèles, même lorsque le saut entre deux modèles est très faible ($< 10\sigma$).

Traditionnellement, la technique du LMS est présentée comme la minimisation de la moindre médiane des carrés des résidus. D'un autre point de vue, on pourrait aussi interpréter cette technique comme la minimisation de l'estimée non biaisée de l'écart type $\hat{\sigma}$ (équation 5.3). Pour toutes les hypothèses \mathbf{F} , les auteurs estiment la variance de l'erreur normalisée ϵ (équation 5.7), puis déterminent la valeur non biaisée de l'écart type du k -ième ordre $\hat{\sigma}_k$ pour \mathbf{F} . Ils calculent la valeur de s_k pour toutes les valeurs de k et sélectionnent la plus petite de ces valeurs qui est renommée s_k^* . Ils choisissent ensuite le modèle $\tilde{\mathbf{F}}$ qui minimise la valeur de $\hat{\sigma}_k$. Après suppression de tous les points qui appartiennent au modèle, on cherche s'il existe un autre modèle dans les données. Dans ce cas, la procédure est relancée avec les données restantes. Dans le second cas, on estime l'écart type robuste $\hat{\sigma}$ de toutes les régions trouvées puis on ré-estime les paramètres de chaque modèle par le RLS.

L'algorithme présenté demande un coût en calcul très important. Pour cette raison, et pour utiliser des points ayant une forte probabilité d'appartenir au modèle recherché, les auteurs proposent d'estimer la valeur de s_k en utilisant uniquement les n points qui se situent dans l'espace physique intérieur des p échantillons du tirage aléatoire.

Le RANSAC : RANdom SAmple Consensus

L'algorithme du RANSAC est une procédure exploratoire qui a été introduite par *M. Fishler* et *R. Bolles* dans le domaine du traitement d'images en 1981 [FB81]. C'est une technique très utilisée qui s'applique directement à la reconstruction robuste d'une scène. Plusieurs articles, notamment [MMRK91], [Ste95], [SL95a], ou [Tor97] se sont intéressés à tester, comparer et améliorer cet algorithme. Tout comme les techniques basées sur le LMS, cet algorithme procède à un échantillonnage aléatoire de l'espace des données. Le principe de l'algorithme est le suivant : extraire des modèles à partir des données, déterminer les points qui vérifient ce modèle puis sélectionner les meilleurs. L'avantage de cette procédure est de pouvoir utiliser plusieurs modèles de surface différents pour représenter les données. Pour chaque modèle de surface que l'on va utiliser dans la procédure, on suppose que l'on peut estimer ses paramètres à partir d'un jeu de points minimal p , et quantifier la distance d'une donnée par rapport aux modèles calculés. La recherche des données qui appartiennent au modèle est obtenue par une technique de croissance de région. On sélectionne ensuite l'hypothèse la plus grande, dans le sens où l'on choisit celle qui possède le support le plus important (nombre de points qui composent l'hypothèse). Nous allons maintenant présenter cet algorithme lorsque plusieurs modèles de surface servent à décrire les données.

Soient n l'ensemble des données disponibles et p_l le nombre de points nécessaires pour définir un modèle de type M_l (l décrit la liste des modèles). La procédure est alors la suivante :

Le test d'arrêt de la procédure s'opère de deux manières différentes : en estimant le nombre d'itérations nécessaires pour pouvoir retrouver la structure complète de la scène, ou en attendant d'avoir utilisé chaque modèle, chaque zone ou chaque donnée assez souvent. L'algorithme utilise trois paramètres dans la procédure :

- le domaine de tolérance utilisé pour déterminer si une donnée est compatible ou non

-
- Tirer aléatoirement un type de modèle M_l .
 - Tirer aléatoirement un jeu S de p_l données de l'ensemble des n points.
 - Calculer les paramètres du modèle initial M_l^S fixé par le jeu S .
 - Rechercher dans l'espace $\{m_i\} - S$ les points qui appartiennent au domaine de tolérance du modèle. On obtient un nouveau support S^* du modèle.
 - Tester si la taille du support (nombre de points dans S^*) du modèle initial M_l^S est suffisante : $(S^*) \geq t$.
 - Si le test est *Vrai*, le modèle M_l^S est validé, retour au début.
 - Si le test est *Faux*, le modèle M_l^S est rejeté, retour au début.
-

FIG. 5.3 – Procédure du RANSAC

avec le modèle fixé. On peut définir ce domaine à partir d'une estimée de la variance de l'erreur.

- le critère d'arrêt de l'algorithme,
- le seuil t utilisé pour déterminer si un modèle est représentatif.

L'article original de *M. Fishler* et *R. Bolles* [FB81] propose deux améliorations. La première est de restreindre le processus d'échantillonnage dans des zones définies pour augmenter la probabilité d'appartenance des points initiaux à un modèle unique. Ils proposent aussi de remplacer le processus aléatoire par une méthode déterministe. La seconde amélioration proposée est de ré-estimer les paramètres du modèle après chaque ajout de point au support, ceci pour rendre le modèle plus cohérent. *M. Stricker* et *A. Leonardis* utilisent cette procédure et proposent dans [SL95a] de conduire la recherche aléatoire par des vecteurs de recensement.

Le MLESAC : Maximum Likelihood Error SAmples Consensus Le RANSAC de *M. Fishler* et *R. Bolles* sélectionne le ou les modèles selon la taille de leur support. *P. Torr* et *A. Zisserman* choisissent dans [TZ00] de sélectionner la solution selon un critère probabiliste. Les auteurs proposent dans un premier temps d'utiliser un M-estimateur pour évaluer le domaine de tolérance (MSAC, M-estimator Sample Consensus). Ils présentent ensuite une sélection par maximisation de la vraisemblance, le MLESAC, en utilisant l'algorithme EM (Expectation-Maximisation).

Le RESC : RESidu Consensus *X. Yu*, *T. Bui* et *A. Krzyzak* adaptent dans [YBK94] la procédure du RANSAC pour définir la procédure RESC : RESidu Consensus. Ils proposent une méthode permettant de séparer les données justes des données aberrantes des modèles à partir de leurs résidus. Pour un modèle défini, ils calculent les résidus, comparent leur histogramme pour le faire correspondre à un modèle de distribution, déterminent l'écart type puis trient les données. L'avantage de leur méthode est de pouvoir adapter la taille de l'intervalle de tolérance des modèles au bruit présent dans les données.

Le MINPRAN : MINimize the Probability of RANdomness

C. Stewart propose dans [Ste95] un nouvel algorithme robuste. L'idée est de considérer le modèle comme l'ensemble de points qui minimise la probabilité de dispersion aléatoire de l'ensemble des données. Sous l'hypothèse que les données aberrantes possèdent une dispersion uniforme, *C. Stewart* écrit la probabilité P qu'au moins $n_{\mathbf{F},r}$ points tirés aléatoirement parmi n appartiennent à l'espace défini par $\mathbf{F} \pm r$, \mathbf{F} étant un modèle et r son intervalle de tolérance (les données justes). Dans le cas d'un modèle de droite et d'un espace bidimensionnel (x,z) , il obtient la fonction de probabilité P suivante :

$$\mathbf{P}(r, n_{\mathbf{F},r}, n) = \sum_{i=n_{\mathbf{F},r}}^n \binom{n}{i} \left(\frac{r}{Z_0}\right)^i \left(1 - \frac{r}{Z_0}\right)^{n-i}$$

avec

$$Z_0 = \frac{(Z_{max} - Z_{min})}{2}$$

Z_{max} et Z_{min} étant les bornes de l'espace. Pour chaque hypothèse de modèle \mathbf{F} , la probabilité $\mathbf{P}(r, n_{\mathbf{F},r}, n)$ nous donne alors la probabilité qu'il puisse y avoir au moins $n_{\mathbf{F},r}$ points corrects dans l'intervalle de tolérance défini par \mathbf{F} et r si les n points sont distribués aléatoirement. Cette probabilité P mesure le *degré de hasard* d'un modèle traité. Il utilise cette probabilité P pour définir le domaine de tolérance (défini par r) et choisir les points qui appartiennent au modèle. Il minimise la fonction d'objectif suivante :

$$H(\mathbf{F}, n) = \min_r \mathbf{P}(r, n_{\mathbf{F},r}, n)$$

où $H(\mathbf{F}, n)$ représente la *probabilité de hasard* du modèle \mathbf{F} . L'auteur propose d'utiliser une technique d'échantillonnage aléatoire semblable à celle du LMS (guidée par une technique de Monte-Carlo) pour construire les modèles \mathbf{F}_j , puis de sélectionner le modèle qui minimise la *probabilité de hasard* comme suit :

$$\hat{\mathbf{F}} = \arg \min_j H(\mathbf{F}_j, n) = \min_j \left(\min_r \mathbf{P}(r, n_{\mathbf{F}_j, r}, n) \right)$$

5.3.4 La transformée de Hough

La transformation de *Hough* est l'une des techniques robustes les plus anciennes et fascinantes. Le principe est d'utiliser l'espace discrétisé des paramètres pour retrouver le modèle. Chaque sous-ensemble de points définit un modèle et vote pour lui dans l'espace des paramètres. Les régions de l'espace dans lesquelles les votes s'accumulent correspondent aux modèles cherchés. Le principal inconvénient de la méthode est son coût lorsque le nombre de paramètres augmente. Il faut aussi disposer d'un nombre important de données pour obtenir une convergence nette (un pic d'accumulation important). Dans des conditions de données bruitées, cette convergence est souvent bien difficile à déterminer. Cette méthode fait partie intégrante des techniques robustes puisqu'elle permet de supprimer l'influence des données aberrantes dans l'estimation. Elle est rarement appliquée pour résoudre des problèmes ayant plus de trois inconnues.

5.4 Sélection de modèles

Dans la section précédente nous supposons connu le modèle d'objet à utiliser pour un jeu de données tiré aléatoirement. Dans certaines applications, plusieurs modèles d'objets simples peuvent représenter nos données. Dans ce cas, il va falloir être capable non seulement de comparer deux modèles d'un même objet mais aussi des modèles correspondant à des objets différents (lignes, courbes, plans, ellipses, etc...). Ce problème est appelé la **sélection de modèles**. L'objectif général est de pouvoir décider automatiquement quel modèle parmi ceux possibles on doit utiliser. Choisir la meilleure fonction, ou le meilleur modèle, pour décrire un jeu de données est un problème qui a été traité en statistique. La sélection de modèle n'est apparue que récemment dans des domaines comme la vision par ordinateur, le traitement d'images, la segmentation, la reconnaissance d'objets ou les problèmes inverses.

5.4.1 Sélection de modèles pour la segmentation de données tridimensionnelles

Pour une application comme la reconstruction tridimensionnelle d'une scène 3D contaminée par des points aberrants, on est directement confronté à ce type de problème. Nous cherchons le meilleur jeu de modèles permettant de décrire les données. La plupart des approches sont basées sur le RANSAC comme *ExSel++* [SL95a]. On utilise tout d'abord un algorithme d'exploration de données pour construire une liste suffisante de modèles correspondant à toutes sortes d'objets. Il faut ensuite procéder au tri de ces modèles, sélectionner les modèles les plus justes et représentatifs de l'ensemble des données. On est alors face à un choix délicat. D'une part, on veut construire avec les données un maximum de modèles pour être certain de posséder le bon modèle dans la liste. D'autre part, plus il y aura de modèles dans la liste, plus la sélection sera longue et difficile. Il faut donc trouver un compromis judicieux qui peut provenir des connaissances a priori sur la scène à modéliser. Dans le cas où l'on ne dispose que d'un seul modèle d'objet pour décrire les données, dans notre cas des morceaux de plans pour construire un objet polyédrique, la sélection de modèle est encore nécessaire pour déterminer le nombre de plans utiles à la description de l'objet.

5.4.2 Les différents critères et mesures de qualité

Tout d'abord, il faut savoir évaluer la qualité d'un modèle. Nous devons déterminer si le modèle est représentatif de son support ou si l'estimation des paramètres n'a que peu de sens. On rencontre dans la littérature plusieurs critères. Un premier critère possible est le nombre de points dans le support. Lorsque celui-ci est suffisamment important, le modèle est validé comme le fait l'algorithme du RANSAC. Un autre critère qui est couramment utilisé est la variance des résidus des points du support. Ce critère est important lorsque les paramètres du modèle sont évalués avec un estimateur non robuste. Lorsque des données aberrantes sont incluses dans le support du modèle, celui-ci est altéré et sera rejeté. Pour comparer des modèles d'objets différents, on utilise des critères supplémentaires liés à la complexité des objets : nombre de paramètres nécessaires pour décrire le modèle ou degrés de liberté de celui-ci. Lorsque nous possédons des connaissances sur la scène, objet polyédrique, convexe ... nous pouvons les utiliser et privilégier les modèles les plus probables.

Un grand nombre de mesures ont été présentées pour comparer et sélectionner les modèles : critères basés sur des seuils (nombre de points, variance ...), sur des tests du Chi-2, règle de Bayes, Longueur de Description Minimum (MDL), distance de Kullback-

Liebler, RUNS-test (séquences consécutives de résidus de même signe) ... Peu d'articles [BS98, Tor99] sont consacrés à la comparaison de ces mesures en vision par ordinateur. *P. Torr* étudie et compare quelques-unes de ces mesures dans [Tor99] pour l'estimation de la géométrie entre deux vues. Il obtient de meilleurs résultats avec les mesures Bayésiennes GBIC et GRIC.

L'étude réalisée par *K. Bubna* et *C. Stewart* [BS98] porte sur plusieurs points : présenter, regrouper et énumérer toutes les mesures employées pour la sélection de modèles dans le domaine de la vision, tenant compte du fait que chacune de ces mesures possède ses propres conditions d'utilisation (bruit connu ou non, gaussien ou autre). Les auteurs adaptent et proposent les critères définis dans le but d'obtenir un ensemble de critères plus homogène. Ils proposent d'utiliser la technique d'estimation par **Bootstrap** pour connaître la distribution du bruit présent dans les données. Ils poursuivent ensuite par l'étude et la comparaison des critères définis. Leurs études portent sur deux classes de tests : l'aptitude à choisir le bon modèle pour un jeu de données fixé (**sélection de modèles**) et leur aptitude de discrimination lorsque deux modèles différents sont proches (**fusion des modèles**). Les deux tests sont réalisés à la fois sur des données expérimentales et des données réelles. Les résultats qu'ils obtiennent tendent à préférer le critère Bayésien qui donne de meilleurs résultats dans la majorité des tests effectués. Le critère d'*Akaike*, l'AIC, qui est similaire à celui du MDL semblerait être moins efficace. Il est à noter que le critère proposé par *Kanatani* [Kan98], le GIC, n'est pas utilisé dans cette étude comparative.

Trouver *la bonne mesure* qui permettra de sélectionner et de comparer les différents modèles extraits dans la phase exploratoire est empirique. Les mesures du AIC et du MDL sont les plus utilisées. Pour une application comme la modélisation 3D, la mesure doit être capable de tolérer à la fois les influences du bruit présent dans les données et surtout celles des données aberrantes, sinon la sélection sera inefficace. La sélection doit valider un modèle selon les erreurs engendrées par son utilisation, la complexité de celui-ci et l'importance que lui procure la taille de son support.

5.5 L'algorithme de segmentation de MNE

Nous allons présenter maintenant l'algorithme de segmentation de MNE en plans utilisé dans cette thèse. Nous avons présenté ces travaux dans les publications suivantes : [VBD97] et [VDM00]. L'objectif est d'extraire une représentation polyédrique simple et représentative de tous les bâtiments de la scène sans aucune connaissance sur leur forme (si ce n'est qu'ils sont composés de facettes planes). C'est le problème de segmentation que nous voulons résoudre. En utilisant uniquement le MNE comme donnée initiale, ce problème peut être interprété comme la modélisation d'un nuage de points 3D altéré par un bruit non gaussien. Nous avons vu dans les sections précédentes que ce problème peut être résolu par l'utilisation de techniques robustes et de sélection de modèles. La segmentation en plans proposée dans cette section est basée sur ces techniques pour éviter l'influence, néfaste sur la reconstruction du modèle, des données aberrantes présentes en quantité importante dans le MNE.

Remarque : La segmentation issue de l'algorithme que nous présentons n'est pas une segmentation au sens strict du terme car certains points (les données aberrantes la plupart du temps) sont affectés à aucun plan. Dans le chapitre 6, nous présentons un pré-traitement qui permet l'affectation de tous ces points.

5.5.1 Stratégie générale de la segmentation

Notre approche est basée sur l'algorithme ExSel++ présenté par *M. Stricker* et *A. Leonardis* dans [SL95a]. Les auteurs définissent dans cet article une méthodologie générale et robuste pour extraire des modèles paramétriques à partir de données denses ou éparses. Une des spécificités de leur méthodologie est sa capacité à utiliser et sélectionner plusieurs modèles d'objets différents pour décrire les données.

Le MNE initial est une carte $2D^{\frac{1}{2}}$. Les données de cette carte correspondent principalement aux toits des bâtiments et au sol de la ville. Nous avons choisi le modèle de morceau de surface plan pour décrire les données. Nous serons ainsi capable de reconstruire un modèle polyédrique simple de chaque bâtiment, excepté pour les quelques surfaces du second ordre qui peuvent être présentes dans la structure du bâtiment (dôme, surface cylindrique . . .). Le processus de segmentation que nous proposons est composé de trois étapes principales que nous présentons indépendamment : une étape d'exploration des données qui génère une liste d'hypothèses de modèles, une étape de fusion qui supprime la redondance des hypothèses et une étape de sélection qui choisit le meilleur jeu de modèles pour décrire les données.

Comme pour l'algorithme de reconstruction automatique de façades, nous appliquons le processus de segmentation sur le voisinage d'un bâtiment pour garder un faible temps de calcul. Nous fusionnons ensuite les résultats obtenus sur chacun des bâtiments pour obtenir la modélisation finale de la scène. Nous détectons et extrayons automatiquement du MNE une zone d'intérêt ($ROI_{\text{bât}}^{2.5D}$) sur chaque bâtiment ou groupe de bâtiments adjacents (section 4.5.2).

5.5.2 Exploration des données

L'objectif de cette étape est de produire une liste d'hypothèses qui seront ensuite triées par l'étape de sélection pour décrire les données. Toutes les différentes parties du modèle final du bâtiment doivent être trouvées dans cette étape. L'étape d'exploration est basée sur l'algorithme du RANSAC (section 5.3.3) qui a été proposé dans [FB81]. Comme l'algorithme de la moindre médiane des carrés (LMS), celui-ci calcule un modèle en résolvant un système d'équations défini pour un jeu de points tirés aléatoirement des données. Toutes les données sont ensuite classées par rapport à ce modèle et les points qui appartiennent à l'intervalle de tolérance de l'erreur sont conservés et appelés **jeu de consensus du modèle**. Si la taille du support de ce jeu de consensus dépasse un seuil défini, le modèle est validé puis est réévalué par un estimateur non biaisé. Nous allons adapter cette procédure pour chercher et calculer les différentes hypothèses de modèles qui permettront de décrire les différentes parties des données.

La procédure exploratoire est itérative et chaque étape peut être décrite comme suit :

- sélectionner aléatoirement un jeu de points minimal pour estimer le modèle initial,
- accroître ce jeu de points avec des données consistantes et rejeter les points invalides,
- tester la validité de l'hypothèse de modèle.

Avec un simple modèle de morceau de surface plan, le jeu de points minimal pour construire un plan est défini par trois points non alignés.

Nous implémentons deux améliorations additionnelles pour conduire la procédure et augmenter la consistance des hypothèses construites. Tout d'abord, nous choisissons une procédure de croissance de région déterministe dont les graines sont les points du jeu minimal pour trouver les points du **jeu de consensus du modèle**. En réalité, seul le premier point du jeu minimal est tiré aléatoirement dans la scène. Les deux autres sont choisis dans une fenêtre

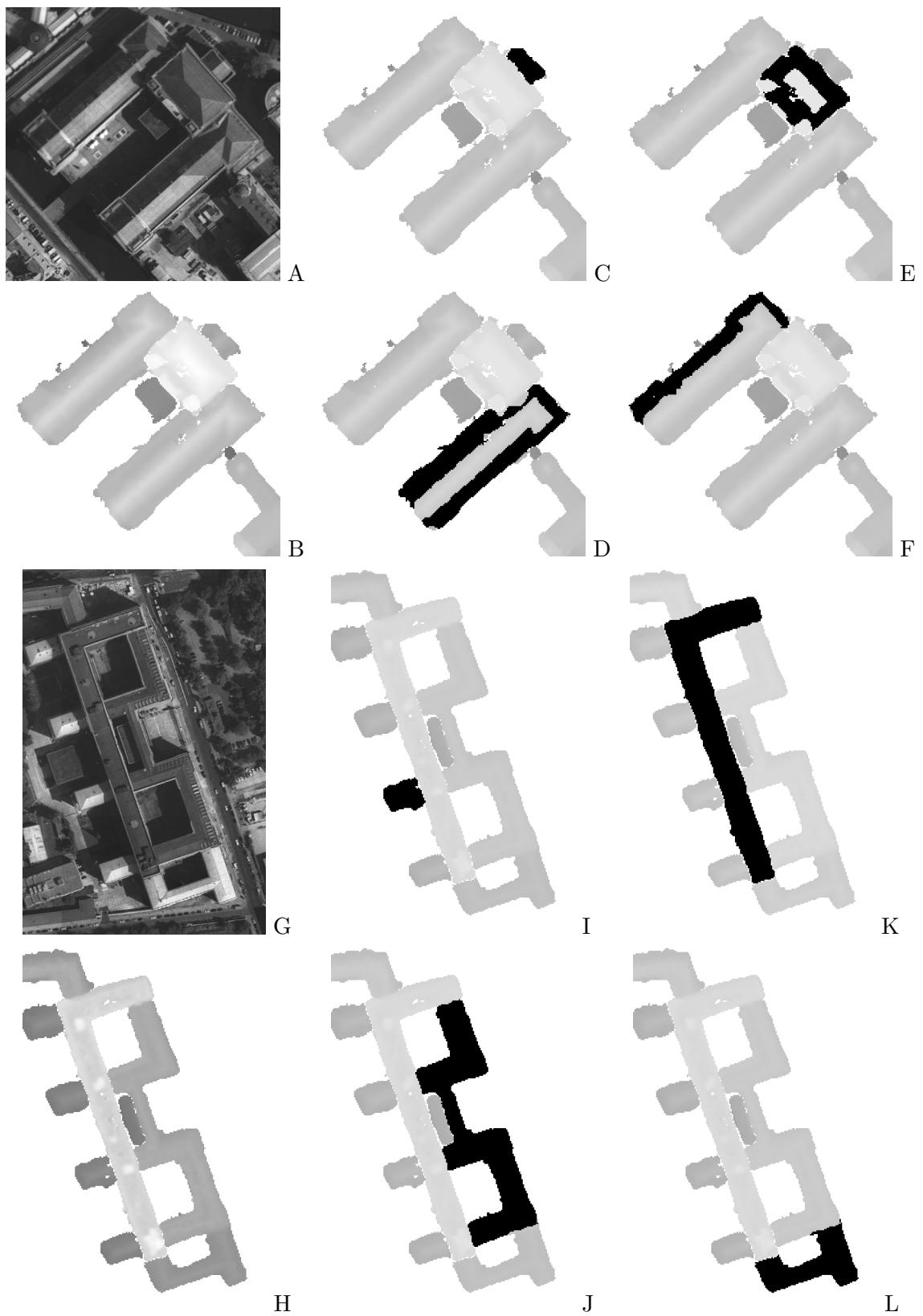


FIG. 5.4 – Résultats de l'étape d'exploration des données : A et G sont les deux ortho-images de deux bâtiments traités et B et H les MNE correspondants. Les régions noires des autres images sont des exemples d'hypothèses de modèles de plan qui ont été extraites des MNE initiaux dans l'étape d'exploration des données.

centrée sur le premier point. Nous validons les trois points initiaux par deux vérifications : nous vérifions qu'ils ne sont pas alignés et nous vérifions qu'ils ne correspondent pas à un plan proche de la verticale. Nous tirons aussi avantage de la carte $2D^{\frac{1}{2}}$ en guidant l'échantillonnage aléatoire et la croissance de région.

La seconde amélioration est l'utilisation d'une carte de recensement pour conduire l'exploration de la scène. Lorsque nous obtenons une hypothèse valide de modèle, nous la stockons dans une carte de recensement pendant un nombre fini d'itérations de la procédure. Les valeurs de la carte de recensement sont décrémenteés après chaque échantillonnage aléatoire même lorsque le modèle obtenu ne correspond pas à une hypothèse valide. Le tirage aléatoire des points du jeu initial ne peut pas prendre les points qui sont recensés dans cette carte.

Nous avons développé deux modes d'exploration pour nos expériences. Dans le premier mode, les hypothèses de modèle sont contraintes à être des plans horizontaux. Dans ce cas, nous calculons simplement la valeur médiane des altitudes des points du modèle pour estimer les paramètres du morceau de surface plan. Dans le second mode, nous n'imposons aucune contrainte sur l'orientation du morceau de surface plan. Nous utilisons un estimateur des moindres carrés linéaire pour calculer les paramètres du morceau de surface plan. Ce second mode nous permet de retrouver les différentes parties de toute sorte de toit. Pour assurer une convergence rapide de l'estimateur, la première estimée des paramètres est contrainte à être horizontale.

Nous utilisons deux conditions d'arrêt pour la procédure d'exploration. Tout d'abord, la procédure est stoppée quand nous estimons avoir suffisamment d'hypothèses de modèle pour retrouver toutes les parties du bâtiment. En second, la procédure est aussi stoppée lorsqu'elle ne trouve plus d'hypothèses dans les données (les points qui ne sont pas dans la carte de recensement). La première condition d'arrêt dépend principalement de la complexité de la scène. Nous évaluons le nombre d'hypothèses à chercher à 50 modèles. La seconde condition d'arrêt est choisie empiriquement à 50 tirages aléatoires sans succès. Le seuil définissant l'intervalle de tolérance d'erreur des hypothèses de modèle dépend de la résolution du MNE et du mode utilisé. Nous en discuterons dans la section 5.5.5. Des exemples de résultats sont présentés dans la figure 5.4. Toutes les parties du modèle ont été retrouvées.

5.5.3 Fusion des hypothèses

Après l'étape d'exploration, nous proposons d'utiliser une étape de fusion des hypothèses dans le processus de segmentation. Cette étape permet de limiter la redondance dans la liste des hypothèses et de diminuer les temps de calcul de l'étape de sélection. Nous avons constaté que les hypothèses conservées après cette étape sont aussi généralement plus cohérentes. Le principe de cette étape est de fusionner deux hypothèses lorsqu'elles ont une surface commune importante ou lorsqu'il y a une haute probabilité qu'elles correspondent à la même surface.

L'implémentation du premier test conditionnel est relativement simple. Nous estimons la taille de la surface commune en utilisant le nombre de points en commun des deux plans. Nous exprimons cette mesure par le taux de surface commune dans le plan qui possède la plus petite surface (le plus petit nombre de points). L'utilisation de ce taux de surface permet d'éviter la confusion entre deux plans qui correspondent à la même surface et une intersection de deux plans différents. Nous utilisons un seuil de décision de 80% pour déterminer si les deux plans doivent être fusionnés.

Le second test conditionnel pour la fusion est basé sur un test statistique : le F-TEST.

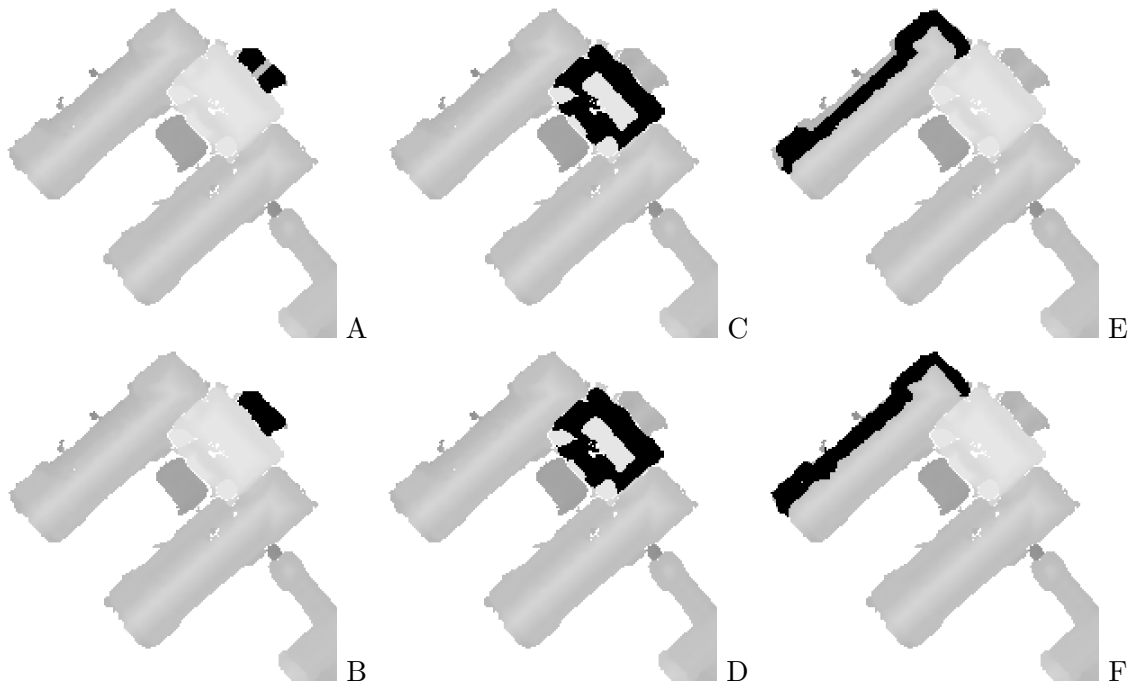


FIG. 5.5 – *Fusion des hypothèses* : Chacune des différentes paires d'images (A B), (C D) et (E F) présente deux hypothèses extraites par l'algorithme d'exploration (régions noires des images des bâtiments) qui correspondent à un même plan. Ces hypothèses sont fusionnées.

Ce test est utilisé uniquement avec le mode de recherche des plans sans contrainte. Nous l'utilisons pour décider si le plan résultant de la combinaison des deux hypothèses de plans est meilleur que chacun d'entre eux dans le sens du test. L'estimateur des paramètres du plan est non biaisé. Nous utilisons la variance de l'erreur des points par rapport à leur modèle pour quantifier la qualité de chacune des trois hypothèses (les deux plans et leur combinaison).

Le F-TEST permet de comparer les variances de deux échantillons de données en maximisant le rejet du cas où ces deux variances sont équivalentes. L'avantage de ce test statistique est de pouvoir comparer les variances de deux échantillons de données qui ont des tailles différentes. Nous utilisons le F-TEST pour comparer individuellement la qualité du plan combiné avec la qualité de chacun des deux plans initiaux. Nous calculons les probabilités que le plan combiné soit meilleur que chacun des deux plans initiaux. Si ces deux probabilités sont supérieures à 0,9, nous fusionnons les deux plans. La figure 5.5 présente trois exemples où les hypothèses sont fusionnées.

5.5.4 Sélection des hypothèses

L'objectif de l'étape de sélection est de décider quelles hypothèses doivent être gardées pour décrire les données. Nous voulons éliminer la partie aléatoire introduite lors de l'étape d'exploration. Nous voulons aussi sélectionner le meilleur jeu d'hypothèses de modèle tout en choisissant le jeu minimum nécessaire pour décrire les données. L'étape de sélection est réalisée en transformant le problème de sélection en un problème d'optimisation. Nous adoptons la solution proposée par A. Leonardis dans [Leo94] qui implémente le principe du

MDL (*Minimum Description Length*).

Les fonctions de longueur de description

Le principe du MDL est basé sur la notion de résolution par codage minimal. Dans le problème de sélection, nous voulons sélectionner le nombre minimal de modèles qui sont nécessaires pour décrire les données. Nous voulons aussi assurer une mesure minimale de l'erreur entre les données et les modèles sélectionnés. Nous utilisons une fonction d'objectif F , calculée pour un sous-ensemble d'hypothèses extraites de la liste de modèles \mathbf{L} . Cette fonction est une combinaison de deux composantes: la première composante $\mathbf{Q}(M_i)$ qui exprime la valeur de bénéfice pour un modèle particulier M_i de la liste, et une seconde composante $\mathbf{I}(M_i, M_j)$ qui exprime la valeur de coût de l'interaction entre les modèles M_i et M_j .

La composante $\mathbf{Q}(M_i)$ exprime la qualité d'une hypothèse de la liste. Cette mesure de qualité est composée de deux termes: un terme de bénéfice et un terme de coût. Le terme de bénéfice exprime l'importance d'une hypothèse. Il dépend de la taille n_i du support D_i du modèle M_i (le nombre de données utilisées pour construire cette hypothèse: $n_i = |D_i|$). Le terme de coût exprime la qualité de description de l'hypothèse. Il dépend de la mesure d'erreur Σ_i des données qui supportent le modèle M_i (Σ_i est la somme des résidus). Nous calculons la composante de qualité $\mathbf{Q}(M_i)$ comme suit :

$$\mathbf{Q}(M_i) = K_1 \cdot n_i - (1 - K_1) \cdot \Sigma_i \quad (5.8)$$

avec $K_1 \in [0,1]$. Nous obtenons une composante plus simple que celle de *A. Leonardis* puisque nous n'utilisons qu'une seule sorte de modèle d'objet. K_1 est une pondération qui nous permet d'ajuster la préférence de l'un des deux termes. La composante de qualité permet de sélectionner les modèles de la liste qui ont un grand support et une faible mesure d'erreur.

Parce que deux hypothèses différentes de modèles peuvent se chevaucher, nous devons prendre en compte cette interaction dans la procédure d'optimisation. Nous voulons limiter ce phénomène. Comme *A. Leonardis*, nous considérons seulement le chevauchement entre deux modèles. Nous utilisons une composante d'interaction entre deux modèles $\mathbf{I}(M_i, M_j)$ qui a la même forme que la composante de qualité. Cette composante est évaluée sur une partie de chevauchement des deux hypothèses de modèles. Mais parce que nous voulons un partie de chevauchement minimale entre les modèles, les termes de la composante d'interaction sont opposés aux termes de la composante de qualité. Nous calculons la composante d'interaction $\mathbf{I}(M_i, M_j)$ comme suit :

$$\mathbf{I}(M_i, M_j) = \frac{(-K_1 \cdot |D_i \cap D_j| + (1 - K_1) \cdot \Sigma_{ij})}{2} \quad (5.9)$$

avec :

$$\Sigma_{ij} = \max \left(\sum_{x \in R_i \cap R_j} d(x, M_i), \sum_{x \in R_i \cap R_j} d(x, M_j) \right)$$

d est la distance Euclidienne entre un point x et un modèle M_i . La composante d'interaction permet de limiter les chevauchements des modèles dans les sous-ensembles d'hypothèses que nous évaluons.

Le problème d'optimisation booléenne

Chaque hypothèse de modèle de la liste \mathbf{L} doit être sélectionnée ou rejetée lors de cette étape, c'est un problème d'optimisation booléenne. Le nombre M d'hypothèses de la liste correspond à la dimension du problème. Considérons un vecteur $m^T = [m_1, m_2, \dots, m_M]$ correspondant à un jeu d'hypothèses. m_i est une variable booléenne qui exprime la présence ($m_i = 1$) ou non ($m_i = 0$) d'un modèle M_i dans la solution m^T . La longueur de description \mathbf{F} du jeu de modèles \tilde{m} est définie de la manière suivante :

$$\begin{aligned} \mathbf{F}(\tilde{m}) &= \sum_{M_i \in \{L\}} m_i \cdot \mathbf{Q}(M_i) \\ &+ \sum_{M_i \in \{L\}, M_j \in \{M - M_i\}} m_i \cdot m_j \cdot \mathbf{I}(M_i, M_j) \end{aligned} \quad (5.10)$$

\mathbf{F} doit être maximisé pour trouver le meilleur jeu de modèles. Nous résolvons ceci comme un problème quadratique booléen. La fonction d'objectif est maximisée comme suit :

$$\mathbf{F}(m) = m^T R m \quad (5.11)$$

Cette fonction permet de prendre en compte la qualité de chaque modèle et l'interaction entre tous ces modèles (par paires). Les termes de la diagonale de la matrice R expriment la valeur de coût-bénéfice pour un modèle particulier M_i , nous prenons la composante de qualité $\mathbf{Q}(M_i)$. Les termes en dehors de la diagonale (i, j) représentent l'interaction du chevauchement des deux modèles M_i et M_j , nous prenons la composante d'interaction $\mathbf{I}(M_i, M_j)$. La matrice R est symétrique. Parce que les termes diagonaux dépendent uniquement du chevauchement entre deux modèles, ils peuvent être nuls et la matrice sera alors creuse ou remplie par blocs.

Optimisation par recherche Tabou

Pour résoudre ce problème d'optimisation booléen, nous avons besoin d'une procédure d'optimisation discrète. Nous utilisons la procédure de **recherche par tabou** pour résoudre le système [SL95b]. La recherche par tabou est une procédure heuristique générale pour l'optimisation globale qui peut être interprétée comme une extension de la méthode de plus grande pente (*steepest ascent method*). L'idée de la recherche par tabou est de lancer la procédure de résolution itérative à partir de n conditions initiales. L'algorithme effectue une itération à chaque lancement de la procédure avec une nouvelle condition initiale. Chaque condition initiale permet de trouver une solution locale. L'utilisation de multiples conditions initiales permet à l'algorithme de recherche par tabou d'obtenir la solution globale du système, c'est la meilleure de toutes les solutions locales obtenues.

La procédure lancée pour chacune des conditions initiales cherche un jeu d'hypothèses de modèle correspondant à une solution locale de notre système booléen. Cette solution est approchée itérativement. Nous appelons "mouvements" les évolutions du vecteur d'hypothèses sélectionnées m . A chaque mouvement, nous évaluons la fonction d'objectif pour le jeu de modèles sélectionnés courant ainsi que pour son voisinage, puis nous choisissons le meilleur mouvement. Ce voisinage ne doit pas être trop grand car on doit évaluer $F(\tilde{m})$ pour tout ce voisinage. Il ne doit pas être trop petit non plus car nous voulons pouvoir atteindre toutes les valeurs discrètes en un petit nombre de mouvements. Pour cette optimisation booléenne, nous définissons le voisinage $\mathbf{V}(\tilde{m})$ de v points correspondant à tous les jeux de modèles

```

Initialisation de la Mémoire des mouvements attribués
Initialisation de la Table des fréquences
Initialisation de la Table des maxima
Pour tous  $\tilde{m}_{o_i}$  tel que  $i = \{0,1, \dots, n\}$ 
  -  $\tilde{m}_c = \tilde{m}_{o_i}$ 
  Tant que critère d'arrêt non rencontré
    - Pour tous  $\tilde{m}^*$  appartenant au voisinage  $\mathbf{V}(\tilde{m}_c)$ ,
      évaluer  $\mathbf{F}(\tilde{m}_c)$ 
      - Si  $\mathbf{F}(\text{meilleur } \tilde{m}^*)$  rencontre le critère d'aspiration
         $\tilde{m}_c = \text{meilleur } \tilde{m}^*$ 
      sinon
      - Si  $\mathbf{F}(\text{meilleur non-tabou } \tilde{m}^*) < F(\tilde{m}_c)$ 
         $\tilde{m}_c = \text{meilleur non-tabou } \tilde{m}^*$ 
      sinon
        ajouter  $\tilde{m}_c$  et  $\mathbf{F}(\tilde{m}_c)$  dans la Table des maxima
         $\tilde{m}_c = \text{meilleur } \tilde{m}^*$  selon la Stratégie de diversification
      fin Si
    fin Si
  fin Pour
  - mise à jour de la Mémoire des mouvements attribués
  - mise à jour de la Table des fréquences
fin Tant que
fin Pour
Solution finale = Solution correspondant à la plus grande valeur de coût de
la Table des maxima

```

FIG. 5.6 – Recherche par Tabou

\tilde{m}^* tels que $|\tilde{m}^*| = |\tilde{m}_c| \pm 1$ avec $|\tilde{m}_c| = \text{card}(\tilde{m}_c)$. L'algorithme va ensuite se déplacer de manière "intelligente" dans l'espace des solutions en s'interdisant de revenir en arrière et en cherchant tous les maxima locaux. La figure 5.6 nous présente la procédure d'optimisation. L'algorithme converge généralement en quatre ou cinq itérations. La procédure est définie avec les quatre composantes suivantes :

Critère d'aspiration Lorsque la plus petite valeur de la fonction d'objectif estimée sur le voisinage $F(\tilde{m}_c)$ n'est pas un maximum local, le point courant $F(\tilde{m}_c)$ prend cette position pour se déplacer vers le maximum auquel elle nous guide. C'est le critère d'aspiration vers le maximum local.

Mémoire des mouvements attribués Cette mémoire des jeux de modèles déjà évalués permet d'implanter la fonction *tabou*. Cette fonction est utilisée pour éviter les retours en arrière vers des maxima locaux déjà trouvés. C'est une mémoire à court terme. On interdit ces jeux de modèles pendant un nombre de mouvements τ , la teneur Tabou.

Stratégie de diversification On utilise une Table des fréquences pour comptabiliser les fréquences de mouvements utilisés. On souhaite donner plus de force aux mouvements les moins utilisés et ainsi permettre l'exploration de tout l'espace de recherche. Lorsque tous les mouvements qui augmentent la valeur de notre fonction d'objectif sont tabou, nous diversifions la recherche en utilisant cette table.

Table des maxima Cette table sert à conserver la totalité des points qui correspondent à des maxima. Ainsi, lorsque le critère d'arrêt est rencontré, on récupère dans ce tableau notre solution optimale qui correspond à la plus grande valeur des maxima locaux.

Critère d'arrêt Après chaque mouvement, toutes les mémoires sont mises à jour. La recherche peut être arrêtée soit après un nombre fixé d'itérations, soit lorsqu'on ne trouve pas de meilleure solution après un nombre défini d'itérations.

5.5.5 Résultats de segmentation

Les résultats de l'étape de segmentation sont présentés dans les figures 5.7 et 5.8. Les MNE correspondant aux bâtiments ont une résolution de $50cm$, ainsi que les ortho-images. La première figure présente les plans extraits par l'algorithme de segmentation sur un bâtiment avec des toits à deux pans. *A* présente l'ortho-image du bâtiment et *B* le MNE correspondant. L'étape d'exploration construit 50 hypothèses de plans à partir des points 3D sur le toit du bâtiment. L'intervalle de tolérance de l'erreur utilise un seuil de $\pm 2m$. L'étape de fusion regroupe 21 plans. L'étape de sélection ne conserve que 22 plans pour décrire le bâtiment. Pour visualiser le résultat, nous reconstruisons un MNE synthétique à partir de tous les plans sélectionnés. Nous pouvons voir dans *C* que les différentes parties du toit sont retrouvées et reconstruites correctement.

La figure 5.8 présente les plans extraits par l'algorithme de segmentation sur trois bâtiments dans le mode avec contrainte horizontale. *A* présente les ortho-images des $ROI_{\text{bât}}^{2.5D}$ des bâtiments et *B* les MNE correspondants. L'intervalle de tolérance de l'erreur utilise un seuil de $\pm 4m$. Ce seuil est plus lâche pour compenser les contraintes d'horizontalité imposées sur les plans. Nous pouvons retrouver aussi les toits en pente comme pour le bâtiment étudié dans la figure 5.7. Les résultats sont présentés dans *C*, les différentes parties du toit sont retrouvées et reconstruites correctement. L'étape d'exploration construit 26 hypothèses de plans à partir des points 3D sur le toit du bâtiment. L'étape de fusion regroupe 14 plans. L'étape de sélection ne conserve que 11 plans pour décrire le bâtiment.

Programmation et paramètres importants La totalité des algorithmes a été développée au cours de la thèse. L'étape de segmentation se compose de quatre programmes développés en C++ : extraction automatique des bâtiments, recherche des modèles en utilisant l'algorithme du RANSAC, fusion des hypothèses redondantes et sélection des plans utiles pour la description du modèle. Une librairie a été développée pour regrouper certaines fonctions communes : les estimateurs, opérations diverses entre les plans, interface I/O (VRML et OGL pour Geomview), ...

Le choix du seuil de l'intervalle de tolérance est un paramètre essentiel. Nous avons choisi ce seuil de manière empirique, privilégiant les principales structures du toit. Sa valeur doit être choisie selon la résolution du MNE, la précision des données qui le composent et le niveau de détail que nous souhaitons reconstruire. Le nombre d'hypothèses construites pour réaliser la liste est de 50. Ce nombre est suffisant pour extraire les principales parties du

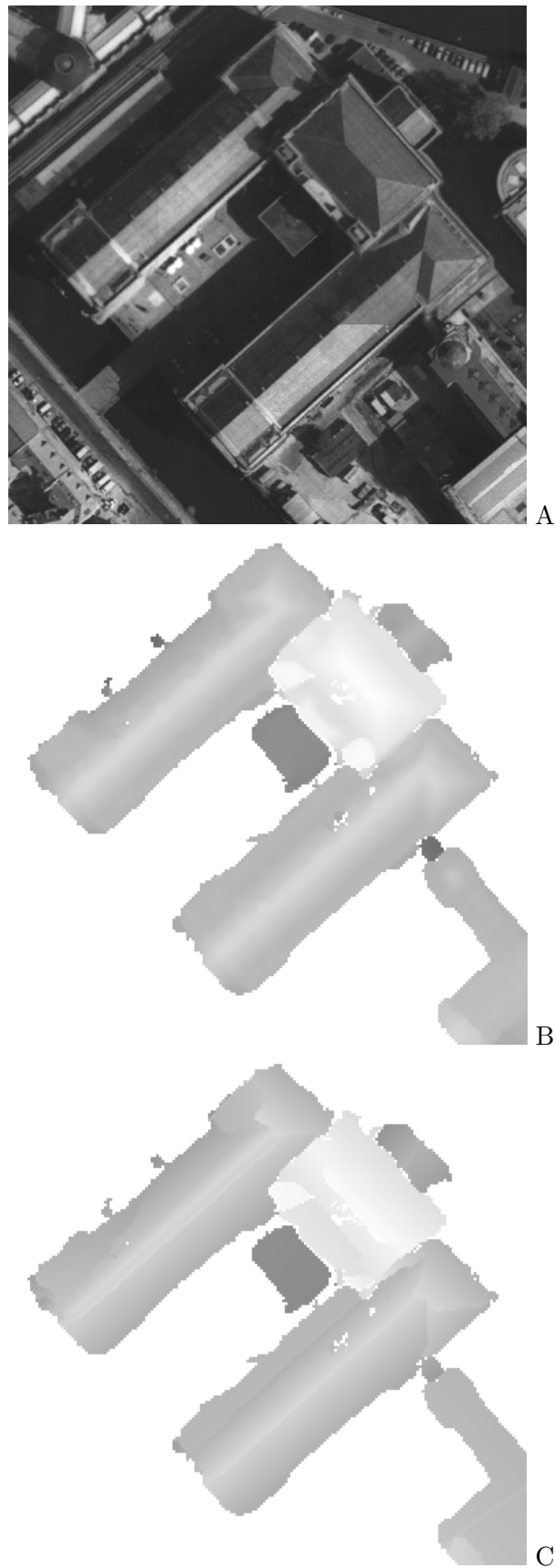


FIG. 5.7 – Résultat de la segmentation : L'algorithme de segmentation utilise 22 plans sur 50 extraits pour décrire le bâtiment, dans le mode sans contrainte.

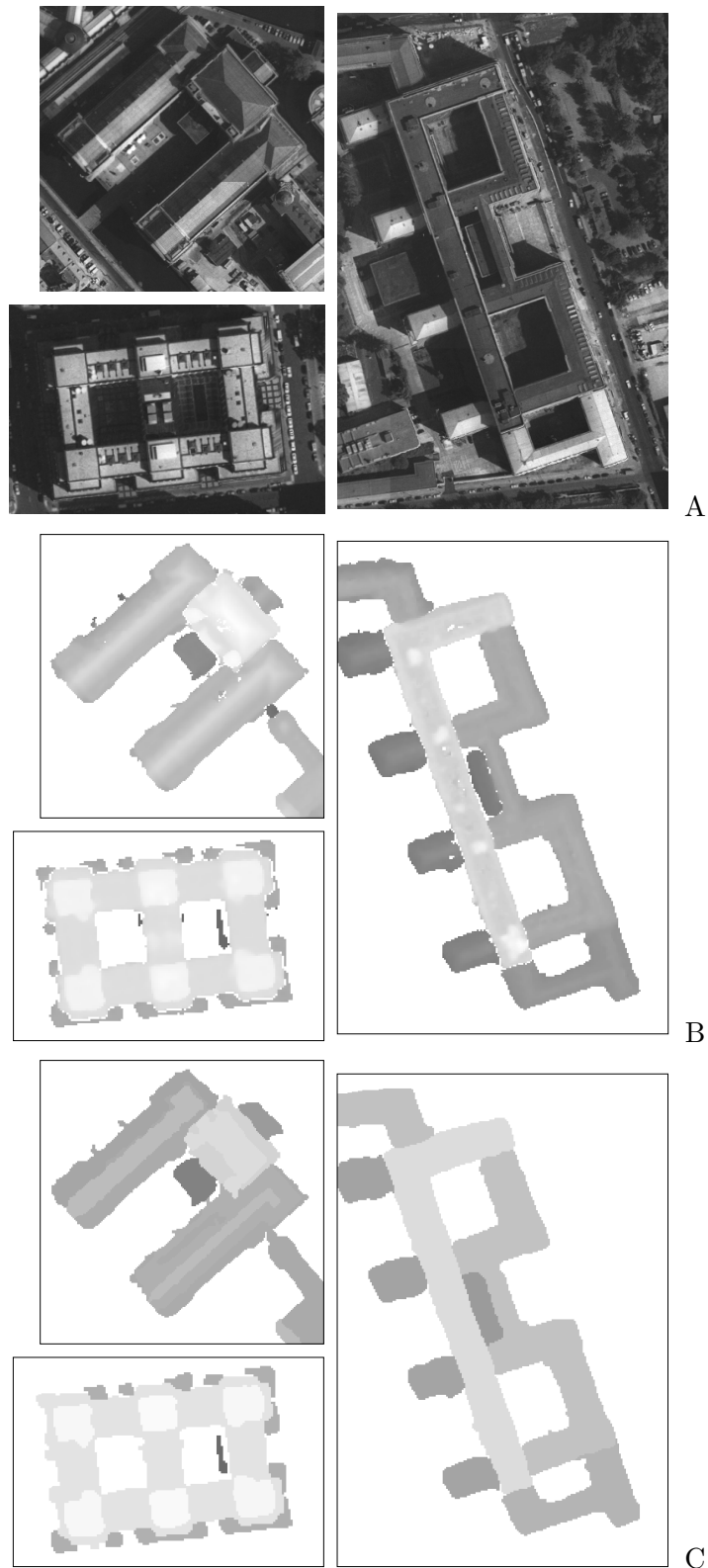


FIG. 5.8 – Résultats de la segmentation, mode plans horizontaux: Les images de A présentent trois ortho-images de la zone d'intérêt $ROI_{bât}^{2.5D}$ sur chacun des bâtiments traités. Les images de B sont les MNE correspondants. Les MNE synthétiques que nous construisons après l'étape de segmentation automatique sont présentés dans C. Nous utilisons uniquement les plans horizontaux pour la description de ces bâtiments.

bâtiment avec des plans sans contrainte. Cette valeur est surestimée pour la construction de plans horizontaux. Cette valeur est fixée, mais elle devra être réévaluée selon la complexité du bâtiment ou sa taille. Si nous voulons un modèle plus précis, nous devons diminuer le seuil de tolérance d'erreur. Nous devons aussi augmenter la taille de la liste des hypothèses, car les hypothèses construites seront beaucoup plus sensibles aux petites structures présentes sur les toits (cheminées . . .) et au bruit présent dans le MNE.

Estimation des plans Nous avons testé la procédure de segmentation en combinant plusieurs estimateurs (traditionnels et robustes) avec différentes procédures d'estimation. Les estimateurs que nous utilisons sont l'estimateur des moindres carrés (LS), un estimateur des moindres carrés non linéaire (NLS), un M-estimateur robuste et l'estimateur de la moindre médiane des carrés (LMS). Nous souhaitons sélectionner l'estimateur le plus adapté, chacun possédant des caractéristiques différentes : rapidité et sensibilité au bruit principalement.

Les différentes procédures que nous utilisons pour l'estimation des modèles sont : (1) chercher tous les points qui vérifient le modèle initial puis estimer les paramètres du modèle, (2) après chaque croissance de région, ré-estimer les paramètres du modèle et (3) après chaque croissance de région, ré-estimer les paramètres du modèle et rejeter les points qui n'appartiennent plus au domaine de tolérance du nouveau modèle estimé.

A partir de nos expériences, nous avons adopté différentes méthodes pour chacun des deux modes : segmentation du bâtiment avec des plans horizontaux et segmentation avec des plans qui peuvent avoir une orientation quelconque. Avec la contrainte horizontale, nous utilisons l'estimateur du LMS et la troisième procédure qui permet de rejeter les données erronées tout au long de la procédure. Dans le mode sans contrainte, nous choisissons d'utiliser la troisième procédure avec un estimateur des moindres carrés (LS) pour conserver un temps de calcul raisonnable.

5.6 Conclusion

Nous avons présenté dans ce chapitre la segmentation d'un MNE en plans. Nous avons introduit l'estimation de modèles puis la segmentation de données en plans, pour reconstruire des modèles correspondant aux différentes parties de la surface. Nous avons ensuite vu les méthodes robustes qui permettent d'estimer les modèles dans le cas où des données aberrantes sont présentes, les faux appariements dans le cas d'un MNE. Nous avons aussi introduit les principaux critères permettant la sélection de modèles avant de présenter la méthode de segmentation des bâtiments que nous utilisons.

Notre approche est basée sur l'algorithme ExSel++ présenté par *M. Stricker* et *A. Leonardis* dans [SL95a]. Elle se compose de trois étapes. La première étape est l'exploration des données, nous utilisons une version améliorée de l'algorithme du RANSAC. La deuxième étape est une étape de fusion qui supprime la redondance des hypothèses. Nous utilisons deux critères, le recouvrement des modèles et un test statistique permettant de comparer les modèles. La dernière étape est la sélection qui choisit le meilleur jeu de modèles pour décrire les données.

Nous avons développé deux modes de segmentation, un premier qui permet de représenter les données par des plans d'orientation quelconque et un second qui contraint les plans à être horizontaux. Nous avons montré des résultats pour ces deux modes. Dans les deux cas, les différentes parties du toit sont retrouvées et reconstruites correctement. Même si le mode de recherche de plans d'orientation quelconque donne des résultats satisfaisants, nous

utiliserons dans la suite uniquement le mode de recherche de plans horizontaux. Du fait que nous utilisons un MNE de $50cm$ de résolution, il est seulement possible d'extraire les plans principaux de la structure du bâtiment, et de retrouver sa forme générale. Seul le mode de recherche de plans horizontaux permet de garantir une constante qualité de segmentation avec cette résolution. Nous pourrions utiliser l'autre mode de recherche avec des MNE de plus haute résolution.

Chapitre 6

Vectorisation

L'objectif de la modélisation est d'obtenir une description exploitable par les applications qui nous intéressent. Nous avons présenté dans la section 3.3 plusieurs modèles de bâtiments pour décrire la structure de ceux-ci et nous avons choisi d'utiliser un modèle de type polyédrique. Nous avons présenté dans le chapitre 5 un algorithme de segmentation du MNE en plans, chacun des plans obtenus correspond à l'une des faces du modèle de bâtiment. Pour construire le modèle final du bâtiment, nous devons tout d'abord extraire les contours de chacune de ces faces. Nous les regroupons ensuite pour construire les arêtes du modèle polyédrique, c'est-à-dire un **modèle polygonal**.

Ce chapitre présente la construction du modèle polygonal à partir du jeu de plans construit par l'algorithme de segmentation. Nous proposons dans une première partie une méthode pour construire le modèle polygonal du bâtiment à partir de la liste de modèles de plans. Cette méthode est basée sur l'approximation polygonale d'une chaîne de points. Dans une seconde partie, nous étudions plusieurs méthodes de recalage et de correction du modèle obtenu. L'objectif de ces méthodes est de supprimer les imperfections du modèle polygonal en s'aidant des contours des images ou de contraintes géométriques ou topologiques. Nous avons développé trois algorithmes différents : un algorithme de recalage basé sur des **snakes polygonaux**, un algorithme de recalage par **programmation dynamique** et un algorithme de correction du modèle polygonal qui impose des contraintes angulaires.

6.1 Modélisation du bâtiment

L'étude et la compréhension d'une scène par l'interprétation directe des images ou d'un ensemble de points tridimensionnels sont difficiles. On cherche donc généralement à extraire des primitives pertinentes et exploitables pour décrire la structure tridimensionnelle de la scène. Dans une scène urbaine, qui est principalement composée d'objets polyédriques, ce sont les discontinuités de surface et de profondeur qui sont le plus souvent utilisées pour décrire les bâtiments, comme nous l'avons présenté dans la section 3.3. L'utilisation de ces primitives permet de structurer la description de la scène en diminuant considérablement la taille des données.

Dans les images qui observent la scène, ces discontinuités tridimensionnelles correspondent le plus souvent à des contours qui sont des variations brusques de l'intensité de l'image. En vision par ordinateur, le contour est l'une des primitives images la plus utilisée, elle permet de représenter la plupart des environnements construits par l'homme. Les plus forts contours des images représentent généralement les frontières (ou arêtes) des différentes

faces des objets de la scène, mais les contours correspondent aussi à des courbes physiques 3D présentes sur une surface continue. L'un des principaux avantages de l'utilisation des contours est leur insensibilité vis-à-vis des variations d'éclairage et des changements de point de vue.

6.1.1 La construction des chaînes polygonales du modèle

La construction des chaînes polygonales du modèle du bâtiment peut être obtenue selon deux stratégies différentes : l'extraction des contours des images ou la détection des discontinuités de surfaces dans un ensemble de données tridimensionnelles qui correspond à l'objet. En théorie, les deux stratégies permettent d'obtenir le même résultat.

L'extraction des contours présents dans les images peut se diviser en deux approches principales : la *détection des points de contours* et la *modélisation des contours*. La première approche consiste à extraire dans une première phase tous les points de contour de l'image en utilisant un détecteur de points de contour comme celui de *Canny* [Can86] ou de *Deriche* [Der87]. Ensuite, on utilise ces points pour construire des primitives de plus haut niveau comme les chaînes, les segments, les polygones ou les courbes splines. La seconde approche consiste à utiliser un modèle de contour, puis à optimiser les paramètres et la structure de ce contour par rapport aux données de l'image. Une des méthodes les plus répandues est celle des contours actifs (snakes) que nous présentons dans la section 6.2.1.

La seconde stratégie est celle que nous avons choisie. Elle est basée sur la détection des discontinuités de surfaces, c'est le problème de segmentation que nous avons traité dans le chapitre 5 précédent. Nous allons utiliser les plans proposés par l'algorithme de segmentation pour retrouver ces discontinuités et modéliser le bâtiment. Chaque plan étant associé à l'une des faces du modèle de l'objet, les limites de ces plans correspondent à une discontinuité d'orientation ou de profondeur. Nous réalisons une approximation des détourages de chacun des plans pour obtenir les chaînes polygonales du modèle.

Approximation polygonale d'une chaîne de points

L'objectif de l'approximation polygonale est d'obtenir un modèle de description simple et représentatif de l'objet traité. Une étape préliminaire de chaînage est nécessaire pour regrouper les points (obtenus par détection ou à partir de plans comme dans notre cas) selon les objets qu'ils représentent. La méthode consiste à approximer ces chaînes de points par des chaînes polygonales possédant la même forme générale que l'objet représenté. Les objets de la scène étant polyédriques, la chaîne polygonale que nous souhaitons construire sera composée uniquement de segments et de sommets (nous n'utilisons pas de modèle de courbe). Les avantages de l'approximation sont de structurer l'objet visualisé et de réduire la taille de description.

Les critères d'une bonne approximation polygonale

La chaîne de points initiale étant déjà une chaîne polygonale, le problème de l'approximation polygonale consiste à approximer ou simplifier une chaîne polygonale P par une chaîne polygonale P' . La qualité de l'approximation va dépendre du nombre de sommets utilisés dans la chaîne polygonale P' . Plus le nombre de sommets utilisés sera élevé, plus la description du contour sera fine, mais en contrepartie la taille de la description sera d'autant plus importante. Il faut donc réduire considérablement le nombre de sommets en simplifiant la description. Un exemple de simplification possible est le cas où plusieurs sommets

consécutifs de la courbe sont alignés. Tous les sommets étant compris entre les deux extrémités peuvent être supprimés de la courbe, le segment défini entre ces extrémités suffit à la description. Malheureusement cette simplification n'est généralement pas suffisante car la description reste encore très volumineuse. L'approximation polygonale doit donc supprimer les sommets les moins significatifs de la chaîne polygonale P .

La difficulté de l'approximation polygonale réside dans le choix des sommets ou des segments qui permettront de décrire au mieux et le plus simplement possible la chaîne de points initiale P . Du fait de la numérisation, de l'imprécision du détecteur ou de toute autre source de bruit (mise en correspondance par corrélation dans notre application), la chaîne de points initiale est déformée, les bords des objets polyédriques ne sont plus linéaires et les angles deviennent des courbes arrondies. Différentes mesures ont été proposées pour évaluer la ressemblance de deux polygones, P et P' dans notre cas, et pour juger de la qualité de l'approximation. Les mesures de distance entre les deux polygones sont peu discriminantes car la chaîne de points initiale est souvent moins proche du modèle réel que son approximation car elle est plus régulière. Plus la chaîne est soumise à des déformations, plus ces mesures perdront de leur sens. Pour la modélisation d'une scène urbaine ou d'objets polyédriques, les critères portant sur la forme sont plus aptes à juger de la qualité de l'approximation. Nous proposons dans la section 6.2.4 une méthode permettant de corriger un modèle de bâtiment polygonal et de privilégier certaines formes régulières, les angles droits et les angles plats.

Intuitivement, les deux caractéristiques essentielles de la chaîne de points qui doivent être préservées sont les **bords** de l'objet et les **angles** du contour de l'objet. Pour préserver les bords, une des méthodes possible consiste à retrouver les sommets de la liste initiale de points qui supportent ces limites puis à estimer le meilleur segment qui leur correspond au sens des moindres carrés (ou par toute autre technique d'estimation). Pour préserver les angles, il est souvent nécessaire de déplacer le sommet de la courbe correspondant à cet angle pour obtenir une meilleure représentation. On peut utiliser par exemple l'intersection des deux segments estimés de part et d'autre du sommet. Nous allons présenter maintenant l'algorithme d'approximation polygonale que nous utilisons.

6.1.2 L'algorithme de construction du modèle polygonal du bâtiment

Nous avons exposé la construction du modèle polygonal dans [VDM00]. Dans la suite du chapitre nous n'utiliserons que des **modèles de plans horizontaux** pour décrire les différentes parties de la surface d'un toit. Ces modèles sont obtenus avec l'algorithme de segmentation de MNE utilisant la contrainte d'horizontalité. Il y a deux problèmes que nous nous proposons de résoudre avant d'appliquer l'approximation polygonale : tout d'abord, il reste des régions de recouvrement entre les différents morceaux de surfaces utilisés, ensuite, quelques trous apparaissent entre les modèles de plans et au milieu de certains de ces modèles. La plupart de ces problèmes proviennent directement du MNE, dans les régions où l'algorithme de mise en correspondance a échoué. Nous proposons une étape de pré-traitement pour corriger ces problèmes avant la construction du modèle polygonal.

Pré-traitement

Nous proposons de construire un MNE local synthétique à partir de la liste des modèles de plans. Ce MNE synthétique permet de garantir une cohérence $2D^{\frac{1}{2}}$ du modèle polygonal que nous allons construire. Chaque pixel est affecté à une seule altitude, donc à un seul

modèle. Si un pixel appartient à plusieurs modèles, le pixel est affecté au modèle de plus basse altitude. Nous avons choisi ce mode d'affectation car les toits débordent généralement de leurs limites réelles. Si un pixel n'appartient à aucun modèle, nous utilisons un autre MNE correspondant à un modèle numérique complet de la région d'intérêt. Ce MNE est une carte interpolée du MNE initial. Nous prenons la valeur de l'altitude du pixel dans ce MNE complet. Ensuite, nous affectons le pixel au modèle qui est le plus près de cette altitude (composante en Z).

Après la construction de ce MNE synthétique et complet, nous appliquons une procédure de filtrage composée de deux étapes. La première étape supprime les petites régions résiduelles qui ont une surface inférieure à $12m^2$. La seconde étape est un filtrage morphologique. Nous utilisons ce filtrage pour lisser les frontières entre les différents modèles, principalement au niveau des jonctions. Nous utilisons les filtres morphologiques d'ouverture/fermeture suivis des filtres de fermeture/ouverture.

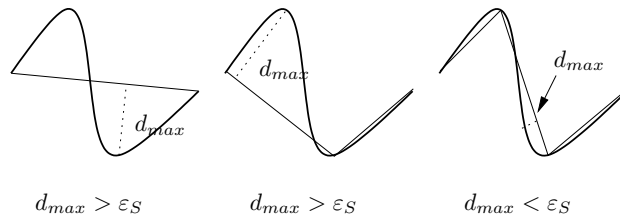
Le MNE synthétique de la région d'intérêt peut être interprété comme une image segmentée en différentes régions. Nous proposons maintenant une méthode pour extraire le modèle polygonal à partir de ce MNE segmenté. Nous commençons par extraire deux types de primitives de cette image : les jonctions et les chaînes. Les chaînes correspondent aux listes de points de l'image qui délimitent les différentes régions. Les jonctions sont les limites de ces chaînes, elles peuvent être de différents types : une jonction simple est l'intersection du bord du MNE et d'une chaîne, une jonction double ferme une chaîne et une jonction triple ou de degré supérieur correspond aux points qui délimitent plusieurs régions. La méthode que nous présentons est composée de deux processus distincts. Le premier processus réalise une approximation polygonale des chaînes, les jonctions étant fixes. Le second processus traite les différentes configurations de jonctions et simplifie le modèle si nécessaire.

Approximation polygonale des chaînes

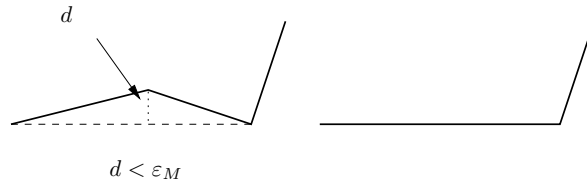
Nous proposons un algorithme d'approximation des chaînes basé sur les travaux de *T. Pavlidis* et *S. Horowitz* [PH74]. Ce choix a été guidé par l'étude comparative de différentes méthodes d'approximation polygonale présentée dans la thèse de *A. Filbois* [Fil95]. Nous avons choisi de baser notre procédure sur cet algorithme pour son efficacité et la facilité de sa mise en oeuvre. L'algorithme original utilise des étapes successives de **split** et de **merge** tant que la chaîne polygonale approximante est modifiée. Ensuite, une étape d'approximation par moindres carrés estime les paramètres de chaque segment et l'étape finale calcule les nouvelles positions des sommets de la chaîne polygonale. Nous adoptons les mêmes tests d'ajout ou de suppression de sommets de la chaîne polygonale dans l'algorithme que nous avons développé. On ajoute un sommet (**split**) lorsque la distance maximum des points de la chaîne initiale d_{max} à la chaîne polygonale courante est supérieure à un seuil ε_S (figure 6.1 A). Deux segments successifs de la chaîne sont regroupés (**merge**) lorsque la distance d entre le sommet milieu (celui se trouvant à l'intersection des segments) et la droite définie par les deux autres sommets de la chaîne polygonale est inférieure à un seuil ε_M (figure 6.1 B).

L'algorithme que nous avons développé est décrit dans la figure 6.2. Nous proposons d'améliorer l'algorithme original défini par *T. Pavlidis* et *S. Horowitz* en ajoutant les composantes suivantes :

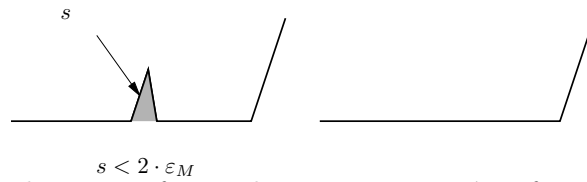
- Nous ajoutons un second critère pour fusionner deux segments (étape de **merge**) basé sur la surface du triangle qui est définie par trois points successifs de la chaîne polygonale. Ce test permet de supprimer le bruit résiduel présent dans la chaîne polygonale



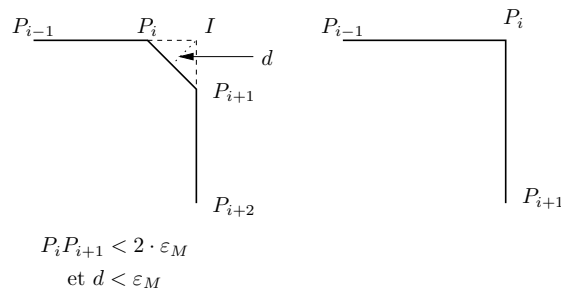
A. Étape du **split** : tant que $d_{max} > \epsilon_S$, on ajoute un sommet à la chaîne.



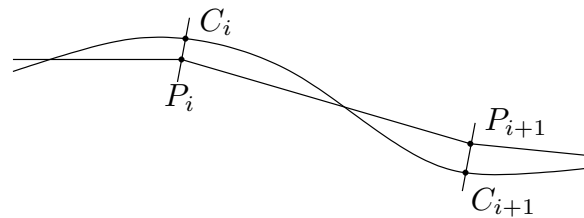
B. Étape du **merge** : fusion de segments consécutifs selon le critère de distance.



C. Autre étape de **merge** : fusion de segments consécutifs selon le critère de surface.



D. Correction des coins : Si les deux tests sont valides dans une partie de la chaîne, nous retrouvons un coin du modèle et corrigeons la chaîne polygonale.



E. Estimation des paramètres d'un segment à partir de la chaîne : nous utilisons la liste de points de la chaîne compris entre C_i et C_{i+1} pour estimer les paramètres du segment $P_i P_{i+1}$ de la chaîne polygonale.

FIG. 6.1 – A, B, C et D présentent les différentes étapes et tests de la procédure d'approximation polygonale. E présente le choix de la liste de points de la chaîne que nous utilisons pour l'estimation d'un segment de la chaîne polygonale.

(figure 6.1 C).

- Nous ajoutons dans la boucle **Tant que** une étape de correction de coins aux étapes de **split** et de **merge**. Cette correction traite les cas où les angles de la chaîne sont trop lisses, lorsqu'elle est décrite par deux points au lieu d'un (figure 6.1 D). Cette correction permet de limiter les déformations des angles pour mieux satisfaire le critère de qualité défini, préserver les angles.
- Les étapes d'estimation des paramètres des segments et du calcul d'intersection des points sont placées dans la boucle **Tant que**. Ces deux étapes peuvent encore demander de relancer des étapes de **split** et de **merge**.
- Nous utilisons l'estimateur de la moindre médiane des carrés (LMS) à la place de l'estimateur des moindres carrés (LS) pour éviter les problèmes d'initialisation et obtenir une solution robuste et représentative des segments. L'utilisation de cet estimateur permet au modèle de satisfaire l'autre critère de qualité correspondant aux bords de l'objet modélisé.

Tant que la chaîne est modifiée

- étape de **split** de la chaîne basée sur la distance entre la chaîne de points initiale et la chaîne polygonale,
- étape de **merge** de la chaîne basée sur les critères de distance et de surface pour trois points consécutifs de la chaîne polygonale,
- correction des coins,
- calcul des segments de la chaîne avec l'estimateur du LMS,
- calcul des nouveaux sommets correspondant aux intersections des segments de la chaîne polygonale,

fin Tant que

FIG. 6.2 – *Procédure d'approximation polygonale des chaînes.*

Parce nous ajoutons l'étape de correction des coins, l'étape de calcul des segments et l'étape de calcul des intersections de ces segments dans la boucle **Tant que**, les sommets de la chaîne polygonale peuvent être en dehors de la chaîne de points initiale. Pour calculer les segments, nous avons besoin de retrouver les points de la chaîne initiale qui correspondent à leurs supports. Pour sélectionner les points de support de chaque segment à ré-estimer, nous cherchons les deux points C_i et C_{i+1} de la chaîne initiale les plus proches des deux sommets P_i et P_{i+1} correspondant au segment initial comme le présente la figure 6.1 E. Nous utilisons les points C_i et C_{i+1} pour délimiter la liste de points définissant le support utilisé pour l'estimation des paramètres du segment.

Traitement des jonctions

Dans la procédure d'approximation polygonale, les limites des chaînes (les jonctions) étaient fixées pour éviter des déconnexions dans le modèle polygonal du bâtiment. Dans la procédure de traitement des jonctions, nous voulons ajuster les positions des jonctions pour obtenir un modèle polygonal plus juste et représentatif. Nous traitons toutes les jonctions en même temps, les sommets des chaînes restent fixes. Chaque type de jonction possède un traitement adapté que nous présentons dans la figure 6.3.

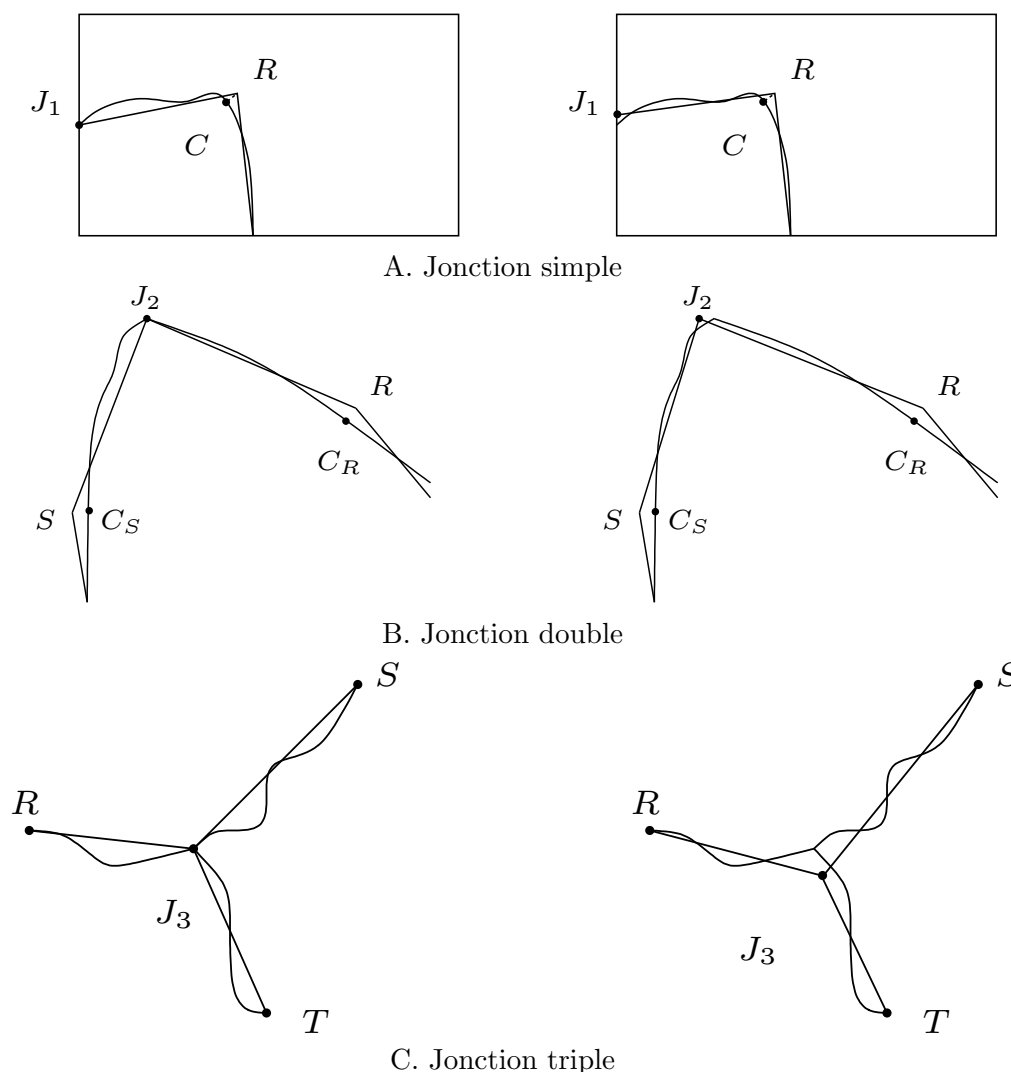


FIG. 6.3 – A, B et C présentent les traitements définis pour les différentes jonctions pour corriger leurs positions dans le modèle polygonal.

Une jonction simple J_1 est l'intersection du bord de l'image du MNE et de la chaîne polygonale. Dans un premier temps, nous cherchons les points de la chaîne initiale qui correspondent au dernier segment (celui qui joint le bord de l'image). Comme dans la procédure d'approximation polygonale, nous calculons le point C de la chaîne qui est associé à l'autre extrémité du segment R (figure 6.3 A). Nous calculons ensuite le nouveau segment, R étant fixe, en estimant ses paramètres avec le support défini par les points de la chaîne compris entre J_1 et C . Nous utilisons pour cela l'estimateur de la moindre médiane des carrés (LMS). La procédure de recherche consiste à tirer aléatoirement un point de la chaîne pour définir une hypothèse de jonction simple. Nous choisissons ensuite la meilleure hypothèse au sens de l'estimateur du LMS et repositionnons la jonction J_1 à l'intersection du nouveau segment et du bord de l'image.

Une jonction double J_2 correspond à une chaîne polygonale fermée. Nous utilisons la même stratégie avec deux segments. Nous calculons les deux parties J_2C_R et J_2C_S de la

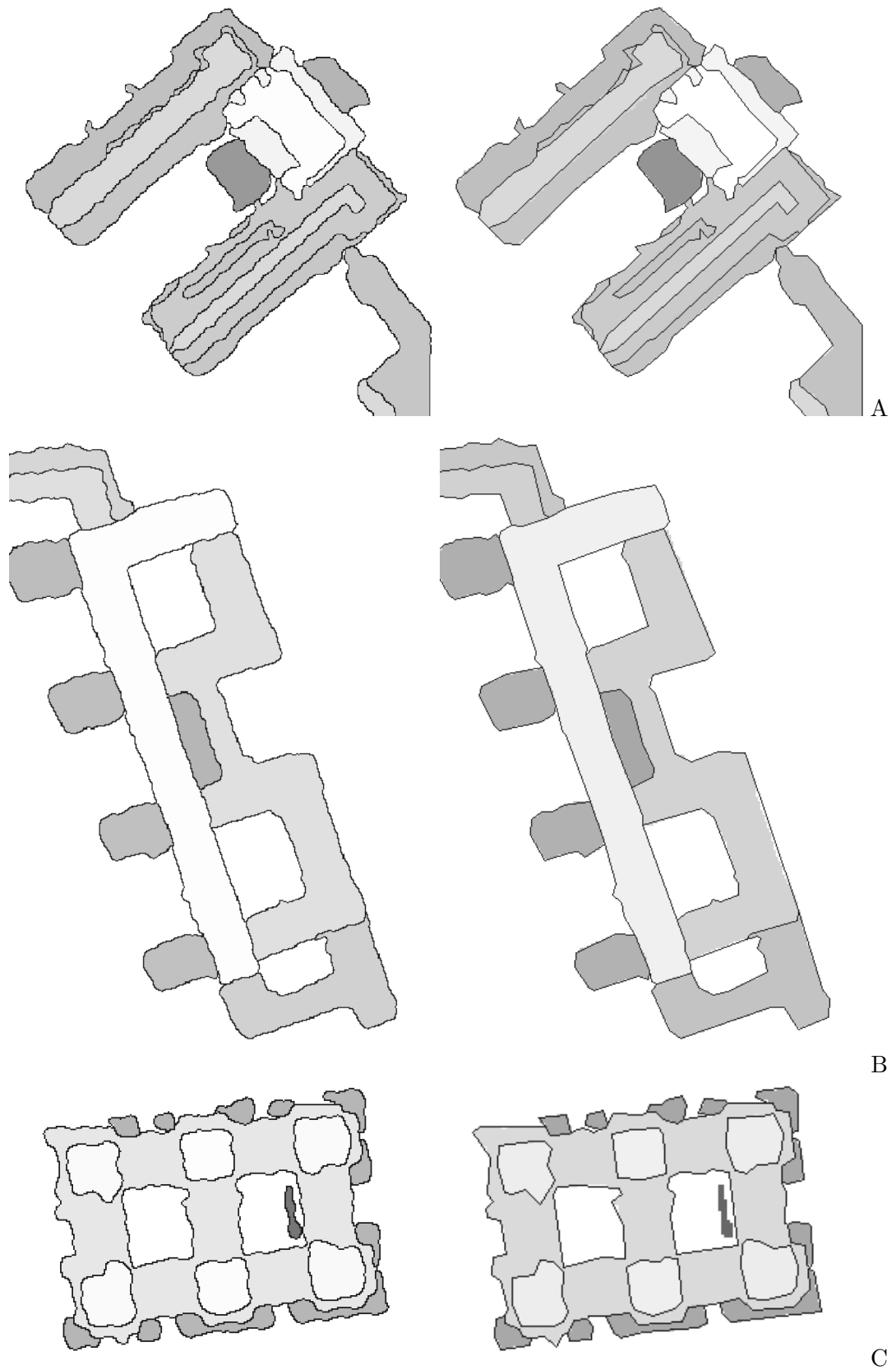


FIG. 6.4 – A, B et C présentent la vectorisation de trois bâtiments. Les images de gauche correspondent aux chaînes de points initiales extraites des MNE segmentés. Les images de droite présentent les résultats de la vectorisation.

chaîne initiale qui correspondent respectivement aux deux segments J_2R et J_2S (figure 6.3 B). Nous calculons ensuite les deux segments passant par les sommets R et S en utilisant l'estimateur du LMS. La procédure de recherche du LMS consiste à tirer aléatoirement un point par chaîne pour construire une hypothèse de jonction double. Nous plaçons ensuite la jonction J_2 à la position correspondant à la meilleure hypothèse.

Pour une jonction triple comme J_3 dans la figure 6.3 C, nous calculons la position qui minimise la somme des résidus des trois segments partant de la jonction. Nous cherchons la meilleure position de la jonction sans déplacer les sommets R , S et T (figure 6.3 C). Nous utilisons les points des trois morceaux de chaînes correspondant aux trois segments de la jonction triple. La procédure aléatoire de la moindre médiane des carrés consiste à tirer aléatoirement deux points sur deux chaînes de points différentes. Nous calculons la position de la jonction à l'intersection des deux segments définis par ces points. Nous calculons ensuite tous les résidus des points des trois chaînes, ceci pour pour chaque tirage aléatoire. Ensuite, nous sélectionnons la position qui minimise les résidus au sens du LMS. Nous ne traitons pas les jonctions de degré supérieur qui sont rarement présentes.

Résultat de la polygonalisation

Les résultats de la vectorisation sont présentés dans la figure 6.4. Les résultats sont présentés pour trois bâtiments différents, A, B et C. Pour chacun de ces bâtiments, l'image de gauche présente les chaînes de points initiales extraites des MNE segmentés. Ces chaînes (en noir) sont présentées sur l'image correspondant au résultat de l'algorithme de segmentation. L'image de droite présente le modèle polygonal construit par l'algorithme d'approximation polygonale. Nous voyons que la forme générale de chaque bâtiment est respectée. Seul le premier des deux critères de qualité définis précédemment est respecté : les segments du modèle polygonal représentent correctement les bords des objets segmentés. Certains angles du bâtiment ont pu être retrouvés grâce à la procédure de correction de coins intégrée dans l'algorithme d'approximation polygonale. Les coins du bâtiment trop lissés sont décrits par plusieurs sommets. Ces imperfections sont principalement dues à la qualité du résultat de segmentation.

6.2 Les méthodes de correction des contours

Les chaînes du modèle polygonal du bâtiment sont construites à partir des modèles de plans générés par l'algorithme de segmentation. Nous avons constaté dans la section 4.1 que les bâtiments débordent de leurs limites réelles dans le MNE. Cet effet n'est pas corrigé par ce dernier, les plans construits par l'algorithme de segmentation possèdent les mêmes imperfections. Nous avons constaté aussi que les détourages des bâtiments ne sont plus rectilignes et que les coins sont arrondis. Nous retrouvons aussi ces imperfections dans la liste de plans qui ont servis à construire le modèle du bâtiment et donc dans le modèle polygonal du bâtiment comme le présente la figure 6.5.

Les méthodes que nous allons présenter maintenant permettent de corriger ces imperfections. Il y a deux catégories de méthodes pour corriger le modèle polygonal : les méthodes de recalage de contour et les méthodes de correction géométriques et topologiques. La première catégorie de méthodes va chercher à repositionner le modèle polygonal sur les contours des images. Nous proposons d'étudier les méthodes de contours actifs qui transforment le problème de recalage en problème de minimisation et les méthodes exploratoires qui recherchent

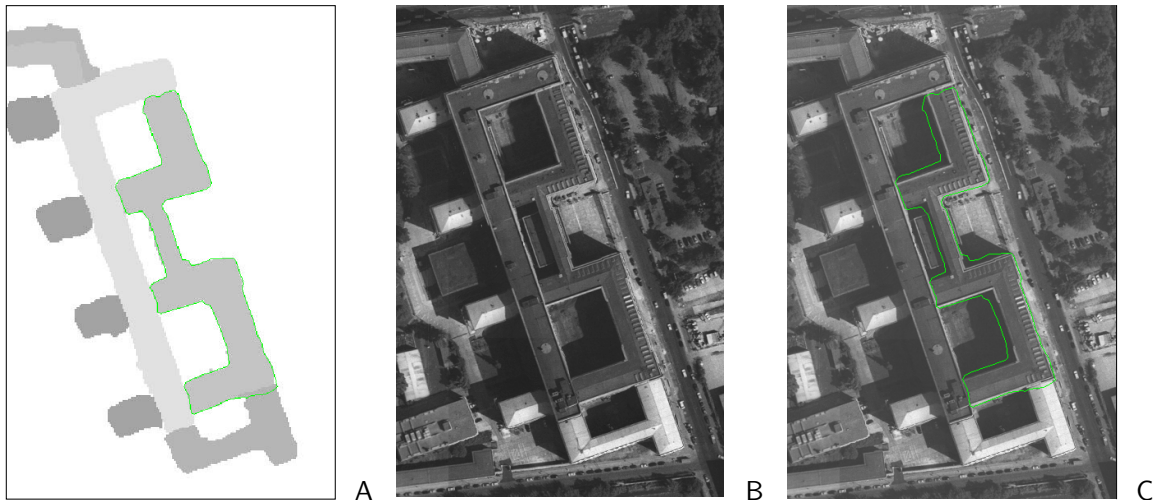


FIG. 6.5 – *Décalage résiduel entre le modèle obtenu à partir du MNE et les contours des images. A présente le modèle polygonal reconstruit projeté sur le MNE initial (trait vert). B est l'ortho-image de la scène. C présente le même modèle polygonal que dans A, projeté sur l'ortho-image correspondante (trait vert). Nous pouvons voir qu'il existe un décalage important du côté des ombres.*

le chemin de coût minimal. La seconde catégorie de méthodes applique des contraintes géométriques et/ou topologiques au modèle pour simplifier et régulariser la forme du bâtiment, ceci pour retrouver les angles et les bords rectilignes de chaque toit.

6.2.1 Les modèles déformables

Les modèles déformables permettent d'approcher un objet à partir de ses lois d'évolution. Ces modèles sont basés sur les lois de la physique et de la dynamique. *M. Kass, A. Witkins et D. Terzopoulos* [KWT87] introduisent dans le domaine de la vision par ordinateur une approche de déformation de modèle basée sur la minimisation d'énergie. Cette approche est appelée **contours actifs** ou **snakes**. Depuis, de nombreux autres modèles ou méthodes de résolution ont été proposés [OS88, FL90, Coh91, Bas94, CKS97]. On peut généralement classer les modèles déformables dans plusieurs groupes : les contours actifs, les modèles déformables rigides ou articulés, les modèles déformables statistiques. Le groupe des contours actifs est certainement celui qui a été le plus traité au cours de ces dernières années, c'est celui que nous nous proposons d'étudier maintenant.

Approches Lagrangiennes et Eulériennes L'approche développée par *M. Kass, A. Witkins et D. Terzopoulos* dans [KWT87] est basée sur la minimisation d'une énergie pour déformer un modèle jusqu'à l'objet cherché. Cette approche fait partie des approches Lagrangiennes, l'évolution du modèle est obtenue en lui appliquant des forces virtuelles de manière explicite. Ces forces sont déterminées à partir des données, des contraintes de régularité du modèle ou du milieu ambiant.

L'approche Eulérienne décrit les mouvements d'un système de manière globale. On ne suit plus les particules du modèle, on applique des déformations à l'espace tout entier. L'approche Eulérienne la plus couramment utilisée est la propagation de fronts (Level-

Set [OS88, Set96]). Le principe est de définir une équation de diffusion pour l'espace (3D pour retrouver un contour 2D) dépendant de l'image. On suit ensuite l'évolution d'une iso-potentielle dans cet espace dont la position de repos finale détermine implicitement la forme de l'objet.

L'approche Eulérienne possède principalement trois atouts. Tout d'abord, la forme du modèle initial est un paramètre beaucoup moins critique. Les approches Lagrangiennes nécessitent généralement un modèle initial proche de la solution. Le deuxième avantage concerne le changement de topologie du contour qui est implicite pour les approches Eulériennes. Les approches Lagrangiennes doivent formuler explicitement cette caractéristique comme dans [DM00]. Le dernier avantage est l'indépendance par rapport à la paramétrisation du modèle.

Il y a néanmoins un grand nombre de lacunes ou de difficultés lorsque l'on utilise une approche Eulérienne comme la propagation de fronts. Tout d'abord le lissage du modèle est intrinsèque. L'ajustement des paramètres et la résolution numérique des équations sont très délicats. Les temps de calcul sont plus importants et la mise en oeuvre plus complexe. De plus, le modèle a tendance à traverser les lignes brisées. Au contraire, les approches Lagrangiennes permettent de combler l'information manquante grâce au principe de régularisation.

Nous avons choisi de développer une méthode de contours actifs (snakes), une approche Lagrangienne donc, pour plusieurs raisons. La raison principale est que nous voulons modéliser un contour polygonal pour l'application qui nous intéresse. L'approche Eulérienne ne permet pas de retrouver de modèle polygonal. De plus, l'algorithme de segmentation présenté dans le chapitre 5 nous permet de disposer d'un contour initial proche de la solution, avec une topologie semblable à celle de la forme du bâtiment. Nous présentons dans la suite les contours actifs et plus particulièrement le modèle des snakes polygonaux.

6.2.2 Les contours actifs

Les contours actifs ont été proposés pour la première fois dans le domaine du traitement d'image par *M. Kass*, *A. Witkins* et *D. Terzopoulos* [KWT87]. Le principe est de définir un modèle déformable par le choix d'une représentation (courbe, segment, surface . . .) et d'une énergie potentielle. Cette énergie potentielle représente à la fois l'interaction entre le modèle et les données, les interactions internes du modèle et les contraintes éventuelles. Le mouvement du modèle n'est pas connu, on minimise l'énergie définie pour déformer le modèle jusqu'à un point d'équilibre, l'objet recherché. Les modèles déformables étant régis par des lois physiques, la minimisation de cette énergie correspond à appliquer des forces physiques au modèle.

Considérons un contour représenté par une courbe $v(s,t)$ où s est un paramètre de parcours et t le temps. Les auteurs proposent de minimiser l'énergie potentielle E le long de la courbe \mathbf{C} suivante :

$$\mathbf{E}(t) = \mathbf{E}_{\text{ext}}(t) + \mathbf{E}_{\text{reg}}(t)$$

Le critère d'optimisation $E_{\text{ext}}(t)$ est l'énergie d'interaction entre le modèle et l'image. Il dépend du type de primitives cherchées dans l'image. Nous cherchons à superposer notre courbe à un contour de l'image. Comme un contour correspond à une zone de forte intensité de gradient, on va chercher à maximiser la somme des intensités de gradient le long de la courbe. L'énergie correspondante à minimiser s'exprime comme suit :

$$\mathbf{E}_{\text{ext}}(t) = - \int_{\mathbf{C}} |\nabla I(v(s,t))| ds$$

L'optimisation de cette unique énergie est un problème mal posé car il existe un bruit présent dans les images. Les auteurs introduisent une énergie de régularisation \mathbf{E}_{reg} qui ajoute des contraintes de lissage sur les contours cherchés. Cette énergie de lissage est composée de deux termes : un premier terme représentant la résistance à la tension, et un second représentant la résistance à la courbure.

$$\mathbf{E}_{\text{reg}}(t) = \int_{\mathbf{C}} \alpha(s) \left| \frac{\partial v}{\partial s}(s,t) \right|^2 ds + \beta(s) \left| \frac{\partial^2 v}{\partial s^2}(s,t) \right|^2 ds$$

M. Kass, A. Witkins et D. Terzopoulos proposent de résoudre le problème sous sa forme stationnaire. Pour diminuer les temps de calcul, on préfère généralement utiliser la formulation dynamique. Pour cela, on considère la courbe comme un objet dynamique se déformant dans un milieu visqueux. Le mouvement est déterminé à l'aide du principe de moindre action généralisé de *Hamilton*. L'équation dynamique obtenue par l'utilisation des équations d'*Euler-Lagrange* est la suivante :

$$\mu \frac{\partial^2 v}{\partial t^2} + \gamma \frac{\partial v}{\partial t} - \frac{\partial}{\partial s} \left(\alpha \frac{\partial v}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left(\beta \frac{\partial^2 v}{\partial s^2} \right) = \frac{\partial |\nabla I(v)|}{\partial v} \quad (6.1)$$

où μ est la densité massive de la courbe (on la suppose généralement nulle pour qu'il n'y ait pas d'inertie) et γ la viscosité. On discrétise ensuite l'équation 6.1 pour obtenir les équations dynamiques du contour actif qui vont permettre de définir son évolution :

$$\begin{aligned} (\gamma I + K) x(t) &= \gamma x(t-1) + \frac{\partial |\nabla I(v)|}{\partial x}(t-1) \\ (\gamma I + K) y(t) &= \gamma y(t-1) + \frac{\partial |\nabla I(v)|}{\partial y}(t-1) \end{aligned}$$

avec K matrice de déformation.

L. Cohen introduit dans [Coh91] un terme énergétique supplémentaire d'inflation du modèle pour permettre au modèle de se déplacer plus rapidement surtout dans les régions ayant un faible gradient. Cette énergie permet au snake de ne pas se refermer sur lui-même mais demande des modifications importantes de l'algorithme. *P. Fua* et *C. Brechbühler* présentent dans [FB96] un algorithme pour imposer des contraintes dures à l'évolution des contours actifs. Cet algorithme procède en deux phases : une phase d'évolution dans le sous espace tangent à la surface des contraintes et une phase de reprojction orthogonale du modèle sur cette même surface.

Snakes polygonaux

Nous utilisons un modèle polyédrique de bâtiment, les contours correspondant aux discontinuités de ce modèle sont des courbes polygonales. Nous utilisons le modèle des snakes polygonaux présenté par *B. Basclé* [Bas94] et inspiré de l'article de *P. Fua* et *Y. Leclerc* [FL90]. *T. Moons et al.* ont proposé dans [MFVG98] une approche similaire. *P. Fua* présente dans [Fua96a] plusieurs modèles pour des application cartographiques, notamment pour retrouver les contours des bâtiments.

Le critère utilisé pour ce type de modèle doit être réadapté. Les points de contrôle de la courbe, qui sont les seuls à se déplacer, correspondent aux sommets ou jonctions de segments des chaînes polygonales du bâtiment. Les angles entre ces segments prennent en général des valeurs quelconques.

Les forces régularisantes utilisées précédemment pour conditionner le système n'ont plus de signification pour ce type de modèle. Ces forces empêcheraient le contour actif de prendre une forte courbure, donc de trouver des angles aigus. Les snakes polygonaux ne comportent donc pas de terme régularisant afin de pouvoir s'adapter à des configurations quelconques. On obtient alors les équations dynamiques simplifiées suivantes :

$$\begin{aligned}\gamma [x(t) - x(t-1)] &= \frac{\partial |\nabla I(v)|}{\partial x} (t-1) \\ \gamma [y(t) - y(t-1)] &= \frac{\partial |\nabla I(v)|}{\partial y} (t-1)\end{aligned}$$

Dans l'algorithme le plus simple de snakes polygonaux, les forces calculées à chaque itération ne sont déterminées qu'à partir du point de contrôle du contour. Le lissage des images et le bruit causent une délocalisation du sommet correspondant à l'angle réel. Dans [FL90, Fua96a] et [Bas94], les auteurs moyennent l'information sur un voisinage pour limiter l'influence du bruit. Ils prennent en compte les forces associées aux points situés le long des segments pour améliorer la localisation (forces obtenues à partir des gradients en chaque point). Un exemple de calcul des forces est illustré par la figure 6.6. La force subie par

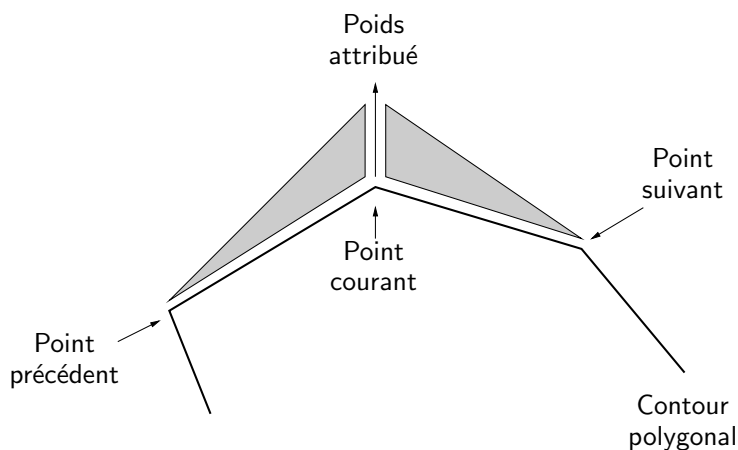


FIG. 6.6 – Calcul des forces introduites par les deux segments.

chaque point de contrôle du snake (sommet de la courbe polygonale) est la somme pondérée des forces mesurées le long des segments. On redéfinit une nouvelle énergie externe pour un sommet du snake. Nous pouvons par exemple utiliser les équations dynamiques suivantes pour chaque point de contrôle des snakes polygonaux :

$$\begin{aligned}\gamma [x(t) - x(t-1)] &= \frac{1}{n_{seg}} \sum_{j=1}^{n_{seg}} \sum_{i=0}^{n_j-1} \frac{1}{n_j} \left(1 - \frac{i}{n_j}\right) \frac{\partial |\nabla I|}{\partial x} (t-1) \\ \gamma [y(t) - y(t-1)] &= \frac{1}{n_{seg}} \sum_{j=1}^{n_{seg}} \sum_{i=0}^{n_j-1} \frac{1}{n_j} \left(1 - \frac{i}{n_j}\right) \frac{\partial |\nabla I|}{\partial y} (t-1)\end{aligned}\tag{6.2}$$

n_{seg} est le nombre de segments liés à la jonction ou au sommet courant et n_j est le nombre de points échantillonnés sur le segment j du snake. Les indices j et i parcourent respectivement les segments et les points appartenant aux segments utilisés pour le calcul de la

force au sommet. Tous les points utilisés sont attirés par le contour le plus proche. La force somme calculée sur les segments entiers permet au point courant (sommet ou jonction) de se déplacer vers ce contour de l'image.

Pondération multi-image

L'utilisation de snakes permet de positionner une courbe sur les contours d'une image. Puisque nous disposons de plusieurs images sur la scène, nous souhaitons toutes les utiliser. Nous proposons maintenant d'utiliser une stratégie multi-image comme *P. Fua* dans [Fua96a] ou *J. Trinder* et *H. Li* dans [TL95].

Nous considérons un snake polygonal dans l'espace tridimensionnel \mathbf{S}_{3D} comme celui que nous avons présenté. Les forces doivent être définies à partir des différentes images disponibles sur la scène. Le principe est de projeter le snake 3D dans les différentes images, puis de sommer les forces mesurées dans chacune de ces images pour déterminer la force à appliquer au snake 3D. La figure 6.7 présente la projection du snake dans les n images \mathbf{I}_i disponibles sur la scène. L'énergie externe du snake est définie comme étant la somme des

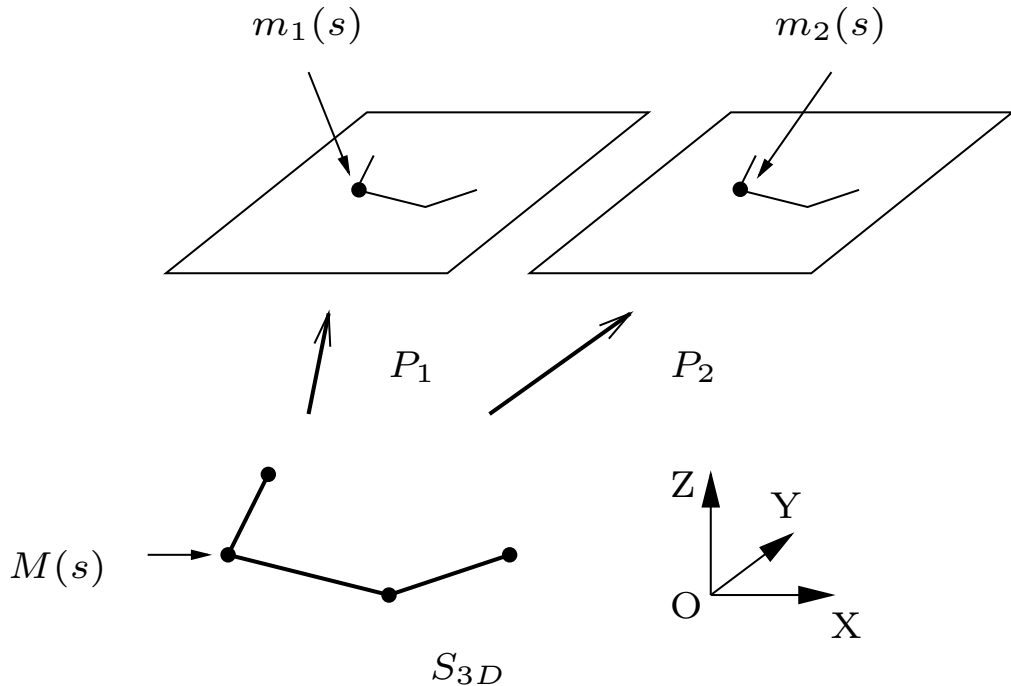


FIG. 6.7 – Projection du snake dans les différentes images.

énergies des snakes polygonaux 2D définis par les sommets $m_i(s)$ dans chaque image \mathbf{I}_i . On obtient la relation suivante :

$$E_{\mathbf{S}_{3D}} = E_1(x_1(s), y_1(s)) + \dots + E_n(x_n(s), y_n(s)) \quad (6.3)$$

Le snake polygonal initial a été construit à partir d'un modèle de plan généré par l'algorithme de segmentation du MNE. Ces modèles étant horizontaux, nous imposons cette contrainte sur le snake \mathbf{S}_{3D} . Celui-ci se va donc déplacer dans le plan horizontal défini par le modèle de plan.

Résultats des snakes polygonaux multi-image

Le recalage par contours actifs que nous avons développé contraint le déplacement du contour à être dans le plan horizontal défini par le modèle de plan initial. Nous avons donc développé un algorithme de recalage qui traite les contours extraits de chaque modèle de plan indépendamment. Cet algorithme est présenté dans la figure 6.8. Le snake défini dans l'espace tridimensionnel est dans un premier temps projeté dans chaque image initiale (tous ses sommets $M(s)$). On calcule la force 2D associée à chaque sommet $m_i(s)$ du contour défini dans l'image I_i . Cette force est déterminée le long des segments adjacents au sommet traité, en utilisant l'équation 6.2 on en déduit le déplacement correspondant dans l'image I_i . On évalue ensuite le déplacement équivalent tridimensionnel associé à cette même image, puis le déplacement somme totale qui doit être appliqué au sommet du contour actif.

```

Tant que le snake bouge ou iter < max_iter
  – Pour chaque point  $M(s)$  (jonction ou sommet) du polygone
    – Pour toutes les images  $i$ 
      – Projection du point 3D et de ses voisins dans l'image
      – Pour tous les segments liés au point
        Évaluer la force 2D du segment
      fin Pour
      – Évaluer le déplacement dans image associé au point  $m_i(s)$ 
      – Calculer le déplacement 2D dans l'espace 3D correspondant à
        ce déplacement dans image au point  $M$  du polygone
      fin Pour
      – Évaluer le déplacement total dans l'espace 3D
      – Déplacer le point  $M(s)$  du polygone
    fin Pour
  Incrémenter iter
fin Tant que

```

FIG. 6.8 – *Algorithme de recalage par les snakes polygonaux multi-images*: L'algorithme traite successivement tous les points de la chaîne polygonale. On projette les segments 3D liés à ce point dans chacune des images et on calcule la force somme des segments 2D leur correspondant. On calcule alors la force 3D équivalente et on déplace le snake 3D jusqu'à sa convergence.

Le déplacement dans chaque image est calculé à partir de plusieurs images de gradients. Ces images sont présentées dans la figure 6.10. A présente l'image représentant la norme du gradient de l'image initiale. On utilise ses images des dérivées selon x et selon y pour définir les images des dérivées secondes de l'intensité selon chacune de ces directions, respectivement B et C. Les intensités des pixels de ces images ont la particularité d'être nulles sur les points des contours réels comme le présente la figure 6.9. La force du contour actif sera nulle en ces points particuliers, le contour actif aura retrouvé le contour réel. Les valeurs de ces images de part et d'autre du contour réel sont positives et négatives, permettant ainsi de donner la direction à suivre au contour actif, lorsque celui-ci est suffisamment proche du contour réel. Les images des dérivées secondes B et C permettent donc de localiser les contours réels des

images.

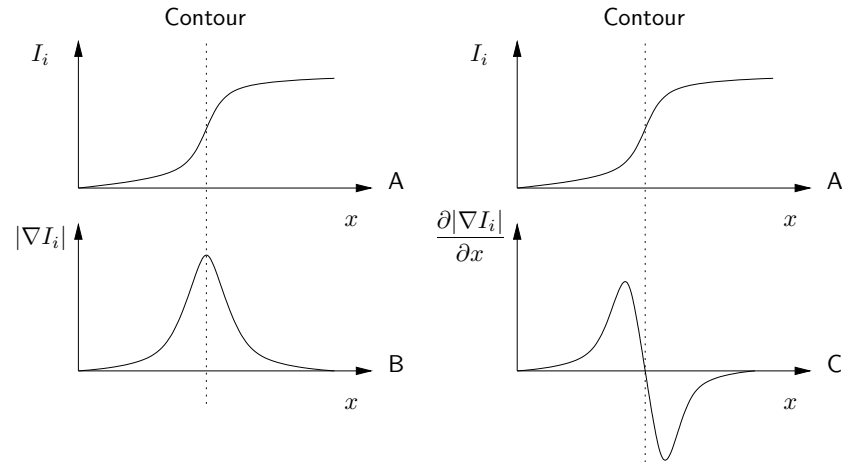


FIG. 6.9 – Relation entre un contour et les images de gradient : Les graphiques présentent le profil d'un contour orthogonal à l'axe des x (trait vertical). A est l'image initiale des intensités. Le contour correspond à une variation importante et rapide de l'intensité. B présente le profil de l'image de la norme du gradient associée. Le contour est théoriquement situé au maximum de la variation de l'intensité, il donne lieu aux maxima locaux de l'image de la norme du gradient. C présente le profil de l'image de la dérivée seconde (dérivée de la norme du gradient). Le contour correspond aux passages par zéro de cette image. L'algorithme des contours actifs utilise les images des dérivées secondes pour se déplacer en direction du contour, les valeurs positives et négatives de part et d'autre du contour génèrent une force qui attire le contour actif, cette force devient nulle lorsque le contour actif est recalé sur le contour (passage par zéro).

Pour faciliter et assurer la convergence des contours, il est nécessaire d'utiliser des images des dérivées secondes lissées. Cette opération de lissage permet aux contours actifs d'une part de ne pas être déviés par les perturbations dues à la présence de bruit, et d'autre part de faciliter la progression vers les zones de forte intensité de gradient. Un premier lissage est réalisé sur l'image de la norme du gradient par la convolution de l'image avec un noyau correspondant à une gaussienne de 4 pixels d'écart type. On calcule les dérivées de l'image de la norme des gradients pour obtenir les images des dérivées secondes de l'image initiale. On applique le même lissage à ces deux images pour obtenir les images utilisées par l'algorithme de recalage.

Les résultats de l'algorithme de recalage par contours actifs sont présentés dans la figure 6.11. Ces résultats sont présentés sur l'ortho-image correspondant à la zone d'intérêt, pour deux modèles de plans du bâtiment extraits par l'algorithme de segmentation, (A, B, C) et (D, E, F). Les images A et D présentent les deux contours initiaux obtenus par l'algorithme d'approximation polygonale. Ces deux contours sont décalés par rapport aux contours réels présents dans l'image. Les images B et E présentent le contour obtenu en utilisant l'algorithme de recalage par contours actifs avec une seule vue de la scène. Nous pouvons voir que les deux contours se sont déplacés vers les contours réels des deux faces traitées du bâtiment. Les images C et F présentent le contour obtenu en utilisant l'algorithme de recalage par contours actifs avec toutes les images disponibles sur le bâtiment étudié, soit douze images. Les contours actifs ont bien été recalés sur les contours réels du

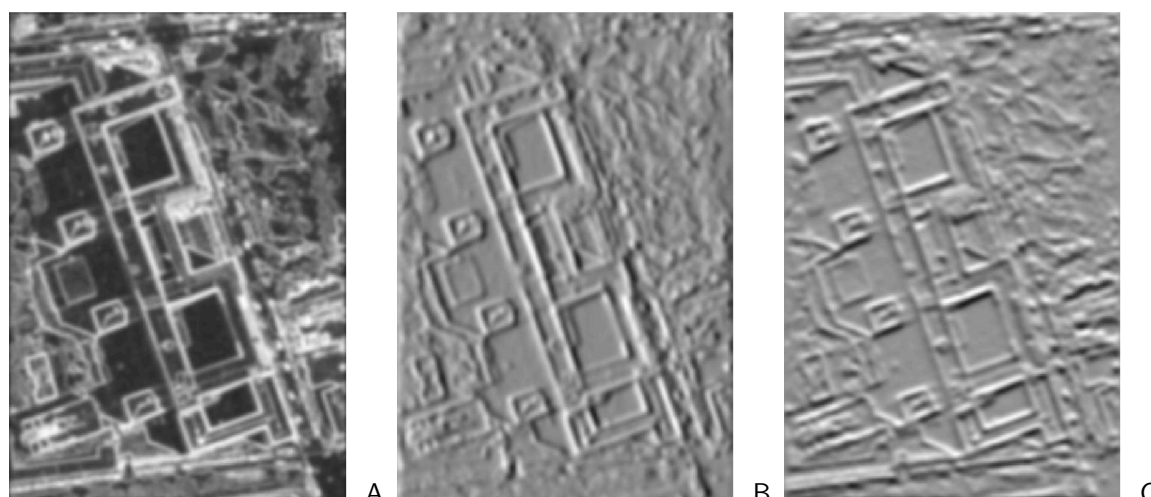


FIG. 6.10 – *Les images de gradient utilisées pour les snakes: Ces trois images sont associées à une vue aérienne. A représente la norme du gradient obtenu après filtrage par une gaussienne de 4 pixels d'écart type. B et C représentent respectivement l'image des dérivées en x et l'image des dérivées en y de l'image A. Ces images sont aussi obtenues après un second filtrage gaussien ayant les mêmes caractéristiques que le premier.*

bâtiment. Les deux contours obtenus sont plus précis que dans le cas où l'on utilisait une seule vue. On peut voir cependant que la différence n'est pas très notable, l'utilisation d'une seule image ayant permis dans ce cas précis d'obtenir une bonne solution. Dans le cas où les contours seront moins visibles, l'utilisation de plusieurs images permettra d'obtenir une bien meilleure solution que l'utilisation d'une seule image. Nous pouvons voir qu'il reste quelques imperfections dans le modèle car nous ne traitons pas le changement de topologie du contour. Nous proposerons dans le chapitre 7 une méthode pour les corriger.

6.2.3 Méthodes exploratoires

Les techniques par contours actifs ne sont pas les seules approches pour résoudre le problème de recalage des contours sur une image. Des méthodes d'exploration déterministe des solutions permettent aussi d'obtenir de bons résultats. La méthode que nous nous proposons d'étudier est basée sur la résolution du problème de recalage par programmation dynamique. Le principe est d'associer un coût à chaque position des segments et de résoudre le problème de recalage comme la recherche du chemin de coût minimal. La technique de résolution par programmation dynamique que nous présentons maintenant est basée sur celle de *A. Grün* et *H. Li* définie dans [GL95].

Recalage par programmation dynamique

Par opposition aux contours actifs, la programmation dynamique est une méthode d'optimisation qui n'est pas fondée sur la dérivabilité de la fonction de coût. Les snakes utilisent l'image lissée du gradient pour attirer le contour. La programmation dynamique possède l'avantage d'être indépendante vis-à-vis des perturbations présentes dans cette image. On dispose aussi d'une plus grande liberté de choix de la fonction de coût.

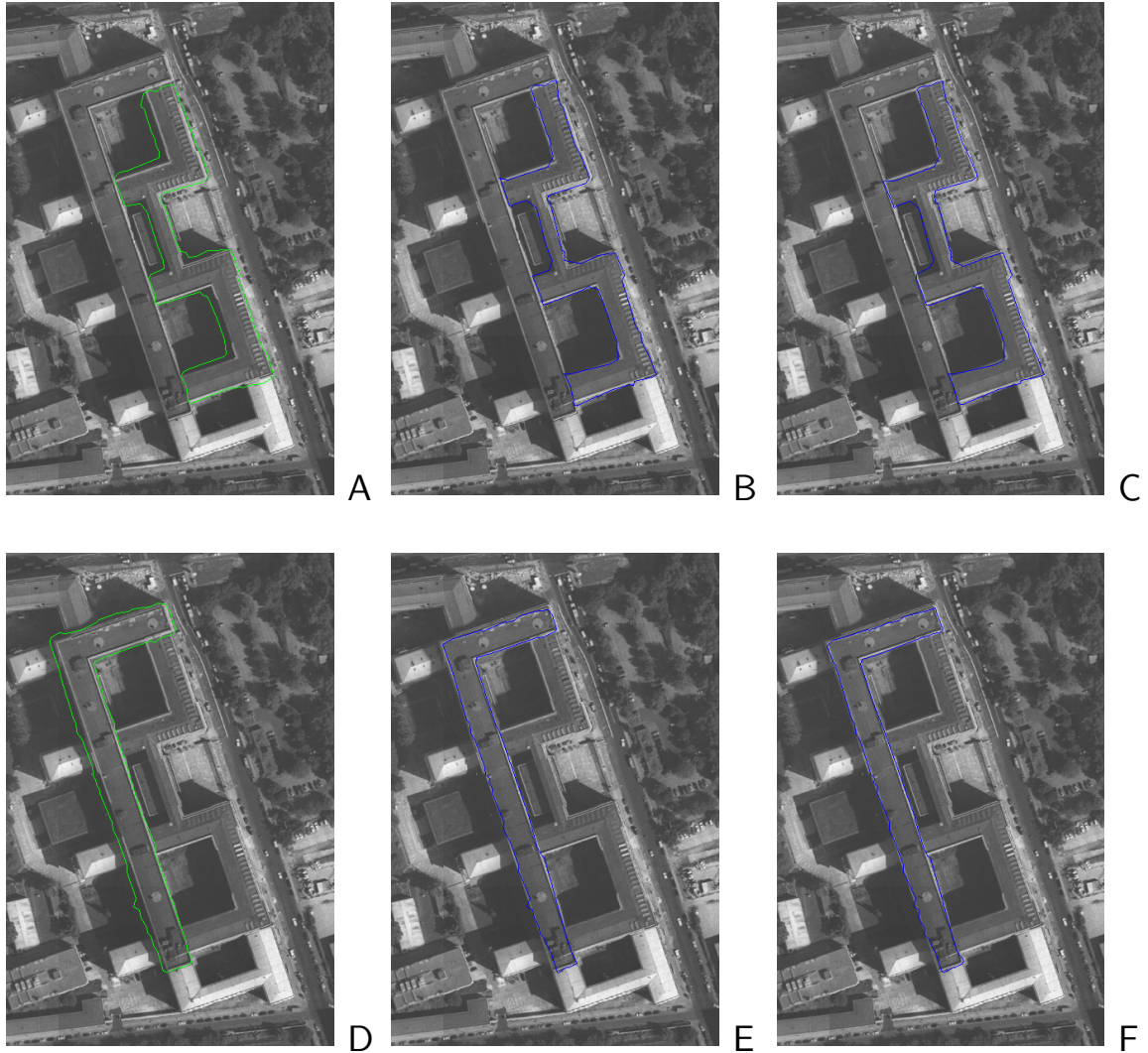


FIG. 6.11 – Résultat du recalage des contours par l'algorithme des contours actifs : (A, B, C) et (D, E, F) présentent les résultats pour deux modèles de plan du bâtiment superposés à l'ortho-image correspondante. A et D présentent le contour initial obtenu par l'algorithme d'approximation polygonale. B et E présentent le contour obtenu après recalage en utilisant une seule vue de la scène. C et F présentent le contour obtenu après recalage en utilisant toutes les vues disponibles sur cette scène, soit douze images. Les contours actifs ont bien été recalés sur les contours réels du bâtiment.

Dans ce type d'algorithme, on explore l'espace des solutions puis on choisit la meilleure, celle qui minimise la fonction de coût globale. La programmation dynamique repose sur le principe de séparabilité, on décompose un problème nécessitant N décisions en une suite de N problèmes nécessitant une décision. Pour le problème de recalage, on cherche les positions optimales des N sommets de la courbe. L'algorithme que nous utilisons est décrit dans la figure 6.12. Nous allons déplacer de manière déterministe les sommets du contour dans un voisinage d'étude autour de leur position initiale et calculer le coût local lui correspondant. Nous calculons ensuite la somme des coûts locaux pour chaque chemin de la chaîne polygonale (liste de segments), ceci pour quantifier la qualité du chemin. Nous sélectionnons ensuite le chemin correspondant au coût global le plus faible pour construire la chaîne polygonale optimale.

Pour tous les points du polygone

- **Pour** chaque position du point dans la fenêtre de voisinage
 - Pour toutes les positions du point précédent du polygone
Évaluer le coût du segment généré
 - Mémoriser la position du point précédent correspondant au coût minimum
 - Mise à jour du coût du point en cours

fin Pour

fin Pour

Rechercher le coût global minimum

Parcours en sens inverse pour récupérer le contour final

FIG. 6.12 – *Algorithme de Programmation Dynamique* : Cet algorithme évalue le coût de toutes les positions admissibles des segments de la chaîne initiale. Chaque sommet peut se déplacer dans un voisinage autorisé. On calcule le coût pour tous les chemins possibles, puis on sélectionne celui correspondant au coût global le plus faible pour construire la chaîne solution.

Le temps de calcul demandé par ce type d'algorithme combinatoire est beaucoup plus important que pour une technique basée sur les contours actifs. Pour étudier les solutions dans un voisinage d'étude étendu et préserver un temps de calcul faible, nous proposons une approche hiérarchique en deux étapes. Nous utilisons deux fenêtres de voisinage différentes comme le présente la figure 6.13, une pour chaque étape. La première fenêtre 6.13 A est de taille 13×13 et centrée sur la position du sommet initial. Cette étape permet d'approcher la solution. Les croix représentent les positions admissibles du sommet initial dans son voisinage, la fonction de coût sera estimée pour chacune de ces positions. La seconde fenêtre 6.13 B est de taille 5×5 et centrée sur la position du sommet correspondant au meilleur coût obtenu lors l'étape d'approche. Cette seconde étape permet d'obtenir la solution finale du problème. Le temps de calcul est approximativement le même pour ces deux étapes puisque chaque sommet est associé à 25 positions admissibles.

A chaque résolution d'étude, l'algorithme doit évaluer le coût de tous les chemins possibles. Nous utilisons l'algorithme de *Viterbi* pour obtenir la solution de coût cumulé minimale. Cet algorithme repose sur la décomposition récursive de la fonction de coût total.

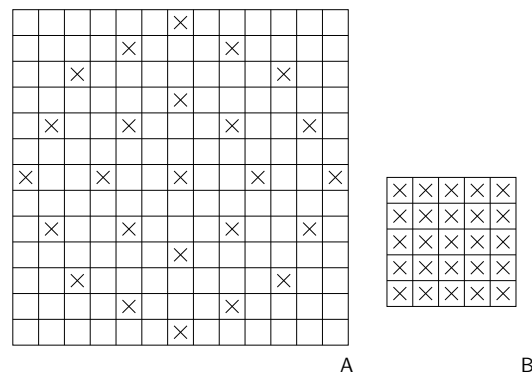


FIG. 6.13 – Fenêtres de voisinage. La fenêtre A définit le voisinage utilisé pour approcher la position du sommet, elle est centrée sur le sommet initial. Sa taille est importante mais on n'évalue le coût que pour les points du voisinage désignés par une croix. La seconde fenêtre B est utilisée pour obtenir la solution finale. Celle-ci définit le voisinage autour du sommet correspondant à la solution approchée. Sa taille est plus petite mais tous les points du voisinage sont évalués. 25 positions sont évaluées pour chacune de ces fenêtres.

Lorsque la fonction de coût utilisée est définie comme l'intégrale d'un coût local le long de la courbe, cet algorithme garantit l'optimalité de la solution calculée dans le domaine exploré. On remarque que l'on peut facilement imposer des contraintes ou créer des pénalités comme le font les algorithmes de mise en correspondance par programmation dynamique.

La figure 6.14 présente l'algorithme de *Viterbi*. Cet algorithme utilise des tables de coûts associées à chaque segment potentiel de la courbe polygonale. Les segments de la chaîne sont traités un par un. On fait varier la position du premier point du segment P_i dans la fenêtre de voisinage choisie (on le déplace sur les positions (*pos* 1, ..., *pos* p) représentant les positions admissibles), le second point P_{i+1} reste fixe. Pour chaque segment ainsi créé, on calcule son coût local. On conserve en mémoire dans la table lui correspondant la position du premier sommet donnant le coût minimum C_m . On mémorise ensuite le coût intermédiaire C_{i+1} en ajoutant au coût du point précédent, C_i , celui associé au segment choisi C_m . On réalise cette opération pour toutes les positions (*pos* 1, ..., *pos* p) du second point du segment en cours. On continue le long du contour en prenant en compte les coûts obtenus pour les segments précédents comme le montre la figure 6.14.

Chaque point du tableau associé à un point du contour pointe vers son prédécesseur optimum. Une fois tous les coûts déterminés, on reconstruit le contour final en choisissant le chemin qui possède le coût minimal, pris dans le dernier tableau. On part de ce point qui pointe vers son prédécesseur optimal et ainsi de suite pour obtenir la chaîne polygonale finale détournant le plan. Puisque le contour est fermé, l'algorithme doit aussi prendre en compte le segment $P_m P_1$. Nous proposons d'utiliser le même algorithme en ajoutant les deux premiers sommets de la chaîne polygonale à la fin de liste. La chaîne finale est obtenue en supprimant les premier et dernier points de la chaîne.

La fonction de coût

Nous avons choisi deux critères pour définir notre fonction de coût locale $f_{P_i P_{i+1}}$ le long du segment $P_i P_{i+1}$. Le premier critère est de maximiser la somme des intensités du gradient le long du contour. Le second critère est relatif à l'orientation du contour, nous souhaitons

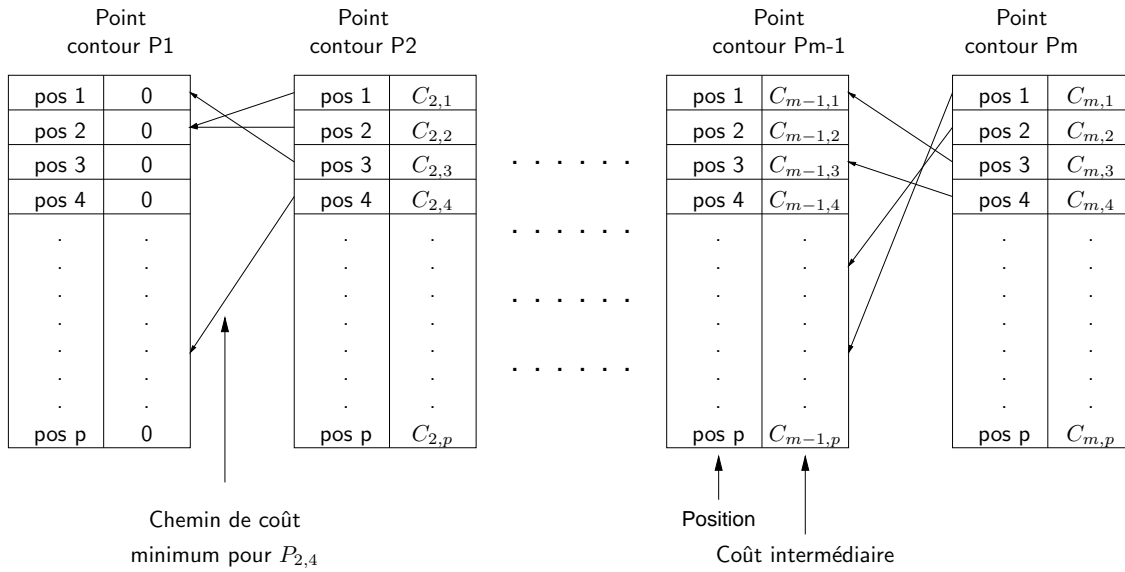


FIG. 6.14 – *Algorithme de Viterbi*: On utilise des tables des coûts pour retrouver la solution du problème. On calcule le coût de chaque segment $P_i P_{i+1}$ pour toutes les positions de ces deux points. On conserve dans le tableau associé au point P_{i+1} la position du premier point P_i correspondant au meilleur segment pour toutes les positions (pos 1, ..., pos p) du premier point. On conserve dans la même case du tableau le coût intermédiaire C_{i+1} correspondant à la somme du coût au point P_i choisi, C_i et du coût local du segment choisi. On retrouve le contour en parcourant les tables en sens inverse, partant du coût total le plus faible pris dans le dernier tableau.

obtenir des segments parallèles aux contours des différentes images. Celui-ci a été ajouté pour améliorer la qualité des résultats. La fonction de coût que nous utilisons le long du segment $P_i P_{i+1}$ est la suivante :

$$f_{P_i P_{i+1}} = \frac{1}{n} \sum_{k=0}^n \left(\frac{1}{n_k} \sum_{j=0}^{n_k} \frac{1 + \cos^2 \alpha_j}{1 + |\nabla I_k|} \right) \quad (6.4)$$

i décrit les sommets, k décrit les images et j les n_k points du segment projeté dans l'image k . ∇I_k est l'image de gradient correspondant à l'image I_k , la valeur du gradient que nous utilisons est codée entre 0 et 255. Nous utilisons une fonction de coût moyennée sur le segment pour ne pas favoriser les contours les plus courts lors de l'optimisation. La figure 6.15 nous présente le coût en un point dans une des images initiales. On mesure tout d'abord l'intensité du gradient au point courant. On évalue ensuite le cosinus de l'angle α_j formé par la normale du segment et la direction du contour de manière à ce que cette valeur soit petite lorsque le segment est tangent au contour. Le coût est évalué sur la totalité du segment pour limiter l'influence du bruit.

Résultats de recalage par programmation dynamique

Comme pour l'algorithme de recalage par contours actifs, l'algorithme de recalage par programmation dynamique que nous avons développé traite les modèles de plans individuellement. Le modèle polygonal initial se déplace aussi dans l'espace tridimensionnel tandis que

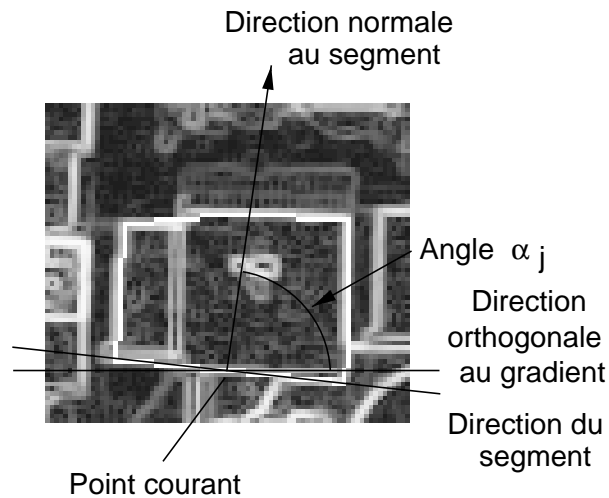


FIG. 6.15 – Mesure de l'angle utilisé pour calculer le coût local du point courant de la chaîne.

le calcul du coût est obtenu à partir des images. Les positions des sommets sont définies par la fenêtre de voisinage. Le voisinage est considéré dans le plan horizontal, le contour est aussi contraint à se déplacer dans le plan. Pour chaque segment déterminé par l'algorithme de programmation dynamique, on projette ses deux extrémités dans les images. On somme alors pour tous les points des segments projetés la fonction de coût locale calculée (équation 6.4). La solution est ensuite obtenue par l'algorithme de *Viterbi*.

La fonction de coût proposée pour l'algorithme de programmation dynamique cherche à la fois à maximiser l'intensité de la norme du gradient le long du contour et à obtenir un contour parallèle au segment. Les images utilisées pour ce recalage sont présentées dans la figure 6.16. Nous utilisons une image de la norme du gradient (A) et les dérivées premières des images initiales (B et C) pour déterminer en chaque point la direction du gradient qui est perpendiculaire au contour. Contrairement aux images utilisées par les contours actifs, ces images ne sont pas lissées pour avoir un profil du pic d'intensité de la norme du gradient de forme plus étroite et permettre à l'algorithme de mieux localiser le contour.

Les résultats de l'algorithme de recalage par programmation dynamique sont présentés dans la figure 6.17. Les résultats sont présentés pour les deux mêmes plans du bâtiment utilisés pour montrer les résultats du recalage par contours actifs. On projette les contours obtenus sur l'ortho-image de la zone d'intérêt. Les images A et D présentent les deux contours initiaux obtenus par l'algorithme d'approximation polygonale. Les images B et E présentent le contour obtenu en utilisant l'algorithme de recalage par programmation dynamique avec une seule vue de la scène. Les images C et F présentent le contour obtenu en utilisant l'algorithme de recalage par programmation dynamique avec les douze vues disponibles. Les résultats obtenus en utilisant une seule image sont beaucoup moins bons que lorsqu'on utilise l'algorithme des contours actifs (figure 6.11). D'importantes erreurs surviennent lorsqu'il y a un autre contour important proche des limites des bâtiments. En utilisant les douze images, la différence est moins importante, les contours ont bien été recalés sur les contours réels du bâtiment même si quelques petites erreurs sont encore visibles. Les contours ne sont pas totalement rectilignes le long des façades.

Les erreurs commises par l'algorithme de programmation dynamique peuvent être expliquées simplement. Contrairement au snake, la forme globale du contour peut être fortement

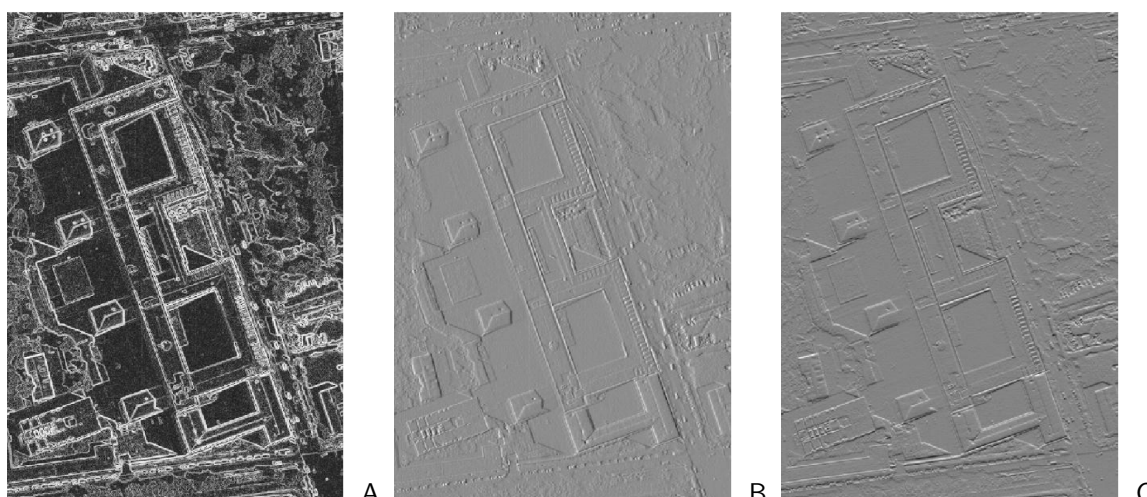


FIG. 6.16 – *Les images de gradient utilisées pour le recalage par programmation dynamique : A représente la norme du gradient. B et C représentent respectivement l'image des dérivées en x et l'image des dérivées en y de l'image initiale. Contrairement aux images utilisées par les contours actifs, celles-ci ne sont pas lissées.*

déformée et assez différente du contour initial. Le snake se déplace continûment vers le contour le plus proche tandis qu'avec la programmation dynamique les sommets des segments peuvent faire des "bonds" dans leurs voisinages et ainsi changer de contour d'image ou générer des contours de topologie différente (croisements).

Pour améliorer les résultats, nous avons tout d'abord ajouter à la fonction de coût le critère d'orientation. Nous avons ensuite introduit un système de pénalité visant à limiter les erreurs commise lorsque l'algorithme change de contour. Le coût local d'un segment projeté dans une des images est multiplié par 10 lorsque le segment est trop court ou lorsque l'angle défini par trois sommets consécutifs est inférieur à 30° . Nous pouvons voir toutefois que les résultats restent beaucoup plus instables que ceux obtenus avec les contours actifs.

6.2.4 Corrections géométriques et topologiques

Nous avons vu dans les sections précédentes des méthodes de correction de la chaîne polygonale utilisant les informations des images initiales de la scène. Le principe de ces méthodes est de recalcr cette chaîne sur les contours des images. Il existe d'autres méthodes de correction qui sont basées uniquement sur les propriétés géométriques et topologiques de la chaîne. Le principe est d'imposer des contraintes de régularité au modèle qui découlent des connaissances disponibles sur la scène.

Plusieurs méthodes de ce type ont déjà été proposées, ce sont des méthodes de simplification de polygones. *U. Weidner* et *W. Förstner* proposent dans [WF95] une méthode de simplification basée sur le critère du MDL. Les auteurs définissent dix configurations possibles de sous-ensembles de chaîne pour lesquelles les coûts sont pré-calculés. Ces configurations tendent à imposer des contraintes d'orthogonalité. La solution choisie est celle possédant la longueur de description minimale. La technique proposée est rapide et permet le changement de topologie mais ne permet pas de retrouver le véritable détournage du bâtiment comme le montrent les résultats présentés dans leur article. Nous proposons une

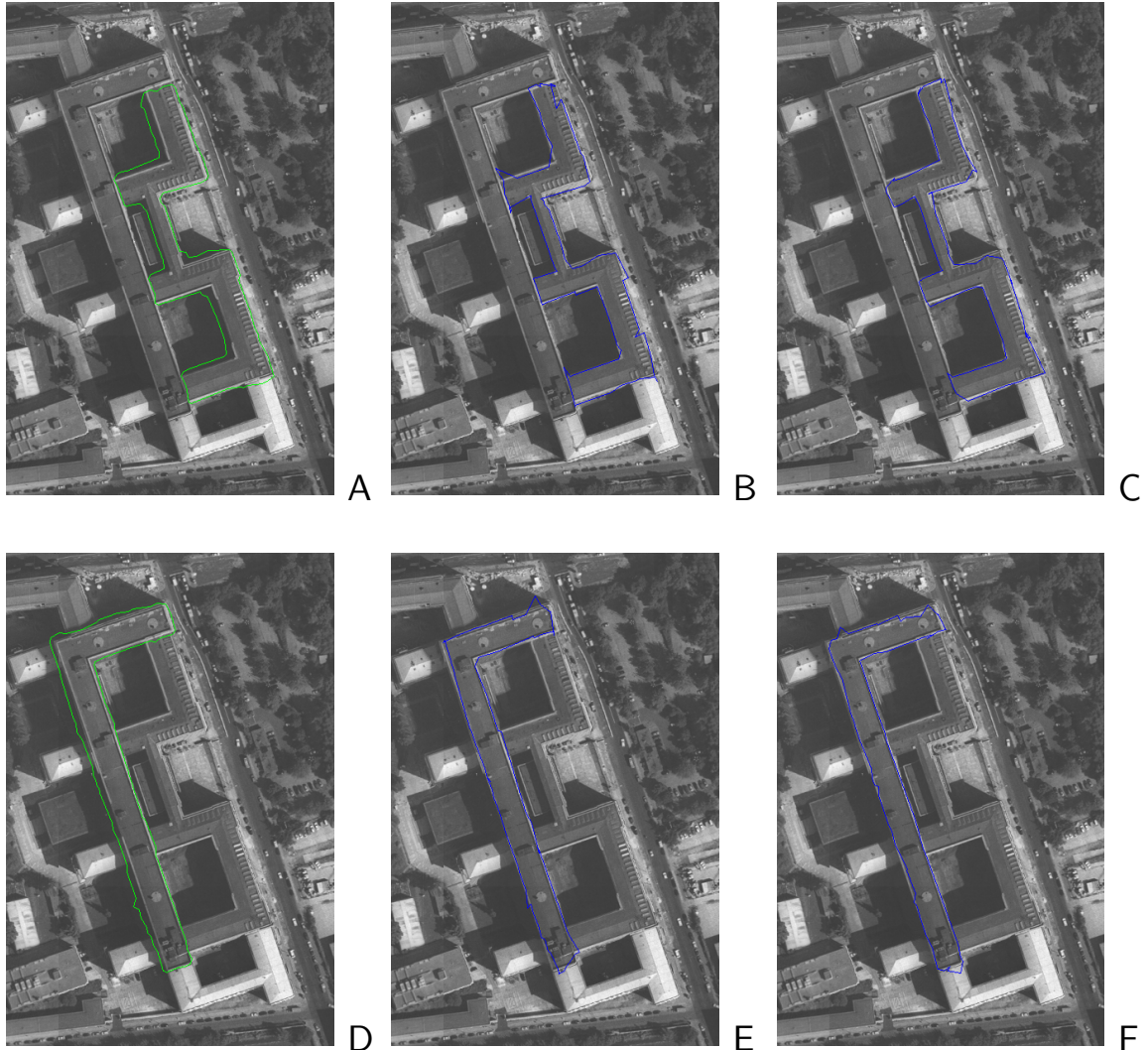


FIG. 6.17 – Résultat du recalage des contours par programmation dynamique : (A, B, C) et (D, E, F) présentent les résultats pour deux modèles de plan du bâtiment superposés à l'ortho-image de la zone d'intérêt. A et D présentent le contour initial obtenu par l'algorithme d'approximation polygonale. B et E présentent le contour obtenu après recalage en utilisant une vue de la scène. C et F présentent le contour obtenu après recalage en utilisant toutes les vues disponibles sur cette scène, 12 images. En utilisant une seule image, les résultats sont beaucoup moins bons que l'utilisation des contours actifs (figure 6.11). En utilisant les douze images, la différence est moins importante, les contours ont bien été recalés sur les contours réels du bâtiment.

nouvelle stratégie d'orthogonalisation du modèle polygonal basée sur les techniques robustes.

Réajustement du modèle avec contraintes angulaires

L'algorithme d'orthogonalisation du modèle polygonal que nous avons développé est présenté dans la publication suivante [VDM00]. Cet algorithme utilise le modèle polygonal du bâtiment obtenu après l'étape de construction par approximation polygonale. Durant l'extraction des plans, nous avons supposé ne disposer d'aucune connaissance a priori sur la forme des contours qui détournent les différentes faces du modèle de bâtiments. Après l'étape de polygonalisation, nous obtenons un modèle polygonal avec des angles quelconques. Dans les environnements construits par l'homme comme l'environnement urbain, les angles droits et les angles plats sont souvent présents. Nous proposons un algorithme qui essaie d'imposer ces contraintes sur tous les angles du modèle de bâtiment.

Le modèle polygonal initial du bâtiment Ω_i est composé de segments qui sont liés par les jonctions du modèle ou les sommets des chaînes polygonales. Parce que nous souhaitons préserver une consistance globale du modèle, la stratégie d'orthogonalisation doit être appliquée sur le modèle complet. Nous proposons de résoudre ce problème d'orthogonalisation du modèle par l'optimisation d'une fonction d'objectif \mathbf{O} . La meilleure solution Ω_b correspond au minimum de la fonction d'objectif suivante :

$$\Omega_b = \min \mathbf{O}(\Omega) = \min (\mathbf{A}(\Omega) + \mathbf{I}(\Omega)) \quad (6.5)$$

Cette fonction d'objectif est définie par deux composantes : une composante \mathbf{A} qui contraint les angles à devenir des angles droits ou des angles plats et une composante \mathbf{I} qui est un terme d'attache aux données initiales.

La première composante donne la priorité aux angles de 90° et de 180° . Nous utilisons la fonction $\sin(2\alpha)$ pour évaluer la pondération associée à l'angle α du modèle polygonal. Cette fonction est minimale pour les angles que nous souhaitons privilégier (0° MODULO 90°). Nous utilisons une variable d'angle pour chaque sommet de la chaîne polygonale, deux variables d'angles pour une jonction triple, etc . . . Nous fixons les jonctions simples car elles correspondent aux bords de l'image.

Considérons Γ l'ensemble de toutes les variables d'angle utilisées par le modèle polygonal Ω , nous définissons la composante \mathbf{A} comme suit :

$$\mathbf{A}(\Omega) = \sum_{\alpha \in \Gamma_\Omega} W_1 \cdot \sin(2\alpha) \quad (6.6)$$

La composante \mathbf{A} permet de forcer les angles du modèle polygonal vers les angles que nous privilégions.

L'algorithme d'orthogonalisation utilise seulement un modèle polygonal comme donnée initiale. Nous avons donc besoin de définir une composante qui permette d'attacher le modèle résultat aux données initiales et d'éviter les déformations trop importantes du modèle polygonal Ω . Considérons Λ l'ensemble des points du modèle polygonal Ω (jonctions et sommets), nous définissons la composante \mathbf{I} comme suit :

$$\mathbf{I}(\Omega) = \sum_{\beta \in \Lambda_\Omega} W_2 \cdot |\beta_i - \beta| \quad (6.7)$$

avec $\beta_i \in \Lambda_{\Omega_i}$

β est un point du modèle polygonal courant et β_i est le même point dans le modèle initial. W_1 et W_2 sont deux pondérations qui permettent de contrôler l'influence des deux composantes de notre fonction d'objectif \mathbf{O} . Nous choisissons de prendre :

$$\begin{aligned} W_1 &= 1/\sin(10^\circ) \\ W_2 &= 1/\varepsilon_M \end{aligned} \tag{6.8}$$

ε_M est le seuil utilisé lors de l'étape de **merge** de l'algorithme de polygonalisation de contour. Ces choix permettent d'avoir le même coût entre un déplacement de la chaîne polygonale de la distance ε_M par rapport au modèle initial et une différence angulaire de 10° .

Parce que le modèle initial est proche de la solution, nous utilisons un **M-estimateur** avec une pondération de *Tuckey* (section 5.3.1). Après l'optimisation, nous utilisons une étape de **merge** pour éliminer des configurations critiques de la chaîne polygonale dues à l'application des contraintes, ce sont principalement des points de rebroussement.

6.2.5 Résultats

La figure 6.18 présente les résultats de l'algorithme d'orthogonalisation du modèle polygonal. Les résultats sont présentés pour trois bâtiments de formes différentes, A, B et C. Pour chaque bâtiment étudié, l'image de gauche présente le modèle polygonal initial construit par l'algorithme d'approximation polygonale. L'image de droite présente le modèle obtenu après application de l'algorithme d'orthogonalisation. Nous pouvons voir que l'algorithme a permis de retrouver la plupart des angles droits et des angles plats du modèle de bâtiment. La forme générale du bâtiment est préservée. L'utilisation d'une technique robuste dans la stratégie d'orthogonalisation permet d'assurer la stabilité de l'algorithme.

Programmation et paramètres importants Les algorithmes d'approximation polygonale et d'orthogonalisation ont tout d'abord été développés par *M. Mesnage* au cours d'un stage d'ingénieur dirigé par *F. Devernay* et co-encadré par moi-même. Les algorithmes de recalage par contours actifs et par programmation dynamique ont été développés au cours de cette thèse. Tous ces algorithmes sont en C++. Des scripts PERL permettent de faciliter l'utilisation de l'ensemble des programmes en regroupant les commandes nécessaires.

Nous n'utilisons qu'un seul paramètre dans l'algorithme d'approximation polygonale ε_M . Ce paramètre est utilisé à la fois pour les étapes de **split** et de **merge**. Nous lui attribuons une valeur de $1,5m$. Augmenter ou diminuer sa valeur permet de contrôler le degré de simplification souhaité.

Un seul paramètre est utilisé dans l'algorithme de recalage par contours actifs, la viscosité γ . Ce paramètre est utilisé pour contrôler la vitesse de déplacement du contour. Cette vitesse doit être suffisamment faible pour que celui-ci converge vers le minimum le plus proche (qui correspond au contour réel du bâtiment si l'initialisation est correcte).

L'algorithme de programmation dynamique dépend de plusieurs paramètres. le premier est la taille des fenêtre de voisinages utilisées. Le choix dépend de la confiance associée aux sommets du contour initial. Plus le contour sera proche de la solution et plus le voisinage utilisé pourra être restreint. Nous utilisons une fenêtre de taille 13×13 pixels². Les autres paramètres que nous utilisons sont les pénalités qui permettent d'interdire certaines configurations peu probables. Le choix s'effectue empiriquement.

L'algorithme d'orthogonalisation n'utilise que deux paramètres, les pondérations W_1 et W_2 . Leurs valeurs permettent de contrôler l'influence entre la composante d'attache aux

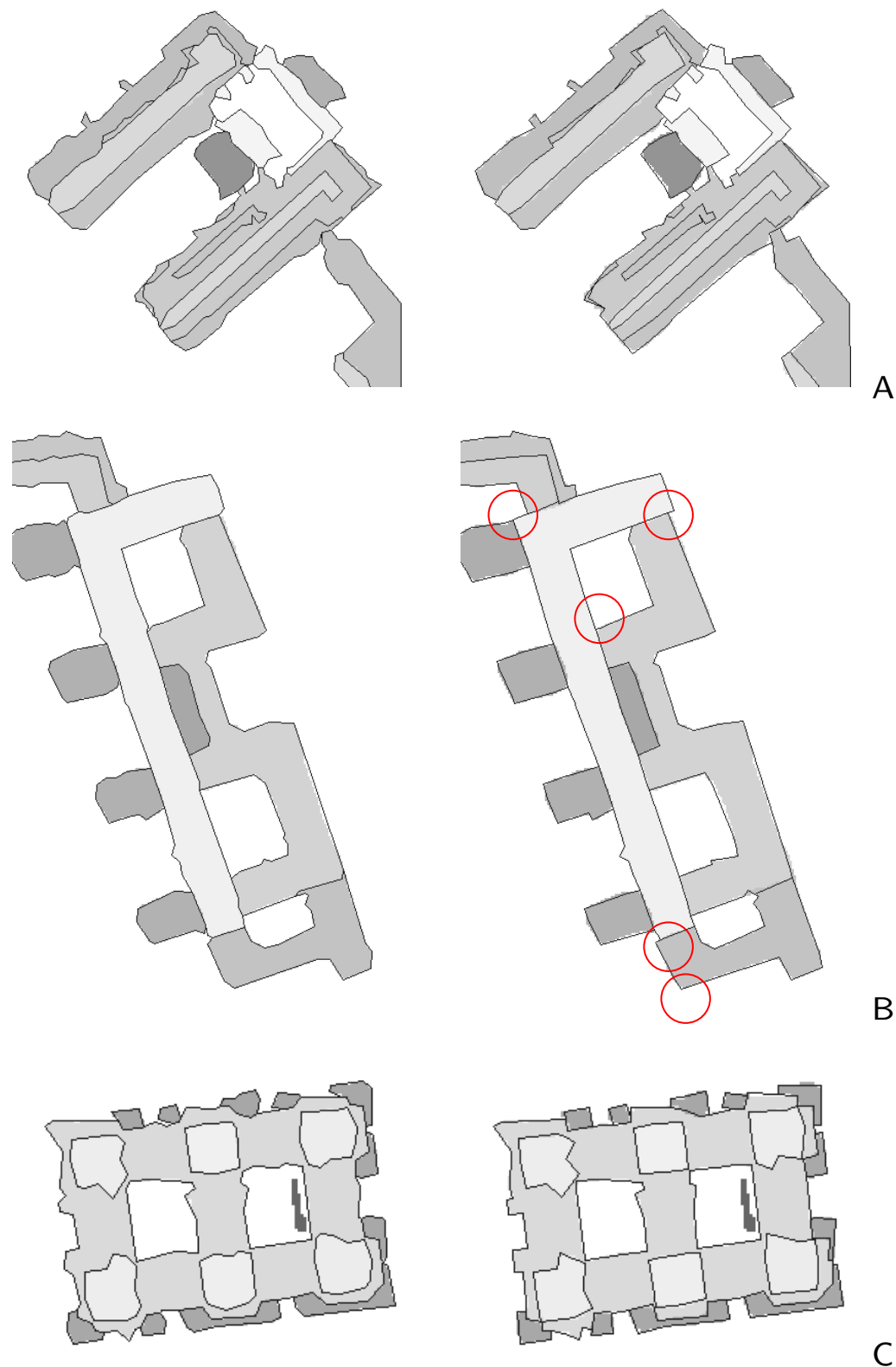


FIG. 6.18 – *Orthogonalisation du modèle polygonal: A, B et C présentent les résultats de l'algorithme sur trois bâtiments de formes différentes. L'image de gauche présente le modèle initial superposé au résultat de l'algorithme de segmentation. L'image de droite présente le modèle obtenu par l'algorithme de correction proposé. Les cercles en rouge présentent les améliorations importantes.*

données et la composante qui contraint les angles à devenir droits ou plats. Nous choisissons $W_1 = 1/\sin(10^\circ)$ et $W_2 = 1/\varepsilon_M$ pour avoir le même coût entre un déplacement de la chaîne polygonale de la distance ε_M par rapport au modèle initial et une différence angulaire de 10° . ε_M est le paramètre utilisé pour l'approximation polygonale.

6.3 Conclusion

Nous avons décrit dans ce chapitre plusieurs algorithmes permettant de construire ou de corriger un modèle polygonal de bâtiment. Nous avons tout d'abord présenté un algorithme d'approximation polygonale basé sur la méthode du *split and merge*. Cet algorithme est utilisé pour construire le modèle polygonal du bâtiment. Il permet de représenter correctement les bords des objets segmentés, premier critère de qualité que nous avons défini. Nous avons aussi introduit une étape de correction des coins pour limiter la déformation des angles et répondre au mieux au second critère de qualité.

A cause des imperfections présentes dans les modèles de plans utilisés pour l'approximation polygonale, il existe un décalage important entre le modèle polygonal du bâtiment et les contours des images correspondant aux limites réelles du bâtiment. Les bords du modèle ne sont pas rectilignes et les angles sont arrondis. Nous avons présenté et étudié trois algorithmes permettant la correction du modèle. Les deux premiers algorithmes procèdent au recalage des contours sur les images. L'algorithme de recalage par contours actifs utilise un contour actif polygonal et l'optimisation est multi-image. Le second algorithme est basé sur la programmation dynamique, le calcul du coût est aussi réalisé avec toutes les images disponibles. Nous avons vu que les résultats obtenus avec l'algorithme des contours actifs sont meilleurs, le contour initial étant assez proche de l'objet cherché. Ces deux approches ont fait l'objet de la publication suivante [VBD97]. Nous avons enfin proposé un algorithme de correction du modèle polygonal complet. Il permet d'imposer la contrainte d'orthogonalité aux sommets du modèle lorsque c'est nécessaire. Cet algorithme a fait l'objet de la publication suivante [VDM00].

Chapitre 7

Résultats

Nous avons développé dans les chapitres précédents plusieurs outils liés à la reconstruction des bâtiments en imagerie aérienne. Chacun d'eux traite une partie du problème. Nous présentons dans ce chapitre les résultats obtenus. L'objectif final visé est de combiner ces outils en vue de développer un système de modélisation des bâtiments robuste, cohérent et de qualité constante.

Dans une première partie, nous présentons le cadre des études. Nous utilisons une zone d'intérêt de taille élevée. Dans la deuxième partie, nous proposons un système de modélisation et présentons ses résultats sur la zone d'intérêt définie. Avec le but de valider les modèles obtenus, nous présentons ensuite les résultats d'une première étude qualitative basée sur des outils télécoms.

Nous présentons dans la troisième partie d'autres résultats, obtenus avec les outils que nous avons développés au cours de cette thèse : reconstruction des points des façades, segmentation de MNE en plans, approximation polygonale des contours, recalage de contour par snake, recalage de contour par programmation dynamique et orthogonalisation d'un modèle polygonal.

7.1 Le cadre

Pour présenter les résultats, nous utilisons un chantier aérien de 24 images sur la ville de Berlin en Allemagne. La scène correspond à une zone urbaine assez dense. La disposition des acquisitions correspond à une grille régulière comme présenté dans la figure 3.1 (6×4 images). Cette disposition permet d'exploiter au mieux la forme carrée des images et d'avoir une redondance d'informations altimétriques. Les recouvrements des images sont de l'ordre de 80% le long de la trace et perpendiculairement à cette direction.

L'image correspondant à chaque prise de vue est obtenue en numérisant un film de 9 pouces par 9 pouces avec un pas de $25\mu m$. La focale est d'environ $150mm$ et la hauteur de l'acquisition de $2300m$, ce qui correspond à des images au $1/15000$. La résolution planimétrique des images numérisées est de $37.5cm$, elle est suffisante pour retrouver les principales composantes de la forme d'un bâtiment.

Nous présentons les résultats sur une zone d'intérêt qui possède une taille de $1km \times 1km$. La résolution des Modèles Numériques que nous utilisons est de $50cm$. Le MNE brut correspondant obtenu par la chaîne de production automatique d'ISTAR est présenté dans la figure 7.1, il possède une taille de 2000×2000 pixels². L'ortho-image correspondant à cette zone d'intérêt est présentée dans la figure 7.2.



FIG. 7.1 – *MNE brut de la scène étudiée : Ce MNE a été reconstruit en utilisant les 24 images disponibles sur le chantier. Dans une première phase, toutes les paires d'images adjacentes sont appariées, elles permettent la construction de MNE locaux. Le MNE brut présenté dans cette figure est obtenu par fusion robuste de tous ces MNE locaux.*



FIG. 7.2 – *Ortho-image de la scène étudiée.*

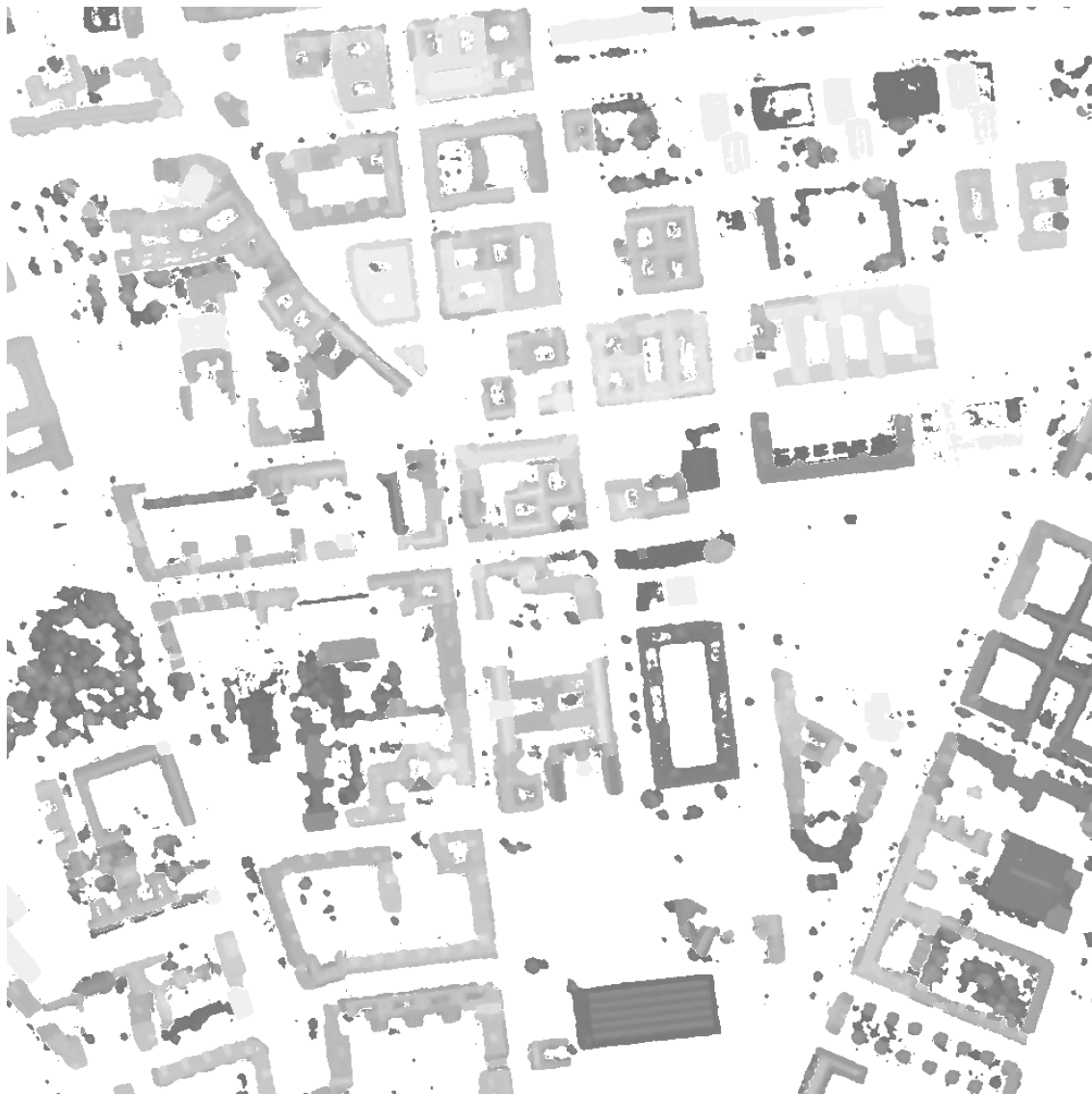


FIG. 7.3 – MNE obtenu après l'extraction automatique du sol. Le sol est extrait en soustrayant le MNT de la zone au MNE brut (figure 7.1), puis en éliminant les régions restantes pas assez élevées et/ou de faible surface.

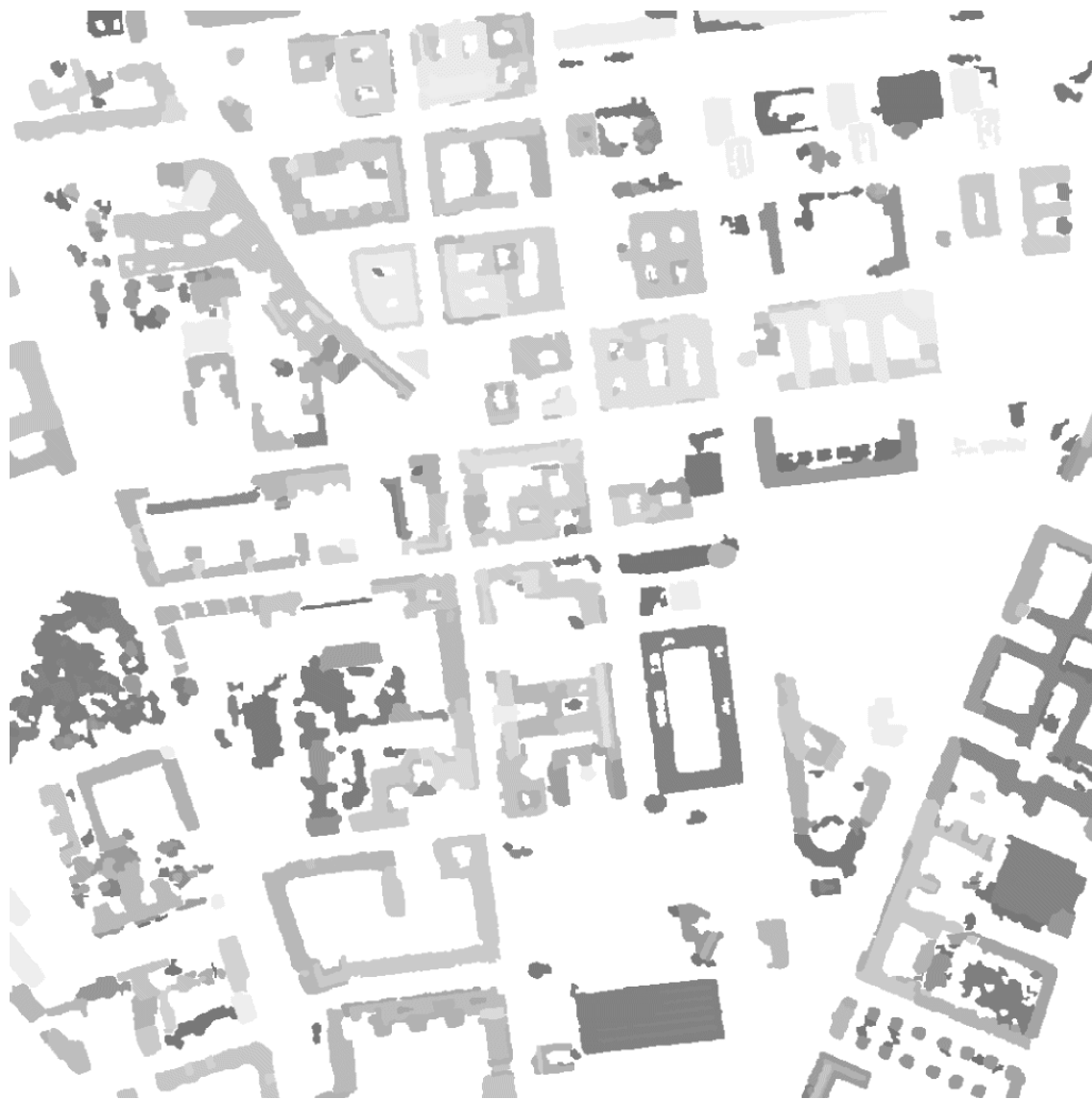


FIG. 7.4 – *Résultat de la segmentation en plans horizontaux : L'algorithme de segmentation de MNE retrouve la plupart des régions planaires des toits des bâtiments.*

7.2 Système de modélisation proposé

La section 3.4 présentait les objectifs que nous nous étions fixés pour la reconstruction des bâtiments, nous les rappelons : détecter tous les bâtiments, traiter toutes les formes de bâtiment, obtenir un modèle représentatif de la forme générale du bâtiment et avoir un détournement du bâtiment de bonne qualité. Pour répondre à ces objectifs, nous avons opté pour une stratégie générale (expliquée en détail dans la section 3.6) en trois étapes : la construction d'un MNE dense et fiable, la segmentation des bâtiments en régions planes et la vectorisation des contours des plans pour obtenir la description finale du bâtiment.

Pour la première étape, nous utilisons la chaîne de production de MNE d'ISTAR qui permet de construire un MNE dense de la scène. Nous proposons ensuite d'utiliser l'algorithme de segmentation proposé dans le chapitre 5. Nous effectuons une segmentation du MNE initial en plans horizontaux pour retrouver les régions planaires qui permettent de décrire les toits des bâtiments. La troisième étape, la vectorisation, se compose des deux processus suivants : un processus d'approximation polygonale des contours des plans pour construire un premier modèle polygonal de chaque bâtiment, puis un processus d'orthogonalisation des modèles des bâtiments pour obtenir la description finale de la scène. Chacun de ces processus a été présenté dans le chapitre 6

Les algorithmes de segmentation et de vectorisation doivent être appliqués localement, nous voulons traiter les bâtiments un à un. Pour cela, nous utilisons le MNE dans lequel le sol a été supprimé pour pouvoir retrouver les régions correspondant aux bâtiments. Nous extrayons le sol en utilisant le MNT de la zone, comme nous l'avons présenté en section 4.5.2. Le MNE ainsi obtenu est présenté dans la figure 7.3.

Nous pouvons voir dans l'image de la figure 7.3 que certaines régions correspondant à de la végétation sont encore présentes (à gauche dans l'image). L'algorithme d'extraction du sol développé n'utilise que l'information de hauteurs, il extrait donc le sur-sol de la zone d'intérêt. Cette information n'a pas suffi à différencier les bâtiments de la végétation. Cet algorithme peut être amélioré en utilisant certaines propriétés de la surface comme *C. Baillard* dans [Bai97], ou en utilisant un autre type d'imagerie comme *A. Huertas, Z. Kim et R. Nevatia* dans [HKN90]. L'extraction obtenue est suffisante pour nos études, l'objectif étant la détection de tous les bâtiments. Nous allons présenter dans la suite les résultats des étapes de segmentation et de vectorisation.

7.2.1 Segmentation du MNE

L'objectif de cette étape est de décrire la scène urbaine au moyen de morceaux de surface qui vont correspondre aux différentes faces des bâtiments. Le principe de cet algorithme repose sur deux phases. La première consiste à générer une liste d'hypothèses de modèles de plans à partir du MNE traité. La seconde phase consiste à sélectionner les meilleurs modèles de la liste pour décrire le bâtiment.

Nous appliquons cet algorithme sur le MNE de la zone d'intérêt que nous avons définie. Les résultats sont présentés dans la figure 7.4. L'algorithme de segmentation de MNE retrouve presque toutes les régions planaires des toits des bâtiments. La qualité du résultat de segmentation dépend de la qualité du MNE utilisé. Nous avons utilisé un intervalle de tolérance de $\pm 2m$ pour chaque modèle de plan. Compte tenu de la résolution et de la qualité du MNE, il serait difficile de descendre en dessous de cette valeur, les résultats ne seraient alors plus représentatifs. Si l'on souhaite obtenir uniquement le détournement des bâtiments, on peut alors augmenter considérablement la valeur de ce seuil.

La scène est composée de 134 bâtiments. La surface moyenne des bâtiments étudiés est de 2080m². La surface moyenne des 582 régions segmentées qui décrivent la scène est de 529m².

7.2.2 Modèle polygonal du bâtiment

Construction du modèle

La construction du modèle polygonal est effectuée à partir du jeu de plans construits par l'algorithme de segmentation. Les plans utilisés sont tous horizontaux. Chaque plan étant associé à l'une des faces du modèle de l'objet, nous réalisons une approximation des détourages de chacun des plans pour obtenir les chaînes polygonales du modèle.

L'algorithme que nous avons proposé dans la section 6.1.2 est basé sur la méthode du *split and merge*. Nous avons amélioré cet algorithme en lui associant l'estimateur de la moindre médiane des carrés (LMS) et avons ajouté une étape de correction des coins. Le LMS est utilisé pour déterminer les segments de la chaîne polygonale, il permet de déterminer les bords des objets segmentés. L'étape de correction des coins permet de limiter les déformations présentes sur les angles des bâtiments.

Les résultats de notre algorithme d'approximation polygonale sont présentés dans la figure 7.5. Nous avons dit dans la section 6.1.1 qu'il était nécessaire de préserver les bords et angles de l'objet pour obtenir une bonne modélisation. Les bâtiments sont extraits puis traités un à un. Nous voyons que la forme générale de chaque bâtiment est respectée. Les segments des modèles polygonaux obtenus représentent correctement les bords de chacun d'eux, le premier critère de qualité du modèle est bien respecté. Pour les angles, l'étape de correction des coins a permis de préserver certains des angles des bâtiments. Beaucoup d'autres angles restent déformés à cause de la forme lissée des bâtiments présents dans le MNE initial. Ni l'algorithme de segmentation, ni l'algorithme d'approximation polygonale ne corrigent ces déformations.

La figure 7.7 présente le résultat de vectorisation (même modèle que 7.5) superposé cette fois-ci sur le MNE synthétique correspondant au résultat de segmentation. Le modèle représente parfaitement le résultat de la segmentation. L'algorithme d'approximation polygonale permet de construire un modèle polygonal de chaque bâtiment à partir des plans. Tout comme l'algorithme de segmentation, il permet de modéliser le bâtiment à partir du MNE initial mais ne corrige pas les imperfections présentes dans celui-ci.

Modèle avec contraintes angulaires

Pour corriger une partie des imperfections présentes dans le modèle polygonal construit, nous avons proposé dans la section 6.2.4 de lui appliquer des contraintes angulaires. Durant l'extraction des plans par l'algorithme de segmentation, nous avons supposé ne disposer d'aucune connaissance a priori sur la forme des contours des bâtiments. Nous avons donc proposé un algorithme qui permet d'imposer des angles droits et des angles plats dans les parties du modèle où cela est nécessaire.

Pour préserver une homogénéité globale du modèle, la stratégie d'orthogonalisation est appliquée sur le modèle complet de chaque bâtiment, traité encore une fois indépendamment pour diminuer les temps de calcul. Nous avons proposé de résoudre ce problème d'orthogonalisation du modèle par l'optimisation d'une fonction d'objectif et l'utilisation d'un estimateur robuste, le *M-estimateur*. Cet estimateur permet d'assurer la robustesse de l'algorithme.

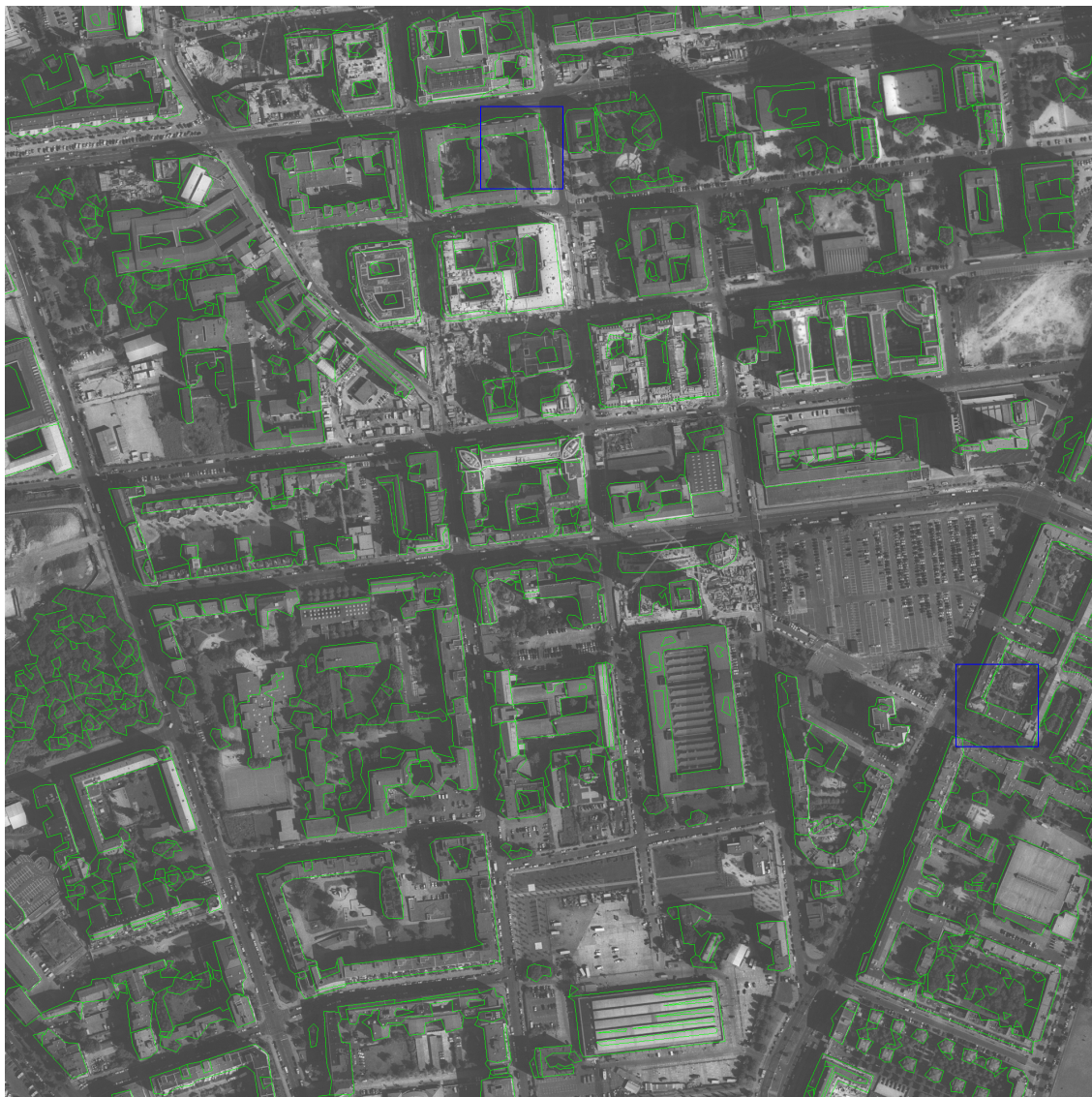


FIG. 7.5 – Construction du modèle polygonal de chaque bâtiment : L'algorithme d'approximation polygonale est appliqué sur chaque plan horizontal proposé par l'algorithme de segmentation automatique. Les modèles obtenus (en vert) sont présentés superposés à une ortho-image. Ces modèles représentent correctement la forme générale des objets qu'ils décrivent.

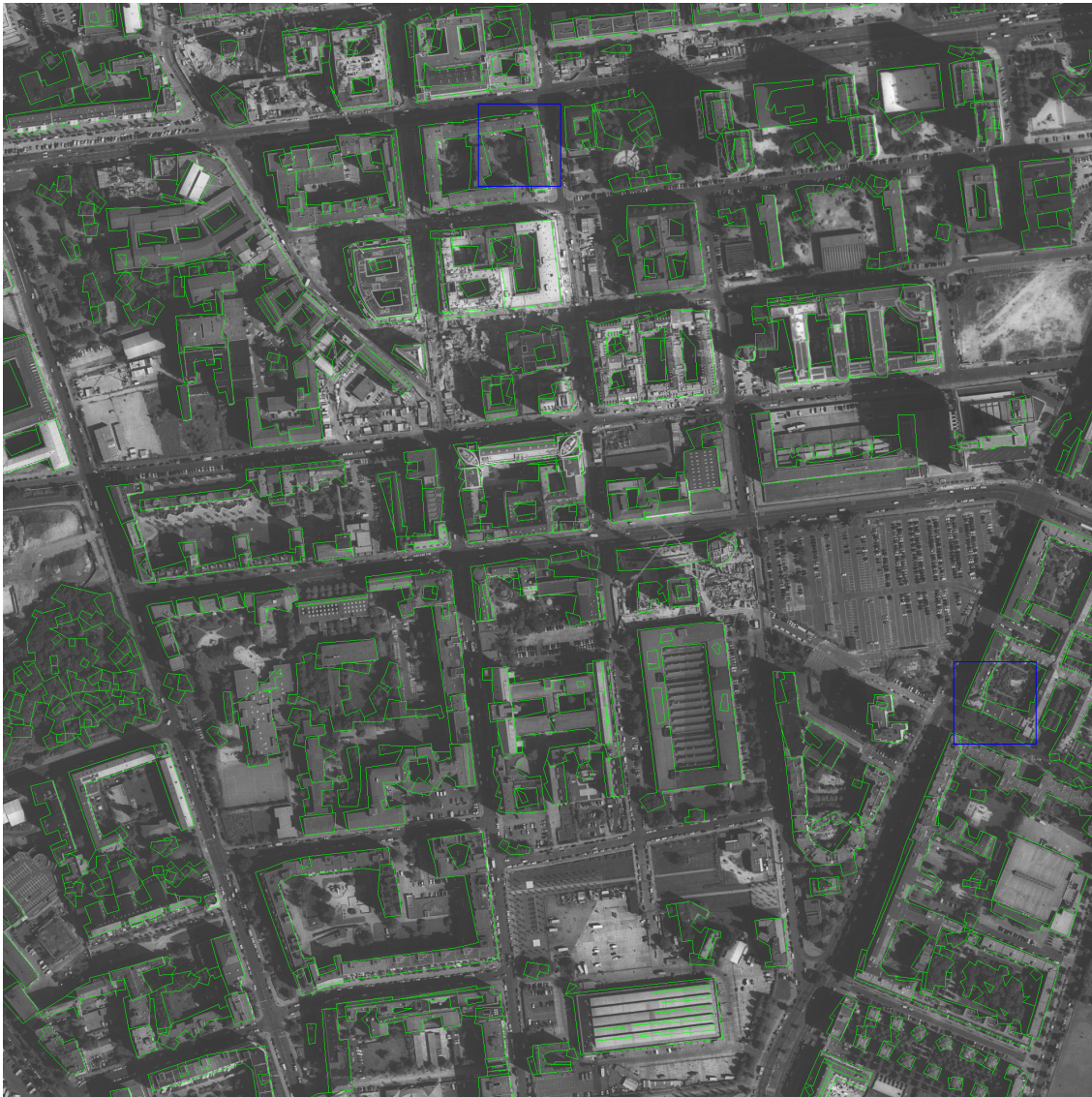


FIG. 7.6 – *Résultat de l'orthogonalisation des modèles de bâtiments : Les modèles des bâtiments sont réguliers, une grande partie des angles a été corrigée. Le modèle décrit beaucoup mieux la forme générale du bâtiment.*



FIG. 7.7 – Construction du modèle polygonal de chaque bâtiment : Présentation du modèle polygonal obtenu, superposé au MNE synthétique utilisé pour la vectorisation. Le modèle représente parfaitement le résultat de la segmentation.

Les résultats de l'algorithme d'orthogonalisation sont présentés dans la figure 7.6. Les modèles polygonaux des bâtiments de la scène sont maintenant beaucoup réguliers. Une grande partie des imperfections liées à l'aspect lisse du MNE initial utilisé a été supprimé : la plupart des angles des bâtiments a été corrigée et une grande partie de leurs bords a retrouvé une forme rectiligne. L'algorithme a permis de retrouver la forme générale des bâtiments de la scène.

7.2.3 Temps de calcul

Nous n'avons jusqu'à présent que très peu parlé des temps de calcul. C'est un paramètre essentiel si nous voulons développer un système de modélisation des bâtiments exploitable dans un contexte industriel. Les temps de calcul qui vont être présentés ont été obtenus sur la zone d'intérêt urbaine dense de 1km^2 de Berlin utilisée pour présenter les résultats, elle se compose de 134 objets du sur-sol. Nous avons utilisé une Sun Ultra 10 avec 128M de Ram.

L'algorithme de segmentation de MNE en plans horizontaux traite la totalité de la scène en moins de 25 minutes. L'algorithme de vectorisation construit tous les modèles polygonaux initiaux en moins de 4 minutes. Enfin, l'algorithme d'orthogonalisation met 20 minutes pour traiter tous les bâtiments de la scène.

7.2.4 Validation des modèles reconstruits

Nous voulons utiliser les modèles résultants du système de modélisation présenté pour alimenter des outils télécoms de planification radio. Ces outils permettent à la fois de déployer, de maintenir et de densifier les réseaux cellulaires. Ils simulent la couverture d'une cellule et permettent de limiter le nombre de campagnes de mesure à effectuer, d'où une économie importante. Pour valider les modèles de terrain que nous obtenons automatiquement, nous avons décidé de mettre en œuvre une procédure de validation basée sur des outils télécoms. Les enjeux de ces études sont importants pour démontrer l'intérêt des produits et anticiper la demande en faisant évoluer ceux-ci avec les besoins des clients d'ISTAR.

Dans une simulation, la valeur calculée du champ reçu dépend principalement du modèle de propagation choisi et du modèle de l'environnement. Dans la procédure de validation que nous proposons de mettre en œuvre, nous cherchons à comparer des mesures de champs et des valeurs simulées prédites par le modèle de propagation couplé avec différents MNE :

- le MNE brut issu de la chaîne de production d'ISTAR,
- le MNE synthétique construit à l'aide de modèles de bâtiments obtenus manuellement par des opérateurs,
- et le MNE synthétique construit à l'aide de modèles obtenus automatiquement avec le système de modélisation que nous proposons.

Ces études ont été effectuées par Benoît Mousseau dans le cadre de son stage de DESS. Elles se déroulent sous la direction conjointe de la société ISTAR et de l'IRCOM-SIC, laboratoire du CNRS associé à l'université de Poitiers. L'équipe radiocommunications de l'IRCOM étudie et développe différents modèles de propagation.

Le modèle de propagation Les mesures de champs que nous utilisons comme référence sont issues d'une campagne de mesure réalisée sur Paris intra-muros (figure 7.8). Nous disposons de 5000 points de mesure avec une précision de 5m en position (DGPS). Le calcul est effectué avec des données numériques prises le long d'un profil 2D (entre l'émetteur et

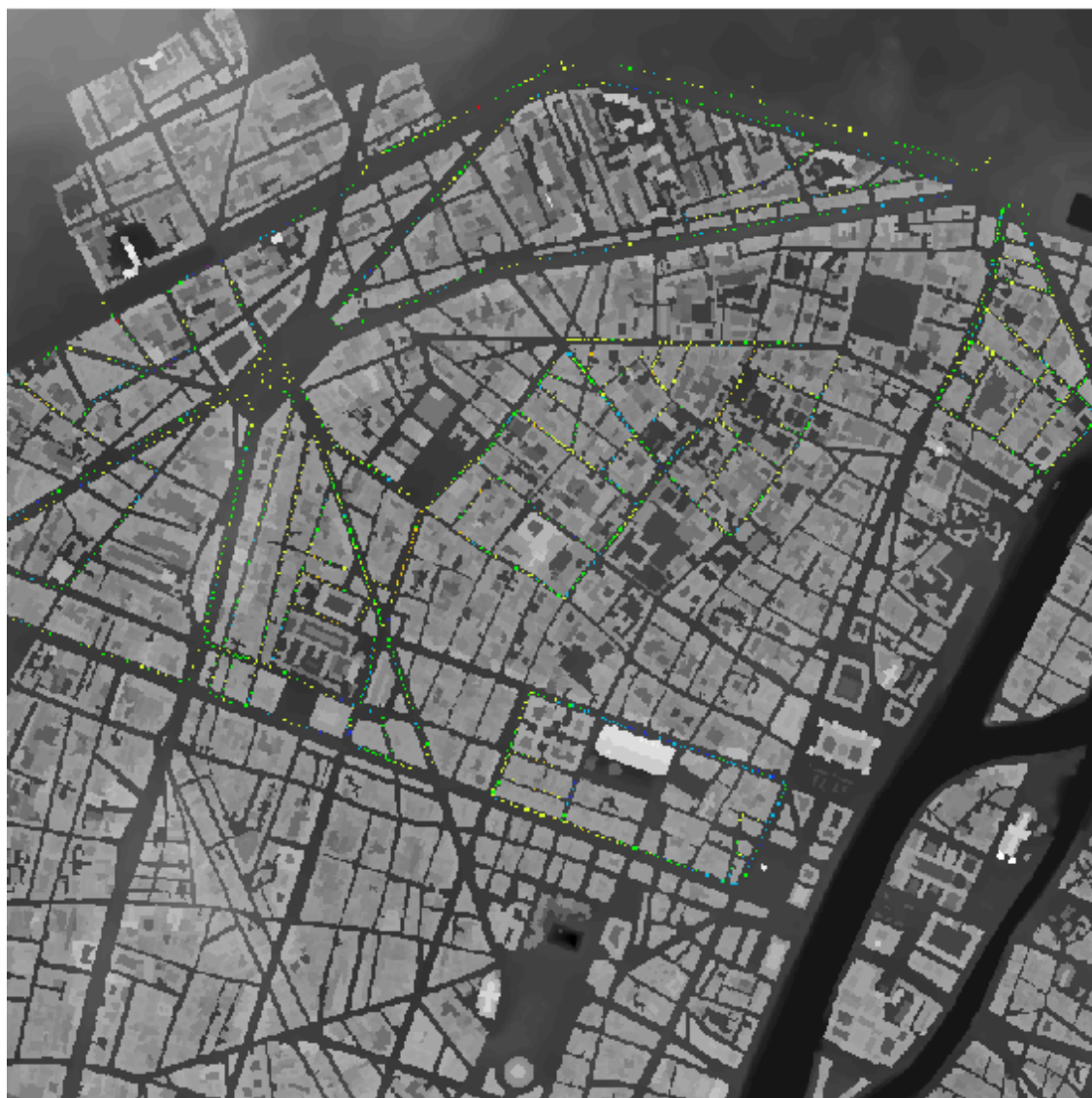


FIG. 7.8 – *Campagne de mesure sur Paris. Les points en couleur correspondent aux mesures effectuées.*

le récepteur). Le modèle de propagation utilisé pour les calculs théoriques du champ en chaque point de la scène prend en compte la diffraction de manière déterministe (modèle de type "knife-edge"), nous utilisons le modèle de Deygout. Les bâtiments sont modélisés par des demi-plans pour le calcul de l'affaiblissement de diffraction. Ce modèle utilise les 2 plans correspondants aux arrêtes contribuant le plus à l'affaiblissement de diffraction. Les autres phénomènes sont pris en compte par un modèle statistique (Okumura-Hata). Ce modèle empirique a été établi à partir d'une longue série de campagnes de mesure au Japon. La combinaison de ces modèles permet d'obtenir une estimée rapide du champ électrique avec une précision jugée satisfaisante pour déployer un réseau et le maintenir dans une zone rurale et sub-urbaine.

Tests comparatifs Le profil utilisé pour le calcul théorique du champ électrique en un point est extrait directement du MNE. Pour valider le système de modélisation proposé, nous allons comparer les valeurs du champ de référence (issues de la campagne de mesure) aux valeurs théoriques obtenues avec le MNE brut, le MNE synthétique reconstruit à partir de modèles de bâtiments obtenus manuellement et le MNE synthétique reconstruit à partir des modèles issues de notre système de modélisation automatique. Tous ces MNE ont une résolution de 1m. La comparaison est obtenue sur 2500 mesures.

La figure 7.9 présente les courbes représentant la distribution des erreurs des champs calculés pour les trois MNE utilisés. En observant les formes des distributions, nous pouvons voir dans un premier temps que les distributions correspondant aux deux MNE synthétiques ont les mêmes caractéristiques. Nous utilisons aussi l'écart type de ces distributions d'erreurs pour comparer la qualité des champs théoriques calculés (figure 7.10). L'écart type des valeurs simulées du champ avec le MNE synthétique automatique est compris entre ceux correspondant à l'utilisation du MNE brut et du MNE synthétique manuel.

En analysant de ces résultats, nous pouvons dire que la vectorisation automatique apporte une amélioration notable avec le modèle de propagation utilisé lors des simulations. Les résultats sont presque comparables à ceux obtenus avec le MNE synthétique manuel. La différence entre l'utilisation des deux MNE synthétiques réside au niveau des erreurs de plus de 20 dB qui sont plus nombreuses avec le MNE synthétique automatique. D'un point de vue pratique, ces erreurs correspondent aux points de mesure situés trop près des bâtiments et des arbres. Lorsque les arbres sont agglomérés aux bâtiments ou que les limites de ces derniers sont trop avancées dans la rue, la valeur du champ n'est plus évaluée au niveau du sol mais à une altitude plus élevée. Ce problème n'est pas dû à une erreur du MNE mais au fait que nous ne prenons pas en compte la possibilité pour le récepteur de se retrouver sous un arbre. Après correction manuelle des rues, l'écart type de l'erreur commise avec le MNE synthétique automatique passe à 8,24 dB et devient comparable au résultat obtenu avec le MNE synthétique manuel.

Pour conclure, nous pouvons dire que la vectorisation automatique donne des résultats de simulation comparables à ceux obtenus avec une vectorisation manuelle avec le modèle de propagation utilisé. Ceci est dû à l'utilisation d'un modèle de propagation basé sur la technique des "lames de couteau". Nous poursuivons actuellement nos études avec un modèle de propagation plus complet basé sur la GTD, modèle de type lancer de rayons. Ces modèles sont beaucoup plus complexes et mieux adaptés aux scènes urbaines et aux données haute résolution.

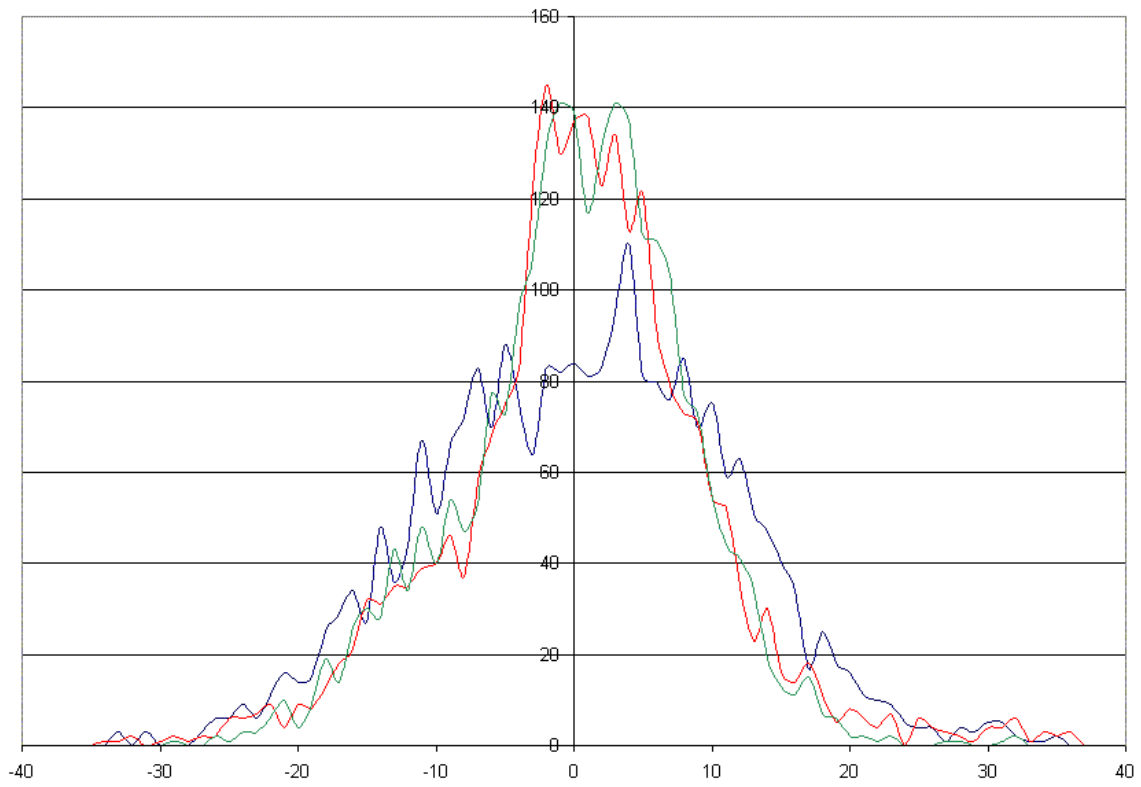


FIG. 7.9 – Courbes représentant la distribution des erreurs des champs calculés pour les trois MNE utilisés. L'abscisse correspond à la valeur des erreurs en dB des mesures simulées du champ et l'ordonnée au nombre de points ayant cette valeur. La courbe bleue correspond au MNE brut, la courbe verte à celle du MNE synthétique manuel et la courbe rouge à celle du MNE synthétique automatique. La forme des courbes correspondant aux deux MNE synthétiques sont semblables.

MNE brut	10,84 dB
MNE synthétique automatique	9,46 dB
MNE synthétique manuel	8,06 dB

FIG. 7.10 – Le tableau présente les écart types des distributions des erreurs des mesures simulées. L'écart type des valeurs simulées du champ avec le MNE synthétique automatique est compris entre ceux correspondant à l'utilisation du MNE brut et du MNE synthétique manuel. La vectorisation automatique apporte donc une amélioration notable avec le modèle de propagation utilisé.

7.3 Autres études et résultats

Nous présentons dans cette section les résultats obtenus avec les autres outils et algorithmes développés au cours de cette thèse. Nous présentons tout d'abord les résultats de correction de MNE à partir de la mise en correspondance des façades. Nous verrons ensuite les résultats du système de modélisation que nous avons défini, appliqué à un MNE corrigé. Nous verrons enfin plusieurs études sur le recalage des modèles de bâtiments.

7.3.1 Reconstruction de façades

Le chapitre 4 présentait un algorithme de reconstruction des façades. Nous proposons cet algorithme pour utiliser les informations correspondant aux façades visibles dans les images. Le principe est de rectifier localement les images initiales par rapport aux façades de la scène pour pouvoir appairer ces surfaces verticales. Cet algorithme traite automatiquement toutes les orientations de façades sans utiliser de connaissance a priori sur la forme des bâtiments.

Nous proposons ensuite d'utiliser ces points 3D particuliers pour corriger la forme des bâtiments dans le MNE. Nous creusons simplement dans le MNE, aux lieux où nous avons reconstruit ces points. Cet algorithme s'applique localement, les bâtiments sont traités un à un. Pour cela, nous utilisons le MNE dans lequel le sol a été extrait.

Reconstruction des points sur les façades

La figure 7.11 présente les résultats de reconstruction des façades sur la zone d'intérêt. L'algorithme traite toutes les orientations de façades possibles pour tous les bâtiments de la scène. Tous les points reconstruits sont ensuite projetés dans une même carte d'accumulation.

En comparant cette carte à l'ortho-image de la zone d'intérêt présentée en figure 7.2, nous pouvons voir qu'une grande partie des façades de la scène a été retrouvée. La reconstruction est de manière générale assez dense. Certaines façades n'ont toutefois pas pu être reconstruites pour plusieurs raisons. Tout d'abord, certaines de ces surfaces ne sont pas visibles dans les images. Cela peut être dû à la présence de végétation ou à l'étroitesse des rues ou des cours intérieures. La seconde raison est le manque de dynamique des images des façades lorsque celles-ci sont à l'ombre. Nous pouvons remarquer aussi en regardant la carte obtenue que très peu de points ont été reconstruits dans les régions correspondant à la végétation. L'algorithme ne reconstruit donc que les façades des bâtiments de la scène.

Correction du MNE

La figure 7.12 présente le résultat de l'algorithme de correction de MNE. Nous l'avons appliqué au MNE initial, après reconstruction des points sur les façades des bâtiments. Les régions en rouge sont les régions du MNE qui ont été corrigées par l'algorithme. Nous pouvons voir qu'une grande partie des limites des bâtiments a été corrigée. Comme pour la carte d'accumulation, nous pouvons remarquer que la végétation n'a pas été corrigée, l'algorithme ne traite que les bâtiments de la scène.

Malheureusement, certains faux appariements des images de façades n'ont pu être éliminés lors de l'étape de filtrage. Ces accumulations ont causé des trous dans les bâtiments (bâtiment au centre). L'étape de filtrage reste le point critique de l'algorithme. Les trous présents dans le bâtiment au centre sont causés par une accumulation de faux appariements au milieu du toit de ce bâtiment. Le filtrage 3D permet de rejeter tous les faux appariements



FIG. 7.11 – *Reconstruction des points sur les façades des bâtiments. L'image présentée correspond à la carte d'accumulation obtenue après avoir traité toutes les orientations de façade des bâtiments de la scène. Nous pouvons voir qu'une grande partie des façades a été corrigée et que peu de points ont été reconstruits dans les régions correspondant à la végétation.*

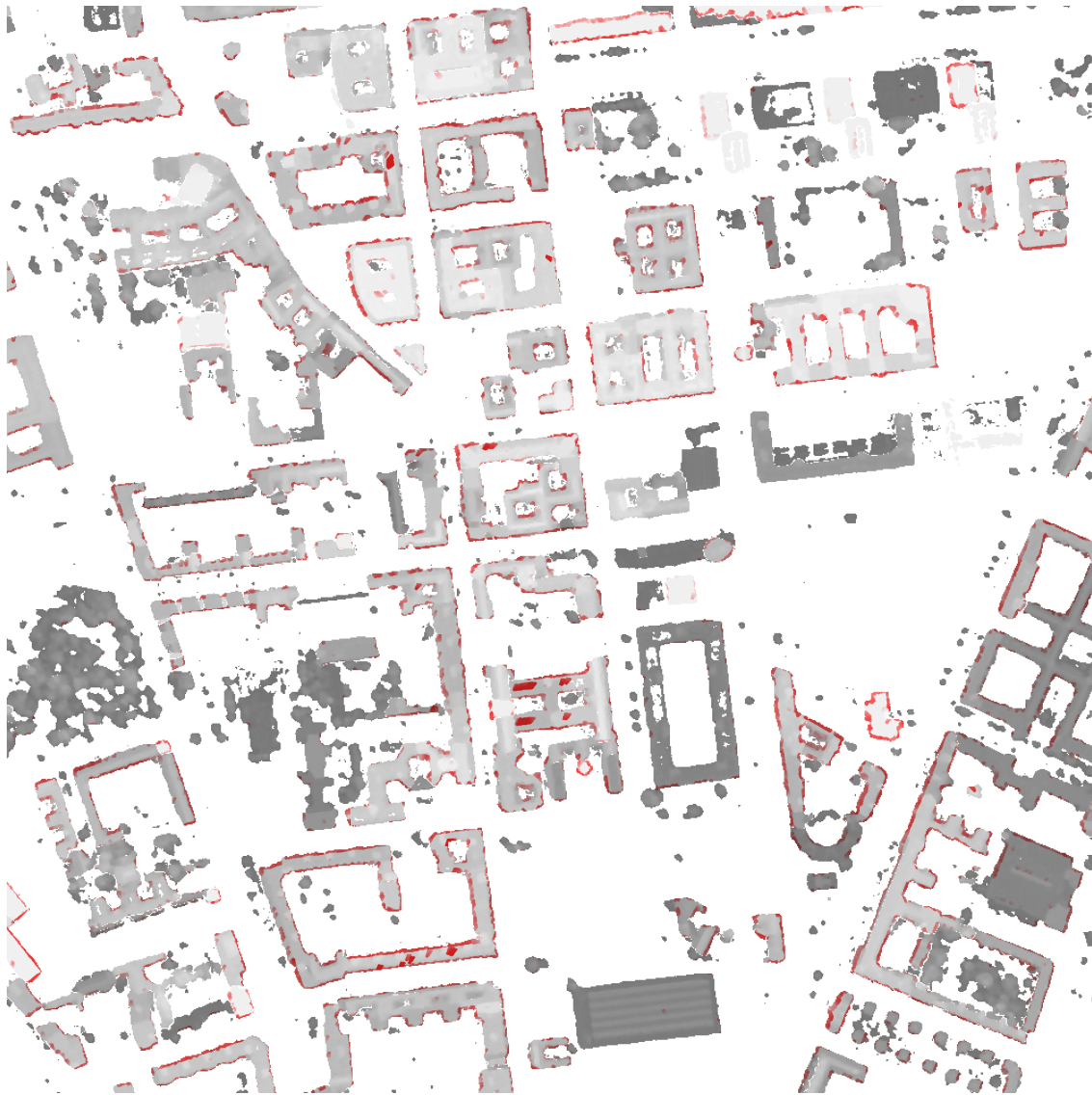


FIG. 7.12 – Correction d'un MNE à partir des points reconstruits sur les façades des bâtiments. Les régions en rouge ont été corrigées par l'algorithme. Une grande partie des façades a été corrigée. On peut remarquer toutefois que l'étape de filtrage n'a pas réussi à éliminer des accumulations de faux appariements dans certains bâtiments. Ceci a causé des trous au milieu des toits.

des façades lorsque ces derniers ne convergent pas dans une même région de l'espace tridimensionnel. Nous étudions le moyen d'éliminer ces erreurs en obligeant l'étape de correction à ne traiter que les bords des bâtiments.

Initialisation à l'aide des vecteurs automatiques

L'algorithme de corrélation automatique des façades n'utilise aucun a priori sur la forme des bâtiments, et surtout sur l'orientation de ses façades. Nous proposons maintenant de modifier l'algorithme pour utiliser un modèle initial du bâtiment. Le modèle est prismatique (section 3.3), c'est un ensemble de polygones dans lequel chaque chaîne correspond à un plan horizontal d'altitude connue. Le modèle polygonal initial que nous utilisons est celui obtenu en utilisant les algorithmes de segmentation et de vectorisation automatique des bâtiments que nous avons développés au cours de cette thèse.

Le principe est d'utiliser les segments de ce modèle pour guider la mise en correspondance des façades. Dans ce cas, il n'est plus nécessaire de lancer l'algorithme pour toutes les orientations de façades du bâtiment. Nous utilisons l'orientation des segments pour construire les images des façades. Nous réduisons la zone d'intérêt au voisinage du segment (nous augmentons sa longueur de 30% et lui associons une épaisseur de $6m$). Nous appliquons l'algorithme localement sur chacune des façades ainsi définie et corrigeons le MNE sur le même principe et avec les mêmes paramètres que l'algorithme automatique.

Les résultats de reconstruction des façades sont présentés dans la figure 7.13. A présente l'ortho-image correspondant au bâtiment traité. B présente le modèle polygonal réel obtenu manuellement (en vert) superposé au MNE synthétique construit à partir des plans issus de la segmentation automatique. C présente le modèle polygonal initial (en bleu) obtenu automatiquement par nos algorithmes de segmentation et d'approximation polygonale. D présente le MNE corrigé obtenu en utilisant l'algorithme automatique de reconstruction des façades. Nous pouvons voir qu'il a permis de retrouver les limites réelles (en vert) du bâtiment. E et F présentent le MNE corrigé obtenu en utilisant l'algorithme modifié qui utilise un modèle polygonal initial. Nous utilisons exactement les mêmes paramètres que dans l'algorithme automatique présenté dans la section 4.5.

G et H présentent les cartes d'accumulation obtenues avec l'algorithme automatique et l'algorithme modifié. Nous retrouvons la forme des façades. Le nombre de points reconstruits est moins important en utilisant l'algorithme présenté dans cette section. Nous pouvons voir aussi que le premier algorithme a permis de retrouver des points sur des façades qui n'ont pas été segmentées, c'est parce qu'il traite toutes les orientations de façades pour tous les bâtiments contenus dans la zone d'intérêt. Le nouvel algorithme ne traite qu'un voisinage du segment traité.

L'algorithme modifié permet aussi de retrouver les limites réelles du bâtiment. Par contre, il a reconstruit un nombre beaucoup moins important de points sur les façades. Ceci est dû au fait que l'on utilise plus qu'une seule paire d'image pour chaque façade. De plus, l'orientation utilisée pour construire les images des façades ne correspond plus exactement à celle de la façade réelle. Nous pouvons voir par contre qu'il n'y a plus de trous dans les bâtiments provoqués par quelques données aberrantes qui n'ont pu être supprimées, comme dans le bâtiment du haut. Les faux appariements proviennent principalement du traitement d'orientations qui ne correspondent à aucune façade réelle de la scène.

Puisque le nouvel algorithme est beaucoup moins sensible à ces erreurs, nous pouvons utiliser une étape de filtrage beaucoup moins sévère. Le temps de calcul devient beaucoup plus important car il est directement lié à la complexité du modèle initial. Pour cet exemple

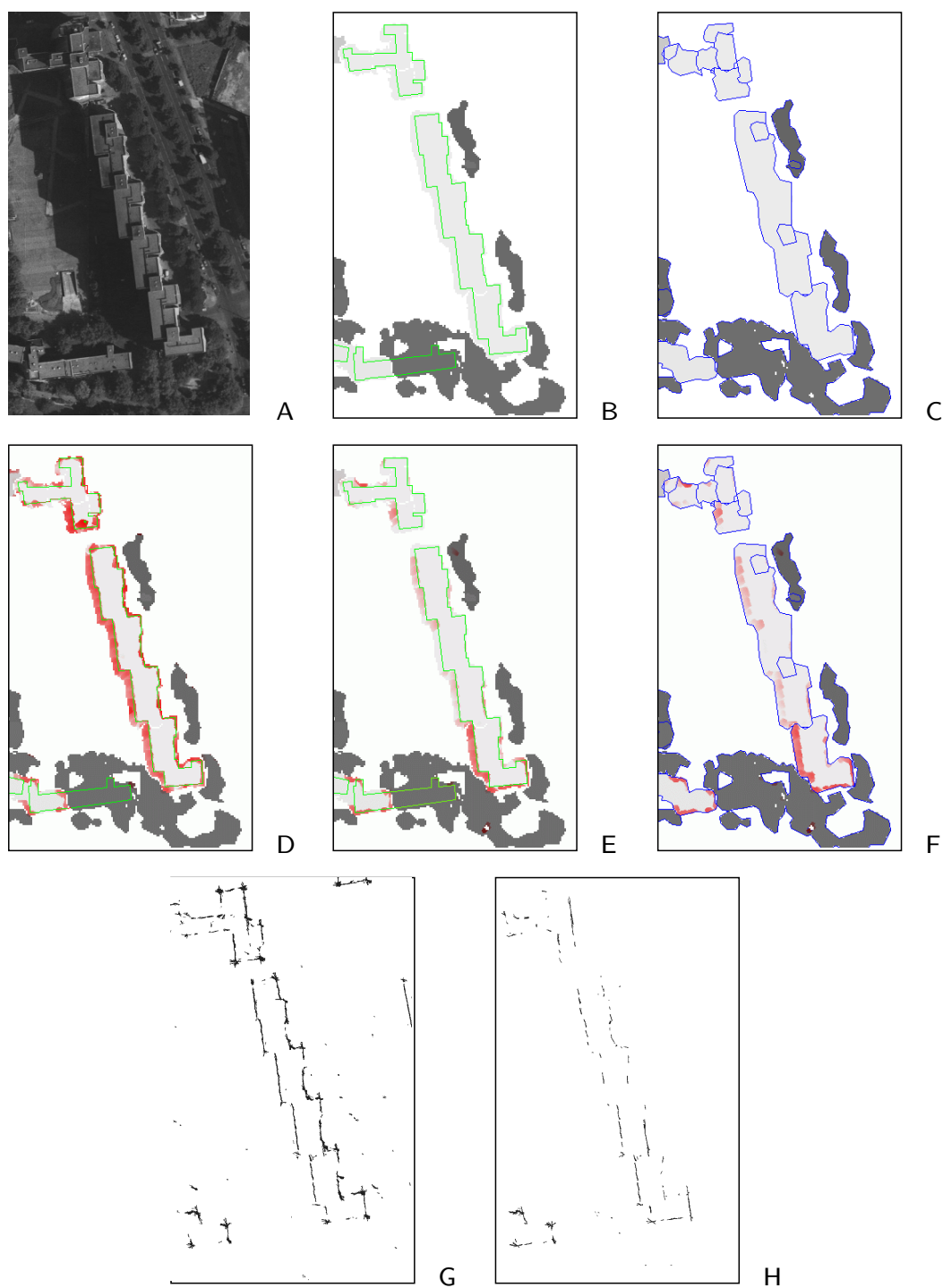


FIG. 7.13 – *Algorithme modifié de correction de MNE. Nous utilisons un modèle polygonal initial pour guider la mise en correspondance des façades. A présente l'ortho-image du bâtiment traité. B et C présentent respectivement le modèle polygonal réel (vert) et le modèle polygonal initial (bleu) superposés au MNE synthétique. D présente la correction du MNE obtenue avec l'algorithme de reconstruction de façades automatique. E et F présentent la correction obtenue avec l'algorithme modifié ainsi que les modèles réels et initiaux. G et H présentent les cartes d'accumulation obtenues avec l'algorithme automatique et l'algorithme modifié. Ce nouvel algorithme n'est plus sensible aux faux appariements mais reconstruit beaucoup moins de points sur les façades.*

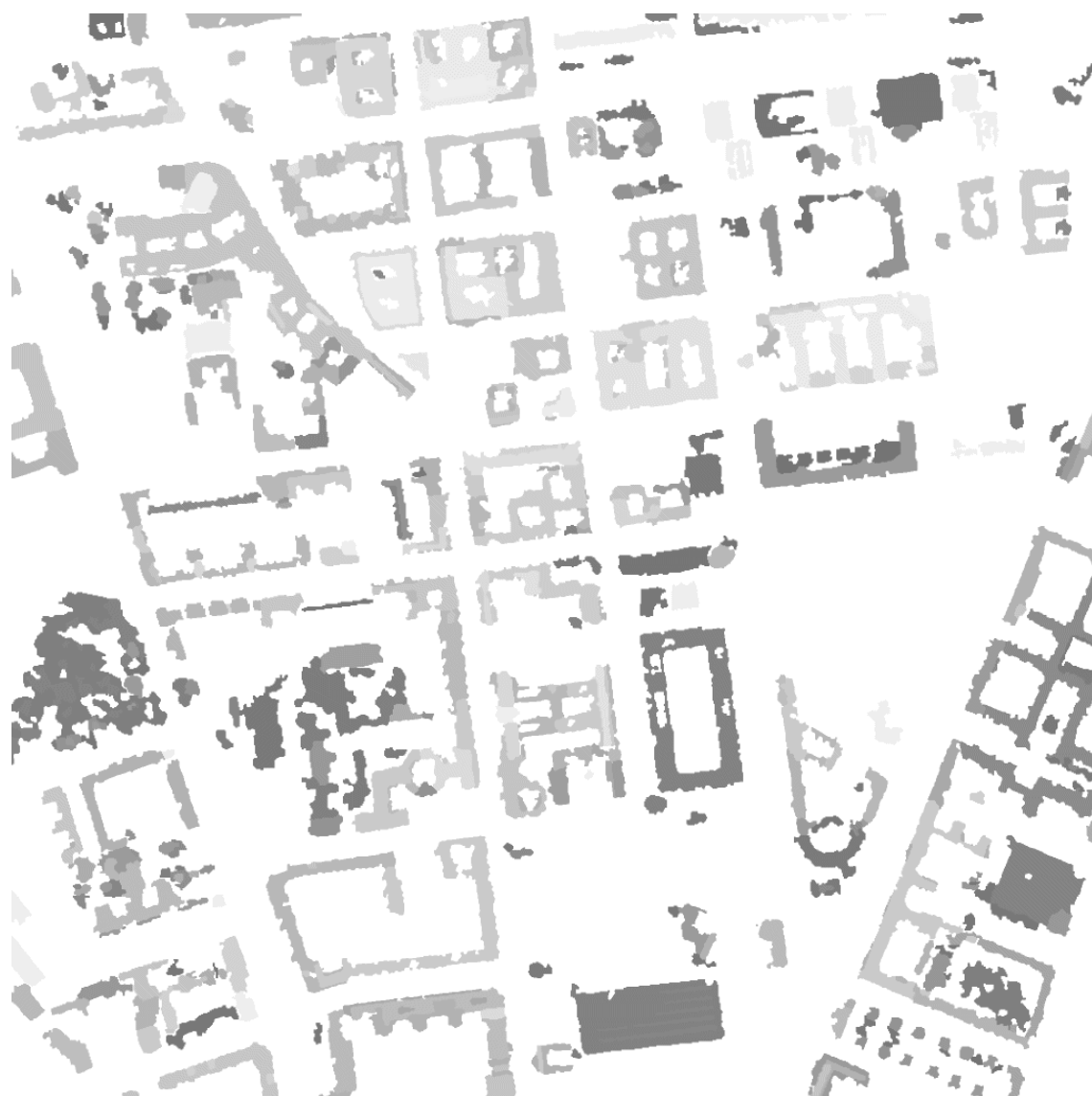


FIG. 7.14 – *Résultat de la segmentation en plans horizontaux après correction des façades : Nous proposons de segmenter le MNE obtenu après la correction des façades. La plupart des façades ont été corrigées, les toits sont moins larges. Des problèmes apparaissent sur certains bords des bâtiments, lorsque les façades ne sont que partiellement corrigées.*

de bâtiment, le temps de calcul est d'environ 30 minutes pour l'algorithme automatique et de plus de 3 heures pour l'algorithme modifié sur un Sun Ultra 10. Une amélioration possible de l'algorithme serait de ne traiter que les orientations principales extraites du modèle initial.

7.3.2 Système de modélisation appliqué sur un MNE corrigé

Segmentation d'un MNE corrigé

Nous proposons maintenant d'appliquer l'algorithme de segmentation sur le MNE obtenu après la correction des façades. Les résultats sont présentés dans la figure 7.14. Les bâtiments dont les limites ont été corrigées sont moins larges. L'algorithme de correction basé sur la reconstruction des façades permet de retrouver les limites réelles d'une grande partie des bâtiments. Toutefois, des problèmes apparaissent sur certains bords des bâtiments, lorsque les façades ne sont que partiellement corrigées. Les limites des bâtiments ne sont plus rectilignes. Il sera plus facile de comparer les deux résultats de segmentation après l'étape de vectorisation (figures 7.5 et 7.15).

Modèle polygonal du bâtiment

Construction du modèle Nous avons appliqué l'algorithme d'approximation polygonale sur le MNE corrigé de la zone étudiée. Les résultats sont présentés dans la figure 7.15. L'algorithme de correction basé sur la reconstruction des façades permet de retrouver les limites réelles des bâtiments. Les modèles polygonaux reconstruits seront plus proches du bâtiment que l'on souhaite modéliser comme le montre l'exemple (B, C). Toutefois, des problèmes apparaissent aux bords des bâtiments lorsque les façades ne sont que partiellement corrigées. Certains limites des bâtiments ne sont plus rectilignes et le modèle peut être considérablement altéré comme le présente (D, E).

Modèle avec contraintes angulaires La figure 7.16 présente les résultats obtenus en appliquant l'algorithme d'orthogonalisation sur des modèles de bâtiments construits à partir d'un MNE corrigé. A présente le modèle polygonal obtenu. Nous avons vu précédemment que l'algorithme de correction automatique des façades permet de retrouver les véritables bords des bâtiments mais que des problèmes apparaissent lorsque les façades ne sont que partiellement corrigées. L'algorithme d'orthogonalisation corrige une partie des déformations engendrées. Dans l'exemple (B, C), nous voyons que l'utilisation d'un MNE corrigé a permis d'obtenir un modèle représentatif de la structure de ce bâtiment. Au contraire, dans l'exemple (D, E) le bâtiment n'a été corrigé que partiellement et le modèle polygonal obtenu n'est plus représentatif.

L'utilisation d'un MNE corrigé peut permettre de construire un modèle polygonal plus représentatif lorsque les façades des bâtiments sont bien visibles dans les images initiales et que le modèle est bien corrigé. Nous pensons que l'amélioration sera beaucoup plus nette en utilisant des images de plus haute résolution.

7.3.3 Orthogonalisation des modèles recalés par contours actifs

Nous avons vu dans le chapitre 6 deux méthodes de correction du modèle polygonal par son recalage sur les contours des images : les contours actifs et la programmation dynamique.

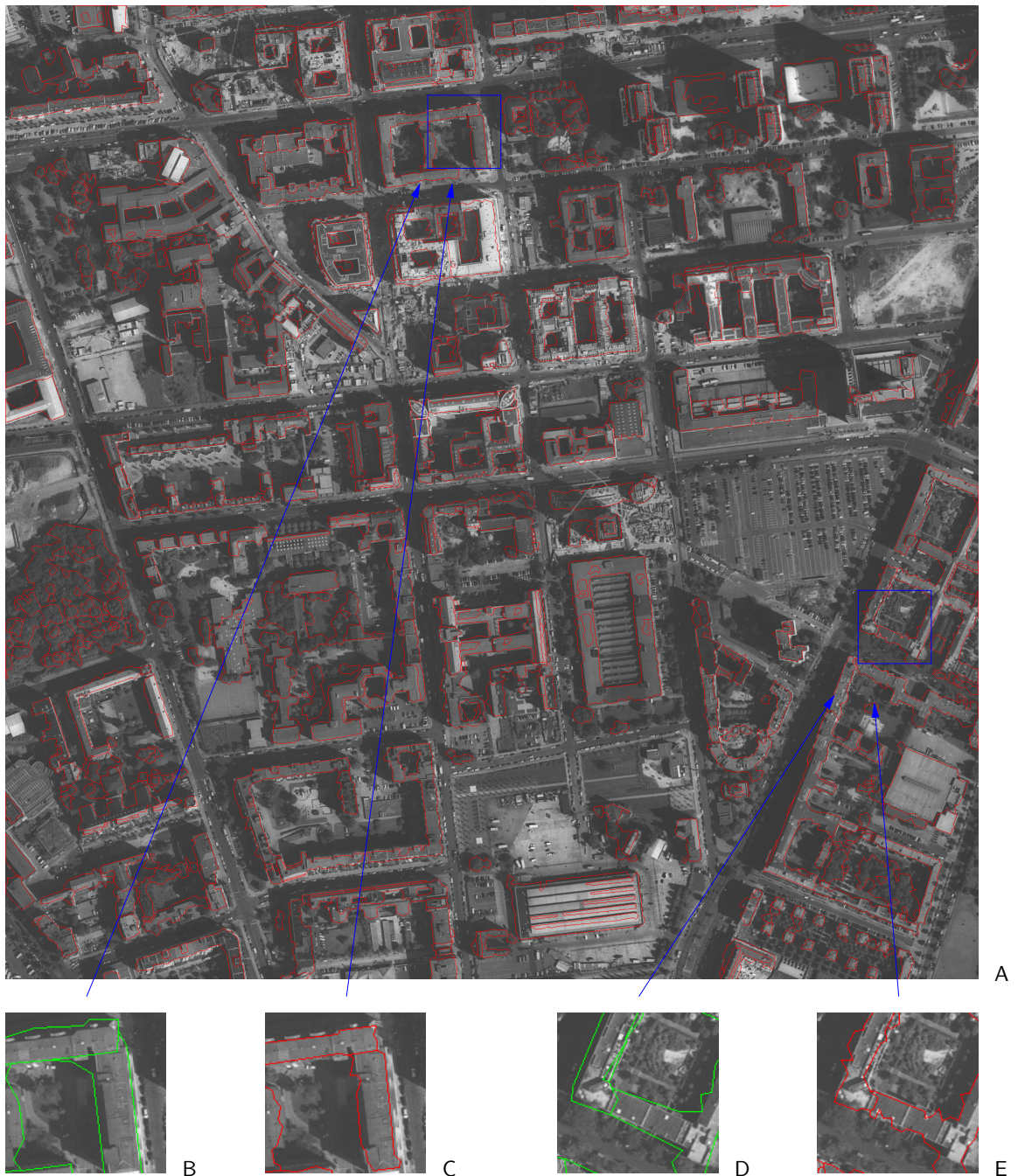


FIG. 7.15 – Construction du modèle polygonal après la correction des bâtiments avec les façades : Nous avons aussi appliqué l'algorithme d'approximation polygonale sur un MNE corrigé à partir des points reconstruits sur les façades des bâtiments, image A. B et D présentent en vert une partie du modèle polygonal obtenu avec le MNE initial (figure 7.5). C et E présentent en rouge le modèle polygonal obtenu avec le MNE corrigé. On voit dans l'exemple (B, C) que l'algorithme de correction a permis de repositionner certains contours sur les limites réelles des bâtiments. Dans l'exemple (D, E), nous pouvons voir que certains contours sont maintenant déformés, lorsque les façades n'ont été que partiellement corrigées.

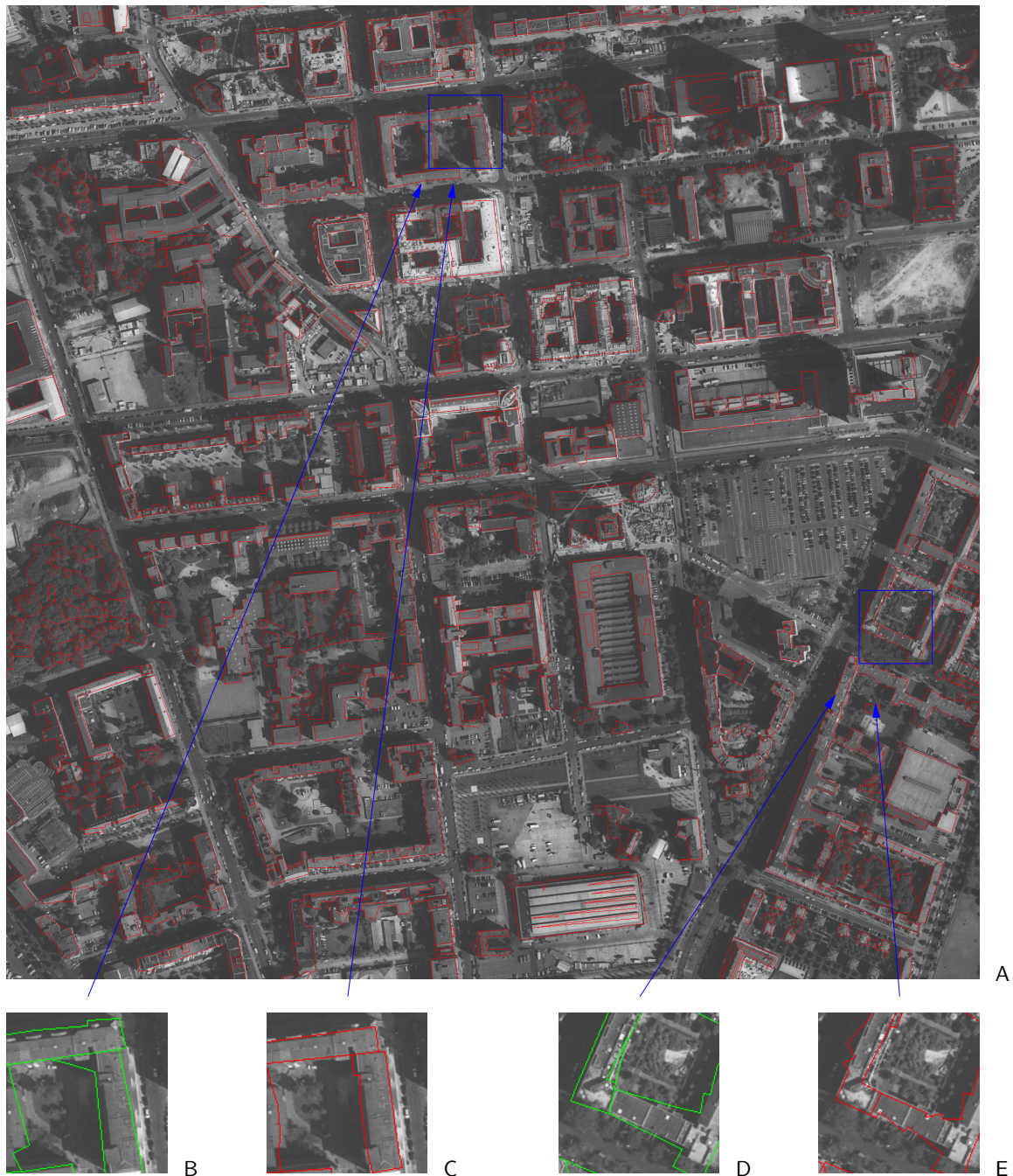


FIG. 7.16 – Résultat de l'orthogonalisation des modèles de bâtiments après la correction des façades : A présente le modèle polygonal obtenu après l'orthogonalisation. B et D présentent en vert une partie du modèle polygonal obtenu en utilisant le MNE initial (figure 7.6). C et E présentent en rouge le modèle polygonal obtenu en utilisant le MNE corrigé (A). Nous voyons dans l'exemple (B, C) que l'utilisation d'un MNE corrigé a permis d'obtenir un modèle représentatif de la structure de ce bâtiment. Au contraire, dans l'exemple (D, E) le bâtiment n'a été corrigé que partiellement et le modèle polygonal obtenu n'est plus représentatif.

Nous n'étudions dans la suite que le recalage par des contours actifs (des snakes polygonaux multi-images) qui nous donnent de meilleurs résultats. Les snakes que nous utilisons sont contraints à se déplacer horizontalement : nous utilisons un snake par plan du modèle polygonal du bâtiment.

Les résultats de recalage ont été présentés en section 6.2.2 dans la figure 6.11. Nous avons vu que les modèles étaient bien recalés sur les contours réels des bâtiments. Toutefois, des imperfections résiduelles restent présentes dans le modèle, notamment parce que le modèle ne prend pas en compte le changement de topologie. Nous proposons dans cette section de corriger certaines des imperfections restantes en appliquant au modèle recalé l'algorithme d'orthogonalisation que nous venons de présenter.

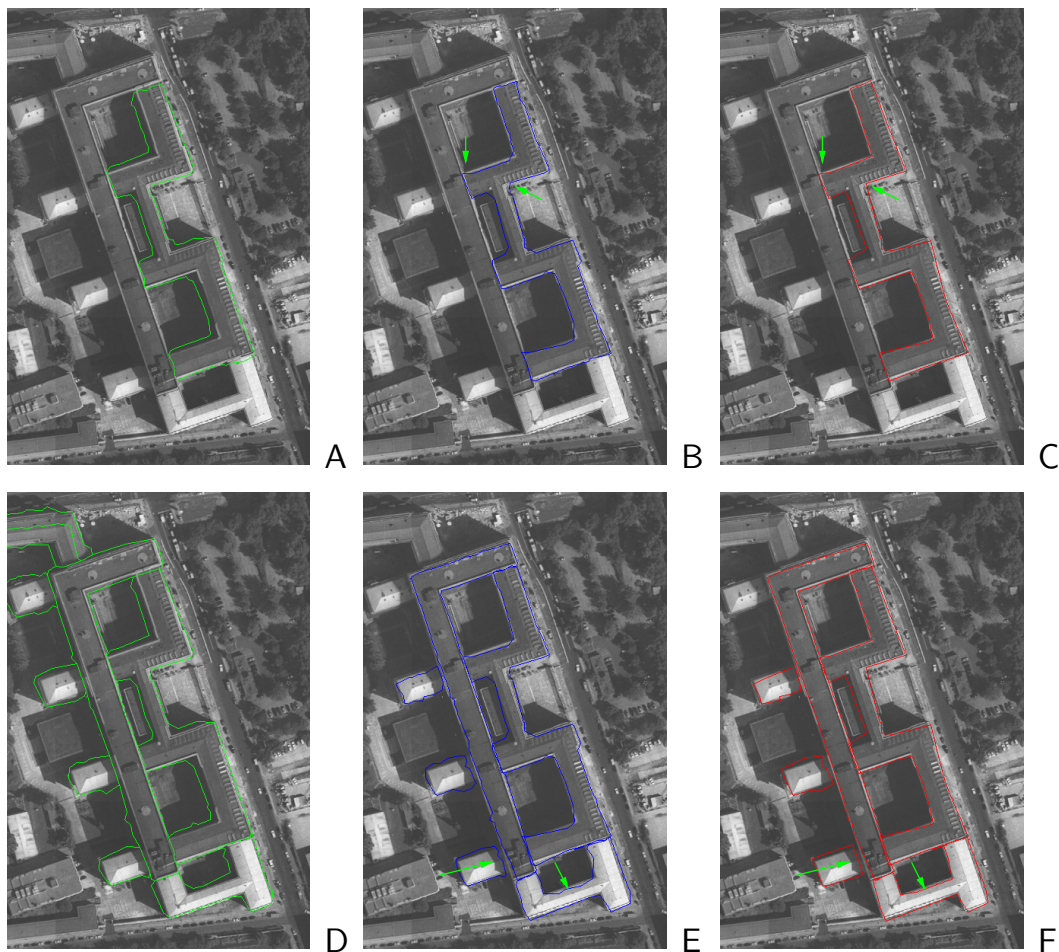


FIG. 7.17 – Résultats de l'orthogonalisation d'un modèle recalé sur les images par contours actifs. (A, B, C) et (D, E, F) présentent l'étude de deux plans du bâtiment. (G, H, I) présente les résultats obtenus pour tous les plans qui composent le bâtiment. La première image, modèle en vert, présente le contour initial. La seconde image, modèle en bleu, présente le modèle recalé par les snakes polygonaux avec douze images du bâtiment. La dernière image, modèle en rouge, présente le modèle obtenu après orthogonalisation du modèle bleu. Les modèles obtenus représentent mieux les bâtiments. Les flèches vertes présentent les améliorations notables du modèle polygonal.

Les résultats sont présentés dans la figure 7.17. (A, B, C) et (D, E, F) présentent l'étude des deux plans du bâtiments que nous avons traité dans le chapitre 6. (G, H, I) présente les résultats obtenus pour tous les plans qui composent le bâtiment¹. La première image de chaque ligne, le modèle en vert, présente le contour initial obtenu après segmentation. La seconde image, le modèle en bleu, présente le modèle recalé par les snakes polygonaux multi-images. L'algorithme utilise les douze images disponibles du bâtiment. La dernière image de chaque ligne, le modèle en rouge, présente le modèle obtenu après orthogonalisation du modèle bleu. Les bords sont plus réguliers. Les modèles obtenus représentent mieux les bâtiments. Les flèches vertes présentent les améliorations notables du modèle polygonal.

7.3.4 Snakes guidés par les façades

Nous avons vu dans le chapitre 4 que les points reconstruits sur les façades par l'algorithme automatique possédaient une bonne localisation planimétrique. Nous proposons maintenant d'utiliser ces points pour guider les snakes polygonaux que nous avons développés. L'idée est d'utiliser ces points pour guider le snake lorsque le bord du bâtiment n'est pas perçu dans les images initiales (contour franc) et ne peut être retrouvé par l'algorithme standard de contours actifs.

Pour cela, nous allons utiliser la carte d'accumulation, obtenue en projetant dans cette carte les points reconstruits pour toutes les orientations traitées. Nous allons utiliser les images des dérivées de cette carte pour guider le snake sur le bord du bâtiment, comme le fait une image de ce même bâtiment. Ceci est équivalent à ajouter une énergie supplémentaire dépendant de la hauteur du pic d'accumulation de cette carte.

Les résultats sont présentés dans la figure 7.18. A présente en vert le contour initial du plan. B présente le MNE corrigé du bâtiment et C la carte d'accumulation correspondante. Nous pouvons remarquer que se sont les façades orientées vers le sud et l'est qui ont été principalement corrigées. D présente le contour initial superposé à la carte d'accumulation. E et F présentent le contour obtenu en utilisant l'algorithme des snakes en utilisant uniquement la carte d'accumulation. Ce modèle est superposé à la carte d'accumulation et l'ortho-image du bâtiment. Nous pouvons voir que le modèle s'est déplacé vers les pics d'accumulations de la carte (E) lorsque celle-ci a été retrouvée. Nous voyons dans F que le modèle est bien recalé sur le bord du bâtiment, dans les lieux où se sont accumulés les points reconstruits. Le contour ne se déplace pas pour certaines des façades ouest et nord car aucun point n'a été reconstruit sur ces façades. G présente en bleu le modèle polygonal obtenu en appliquant l'algorithme des snakes avec toutes les images du bâtiment. H et I présentent le contour obtenu avec l'algorithme des snakes, en utilisant cette fois la carte d'accumulation et toutes les images disponibles. La carte est considérée comme une image supplémentaire qui possède une confiance supérieure (équivalente à deux images). Les résultats sont comparables.

La figure 7.19 compare les résultats de recalage par contours actifs avec et sans la carte d'accumulation. Nous comparons les modèles obtenus sur le bâtiment avant et après l'étape d'orthogonalisation. Les images A et D présentent les modèles polygonaux initiaux (en vert). La première ligne (B,C) présente les modèles obtenus avec les snakes standards (uniquement les images). La seconde ligne (E,F) présente les modèles reconstruits en utilisant en plus la carte d'accumulation. Les modèles en bleu correspondent aux résultats de l'algorithme des snakes. Les modèles en rouge présentent les modèles obtenus après l'étape d'orthogonalisation. L'apport de la carte d'accumulation a permis de retrouver certaines limites réelles

1. Nous faisons remarquer que notre algorithme ne traite actuellement que les chaînes fermées. Il est toutefois possible d'appliquer ces algorithmes à des chaînes ouvertes.

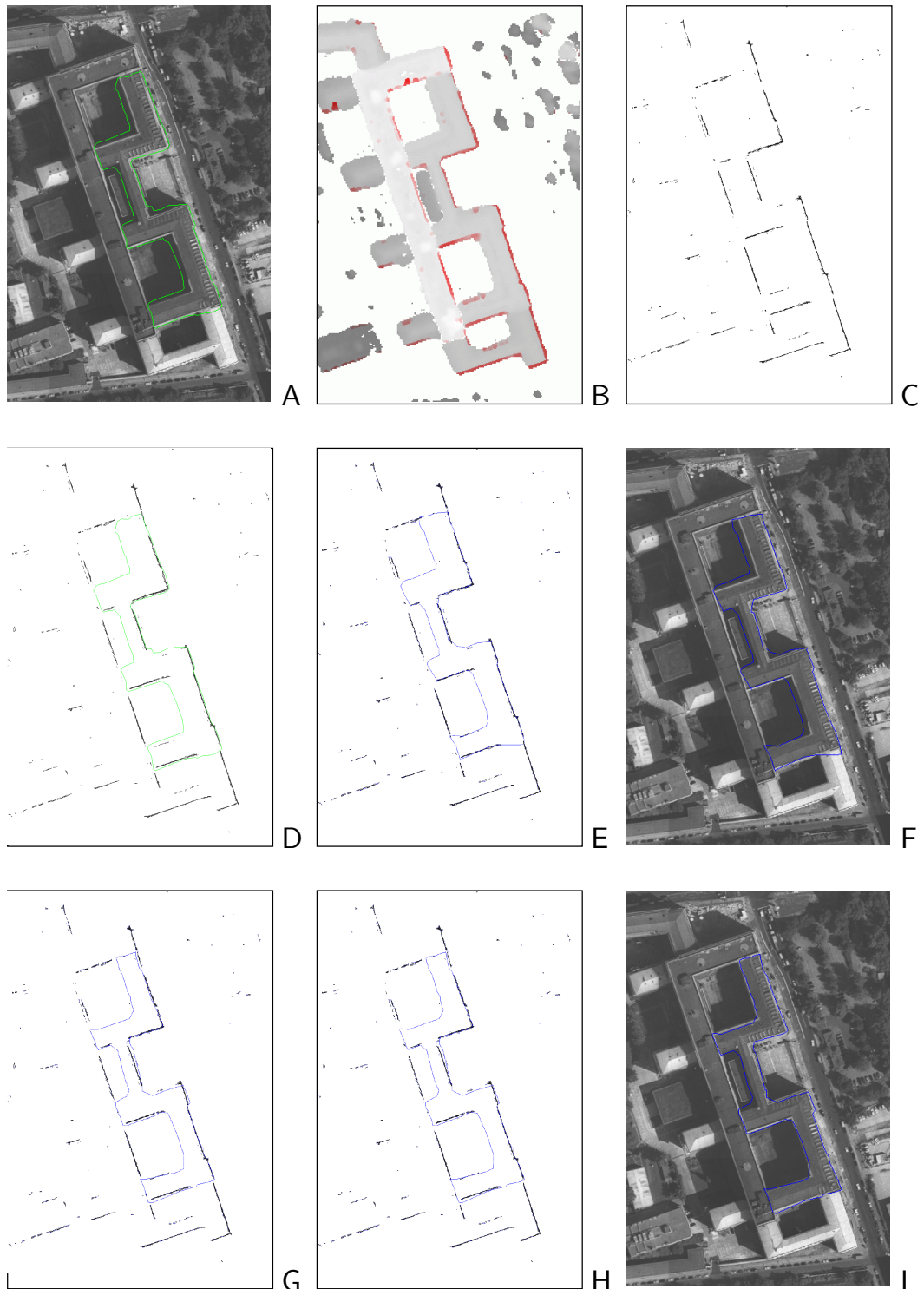


FIG. 7.18 – *Résultat de recalage par contours actifs en utilisant la carte d'accumulation. A présente en vert le contour initial du plan. B présente le MNE corrigé du bâtiment et C la carte d'accumulation correspondante. D présente le contour initial superposé à la carte d'accumulation. E et F présentent le contour obtenu avec l'algorithme des snakes, en utilisant uniquement la carte d'accumulation. G présente en bleu le modèle polygonal obtenu en utilisant l'algorithme des snakes que nous avons développé. H et I présentent le contour obtenu avec l'algorithme des snakes, en utilisant cette fois la carte d'accumulation et toutes les images disponibles. La carte d'accumulation apporte une information supplémentaire pour guider le contour actif.*

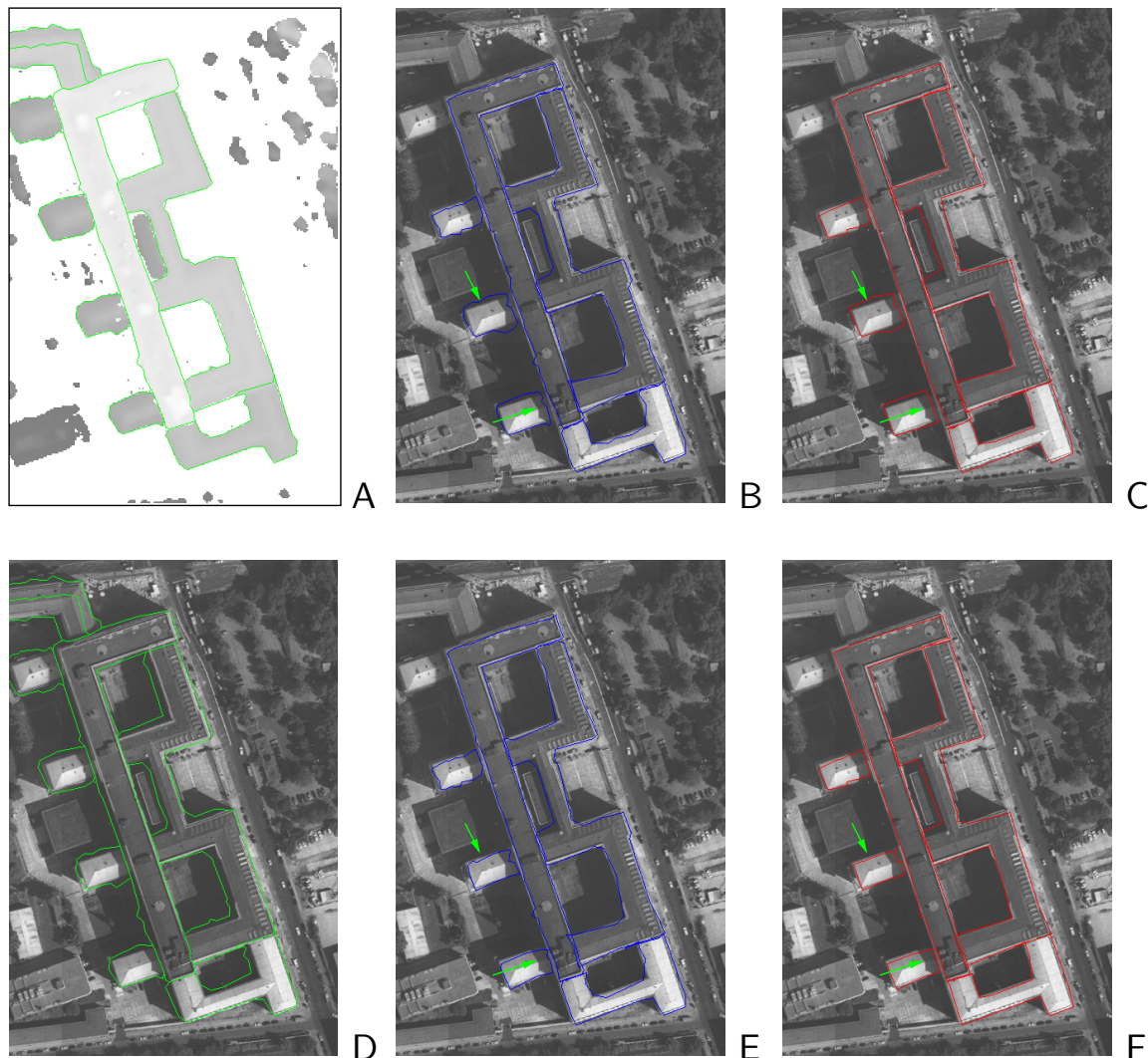


FIG. 7.19 – Comparaison des modèles de bâtiment obtenus par des contours actifs avec et sans utilisation de la carte d'accumulation : A et D présentent en vert les modèles polygonaux initiaux. La première ligne (B,C) présente les modèles obtenus avec les snakes standards. La seconde ligne (E,F) présente les modèles reconstruits en utilisant en plus la carte d'accumulation. Les modèles en bleu correspondent aux résultats de l'algorithme des snakes. Les modèles en rouge présentent les modèles obtenus après une étape d'orthogonalisation. L'apport de la carte d'accumulation a permis de retrouver certaines limites du bâtiment (flèches vertes).

du bâtiment (flèches vertes). Elle permettra aussi d'apporter une information lorsqu'il n'y a pas de contour dans les images ou lorsqu'un autre contour qui ne correspond à aucune discontinuité vient perturber l'optimisation.

7.3.5 Temps de calcul

L'algorithme de reconstruction des façades est le traitement le plus long. Il nous aura fallu un peu plus de 1 jour (1580 minutes) pour obtenir les résultats. L'algorithme traite 18 orientations de façade pour chaque bâtiment. C'est l'algorithme de mise en correspondance par corrélation des images des façades qui prend le plus de temps. Ce temps pourrait être diminué en supprimant les objets du sur-sol qui ne correspondent pas à des bâtiments. Nous estimons le temps moyen de traitement d'un bâtiment aux environs de 40 à 50 minutes. L'algorithme modifié qui utilise un modèle initial traite un bâtiment de faible taille comme celui présenté dans la figure 7.13 en 190 minutes.

Les autres algorithmes que nous utilisons sont beaucoup plus rapides. L'algorithme de segmentation de MNE en plans horizontaux traite la totalité de la scène en moins de 25 minutes. L'algorithme de vectorisation construit tous les modèles polygonaux initiaux en moins de 4 minutes. Enfin, l'algorithme d'orthogonalisation met 20 minutes pour traiter tous les bâtiments de la scène. L'algorithme de recalage par contours actifs traite un plan du bâtiment en moins de 10 secondes, le chargement des différentes images représentant environ 90% du temps de l'algorithme.

7.4 Conclusion

Nous avons présenté dans ce chapitre les résultats obtenus avec les algorithmes développés au cours de cette thèse. Ces résultats sont présentés sur une zone d'intérêt urbaine dense de $1km^2$ sur Berlin.

Nous avons étudié les différents outils présentés dans cette thèse pour développer un système de modélisation des bâtiments. Nous avons fixé quatre objectifs pour qualifier une bonne modélisation : détecter tous les bâtiments, traiter toutes les formes de bâtiment, obtenir un modèle cohérent de la forme générale du bâtiment et avoir un détourage du bâtiment de bonne qualité. Tous les algorithmes que nous avons présentés ont été développés pour répondre au mieux à ces objectifs.

Système de modélisation étudié Nous avons proposé le système de modélisation de scène urbaine suivant : (1) segmentation du MNE initial en plans horizontaux pour retrouver les régions planaires qui permettent de décrire les toits des bâtiments, (2) approximation polygonale des contours des plans pour construire un premier modèle polygonal de chaque bâtiment et (3) orthogonalisation des modèles des bâtiments pour obtenir la description finale de la scène. Les résultats de ce système sont présentés en figure 7.6. Nous avons ensuite défini une procédure de validation basée sur des outils télécoms.

Comme nous l'avons vu dans ce chapitre, un tel système permet de détecter tous les bâtiments de la scène pour obtenir une modélisation cohérente et uniforme des bâtiments de forme quelconque. Trois des quatre objectifs fixés au début de la thèse sont atteints. Ce système permet de traiter une zone urbaine de $1km^2$ en moins d'une heure.

Le dernier objectif visé concerne la qualité de la modélisation. Les premières simulations de champs sont encourageantes puisque le système de modélisation permet d'obtenir

des résultats comparables à ceux obtenus avec des modèles construits manuellement. Ces données peuvent permettre d'obtenir une estimée rapide du champ électrique avec une précision jugée satisfaisante pour déployer un réseau et le maintenir dans une zone rurale et sub-urbaine. Toutefois, des déformations morphologiques subsistent. Ces déformations sont principalement dues au traitement direct du MNE qui est une reconstruction fortement déformée de la scène. Nous poursuivons les études pour déterminer si cette modélisation est suffisante pour être utilisée par des outils télécoms plus complets basés sur la GTD.

Nous travaillons actuellement sur l'amélioration de la qualité en intégrant une étape de recalage basée sur les contours actifs. Nous souhaitons utiliser à la fois les images initiales et la carte d'accumulation des points reconstruits sur les façades pour obtenir un recalage correct du modèle sur les bords du bâtiment. Nous devons développer pour cela une stratégie qui permette de recaler un modèle polygonal complet sur ces images. Cette étape supplémentaire interviendra avant celle de l'orthogonalisation pour permettre de corriger les dernières déformations présentes dans le modèle du bâtiment. En observant les résultats obtenus avec une telle technique, ce système de modélisation devrait permettre d'obtenir une description beaucoup plus proche des données obtenues manuellement. Elles pourront alors devenir exploitables par des applications industrielles.

Études réalisées Nous avons étudié l'algorithme de reconstruction des façades présenté dans le chapitre 4. Nous avons vu qu'une grande partie des façades de la scène a été retrouvée. La carte d'accumulation présentant les points reconstruits est dense et le MNE a été corrigé correctement. Nous avons aussi proposé de modifier cet algorithme en utilisant un modèle initial du bâtiment. La reconstruction est moins perturbée mais la complexité combinatoire du nouvel algorithme le rend difficilement exploitable.

Nous avons ensuite appliqué l'algorithme de segmentation en plans sur le MNE corrigé. La représentation obtenue reste cohérente mais des problèmes apparaissent aux bords des bâtiments lorsque les façades ne sont que partiellement corrigées. Nous poursuivons ces études pour améliorer la reconstruction des points 3D. Nous souhaitons aussi les utiliser pour guider la segmentation et la reconstruction des bâtiments.

Les algorithmes de l'étape de vectorisation sont utilisés pour construire le modèle polygonal de chaque bâtiment et corriger les imperfections présentes. Nous avons tout d'abord présenté les résultats de l'algorithme d'approximation polygonale développé. La forme générale de chaque bâtiment est respectée par l'algorithme. La majorité des segments du modèle correspond aux bords des bâtiments. Certains angles restent déformés à cause de l'aspect lisse du MNE initial.

Nous utilisons deux algorithmes différents pour corriger le modèle: un algorithme de recalage du modèle sur les contours des images en utilisant des snakes polygonaux et un algorithme d'orthogonalisation qui impose des contraintes angulaires. Nous avons montré des résultats pour chacun de ces algorithmes utilisés individuellement et en les combinant. Chacun d'eux permet de supprimer une partie des imperfections du modèle polygonal initial. La combinaison des deux algorithmes permet d'obtenir une bonne description finale pour chaque bâtiment de la scène. Le nombre important d'images permet d'assurer la convergence des snakes. La carte d'accumulation aide l'algorithme lorsque les contours n'engendrent pas une force assez importante. L'algorithme d'orthogonalisation permet quant à lui d'assurer l'obtention d'un modèle cohérent et de forme régulière du bâtiment traité.

Chapitre 8

Conclusion

Nous avons présenté dans cette thèse plusieurs outils pour la reconstruction automatique de bâtiments à partir d'imagerie aérienne. Nous nous sommes placés dans le contexte d'une application de télécommunications. Nous avons choisi une stratégie de reconstruction en trois étapes : construction d'un Modèle Numérique d'Élévation (MNE) dense et fiable, segmentation des bâtiments en régions planes puis vectorisation des contours des plans pour obtenir la description finale du bâtiment.

Les objectifs visés étaient : de détecter tous les bâtiments, de traiter toutes les formes de bâtiment, d'obtenir un modèle représentatif de la forme générale du bâtiment et d'avoir un détourage du bâtiment de bonne qualité. Tous les outils proposés dans cette thèse ont été développés pour répondre au mieux à ces objectifs. Ils sont présentés en détail dans les chapitres 4, 5 et 6. Dans le chapitre présentant les résultats (7), nous les avons mis en œuvre sur une scène urbaine assez dense de $1km^2$ correspondant à la ville de Berlin. Nous avons présenté un système de modélisation qui nous permet de construire une description de la scène urbaine qui répond aux objectifs visés.

8.1 Contributions

Les principales contributions de cette thèse sont les suivantes :

Outils pour la reconstruction automatique de bâtiments La contribution principale de cette thèse est la présentation, l'étude et le développement de plusieurs outils pour la reconstruction automatique de bâtiments. Nous avons fait un état important des publications sur le sujet avant de définir notre stratégie pour reconstruire les bâtiments. Nous avons ensuite présenté et intégré plusieurs algorithmes dans chacune des étapes de notre stratégie de reconstruction.

La plupart des algorithmes que nous avons étudiés sont utilisés habituellement dans des domaines scientifiques différents. Nous avons intégré et adapté ces algorithmes pour l'application que nous voulons traiter. Le point fort de thèse est d'avoir réussi à assembler tous ces algorithmes pour obtenir des résultats cohérents et utilisables dans un contexte industriel. Nous avons aussi voulu que cette thèse soit un bon point de départ pour toute personne s'intéressant à la reconstruction automatique de bâtiments.

Système complet de modélisation Nous avons présenté dans la section 7.2 un système de modélisation des bâtiments basé sur les outils que nous avons développés. Ce système de modélisation de scène urbaine est le suivant : (1) segmentation du MNE initial en plans horizontaux pour retrouver les régions planes qui permettent de décrire les toits des bâtiments, (2) approximation polygonale des contours des plans pour construire un premier modèle de chaque bâtiment et (3) orthogonalisation des modèles des bâtiments pour obtenir la description finale de la scène.

Ce système permet de détecter tous les bâtiments de la scène pour obtenir une modélisation cohérente et uniforme des bâtiments de forme quelconque. Ce système permet de traiter une zone urbaine de $1km^2$ en moins d'une heure. Nous avons proposé une procédure de validation des résultats basée sur des outils télécoms. Les résultats montrent que ces données sont satisfaisantes pour déployer un réseau et le maintenir dans une zone rurale et sub-urbaine. Des déformations morphologiques subsistent toutefois dans certains modèles de bâtiments, rendant l'utilisation de ces modèles moins certaine avec des outils télécoms plus complexes. Ces études continuent.

Reconstruction automatique des façades Une contribution originale et importante est décrite dans le chapitre 4, l'algorithme de reconstruction et de correction des façades. Nous proposons cet algorithme pour utiliser les informations correspondant aux façades visibles dans les images. Le principe consiste à rectifier localement les images initiales par rapport aux façades de la scène pour pouvoir apparier ces surfaces verticales. Nous avons vu dans la section 4.4.3 que la localisation des points reconstruits sur les façades était meilleure en utilisant les images des façades qu'en utilisant les images initiales. Nous avons montré aussi dans la section 4.4.4 que la sensibilité par rapport à l'orientation fixée pour la transformation des façades était peu importante.

Nous avons ensuite présenté un algorithme permettant la reconstruction automatique des façades sans aucun a priori sur la forme du bâtiment traité ni sur l'orientation de ses façades. Nous avons proposé aussi une méthode qui utilise les points reconstruits pour corriger les limites des bâtiments dans le MNE. Les résultats ont été présentés sur une scène synthétique et une scène urbaine assez complexe de $1km^2$. Nous obtenons une reconstruction dense de points sur la plupart des façades des bâtiments. Nous avons montré comment utiliser les points reconstruits sur les façades pour corriger les limites des bâtiments ou pour guider des contours actifs.

Segmentation en plans Nous avons ensuite développé un algorithme de segmentation de MNE en plans. Nous avons proposé de décomposer le bâtiment en plans horizontaux ou d'orientation quelconque. Cet algorithme permet de retrouver les régions planes correspondant aux différentes parties des toits des bâtiments. Il comprend trois phases. Une phase d'exploration basée sur la technique robuste du RANSAC (RANdom SAmple Consensus) permet de construire un jeu d'hypothèses de modèles de plans. Une phase de fusion supprime ensuite la redondance des modèles. Enfin, une phase de sélection basée sur le MDL (Minimum Description Length) et la recherche par tabou permet de ne conserver que les meilleures hypothèses pour décrire le bâtiment.

Du fait de la faible résolution du MNE ($50cm$) et de son aspect lisse, nous réalisons une segmentation en plans horizontaux pour obtenir une modélisation représentative. Cet algorithme peut être appliqué au MNE initial ou au MNE corrigé par l'algorithme de reconstruction automatique des façades. Les résultats sur la scène urbaine ont montré que

l'algorithme permettait d'extraire les principales régions planes qui composent les toits des bâtiments. Cet algorithme nous permet d'obtenir une représentation correcte et robuste des bâtiments de la scène.

Vectorisation Le chapitre vectorisation (6) représente un point clé de notre stratégie, nous avons développé et testé quatre algorithmes pour l'obtention d'un modèle de bâtiment représentatif. Nous avons tout d'abord proposé un algorithme d'approximation polygonale pour construire le modèle de chaque bâtiment. Cet algorithme utilise actuellement les plans horizontaux proposés par l'algorithme de segmentation.

Nous avons ensuite proposé trois techniques différentes pour corriger le modèle polygonal initial. Les deux premières sont des techniques qui permettent de recalculer les plans du modèle polygonal sur les contours des images. Nous avons développé une technique basée sur des snakes polygonaux multi-images et une technique qui utilise l'algorithme de programmation dynamique. Nous avons comparé les résultats obtenus pour un bâtiment de ces deux techniques, puis avons choisi l'algorithme des snakes qui est plus stable et donne de meilleurs résultats avec les douze images disponibles du bâtiment.

Nous avons aussi proposé une troisième technique de correction qui impose des contraintes d'orthogonalité au modèle polygonal. Cette technique est basée sur l'optimisation d'une fonction d'objectif couplée à un estimateur robuste, le M-estimateur. Cet estimateur permet d'assurer la stabilité des résultats. Nous avons appliqué cette technique sur le modèle polygonal de la scène urbaine. Il a permis de corriger la plupart des angles des bâtiments et une grande partie de leurs bords a retrouvé une forme rectiligne. Le modèle corrigé respecte beaucoup mieux la forme générale des bâtiments de la scène.

Nous avons présenté et comparé les résultats de ces trois méthodes de correction. Les meilleurs résultats sont obtenus avec l'algorithme de recalage par contours actifs et l'algorithme d'orthogonalisation. Chacun d'eux permet de corriger une partie des déformations présentes dans les modèles. Nous avons montré que ces deux algorithmes sont complémentaires et que leur association permet d'obtenir un résultat de très bonne qualité.

Outils développés Cette thèse aura aussi permis de développer une série de programmes et une librairie dédiés à la modélisation de bâtiments. La majorité des programmes a été développée en C++ conformément aux règles de codage de l'environnement SPOT3D d'ISTAR (segmentation en plans, recalage par snakes, recalage par programmation dynamique, ...). Les algorithmes d'approximation polygonale et d'orthogonalisation ont été développés par *M. Mesnage* au cours d'un stage d'ingénieur dirigé par *F. Devernay* et co-encadré par moi-même. L'algorithme de mise en correspondance des façades utilise plusieurs programmes C++, développés à ISTAR (corrélations, projections, ...) ou par moi-même (construction des images de façades, choix des images, différents filtres, cartes d'accumulations, correction de MNE, ...), le tout englobé dans des scripts PERL. Une documentation sous format HTML est en cours de rédaction pour l'exploitation de ces outils.

États de l'art Nous avons réalisé dans cette thèse un état de l'art pour chacun des domaines étudiés. Nous avons tout d'abord présenté dans le chapitre 2 l'étude d'un système de vision permettant la reconstruction d'objets en présentant des références pour chacune des étapes. Nous avons ensuite présenté dans le même chapitre un état de l'art des méthodes d'appariement d'images. Dans le chapitre 3, nous avons repris les étapes du système de vision dans le contexte du traitement d'un chantier aérien. Nous avons présenté ensuite un état de

l'art des méthodes de reconstruction de bâtiments. Dans le chapitre 5, nous avons proposé deux revues, une première sur les principales méthodes d'estimation robuste et une seconde sur les algorithmes de reconstruction de surfaces à partir de données tridimensionnelles. Nous avons présenté aussi des références sur la sélection de modèles. Enfin, dans le chapitre 6, nous avons présenté les principaux algorithmes à base modèles déformables.

8.2 Perspectives

Validation des résultats du système de modélisation Tout d'abord, nous continuons les tests de comparaison et de validation des données numériques de terrain avec d'autres modèles de propagation. Il sera intéressant d'étudier les MNE synthétiques obtenus automatiquement et manuellement avec des modèles de propagation basés sur le lancer de rayon. Les enjeux de ces études sont importants pour certifier les produits fournis par ISTAR auprès de ses clients.

Travaux et améliorations envisagés Les premières améliorations envisagées concernent la qualité de la modélisation. Pour cela, nous souhaitons intégrer une étape de recalage par contours actifs, utilisant les images initiales et la carte d'accumulation des points reconstruits sur les façades, dans le système de modélisation. Au vu des résultats obtenus avec le recalage par contours actifs, ce nouveau système de modélisation devrait permettre d'obtenir une description exploitable par des applications industrielles. Nous devons développer pour cela une nouvelle stratégie qui permette de recalibrer un modèle polygonal tridimensionnel de bâtiment complet. Nous pensons pour cela développer un algorithme tel que celui proposé par *P. Fua* dans [Fua96a] ou *T. Moons et al.* dans [MFVG98].

Nous pensons aussi tester le système proposé avec un MNE initial de plus haute résolution. Dans ce cas, nous pourrions utiliser pleinement les capacités de l'algorithme de segmentation avec des plans d'orientation quelconque. Nous devons utiliser un nombre de tirages aléatoires de modèles plus important. Les phases de fusion et de sélection seront alors d'autant plus importantes dans le processus de segmentation. Nous devons par conséquent adapter l'algorithme d'orthogonalisation en ajoutant les variables d'angle supplémentaires au niveau des arrêtes verticales du nouveau modèle à corriger.

Dans cette thèse, nous avons choisi d'utiliser une stratégie basée sur les surfaces. Nous utilisons le MNE de la scène pour construire le modèle de bâtiment. Nous souhaitons dans le futur développer et étudier une stratégie basée sur les modèles. Nous pensons notamment construire le modèle d'un bâtiment à partir des segments 3D extraits des images. Il serait aussi possible d'utiliser dans la procédure des segments construits à partir de la carte d'accumulation des points reconstruits sur les façades.

Suite de la thèse Après cette thèse, un projet de collaboration entre le laboratoire de recherche *Computer Vision* du groupe IRIS de USC (University of South California) et ISTAR débutera en octobre 2000. Ce projet se déroulera sous la forme d'une coopération du service national.

L'approche du laboratoire de recherche *Computer Vision* de USC est très différente de celle que nous avons développée dans cette thèse puisqu'elle est basée sur les modèles. L'idée générale consiste à extraire des primitives des images (comme les segments), puis à les regrouper par une méthode hiérarchique de groupement perceptuel pour former des

structures cohérentes et construire le modèle de bâtiment. Leurs approches ont été développées au cours de projets militaires américains de l'ARPA (RADIUS, APGD et MURI).

Le projet de recherche se compose de trois axes de recherche majeurs. Le premier axe sera de tester et d'évaluer les algorithmes développés par le laboratoire avec des données fournies par ISTAR. Le but est d'évaluer les différentes approches par modèle de USC pour la reconstruction de bâtiments. Nous pourrions comparer les résultats obtenus par les approches "modèle" à ceux que nous avons pu obtenir avec une approche "surface" au cours de cette thèse.

Le deuxième axe de recherche sera l'étude et le développement d'algorithmes de reconstruction de bâtiments en utilisant les expériences des deux laboratoires. Nous poursuivrons tout d'abord les études des sujets présentés dans le paragraphe précédent : *Travaux et améliorations envisagés*. Nous pensons tester aussi la technique de segmentation de surfaces tridimensionnelle basée sur le *tensor voting*. Nous pourrions utiliser les points 3D reconstruits par le processus de corrélation standard qui utilise les images initiales, mais aussi les points 3D reconstruits sur les façades. Ces points permettront de mieux délimiter les bâtiments pour obtenir une reconstruction plus cohérente.

Le dernier axe de recherche portera sur l'étude des bâtiments avec le nouveau capteur HRSC (High-Resolution Stereo Camera) d'ISTAR. Ce capteur permet d'obtenir des images de très haute résolution avec une plus grande dynamique de l'intensité, surtout dans les zones d'ombres. Cette caractéristique devrait permettre d'obtenir des MNE de meilleure qualité, les contours des bâtiments seront plus réguliers. Cette étude permettra d'expérimenter les algorithmes dont nous disposons pour évaluer le potentiel de ce nouveau capteur pour l'extraction automatique des bâtiments.

Bibliographie

- [AF87] N. AYACHE ET B. FAVERJON, Efficient registration of stereo images by matching graph descriptions of edge segments, *The International Journal of Computer Vision* **1,2** (avril 1987).
- [AF99] B. AMERI ET D. FRITSCH, 3-d reconstruction of polyhedral-like building models, in *ISPRS Conference: Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, 1999, pages 15–20.
- [AF00] B. AMERI ET D. FRITSCH, Automatic 3d building reconstruction using plane-roof structures, in *A paraître dans Annual ASPRS conference 2000*, Washington DC, 2000.
- [AG92] P. ASCHWANDEN ET W. GUGGENBÜHL, Experimental results from a comparative study on correlation-type registration algorithms, in *ISPRS Workshop*, Bonn, Germany, 1992.
- [AGLM92] LUIS ALVAREZ, FRÉDÉRIC GUICHARD, PIERRE-LOUIS LIONS ET JEAN-MICHEL MOREL, Axioms and Fundamental Equations of Image Processing, rapport technique numéro 9231, CEREMADE, 1992.
- [ALM92] L. ALVAREZ, P-L. LIONS ET J-M. MOREL, Image selective smoothing and edge detection by nonlinear diffusion (II), *SIAM Journal of numerical analysis* **29** (1992), 845–866.
- [Alo90] J.Y. ALOIMONOS, Perspective approximations, *Image and Vision Computing* **8,3** (août 1990), 179–192.
- [Aya89] N. AYACHE, *Vision Stéréoscopique et Perception Multisensorielle: Applications à la Robotique Mobile*, InterEditions, Paris, 1989.
- [Bai97] CAROLINE BAILLARD, *Analyse d'image aériennes stéréoscopiques pour la restitution 3-D des milieux urbains*, PhD thesis, École Nationale Supérieure des Télécommunications, octobre 97.
- [Bar89] S.T. BARNARD, Stochastic Stereo Matching Over Scale, in *The International Journal of Computer Vision*, volume 3, pages 17–32, 1989.
- [Bas94] BÉNÉDICTE BASCLE, *Contributions et applications des modèles déformables en vision par ordinateur*, PhD thesis, Université de Nice Sophia-Antipolis, juillet 1994.
- [Bel96] P. N. BELHUMEUR, A bayesian approach to binocular stereopsis, *The International Journal of Computer Vision*, 1996.
- [BGGL95] M. BERTHOD, L. GABET, G. GIRAUDON ET J. LOTTI, High-resolution stereo for the detection of buildings, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, A. Gruen, O. Kuebler et P. Agouris (édité par), ETH, Ascona, Switzerland, avril 1995, pages 135–144, Birkhauser.
- [BH98] C. BRENNER ET N. HAALA, Fast production of virtual city models, in *IAPRS Vol.32, Part 4*, Stuttgart, 1998, pages 77–84.
- [BHFS96] FRANK BIGNONE, OLOF HENRICSSON, PASCAL FUA ET MARKUS STRICKER, Automatic extraction of generic house roofs from high resolution aerial imagery, in *Proceedings of the 4th European Conference on Computer Vision*, Bernard Buxton (édité par), Cambridge, UK, avril 1996, pages 85–96.
- [BS98] K. BUBNA ET C.V. STEWART, Model selection and surface merging in reconstruction algorithms, in *Proceedings of the 6th International Conference on Computer Vision*, Bombay, India, janvier 1998, pages 895–902, IEEE Computer Society Press.
- [BV91] RUUD BOLLE ET BABA VEMURI, On three-dimensional surface reconstruction methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14,1** (janvier 1991), 1–13.
- [BZF99] C. BAILLARD, A. ZISSERMAN ET A. FITZGIBBON, Automatic line matching and 3d reconstruction of buildings from multiple views, in *ISPRS Conference: Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, 1999, pages 69–80.

- [Can86] J. F. CANNY, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**,6 (novembre 1986), 769–798.
- [Can97] DAVID CANU, *Vision Stéréoscopique à Partir d'un grand nombre d'images: Application à l'Imagerie Aérienne*, PhD thesis, Université de Rouen, décembre 1997.
- [CBL⁺99] R. CHELLAPPA, P. BURLINA, C.L. LIN, X. ZHANG, L.S. DAVIS ET A. ROSENFELD, Site model based image registration and change detection, in *University of Maryland*, 1999.
- [CHR⁺96] R. COLLINS, A. HANSON, E. RISEMAN, C. JAYNES, F. STOLLE X. WANG ET Y. CHENG, Umass progress in 3d building model acquisition, in *ARPA IUW*, Palm Springs, CA, 1996.
- [CJCP99] M. CORD, M. JORDAN, J. COCQUEREZ ET N. PAPARODITIS, Automatic extraction and modelling of urban building from high resolution aerial images, in *ISPRS Conference: Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, 1999, pages 187–192.
- [CJS⁺95] R. COLLINS, C. JAYNES, F. STOLLE, X. WANG, Y. CHENG, A. HANSON ET E. RISEMAN, A system for automated site model acquisition, in *SPIE*, Orlando, FL, 1995.
- [CKS97] VINCENT CASELLES, RON KIMMEL ET GUILLERMO SAPIRO, Geodesic active contours, *The International Journal of Computer Vision* **22**,1 (1997), 61–79.
- [CL97] LIANG-HUA CHEN ET WEI-CHUNG LIN, Visual surface segmentation from stereo, *Image and Vision Computing* **15** (1997), 95–106.
- [Coh91] L.D. COHEN, On active contour models and balloons, *CVGIP: Image Understanding* **53** (1991), 211–218.
- [Col96] R. COLLINS, A space-sweep approach to true multi-image matching, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, San Francisco, CA, juin 1996, IEEE.
- [Coo98] S. COORG, *Pose Imagery and Automated Three-Dimensional Modeling of Urban Environments*, PhD thesis, Massachusetts Institute of Technology, 1998.
- [Cor98] MATTHIEU CORD, *Analyse d'images aériennes haute résolution: détection et modélisation du bâti en zone urbaine*, PhD thesis, Université de Cergy-Pontoise, ENSEA-ETIS, décembre 1998.
- [CT98] SATYAN COORG ET SETH TELLER, Automatic extraction of textured vertical facades from pose imagery, rapport technique numéro 729, M.I.T. Laboratory for Computer Science, 1998.
- [CZD⁺93] R. CHELLAPPA, Q. ZHENG, L.S. DAVIS, D. DEMENTHON ET A. ROSENFELD, Site-model-based change detection and image registration, in *DARPA93*, pages 205–216, 1993.
- [DA89] UMESH R. DHOND ET J.K. AGGARWAL, Structure from stereo - a review, *IEEE Transactions on Systems, Man, and Cybernetics* **19**,6 (1989), 1489–1510.
- [DBY98] PAUL E. DEBEVEC, GEORGE BORSHUKOV ET YIZHOU YU, Efficient view-dependent image-based rendering with projective texture-mapping, in *9th Eurographics workshop on Rendering*, Vienna, Austria, juin 1998.
- [Deb96] PAUL E. DEBEVEC, *Modeling and Rendering Architecture from Photographs*, PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996.
- [Der87] R. DERICHE, Using canny's criteria to derive a recursively implemented optimal edge detector, *The International Journal of Computer Vision* **1**,2 (mai 1987), 167–187.
- [Dev97] FRÉDÉRIC DEVERNAY, *Vision stéréoscopique et propriétés différentielles des surfaces*, PhD thesis, École Polytechnique, février 97.
- [DF95] R. DERICHE ET O. FAUGERAS, Les EDP en traitement des images et vision par ordinateur, rapport technique numéro 2697, INRIA, novembre 1995. A more complete version of this Research Report has appeared in the French Revue "Traitement du Signal". Volume 13 - No 6 - Special 1996.
- [DF96] R. DERICHE ET O. FAUGERAS, Les EDP en Traitement des Images et Vision par Ordinateur, *Traitement du Signal* **13**,6 (1996).
- [DF00] F. DEVERNAY ET O. FAUGERAS, Shape from stereo using fine correlation: Method and error analysis, *Image and Vision Computing*, 2000. Submitted.
- [DJ95] OLIVIER DISSARD ET OLIVIER JAMET, 3d-reconstruction of buildings from stereo images using both monocular analysis and stereo-matching: an assessment within the context of cartographic production, in *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, SPIE (édité par), Orlando, Floride, 1995, volume 2486, pages 255–266.
- [DM00] H. DELINGETTE ET J. MONTAGNAT, Topology and shape constraints on parametric active contours, in *Inria Research Report*, INRIA numéro 3880, 2000.

- [DTM96] P.E. DEBEVEC, C.J. TAYLOR ET J. MALIK, Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach, in *SIGGRAPH*, New Orleans, août 1996, pages 11–20.
- [Fau93] O. FAUGERAS, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.
- [Fau95] OLIVIER FAUGERAS, Stratification of 3-D vision: projective, affine, and metric representations, *Journal of the Optical Society of America A* **12**,3 (mars 1995), 465–484.
- [FB81] M.A. FISCHLER ET R.C. BOLLES, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* **24** (1981), 381–385.
- [FB96] P. FUA ET C. BRECHBUHLER, Consistent site modeling: Imposing hard constraints on deformable models, in *ARPA IUW*, Palm Springs, CA, 1996, pages 1077–1094.
- [Fil95] ALAIN FILBOIS, *Contributions à la modélisation automatique d'objets polyédriques 3D: extraction de primitives 3D, facettes et segments*, PhD thesis, Institut National Polytechnique de Lorraine, juillet 1995.
- [FK98] OLIVIER FAUGERAS ET RENAUD KERIVEN, Complete dense stereovision using level set methods, in *Proceedings of the 5th European Conference on Computer Vision*, H. Burkhardt et B. Neumann (édité par), Freiburg, Germany, juin 98, pages 379–393, *Lecture Notes in Computer Science*, Springer-Verlag.
- [FKL⁺98] F. FISHER, T. KOLBE, F. LANG, A. CREMERS, W. FÖRSTNER, L. PLÜMER ET V. STEINHAGE, Extracting building from aerial images using hierarchical aggregation in 2d and 3d, *Computer Vision and Image Understanding* **72**,2 (février 1998), 195–203.
- [FL90] P. FUA ET Y.G. LECLERC, Model driven edge detection, *Machine Vision and Application* **3** (1990), 45–56.
- [FR96] OLIVIER FAUGERAS ET LUC ROBERT, What can two images tell us about a third one?, *The International Journal of Computer Vision* **18**,1 (avril 1996), 5–20.
- [FRM99] M. FRADKIN, M. ROUX ET H. MAÎTRE, Building detection from multiple aerial images, in *ISPRS Conference: Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, avril 1999.
- [FS92] P. FUA ET P. SANDER, Reconstructing surfaces from unstructured 3d points, in *Proc. Image Understanding Workshop*, San Diego, California, janvier 1992.
- [FT86] OLIVIER FAUGERAS ET GIORGIO TOSCANI, The calibration problem for stereo, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, juin 1986, pages 15–20, IEEE.
- [Fua91] P. FUA., Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities, in *International Joint Conference on Artificial Intelligence*, Sydney, Australia, août 1991.
- [Fua95] P. FUA, Parametric models are versatile: The case of model based optimization, in *ISPRS WG III/2 Joint Workshop*, Stockholm, Sweden, septembre 1995.
- [Fua96a] P. FUA, Cartographic applications of model-based optimization, in *ARPA96*, pages 409–420, 1996.
- [Fua96b] PASCAL FUA, Model-based optimization: Accurate and consistent site modeling, in *XVIII ISPRS Congress*, Vienna, Austria, juillet 1996.
- [Fua97] PASCAL FUA, From multiple stereo views to multiple 3-d surfaces, *The International Journal of Computer Vision* **24**,1 (août 1997), 19–35.
- [GF99] J. GOMES ET O. FAUGERAS, Reconciling distance functions and level sets, in *Inria Research Report*, INRIA numéro 3666, 1999.
- [GGR94] LAURENT GABET, GÉRARD GIRAUDON ET LAURENT RENOARD, Construction automatique de modèle numérique de terrain haute résolution en zone urbaine, in *Bulletin S.F.T.P.*, volume 135, pages 9–25, 1994.
- [GGR97] L. GABET, G. GIRAUDON ET L. RENOARD, Automatic generation of high resolution urban zone digital elevation model, *ISPRS Journal of Photogrammetry and Remote Sensing* **52**,1 (février 1997), 33–47.
- [GL95] A. GRUEN ET H. LI, Road extraction from aerial and satellite images by dynamic-programming, *ISPRS Journal of Photogrammetry and Remote Sensing* **50**,4 (august 1995), 11–20.
- [GLY95] DAVI GEIGER, BRUCE LADENDORF ET ALAN YUILLE, Occlusions and binocular stereo, *The International Journal of Computer Vision* **14** (mars 1995), 211–226.

- [GML99] E. GÜLCH, H. MÜLLER ET T. LÄBE, Integration of automatic processes into semi-automatic building extraction, in *ISPRS Conference: Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, 1999, pages 177–186.
- [Gri81] W.E.L. GRIMSON, A computer implementation of a theory of human stereo vision, *Philosophical Transactions of the Royal Society of London, B.* **292**,1058 (1981), 217–253.
- [Gri85] W.E.L. GRIMSON, Computational experiments with a feature based stereo algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **7**,1 (1985), 17–34.
- [GW98] A. GRUEN ET X. WANG, Cc-modeler: a topology generator for 3-d city models, *ISPRS Journal of Photogrammetry and Remote Sensing* **53**,5 (octobre 1998), 286–295.
- [HA89] WILLIAM HOFF ET NARENDRA AHUJA, Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**,2 (février 1989), 121–136.
- [HB97] N. HAALA ET C. BRENNER, Interpretation of urban surface models using 2d building information, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images II*, A. Gruen, E. Baltsavias et O. Henricsson (édité par), ETH, Ascona, Switzerland, mai 1997, pages 213–222, Birkhauser.
- [Hen96] OLOF HENRICSSON, *Analysis of image structures using color attributes and similarity relations*, PhD thesis, Swiss Federal Institute of Technology Zurich, novembre 96.
- [HFC96] AARON J. HELLER, PASCAL FUA, C. CONNOLLY ET J. SARGENT, The site-model construction component of the RADIUS testbed system, in *1996 Proceedings, Image Understanding Workshop*, page 345, 1996.
- [HG94] R. HARTLEY ET R. GUPTA, Linear pushbroom cameras, in *Proceedings of the 3rd European Conference on Computer Vision*, J.-O. Eklundh (édité par), Stockholm, Sweden, mai 1994, *Lecture Notes in Computer Science* volume 800-801, Springer-Verlag.
- [HH95] N. HAALA ET M. HAHN, Data fusion for the detection and reconstruction of buildings, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, A. Gruen, O. Kuebler et P. Agouris (édité par), ETH, Ascona, Switzerland, avril 1995, pages 211–220, Birkhauser.
- [HJB⁺96] A. HOOVER, G. JEAN-BAPTISTE, X.Y. JIANG, P.J. FLYNN, H. BUNKE, D.B. GOLDFOG, K. BOWYER, D.W. EGGERT, A.W. FITZGIBBON ET R. B. FISHER, An experimental comparison of range image segmentation algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**,7 (july 1996), 673–689.
- [HKN90] A. HUERTAS, Z. KIM ET R. NEVATIA, Multisensor integration for building modeling, in *CVPR00*, pages II:203–210, 190.
- [HNT⁺96] W.H. HUDSON, D.C. NADADUR, K.B. THORNTON, X. LIU ET R.M. HARALICK, The radius cdrom ground truthed data set, in *ARPA96*, pages 511–519, 1996.
- [Hor86] B.K. HORN, *Robot Vision*, MIT Press, 1986.
- [HQ97] AARON J. HELLER ET L. H. QUAM, The RADIUS common development environment, in *RADIUS: Image Understanding for Imagery Intelligence*, Oscar Firschein et Tom Strat (édité par), San Mateo, CA, 1997.
- [HS89] R. HORAUD ET TH. SKORDAS, Stereo correspondence through feature grouping and maximal cliques, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**,11 (1989), 1168–1180.
- [Hsi96] Y. HSIEH, Sitecity: A semi-automated site modeling system, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, San Francisco, CA, juin 1996, pages 499–506, IEEE.
- [Hub81] P.J. HUBER, *Robust Statistics*, John Wiley & Sons, New York, 1981.
- [HVF⁺97] M. HENDRICKX, J. VANDEKERCKHOVE, D. FRERE, T. MOONS ET L. VAN GOOL, 3d reconstruction of house roofs from multiple aerial images of urban areas, in *ISPRS Workshop: 3D Reconstruction and Modelling of Topographic Objects*, Stuttgart, Germany, 1997, pages 88–95.
- [IB95] STEPHEN S. INTILLE ET AARON F. BOBICK, Incorporating intensity edges in the recovery of occlusion regions, rapport technique numéro 246, M.I.T. Media Lab Perceptual Computing Group, 1995.
- [IG98] H. ISHIKAWA ET D. GEIGER, Occlusions, discontinuities, and epipolar line in stereo, in *Proceedings of the 5th European Conference on Computer Vision*, H. Burkhardt et B. Neumann (édité par), Freiburg, Germany, juin 1998, *Lecture Notes in Computer Science*, Springer-Verlag.
- [JC95] M. JORDAN ET J.-P. COCQUEREZ, Three-dimensional description of scenes observed in aerial photography, *Pattern Recog.* **28**,7 (1995), 931–947.

- [JHR97] C. JAYNES, A. HANSON ET E. RISEMAN, Building reconstruction from optical and range images, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, San Juan, Puerto Rico, juin 1997.
- [JJ94] MCGLONE J.C. ET SHUFELT J.A., Projective and object space geometry for monocular building extraction, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Seattle, WA, juin 1994, pages 54–61, IEEE.
- [JMM99] O. JAMET, H. MAÎTRE ET H. LE MEN, Applying the theory of evidence to vector-d.e.m. comparison for the building planimetric change detection, in *ISPRS Conference: Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, 1999.
- [Kan96] K. KANATANI, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, 1996.
- [Kan98] KENICHI KANATANI, Statistical optimization and geometric inference in computer vision, *Phil. Trans. R. Soc. Lond. A*, 356 (1998), 1303–1320.
- [KM95] T. KIM ET J. MULLER, Building extraction and verification from spaceborn and aerial imagery using image understanding fusion techniques, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, A. Gruen, O. Kuebler et P. Agouris (édité par), ETH, Ascona, Switzerland, avril 1995, pages 221–230, Birkhauser.
- [KO94] T. KANADE ET M. OKUTOMI, A stereo matching algorithm with an adaptive window: Theory and experiment, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**,9 (septembre 1994), 920–932.
- [Kos93] ANDREAS KOSCHAN, What is new in computational stereo since 1989: A survey on current stereo papers, rapport technique numéro 93-22, Technical University of Berlin, Department of Computer Science, août 1993.
- [KWT87] M. KASS, A. WITKIN ET D. TERZOPOULOS, Snakes: Active contour models, in *First International Conference on Computer Vision*, London, juin 1987, pages 259–268.
- [L99] Y. LÉCHERVY, *Acquisition vectorielle 3D de milieux urbains pour les télécommunications mobiles*, PhD thesis, Université Paris Sud - Paris XI, septembre 1999.
- [Leo94] ALEŠ LEONARDIS, Recover and select paradigm—a robust approach to estimation of parametric models, in *Performance versus Methodology in Computer Vision NSF/ARPA Workshop*, Seattle, WA, juin 1994, pages 78–87.
- [LG98] T. LÄBE ET E. GÜLCH, Robust techniques for estimating parameters of 3d building primitives, in *ISPRS Symposium, Comm. II, IAPR Vol.32, Part 2*, Cambridge, 1998.
- [LH81] H.C. LONGUET-HIGGINS, A computer algorithm for reconstructing a scene from two projections, *Nature* **293** (1981), 133–135.
- [LHN94] CHUNGAN LIN, ANDRES HUERTAS ET RAMAKANT NEVATIA, Detection of buildings using perceptual grouping and shadows, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Seattle, WA, juin 1994, pages 62–69, IEEE.
- [LHT96] X. LIU, R.M. HARALICK ET K. THORNTON, Site model construction using geometric constrained optimization, in *ARPA96*, pages 357–372, 1996.
- [Lin94] TONY. LINDBERG, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, 1994.
- [LMP98] KIL-MOO LEE, PETER MEER ET RAE-HONG PARK, Robust adaptive segmentation of range images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**,2 (février 1998), 200–205.
- [LNN99] J. LI, R. NEVATIA ET S. NORNOHA, User assisted modeling of buildings from aerial images, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, 1999, pages II:274–279, IEEE.
- [Lot96] JEAN-LUC LOTTI, *Mise en correspondance stéréo par fenêtres adaptatives en imagerie haute résolution*, PhD thesis, Université de Nice - Sophia Antipolis, France, février 1996.
- [LT88] R. K. LENZ ET R. Y. TSAI, Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10** (1988), 713–720.
- [May98] H. MAYER, Three dimensional generalization of buildings based on scale-spaces, rapport technique, Technische Universität München, 1998.
- [May99] HELMUT MAYER, Automatic object extraction from aerial imagery—a survey focusing on buildings, *Computer Vision and Image Understanding* **74**,2 (mai 1999), 138–149.

- [MD94] ROUX M. ET MCKEOWN D.M., Feature matching for building extraction from multiple views, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Seattle, WA, juin 1994, pages 46–53, IEEE.
- [MD96] M. MARESH ET P. DURACHER, The geometric design of a vehicle based 3 line ccd camera system, in *ISPRS 96, Commission I, WG 1/3*, pages 121–127, 1996.
- [MF81] J.E.W. MAYHEW ET J.P. FRISBY, Psychophysical and computational studies towards a theory of human stereopsis, *Artificial Intelligence* **17** (1981), 349–386.
- [MFVG98] THEO MOONS, DAVID FRÈRE, JAN VANDEKERCKHOVE ET LUC VAN GOOL, Automatic modeling and 3d reconstruction of urban house roofs from high resolution aerial imagery, in *Proceedings of the 5th European Conference on Computer Vision*, H. Burkhardt et B. Neumann (édité par), Freiburg, Germany, juin 98, pages 410–425, *Lecture Notes in Computer Science*, Springer-Verlag.
- [ML92] HENRI MAÎTRE ET WEI LUO, Using models to improve stereo reconstruction, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14,2** (février 1992), 269–277.
- [MMRK91] P. MEER, D. MINTZ, A. ROSENFELD ET D.Y. KIM, Robust regression methods for computer vision: A review, *The International Journal of Computer Vision* **6,1** (1991), 59–70.
- [MP76] D. MARR ET T. POGGIO, Cooperative computation of stereo disparity, *Science* **194** (1976), 283–287.
- [MP77] D. MARR ET T. POGGIO, A theory of human stereo vision, memorandum aim numéro 451, MIT Artif. Intell. Lab, 1977.
- [MS96a] M. MARESH ET D. SCHEIBLHOFER, A vehicle based multi-sensor platform for facade recording, in *Proceedings of the 20th Workshop of Austrian Association for Pattern Recognition*, pages 195–206, 1996.
- [MS96b] J.V. MILLER ET C.V. STEWART, Muse: Robust surface fitting using unbiased scale estimates, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 300–306, 1996.
- [MV99] H. MAAS ET G. VOSSELMAN, Two algorithms for extracting building models from raw laser altimetry data, *ISPRS Journal of Photogrammetry and Remote Sensing* **54,2-3** (juillet 1999), 153–163.
- [NFI⁺94] W. NEUENSCHWANDER, P. FUA, L. IVERSON, G. SZÉKELY ET O. KUBLER, Ziplock Snakes, *The International Journal of Computer Vision*, 1994. In press, available as Tech Note 548, Artificial Intelligence Center, SRI International.
- [NHK99] R. NEVATIA, A. HUERTAS ET Z. KIM, The muri project for rapid feature extraction in urban areas, in *ISPRS Conference Automatic Extraction of GIS Objects from Digital Imagery*, München, Germany, septembre 1999, pages 3–14.
- [NN97] S. NORONHA ET R. NEVATIA, Detection and description of buildings from multiple aerial images, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Puerto Rico, juin 1997, pages 588–594.
- [OK85] Y. OHTA ET T. KANADE, Stereo by intra- and inter-scanline search, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **7** (1985), 139–154.
- [OK93] M. OKUTOMI ET TAKEO KANADE, A multiple-baseline stereo, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15,4** (avril 1993), 353–363.
- [OS88] S. OSHER ET J. SETHIAN, Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation, *Journal of Computational Physics* **79** (1988), 12–49.
- [Pap98] N. PAPARODITIS, *Reconstruction Tridimensionnelle de Paysages Urbains en Imagerie Stéréoscopique Haute Résolution*, PhD thesis, Université de Nice Sophia-Antipolis, 1998.
- [PH74] THEODOSIOS PAVLIDIS ET STEVEN L. HOROWITZ, Segmentation of plane curves, *IEEE Transactions on Computers* **23,C** (1974), 860–870.
- [PMF91] S.B. POLLARD, J.E.W. MAYHEW ET J.P. FRISBY, Implementation details of the pmf stereo algorithm, *3D Model Recognition From Stereoscopic Cues*, 1991, 33–39.
- [RC98] S. ROY ET I. COX, A maximum-flow formulation of the n-camera stereo correspondence problem, janvier 1998, 492–499.
- [RD89] IRVIN R.B. ET MCKEOWN D.M., Methods for exploiting the relationship between buildings and their shadows in aerial imagery, *SMC* **19,6** (1989), 1564–1575.
- [RD95] L. ROBERT ET R. DERICHE, Dense depth map reconstruction using a multiscale regularization approach which preserves discontinuities, in *Proceedings of the International Workshop on Ste-*

- reoscopic and Three-Dimensional Imaging*, S. Efstratiadis et al. (édité par), pages 32–39, The European Project RACE 2045-DISTIMA, septembre 1995.
- [RL87] P.J. ROUSSEEUW ET A.M. LEROY, *Robust Regression and Outlier Detection*, John Wiley & Sons, New York, 1987.
- [RM97] M. ROUX ET M. MAÎTRE, Three-dimensional description of dense urban areas using maps and aerial images, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images II*, A. Gruen, E. Baltsavias et O. Henricsson (édité par), ETH, Ascona, Switzerland, mai 1997, pages 311–322, Birkhauser.
- [RMG97] M. ROUX, H. MAÎTRE ET S. GIRARD, A step toward stereo reconstruction of urban aerial images, in *ISPRS Workshop: 3D Reconstruction and Modelling of Topographic Objects*, Stuttgart, Germany, 1997, pages 107–114.
- [SB94] B. SERRA ET M. BERTHOD, Subpixel contour matching using continuous dynamic programming, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Seattle, WA, juin 1994, pages 202–207, IEEE.
- [Set96] J. A. SETHIAN, *Level Set Methods*, Cambridge University Press, 1996.
- [SFB96] CHARLES STEWART, ROBIN FLATLAND ET KISHORE BUBNA, Geometric constraints and stereo disparity computation, *The International Journal of Computer Vision* **20**,3 (1996), 143–168.
- [SGJ97] U. STILLA, R. GEIBEL ET K. JURKIEWICZ, Building reconstruction using different views and context knowledge, in *ISPRS Workshop: 3D Reconstruction and Modelling of Topographic Objects*, Stuttgart, Germany, 1997, pages 129–136.
- [Sha93] J. SHAH, A nonlinear diffusion model for discontinuous disparity and half-occlusions in stereo, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, New-York, NY, juin 1993, IEEE.
- [SL95a] MARKUS STRICKER ET ALEŠ LEONARDIS, Exsel++ a general framework to extract parametric models, rapport technique numéro BIWI-TR-159, ETH Zürich, février 1995.
- [SL95b] MARKUS STRICKER ET ALEŠ LEONARDIS, Tabu search: An efficient solution to discrete optimization, rapport technique numéro BIWI-TR-161, ETH Zürich, mars 1995.
- [Sla80] C. C. SLAMA (édité par), *Manual of Photogrammetry*, American Society of Photogrammetry, édition fourth, 1980.
- [SML96] U. STILLA, E. MICHAELSEN ET K. LÜTJEN, Automatic extraction of buildings from aerial images, *IAPR Workshop Graz Austria*, 1996.
- [SS88] D. SCHARSTEIN ET R. SZELISKI, Stereo matching with nonlinear diffusion, *The International Journal of Computer Vision* **2**,28 (juillet 1988), 155–174.
- [Ste95] C.V. STEWART, Minpran: A new robust estimator for computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**,10 (octobre 1995), 925–938.
- [Ste99] CHARLES V. STEWART, Robust parameter estimation in computer vision, *SIAM Reviews* **41**,3 (septembre 1999), 513–537.
- [TA77] A.N. TIKHONOV ET V.Y. ARSENIN, *Solutions of Ill-posed Problems*, Winston and Sons, Washington, D.C., 1977.
- [Tau91] G. TAUBIN, Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**,11 (novembre 1991), 1115–1138.
- [Ter86] D. TERZOPOULOS, Image analysis using multigrid relaxation methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**,2 (mars 1986), 129–139.
- [TL95] J. TRINDER ET H. LI, Semi-automatic feature extraction by snakes, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, A. Gruen, O. Kuebler et P. Agouris (édité par), ETH, Ascona, Switzerland, avril 1995, pages 95–104, Birkhauser.
- [TM97] P. H. S. TORR ET D. W. MURRAY, The development and comparison of robust methods for estimating the fundamental matrix, *IJCV* **24**,3 (1997), 271–300.
- [TM98] CHI-KEUNG TANG ET GERARD MEDIONI, Inference of integrated surface, curve, and junction descriptions from sparse 3d data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**,11 (novembre 1998), 1206–1223.
- [Tor97] P.H.S. TORR, An assessment of information criteria for motion model selection, in *Comp. Vision and Pattern Rec.*, IEEE Computer Society Press (édité par), pages 47–52, 1997.
- [Tor99] PHILIP H. S. TORR, Model selection for structure and motion recovery from multiple images, rapport technique numéro MSR-TR-99-16, Microsoft, mars 1999.

- [TZ00] P. TORR ET A. ZISSERMAN, Mlesac: A new robust estimator with application to estimating image geometry, *Computer Vision and Image Understanding* **78**,1 (avril 2000), 138–156.
- [VBD97] CHRISTOPHE VESTRI, FRANK BIGNONE ET FRÉDÉRIC DEVERNAY, Extraction semi-automatique de bâtiments à partir d'imagerie aérienne et spatiale haute résolution, in *Journées ORASIS'97*, INRIA (édité par), pages 173–183, octobre 1997.
- [VD99] CHRISTOPHE VESTRI ET FRÉDÉRIC DEVERNAY, "redressement de façades", propositions et étude d'une nouvelle méthodologie pour l'aide à la reconstruction automatique de bâtiments, in *ISPRS Workshop (submission)*, Institut Géographique National, St Mandé, France, avril 1999.
- [VD00] CHRISTOPHE VESTRI ET FRÉDÉRIC DEVERNAY, Improving correlation-based dems by image warping and facade correlation, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2000.
- [vdH98] F. VAN DEN HEUVEL, 3d reconstruction from single image using geometric constraints, *ISPRS Journal of Photogrammetry and Remote Sensing* **53**,6 (décembre 1998), 354–368.
- [vdH99] F. VAN DEN HEUVEL, Estimation of interior orientation parameters from constraints on line measurements in a single image, in *International Archives of Photogrammetry and Remote Sensing, Vol. 32, Part 5W11*, pages 81–88, 1999.
- [VDM00] CHRISTOPHE VESTRI, FRÉDÉRIC DEVERNAY ET MARION MESNAGE, Using robust methods to model buildings with a dem, in *soumis à Models and Strategies for Object Reconstruction, ISPRS Workshop*, Bonn, Allemagne, avril 2000. Note: workshop annulé, à paraître RR-INRIA.
- [Wei97] U. WEIDNER, Digital surface models for building extraction, in *Automatic Extraction of Man-Made Objects from Aerial and Space Images II*, A. Gruen, E. Baltsavias et O. Henricsson (édité par), ETH, Ascona, Switzerland, mai 1997, pages 193–202, Birkhauser.
- [WF95] UWE WEIDNER ET WOLFGANG FÖRSTNER, Toward automatic building extraction from high resolution digital elevation models, in *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 50, pages 38–49, 1995.
- [WG96] S. WEIK ET O. GRAU, Recovering 3-d object geometry using a generic constraint description, in *18th Congress of the International Society for Photogrammetry and Remote Sensing*, Vienne, 1996, pages 9–19.
- [WH97] X. WANG ET A. HANSON, Extracting surface microstructures from multiple aerial images, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, San Juan, Puerto Rico, juin 1997.
- [XZ96] GANG XU ET ZHENGYOU ZHANG, *Epipolar Geometry in Stereo, Motion and Object Recognition*, Kluwer Academic Publishers, 1996.
- [YBK94] X.M. YU, T.D. BUI ET A. KRZYSAK, Robust estimation for range image segmentation and reconstruction, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**,5 (may 1994), 530–538.
- [Zel96] CYRIL ZELLER, *Calibration Projective Affine et Euclidienne en Vision par Ordinateur*, PhD thesis, École Polytechnique, février 1996.
- [Zha93] ZHENGYOU ZHANG, Le problème de la mise en correspondance: l'état de l'art, rapport de recherche numéro 2146, INRIA, décembre 1993.
- [Zha96] Z. ZHANG, Determining the epipolar geometry and its uncertainty: A review, rapport technique numéro 2927, INRIA Sophia-Antipolis, France, juillet 1996.
- [Zha97] ZHENGYOU ZHANG, Parameter estimation techniques: A tutorial with application to conic fitting, *Image and Vision Computing Journal* **15**,1 (1997), 59–76.

Résumé

Disposer de descriptions tridimensionnelles d'une scène urbaine intéresse de plus en plus d'applications industrielles. Les fournisseurs de services en télécommunications ont besoin de Modèles Numériques Urbains (MNU) précis pour alimenter les outils de planification qui leur permettent de déployer et de densifier leurs réseaux cellulaires. Avec comme objectif la fourniture d'un produit destiné aux applications des télécommunications, cette thèse traite de la modélisation de bâtiments.

Les contributions principales de cette thèse sont la présentation, l'étude et le développement de plusieurs outils pour la reconstruction automatique de bâtiments. Une contribution originale et importante est l'algorithme de reconstruction automatique des façades. Il permet la reconstruction des façades sans aucun a priori sur la forme du bâtiment traité ni sur l'orientation de ses façades. Nous montrons comment utiliser les points reconstruits sur les façades pour corriger les limites des bâtiments dans un MNE (Modèle Numérique d'Élévation) ou pour guider des contours actifs.

Nous présentons aussi un système de modélisation automatique des bâtiments basé sur les outils que nous avons développés. Ce système de modélisation de scène urbaine se compose des trois étapes suivantes : (1) segmentation du MNE initial en plans horizontaux pour retrouver les régions planes qui permettent de décrire les toits des bâtiments, (2) approximation polygonale des contours des plans pour construire un modèle initial et (3) orthogonalisation de ces modèles de bâtiments. Nous proposons une procédure de validation des résultats basée sur des outils télécoms. Les résultats montrent que ces données sont satisfaisantes pour déployer un réseau et le maintenir dans une zone rurale et sub-urbaine.

Mots-clé : Reconstruction de bâtiments, Reconstruction de façades, MNE, Image-rie aérienne, Segmentation en plans, Vectorisation, Stéréoscopie, Télécommunications mobiles

Abstract

Extracting building descriptions from an urban scene in 3D is an essential task for many industrial applications. Telecom Service providers require accurate Digital Elevation Models (DEM) for their planification tools which permit to deploy or to densify their cellular phone networks. With the objective of providing a product for telecom applications, this thesis focuses on building modeling.

The main contributions of this thesis are the presentation, the study and the development of multiple tools for automatic building modeling. An original and important contribution is the automatic algorithm for reconstruction of the facades. It enables the reconstruction of facades without any a priori information about the orientation of the facades and the shape of the building. We propose to use the reconstructed points on the facades to correct the limits of the buildings in a DEM or to drive active contours.

We also present an automatic modeling system of buildings based on the tools that we have proposed. This system of modeling is composed of the three following stages: (1) segmentation of the initial DEM in planar patches, (2) polygonal approximation of the contours of the planar patches for building an initial model and (3) orthogonalization of the building model. Finally, we propose to validate the results with telecom tools. Results seem satisfactory to deploy a mobile network in an rural or sub-urban area.

Keywords: Building modeling, Facades modeling, DEM, Aerial imagery, Planar segmentation, Vectorization, Stereoscopy, Mobile telecommunications