Localisation par l'image en milieu urbain : application à la réalité augmentée

Antoine Fond

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Localisation par l'image en milieu urbain : application à la réalité augmentée

THÈSE

présentée et soutenue publiquement le 06 avril 2018

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Antoine Fond

Composition du jury

Rapporteurs :	Vincent Lepetit, Professeur – Université de Bordeaux Frédéric Jurie, Professeur – Université de Caen - Normandie
Examinateurs :	Nicolas Paparoditis, Directeur de Recherche – ENSG Antoine Tabbone, Professeur – Université de Lorraine
Encadrants :	Marie-Odile Berger, Directrice de Recherche – INRIA Gilles Simon, Maitre de Conférence – Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

« ช้าๆได้พร้าสองเล่มงาม »

proverbe thaï

Remerciements

Je tiens tout d'abord à remercier Vincent Lepetit et Frédéric Jurie d'avoir accepté de relire ma thèse. Leurs remarques éclairées sur mes travaux ont été très précieuses. Merci également à Antoine Tabbone d'avoir accepté de participer à mon jury et à Nicolas Paparoditis qui m'a fait l'honneur de le présider.

Je souhaite remercier ma directrice de thèse Marie-Odile Berger qui a su trouver le bon équilibre de travail entre libertés d'exploration et nécessité d'avancement. Sa confiance en ma compréhension des réseaux de neurones convolutionnels m'a permis de développer de nouvelles idées sans perdre de vue les objectifs concrets de rédaction et de publication. Un grand merci également à mon codirecteur de thèse Gilles Simon pour sa disponibilité et son regard toujours pertinent sur mes travaux. Merci aussi à tous mes collègues et amis de l'équipe Magrit, Pierre, Pierre-Frédéric, Erwan, Cong, Vincent et les autres avec qui j'ai partagé des moments formidables.

Bien sur je n'aurais pas pu mener à bien cette thèse sans le soutien inconditionnel de ma famille, ma mère Marie-Jeanne, mon père Alain et ma soeur Anna. Enfin merci à ma femme Sasithon pour sa patience et sa confiance inébranlable durant ces trois années de thèse.

Table des matières

Table des matières **i**

Introduction **1**

1 Etat de l'art **5**
1.1 Solutions généralistes 5
1.1.1 Reconnaissance de lieux 5
1.1.2 Calcul de pose pour les scènes planes 7
1.2 Approches spécifiques aux milieux urbains 10
1.2.1 Détection et reconnaissance de façades (*bottom-up*) 10
1.2.2 Calcul de pose par recalage de modèle (*top-down*) 12
1.3 Outils mathématiques 14
1.3.1 Réseau de neurones convolutionnels 14
1.3.2 Algorithme d'Espérance-Maximisation 15

2 Estimation des points de fuite de Manhattan **19**
2.1 Travaux liés 20
2.2 Travaux précédents sur la recherche de points de fuite avec *A Priori* 21
2.3 Estimation et segmentation conjointes des directions de Manhattan 22
2.3.1 Architecture du réseau 24
2.3.2 Bases d'apprentissage 25
2.3.3 Entrainement du réseau 26
2.4 Raffinement du repère de Manhattan utilisant les segments de droites 28
2.4.1 Classification *A Priori* des segments de droite 29
2.4.2 Formulation bayésienne 30
2.4.3 Résolution par Espérance-Maximisation 33
2.5 Efficacité et applications 36
2.5.1 Détails d'implémentation et efficacité 36
2.5.2 Orthorectification des images 36
2.6 Résultats et limites 37
2.6.1 Jeux de données de tests 37
2.6.2 Résultats 38

2.6.3 Limites . . . 43
2.7 Conclusion . . . 44

3 Propositions de façades pour la détection et la reconnaissance de bâtiments 47
3.1 Travaux liés . . . 48
3.2 Propositions de façades . . . 49
3.2.1 Segmentation sémantique et détection conjointe de contour . . . 50
3.2.2 Génération de candidats rectangulaires . . . 52
3.2.3 Indices caractéristiques de façade . . . 52
3.2.4 Combinaison des indices . . . 61
3.3 Détection et reconnaissance de façades . . . 63
3.3.1 Classification de façades . . . 63
3.3.2 Mise en correspondance de façades . . . 64
3.4 Résultats expérimentaux . . . 65
3.4.1 Résultats concernant la proposition de façades . . . 65
3.4.2 Résultats relatifs à la reconnaissance de façades . . . 67
3.4.3 Applications à la réalité augmentée et au calcul de pose . . . 70
3.4.4 Réalité augmentée . . . 70
3.4.5 Initialisation de pose de caméra . . . 73
3.5 Conclusion . . . 76

4 Segmentation et recalage de façade conjoint 77
4.1 Travaux liés . . . 77
4.1.1 Segmentation sémantique de façades . . . 77
4.2 Initialisation du recalage et de la segmentation sémantique . . . 80
4.2.1 Initialisation du recalage par détection . . . 80
4.2.2 Initialisation de la segmentation sémantique . . . 80
4.3 Résolution jointe des problèmes de recalage et segmentation sémantique . . . 82
4.3.1 Formulation bayésienne . . . 82
4.3.2 Résolution par Espérance-Maximisation . . . 85
4.3.3 Discussions sur les cas de résolution . . . 87
4.4 Détails d'implémentation . . . 89
4.4.1 Efficacité de la méthode . . . 89
4.4.2 Schéma multi-résolutions . . . 89
4.5 Résultats et limites . . . 90
4.5.1 Jeux de données utilisés . . . 90
4.5.2 Résultats quantitatifs . . . 91
4.5.3 Résultats qualitatifs . . . 96
4.5.4 Limites de la méthode . . . 97
4.6 Conclusion . . . 101

Conclusion 103

A Annexe : Partitionnement optimal de l'espace couleur **107**

B Annexe : résolution du système polynomial $p = 4$ **109**
B.1 Système polynomial pour $p = 4$ 109
B.2 Gradient et hessien pour $p = 4$ 110

Publications personnelles **113**

Bibliographie **115**

Introduction

Définition générale du problème

Contexte

Dans cette thèse, nous nous intéressons au problème de la localisation en milieux urbains. Il s'agit pour une personne de pouvoir localiser précisément sa position et son orientation en ville. Ce positionnement est relatif à un modèle de l'environnement qui sert de référentiel (une carte par exemple). Un tel positionnement est fondamental pour des applications de réalité augmentée ou de robotique mobile.

En effet, l'objectif de la réalité augmentée est d'augmenter la vue du monde d'un utilisateur avec de l'information contextuelle. L'affichage de cette information dépend ainsi de la localisation de l'utilisateur. Bien souvent cette information, qui prend la forme d'objets virtuels, doit en plus se fondre naturellement avec la scène observée. Ainsi la géométrie, la texture, l'éclairage de ces objets virtuels ajoutés à la scène doivent être en accord avec l'image. Le rendu réaliste de tels objets nécessite donc un positionnement précis de l'utilisateur en translation et en rotation. Ce positionnement doit également être stable temporellement. En effet la qualité du suivi (*tracking*) dans les vidéos est critique pour garder la sensation d'immersion et éviter le phénomène de scintillement.

Les milieux urbains et les environnements industriels sont sans aucun doute les deux domaines les plus riches en applications de la réalité augmentée. Si l'exemple du rendu *in situ* d'un bâtiment pour un projet architectural est un exemple régulièrement utilisé [1], les secteurs d'application sont beaucoup plus diversifiés. Le secteur du tourisme pourrait par exemple introduire des anecdotes historiques sur les façades des bâtiments à visiter. Des informations en surimpression de l'image pourraient également bénéficier à la publicité dans les quartiers marchands. Si ce type d'information ne nécessite qu'un positionnement relatif une fois le bâtiment identifié, d'autres application ont besoin d'un positionnement global géoréférencé. Ainsi, on peut également penser à des annotations visuelles (flèches ou chemin tracé au sol) pour aider à trouver sa route vers une destination spécifique dans une ville inconnue. Le domaine de la maintenance pourrait également profiter d'applications en réalité augmentée pour les services de voirie en incorporant par exemple à l'image les plans des réseaux souterrains (électricité, gaz) [2]. Enfin le secteur du divertissement a déjà démontré son intérêt pour ce type d'application en réalité augmentée urbaine avec des

1. Trimble : https ://youtu.be/kXVW4sUsh3A
2. Bentley Systems : https ://youtu.be/KS_5OHoHHuo

succès vidéo-ludiques comme Ingress ou Pokémon Go.

Les milieux urbains sont également un contexte privilégié pour la robotique mobile. Localiser le robot dans son environnement est une problématique centrale dès lors que le robot peut se déplacer. D'autres tâches tirent directement parti d'un bon positionnement comme la planification de trajectoires ou le contrôle en boucle fermée. Dans les milieux urbains, cette étape est d'autant plus critique que le robot est amené à se déplacer dans un environnement dynamique densément peuplé.

Maitriser cette étape ouvre la porte aux véhicules autonomes en ville. Depuis les expérimentations de la Google Car en 2010, les projets de voiture autonome se sont multipliés chez les constructeurs automobiles dont certains proposent déjà des produits opérationnels (Tesla AutoPilot). Les législations évoluent également sur ce sujet anticipant ce qui pourrait être une véritable révolution dans le domaine du transport.

Les progrès de la localisation, autant pour la robotique que pour la réalité augmentée, ont été notamment permis par l'utilisation de capteurs GPS et de centrales inertielles (*IMU*) sur les appareils à localiser en plus de l'odométrie mécanique classique. Ces deux capteurs ont cependant des limites. Les gyromètres et accéléromètres des centrales inertielles accumulent des erreurs au cours du temps amplifiées par l'intégration (les mesure étant différentielles). Ce phénomène de dérive, très présent sur les centrales inertielles à bas coûts, rend un tel système inutilisable sans une autre mesure pour la corriger. Ainsi ces données sont généralement couplées à des données GPS. Or dans les milieux urbains, la présence d'immeubles de part et d'autre de la route peut obstruer le signal satellite, ce qui dégrade la précision de la localisation (effet de vallée). Même non dégradée, avec des erreurs qui varient de 5 à 10 mètres, la mesure GPS est trop imprécise pour des applications de réalité augmentée ou de robotique mobile performantes.

Vision par ordinateur

Une solution pour améliorer la précision de la localisation en milieux urbains est alors de se reposer sur la vision par ordinateur. En effet, un des problèmes fondamentaux de cette branche de l'informatique est de déduire d'une image la pose (en translation et en rotation) de la caméra qui en a fait l'acquisition.

L'avantage de cette mesure de localisation est qu'elle est issue d'un capteur très bon marché. D'ailleurs la plupart des appareils mobiles embarquent désormais une caméra. Si les techniques de vision par ordinateur sont souvent coûteuses en capacités de calculs, les nouveaux appareils mobiles se dotent d'unité de calcul dédiées images (puces Pixel Visual Core pour le smartphone de Google, calculateur Nvidia Drive PX pour la conduite autonome) qui permettent un traitement temps réel.

La vision est également le moyen de localisation principal d'une grande partie du règne animal dont l'Homme. En effet, c'est en se focalisant sur des repères visuels que l'Homme est capable de se repérer et de s'orienter. En milieu urbain cela se traduit souvent par une identification des bâtiments visibles dans la scène. Si les façades de bâtiments sont les repères visuels naturels en ville, ils sont également l'objet d'intérêt principal de la plupart des applications de réalité augmentée urbaine que l'on a citées.

Ce double rôle des bâtiments, qui sont par ailleurs des éléments immuables des scènes urbaines, nous a poussé à nous reposer sur eux pour résoudre le problème du calcul de pose de caméra en milieu urbain. Si l'on reconnait les bâtiments de l'image, on peut se positionner à la fois localement par rapports à eux mais aussi globalement dans la ville. Pour ne pas trop limiter le champ d'application, nous visons à résoudre ce problème à partir d'une seule image. Ainsi la méthode pourra être utilisée avec une caméra monoculaire standard aussi bien sur des images statiques que sur des vidéos.

Le problème de localisation que nous considérons est a distinguer du SLAM visuel qui reconstruit un modèle de l'environnement en ligne soit en utilisant un système multi-caméras, soit en utilisant la cohérence temporelle. Dans notre cadre, un modèle de l'environnement est supposé connu. Un positionnement n'ayant de sens que par rapport à un certain référentiel, pour pouvoir localiser la caméra dans un environnement, le modèle est attaché à un certain référentiel global. Localiser la caméra nécessite donc d'identifier quelle partie du modèle est observée dans l'image.

Il existe une grande diversité de modèles d'environnements urbains pour la localisation, le plus simple étant surement une carte avec les empreintes au sol des bâtiments. Ce modèle peut être enrichi en y ajoutant la hauteur des bâtiments pour établir un modèle 3D très grossier de la ville [8]. La grande régularité architecturale des villes fait que ces seules informations géométriques ne sont pas suffisamment discriminantes pour identifier la partie observée du modèle sans une première estimation via le GPS. Au contraire de ces modèles très simples, les modèles issus de méthodes de type *Structure From Motion*, sont très riches en information. Il sont constitués d'un nuage de points 3D qui représente la géométrie de la scène. Chaque point est aussi associé à un descripteur d'image local (classiquement SIFT) qui confère au modèle de l'information photométrique. A l'échelle d'une ville, ces modèles sont très lourds en mémoire (plusieurs Go pour Dubrovnik [71]) et le très grand nombre de points (plusieurs millions) agit directement sur la complexité des algorithmes de localisation les utilisant. En plus de cette distinction des modèles par leur complexité, ils présentent des niveaux de précision différents. Ainsi les modèles basés sur les empreintes au sol proviennent du cadastre qui a une fiabilité moindre vis-à-vis des images qu'un modèle SFM directement issu d'elles.

Nous proposons ne nous appuyer sur un modèle qui se concentre sur les bâtiments et qui apparait comme un compromis entre les modèles purement géométriques et les modèles de type *Structure From Motion*. Notre modèle de ville est alors constitué d'un ensemble de façades rectangulaires attachées à un référentiel global qui essentialise la géométrie. Chaque façade est par ailleurs décrite par un descripteur photométrique compact et une décomposition sémantique structurée. Ce modèle présente l'avantage d'être léger ce qui autorise son passage à l'échelle d'une ville tout en gardant de l'information photométrique et géométrique fine pour la précision. Ainsi il est utilisable autant pour des applications qui nécessitent un positionnement local (pour intégrer un objet virtuel à un bâtiment particulier) que pour des applications où un positionnement global est requis (aide à la navigation par exemple). Enfin il est possible de le construire à partir des données disponibles sur Google Street View [3] et iTowns [3].

L'objectif de la thèse a été de développer une méthode pour calculer la pose de la caméra vis-

3. www.ign.fr/institut/innovation/stereopolis

à-vis de ce modèle de ville à partir d'une seule image en vue piétonne. On impose également une contrainte d'efficacité compatible avec le temps réel en vue d'une application de la méthode à la réalité augmentée. L'idée principale est d'introduire de l'information haut-niveau et notamment de la sémantique aux différentes étapes d'une approche mixte *bottom-up*/*top-down* qui décompose le problème de la façon suivante.

Dans un premier temps, nous combinons la capacité de généralisation d'un réseau de neurones convolutionnels à la précision d'un modèle bayésien pour trouver les points de fuite principaux de l'image. Ceci fait l'objet du chapitre 2, dans lequel on montre qu'un telle approche permet de résoudre des cas très bruités avec très peu de segments convergents vers les points de fuite dans l'image. Les trois points de fuite détectés, dits de Manhattan, donnent une estimation de la rotation et permettent de rectifier l'image courante.

Dans ces images rectifiées les façades de bâtiments apparaissent rectangulaires. La deuxième partie consiste alors à détecter ces façades et les identifier par rapport à aux façades de référence de notre modèle de ville. Des caractéristiques spécifiques aux bâtiments sont résumées en ensemble d'indices qui exploitent une segmentation sémantique de l'image. Passé une étape sélection rapide de candidats sur ces indices, les façades sont classifiée et reconnues en utilisant des méthodes d'apprentissage. Cette méthode efficace, décrite en détails au chapitre 3, offre des performances accrues en terme de taux de détection et de reconnaissance par rapport à l'état de l'art.

Cette détection des façades de l'image permet d'estimer grossièrement la translation de la caméra. Celle-ci est raffinée dans une ultime étape qui tire parti de la segmentation sémantique pour recaler précisément la façade de référence dans un cadre bayésien. Cette étape est le sujet du chapitre 4. Les résultats sont évaluées sur trois jeux de données différents qui illustrent la robustesse de l'approche aux changements de points de vue, aux changements d'illumination, aux occultations et aux répétitions.

Chapitre 1

Etat de l'art

Le problème de localisation par l'image tel que nous venons de le formuler peut en fait se décomposer en deux sous-problèmes. Le premier problème consiste dans un premier temps à reconnaitre les façades de référence qui sont visibles dans l'image. L'identification de ces façades nous renseigne déjà sur une localisation sommaire dans la ville. Le second problème renvoie au calcul de pose de caméra, à proprement parler, par rapports à ces façades identifiées. Ce positionnement local couplé aux données géométriques géoréférencées associées aux façades de références du modèle permet alors un positionnement global précis.

On présente d'abord les solutions généralistes de la littérature pour ces deux sous-problèmes. Si la géométrie particulière des villes peut être exploitée notamment dans le calcul de pose (scène planes par morceaux), des difficultés propres aux milieux urbains (perspective prononcée, répétitions) peuvent en revanche faire échouer ces approches. Des méthodes spécifiques ont alors été développées pour exploiter les régularités de ces environnements plutôt que d'en subir les effets négatifs.

1.1 Solutions généralistes

1.1.1 Reconnaissance de lieux

Le premier problème d'identification de façade renvoie au problème appelé « Reconnaissance de lieux » (*Place Recognition*) dans la littérature. Il s'agit, à partir d'une nouvelle image d'un lieu, d'identifier le lieu en question parmi une base de connaissances. C'est un problème important en robotique mobile car identifier un lieu qui a déjà été parcouru par le robot est très utile pour corriger la dérive de l'odométrie (on parle de fermeture de boucle). Dans leur étude de 2016, Lowry et al. [76] distinguent la description de l'image du processus de reconnaissance par rapport à un modèle des lieux. Celui-ci prend le plus souvent la forme d'une base d'images étiquetées avec les différents lieux que l'on souhaite reconnaitre. Par ailleurs, l'étude [76] scinde les descriptions d'image pour la reconnaissance en deux catégories suivant qu'elles utilisent des descripteurs locaux ou directement une représentation globale.

Les descripteurs locaux comme SIFT [75], SURF [15] ou ORB [97] ont été construits pour être robustes aux changements de points de vue (invariants par similitude) ce qui présente un

avantage certain pour la reconnaissance. S'il n'est pas envisageable en terme de complexité de reconnaitre un lieu par mise en correspondance avec toutes les images de la base, Sivic et al. [103] déduisent une représentation globale compacte de l'image en utilisant ces descripteurs locaux. Cette représentation, dite en sac de mots et inspirée de la recherche de documents, décompose une image selon un vocabulaire de mots visuels. En apprenant hors ligne un partionnement de l'espace des descripteurs locaux (un vocabulaire), on peut quantifier les descripteurs locaux d'une image en un histogramme de mots visuels. Le partionnement, originellement fait par *K-Means*, a été par la suite amélioré en utilisant un mélange de Gaussiennes et une représentation par un vecteur de Fisher [90] ou par VLAD [5].

D'autres descripteurs sont déduits directement de l'image globale comme GIST [89]. Ces descriptions globales ont connu un regain d'intérêt avec l'utilisation de réseaux de neurones convolutionnels (CNN). Ainsi dans [84], les auteurs montrent que ces descripteurs sont plus robustes aux changements d'illumination ou de conditions climatiques que les approches locales. [105] propose une description hybride locale/globale en calculant des descripteurs CNN sur des zones d'intérêts de l'image issues d'une méthode de proposition d'objets [122] (Fig. 1.1). Cette approche qui cherche à associer le meilleur des deux mondes (la robustesse aux changements de points de vue et aux changements d'illumination) est assez proche dans le principe de celle que l'on propose dans cette thèse au chapitre 3.

FIGURE 1.1 – Exemple de reconnaissance de lieux par description hybride locale/globale [105]. Les régions d'intérêts proposées par EdgesBoxes [122] et mises en correspondance par descripteur CNN.

Concernant le processus qui vise à décider si un lieu a été reconnu ou non, plusieurs méthodes visent à lever les ambiguïtés de la description visuelle. Si ce processus prend généralement la forme d'une recherche d'image (*image retrieval*) au plus proche voisin dans la base suivant les descripteurs, des améliorations ont été apportées pour gérer les répétitions de motifs qui biaisent fortement les descripteurs en sac de mots. Ainsi dans [99], le vocabulaire est adapté de sorte à se concentrer sur les mots visuels discriminants pour leur modèle de lieux en utilisant un arbre de vocabulaire. La distribution des occurrences de mots visuels est également apprise dans FAB-MAP [27] cette fois via un modèle génératif. Torii et al. [109] détectent eux directement les motifs

répétitifs dans l'image pour adapter leurs poids dans l'histogramme des mots visuels.

L'amélioration du processus de décision permet également de corriger un autre défaut de la description par sac de mots qui est la disparition de l'information géométrique. Ainsi dans [67], les auteurs réintroduisent cette information en comparant les descriptions via une pyramide spatiale (*Spatial Pyramid Kernels*). Les relations géométriques entre co-occurrences de mots visuels sont aussi explicitement intégrées dans [86] ce qui améliore le pouvoir discriminant de cette description.

Néanmoins des difficultés de reconnaissance persistent dans les environnements urbains. En effet, la régularité des règles architecturales fait qu'une description par sac de mots reste trop peu discriminante. Les façades sont systématiquement composées des mêmes mots visuels en proportions très voisines l'une de l'autre et avec peu voire aucune information géométrique les liant. Ceci est renforcé par les nombreuses répétitions de mots visuels au sein d'une même image (*Perceptual Aliasing*). Cette grande similarité visuelle entre les façades est une source importante de confusions et d'échecs de reconnaissance en milieu urbain. Les changements d'apparences importants d'un même lieu est une autre source de difficulté de ces environnements. Par exemple les éclairages artificiels créent des changements de luminosité qui peuvent perturber la reconnaissance d'un lieu acquis sous d'autres conditions dans la base de connaissance. Par ailleurs les effets de la perspective sont très prononcés à cause des dimensions relatives des piétons, des bâtiments et des rues. Suivant l'orientation de la caméra, la vue de la scène est donc extrêmement différente ce qui affecte fortement les descriptions d'image globales mais aussi les descripteurs locaux qui peuvent alors être hors de leur domaine d'invariance.

Dans notre modèle, les images sont peu denses et sans aucune redondance contrairement à de nombreuses méthodes de reconnaissance qui reposent sur des bases de lieux acquises continûment par un robot mobile. Aussi, dans notre cadre, les changements d'apparences entre une nouvelle image et les images du modèle peuvent être plus marqués. De plus les façades ne représentent qu'une partie de l'image courante contrairement aux références de la base qui se réduisent aux façades. Cette asymétrie du problème renvoie alors plus à une problématique de détection d'objets que l'on discute dans le chapitre 3.

1.1.2 Calcul de pose pour les scènes planes

Selon notre décomposition du problème de positionnement global en milieu urbain, une fois la façade du bâtiment de l'image courante identifiée parmi les références du modèle, le second sous-problème vise à trouver la rotation R et la translation T de la caméra dans le repère de cette façade. Pour ce problème de calcul de pose locale il est nécessaire de connaitre les paramètres intrinsèques de la caméra qui constituent la matrice de calibration K. Comme notre modèle est composé de façades planes, le problème se réduit à un problème de recalage paramétré par l'homographie $H = K\,(R|T)\,K^{-1}$ qui transforme la façade de référence identifiée I_{ref} en la façade du bâtiment visible dans l'image courante I dite aussi image cible [80]. Dans la suite on ne distinguera les méthodes de recalage des méthodes du suivi (*tracking*) que par le fait que ces dernières sont des méthodes de recalage qui nécessitent une initialisation.

On peut alors distinguer 3 approches différentes pour calculer cette homographie. Les ap-

proches denses appelés parfois *Template-Matching* cherchent à maximiser une mesure de similarité entre les images I et $I_{ref} \circ H$. Les approches *features-based* déduisent l'homographie par mise correspondance de points d'intérêt dans les deux images. Enfin il existe des méthodes qui estiment directement l'homographie à partir d'un modèle appris par régression.

Dans la première catégorie du recalage par méthodes denses, la mesure de similarité choisie est très importante. La première mesure utilisée a été la somme des différences de pixels au carrés entre les deux images (norme L_2) [77]. L'optimisation se fait très rapidement par descente de gradient par l'algorithme de Gauss-Newton. Si la transformation était au début cantonnée à une simple translation 2D, les modèles géométriques ont ensuite été enrichis pour couvrir les transformations affines [48] et les homographies [10]. L'efficacité en temps de calcul de la minimisation a également été améliorée dans [18] par une approximation au second-ordre sans calculer le Hessien. La mesure de similarité par norme L_2 reste sensible aux changements d'illumination et aux occultations. Dans [58], l'image de référence est décomposée en une pyramide de sous-images recalées indépendamment vis-à-vis de l'image cible selon la norme L_2. La solution globale du recalage est cherchée récursivement dans l'espace des paramètres telle qu'elle maximise le nombre de sous-recalage. Si la décomposition permet de traiter efficacement les occultations, elle peut être sensible aux répétitions très fréquentes sur les façades.

Kim et al. [61] utilisent un M-Estimator pour une mesure de similarité plus robuste. L'information mutuelle entre les images est également une mesure de similarité moins sensible aux changements d'illumination et aux occultations [113] et utilisée depuis longtemps pour le recalage multimodal en imagerie médicale [91]. Si ces mesures augmentent significativement la complexité de l'optimisation, des progrès ont été apportés depuis qui permettent une résolution efficace [29] (Fig. 1.2). Néanmoins toutes ces méthodes restent itératives et nécessitent une initialisation dont dépend fortement la convergence de l'algorithme vers la solution globale.

FIGURE 1.2 – Suivi par recalage d'image en utilisant l'information mutuelle comme mesure de similarité entre images [29]. L'homographie entre la référence (en bas) et l'image courante (en haut) est calculée à chaque pas de temps.

Dans un cas de recalage en translation seule, la solution globale peut rapidement être trouvée par décalage de phase dans le domaine fréquentiel. Cette méthode peut être généralisée à des similarités [92] et à des homographies [123]. Cependant les zones de l'image courante qui ne correspondent pas à l'image de référence peuvent perturber la transformée de Fourier et faire

ainsi échouer la méthode. Cela arrive régulièrement sur les images urbaines où un bâtiment peut être observé sous des échelles très différentes.

La seconde catégorie d'approches de recalage concerne les méthodes *features-based*. Il s'agit d'extraire des points d'intérêt similaires dans les deux images et de les mettre en correspondance. Les points SIFT [75] sont définis comme des maxima de l'espace d'échelle des différences de gaussiennes de l'image. Chaque point, associé à une échelle et une orientation caractéristique, est alors rendu invariant aux similarités. Les points sont mis en correspondance selon leur plus proche voisin vis-à-vis de leur descripteur calculé comme un histogramme de gradient orientés local. Couplé à la méthode d'estimation RANSAC cela permet de calculer l'homographie H entre les deux images [49], en étant robuste aux occultations et aux changements d'échelle d'observation importants.

Malgré ces atouts, le temps de calcul des descripteurs SIFT rend la méthode lente. La détection et la description de points d'intérêts robuste a été accélérée avec SURF [15] en utilisant des réponses d'ondelettes de Haar calculées efficacement par images intégrales. ORB combine la rapidité d'extraction de FAST [96] avec la rapidité de calcul des descripteurs BRIEF [20] pour offrir des performances similaires à SIFT en un temps très réduit. Si FAST utilisait déjà en partie de l'apprentissage automatique, l'émergence des réseaux de neurones convolutionels a conduit à LIFT [118], une procédure calquée sur SIFT mais apprise de bout-en-bout. Ceci lui confère une plus grande robustesse aux changements de points de vue extrêmes et améliore la mise en correspondance (Fig. 1.3).

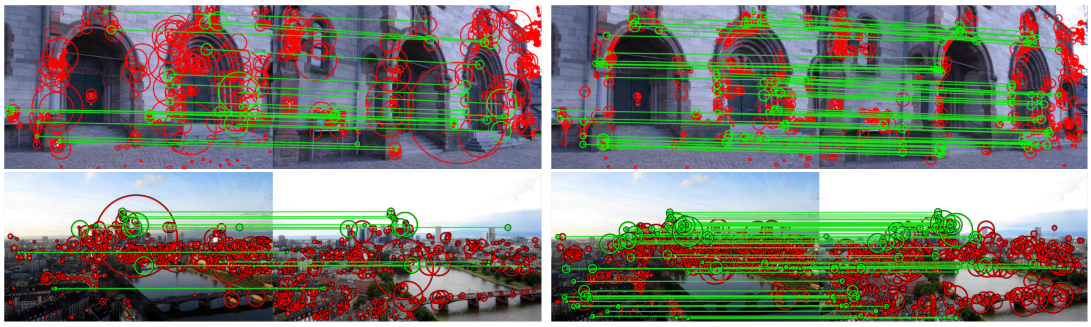

FIGURE 1.3 – Mise en correspondance de points d'intêret SIFT (à gauche) et LIFT (à droite) après filtrage par RANSAC [118].

Les performances des réseaux de neurones convolutionnels profonds ont ouvert récemment une nouvelle catégorie de méthodes de recalage qui vise directement à renvoyer une estimation de la transformation en sortie de réseau [32]. Plutôt que de concaténer les deux images en entrée de réseau, Rocco et al. [94] combinent les résultats d'un premier étage de description en une carte de corrélations à partir de laquelle un second réseau apprend à régresser la transformation. Ces approches par régression sont directement reliés aux travaux de Kendall et al. [60] qui infèrent la pose en rotation et en translation à partir d'une seule image, le modèle de scène étant en un sens inclue dans le réseau de neurones. Cependant la robustesse intrinsèque de ces réseaux aux translations et aux petites déformations ainsi que la taille fixe des entrées limite la précision des transformation estimées.

Ces nouvelles approches semblent néanmoins fournir une estimation grossière correcte même dans des cas de changements de points de vue extrême pour lesquels la mise en correspondance de points SIFT échoue. Ces cas ne sont pas rares en environnements urbains où les effets de perspectives sont très marqués en raison des tailles et distances relatives des bâtiments et des piétons. Ces approches par apprentissage automatiques sont également plus robustes aux changements d'illumination qui affectent autant les méthodes denses que les méthodes par points d'interêt.

La présence de motifs répétés très fréquents le long des façades (fenêtre, balcons, décorations, ...) crée aussi fréquemment des minima locaux pour ces deux catégories d'approches. Ces problèmes sont renforcés par le fait que les façades sont souvent des zones peu texturées (quelques fenêtres sur un crépi uniforme). Cela a pour conséquence l' extraction d'un faible nombre de points d'intérêts qui perturbe les votes de consensus pour les approches de type RANSAC. Cela crée également, pour les approches denses, de nombreux minima locaux dès lors que des contours se retrouvent alignés. Enfin les occultations sont courantes au niveau du rez-de-chaussée à cause du mobilier urbain, des voitures et des piétons. La distance typique des observateurs par rapport à la taille des bâtiments fait également que les façades sont souvent partielles sur les images.

1.2 Approches spécifiques aux milieux urbains

Pour résoudre ces problèmes caractéristiques des milieu urbains pour la reconnaissance et le calcul de pose, des méthodes ont cherché à exploiter les régularités particulières de ces environnements. On peut distinguer deux grands types d'approches, celles basées images (*bottom-up*) qui cherchent à détecter les façades précisément dans l'image courante et celles basées modèles (*top-down*) qui visent à recaler le modèle de la scène urbaine sur l'image.

1.2.1 Détection et reconnaissance de façades (*bottom-up*)

Plusieurs méthodes ont été proposées dans le passé pour détecter les structures rectangulaires (en fait des rectangles déformés par la transformation perspective) dans les environnements de type Manhattan. Dans [63], les segments de ligne sont automatiquement détectés et recoupés pour générer des hypothèses des structures rectangulaires en accord avec les points de fuite. Pour chaque hypothèse, l'image d'entrée est orthorectifiée et un histogramme de gradient (HOG) est calculé à l'intérieur du rectangle déformé. Les hypothèses dont le descripteur HOG associé contient plus de deux directions horizontales et verticales dominantes sont rejetées. Cette méthode coûteuse génère de nombreuses hypothèses superflues. Pour améliorer la qualité et réduire le nombre d'hypothèses, Micusik et al. [83] formulent la détection de structures rectangulaires sur un graphe des segments de l'image. Le voisinage est restreint par l'utilisation d'une triangulation de Delaunay contrainte pour connecter les segments. Le problème est alors exprimé comme la résolution du maximum de probabilité *a posteriori* d'un champ aléatoire de Markov. Toutes ces méthodes permettent de détecter des structures rectangulaires qui apparaissant sur les façades, comme des fenêtres ou des rangées de fenêtres, mais pas, en général, des façades entières.

Dans [73], les auteurs cherchent à caractériser les façades par leur régularité. L'indice de Gini est utilisé pour définir une mesure de régularité basé sur le caractère épars des distributions de

contours dans les façades. La détection de façades est traitée comme un problème de maximisation de cette régularité sur une région, qui est résolu en utilisant une méthode d'expansion de région gloutonne à partir d'une grille sous-échantillonnée. La Programmation Quadratique Intégrée est ensuite utilisée pour sélectionner un sous-ensemble de façades qui ont un score de régularité maximum et une couverture de façade, avec un chevauchement minimum. Cependant la méthode souffre d'une grande sensibilité à l'initialisation de la grille originale. L'hypothèse de régularité, si elle est vérifiée pour les centres d'affaires en vue aérienne, l'est beaucoup moins pour des vues piétonnes en vue perspectives d'environnements urbains architecturalement plus complexes.

Liu et al. [72] combinent d'une certaine façon les deux approches précédentes. Dans un premier temps des segments sont détectés dans l'image. Ils sont ensuite filtrés pour ne garder que ceux participant aux deux points de fuites principaux. Aux lignes supports de ces segments horizontaux et verticaux sont associés des descripteurs photométriques basés sur une décomposition fréquentielle. Les dissimilarités entre ces lignes délimitent l'image en régions homogènes qui sont ensuite fusionnées par adjacence et similarité de texture via un descripteur par co-occurences. Si cette méthode distingue bien la façade du fond lorsque celle-ci occupe une large portion de l'image, elle est mise en défaut lorsque celle-ci est plus petite. De plus l'étape de fusion a tendance à regrouper les façades adjacentes d'une scène au sein de la même détection (Fig. 1.4, en bas).

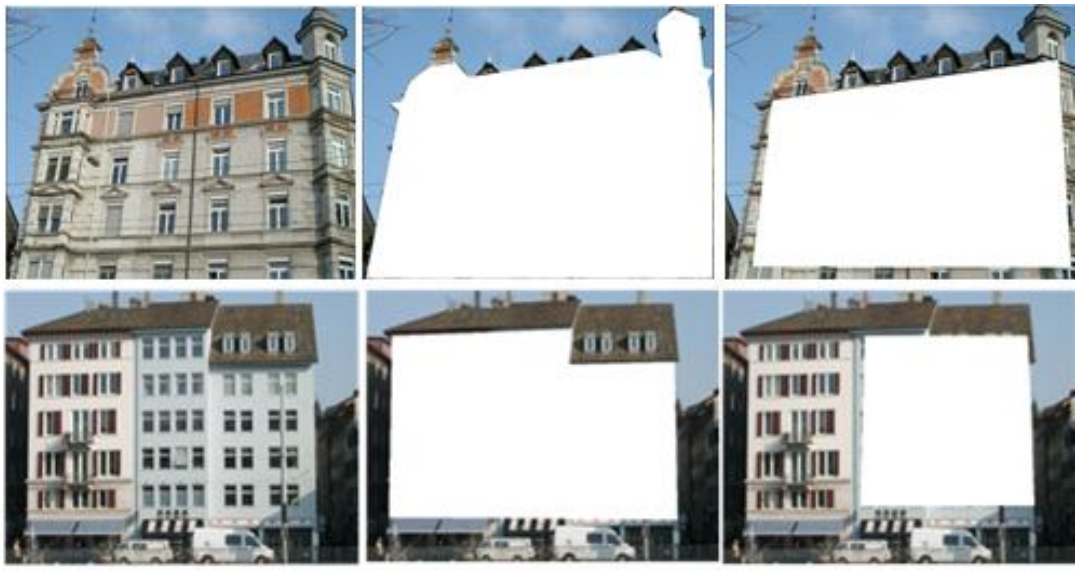

FIGURE 1.4 – Exemples de résultats de détection de façade de l'algorithme [72] (à droite). A milieu les images montrent le détourage manuel de la vérité-terrain par les auteurs.

Burochin et al. [19] cherchent à identifier des façades sans ouvertures (fenêtres ou portes) dans des images aériennes où la résolution des façades est très dégradée. Si l'objectif et le contexte sont différents de notre problème, le descripteur de façade qui y est développé est tout à fait pertinent pour la détection de façades. Celui-ci combine des informations statistiques globales sur la façade déclinés en différents critères (uniformité de la radiométrie, structure, répétitions) et les détections locales de portes et fenêtres. Celles-ci sont déduites d'une formulation en minimisation

d'énergie résolue par *reversible jump Markov Chain Monte Carlo* et recuit simulé. La lenteur de cette étape exclut cependant un tel descripteur de son utilisation pour la localisation visuelle urbaine.

1.2.2 Calcul de pose par recalage de modèle (*top-down*)

Si les approches *bottom-up* de détection de façades peuvent être utilisées aussi bien pour la reconnaissance que pour le calcul de pose, les approches *top-down* s'appliquent essentiellement au calcul de pose et nécessitent une initialisation qui vient le plus souvent du couple de capteurs GPS/IMU. Le problème est alors formulé comme la maximisation d'un critère de similarité entre les bâtiments du modèle projetés dans l'image et l'image courante. Dans [93], cette mesure fait intervenir une mise en correspondance des éléments de contours de l'image (*edgels*) avec ceux du modèle projeté. Ces données images sont fusionnés aux données inertielles par un filtre de Kalman. L'initialisation doit cependant être très proche de la solution du recalage car les alignements de segments créent facilement des minima locaux. [59] propose de se reposer sur les silhouettes de bâtiments plutôt que sur des contours internes à la façades. Le recalage initial est alors raffiné en mettant en correspondance la silhouette extraite des contours dans l'image avec la silhouette de la projection du modèle en utilisant des descripteurs *shape-context*. Le suivi est alors exécuté de manière très similaire à [93]. Si le modèle n'a plus besoin d'être texturé, il doit être géométriquement très détaillé pour être suffisamment discriminant. En pratique un tel modèle est très rarement disponible à l'échelle d'une ville. De plus si cette approche réduit les minima locaux, elle s'appuie toujours sur des contours dont les alignements non désirés en occasionnent encore trop.

Pour surmonter ces difficultés, des méthodes proposent d'utiliser une segmentation de l'image qui associe à chaque pixel une étiquette décrivant son appartenance à une classe sémantique de la scène (premier plan/arrière plan, classe d'objet). Ainsi dans [8], le GPS et l'IMU fournissent une pose grossière qui est raffinée par la suite. Les bâtiments sont modélisés par leur empreinte au sol ainsi que leur hauteur. Ce modèle 2.5D très simple est recalé en rotation en utilisant les points de fuites. Des hypothèses de translation sont générées par mise en correspondance des arêtes verticales du modèle avec les contours verticaux dans l'image. Ces hypothèses sont évaluées en comparant la projection du modèle dans l'image selon celles-ci à une segmentation sémantique de l'image qui distingue façade/non-façade. Si cette approche est intéressante, elle reste peux précise car elle n'est pas capable de distinguer deux façades adjacentes dont les bords ne serait pas directement visibles. Seule l'enveloppe de la façade est utilisée alors que des éléments architecturaux plus structurels pourraient justement résoudre ces ambiguïtés. C'est notamment ce que font Chu et al. [24] qui utilisent, en plus des contours et des étiquettes sémantiques de façade (basé sur SegNet [9]), une détection de portes et fenêtres par R-CNN [44] dans leur fonction de coût de recalage.

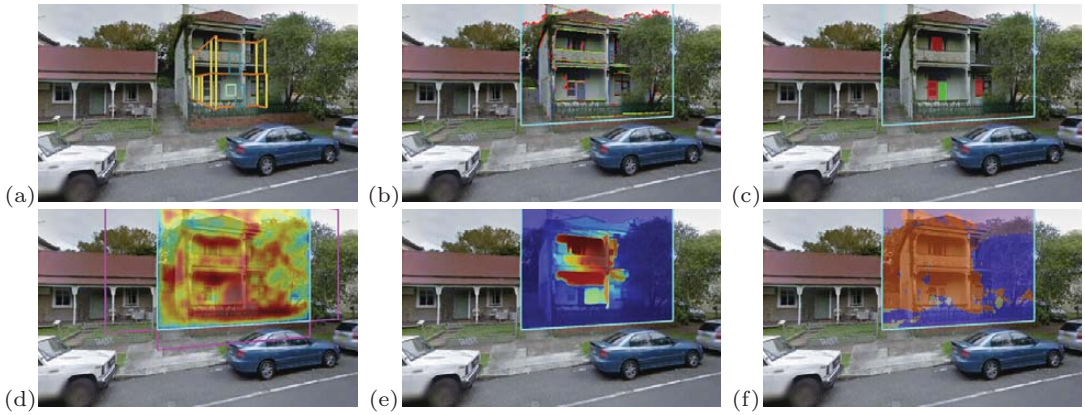

FIGURE 1.5 – Illustration des différents composantes de l'énergie à minimiser dans HouseCraft. (a) Recalage initial (GPS/IMU) (b) Contours verticaux et horizontaux (c) Détections de porte et fenêtre (d) Segmentation de premier plan (e) Carte de saillance (f) Segmentation sémantique

Cependant la complexité de l'inférence qui repose sur un espace de recherche discrétisé et l'utilisation de vues multiples sont des freins à une application temps réel pour la localisation en milieu urbain. Les approches points peuvent également bénéficier de la segmentation sémantique. Dans [62], l'auteur propose de filtrer les correspondances de points entre images à partir de leur contexte sémantique local. Si cela réduit un peu les ambiguïtés, dans les images urbaines avec des classes très fréquentes comme les fenêtres cela peut n'avoir qu'un faible impact. La robustesse des régions sémantiques aux changements de conditions d'illumination et de climat peut aussi être exploitée. Dans [108], un modèle SFM est augmenté de courbes 3D qui représentent les démarcations de régions sémantiques dans les images. Le recalage repose alors sur la mise en correspondance de ces points et ces courbes avec leur projection dans l'image. Si cette approche démontre la robustesse des CNNs à de forts changements dans les conditions d'acquisition, la stabilité de telles courbes n'est pas garantie dans le cas des façades (contours de fenêtres, de portes).

Parallèlement à nos travaux, la méthode de Arth et al. [8] a été améliorée en utilisant ces réseaux de neurones convolutionels pour générer une segmentation intermédiaire qui distingue les limites verticales entre façades en plus des limites horizontales, de la façade elle-même et de l'arrière-plan [7]. Deux réseaux supplémentaires ont été appris pour optimiser itérativement la pose à partir de cette segmentation et des segmentations issues du modèle projeté. Une alternative qui apporte une meilleure précision est d'utiliser une carte de normale inférée par un autre CNN [6]. Celle-ci combinés aux limites de façades de la segmentation permet de générer des hypothèses de pose dans un schéma RANSAC à partir de correspondances de 3 coins de façade ou de correspondances de 2 coins et des normales à la façade. Si cette méthode relaxe la contrainte de proximité de la pose initiale, elle reste nécessaire, la seule géométrie n'étant pas suffisante pour une localisation globale à l'échelle de la ville.

1.3 Outils mathématiques

1.3.1 Réseau de neurones convolutionnels

Inspirés des travaux sur le cortex visuel humain de Hubel et Wiesel [54], les réseaux de neurones convolutionnels ont été introduits dans les années 1980 [40]. Ils consistent en une succession de couches de neurones convolutionels, de fonctions d'activation non-linéaires, de couches de mutualisation (*pooling*) et de couches de normalisations.

Leur architecture en couches de dimensions toujours réduites cherche à traduire la hiérarchisation des images naturelles : des contours et des couleurs sont combinés en formes qui elles mêmes sont combinées en motifs plus complexes pour aboutir à un objet. La convolution permet d'extraire ces différents éléments et de les localiser quand la non-linéarité et le *pooling* renforcent l'invariance aux petites transformations comme l'a montré Mallat dans ses travaux voisins sur le *Scattering* d'ondelettes [78] [79].

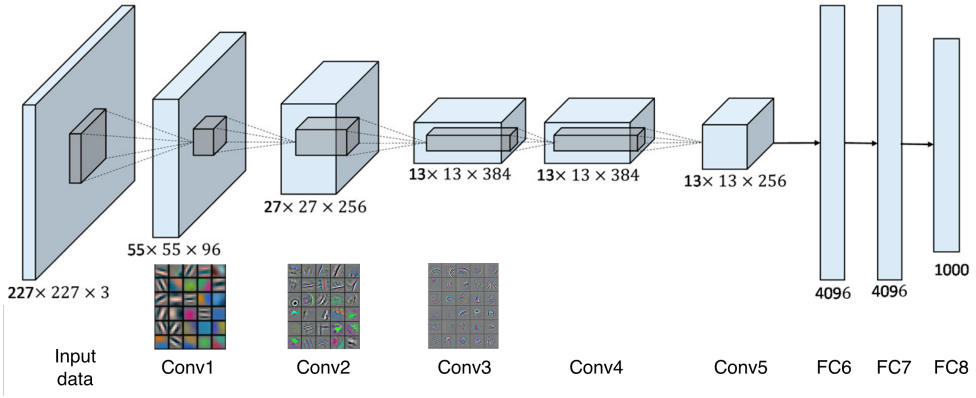

FIGURE 1.6 – Architecture du réseau de neurones convolutionnels AlexNet. La hierarchisation du réseau est illustrée par quelques filtres des trois premières couches visualisés par déconvolution.

La dernière couche du réseau donne des sorties qui sont typiquement de deux types possibles. Soit le réseau a été construit pour résoudre un problème de classification auquel cas la couche de sortie est un vecteur dont la taille est le nombre de classes. Soit le réseau a été construit pour résoudre un problème de régression auquel cas la couche de sortie est simplement le vecteur des paramètres à régresser.

Pour mener à bien ces tâches, le réseau est entrainé de manière supervisée. Cet entrainement est formulé comme un problème d'optimisation où les paramètres sont les poids des filtres de convolution. Pour tous les couples image/résultat de la base d'entrainement, il s'agit de trouver les poids qui minimisent une certaine fonction de coût qui traduit l'adéquation entre les valeurs de sortie de réseau et le résultat attendu pour une image d'entrée. Les fonctions de coût logistique associées à une fonction non-linéaire *SoftMax* et les fonctions d'entropie croisée sont généralement utilisées pour la classification. Pour les problèmes de régression la fonction de coût utilisant la norme euclidienne L_2 est préférée.

Tous les types de couches étant différentiables, ces réseaux sont entrainés par descente de gradient stochastique en utilisant la rétropropagation du gradient au travers des couches [68].

Pour renforcer l'indépendance des filtres pendant l'apprentissage, la technique du *Dropout* qui désactive aléatoirement des neurones est souvent utilisée.

Si ces systèmes avaient déjà démontré leur forces pour la reconnaissance de caractères dans les années 1990 [68], ils ont ensuite perdu en visibilité dans le domaine de la vision par ordinateur jusqu'à l'introduction d'AlexNet en 2012 dont les résultats sur ImageNet [98] ont révolutionné le domaine [65]. L'accès à de grande capacités de calcul par le développement des architectures d'unités graphiques, la disponibilité de grandes quantités de données d'entrainement et des améliorations sur l'apprentissage sont à l'origine du succès de ces nouveaux réseaux de neurones convolutionels profonds, selon Yann Lecun l'un des pères de cette approche.

Les approches d'apprentissage profond par réseaux de neurones convolutionnels (CNN) ont démontré leurs performances pour la segmentation sémantique dans des *benchmarks* CityScape [1] ou COCO [2]. Les travaux de Farabet et al. [35] ont ouvert la voie en proposant une approche où les descripteurs locaux sont appris par un CNN multi-résolution. L'image est d'abord décomposée via une pyramide Laplacienne. Sur chacun des étages de la pyramide est appliqué un CNN. Les dernières cartes de caractéristiques (*features maps*) des CNN des différents étages sont concaténées après interpolation et un perceptron multicouche est utilisé comme classifieur. Finalement le résultat de la segmentation est régularisé avec un *Conditional Random Field* basé sur les superpixels.

Les réseaux dits *Fully Convolutional* [74] abrégés FCN sont basés sur une architecture encodeur-décodeur et ne contiennent pas de couches complètement connectés comme dans les premiers modèles de réseaux pour la classification. Plusieurs raffinements ont été apportés à cette architecture pour gérer la multi-résolution. On peut citer notamment U-Net [95] qui concatene les couches encodeur-décodeur de même échelle ou DeepLab [22] qui utilise des convolutions à trou pour produire efficacement des *features maps* à différentes échelles. Le réseau SegNet [9] lui, conformément à la décomposition en « What » et « Where » théorisé par Zhao et Lecun [121], conserve l'information de localisation perdue lors des étapes de *pooling* dans l'encodeur et la réutilise pour les *unpooling* du décodeur.

Aujourd'hui, ces approches sont prometteuses ou font déjà état de l'art, non seulement sur des problématiques de haut-niveau comme la segmentation sémantique, la classification d'image (challenge sur ImageNet [3]), la détection d'objets (benchmarks sur Pascal VOC [4]), le calcul de pose [60], le calcul de cartes de normale ou de profondeur [12] [33] mais aussi des problématiques bas-niveau utilisant notamment des réseaux générateurs adversaires (GAN) [46] comme la super-résolution [69], le débruitage [111], ou l'*inpainting* [56], ...

1.3.2 Algorithme d'Espérance-Maximisation

L'algorithme Espérance-Maximisation (*Expectation-Maximization*, souvent abrégé EM) [30] est un algorithme itératif d'estimation de paramètres dans un cadre probabiliste de maximum de vraisemblance (ou de Maximum *A Posteriori*).

1. https ://www.cityscapes-dataset.com/benchmarks
2. http ://cocodataset.org
3. http ://image-net.org/challenges/LSVRC/2017/results
4. http ://host.robots.ox.ac.uk/pascal/VOC/index.html

Il est particulièrement utile lorsque la maximisation directe de la vraisemblance est impossible analytiquement mais que l'introduction de variables latentes non observables rend l'estimation des paramètres simple.

Soit $X = (x_1, \ldots, x_n)$ un ensemble des données d'observation indépendantes et identiquement distribuées suivant une loi $p(X|\Theta)$ paramétrée par le vecteur d'état Θ. On cherche à estimer le paramètre Θ maximisant la log-vraisemblance donnée par :

$$\ln p(X|\Theta) = \ln \prod_{i=1}^{n} p(x_i|\Theta) = \sum_{i=1}^{n} \ln p(x_i|\Theta) \tag{1.1}$$

On complète alors les données par les variables latentes $Z = (z_1, \ldots, z_n)$ inconnues. En notant $p(z_i|x_i, \Theta)$ la probabilité de z_i sachant x_i et le paramètre Θ, on définit la log-vraisemblance complétée $\ln p(X, Z|\Theta)$:

$$\ln p(X|\Theta) = \ln p(X, Z|\Theta) - \ln p(X|Z, \Theta) \tag{1.2}$$

L'algorithme EM est une procédure itérative qui cherche à construire une suite de paramètres $\Theta^{(t)}$ qui converge vers un maximum local de la log-vraisemblance $\ln p(X|\Theta)$.

Comme Z n'est pas observable, on estime la vraisemblance des données complètes $p(X, Z|\Theta)$ par son espérance conditionnellement aux données observées X et au paramètre courant $\Theta^{(t)}$. $\ln p(X|\Theta)$ ne dépendant pas de Z, on peut écrire :

$$\begin{aligned} \ln p(X|\Theta) &= \mathbb{E}_{Z|X,\Theta^{(t)}} \ln p(X, Z|\Theta) - \mathbb{E}_{Z|X,\Theta^{(t)}} \ln p(X|Z, \Theta) \\ &= Q\left(\Theta|\Theta^{(t)}\right) - H\left(\Theta|\Theta^{(t)}\right) \end{aligned} \tag{1.3}$$

On peut alors prouver en utilisant l'inégalité de Jensen [30] que :

$$\ln p(X|\Theta) - \ln p(X|\Theta^{(t)}) \geq Q\left(\Theta|\Theta^{(t)}\right) \tag{1.4}$$

Or l'introduction de Z a en principe été faite pour que $Q\left(\Theta|\Theta^{(t)}\right)$ soit maximisable facilement. L'algorithme peut alors se résumer en deux étapes itérées jusqu'à convergence :

— **E-Step :** $Q\left(\Theta|\Theta^{(t)}\right) = \mathbb{E}_{Z|X,\Theta^{(t)}} \ln p(X, Z|\Theta)$

— **M-Step :** $\Theta^{(t+1)} = \text{argmax}_{\Theta}\, Q\left(\Theta|\Theta^{(t)}\right)$

Dans le cadre d'une loi de mélange de distributions $f(X|\theta_j)$ paramétré par $\Theta = (\pi_j, \theta_j)_{1 \leq j \leq k}$:

$$p(X|\Theta) = \sum_{j=1}^{k} \pi_j f(X|\theta_j) \tag{1.5}$$

on introduit les variables latentes $z_{i,j}$ qui valent 1 quand l'observation i est associée à la loi de paramètre θ_j et 0 sinon. On peut alors réécrire la log-vraisemblance complétée :

$$\ln p(X, Z|\Theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{i,j} \left(\ln f(x_i|\theta_j) + \ln \pi_j\right) \tag{1.6}$$

En utilisant la règle de Bayes, on peut réduire les deux étapes de l'algorithme à :

— **E-Step :** calculer $p(z_{i,j}|x_i, \Theta^{(t)}) = \frac{\pi_j f(x_i|\theta_j^{(t)})}{\sum_{j'} \pi_{j'} f(x_i|\theta_{j'}^{(t)})}$

— **M-Step :** $\Theta^{(t+1)} = \text{argmax}_\Theta \sum_{i=1}^{n} \sum_{j=1}^{k} p(z_{i,j}|x_i, \Theta^{(t)}) \left(\ln f(x_i|\theta_j) + \ln \pi_j\right)$

Cet algorithme a été très largement utilisé en vision par ordinateur notamment pour le recalage de nuages de points [51] [34], le suivi de formes [38] ou la segmentation [25].

Chapitre 2

Estimation des points de fuite de Manhattan

Les scènes urbaines répondent à une géométrie particulière dictée par les règles d'architecture et d'aménagement urbain. Ainsi la géométrie des bâtiments est le plus souvent assimilable à un parallélépipède rectangle. De plus ces parallélépipèdes rectangles sont généralement alignés le long des routes et parallèles entre eux. Aussi on distingue le plus souvent 3 directions orthogonales prédominantes dans les scènes urbaines qui correspondent aux trois axes des parallélépipèdes. Ces directions sont appelées directions de Manhattan. Représentées par leur vecteur directeur unitaire orthogonaux entre-eux, elles forment une matrice orthonormale dite rotation de Manhattan. Trouver cette matrice permet ainsi de localiser la caméra en rotation par rapport à la scène.

Avec un modèle de caméra sténopé, ces vecteurs directeurs dans l'espace 3D de la scène sont projetés dans l'image en des points appelés points de fuite de Manhattan. Toute droite de la scène 3D parallèle à une direction de Manhattan est alors projetée dans l'image en une droite qui passe par le point de fuite de Manhattan correspondant. Les objets fabriqués par l'Homme possèdent souvent de telles lignes parallèles en grand nombre (bordures de fenêtres, de porte, de briques, signalisation sur la route, ...). Trouver les points où concourent le plus de droites de l'image est donc une manière d'estimer les points de fuite dont on peut déduire la rotation de Manhattan. Cette hypothèse est cependant parfois mise à mal en environnements urbains. En effet ceux-ci créent également beaucoup de fausses détections de points de fuite qui sont préjudiciables pour l'estimation de la rotation.

Nous proposons dans ce chapitre une méthode qui s'appuie sur un réseau de neurones convolutionnels pour estimer, à partir de l'image globale, les points de fuite de Manhattan ainsi qu'une segmentation de l'image relativement à eux. Une seconde étape de raffinement utilise alors ces informations et les segments de l'image pour estimer plus précisément ces points dans une formulation bayésienne efficace.

2.1 Travaux liés

Il existe une littérature importante autour de la détection de points de fuite dans les images. Pour l'essentiel, ces méthodes d'estimation reposent sur une première étape de détection de lignes. Si cela était traditionnellement fait par une transformée de Hough sur les contours de l'image, le détecteur de segments LSD basé sur une approche *a contrario* de l'alignement de contours [114] montre de bien meilleurs résultats. Celui-ci a contribué aux progrès récents de résolution de ce problème sur les bases de York Urban [31] ou Eurasian Cities [13]. Si toutes les méthodes cherchent ensuite les points de fuite comme des accumulations d'intersections de ces droites extraites, certaines supposent connues les paramètres intrinsèques de la caméra quand d'autres ne font pas cette hypothèse. Dans le contexte qui est le notre de calcul de pose de caméra, cette connaissance est nécessaire.

L'interêt de la connaissance des paramètres de calibration de la caméra est qu'elle permet de changer la représentation du problème. En effet, en géométrie projective, les intersections peuvent également avoir lieu à l'infini dans le cas de droites parallèles. Cela crée des cas singuliers lorsque l'on recherche des intersections dans l'image. Cependant avec les paramètres de calibration, on peut représenter une droite par un vecteur normal de la sphère unitaire (dite sphère Gaussienne) ce qui garantit une égalité de traitement de ces cas singuliers (Fig. 2.1).

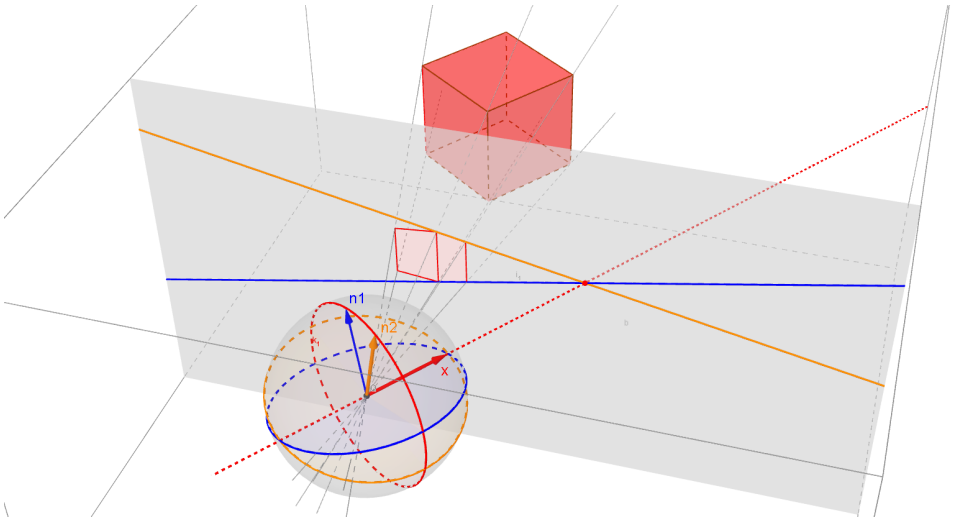

FIGURE 2.1 – Représentation des droites du parallélépipède modélisant un bâtiment sur la sphère Gaussienne. Cette représentation est faite par le vecteur normal n_i au plan qui passe par la droite projetée dans l'image et par le centre de la caméra.

Ainsi dans [4] les auteurs utilisent cette représentation sur la sphère Gaussienne dans un cadre bayésien. Un algorithme d'Espérance-Maximisation estime itérativement les coordonnées des points de fuite en même temps que la probabilité pour un segment de droite d'appartenir à un certain point de fuite. Bien que la convergence de l'algorithme EM soit souvent sensible à l'initialisation, une procédure très grossière utilisant la transformée de Hough est utilisée pour cette étape. Depuis plusieurs travaux se sont intéressés à améliorer cette initialisation.

Tardif et al. [106] génèrent des hypothèses de points de fuite en utilisant des paires de segments. L'ensemble de consensus est ensuite calculé par l'algorithme J-linkage ce qui réduit les minima locaux possibles. La même méthode est employée dans [116] avec une nouvelle mesure de consistance point/segment qui apporte une meilleure précision. Dans [70], le problème est résolu dans le domaine dual où des droites convergentes deviennent des alignements de points. Une détection robuste d'alignements de points exprimée dans un cadre *a contrario* permet de générer des points de fuite potentiels. Malheureusement, puisque pour toute paire de droites, celles-ci s'intersectent forcément (à l'infini si elles sont parallèles), le regroupement de lignes pour en déduire les points de fuite reste un problème difficile qui génère souvent de nombreuses fausses détections de points de fuite.

Pour résoudre ce problème, certains travaux ont cherché à introduire des contraintes géométriques sur les points de fuite lors de leur détection. L'orthogonalité entre les points de fuite de Manhattan est notamment utilisée en reformulant le problème de manière à chercher une rotation dont les trois vecteurs supports sont les directions de Manhattan. Une méthode de résolution de ce problème d'optimisation par moindres-carrés non-linéaires est proposée dans [85] qui utilise des bases de Gröbner pour résoudre le système polynomial associé. Si la résolution est efficace, elle reste sensible aux anomalies (*outliers*).

Pour une estimation plus robuste, Bazin et al. [16] utilisent un algorithme RANSAC en échantillonnant des triplets de lignes pour définir des rotations dont on évalue le score de consensus. Les mêmes auteurs ont également proposé une autre approche selon cette même formulation robuste du problème en tant que problème d'optimisation sur les rotations [17]. L'optimisation globale est faite par un algorithme *Branch-and-Bound* utilisant des notions de théorie des intervalles. Si la méthode est plus lente elle garantie une solution globale au problème.

2.2 Travaux précédents sur la recherche de points de fuite avec *A Priori*

Dans la lignée de ces derniers travaux cités, on peut ajouter un *a priori* sur la distribution géométrique de ceux-ci dans la plupart des applications urbaines en vue piétonne qui nous intéressent ici. Nous avons contribué à plusieurs méthodes qui utilisent un tel *a priori* pour régulariser le problème.

Ainsi dans une première approche [36] nous avons défini une distribution de points de fuite *a priori* (Fig. 2.2) qui repose sur les observations suivantes :

- le zénith est généralement proche de la direction verticale du repère caméra et l'angle de roulis faible
- les autres points de fuite lui sont globalement perpendiculaires
- parmi ceux-ci les deux points de fuite de Manhattan sont eux-mêmes globalement orthogonaux entre eux

Cet *a priori* est formulé dans un cadre bayésien en utilisant différentes distributions de Kent définies sur la sphère. En effet, ayant supposé connue la matrice de calibration de la caméra, travailler sur le domaine sphérique est plus commode pour imposer des contraintes sur la géométrie

projective. Une pseudo-vraisemblance basée sur l'accumulation d'intersections de segments peut être calculée directement sur la sphère. L'*a priori* sert alors à la fois dans une première étape de recherche de points de fuite candidats en biaisant l'échantillonnage d'un algorithme *Mean-Shift* sphérique. Il sert aussi à régulariser la pseudo-vraisemblance dans la formulation en Maximum *A Posteriori* de la sélection du triplet de Manhattan parmi les candidats.

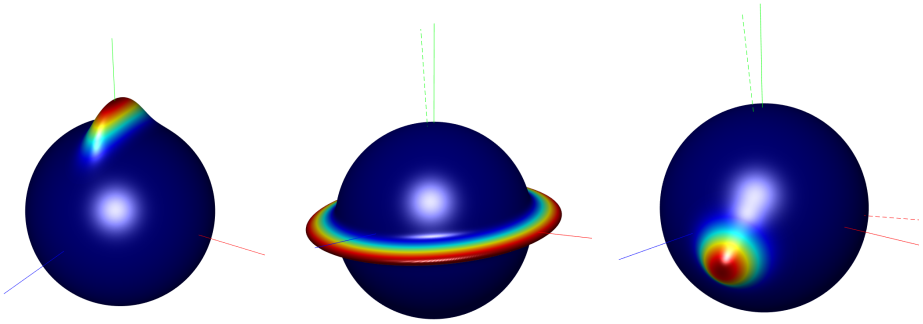

FIGURE 2.2 – Distributions de probabilité conditionnelles de la distribution *a priori* des points de fuite $P(X,Y,Z) = P(Z|X,Y)P(X|Y)P(Y)$. De la gauche vers la droite : $P(Y)$, $P(X|Y)$ et $P(Z|X,Y)$. L'axe x est en rouge, l'axe y en vert et l'axe z en bleu. Le zénith est en tirets vert.

Nous avons également participé à une seconde approche [101]. Pour celle-ci, l'*a priori* de proximité du zénith avec la verticale de la caméra a été conservé. Mais plutôt que de travailler sur la sphère, on s'affranchit de la matrice de calibration pour travailler directement dans l'image. Cela fait d'autant plus sens que ce simple *a priori* sur le zénith en vue piétonne entraine la localisation de la ligne d'horizon systématiquement dans les limites de l'image. De plus, cette ligne d'horizon apparait dans la plupart des images en environnement urbains comme une ligne matérielle faite d'une accumulation de segments et pas seulement comme une abstraction mathématique (Fig. 2.3). Cela s'explique par le fait que la plupart des objets d'une scène urbaine ont soit une hauteur adaptée à l'humain (voitures, portes, ...), soit sont placés à hauteur de regard (affiches, panneaux, ...). La méthode se décompose alors en 3 parties. Premièrement on cherche le zénith proche de la verticale et on corrige le roulis par une rotation de l'image. La ligne d'horizon est alors horizontale dans cette image corrigée. Sa hauteur dans l'image est estimée à partir des pics dans les histogrammes de segments horizontaux accumulés sur la direction verticale. On cherche alors les autres points de fuite comme étant des accumulations d'intersections avec la ligne d'horizon. On peut alors estimer la matrice de calibration.

2.3 Estimation et segmentation conjointes des directions de Manhattan

Comme on l'a vu précédemment, la plupart des méthodes d'estimation des points de fuite reposent sur une première étape d'extraction de segments de droites dans l'image. Or dans les environnements fabriqués par l'Homme, les lignes peuvent s'avérer être des indices peu fiables.

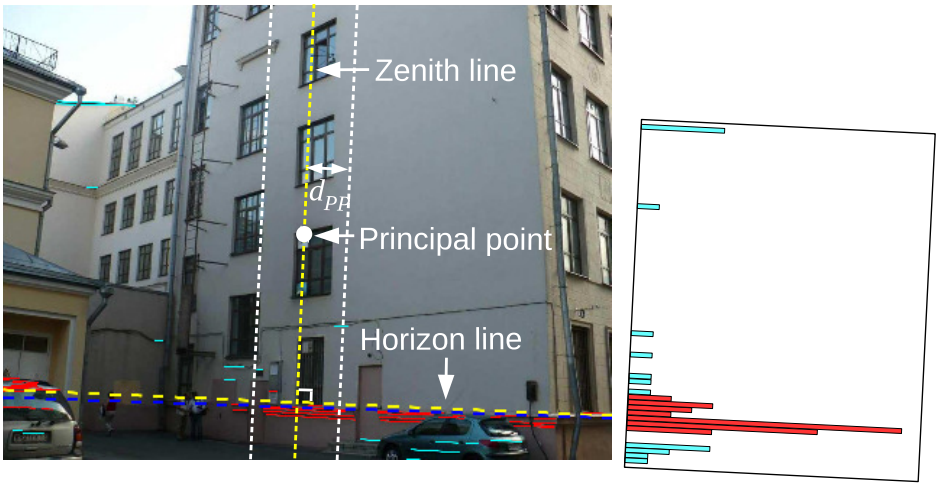

FIGURE 2.3 – La ligne d'horizon peut être détectée comme une accumulation de segments de droite dans l'image orthogonaux à la direction du zénith.

Si les alignements de motifs le long des façades sont bien visibles dans les images de faibles résolutions, il n'est pas forcément facile d'en extraire des segments de droites convergeant vers les points de fuites. En effet dans ces conditions et pour des façades soumises à de forts effets de perspectives, les segments de droites provenant des fenêtres sont par exemple rarement détectés. Cela s'explique par la difficulté à estimer l'orientation locale du gradient sur ces structures comprimées.

Les images peuvent manquer de segments de droites convergeant vers les points de fuite de Manhattan. Ce petit nombre d'*inliers* est une difficulté pour la détection de ces points de fuite. Ce problème est renforcé par l'abondance de segments de droites extraits qui n'ont aucune cohérence avec le repère de Manhattan, augmentant d'autant plus le taux d'anomalies (*outliers*). Ces anomalies proviennent le plus souvent de structures linéaires venant par exemple des voitures, du mobilier urbain, des lignes électriques ou de façades qui ne sont pas alignés avec les points de fuites de Manhattan.

Même dans des cas idéaux sans anomalies ni perspective trop prononcée, la densité de motifs répétés verticalement et horizontalement sur les façades crée facilement des points de convergence au centre de celles-ci qui ne sont pas des points de fuite. Combinés aux problèmes précédents, cette multiplication des minima locaux peu différents les uns des autres rend difficile l'estimation du repère de Manhattan à partir de segments de droites.

Au contraire nous proposons de nous baser, dans un premier temps, sur l'ensemble de l'image pour estimer le repère de Manhattan. C'est également le choix de Zhai et al. dans [119] qui profitent de l'inférence d'un réseau de neurones convolutionnels pour estimer le zénith et la ligne d'horizon. Notre approche va plus loin et utilise aussi un réseau de neurones convolutionnels pour régresser cette fois l'orientation du repère Manhattan tout en segmentant les plans de l'image selon ces orientations. On espère ainsi intégrer davantage d'information géométrique que seulement les segments de droites comme des répétitions linéaires de motifs plus complexes ou

des déformations de structures régulières (rectangles transformés en trapèzes par exemple).

2.3.1 Architecture du réseau

Le réseau de neurones convolutionnels est composé de deux parties qui sont directement liées aux deux problèmes que nous résolvons conjointement : l'estimation de l'orientation du repère de Manhattan et la segmentation de l'image selon ces orientations (Fig. 2.4).

La partie segmentation est constituée d'un réseau SegNet [9] modifié pour avoir 4 étiquettes sémantiques de sorties "X", "Y", "Z" et "anomalie". Cet étiquetage sémantique de l'image signifie que, pour un pixel de l'image, le point de la scène 3D dont il est projeté appartient à un plan orthogonal à la direction de l'étiquette. Les noms des directions sont choisis avec la convention des repères caméra dans le cadre projectif. Ainsi par exemple un point 3D projeté sur un pixel étiqueté "Y" appartient au sol dans la scène quand un point projeté sur un pixel étiqueté "Z" appartient à un plan est globalement orienté face à la caméra.

La partie estimation du réseau est composée de deux couches complètement connectées (*fully connected layers*) de taille 4096 qui sont directement rattachées à la couche de convolution Conv5 de la partie segmentation en entrée. La sortie de ce perceptron multi-couches pour la régression est constitué d'un couche de taille 4 qui représente les 4 composantes des quaternions qui définissent le repère de Manhattan dans le repère de la caméra.

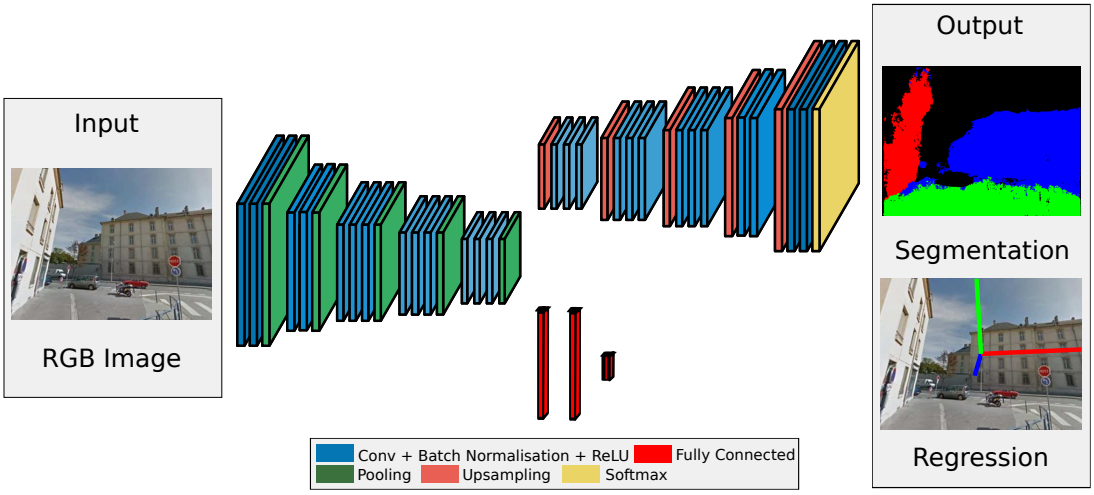

FIGURE 2.4 – Architecture du réseau SegNet modifié pour avoir deux sorties, l'une pour la régression de la rotation de Manhattan, l'autre pour la segmentation de l'image selon les directions de Manhattan.

La fonction de coût finale J est la somme pondérée (par le paramètre α) de la fonction de coût par entropie croisée (*cross-entropy*) pour la segmentation et de la norme euclidienne pour la régression :

$$J = -\frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=0}^{3} y_{i,j}^{(k)} \ln \frac{\exp x_{i,j}^{(k)}}{\sum_{j'} \exp x_{i,j'}^{(k)}} + \alpha \frac{1}{K}\sum_{k=1}^{K} \left\| p^{(k)} - q^{(k)} \right\|_2 \tag{2.1}$$

avec $x_{i,j}^{(k)}$, la valeur de sortie de réseau au pixel i pour l'étiquette j de l'échantillon k. $y_{i,j}^{(k)}$ vaut 1 si l'étiquette vaut j au pixel i pour ce même échantillon k et 0 sinon (les étiquettes sont codées par $j = 0$ pour une anomalie, $j = 1$ pour "X", $j = 2$ pour "Y" et $j = 3$ pour "Z"). De même $p^{(k)}$ représente le vecteur de sortie du réseau de régression et $q^{(k)}$ les 4 composantes du quaternion attendu de l'échantillon k.

2.3.2 Bases d'apprentissage

Nous entrainons le réseau sur deux jeux de données qui correspondent respectivement à des environnements de Manhattan extérieurs et intérieurs.

Pour les environnements extérieurs urbains il s'agit d'images de la ville de Nancy issues de Google Street View [1]. Nous avons d'abord extrait 3000 images panoramiques 360° uniformément sur les routes de Nancy à partir des coordonnées GPS des panoramas et des cartes du réseau routier d'OpenStreetMap [2]. En plus de ces panoramas images, il est possible d'extraire de Google Street View des cartes de normales panoramiques aux mêmes localisations qui sont établies à partir du modèle 3D de la ville (Fig. 2.5). Pour chacun de ces couples de panoramas équirectangulaires (image, carte de normales) nous échantillonnons 10 vues différentes (Fig. 2.6). L'échantillonnage est fait selon une paramétrisation des vues en angles de lacet, roulis et tangages. Le lacet est uniformément distribué entre 0 et 2π, tandis que le tangage suit une loi normale centrée dont l'écart-type est de $\sigma = 0.26$ (15°). Le roulis n'est pas considéré ici. Il en résulte une base de données de 30000 images piétonnes de diverses scènes urbaines et leur carte de normales correspondante (Fig. 2.7).

A partir de la carte de normales d'une vue, on en déduit le repère de Manhattan vérité-terrain en utilisant la méthode de Ghanem et al. [43] pour y extraire les 3 directions orthogonales prédominantes. Cette même méthode permet également de segmenter la carte de normales relativement aux trois directions de Manhattan. Cependant nous y rajoutons un masque sémantique qui ne garde que les étiquettes de façades et de route dans l'inférence de la segmentation sémantique par SegNet. On espère ainsi filtrer les éléments sémantiques vecteurs d'anomalies pour le raffinement du repère basé sur les lignes.

Pour les environnements intérieurs, la base est construite à partir de 30000 vues de bureaux, de salles de cours et de chambres du jeu de données de Stanford [3]. Chaque vue est également associée à une carte de normales directement disponible. Ces normales sont issues du maillage triangulaire régularisant le nuage de point venant d'un algorithme de *Structure From Motion*. Elles sont ainsi plus précises que celles de Google Street View qui dépendent de la précision de la cartographie des bâtiments. On utilise la même méthode pour extraire le repère de Manhattan et sa segmentation de normales correspondante. Aucun post-traitement par masquage sémantique n'y est appliqué cependant. En effet les murs étant peu texturés et le mobilier généralement aligné avec le cadre de la pièce, un filtrage sémantique limiterait trop l'extraction de lignes pour le raffinement.

1. http ://www.google.fr/maps
2. http ://openstreetmap.fr
3. http ://buildingparser.stanford.edu/dataset.html

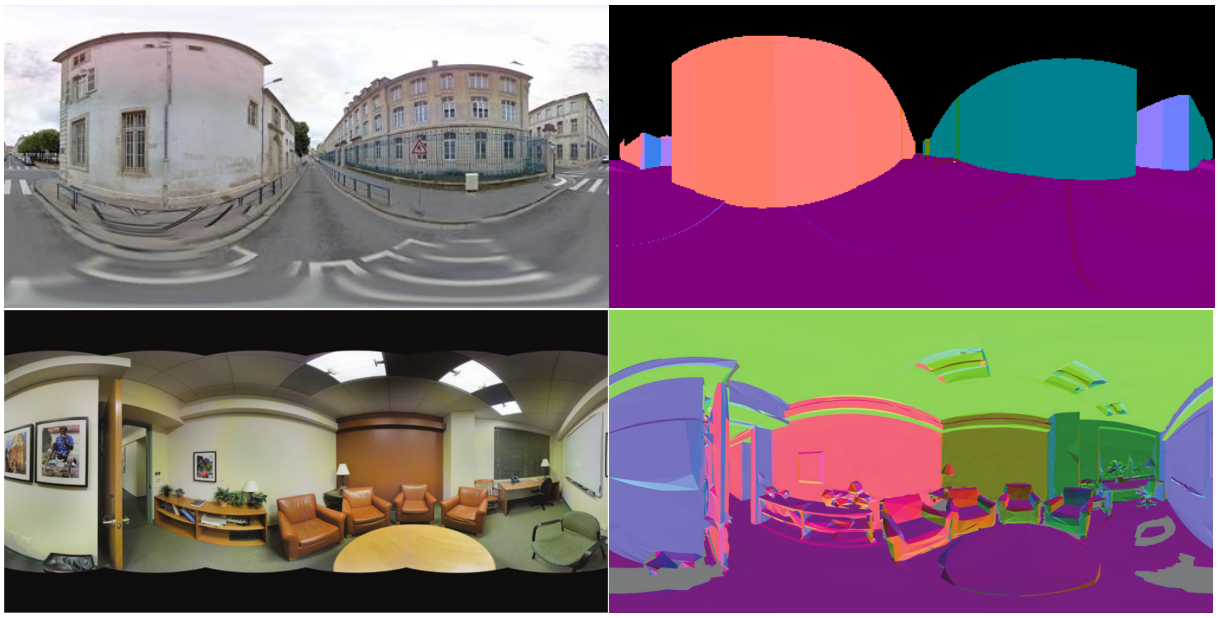

FIGURE 2.5 – Données d'apprentissage équirectangulaires brutes des bases Google Street View (en haut) et Stanford (en bas). A gauche l'image équirectangulaire et à droite la carte de normales correspondante.

FIGURE 2.6 – Différentes vues échantillonnées à partir des panoramas équirectangles (en haut Google Street View et en bas Stanford).

2.3.3 Entrainement du réseau

Nous utilisons une descente de gradient stochastique sur la fonction de coût J (Eq. 2.1) pour entrainer le réseau avec Caffe [57] avec des *batches* de taille $K = 10$. Cependant nous avons dû isoler les différentes parties du réseau pour assurer la convergence. La partie encodeur convolutif du réseau jusqu'à Conv5 correspond à une réduction de dimension que l'on peut espérer assez générique. Elle devrait alors être assez stable aux neutralisations de couches plus profondes. Dans

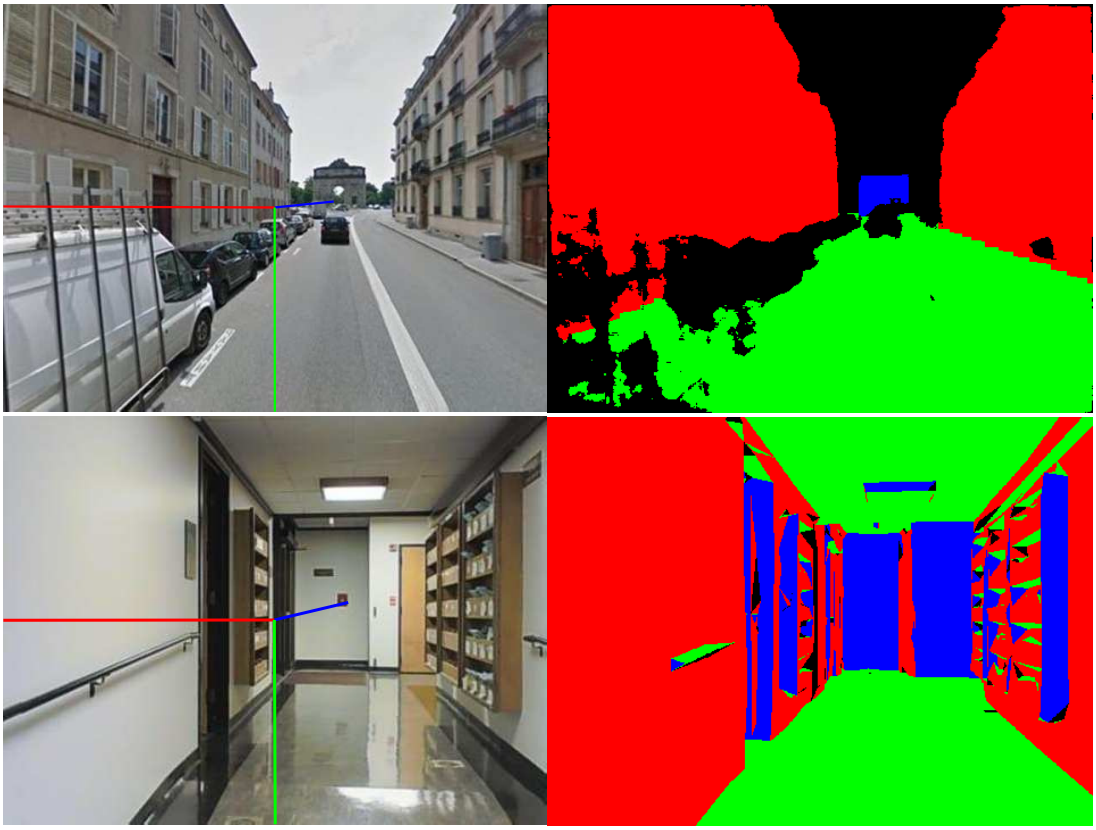

FIGURE 2.7 – Vérité terrain pour l'apprentissage déduite des panoramas équirectangulaires pour les bases Google Street View (en haut) et Stanford (en bas). A gauche l'image en projection perspective avec les points de fuite de Manhattan également projetés. A droite la segmentation de l'image en plans selon les directions de Manhattan.

un premier temps nous n'avons entrainé que le réseau SegNet seul (sans la partie régression donc) pour inférer uniquement la segmentation de directions de Manhattan (Fig. 2.8).

Ensuite nous avons supprimé l'étage de déconvolution du réseau à partir de la couche Conv5 pour y rajouter les deux couches complètement connectées de la régression. L'architecture de ce réseau est alors très similaire au réseau VGG [21]. Les poids du réseau sont initialisés avec les poids de l'apprentissage précédent pour les couches de convolutions et au hasard pour les deux nouvelles couches de régression. Ce nouveau réseau est alors entrainé pour la régression du repère de Manhattan exclusivement (Fig. 2.9).

Enfin, on rajoute les couches de déconvolutions à la couche Conv5 dont les poids sont initialisés aux poids de l'apprentissage de segmentation seule. Les autres poids (étage de convolution et de régression) sont initialisés aux poids de l'apprentissage de régression seule. L'étage de déconvolution est alors raffiné à cette étape par rapport aux nouvelles entrées de l'étage de convolution. Cette dernière étape permet en outre un apprentissage complet des deux tâches de bout-en-bout. Cet apprentissage conjoint améliore les résultats de ces deux problèmes non-indépendants tout en mutualisant les calculs.

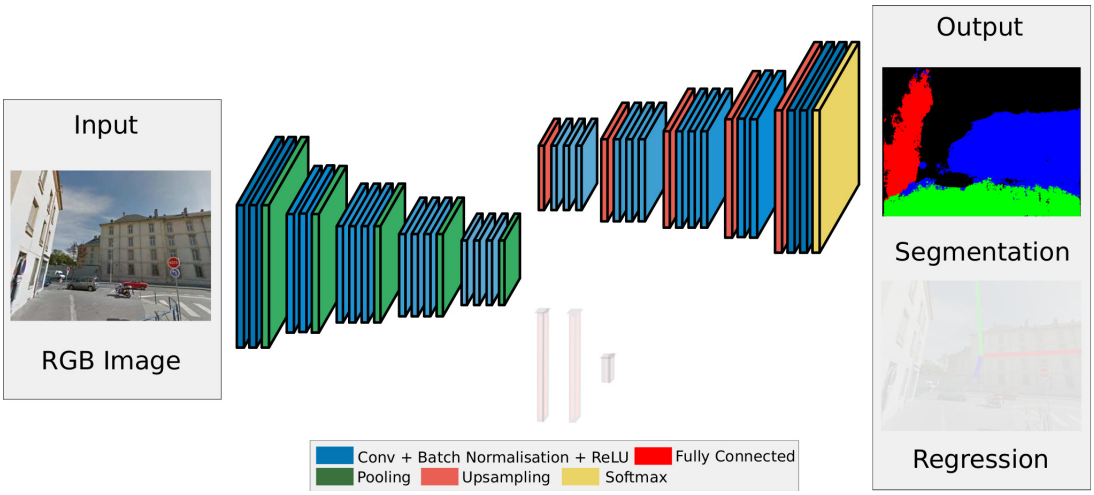

FIGURE 2.8 – La partie régression du réseau est neutralisée dans un premier temps.

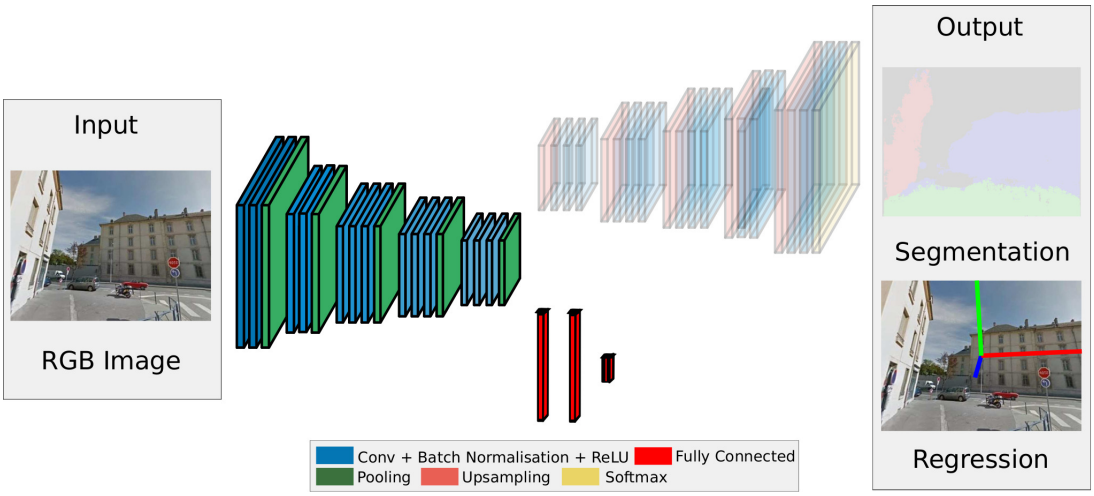

FIGURE 2.9 – Dans un second temps, c'est la partie segmentation qui est neutralisée pour entrainer les 2 couches complètement connectée de régression.

2.4 Raffinement du repère de Manhattan utilisant les segments de droites

A cause de la robustesse intrinsèque des réseaux de neurones convolutionnels aux petites déformations [79], on ne peut espérer une très grande précision de l'estimation du repère de Manhattan par le CNN. Nous pouvons néanmoins dépasser cette limite en raffinant le résultat de la régression en utilisant cette fois les segments de lignes comme dans les approches plus classiques. Nous proposons pour cela un modèle bayésien qui lie directement le repère de Manhattan aux couples de droites plutôt qu'aux droites elles-mêmes. La formulation de ce problème bénéficie de la segmentation et de l'initialisation du réseau précédent et se résout très rapidement par Espérance-Maximisation.

2.4.1 Classification *A Priori* des segments de droite

On note $L = \{l_i\}_{1 \leq i \leq M}$, l'ensemble des M segments de droite extraits de LSD [114]. On suppose connus les paramètres intrinsèques de la caméra au travers de la matrice de projection K :

$$K = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

avec la longueur focale f et le centre optique de la caméra (c_x, c_y). Les deux sorties du réseau fournissent une initialisation du repère de Manhattan $R_0 = (X_0|Y_0|Z_0)$ ainsi qu'une segmentation des plans de l'image selon ce repère D.

Chacune des droites supports des segments l_i définit avec le centre de la projection $(0, 0, 0)$ un plan qui intersecte la sphère unitaire (dite sphère Gaussienne [14]) en un grand-cercle. Les intersections entre ces grand-cercles correspondent aux intersections projectives de leurs droites associées dans le plan. Ainsi les points de fuite de l'image peuvent être déduits des intersections de grand-cercles (Fig. 2.1, le point de fuite x est déduit des grand-cercles de normales n_1 et n_2). Ceux-ci sont représentés par leur normale dans l'ensemble N :

$$N = \left\{ \frac{K^{-1}l_{i,1} \times K^{-1}l_{i,2}}{\|K^{-1}l_{i,1} \times K^{-1}l_{i,2}\|_2} | \, l_i \in L \right\} \tag{2.3}$$

avec $l_{i,1}$ et $l_{i,2}$, les extrémités du segment l_i dans le repère 2D du plan.

La plupart des approches d'estimation du repère de Manhattan cherchent d'abord les points de fuite soit par accumulation d'intersections de grand-cercles sur la sphère Gaussienne (Fig. 2.11, à gauche) soit par discrétisation de l'espace de recherche, soit par échantillonnage de modèles parmi les données (RANSAC). C'est seulement dans un second temps et de manière heuristique que sont extraits les points de fuite de Manhattan. Notre approche est différente. On génère d'abord toutes les intersections possibles sur la sphère (Fig. 2.11, au milieu) et on détermine le repère orthogonal qui maximise le nombre de ces intersections à partir de la solution initiale proposée par le réseau.

Cette formulation se heurte néanmoins à une complexité plus importante ($O(M^2)$ au lieu de $O(M)$) ainsi qu'à une multiplication des minima locaux. La classification *a priori* des segments de droites à partir de D répond à ce problème. En effet elle permet à la fois de limiter ces minima mais aussi de réduire la combinatoire en imposant des conditions sur les intersections possibles entre grand-cercles.

La direction de Manhattan qui a le plus de chance d'être bien estimée est le zénith $m_{y_0} = KY_0$. En effet, cette direction est partagée par la plupart des objets de la scène fabriqués par l'homme et elle est le plus souvent proche de la direction verticale de la caméra $Y_{cam} = (0, 1, 0)$. En utilisant la mesure de consistance entre une droite l_i et un point de fuite $c(l_i, m_{y_0})$ définie par Xu et al. [116], nous classifions un segment de droite comme « vertical » (notés "y") s'il est plus consistant avec le zénith initial m_{y_0} qu'avec les autres directions de Manhattan estimées initialement ($m_{x_0} = KX_0$ et $m_{z_0} = KZ_0$) et s'il recouvre majoritairement ($\geq 70\%$) des zones d'étiquettes "X" ou "Z" de la segmentation de Manhattan D. En effet les étiquettes donnant la

direction des normales, il est impossible d'avoir un segment « vertical » au sol d'étiquette "Y". Ils ne peuvent être que sur des plans "X" où "Z".

Pour classifier les autres segments de droite, nous exploitons encore les contraintes de la segmentation de Manhattan (Fig. 2.10). Les façades de la scène sont assimilables à des plans « verticaux » étiquetés soit "X" soit "Z". La structure très régulière des façades fait qu'une droite sur une façade est très probablement soit « verticale » ("y"), soit dans la direction complémentaire à "y" et à l'étiquette normale de la façade. Ainsi les segments de droite qui recouvrent majoritairement des zones de plans « verticaux » "X" et qui ne sont pas déjà classifiés "y" sont classifiés "z". De même les segments qui recouvrent des zones de plan « verticaux » "Z" et qui ne sont pas déjà classifiés "y" sont classifiés "x".

Les segments de droite qui recouvrent des zones de plans « horizontaux » d'étiquette "Y" ne peuvent être verticaux. C'est la seule condition que donne la segmentation de Manhattan à ces segments pour les classifier. Pour ne pas injecter un *a priori* trop fort en s'appuyant sur l'estimation initiale, nous choisissons de dupliquer ces segments et de les classer aussi bien "x" que "z". Comme ces segments correspondent essentiellement à des éléments au sol, qui est généralement assez uniforme, il ne devrait pas y avoir trop de segments dans ces zones.

Enfin les segments qui recouvrent majoritairement des zones d'anomalies dans la segmentation de Manhattan sont écartés.

On note $L^{(p)}$, $p \in \{x, y, z\}$, l'ensemble des segments l_i classifiés "p" et $N^{(p)}$ son ensemble associé sur les normales de grand-cercles. On génère alors toutes les intersections de grand-cercles dans chacun des sous-ensembles $N^{(p)}$ en prenant le produit vectoriel des normales de tous les couples de grand-cercles de même classe.

$$V^{(p)} = \left\{ \frac{n_i \times n_j}{\|n_i \times n_j\|_2} \,|\, n_i \in N^{(p)},\, n_j \in N^{(p)},\, i \neq j \right\} \tag{2.4}$$

L'ensemble $V = \left\{V^{(x)}, V^{(y)}, V^{(z)}\right\}$ est l'ensemble de données de la sphère Gaussienne sur lesquelles on cherche des accumulations orthogonales entre elles qui correspondent aux points de fuite de Manhattan (Fig. 2.11, à droite).

2.4.2 Formulation bayésienne

Nous formulons ce problème dans un cadre bayésien. La distribution des données observées $V = \{V_i\}_{1 \leq i \leq n}$ sur la sphère Gaussienne est modélisée par un mélange de distributions de Von Mises-Fisher $P(V_i|\mu, \kappa)$ (Eq. 2.6 et Fig. 2.12) et d'une distribution uniforme $P(V_i|o) = \frac{1}{4\pi}$ pour les anomalies :

$$\begin{aligned} P(V_i|\Theta) = \quad & \sum_{j=1}^{3} \pi_j P(V_i|\mu_j, \kappa_j) + \alpha P(V_i|o) \\ \text{t.q.} \quad & (\mu_1|\mu_2|\mu_3) \in SO(3) \end{aligned} \tag{2.5}$$

avec

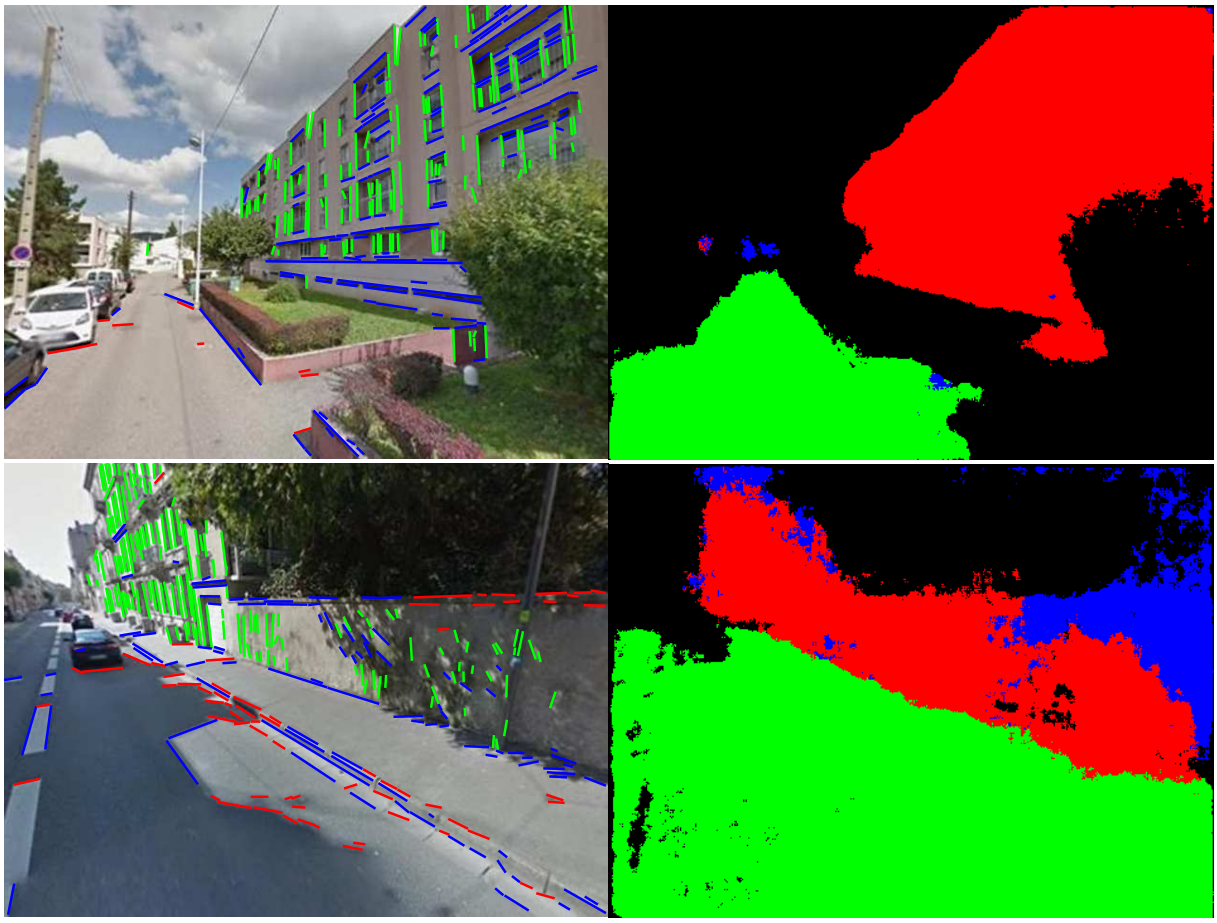

FIGURE 2.10 – Classification *a priori* des segments de droite de LSD (à gauche) à partir de la segmentation de l'image en directions de Manhattan (à droite).

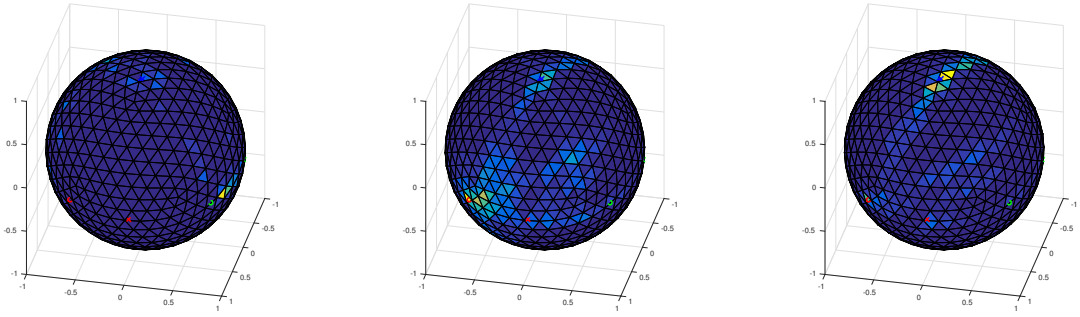

FIGURE 2.11 – Histogramme sphérique des normales des grand-cercles n_i (à gauche). Histogramme sphérique de toutes les intersections de grand-cercles $v_k = n_i \times n_j$ (au milieu). Histogrammes sphériques des intersections de grand-cercles filtrés par la classification *a priori* $v_k \in V$ (à droite). La vérité terrain de la rotation est représentée par des cercles et l'estimation initiale par des croix.

$$P(V_i|\mu,\kappa) = \frac{\kappa}{4\pi \sinh\kappa} \exp\left(\kappa \mu^T V_i\right) \tag{2.6}$$

$\{\pi_j\}_{1\leq j\leq 3}$ représentent les poids du mélange et α le taux d'anomalies (*outliers*). $\{(\mu_j,\kappa_j)\}_{1\leq j\leq 3}$ sont les paramètres des distributions de Von-Mises Fisher où les centres μ_j sont liés par une contrainte d'orthonormalité. On note alors $R = (\mu_1|\mu_2|\mu_3)$ la rotation qui correspond à la transformation du repère caméra $(e_1|e_2|e_3) = I_3$ en le repère de Manhattan.

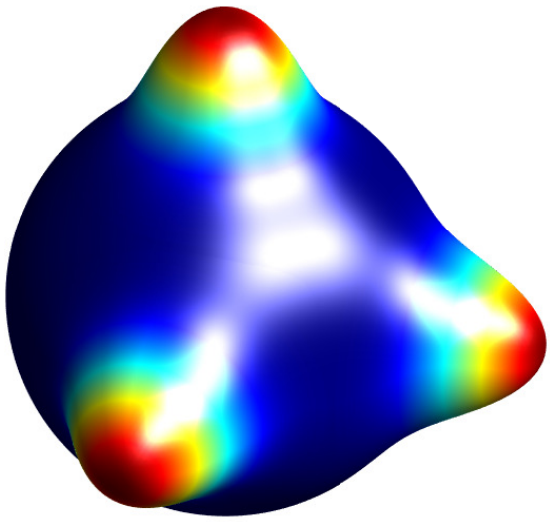

FIGURE 2.12 – Exemple de mélange de distributions de Von Mises-Fisher sur la sphère Gaussienne avec $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$, $\kappa_1 = \kappa_2 = \kappa_3 = 50$ et $R = I_3$

Le but est d'estimer le paramètre d'état $\Theta = (R, \alpha, \{\pi_j\}_{1\leq j\leq 3}, \{\kappa_j\}_{1\leq j\leq 3})$ qui contient la transformation géométrique R ainsi que tous les paramètres des distributions. En faisant l'hypothèse que les données observées sont indépendantes et en prenant le logarithme, la vraisemblance $P(V|\Theta)$ peut alors être maximisée pour trouver Θ :

$$\begin{aligned} \Theta^\star &= \underset{\Theta}{\operatorname{argmax}} \ln P(V|\Theta) \\ &= \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{n} \ln P(V_i|\Theta) \end{aligned} \tag{2.7}$$

2.4.3 Résolution par Espérance-Maximisation

Ce problème de Maximum de Vraisemblance peut être résolu dans un cadre d'Espérance-Maximisation. On définit les variables latentes $Z = \{z_{i,j} \in \{0,1\}, z_{i,o} \in \{0,1\}\}_{1 \leq i \leq n, 1 \leq j \leq 3}$ telles que $z_{i,j}$ assigne un point V_i à une distribution de Von mises-Fisher (Re_j, κ_j) et $z_{i,o}$ assigne V_i à la classe d'anomalie supplémentaire o. L'algorithme d'Espérance-Maximisation cherche à trouver la solution itérativement en alternant entre deux étapes (Fig. 2.13). L'étape E (*E-Step*) calcule l'espérance (par rapport à Z) de la log-vraisemblance complétée $Q(\Theta|\Theta^{(t)})$ conditionnellement à V et au paramètre courant $\Theta^{(t)}$. L'étape M (*M-Step*) cherche le paramètre d'état Θ qui maximise cette espérance :

$$\begin{aligned} Q\left(\Theta|\Theta^{(t)}\right) &= \mathbb{E}_{Z|V,\Theta^{(t)}} \ln P(V, Z|\Theta) \\ &= \sum_{Z} P(Z|V, \Theta^{(t)}) \ln P(V, Z|\Theta) \\ &= \sum_{i} \sum_{j} \beta_{i,j} \left(\ln \pi_j + \ln P(V_i|R, \kappa_j)\right) \\ &+ \sum_{i} \gamma_i \ln \frac{\alpha}{4\pi} \end{aligned} \tag{2.8}$$

avec $\beta_{i,j} = \mathbb{E}\left(z_{i,j}|V, \Theta^{(t)}\right)$ et $\gamma_i = \mathbb{E}\left(z_{i,o}|V, \Theta^{(t)}\right)$
Ainsi l'algorithme d'Espérance-Maximisation itère entre ces deux étapes, ce qui donne en détail :

- **E-Step :** calcule $\beta_{i,j}$ et γ_i
- **M-Step :** $\Theta^{(t+1)} = \text{argmax}_{\Theta}\, Q\left(\Theta|\Theta^{(t)}\right)$

L'étape **E-Step** peut être vue comme le calcul de la probabilité d'assignation de chaque point de donnée observé V_i à une distribution de Von mises-Fisher $P(V_i|Re_j, \kappa_j)$ connaissant les paramètres $\Theta^{(t)} = \left(R^{(t)}, \alpha^{(t)}, \{\pi_j\}^{(t)}_{1 \leq j \leq 3}, \{\kappa_j\}^{(t)}_{1 \leq j \leq 3}\right)$. En utilisant la règle de Bayes, on peut écrire :

$$\begin{aligned} \beta_{i,j} &= \mathbb{E}\left(z_{i,j}|V_i, \Theta^{(t)}\right) \\ &= \frac{\frac{\pi_j \kappa_j}{\left(1 - e^{-2\kappa_j}\right)} \exp\left(\kappa_j V_i^T R e_j - 1\right)}{\sum_{j'} \frac{\pi_{j'} \kappa_{j'}}{\left(1 - e^{-2\kappa_{j'}}\right)} \exp\left(\kappa_{j'} V_i^T R e_{j'} - 1\right) + \alpha/2} \end{aligned} \tag{2.9}$$

$$\begin{aligned} \gamma_i &= \mathbb{E}\left(z_{i,o}|V_i, \Theta^{(t)}\right) \\ &= \frac{\alpha/2}{\sum_{j'} \frac{\pi_{j'} \kappa_{j'}}{\left(1 - e^{-2\kappa_{j'}}\right)} \exp\left(\kappa_{j'} V_i^T R e_{j'} - 1\right) + \alpha/2} \end{aligned} \tag{2.10}$$

Dans l'étape **M-Step** nous visons à maximiser $Q\left(\Theta|\Theta^{(t)}\right)$ connaissant les assignations $\beta_{i,j}$

et γ_i. En développant les expressions des distributions de l'équation 2.6 et en ignorant les termes constants, Q peut être réécrit $\tilde{Q}$:

$$\tilde{Q} = \sum_{i,j} \beta_{i,j} \left(\ln \pi_j + \ln \kappa_j - \ln \left(1 - e^{-2\kappa_j} \right) + \kappa_j V_i^T R e_j \right) + \sum_i \gamma_i \ln \frac{\alpha}{4\pi} \tag{2.11}$$

La solution $\Theta^{(t+1)}$ de la maximisation de cette quantité $\tilde{Q}$ est calculée analytiquement. Elle correspond à une mise à jour des paramètres $\Theta^{(t)}$ selon les formules établies ci-après.

Mise à jour des poids du mélange π_j et α

En dérivant l'équation 2.11 à laquelle on a rajouté la contrainte de normalisation des poids on obtient :

$$\pi_j^{(t+1)} = \frac{\sum_i \beta_{i,j}}{n} \tag{2.12}$$

$$\alpha^{(t+1)} = \frac{\sum_i \gamma_i}{n} \tag{2.13}$$

Mise à jour des paramètres de concentration κ_j

La dérivation de l'équation 2.11 entraine la résolution des équations suivantes :

$$\frac{\sum_i \beta_{i,j}}{\kappa_j} + \frac{2 \sum_i \beta_{i,j}}{1 - e^{2\kappa_j}} + \sum_i \beta_{i,j} V_i^T R e_j = 0,\ 1 \leq j \leq 3 \tag{2.14}$$

Si ces équations ne peuvent être résolues sous forme analytique, plusieurs approximations des solutions ont été proposées dans la littérature. La solution introduite par Banerjee et al. [11] montre la meilleure qualité d'approximation :

$$\kappa_j^{(t+1)} = \frac{r_j \left(2 - r_j^2 \right)}{1 - r_j^2} \tag{2.15}$$

avec $r_j = \frac{\left\| \sum_i \beta_{i,j} V_i \right\|_2}{n}$

Mise à jour de la rotation R

Trouver R revient à résoudre le problème d'optimisation suivant :

$$\begin{aligned} R^{(t+1)} = \quad & \underset{R}{\operatorname{argmax}} \sum_{i,j} \beta_{i,j} \kappa_j V_i^T R e_j \\ \text{t.q.} \quad & R^T R = I_3 \end{aligned} \tag{2.16}$$

Or en posant les matrices concaténés de tailles $(3n, 3)$ $X = (B_1 V | B_2 V | B_3 V)^T$ ($B_j = diag(\beta_{i,j})$) et $Y = (\kappa_1 e_1 | \dots | \kappa_1 e_1 | \kappa_2 e_2 | \dots | \kappa_2 e_2 | \kappa_3 e_3 | \dots | \kappa_3 e_3)^T$, on peut remarquer que ce problème peut se réécrire avec le produit scalaire matriciel $\langle , \rangle$ issu de la norme de Frobenius :

$$\begin{aligned} R^{(t+1)} &= \underset{R}{\operatorname{argmax}} \langle X,\, RY \rangle \\ &= \underset{R}{\operatorname{argmax}} \left\langle R,\, XY^T \right\rangle \end{aligned} \tag{2.17}$$

Sous cette forme il s'agit d'une version du problème de Procruste Orthogonal Pondéré (*Weighted Orthogonal Procrustes Problem*) dont la solution est calculée à l'aide d'une décomposition en valeurs singulières (SVD) [100] :

$$R^{(t+1)} = U_1 U_2^T \text{ avec la SVD } XY^T = U_1 \Sigma U_2^T \tag{2.18}$$

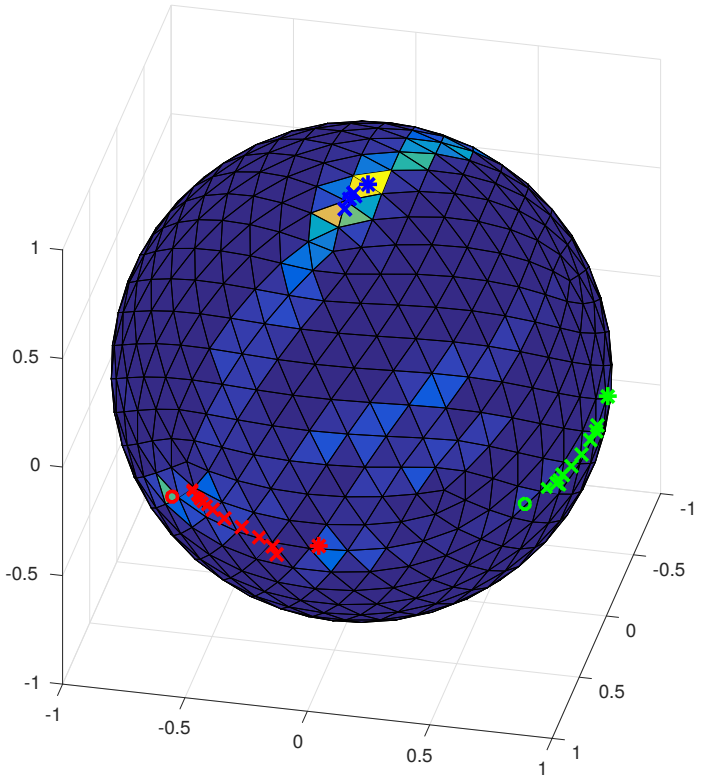

FIGURE 2.13 – Itérations de l'algorithme EM sur l'histogramme sphérique des intersections de grand-cercles filtrés par la classification *a priori*. La vérité terrain de la rotation est représentée par des cercles, l'initialisation par des étoiles et les différentes itérations par des croix.

2.5 Efficacité et applications

2.5.1 Détails d'implémentation et efficacité

L'algorithme EM nécessite une initialisation. Si la rotation est initialisée à partir de l'estimation du réseau, les autres paramètres ne profitent pas d'une estimation spécifique à l'image. Les poids du mélange sont initialisés sans préférence $\pi_1^{(t_0)} = \pi_2^{(t_0)} = \pi_3^{(t_0)} = \frac{1}{3}$. Les paramètres de concentration sont eux aussi initialisés sans distinction à des valeurs $\kappa_1^{(t_0)} = \kappa_2^{(t_0)} = \kappa_3^{(t_0)} = 20$ qui correspondent à des étendues larges sur la sphère. Enfin le taux d'anomalies $\alpha^{(t_0)} = 0.8$ correspond à la moyenne des taux d'anomalies parmi les segments de LSD sur la base d'apprentissage.

Grace aux formules analytiques de la *M-Step*, la complexité de l'algorithme EM pour une itération est en $O(n)$, avec n le nombre d'intersections de grand-cercles. Si la convergence de l'algorithme dépend de l'estimation initiale de la rotation par le CNN, elle se fait en moyenne en moins de 20 itérations sur la base d'apprentissage.

Si on considère toutes les intersections de segments on a $n = \frac{M(M-1)}{2}$, avec M le nombre de segments extraits par LSD (en moyenne $M \approx 450$). Le filtrage par les contraintes de la segmentation de Manhattan réduit en pratique ce nombre d'un ordre de grandeur avec une moyenne de seulement $n \approx 9765$ intersections.

Les contraintes de la segmentation de Manhattan sont établies par la consistance des segments de droite avec le zénith et par des tests de recouvrement de ces segments avec la segmentation. Ceux-ci sont effectués en appliquant l'algorithme de Bresenham sur les segments de droite pour en déduire les indices des pixels de la segmentation qu'ils recouvrent. Toute cette partie sur la classification *a priori* des segments et le filtrage des intersection qui en découle est codée en Matlab.

Les autres parties de la méthode sont codées en C. L'estimation initiale et la segmentation de Manhattan profitent en plus d'une accélaration GPU via le *framework* Caffee [57]. Malgré les améliorations de complexité que l'on vient de discuter, l'algorithme EM demeure la partie critique de la méthode en terme de temps de calcul. Celui-ci est également codé en C. On tient cependant à préciser que c'est le cas avec la plupart des méthodes de l'état de l'art avec lesquelles nous nous comparons [106][85][70]. Il serait par ailleurs facile de paralléliser la *E-Step* dont toutes les évaluations sont indépendantes. Un soin particulier a d'ailleurs été apporté à la formulation de ces évaluations pour éviter les erreurs numériques. Par exemple le sinh de la distribution de Von Mises-Fisher a dû être transformé en $\sinh(\kappa) = \frac{e^{\kappa}(1-e^{-2\kappa})}{2}$. Ainsi l'execution de l'ensemble de la méthode est très rapide avec une moyenne de 0.12 s par image sur une installation de bureau avec un processeur I7-3520M et une carte graphique Nvidia TITAN X.

2.5.2 Orthorectification des images

En plus de fournir une estimation de la rotation de la caméra, le repère de Manhattan permet de transformer l'image de sorte que les façades orthogonales à un point de fuite horizontal apparaissent comme vues de face (Fig. 2.14). Pour les deux points de fuite horizontaux "X" et "Z", on peut alors générer deux images rectifiées. Dans les conditions de Manhattan toutes les façades de l'image sont alors rectifiées sur l'une ou l'autre des deux images.

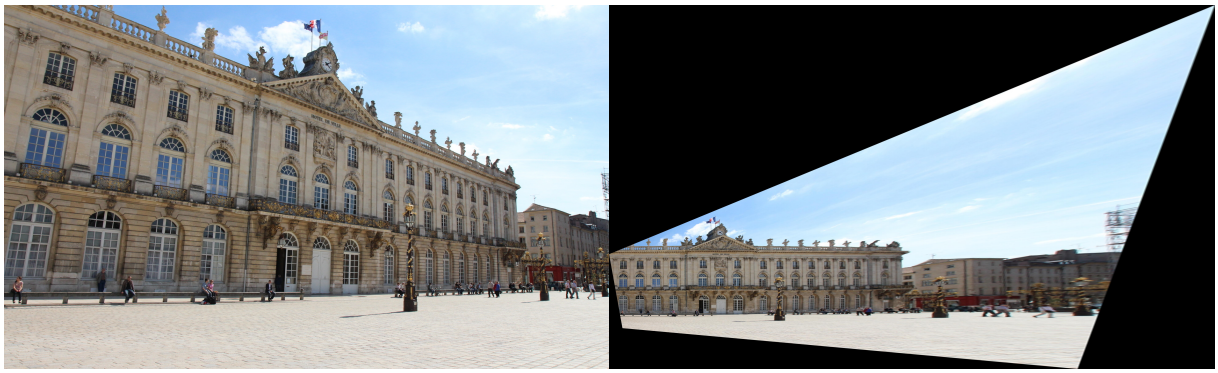

FIGURE 2.14 – Image de l'hôtel de ville de Nancy (à gauche) et l'image rectifiée de la façade de ce même bâtiment (à droite).

La transformation d'image H est une homographie que l'on peut calculer à partir de la matrice de calibration K :

$$H = KRK^{-1} \tag{2.19}$$

avec $R = (X|Y|Z)$ ou $R = (Z|Y|X)$.

Dans la suite de la thèse on utilise ces images transformées pour estimer la translation. Cela nous permet de nous affranchir des effets de perspective qui peuvent être importants dans les environnements urbains. En effet dans ces images, les façades apparaissent alors rectangulaires ce qui facilite leur détection.

En pratique on réduit l'image transformée à une sous-image dans laquelle les déformations ne sont pas trop importantes (un pixel n'est pas transformé en plus que 8×8 pixels). En effet dans les cas de très forte perspective, l'homographie H peut transformer l'image de sorte que sa taille ait des valeurs déraisonnablement grandes. Cette taille est directement liée à la complexité des algorithmes qui l'utilise et peut même être problématique pour le stockage en mémoire. De plus les zones très étirées n'apportent finalement que peu d'information car elle sont issues d'une interpolation très sévère (plusieurs centaines de pixels pour un pixel de l'image originale). Ce recadrage entraine un décalage en translation qui peut se traduire dans les équations comme un changement du centre optique de la matrice de calibration K'. On a alors $H = K'RK^{-1}$.

2.6 Résultats et limites

2.6.1 Jeux de données de tests

Nous avons évalué les performances de notre méthode sur deux jeux de données tests correspondant à deux environnements fabriqués par l'Homme (*Man-made*).

Le premier illustre les environnements urbains en vue piétonne et est constitué de 2000 images de la ville de Nancy issues de Google Street View. La même procédure que pour l'apprentissage (Sec. 2.3) est employée pour déduire les points de fuite de Manhattan des données brutes (i.e. les panoramas équirectangles). La courte focale du système d'acquisition 360° accentue les effets de

perspective qui sont déjà importants dans les vues piétonnes en raison de la distance typique de l'observateur aux bâtiments et des dimensions de ceux-ci. Associé à une résolution assez faible (480×640) des images, cela réduit le nombre et la qualité des segments extraits comme on l'a décrit dans la section 2.3.

Le second jeu de données se concentre sur les environnements intérieurs comme des bureaux, ou des salles de cours. Il est constitué de 2000 images tests issues de la base de Stanford qui n'ont pas servies pour l'apprentissage. Si les dimensions des objets de la scène font que la perspective est moins marquée que pour la base urbaine, la complexité de le scène y est plus importante. Il y a en effet beaucoup plus d'objets qui ne sont pas forcément en accord avec le repère de Manhattan (mobilier, cables, ...).

2.6.2 Résultats

On cherche dans cette partie à estimer la rotation de la caméra en la déduisant du repère de Manhattan. Afin de comparer les résultats de notre méthode par rapport aux méthodes de l'état de l'art on utilise une métrique d'erreur qui a réellement du sens sur $SO(3)$. Parmi les différentes métriques définies dans [55], on choisit la distance $d(R_1, R_2) = \left\|\log(R_1 R_2^T)\right\|$ qui a l'interprétation géométrique la plus évidente.

L'architecture typique des bâtiments avec des murs homogènes et des fenêtres étroites ne laisse possible la détection que de quelques segments très courts sur les fenêtres pour estimer les points de fuite horizontaux. Combinée aux difficultés liées à la faible résolution et aux déformation perspectives, la convergence de ces segments se distingue alors mal des fausses détections de points de fuite issues du bruit (arbres, voitures, piétons, ...) ou de la répétitions de motifs. Les méthodes itératives qui nécessitent une initialisation convergent alors souvent vers un minimum local [4]. Les approches robustes par échantillonnage sur les données [106] souffrent aussi de fausses détections de points de fuite à cause du taux d'*inliers* faible et quasi-identique entre les modèles. Même le modèle *a contrario* proposé dans [70] trouve des alignements dans le domaine dual qui ne correspondent pas à des points de fuite réels de l'image. La multiplication des objets non-Manhattan dans les environnements intérieurs est aussi une source de fausses détections même si elle est mieux tolérée par les différentes approches comme le montre les résultats sur la base de Stanford (Fig. 2.15).

On constate également que, lorsque la rotation est estimée *a posteriori* à partir des points de fuite détectés, les résultats en pâtissent (Fig. 2.15). En effet la contrainte d'orthogonalité permet de régulariser le problème. Sans elle, il y a plus de fausses détections et il faut choisir le triplet de Manhattan parmi les points de fuite détectés. Cela est généralement fait par un critère heuristique d'orthogonalité. Or l'absence d'un des 3 points de fuite dans l'image peut alors conduire à un mauvais choix de solution. Enfin même quand ceux-ci ont été choisis correctement, une orthogonalisation du triplet purement géométrique par SVD peut écarter la rotation de sa solution optimale vis-à-vis de l'image.

La non-linéarité des équations issues de ces formulations sur $SO(3)$ entraine cependant une forte complexité de résolution pour les méthodes [85] et [16]. Si leur précision est meilleure, le temps de calcul est très significativement augmenté ce qui exclut ces méthodes d'applications

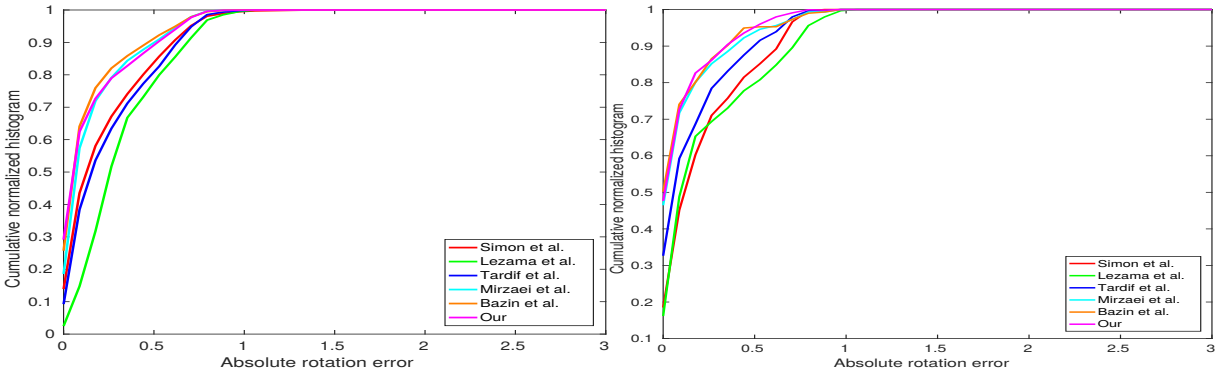

FIGURE 2.15 – Résultats statistiques de l'estimation de la rotation de Manhattan selon la métrique d'erreur sur $SO(3)$. Les résultats sont présentés par l'histogramme cumulé normalisé de cette erreur sur les bases de test de Google Street View (à gauche) et Stanford (à droite). Notre méthode est comparée à Simon et al. [101], Lezama et al. [70], Tardif et al. [106], Mirzaei et al. [85] et Bazin et al. [17]

TABLE 2.1 – Temps d'exécution moyen par image des méthodes comparées

	Simon et al. [101]	Lezama et al. [70]	Tardif et al. [106]	Mirzaei et al. [85]	Bazin et al. [17]	Nous
Temps (secondes)	0.28	58.14	0.75	1.34	0.29	0.12

temps-réel pour la réalité augmentée ou la robotique mobile (Tab. 2.1). Pour [16], le temps de calcul moyen étant de plus de 10 min, il n'a pas été possible de calculer les résultats sur l'intégralité des images de test. Seule l'approche de [17] offre de bon résultats statistiques tout en restant efficace pour une formulation dans $SO(3)$.

Notre méthode permet de partir d'une estimation qui tient compte de l'image entière et qui s'appuie potentiellement sur plus d'information que les seules convergences de segments. Cela permet d'initialiser une solution moins sensible aux minima locaux très communs dans les approches basées uniquement sur les segments. Le raffinement permet alors de gagner en précision avec une formulation dans $SO(3)$ tout en gardant une résolution très rapide grâce au filtrage des données V_i par la segmentation de Manhattan et aux solutions analytiques de la *M-Step* de l'algorithme EM (Fig. 2.18, deux premières lignes). Cette formulation bayésienne est également robuste en cela qu'elle gère les anomalies et les cas d'absence d'un des trois points de fuite dans l'image via les poids du mélange π_j. Ainsi si [17] est plus précise statistiquement (Fig. 2.15), notre méthode résout des cas complexes pauvres en segments pertinents sur lesquels cette méthode échoue. Il s'agit notamment de situations où le taux d'anomalie est très supérieur à 80 % dû à la présence d'arbres ou de mobilier et à la faible résolution de bâtiments partiellement visibles (Fig. 2.16 et 2.17).

De plus lorsque le nombre de segments extraits est vraiment trop faible ($M < 200$), on garde l'estimation initiale du réseau sans raffinement (Fig. 2.18, deux dernière lignes). Dans ces cas, l'ensemble des méthodes basées segments échouent. S'il n'est pas précis le réseau estime néanmoins une rotation cohérente avec le contenu visuel de l'image. La forte présence de ciel ou de plafond est fortement corrélée à un angle de tangage important, ce que le réseau est capable d'apprendre.

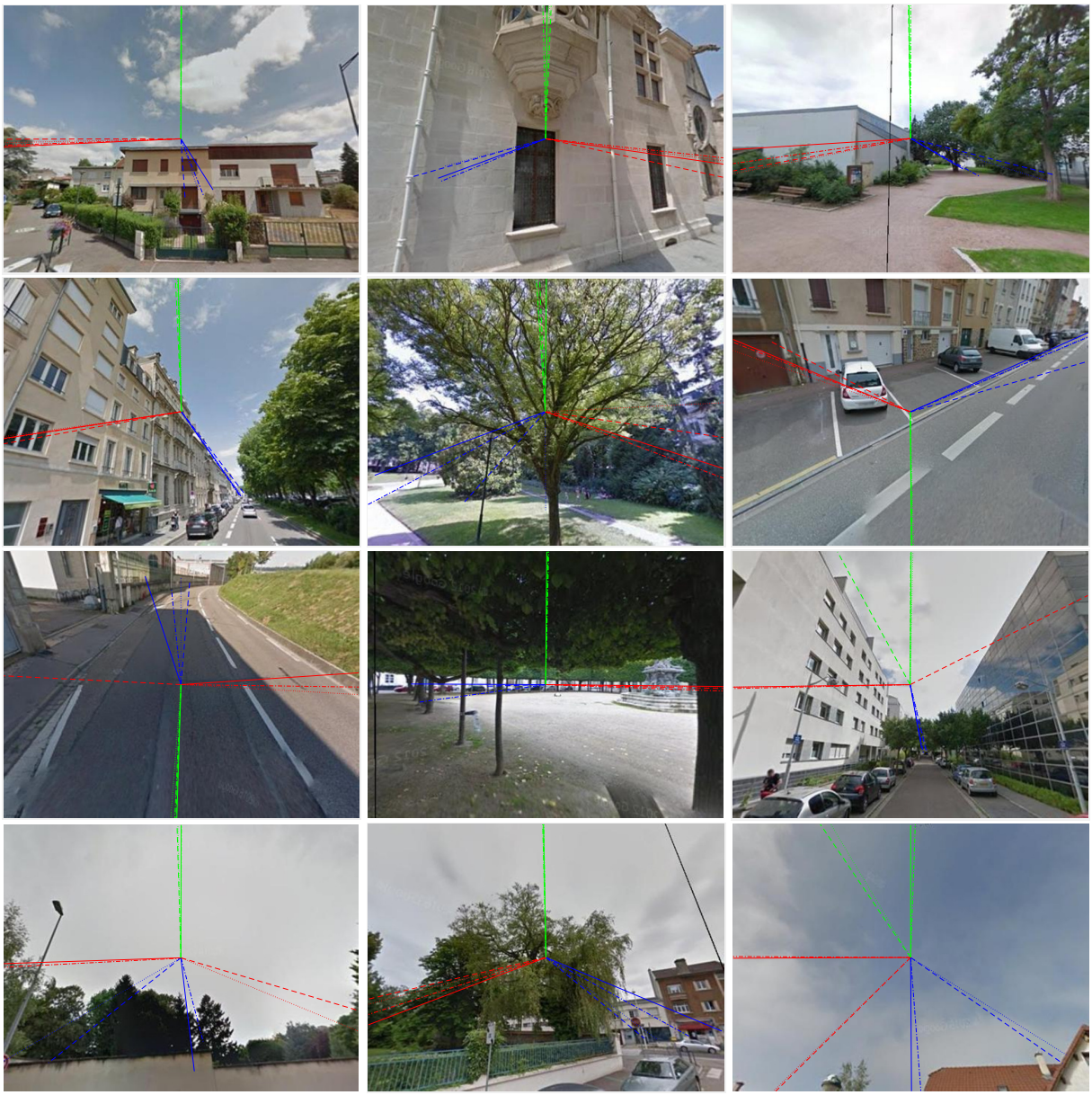

FIGURE 2.16 – Cas complexes de la base de tests Google Street View. La vérité-terrain est en traits pleins, [17] en pointillés, [85] en tirets et notre méthode en alternance de tirets et de points.

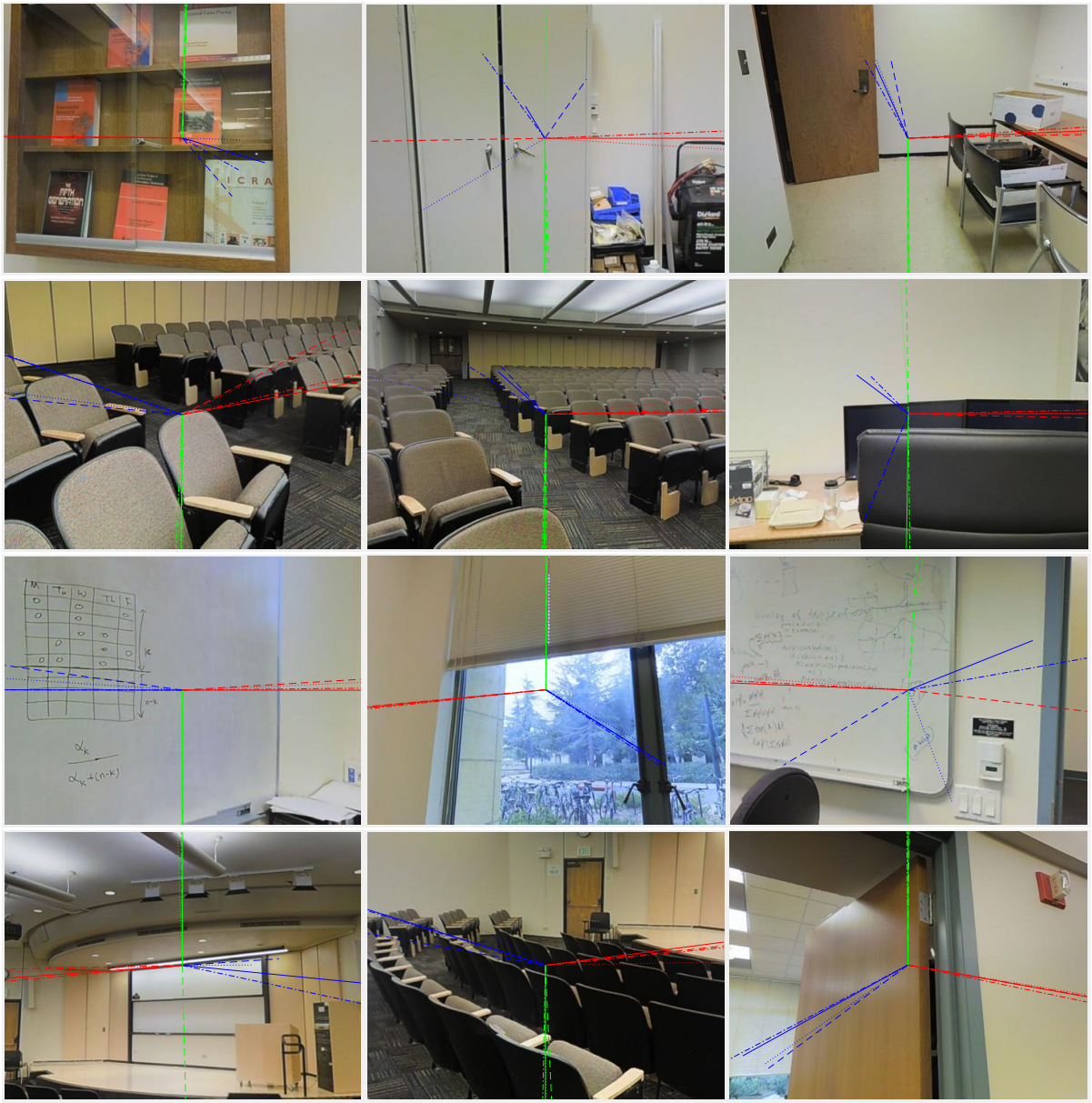

FIGURE 2.17 – Cas complexes de la base de tests Stanford. La vérité-terrain est en traits pleins, [17] en pointillés, [85] en tirets et notre méthode en alternance de tirets et de points.

Ainsi notre méthode offre des résultats autant, voire plus précis, que les approches lentes dans $SO(3)$ et largement plus précises que les approches par orthogonalisation *a posteriori* tout en garantissant un temps d'exécution très faible (≈ 0.1 s) (Fig. 2.15).

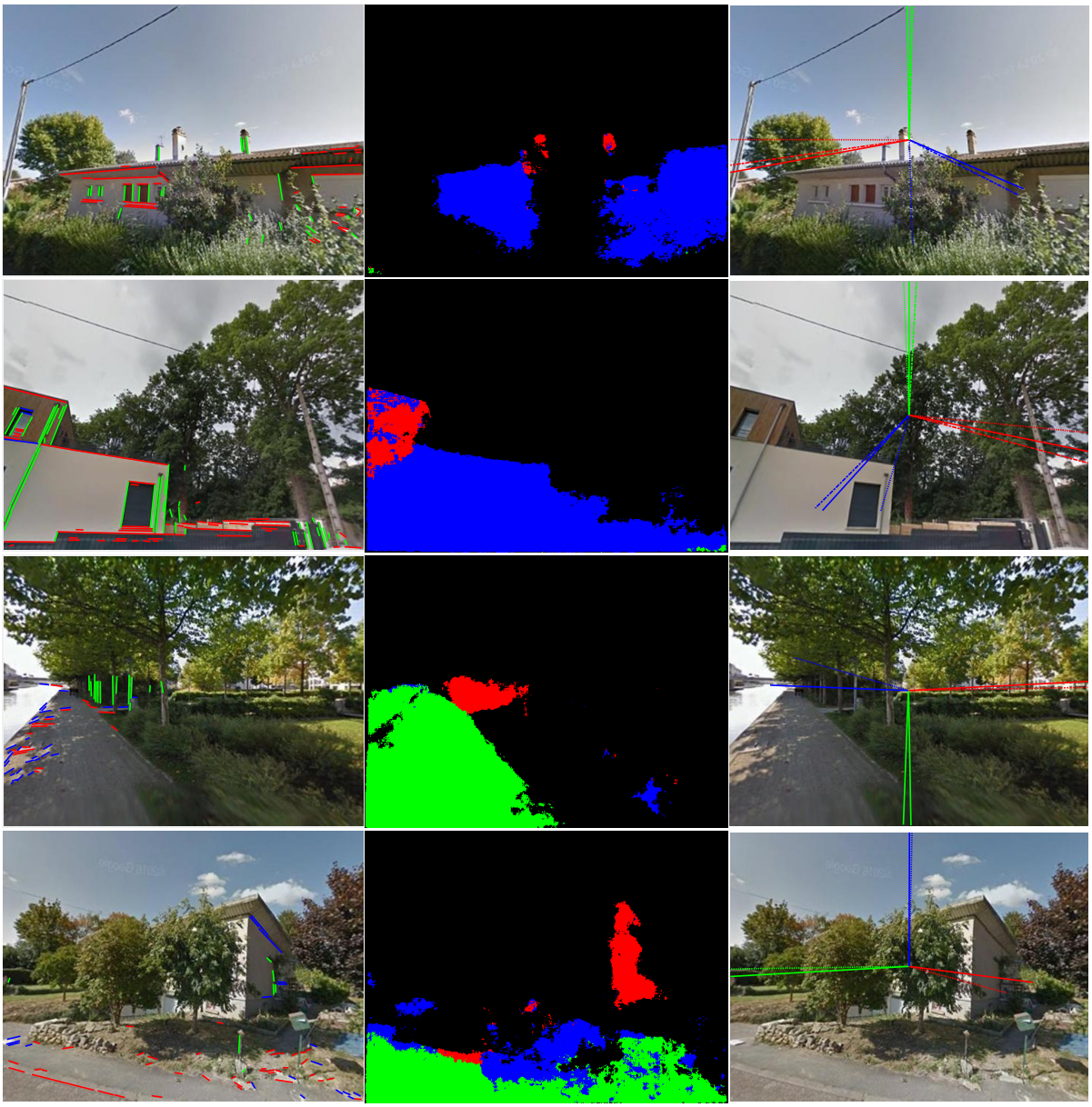

FIGURE 2.18 – Exemples de résultats sur la base de tests Google Street View. On présente de gauche à droite, la classification *a priori* des segments, la segmentation en plan de Manhattan et la rotation de Manhattan projetée dans l'image. La vérité-terrain est en traits pleins, l'initialisation du réseau en pointillés et le raffinement en tirets en alternance de tirets et de points. Les deux premières lignes montrent des cas où l'initialisation a été améliorée par le raffinement et les deux dernières des cas où la raffinement n'a pas été fait faute d'un nombre suffisant de segments.

Malgré tout, notre méthode peut échouer lorsque la segmentation de Manhattan inférée est

très erronnée. En effet la segmentation fait parfois des erreurs sur les zones proches de la caméra où l'échelle des déformations est encore trop grande pour distinguer si la façade est vue de face ou de côté (Fig. 2.19). Cela peut mener à un mauvais filtrage des intersections qui créent alors des minima locaux sur la sphère. Il peut également y avoir des erreurs d'étiquetage entre les points de fuite horizontaux pour des façades orientées selon l'angle limite de 45° qui contrôle le basculement de l'étiquette "Z" (en face) de l'étiquette "X" (de côté). Ces cas ne sont cependant pas problématiques du moment que la façade est segmentée uniformément. L'estimation de la rotation par le réseau a moins d'influence sur les résultats, l'étape de raffinement arrivant le plus souvent à rattraper l'erreur initiale comme on le voit sur la figure 2.13.

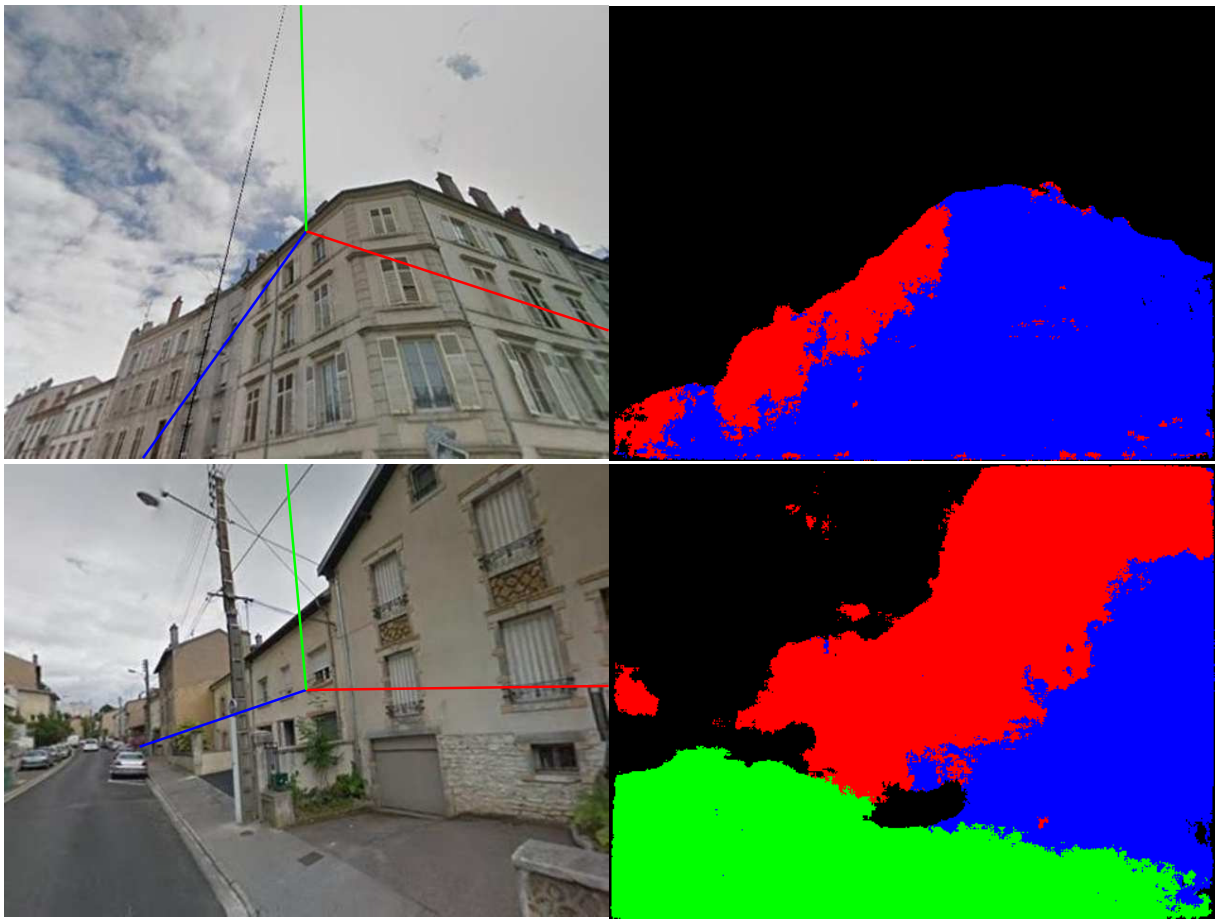

FIGURE 2.19 – Cas d'échecs de l'estimation de la rotation (projetée sur l'image à gauche) à cause d'une mauvaise segmentation en plans de Manhattan (à droite).

2.6.3 Limites

La base Google Street View souffre d'un problème de fiabilité de la segmentation. En effet, les cartes de normales équirectangles proviennent de la géométrie de Google Map et il arrive qu'un bâtiment manque ou que sa hauteur soit mal renseignée. Les anomalies de segmentation

qui en découlent perturbent l'apprentissage en maintenant une erreur anormalement grande dans certaines zones de l'image. L'utilisation d'un modèle géométrique plus précis (iTowns IGN, TorontoCity [115]) devrait permettre des améliorations substantielles sachant que l'essentiel des cas d'échecs viennent d'une mauvaise segmentation.

Un biais dans les bases d'apprentissage limite la généralisation de notre méthode à tout type d'images. En effet autant pour la base de scènes d'extérieur que pour celle de scènes d'intérieur, les images ont toutes été prises avec une même caméra dont les paramètres intrinsèques sont fixés y compris la taille de la matrice CCD. Cette absence de variabilité dans les paramètres intrinsèques est préjudiciable à notre méthode pour des images issues d'une autre caméra (dont la matrice de calibration est notée K'). Les déformations de structures sur laquelle s'appuient le CNN n'ont alors plus le même aspect à cause du changement de projection et du redimensionnement de l'image en entré de réseau.

Ces effets pourraient en théorie être compensés par une transformation de l'image courante KK'^{-1} pour se ramener dans les conditions d'acquisition de l'apprentissage. Cependant, dans notre cas, la longueur focale de l'apprentissage étant très petite (5mm), l'image courante corrigée est limitée par un champ de vision plus faible. Cela ce traduit concrètement sur l'image corrigée par une large bordure noire autour du contenu de l'image courante. Cette correction provoque un effet de bords qui perturbe l'inférence du réseau de neurones. Bien que l'utilisation de convolutions à trou puisse résoudre ce problème, notre méthode n'est actuellement utilisable de façon optimale que pour des images issues d'un caméra possédant grossièrement les mêmes paramètres intrinsèques que ceux de la caméra d'acquisition de la base d'apprentissage.

Si cette dépendance au matériel peut sembler une contrainte forte de la méthode, de nombreuses applications en milieux urbains relèvent de celle-ci notamment en robotique mobile ou en réalité augmentée embarquée. En effet dans ces conditions, il est facile de constituer une nouvelle base d'apprentissage. De plus le centre optique étant généralement assimilable au centre de l'image, le seul paramètre vraiment critique est la longueur focale. Sur la plupart des appareils mobiles, la gamme de valeur de longueur focale est limitée, ce qui garantie une certaine transferabilité de la méthode à des caméras qui n'ont pas fait l'acquisition de la base d'apprentissage.

Pour illustrer cela, nous avons calculé l'erreur sur la ligne d'horizon pour le jeu de données standard pour la détection de points de fuite York Urban (Fig. 2.20). Nous reprenons les résultats de l'état de l'art sur cette base avec cette métrique à partir de la courbe de [101]. Malgré une longueur focale différente, les performances de notre méthode restent dans la moyenne de celles de l'état de l'art, et ce sur une métrique qui ne tient pas compte de l'orthogonalité.

2.7 Conclusion

Nous avons proposé un réseau de neurones convolutionnels qui estime la rotation de Manhattan et infère la segmentation de l'image selon les directions de Manhattan conjointement. Cette approche s'appuie sur l'ensemble de l'image et donc sur une information plus riche pour déduire ces points de fuite que les seuls segments de droites. Si la précision est limitée, nous introduisons également une méthode de raffinement rapide qui repose cette fois sur les segments de droites. Dans un premier temps les intersections de segments sont filtrées par les contraintes

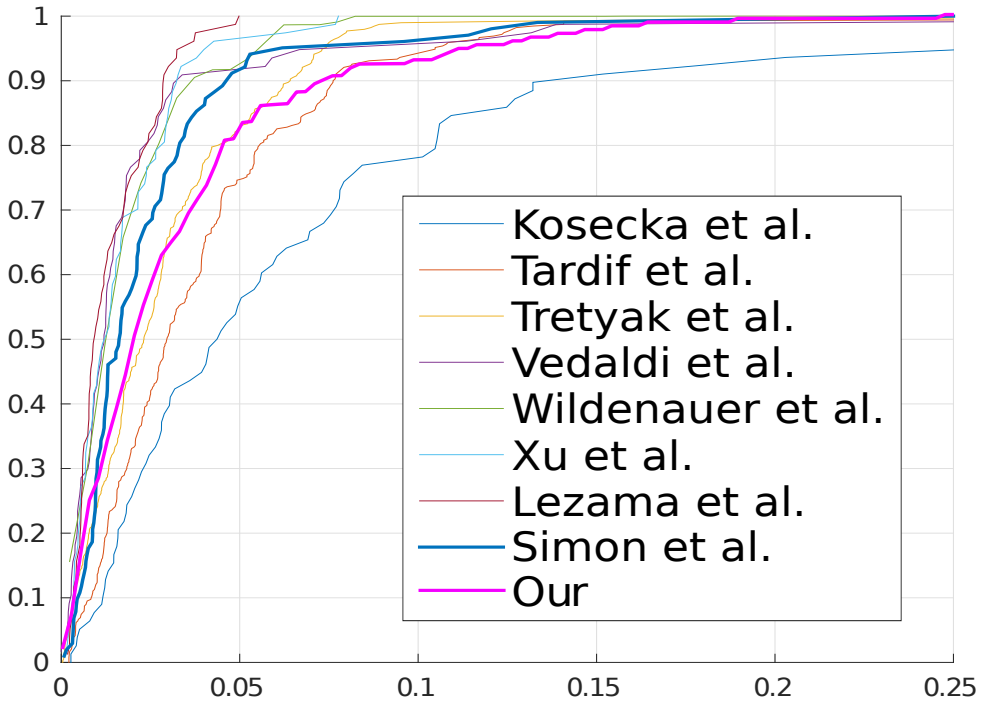

FIGURE 2.20 – Résultats statistiques de notre méthode sur la base de test York Urban dont la calibration de la caméra est très différente de celle de l'apprentissage. Les résultats sont présentés par l'histogramme cumulé normalisé de l'erreur sur la ligne d'horizon.

de la segmentation. Ensuite le problème de l'estimation du repère de Manhattan est formulé dans un cadre bayésien directement sur $SO(3)$. La résolution prend alors la forme d'un algorithme EM dont la résolution analytique de la *M-Step* permet une grande efficacité. Dans les mêmes conditions d'acquisition, les résultats statistiques montrent une meilleure précision et une plus grande robustesse que les méthodes de l'état de l'art dans un temps d'exécution réduit. Même dans des conditions différentes, les résultats restent comparables à l'état de l'art. Une amélioration possible serait de profiter des couches du réseau pour extraire des alignements de structures complexes (motifs architecturaux, fenêtres, roues de voitures,...) qui renforceraient les segments de droites de l'image dans l'étape de raffinement.

Chapitre 3

Propositions de façades pour la détection et la reconnaissance de bâtiments

Si le chapitre précédent nous a permis de rectifier l'image courante relativement aux différents bâtiments de la scène, seule la rotation de la caméra a été estimée. Pour compléter la pose de la caméra il faut aussi estimer sa translation. On suppose que l'on dispose d'une base de données de façades de référence qui constitue notre modèle de scène. On peut alors inférer la translation en mettant en correspondance les façades visibles dans l'image courante avec celles de la base. Pour cela dans un premier temps on cherche à détecter toutes les façades de l'image rectifiée puis identifier ces détections relativement aux façades de référence. Si cette approche « Bottom-Up », de l'image vers le modèle permet de gérer efficacement la mise en correspondance avec un nombre important de façades de référence, la précision géométrique de ces mises en correspondance est limitée. Cette initialisation grossière suffit cependant néanmoins à certaines applications de réalité augmentée. Pour les applications nécessitant davantage de précision dans le calcul de pose, nous présentons une étape de raffinement « Top-Down » dans le chapitre 4.

Nous proposons donc dans ce chapitre une nouvelle méthode efficace de détection et de reconnaissance de façades. Celle-ci repose sur une étape de propositions d'objets spécifique aux façades. Nous définissons notamment de nouveaux indices qui mesurent des caractéristiques typiques des façades comme leur sémantique, leur symétrie, ou la répétition de motifs à leur surface. Ces indices sont combinés pour générer un nombre réduit de bons candidats façades rapidement. Nous montrons que notre méthode surpasse les autres méthodes de propositions d'objets pour cette tâche particulière sur les 1000 images de la Zurich Building Database ainsi que sur une partie de la base d'Oxford. Nous démontrons l'intérêt d'une telle procédure pour la reconnaissance de façades et l'initialisation de pose de caméra. Cette étape est effectuée dans un système efficace qui classifie et met en correspondance les candidats avec des descripteurs basés sur un réseau de neurones convolutionnels. Nous prouvons que cette approche est plus robuste aux changements de points de vue et aux occultations que les méthodes de reconnaissance d'objets classiques.

3.1 Travaux liés

La reconnaissance d'objets a fait de gros progrès ces dernières années avec l'émergence de représentations d'image compactes et robustes aux changements d'apparence (des descripteurs) combinées à de nouvelles méthodes de classification [26][28][90]. Si les étapes de description et de classification ont longtemps été indépendantes, les approches actuelles les plus performantes reposent sur l'entrainement de l'ensemble de la chaîne de reconnaissance par des algorithmes d'apprentissage profond utilisant des réseaux de neurones convolutionnels [66][102].

La détection d'objets est traditionnellement formulée comme un problème de reconnaissance d'objets par fenêtre glissante. Pour chaque localisation et échelle de la fenêtre dans l'image, un descripteur est extrait puis classifié. Cette formulation présente une complexité algorithmique importante. Ceci a longtemps limité la détection d'objets à l'emploi de descripteurs très simples couplés à des méthodes de classification rapides [112]. Pour bénéficier des progrès de l'apprentissage profond sur la reconnaissance d'image sans faire exploser la complexité, les méthodes de détection récentes [45][44] reposent sur des techniques de propositions d'objets [1].

L'objectif de ces méthodes est de générer un ensemble réduit de boites englobantes qui sont potentiellement des objets et cela très rapidement. Un classifieur plus complexe peut alors traiter ces candidats dans un second temps. L'idée d'une mesure d' « *Objectness* » comme étape de pré-selection destinée à produire un faible nombre de régions de sorte que les premières régions contiennent probablement un objet a été développée dans [1]. Leur mesure est basée sur la combinaison dans un cadre bayésien de plusieurs indices visuels. Certains indices cherchent à caractériser la différence d'un objet par rapport à l'arrière plan (contraste de couleur entre une région et ses alentours, saillance d'une région par rapport à la distribution fréquentielle des images naturelles). D'autres indices visent à caractériser le fait qu'un objet a des frontières fermées (*closed-boundaries*) comme avec la densité de contours aux alentours de la région où les chevauchements de celle-ci avec une segmentation en super-pixels.

Depuis lors, plusieurs autres méthodes de propositions d'objets ont été proposées qui reprennent cette notion de frontière fermée pour caractériser un objet. Dans Selective Search [110], les auteurs utilisent une segmentation multi-échelle de l'image pour générer les propositions (Fig. 3.1). Cette segmentation est obtenue par une procédure gloutonne de fusion hiérarchique de régions selon une mesure de similarité entre régions qui intègre leur couleur, leur texture et leur forme.

Le score d'EdgeBoxes[122] est calculé en sommant les contours qui sont entièrement contenus dans la région considérée. Cette mesure simple et rapide à calculer apparaît comme le meilleur compromis relativement au temps de calcul dans une étude comparative sur les méthodes de propositions d'objets [52].

Bien que ces méthodes intègrent des critères qui peuvent bien s'adapter aux façades comme le contraste de couleurs fort entre la façade et ses alentours, d'autres hypothèses sur lesquelles reposent ces méthodes peuvent ne pas être vérifiées sur les façades (frontières non toujours fermées, grande similarité d'apparence entre façades adjacentes). De plus ces méthodes sont très génériques et ne profitent donc pas de la structure très particulière des façades ce qui a un impact sur leur efficacité dans les milieux urbains.

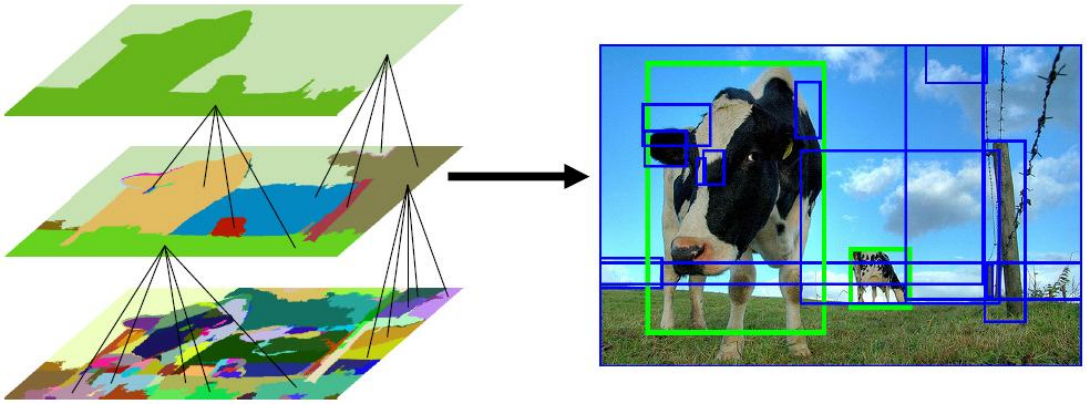

FIGURE 3.1 – Illustration de la segmentation hiérarchique de Selective Search [110].

Si l'application naturelle de ces techniques de propositions d'objets est la détection, elles ont également été récemment appliquées à la reconnaissance de lieux. En effet la comparaison entre deux images entières est très sensible aux changements de points de vues, alors que l'extraction de points de repère visuels (*Visual Landmarks*) y est plus robuste [105]. Si les descripteurs utilisant des réseaux de neurones convolutionnels (CNN) surpassent largement leurs concurrents pour comparer ces points de repère dans les deux images, ils sont cependant coûteux en temps de calcul. Il est alors nécessaire de sélectionner des candidats par une méthode de propositions d'objets pour extraire et mettre en correspondance ces repères visuels en temps raisonnable. Si EdgeBoxes semble le meilleur compromis qualité des proposition/vitesse [52], sa trop grande généricité entraine le besoin d'un nombre important de candidats en milieu urbain, alors qu'il pourrait être réduit en se focalisant sur les bâtiments comme repères visuels.

Comme on l'a montré, l'efficacité est une mesure clé pour la détection et la reconnaissance d'objets. Pour garantir cette efficacité, notre méthode se décompose en trois étapes. Le première étape vise à proposer des candidats façades rapidement en se basant sur des particularités structurelles des façades (symétrie, répétitions, ...). Ces caractéristiques sont évaluées par des indices calculés à partir des contours et d'une segmentation sémantique de l'image inférés par une réseau de neurones convolutionnels. Cette inférence est ensuite re-exploitée pour construire des descripteurs qui servent à la classification des candidats en façade ou non. Aucun autre passage dans le réseau n'est nécessaire pour le calcul de ces descripteurs dérivés de [50]. Enfin ces mêmes descripteurs sont utilisés pour mettre en correspondance les façades détectées avec la base de référence. Ces différentes étapes sont illustrées en Fig. 3.2 et nous les détaillons dans la suite.

3.2 Propositions de façades

Notre algorithme pour la proposition de façades est une procédure en deux étapes. L'image est préalablement rectifiée suivant la méthode introduite dans le chapitre précédent. Un premier ensemble de candidats est initialisé à partir des contours de l'image. Puis des indices propres aux façades sont évalués sur cet ensemble et les meilleurs candidats sont sélectionnés sur un

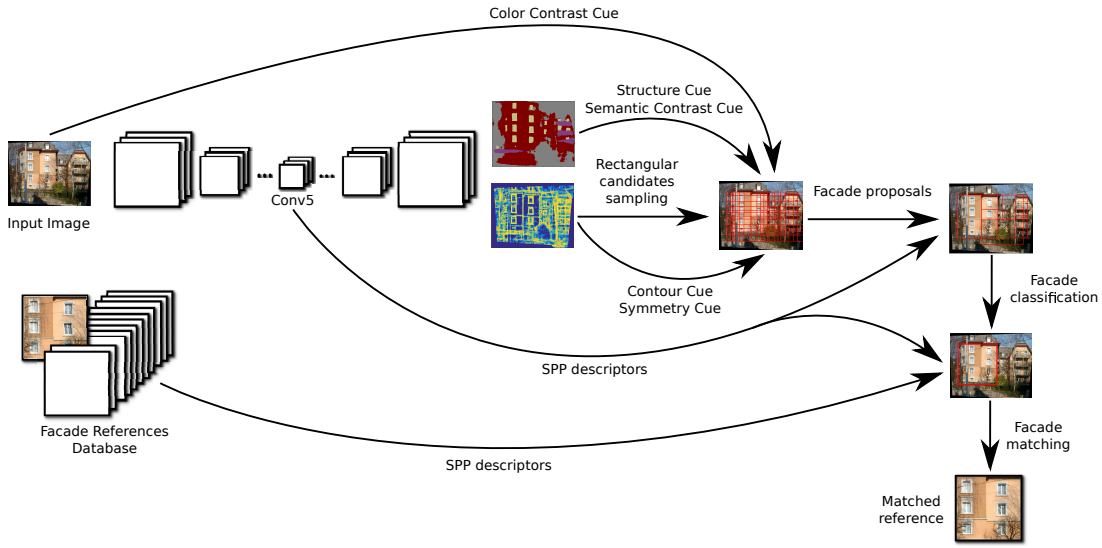

FIGURE 3.2 – Vue d'ensemble de notre méthode de détection et de reconnaissance de façades.

score qui combine les valeurs obtenues dans un cadre d'apprentissage automatique. Une base de données constituée de 1500 images de Google Street View et d'ImageNet a été utilisée pour l'apprentissage tandis que les 1000 images de la Zurich Building Database (ZuBuD) [1] ont été utilisées pour le test. Les boites englobantes de façade qui ont été placées manuellement sont appelées vérité-terrain (GT) dans la suite du chapitre.

3.2.1 Segmentation sémantique et détection conjointe de contour

La segmentation sémantique ne résout pas à elle seule le problème de détection de façades car elle est incapable de distinguer des façades adjacentes. Cependant nous exploitons cette classification au niveau pixel dans nos indices de proposition de façades.

Nous entrainons une version modifiée du réseau SegNet [9] pour inférer 7 classes sémantiques différentes ("arrière-plan", "façade", "fenêtre", "balcon", "porte", "ciel" et "route") ainsi que les contours (Fig. 3.3). Résoudre conjointement ces deux problèmes permet à la segmentation sémantique d'être plus sensible aux bords des éléments et aux contours d'être localisés sur les bordures qui ont du sens (Fig. 3.4). La base de données d'entrainement et de test est une fusion des bases CMPfaçadeDB1 [2], eTrims [3], ECP [4], INRIA [5] and LabelMefaçade [39]. Elle contient une variété de bâtiments de style classique ou plus moderne de villes européennes (Paris, Prague, Berlin, ...). La vérité terrain des contours est choisie comme les bords des éléments de segmentation sémantique. L'architecture du réseau est la même que celle du réseau original mais la dernière couche de déconvolution a 9 sorties (7 pour la sémantique et 2 pour les contours) qui

1. http ://www.vision.ee.ethz.ch/showroom/zubud/
2. http ://cmp.felk.cvut.cz/ tylecr1/façade/
3. http ://www.ipb.uni-bonn.de/projects/etrims_db/
4. http ://vision.mas.ecp.fr/Personnel/teboul/data.php
5. https ://github.com/raghudeep/ParisArtDecofaçadesDataset/

sont découpées en deux couches différentes. La fonction de coût final utilisée pour l'entrainement est une somme pondérée des fonctions de coût par entropie croisée (*cross-entropy*) de ces deux couches.

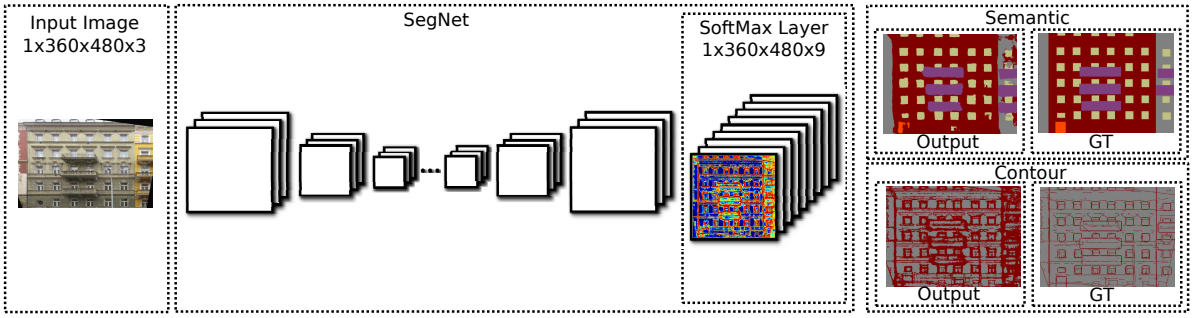

FIGURE 3.3 – Architecture de notre version modifiée de SegNet à deux sorties : une pour la segmentation sémantique et l'autre pour les contours.

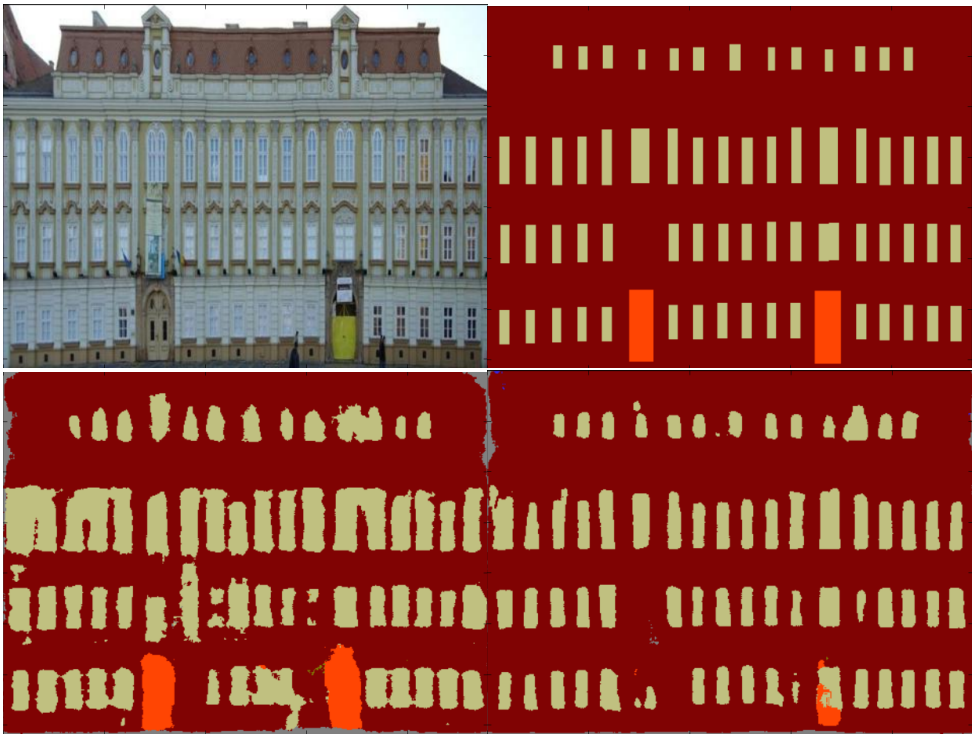

FIGURE 3.4 – Résultat de l'apprentissage conjoint sur la segmentation sémantique. Le premier rang montre l'image de la façade (gauche) et la vérité terrain de la segmentation sémantique (droite). Le second rang montre l'inférence par le réseau SegNet classique (gauche) et notre version modifiée (droite). La forme des fenêtres est plus rectangulaire dans notre version grâce à l'entrainement joint des contours et de la sémantique.

Le réseau pouvant se montrer sensible aux changements d'échelle, une approche multi-résolution

est employée pour résoudre les cas où le bâtiment est petit dans l'image originale. En plus de l'image rectifiée entière, 9 sous-images de dimensions moitié qui se chevauchent sont fournies en entrée du réseau (Fig. 3.5).

FIGURE 3.5 – Approche multi-résolutions pour le calcul des indices et la détection. A gauche l'image originale rectifiée et à droite les 9 sous-images de dimensions moitiés (zoom).

3.2.2 Génération de candidats rectangulaires

La principale hypothèse que l'on fait sur les façades est qu'elles sont rectangulaires. Comme on travaille sur des images rectifiées, de telles façades apparaissent véritablement comme des rectangles dans l'image. Nous choisissons de nous reposer sur les contours de l'image pour générer un premier ensemble de candidats. En effet, les bords des façades créent des gradients de forte valeur dans l'image. La carte de contours E est une des deux sorties de notre réseau SegNet modifié. Ces contours sont agrégés en deux histogrammes, l'un pour les contours accumulés verticalement H_x, l'autre pour les contours accumulés horizontalement H_y. Le produit de ces deux histogrammes $H_x \times H_y^T$ peut alors être vu comme une carte de vraisemblance de coins (Fig. 3.6). Les n maxima locaux de cette carte génèrent $\frac{n(n-1)}{2}$ rectangles. En fait, comme les deux paires de coins (haut-gauche,bas-droite) et (haut-droite,bas-gauche) définissent le même rectangle, seul un ensemble de $\frac{n(n-1)}{2}$ candidats façades sont retenus. Pour l'exemple, le nombre moyen de candidats façades par image est de 16288 à cette toute première étape sur l'ensemble des 1500 images de notre base d'apprentissage.

3.2.3 Indices caractéristiques de façade

Les façades partagent plusieurs caractéristiques visuelles. Elles sont généralement composées de structures rectangulaires comme les étages, les fenêtres, les balcons et les portes. Ces structures se répètent le long de la façade verticalement et horizontalement. Les façades présentent souvent une symétrie horizontale grossière (réflexion). Enfin les façades sont relativement homogènes en couleurs comparée à leur environnement direct. Pour chacune des façades candidates nous

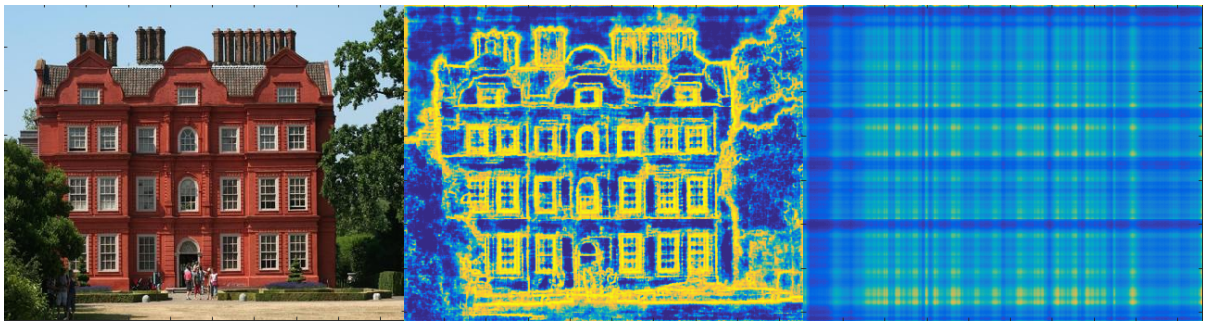

FIGURE 3.6 – Exemple d'image de la base d'apprentissage (à gauche), le contour de celle-ci par SegNet (au milieu) et la carte de vraisemblance de coins (à droite).

évaluons 6 différents critères « Had Hoc ». Nous réutilisons ici le critère de forme établi par Alexe et al. dans [2] et adaptons leur indice de contraste de couleurs. Notre critère de contours favorise les contours verticaux et horizontaux le long des arêtes ce qui le rend plus discriminant que celui également défini dans [2]. Trois nouveaux indices sont introduits cherchant à caractériser le contraste sémantique, la symétrie et les répétitions de motifs sur les façades. La combinaison de tous ces indices en un score nous permet ensuite de rejeter les candidats qui ne correspondent pas à nos hypothèses de façade et de ne garder que les meilleurs. Pour chacun des indices présentés ci-dessous, les figures 3.7, 3.9, 3.10, 3.13, 3.15 et 3.17 montrent le rectangle de plus haut indice parmi l'ensemble des candidats sur un exemple de notre base d'apprentissage (colonne de droite). Sur ces mêmes figures on montre également à gauche les probabilités pour une valeur d'un indice s d'être une façade $P(s|\text{façade})$ (en vert) ou de ne pas en être une $P(s|\text{non-façade})$ (en rouge).

Indice de forme

Les façades sont typiquement rectangulaires, mais tous les rectangles n'ont pas la même probabilité d'être observés. En effet, les lois architecturales n'autorisent que quelques valeurs de format de rectangle (*aspect-ratio*). Les façades infiniment fines sont par exemple presque impossibles. Nous avons appris la distribution de probabilité des deux paramètres de forme des rectangles (hauteur et largeur) sur notre ensemble d'apprentissage de 1500 images de la même manière que dans [2]. Nous avons utilisé une version discrétisée de cette distribution H à 24×24 cases pour plus d'efficacité dans le calcul de l'indice $s_{shape}(r) = H(h, w)$ (voir Fig. 3.7).

Indice de couleur

Si la couleur seule ne suffit pas à caractériser une façade, la différence de distribution de couleur entre la façade et son contexte local est une mesure bien plus intéressante pour cela comme décrit dans [2]. Cependant la faible combinaison de couleurs autorisées sur les façades (couleurs essentiellement pastel, désaturées) fait que résumer la distribution de couleurs par un histogramme uniforme n'est pas la meilleure solution notamment pour distinguer les façades adjacentes. Les distributions sont alors calculées sur un *Octree* qui partitione l'espace couleur de manière à renforcer leur différence entre l'intérieur et le contexte local (Fig. 3.8) tout en

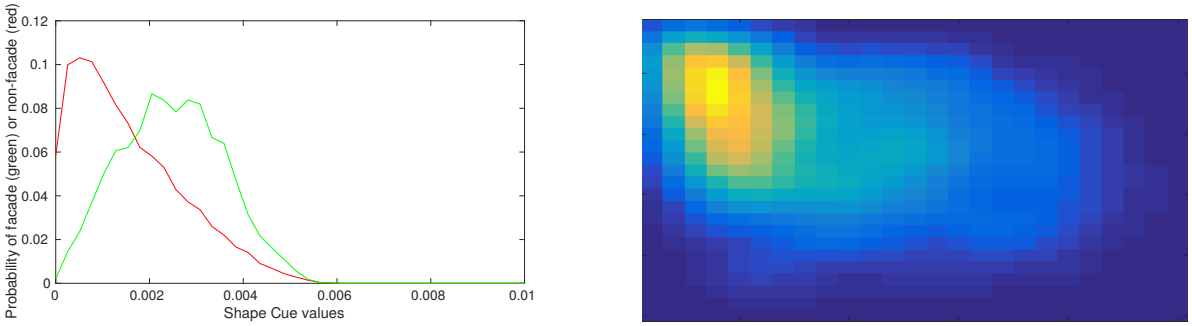

FIGURE 3.7 – Colonne de gauche : probabilité pour une valeur de l'indice de forme d'être obtenue sur une façade (en vert) ou ailleurs (en rouge). Colonne de droite : l'histogramme H qui représente la distribution en (hauteur,largeur) des rectangles vérité-terrain de la base d'apprentissage.

réduisant la complexité de calcul. Cet indice de « contraste de couleurs » entre l'intérieur du rectangle candidat et une région environnante permet alors de distinguer des façades adjacentes comme dans le cas de la figure 3.9 :

$$s_{color}(r) = 1 - \exp\left(-d_{\chi^2}\left(H_c^{b(r,\alpha)}, H_c^r\right)/\sigma_c\right) \tag{3.1}$$

où H_c^r et $H_c^{b(r,\beta)}$ sont respectivement l' histogramme couleur biaisé de l'intérieur de r et l' histogramme couleur biaisé de la bande d'épaisseur α qui entoure r. L'espace couleur LAB est discretisé en 38 cases selon un *Octree* dont la construction est détaillée en annexe.

Indice de contours orientés

Comme les façades sont rectangulaires, nous nous attendons à ce que les valeurs du gradient à la frontière de la boite englobante soient élevées (Fig. 3.10). Plus précisément nous pouvons espérer des contours verticaux (respectivement horizontaux) le long des arêtes verticales (respectivement horizontales) du candidat rectangle :

$$s_{cont}(r) = \frac{1}{2\beta(l+h)}\left(\sum_{b_t(r,\beta)\cup b_b(r,\beta)} E_x + \sum_{b_l(r,\beta)\cup b_r(r,\beta)} E_y\right) \tag{3.2}$$

où β est l'épaisseur de la bande $b_x(r,\beta)$ accolée au bord $x \in \{haut, bas, gauche, droite\}$ du rectangle r. E_x et E_y sont les images binaires des contours issus du réseau SegNet. (h,w) sont les hauteur et largeur de r, respectivement.

Indice de structure

Les fenêtres et les balcons se répètent le long des façades verticalement et horizontalement. Pour traduire cette remarque en une mesure, les étiquettes "fenêtre" et "balcon" sont tout d'abord accumulées horizontalement dans un histogramme H_x^r de 64 cases (Fig. 3.12). De même ces

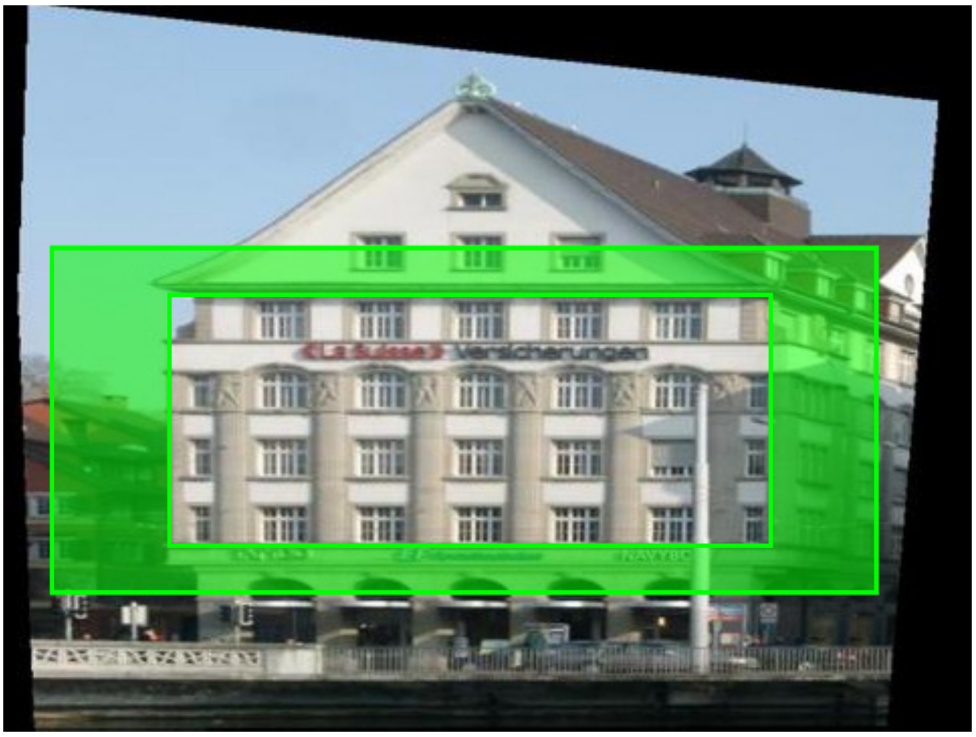

FIGURE 3.8 – La zone verte correspond au contexte local du rectangle intérieur en traits verts.

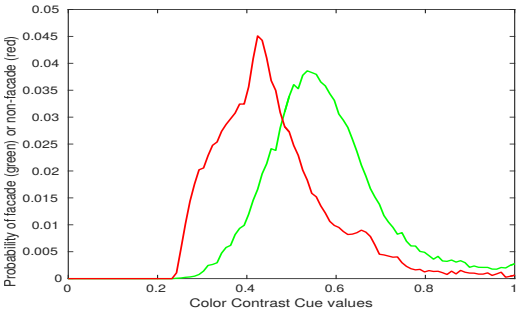

FIGURE 3.9 – Colonne de gauche : probabilité pour une valeur de l'indice de couleur d'être obtenue sur une façade (en vert) ou ailleurs (en rouge). Colonne de droite : rectangle de plus haute valeur d'indice parmi l'ensemble des rectangles candidats sur un exemple de la base d'apprentissage.

étiquettes sont accumulées verticalement dans un histogramme H_y^r. Ces deux signaux 1D sont très structurés avec des répétitions de rectangles. Dans une première version de cet indice de structure [37], celui-ci était basé sur la parcimonie des autocorrelations de ces signaux. Malgré le passage dans le domaine fréquentiel pour un calcul plus rapide de l'autocorrélation (Thérorème de Wiener-Khinchin), le nombre d'opérations restait important.

Nous lui préfèrons donc une approche qui compare ces signaux à des trains d'impulsions idéaux générés à partir d'un graphe qui sert de représentation compacte de la segmentation sémantique de l'image (Fig. 3.11). Pour construire ce graphe on extrait d'abord les composantes

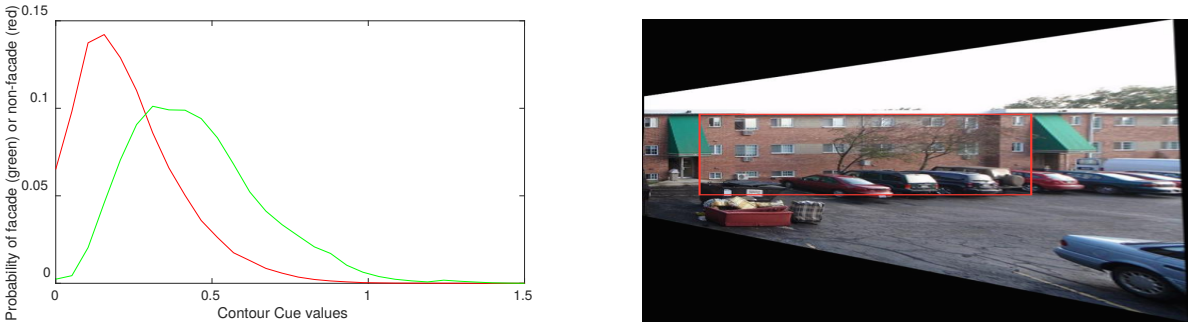

FIGURE 3.10 – Colonne de gauche : probabilité pour une valeur de l'indice de contours d'être obtenue sur une façade (en vert) ou ailleurs (en rouge). Colonne de droite : rectangle de plus haute valeur d'indice parmi l'ensemble des rectangles candidats sur un exemple de la base d'apprentissage.

connexes de la segmentation sémantique. Pour chacune d'elles, on calcule son centre C et sa matrice de covariance $\Sigma = diag(\sigma_x, \sigma_y)$. Les noeuds du graphe sont identifiés aux centres et la jonction entre deux noeuds C_i, C_j suit la procédure suivante :

— Pour chaque noeud C_i on définit $B_{i,x,r} = \left\{C_j,\ C_{j,x} > C_{i,x} \text{ et } |C_{i,x} - C_{j,x}| < \frac{3}{2}\min\left(\sigma_{i,x}, \sigma_{j,x}\right)\right\}$, l'ensemble des centres qui sont dans une bande verticale à droite de C_j de hauteur la dimension verticale minimale des deux composantes connexes. Symétriquement on définit $B_{i,x,l} = \left\{C_j,\ C_{j,x} < C_{i,x} \text{ et } |C_{i,x} - C_{j,x}| < \frac{3}{2}\min\left(\sigma_{i,x}, \sigma_{j,x}\right)\right\}$, l'ensemble des centres qui sont dans une bande verticale à gauche de C_j. On définit de même les ensembles $B_{i,y,r}$ et $B_{i,y,l}$.

— On joint le noeud C_i avec le noeud C_j dans le graphe si et seulement si C_j est le plus proche voisin de C_i dans $B_{i,x,r}$ ou dans $B_{i,x,l}$ ou dans $B_{i,y,r}$ ou dans $B_{i,y,l}$.

— Le poids de l'arête entre deux noeuds joints est la distance euclidienne $\|C_i - C_j\|_2$

Un train d'impulsions $H_{\phi,f,l,m}$ est défini par quatre paramètres : le décalage de phase ϕ, la fréquence f, la largeur l et la hauteur m des rectangles (Fig. 3.12 dernière colonne). Pour calculer les paramètres des 2 trains d'impulsions idéaux qui représentent les signaux H_x^r et H_y^r, on extrait tout d'abord le sous-graphe sémantique dont les noeuds sont inclus dans le rectangle considéré r. On ne décrit ici que le cas du signal vertical H_x^r mais la même procédure est employée pour le signal horizontal H_y^r. La médiane des poids des arêtes verticales donne une estimation robuste de $\frac{1}{f_x}$ et on estime la largeur l_x comme la moyenne des supports verticaux des composantes connexes (3 fois la moyenne des variances verticales). Le décalage de phase est ϕ_x est calculé comme l'abscisse de la première valeur du signal tel que $H_x^r > \frac{\sum H_x^r}{64}$ (l'histogramme possédant 64 cases). Enfin la hauteur est définie comme $m_x = H_x^r(\phi_x + l_x/2)$. L'indice de structure est alors calculé par différences absolues entre le signal et son train d'impulsion idéal. Il est plus discriminant que celui reposant sur l'autocorrélation car il caractérise à la fois la répétition mais aussi la rectangularité attendue d'un tel signal (Fig. 3.13) :

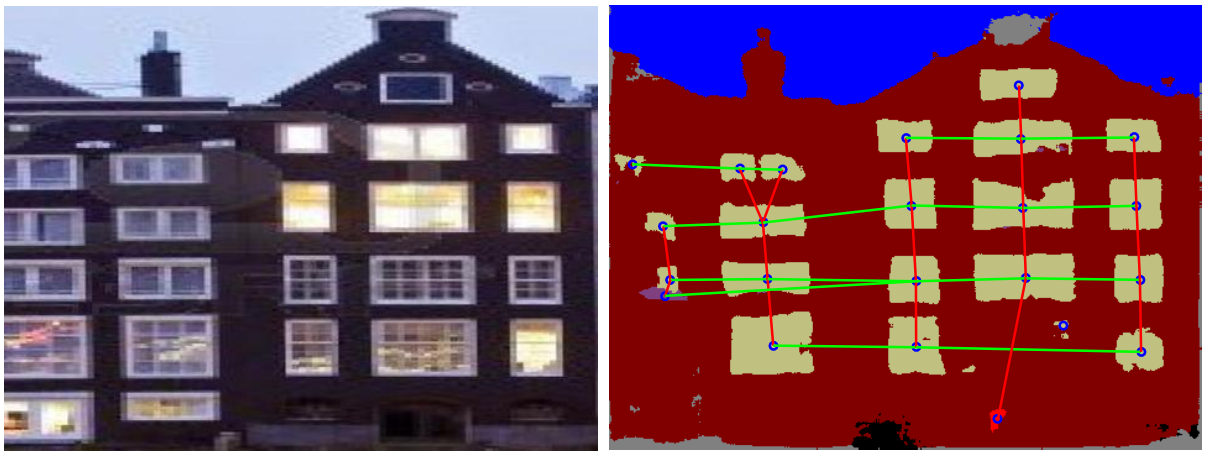

FIGURE 3.11 – A gauche une image de la base de données d'apprentissage. A droite le graphe de la segmentation sémantique lui correspondant. La distinction entre arêtes verticales (vert) et horizontales (rouge) est faite à la construction du graphe.

$$s_{struct}(r) = \left(\sum_{i=0}^{64} |H_x^r(i) - H_{\phi_x, f_x, l_x, m_x}(i)| \right) . \left(\sum_{i=0}^{64} |H_y^r(i) - H_{\phi_y, f_y, l_y, m_y}(i)| \right) \tag{3.3}$$

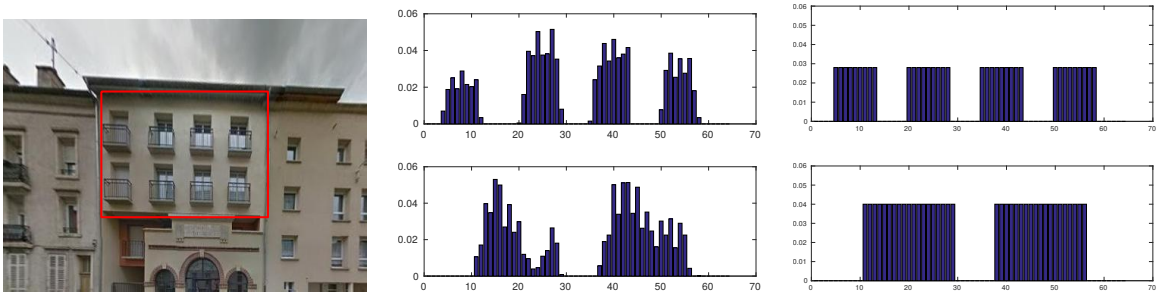

FIGURE 3.12 – Indice de structure : le rectangle candidat (à gauche), les histogrammes d'étiquette sémantiques accumulées verticalement et horizontalement dans ce rectangle (respectivement au milieu en haut et en bas) et les trains d'impulsions idéaux de ces mêmes signaux (à droite).

Indice de symétrie

Les façades ont une symétrie verticale qui n'est pas parfaite. Ce que l'on veut c'est une mesure qui évolue continuement avec l'aspect symétrique de la façade. Par exemple la corrélation croisée entre la moitié gauche et la moitié droite de l'image ne convient pas car elle serait très élevée même pour une toute petite asymétrie. Nous proposons donc un critère basé sur une description plus robuste utilisant les contours. Pour cela nous subdivisons d'abord le rectangle en $4 \times 4 = 16$ patchs. Pour chacun des patchs, on calcule un histogramme de gradient orienté (HOG) de 8 cases sur l'ensemble des contours issus de SegNet (Fig. 3.14). Nous évaluons la distance entre les 8

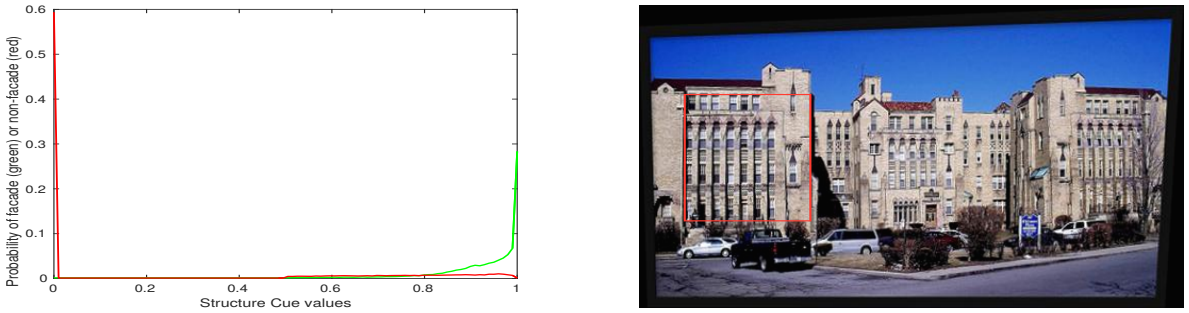

FIGURE 3.13 – Colonne de gauche : probabilité pour une valeur de l'indice de structure d'être obtenue sur une façade (en vert) ou ailleurs (en rouge). Colonne de droite : rectangle de plus haute valeur d'indice parmi l'ensemble des rectangles candidats sur un exemple de la base d'apprentissage.

patchs de gauche et leur symétrique à droite :

$$s_{sym}(r) = \exp\left(-\sum_{i=1}^{4}\sum_{j=1}^{2}\frac{d_{\chi^2}\left(H_e^{sym}(s(i,j)), H_e(i,j)\right)}{8\sigma_s}\right) \tag{3.4}$$

où $H_e(i,j)$ est le descripteur HOG à 8 cases du patch (i,j). $H_e^{sym}(i,j)$ est la version renversée du vecteur $H_e(i,j)$. s est la symétrie axiale d'axe la médiane verticale du rectangle r et d_{χ^2} la distance χ^2.

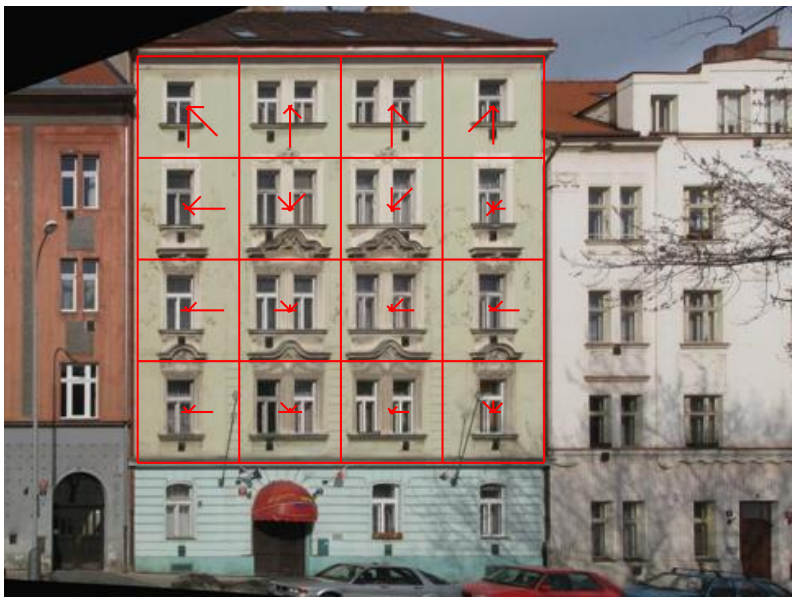

FIGURE 3.14 – Indice de symétrie : pour chacun des 16 patchs, les histogrammes de gradients orientés (HOG) sont représentés par un ensemble de segments. L'orientation d'un segment correspond à une case, et sa longueur à la valeur d'une case.

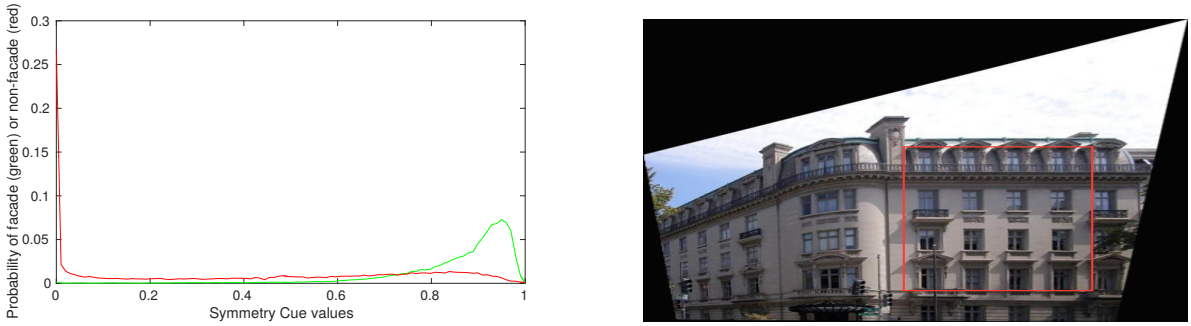

FIGURE 3.15 – Colonne de gauche : probabilité pour une valeur de l'indice de symétrie d'être obtenue sur une façade (en vert) ou ailleurs (en rouge). Colonne de droite : rectangle de plus haute valeur d'indice parmi l'ensemble des rectangles candidats sur un exemple de la base d'apprentissage.

Indice de rectification

Selon l'hypothèse des mondes de Manhattan les façades sont globalement planaires. De plus les bâtiments y étant assimilés à des parallélépipèdes rectangles, il n'est pas rare qu'une façade soit adjacente à deux autres façades qui lui sont perpendiculaires. Si les façades qui appartiennent au plan de rectification apparaissent comme en vue frontale sur l'image rectifiée, ce n'est pas le cas de ces façades perpendiculaires. On définit alors un indice de rectification en mesurant l'homogénéité de la segmentation en plans à l'intérieur du rectangle et dans le contexte local de celui-ci à partir des cartes de segmentation en plans de Manhattan du chapitre précédent :

$$s_{rect}(r) = \exp\left(-\left(1 + \frac{S_Z^r}{area(r)}\right) - \frac{S_Z^{b(r,\gamma)}}{area(b(r,\gamma))}\right) \tag{3.5}$$

où S_Z^r et $S_Z^{b(r,\gamma)}$ sont respectivement les sommes des orientations étiquetés "Z" à l'intérieur de r et dans la bande d'épaisseur γ qui entoure r.

Efficacité et optimisation des paramètres

Le calcul de tous les indices est fait en temps constant par rectangle grâce à l'utilisation d'intégrales images. Cette astuce détaillée dans [112] permet de calculer des sommes dans une région rectangulaire en 4 opérations seulement une fois l'image intégrale créée. On l'utilise ici pour calculer rapidement les histogrammes avec une intégrale image par case. Dans sa version originale dans [2], l'indice de contraste de couleur est calculé en utilisant un histogramme couleur uniforme sur $256 = 4 \times 8 \times 8$ cases ce qui en faisait l'indice le plus coûteux présenté. Au contraire, nous utilisons un partitionnement appris sur les données d'apprentissage de manière à maximiser la séparabilité des distributions de probabilité des valeurs de l'indice pour être une façade (Fig. 3.7, 3.9, 3.10, 3.13, 3.15, 3.17, colonne de gauche en vert) ou pour ne pas l'être (Fig. 3.7, 3.9, 3.10, 3.13, 3.15, 3.17, colonne de gauche en rouge). Le partitionnement est codé comme un *Octree* et

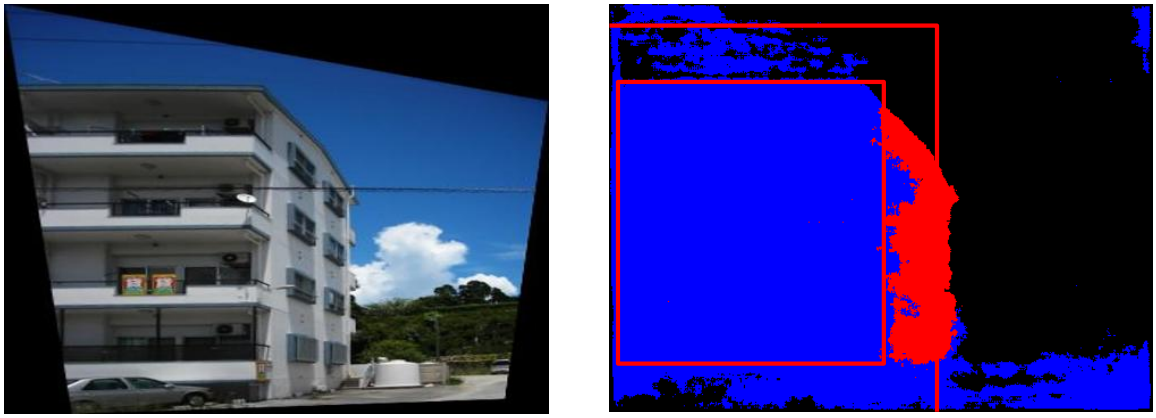

FIGURE 3.16 – A gauche : une image rectifiée de la base d'apprentissage. A droite : la carte d'orientation inférée par le CNN du chapitre précédent et les zones de contexte local considérées pour le rectangle candidat rouge.

FIGURE 3.17 – Colonne de gauche : probabilité pour une valeur de l'indice de rectification d'être obtenue sur une façade (en vert) ou ailleurs (en rouge). Colonne de droite : rectangle de plus haute valeur d'indice parmi l'ensemble des rectangles candidats sur un exemple de la base d'apprentissage.

l'optimisation est faite par recuit simulé (cf. Annexe). Le même critère de séparabilité est utilisé pour optimiser les valeurs des paramètres α, β et γ sur la base d'apprentissage. Cette fois une simple discrétisation de l'espace des paramètres est utilisée pour trouver les valeurs optimales qui sont respectivement 5%, 32% et 25% des dimensions du rectangle. La quantification des autres histogrammes et le nombre de patchs ont été choisis pour garder raisonnable la complexité des autres indices (sous les 10^3 opérations par rectangle). Enfin σ_c, σ_s représentent respectivement les écarts-types des distances du χ^2 des indices de couleur et de symétrie.

L'indice de structure étant basé sur la segmentation sémantique dont l'inférence par SegNet peut se montrer sensible aux échelles, celui-ci est calculé non pas sur l'image rectifiée mais sur une des sous-images de la décomposition multi-résolutions dont les dimensions sont plus proches des dimensions du rectangle candidat considéré (Fig. 3.5). Plus précisément, on choisit la sous-image telle que la valeur d'intersection sur union (*Intersection over Union* notée s_{IoU} [122]) entre le rectangle et la sous-image soit maximale. Si ce score maximal ne dépasse pas la valeur seuil de

$s_{IoU} \geq 0.5$, le calcul sera alors fait sur l'image rectifiée entière. Par ailleurs, les 10 images de la représentation multi-résolutions sont regroupées au sein de deux *batchs* en entrée du réseau pour profiter de la parallélisation et ainsi gagner en temps de calcul.

3.2.4 Combinaison des indices

La séparabilité entre les distributions de probabilité des valeurs d'indice pour les façades $P(s|\text{façade})$ et pour les non-façades $P(s|\text{non-façade})$ (Fig. 3.7, 3.9, 3.10, 3.13, 3.15, 3.17, colonne de gauche) est une indication de la capacité des différents indices à caractériser une façade. Nos critères spécifiques de structure, et de symétrie apparaissent alors bien plus discriminants que les autres indices. Néanmoins ces distributions ont été calculées dans des conditions très favorables sur les images de la base d'apprentissage. Dans les conditions réelles de tests, les façades sont souvent vues partiellement et les occultations peuvent être importantes ce qui peut casser la symétrie ou les répétitions. Au contraire, les indices de rectification et de couleurs sont peu sensibles aux parties cachées. Aussi combiner tous ces indices permet de profiter de leur complémentarité (Fig. 3.18). Pour cette combinaison en un seul score nous utilisons un perceptron multi-couche. Celui-ci est composé de 2 couches cachées de 8 neurones et d'une couche finale de régression du score.

Les données d'entrainement sont des rectangles générés par la procédure de la section 3.2.2, issus de diverses images de la base d'apprentissage. Le réseau de neurones est entrainé pour régresser la valeur d'intersection sur union (s_{IoU}) entre un de ces rectangles et le rectangle vérité-terrain de l'image dont il est extrait. Dans la littérature un seuil $s_{IoU} > 0.5$ est souvent utilisé pour décider si deux régions coïncident. Différentes situations avec leur valeur de s_{IoU} sont illustrées dans [122] montrant que ce seuil fait visuellement sens. Si, dans une version antérieure de la méthode, nous avions entrainé le réseau comme un classifieur avec cette valeur seuil entre façade et non-façade, le score *SoftMax* de classification s'est avéré être un score de combinaison d'indices perfectible. En effet, fortement binarisé, il ne donnait pas assez d'information sur la qualité de la coïncidence. Cela pouvait favoriser de mauvais candidats dans la suite de la procédure. En effet les candidats sont ensuite triés selon ce score et un algorithme glouton supprime les rectangles de faible score qui se chevauchent beaucoup. Cette procédure simple est décrite dans [2]. Elle consiste à sélectionner successivement les rectangles de plus haut score qui ne recouvrent pas (selon le critère $s_{IoU} \geq 0.5$) les rectangles précédemment sélectionnés.

Cette dernière étape vise à éviter les recouvrements. En effet deux rectangles qui coïncident presque parfaitement ont très probablement un score très proche. Ainsi on ne peut espérer prendre simplement les k premiers rectangles aux sens du score sans avoir de multiples instances de la même façade. On pourrait penser que la formulation de ce problème de recouvrement relève de la définition du problème de maximisation des poids d'un sous-ensemble de rectangles ne s'intersectant pas (*Maximum Weight Independent Set of Rectangles* abrégé *MWISR*). Si on en a implémentée une résolution relaxée par programmation linéaire, il s'avère que la qualité des solutions proposées diffère entre les images. Sur les images avec de nombreux bâtiments visibles les résultats sont très bons mais dès qu'il n'y a qu'un bâtiment majeur dans l'image les résultats chutent. En effet dans un tel cas, une meilleure solution au sens de *MWISR* est

souvent de trouver un quasi pavage de l'image en sous-parties rectangulaires de façade plutôt que de se focaliser sur le meilleur rectangle. Au contraire la résolution gloutonne de ce problème de recouvrement présentée précédemment est très efficace et donne des résultats consistants sur l'ensemble des images testées.

La figure 3.18 montre les courbes du taux de rappel (*recall*) obtenues sur la base de test ZuBuD en fonction du nombre de propositions. On compare la combinaison de nos indices propres aux façades (structure, symétrie and rectification) à la combinaison des indices génériques inspirés de [2] (forme, contour et contraste de couleur) et à la combinaison de tous les indices. Un rectangle vérité-terrain (GT) est compté comme détecté sur au moins une des propositions qui l'intersecte avec une score $s_{IoU} \geq 0.5$. En pratique on n'utilise que 100 propositions ce qui correspond à un taux de rappel de 87.5 %. Nous pouvons ainsi utiliser moins de propositions que les méthodes concurrentes pour des performances similaires dans un temps également comparable (0.31s).

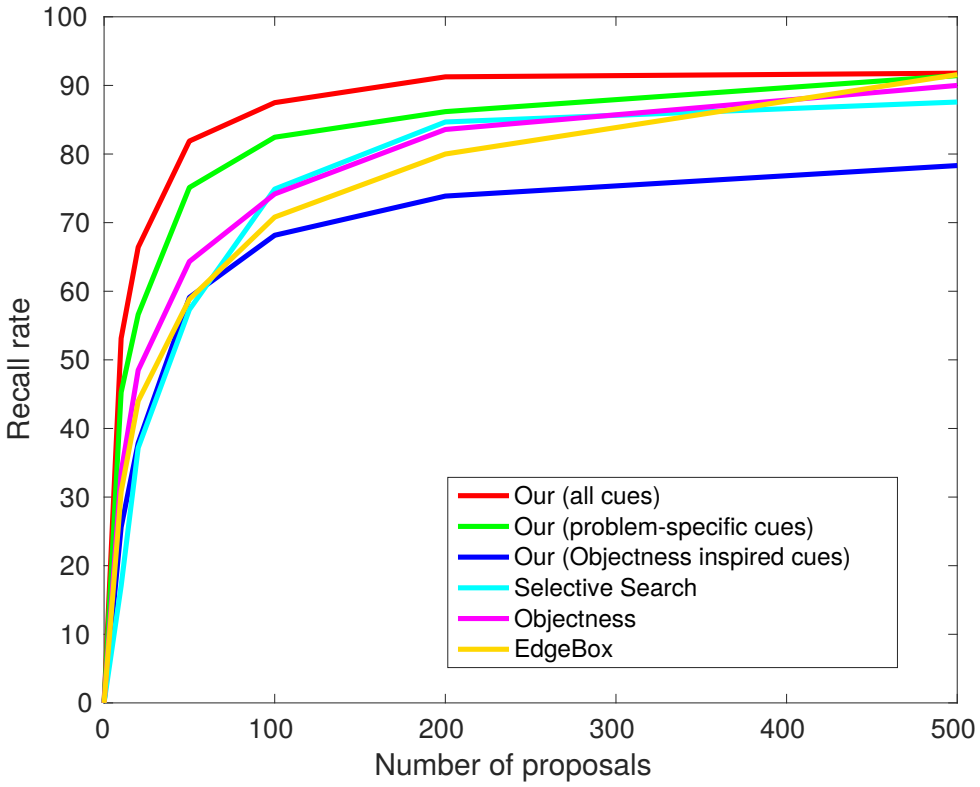

FIGURE 3.18 – Taux de rappel en fonction du nombre de candidats sélectionnés sur la base de test ZuBuD. 3 versions de notre méthode de proposition de façades avec différentes combinaisons d'indices sont montrés : (en rouge) l'ensemble des indices, (en vert) seulement les indices propres aux façades (contour, structure, symétrie et sémantique), (en bleu) seulement les indices inspirés de l'objectness (contour, forme, contraste de couleur). Nous nous comparons à d'autres méthodes de proposition d'objets.

3.3 Détection et reconnaissance de façades

Notre méthode de propositions de façades peut être utilisée pour améliorer [8]. En effet cela pourrait faciliter la génération des hypothèses de translation et résoudre l'ambiguité dans les situations où des façades adjacentes sont sur le même plan dans l'étape de recalage. Cependant nous choisissons de poursuivre la classification jusqu'à obtenir une détection fiable plutôt que simplement des candidats façades probables. On cherche également à ce que ces détections soient reconnues, c'est à dire qu'on soit capable des les identifier précisément en les mettant en correspondance avec des façades de référence d'une base de façades déjà connue. En cela ce problème est apparenté aux problèmes de reconnaissance de lieux, sauf qu'ici on localise aussi grossièrement la façade dans l'image. Notre méthode de détection et de reconnaissance se décompose en trois étapes (Fig. 3.2) :

1. **Générer des candidats façades** par la méthode de proposition de façades précédente.
2. **Classifier ces candidats en "façade" et "non-façade"** par un réseau de neurones utilisant des descripteurs *Spatial Pyramid Pooling* (SPP) [50] tenant compte du contexte alentour comme entrée. Le recours à ces descripteurs permet de ne faire qu'un nombre constant de passages dans le réseau de neurones convolutionnel pour calculer les descripteurs.
3. **Mettre en correspondance les façades détectées avec les façades de référence** en utilisant une métrique apprise par un réseau de neurone Siamois qui prend en entrée les descripteurs SPP.

3.3.1 Classification de façades

La principale singularité de la détection de façades par rapport à la détection d'objets classiques est l'importance du contexte car une façade ne peut être décrite uniquement par son intérieur. En effet, une sous-partie d'une façade peut être tout à fait visuellement identique à une façade complète (par exemple la façade entière amputée d'un étage). L'unique solution pour éviter ce problème de détections multiples de sous-parties est de considérer le contexte visuel alentour. Une véritable façade doit avoir son intérieur qui ressemble à une façade mais son contexte doit s'en distinguer. Nous proposons de construire un descripteur en concaténant le descripteur SPP à l'intérieur du rectangle et le descripteur SPP d'une bande entourant celui-ci. Le descripteur à l'intérieur est calculé sur les cartes de caractéristiques (*feature maps*) de la quatrième couche de convolution (Conv4) de notre version de SegNet pour garder suffisamment d'information de localisation. Pour garantir une plus grande robustesse du descripteur aux changements d'échelles nous choisissons la couche de convolution Conv4 de la sous-image la plus ajustée au rectangle considéré (Fig. 3.5). Pour la structure du descripteur SPP, nous utilisons une pyramide spatiale à trois niveaux (1×1, 2×2, 4×4) pour mutualiser les valeurs des 512 cartes de caractéristiques de Conv4. La résolution de Conv4 (46×60) est cependant trop faible et la bande alentour trop fine (25 % des dimensions du rectangle) pour calculer le descripteur selon le même schéma multi-résolution de mutualisation spatiale (*Spatial Pyramid Pooling*) dans cette bande. Celle-ci peut être divisée en 4 bandes (haut, bas, gauche, droite). Pour chaque bande nous utilisons deux

niveaux de pyramide spatiale (1×1, 1×4). En conséquence notre descripteur SPP à l'intérieur a une dimension de 10752 tandis que celui de la bande totalise 10240 dimensions (Fig. 3.19).

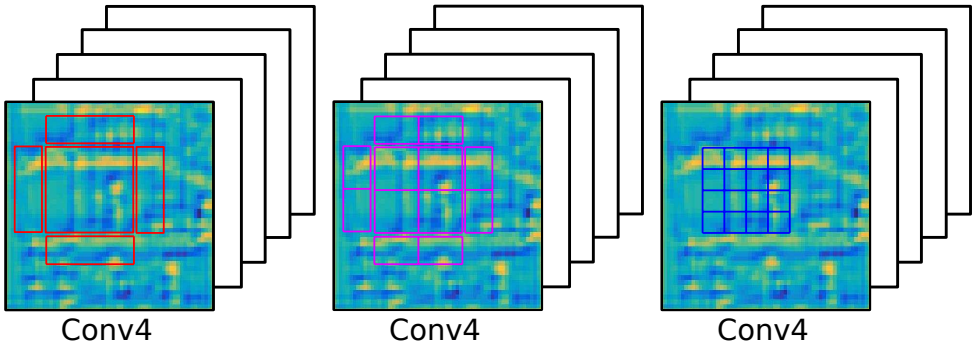

FIGURE 3.19 – Schéma de *Spatial Pyramid Pooling* sur trois niveaux (le premier en rouge, le second en magenta et le dernier en bleu) du descripteur SPP sur la couche Conv4.

Le vecteur concatené de taille 20992 est pris en entrée d'un classifieur par réseau de neurones. Le classifieur est composé de 2 couches cachées complètement connectées de taille 4096. Le réseau est entrainé sur une version augmentée de notre base d'entrainement. En effet nous avons ajouté des données synthétiques à l'ensemble d'entrainement utilisé pour la proposition de façades. Ces données synthétiques supplémentaires sont issues d'images de façades complètes venant d'ImageNet et de Google Street View collées dans des images d'environnements urbains. Les exemples positifs sont générés en prenant les propositions de façades qui coïncident avec la vérité-terrain de façade ($s_{IoU} \geq 0.5$) sur toutes les images de la base d'entrainement (synthétiques ou non). Les exemples négatifs sont générés de la même façon avec un score de coïncidence ($s_{IoU} < 0.5$). Comme il n'est pas garanti qu'il y ait forcément des pixels dans la bande (qui peut être en dehors de l'image rectifiée), un biais d'entrainement peut se produire amenant de nombreuses fausses détections sur les bords de l'image. Pour éviter cela on choisit de répliquer l'image en miroir sur les 4 bords avant la rectification. Cette réplication du signal utilisée classiquement pour les calcul de convolutions nous permet ainsi de gérer les effets de bords problématiques (voir 3.20).

3.3.2 Mise en correspondance de façades

A ce niveau, le descripteur associé à chaque façade détectée est le descripteur SPP interne qui a déjà été calculé à l'étape précédente.

Cependant distinguer deux façades différentes tout en étant robuste aux changements d'apparence d'une même façade peut être un challenge compliqué. Ce problème est similaire à celui de la classification à grain fin (*fine-grained classification*) qui se résout classiquement par l'apprentissage d'une métrique de similarité spécifique. C'est cette approche que nous suivons ici en utilisant un réseau de neurones siamois [23]. L'idée est de trouver une fonction qui envoie les images dans un espace de dimension faible où la distance euclidienne entre points de la même classe est faible et celle entre points de classes différentes importante.

Ici, des paires de descripteurs SPP internes (a_n, b_n) sont utilisées comme entrées d'un réseau

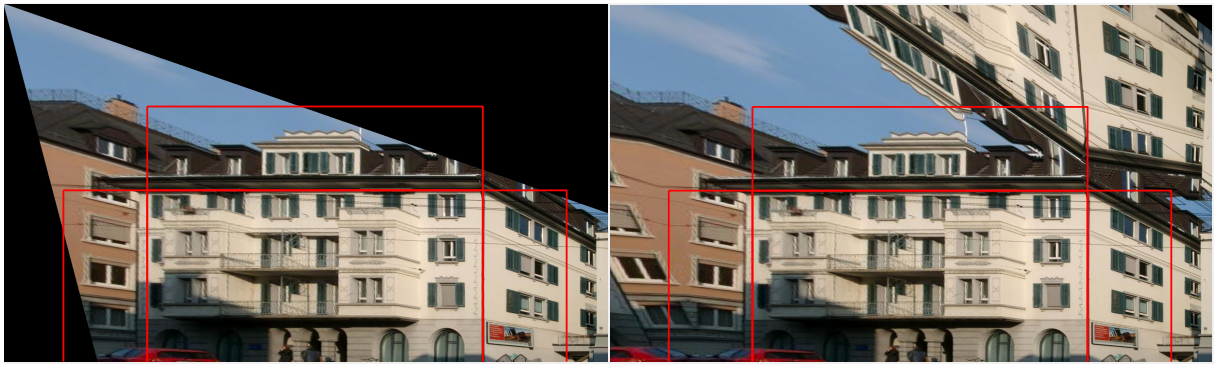

FIGURE 3.20 – A gauche : image rectifiée et zones concernées par la détection contextuelle. A droite : résultat de la réplication miroir sur l'image rectifiée. Les zones contextuelles sont moins susceptibles de biaiser l'apprentissage.

de neurones Φ à deux couches cachées de taille 2048 entrainé en mode siamois [23] sur un tiers de la base ZuBuD avec la fonction de cout suivante :

$$L = \frac{1}{N}\sum_{n=1}^{N} yd^2 + (1-y)\max(m-d,0)^2 \tag{3.6}$$

avec $d = \|\Phi(a_n) - \Phi(b_n)\|_2$, m la marge négative, et $y = 1$ si la paire est positive $y = 0$ sinon

Les paires positives sont générées à partir des candidats positifs ($s_{IoU} \geq 0.5$) des différentes vues d'une même façade tandis que les paires négatives sont générées à partir de candidats positifs de façades différentes. L'espace induit par le réseau de neurones siamois est ainsi de plus petite taille et ajusté pour distinguer des façades qui peuvent sembler proches visuellement. Nous calculons ensuite la distance euclidienne entre les sorties de ce réseau des détections et celles des façades de référence connues. Pour chaque détection nous choisissons le plus proche voisin dans la base de référence. Pour nous assurer que cette mise en correspondance est correcte nous imposons que l'association par plus proche voisin soit réciproque.

3.4 Résultats expérimentaux

3.4.1 Résultats concernant la proposition de façades

Pour les tests nous utilisons la Zurich Buildings Database (ZuBuD). Cette base de données est composée de 1000 vues piétonnes de rues de Zurich. Elle est divisée en 200 scènes, chacune se concentrant sur un bâtiment particulier. Les changements de points de vue sont importants (Fig. 3.21, première ligne) et il n'est pas rare que les façades soient occultées par un arbre (Fig. 3.21, deuxième ligne), du mobilier urbain ou des lignes électriques (Fig. 3.21, deuxième ligne). Elle présente également une grande diversité d'architectures européennes classiques mais aussi des styles plus modernes. Pour toutes ces raisons nous évaluons notre méthode de propositions de façades sur cet ensemble. Pour cela nous avons tout d'abord constitué une verité terrain qui correspond au placement manuel des limites de chacune des façades visibles dans les images de

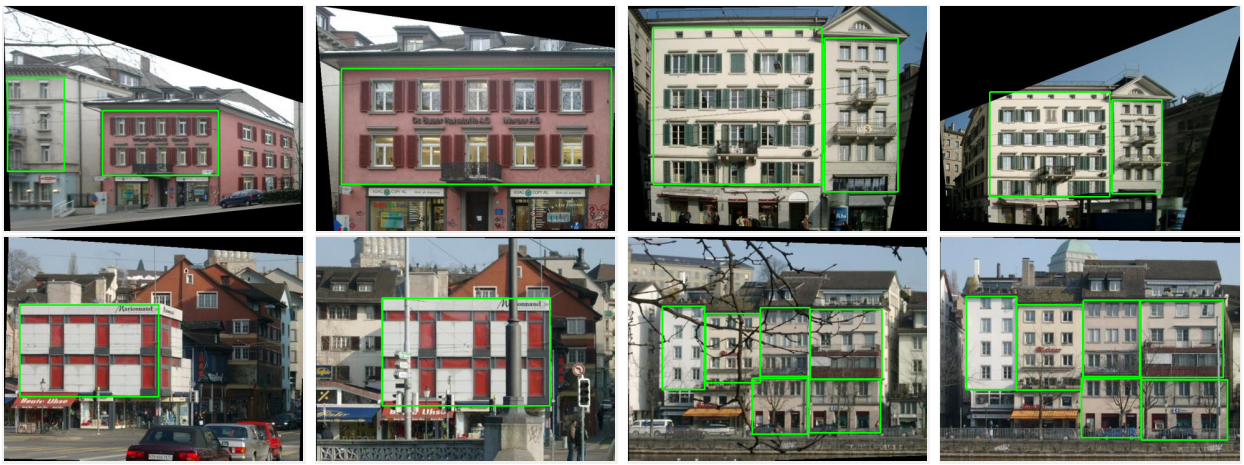

FIGURE 3.21 – Examples d'images de la base ZuBuD rectifiées. La vérité terrain de la détection de façade est en traits verts.

la base. Nous nous comparons également à d'autres méthodes de propositions d'objets selon 3 perspectives différentes : le taux de rappel, la précision et le temps de calcul (Tab. 3.1).

Le taux de rappel est certainement l'une des mesures les plus importantes pour les méthodes de proposition d'objets. Celui-ci est calculé comme le taux d'objets recouverts par au moins une des n premières propositions. Si la plupart des méthodes génériques de propositions d'objets présentent de bons résultats sur ZuBuD au delà de $n = 500$ propositions avec dans ces cas plus de 90 % de rappel, notre méthode montre de meilleurs résultats avec beaucoup moins de propositions. En effet avec seulement $n \leq 100$ propositions, notre méthode présente un gain constant de plus de 10 % comparé aux méthodes de propositions d'objets de l'état de l'art (voir Fig. 3.18).

Cette amélioration peut s'expliquer par l'utilisation d'indices qui sont plus discriminants pour les façades que pour des objets génériques. Plus précisément, certaines hypothèses qui sont faites par des méthodes de propositions d'objets génériques sont violées dans les environnements urbains. Selective Search [110] est basée sur une fusion de super-pixels. Comme le contexte local n'est pas considéré, il n'y a pas de moyen de distinguer des sous-parties de façades des façades complètes ce qui mène à une chute du taux de rappel avec peu de propositions. La principale hypothèse de EdgeBoxes [122] est que les contours sont entièrement inclus à l'intérieur des propositions. Dans le cas de façades adjacentes, il y a bien souvent des contours communs entre les façades qui intersectent les rectangles candidats. Dans l'Objectness [2] la manière de générer les propositions en première étape repose sur des anomalies dans la distribution fréquentielle des images naturelles. Ceci n'est pas adapté aux environnements urbains dans lesquels la distribution fréquentielle est déjà biaisée par les contours verticaux très présents et les nombreuses répétitions de motifs. Enfin aucune de ces méthodes ne prend en compte la rectification de l'image quand nos indices de symétrie et de structure utilisent cette information. Toutes ces hypothèses que font les méthodes génériques de propositions d'objets sont finalement peu adaptées pour les environnements urbains ce qui affecte le classement des propositions. Plus particulièrement, les

TABLE 3.1 – Statistiques sur les propositions de façades

	Selective Search	Objectness	EdgeBox	Nous
Rappel ($n = 100$)	74.89	74.18	70.81	87.49
Précision (s_{IoU})	0.59	0.63	0.61	0.69
Temps (secondes)	0.28	1.42	0.35	0.31

candidats rectangles qui fusionnent des façades adjacentes sont souvent classés en premier par les méthodes génériques ce qui relègue les propositions de façades correctes en queue de classement. Ainsi davantage de propositions sont nécessaires pour effectuer des tâches qui dépendent de ces propositions (comme la détection et la reconnaissance) alors que $n = 100$ propositions suffisent pour notre méthode.

La précision est calculée ici comme la valeur moyenne des scores s_{IoU} de toutes les propositions qui coïncident avec la vérité-terrain ($s_{IoU} \geq 0.5$) sur les 100 premières propositions. Cette mesure n'est pas critique pour la proposition d'objets mais l'est nettement plus pour la détection de façades en vue d'initialisation du calcul de pose de caméra. Nos meilleurs résultats en précision par rapports aux méthodes concurrentes peuvent s'expliquer par l'utilisation d'indices basés sur les contours le long des frontières de la façade ainsi que des indices contextuels.

Une autre mesure qui importe réellement pour la proposition d'objets est le temps. Tous nos indices sont calculés en temps constant grâce aux intégrales images et une mutualisation locale (*Spatial Pooling*). Cela permet d'avoir un temps de calcul qui est meilleur ou du même ordre que les approches concurrentes. Ce temps est déjà compatible avec des applications de réalité augmentéee et pourrait facilement bénéficier d'une parallélisation. Le code est écrit en Matlab avec les parties critiques exécutées en C. Les temps de calculs montrés sur le tableau Tab. 3.1 sont les temps moyens pour $n = 100$ propositions sur un processeur I7-3520M avec une carte graphique Nvidia TITAN X pour le passage dans le réseau SegNet.

3.4.2 Résultats relatifs à la reconnaissance de façades

Nous effectuons les tests de détection et de reconnaissance sur les 3/4 de ZuBuD qui n'ont pas été utilisés pour l'apprentissage de métrique (voir Sec. 3.3.2). Cela consiste en 937 façades dont la vérité terrain de leurs frontières a été délimitée manuellement. Ces façades sont groupées en 171 classes, chacune représentant la même façade d'un même bâtiment. Chaque classe rassemble des images de la même façade observée sous différents points de vue. En conséquence les images de la même classe présentent des artefacts de rectification différents et différentes résolutions (Fig. 3.22). A cause des occultations et de la non visibilité de l'intégralité de la façade certaines façades sont rognées. En plus de toute cette diversité intra-classe les différentes classes peuvent être très proches visuellement. Pour représenter l'ensemble de la classe d'un bâtiment nous choisissons une référence comme l'image la plus proche de la vue frontale dont l'intégralité est visible sans avoir d'occultations.

Ces conditions extrêmes mettent en défaut les approches de reconnaissance classiquement qui échouent à identifier la référence correctement (Tab. 3.2). Nous évaluons les performances de notre méthode avec des méthodes classiques utilisées en reconnaissance de lieux. On rappelle

FIGURE 3.22 – Illustration de la diversité des façades qui appartiennent à la même classe avec des artefacts de rectification, des occultations et des observation partielles.

qu'un des objectifs de ces méthodes est de trouver le lieu (ou ici la façade) rapidement parmi une base connue. Cette contrainte de temps exclut donc les méthodes de mise en correspondance où la combinatoire est importante (RANSAC par exemple). Pour chacune des différentes approches nous exécutons d'abord notre méthode de propositions de façades avec $n = 100$ candidats. Pour chacune des vérités terrains de façade de la base de tests, nous sélectionnons le candidat qui coïncide le mieux avec celles-ci (au sens de s_{IoU}). Ainsi la façade détectée est la meilleure que l'on puisse espérer trouver à partir de nos propositions indépendamment de la performance de notre étape de classification. Pour chacune de ces façades « parfaitement » détectées, on calcule leur descripteur et on cherche leur plus proche voisin parmi les 171 références selon la distance euclidienne. L'identification est considérée correcte si la classe de la plus proche référence est la même que la classe de la vérité-terrain de l'image considérée. Le tableau 3.2 montre les résultats de différents descripteurs utilisés en reconnaissance de lieux. BoW (Sac de Mots en anglais *Bag of Words*) est un descripteur par histogramme de mots visuels (SIFT) à l'intérieur du candidat façade. Le vocabulaire utilisé par ce descripteur est composé de 10000 mots appris sur le tiers d'apprentissage de ZuBuD. VGG est le vecteur de 4096 dimensions de sortie du réseau de neurones convolutionnels VGG [102] appliqué à la sous-image de la proposition redimensionnée à la taille d'entrée du réseau. SPP est le descripteur par schéma multi-résolutions de mutualisation spatiale calculé à partir de la couche Conv4 de notre version de SegNet et SPP (*siamese*) la sortie du réseau siamois de ce même vecteur qui correspond à la surcouche d'apprentissage de métrique.

TABLE 3.2 – Statistiques sur la reconnaissance de façades

	BoW	VGG	SPP (SegNet)	SPP (siamese)
Correspondances correctes (%)	44.06	72.80	79.96	84.48
Temps (secondes)	0.29	2.19	0.07	0.05

Le peu de points clés SIFT extraits des façades et la similarité de leurs descripteurs sur des structures répétitives peut expliquer les mauvais résultats de l'approche par Sac de Mots. De plus sans information spatiale celle-ci ne peut discriminer les façades que sur leur proportions de mots visuels ce qui n'est clairement pas suffisant. La différence entre VGG et SPP est sûrement imputable à l'ajustement fin (*Fine-Tuning*) de notre version du réseau SegNet pour la segmen-

tation de bâtiments. En effet les architectures des deux réseaux sont très voisines et SPP est essentiellement une approximation rapide d'un véritable descripteur CNN de sortie de réseau. La couche Conv4 de notre réseau contient ainsi davantage d'informations intéressantes sur les façades que la version pré-entrainée de VGG sur ImageNet. Enfin la surcouche d'apprentissage de métrique par réseau siamois aide à la classification fine comme attendu.

Nous évaluons maintenant l'ensemble de la méthode en incluant l'étape de détection. Les résultats visuels sont montrés sur la Fig. 3.23. A la fin de l'étape de propositions de façades nous considérons 100 façades candidates. Sans appliquer ensuite l'étape de classification (détection), il ne serait pas possible de conclure sur la reconnaissance de façades. En effet cette ultime étape étant basée sur un appariement au plus proche voisin, les nombreuses fausses détections encore présentes à l'étape de propositions seraient forcément mises en correspondance avec des façades références. La méthode reconnaitrait alors des façades qui ne sont pas présentes dans l'image. Il est donc nécessaire de réduire les fausses détections qui sont souvent des sous-parties de façades parmi les 100 propositions.

La classification de façades basée sur l'intérieur et la région environnante supprime la plupart de ces candidats et la validation réciproque assure que la mise en correspondance est correcte. Ainsi nous avons peu de fausses détections (14.11 %) avec un taux de rappel élevé (75.02 %) pour l'ensemble de la base de test. Ces résultats surpassent l'approche par Sac de Mots et sont comparables aux résultats de VGG et cela même si dans ces deux cas les résultats étaient biaisés de manière à avoir une détection « parfaite » parmi les propositions. Par ailleurs les fausses détections ne veulent pas forcément dire que la détection a complètement échoué. Cela peut souvent être expliqué par une des situations suivantes : la façade détectée est trop petite mais tout de même incluse dans la vérité terrain (façade bleue à gauche de l'image 1 dans Fig. 3.25), la façade détectée rassemble plusieurs façades vérité-terrain en une seule (façade violette de l'image 3 dans Fig. 3.25) la façade détectée a été oubliée pendant l'étiquetage manuel de l'image (façade bleue à gauche de l'image 1 dans Fig. 3.25). Cependant nous avons noté des cas d'échec réguliers qui apparaissent lorsqu'une petite façade détectée avec peu d'information photométrique est mise en correspondance avec une façade de référence quasi-homogène (façade bleue au milieu de l'image 1 dans Fig. 3.25). Cette mauvaise identification peut également arriver quand deux façades sont très similaires dans la base de références. Par exemple, cette situation peut se passer pour deux façades qui viennent de façades perpendiculaires du même bâtiment (image 4 dans Fig. 3.26).

Nous évaluons également la méthode sur un ensemble plus petit qui vient de la partie Rue (Street) du jeu de données Cambridge Relocalisation Dataset [6]. Ce jeu de données est composé de 80 images divisé en 20 classes. Contrairement à ZuBuD il n'y a pas d'occultations mais les changements de points de vue sont plus extrêmes. Les résultats quantitatifs sont très proches de ceux obtenus sur ZuBuD avec 73.88 % de rappel et 16.23 % de fausses détections. Ainsi notre méthode démontre sa robustesse aux changements de points de vue sévères autant qu'aux observations partielles et aux occultations (Fig. 3.24).

Un autre avantage de notre méthode de détection et de reconnaissance de façades est la

6. http ://mi.eng.cam.ac.uk/projects/relocalisation/#dataset

vitesse. Comme nous utilisons des descripteurs SPP nous avons seulement besoin d'un nombre constant de passages (10 avec notre approche multi-résolutions) dans notre version modifiée de SegNet pour l'ensemble de la méthode. Si les changements de points de vue ne sont pas trop importants et que les échelles sont globalement préservées un seul passage peut suffire. Nous recyclons les sorties de différentes parties de l'algorithme (dans la génération de candidats, le calcul des indices,...) et nous réexploitons la couche Conv4 du réseau pour le calcul des descripteurs. Notre méthode peut être vue comme une initialisation de la pose de caméra en prévision d'un raffinement ([59, 93] ou chapitre suivant). En tant que procédé d'initialisation il n'est pas nécessaire de l'exécuter à chaque nouvelle trame mais seulement au début et quand le suivi est perdu. Le temps de calcul de la méthode (0.45s) est compatible avec des applications de réalité augmentée (AR) car une nouvelle détection peut être calculée toutes les 11 trames. Nous pourrions également imaginer que les détections seraient faites côté serveur tandis que la partie suivi elle seule serait traitée en temps réel sur l'appareil mobile. Bien que le manque de puissance GPU soit encore une limite de notre méthode pour être exécutée sur un appareil mobile standard, nous pensons que les prochaines générations en seront capables (l'inférence de SegNet sur une carte graphique embarquée Nvidia TX1 requiert 0.7s contrairement aux 0.1s de notre installation de bureau). Cette tendance semble être confirmée par les constructeurs de smartphones et de tablettes qui équipent de plus en plus leurs appareils d'unités graphiques dédiées. Cela va de pair avec le développement par les grands acteurs du web de logiciels pour l'utilisation de réseaux de neurones profonds optimisés pour mobiles comme Facebook avec Caffe2Go et Google avec Tensorflow Lite.

3.4.3 Applications à la réalité augmentée et au calcul de pose

3.4.4 Réalité augmentée

Une fois les façades détectées et identifiées, il est déjà possible d'y superposer des objets planaires virtuels dans un contexte de réalité augmentée. En effet il suffit pour cela de retro-projeter les frontières de la façade détectée dans l'image originale en utilisant l'homographie inverse de celle qui a permis la rectification. Par exemple dans la figure 3.23 chaque couleur représente une façade qui a été détectée et reconnue avec succès parmi les façades de référence de la base de données. Ces zones pourraient facilement être remplacées par des informations pertinentes sur les bâtiments (Fig. 3.27). En effet la superposition d'objets virtuels planaires est une part importante des applications de réalité augmentée en milieu urbain. Elle peut servir à afficher sur les bâtiments des anecdotes dans un cadre touristique, des publicités ciblées pour les magasins ou bien les adresses dans un contexte d'aide à la navigation en ville. Ici la texture d'un modèle 3d public [7] de la Cité des Congrès de Nantes a été ajoutée aux façades de référence de la base de test ZuBuD. La méthode a permis de reconnaitre et détecter cette même façade dans des images issues de Google Street View. Sur ces images a alors été superposé le logo de la conférence ISMAR dont l'édition avait lieu dans ce même bâtiment.

7. http ://www.3dwarehouse.sketchup.com

FIGURE 3.23 – Examples de résultats de la méthode de détection et reconnaissance obtenue sur ZuBuD. Les polygones verts représentent la vérité-terrain et les zones colorées les détections de façades reprojetée dans l'image non-rectifiée. Toutes les façades ici ont été reconnues correctement.

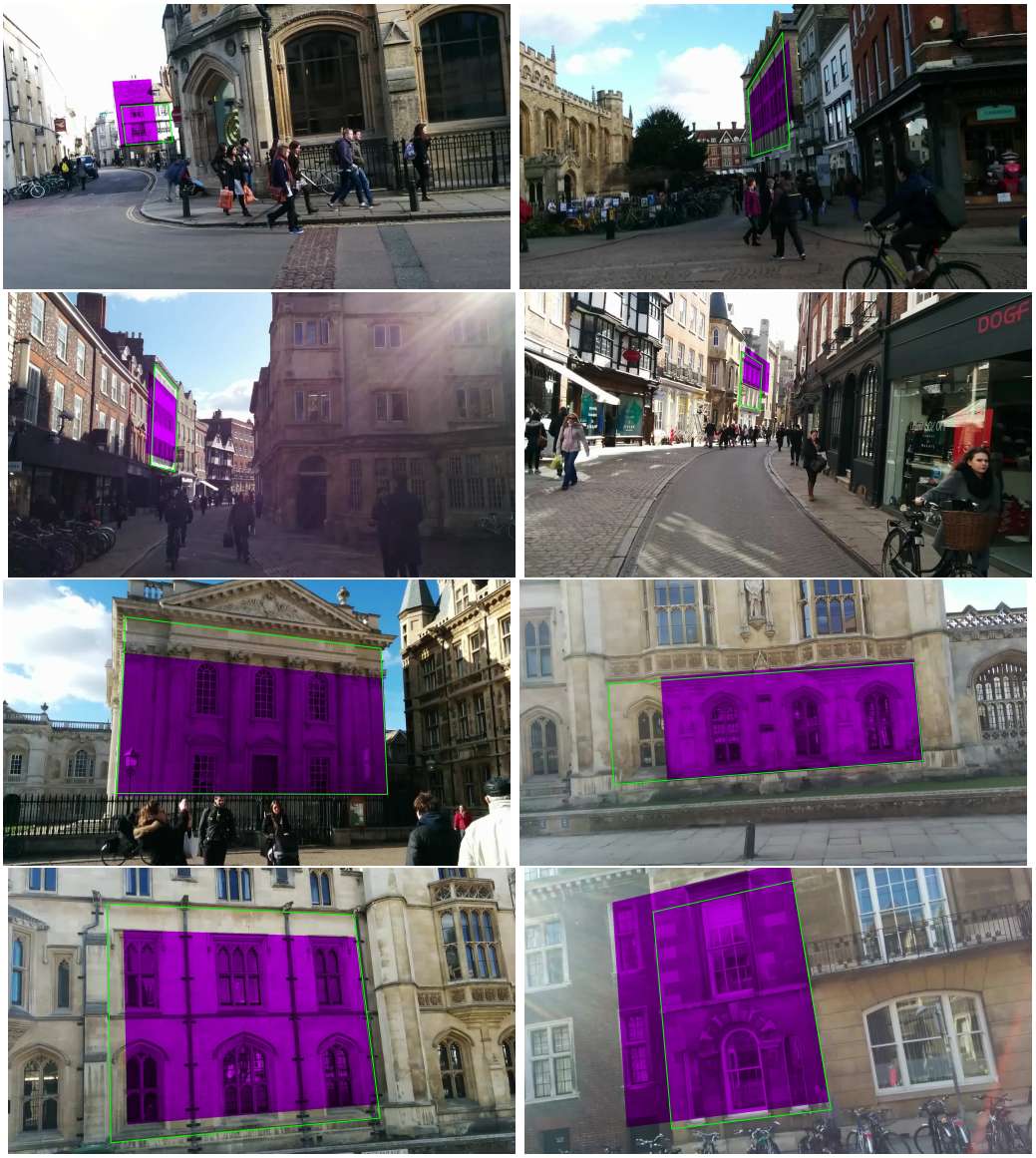

FIGURE 3.24 – Examples de résultats de la méthode de détection et reconnaissance obtenue sur Cambridge. Les polygones verts représentent la vérité-terrain et les zones colorées les détections de façades reprojetée dans l'image non-rectifiée. Toutes les façades ici ont été reconnues correctement.

FIGURE 3.25 – Exemples d'échecs dans différents cas de fausses détections. L'image 1 et 2 montrent par de vraies fausses détections respectivement en rouge et en violet. La détection bleue de l'image 1 à gauche est également considérée comme fausse détection car elle est trop petite par rapport à la vérité terrain en traits verts. Sur l'image 3, la détection violette regroupe deux façades de la vérité terrain. Enfin la détection bleue de l'image 4 est une fausse détection consécutive à un oubli de vérité terrain.

3.4.5 Initialisation de pose de caméra

Néanmoins, d'autres applications nécessitent d'inférer les 6 degrés de liberté de la pose de la caméra. Il peut s'agir de localisation en robotique mobile ou de réalité augmentée avec du rendu d'objets 3d virtuels (Fig. 3.29). On peut pour cela enrichir la base de données de façades de référence avec de l'information géométrique (le géo-référencement des façades ainsi que leurs dimensions réelles). Cela revient à considérer que l'on dispose d'un modèle 3d simple (les bâtiments y sont des parallélépipèdes rectangles) texturé de la ville (Fig. 3.28). Supposons qu'un tel modèle est disponible (via Google Street View par exemple) et qu'une façade de ce modèle a été détectée et reconnue dans l'image courante. L'étape de rectification du chapitre précédent nous donne déjà la rotation R qui transforme le repère caméra en le repère local de cette façade. Seule la translation T entre ces deux repères reste à calculer. Il suffit d'associer les coins du rectangle de détection dans l'image rectifiée (u_1, v_1),(u_2, v_2),(u_3, v_3) et (u_4, v_4) à ceux de la façade du modèle 3d exprimés son repère local $(0, 0, 0)$, $(h, 0, 0)$,$(0, w, 0)$ et $(h, w, 0)$. Ainsi pour chaque coin x de l'image rectifiée associé au point X du modèle 3d on peut écrire :

FIGURE 3.26 – Exemple d'échec de la mise en correspondance. La détection violette est correctement reconnue par mise en correspondance avec la façade de référence de la colonne droite en haut. La détection rouge en revanche est aussi mise en correspondance de cette même façade de référence alors qu'elle devrait l'être avec celle de la colonne droite en bas.

FIGURE 3.27 – A gauche, une des façades de référence de la Cité des Congrès de Nantes. Celle-ci est automatiquement détectée et reconnue dans deux vues de ce même bâtiment (polygones rouges). A partir de ces résultats on a ajouté le logo de conférence ISMAR déformé conformément à l'angle de vue du bâtiment.

$$\begin{aligned} x &\propto KR^T K^{-1} K \left(RX + T\right) \\ &\propto K \left(X + R^T T\right) \end{aligned} \tag{3.7}$$

On peut alors en tirer un système linéaire surdéterminé en prenant le produit vectoriel dans l'équation 3.7. Ce système est alors résolu par moindres carrés :

$$\begin{pmatrix} 0 & f & -u_1 \\ -f & 0 & v_1 \\ u_1 f & -v_1 f & 0 \\ 0 & f & -u_2 \\ -f & 0 & v_2 \\ u_2 f & -v_2 f & 0 \\ 0 & f & -u_3 \\ -f & 0 & v_3 \\ u_3 f & -v_3 f & 0 \\ 0 & f & -u_4 \\ -f & 0 & v_4 \\ u_4 f & -v_4 f & 0 \end{pmatrix} T' = \begin{pmatrix} 2o_y \\ -2o_x \\ v_1 o_x - u_1 o_y \\ 2o_y \\ fh - 2o_x \\ v_2 fh + v_2 o_x - u_2 o_y \\ fw + 2o_y \\ -2o_x \\ u_3 fw + v_3 o_x - u_3 o_y \\ fw + 2o_y \\ fh - 2o_x \\ u_4 fw - v_4 fh + v_4 o_x - u_4 o_y \end{pmatrix} \quad \text{et} \quad T = RT' \qquad (3.8)$$

f est la longueur focale et (o_x, o_y) le centre optique de la caméra. Ces paramètres intrinsèques de la caméra sont supposés connus.

Les dimensions réelles sont importantes pour avoir une échelle commune entre la façade détectée et les objets virtuels qu'on voudrait leur associer. C'est également fondamental dans le cadre de la navigation mobile pour évaluer les distances. Le géo-référencement des façades du modèle permet de transformer la pose de camera relative (R, T) en une pose absolue pour la géolocalisation.

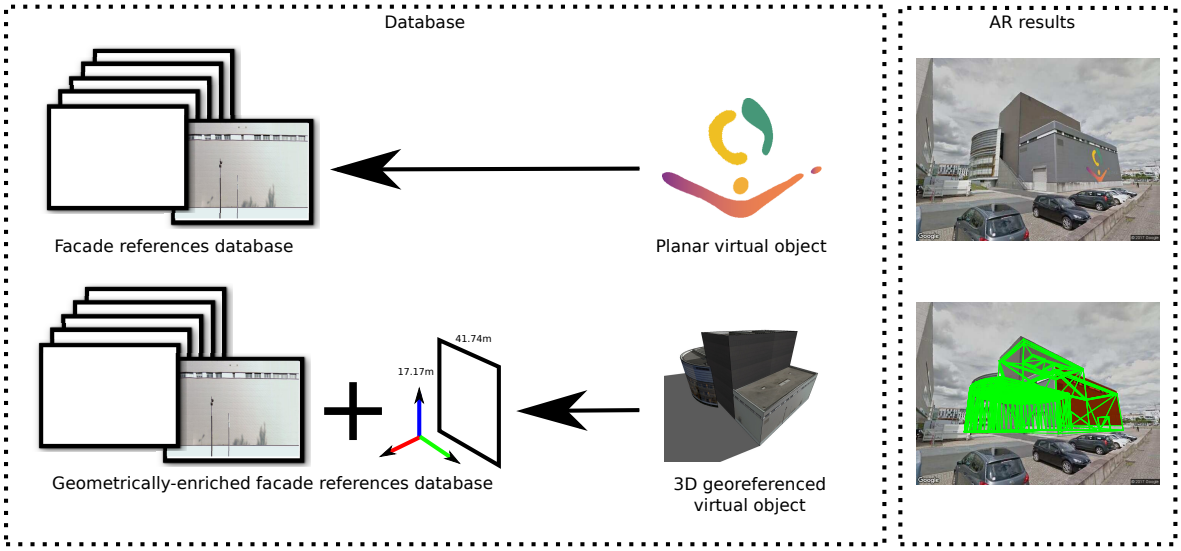

FIGURE 3.28 – Applications possibles de notre méthode à la réalité augmentée. Le premier rang montre le cas d'une base de données de références constitué uniquement d'image de façades et de leurs objets virtuels planaires associés. Le second rang montre le cas d'une base de données enrichie géométriquement avec un modèle 3d simple dont une version fil-de-fer est projetée dans l'image.

Pour illustrer cette initialisation de pose de caméra, on réutilise l'exemple du bâtiment de la

Cité des Congrès de Nantes dont on dispose d'un modèle 3d texturé. Chacune de ses façades a été ajoutée à la base de test et on applique notre méthode de détection et de reconnaissance sur des vues piétonnes de ce bâtiment. Sur toutes ces images la façade a été détectée (à chaque fois parmi les 10 premières propositions) et reconnue comme appartenant au bâtiment de la Cité des Congrès. La pose relative a alors été calculée et un rendu fil-de-fer du modèle 3d du bâtiment a été superposé à l'image courante par projection.

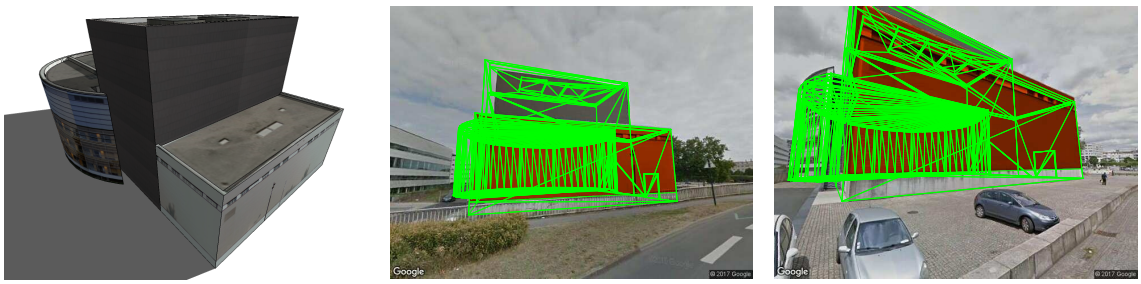

FIGURE 3.29 – A gauche, le modèle 3d texturé de la Cité des Congrès de Nantes. Une façade de ce modèle est automatiquement détectée et reconnue dans deux vues de ce même bâtiment (zones rouges). La pose de caméra calculée a été calculée à partir de ces résultats et on a reprojeté une version fil-de-fer du modèle dans ces vues.

3.5 Conclusion

Nous avons présenté une méthode de détection et de reconnaissance de façades qui repose sur une première étape de proposition de façades. Pour cette étape, nous avons adapté trois indices issus de [2] aux spécificités des milieux urbains. Nous avons également proposé trois nouveaux indices (structure, symétrie et rectification) pertinents vis-à-vis de la structure sémantique et géométrique particulière des façades. Tous ces indices répondent à la contrainte de temps de calcul réduit attendue d'une méthode de proposition d'objets. 100 propositions triées selon un score combinant ces indices suffisent à détecter plus de 87% des façades de ZuBuD, ce qui dépasse largement les performances des méthodes de propositions d'objets génériques de l'état de l'art. S'ensuit une étape de classification forte basée sur des descripteurs CNN qui tiennent compte du contexte local de la façade. Ces mêmes descripteurs sont utilisés en entrée d'un réseau siamois pour reconnaitre les façades détectées parmi une base de références connues. Ces deux dernières étapes pourraient bénéficier d'une description additionnelle moins dense et donc plus robuste aux parties cachées sous la forme du graphe sémantique décrit en 3.11. Ce graphe pourrait être classifié par une approche profonde (MoNet [87]) pour la détection mais également servir pour un test de compatibilité géométrique lors de la mise en correspondance avec la base de références. Quoiqu'il en soit la méthode actuelle montre des bons résultats ($\approx$ 70% de rappel et $\approx$ 15% de fausses détections) sur des jeux de données urbains complexes. La rapidité de la méthode la rend compatible autant avec des applications de réalité augmentée simples qu'avec des applications plus complexes nécessitant une initialisation de calcul de pose de caméra.

Chapitre 4

Segmentation et recalage de façade conjoint

Dans le chapitre précédent la détection et la reconnaissance de façades permettaient d'initialiser le recalage d'une façade de référence connue dans une nouvelle image cible. On cherche ici à raffiner cette première estimation grossière en s'aidant de la segmentation sémantique de l'image. Si celle-ci est directement disponible à partir du chapitre précédent, des erreurs de classification demeurent qui pourraient perturber un recalage qui s'appuierait sur elle. Si une meilleure segmentation améliorerait le recalage, le recalage lui même permettrait d'améliorer la segmentation en profitant de la géométrie particulière de l'architecture d'une facade. Ces deux problèmes étant liés, on cherche à les résoudre conjointement. Nous introduisons notamment ici un modèle bayesien qui lie explicitement la segmentation sémantique de la facade cible aux structures architecturales de la facade de référence modélisées par une mélange de Gaussiennes L_p. Ce modèle gère également le bruit de contexte de l'image cible ainsi que les éventuelles occultations. L'inférence est conduite efficacement par un algorithme d'Espérance-Maximisation dont l'étape de maximisation est résolue partiellement sous forme analytique.

4.1 Travaux liés

4.1.1 Segmentation sémantique de façades

Comme on l'a vue en introduction générale, les méthodes récentes de recalage en milieux urbain tendent à utiliser des éléments de sémantique de haut-niveau (étiquettage sémantique, cartes de normales, séparations de façades,...) qui sont inférés par réseaux de neurones convolutionnels. L'avantage principal est que cela permet de découpler les problèmes de géométrie et d'apparence. Les changements d'apparence des images (points de vue, illumination, ...) sont pris en compte par les réseaux tandis que le calcul de pose se focalise sur ces composantes haut-niveau de l'image réduisant les minima locaux possibles.

La précision et l'efficacité de ces réseaux ne sont plus à démontrer pour la segmentation sémantique mais leur généricité ne permet pas d'éviter des erreurs de classification dont l'am-

biguïté visuelle est telle qu'elle ne peut être levée que par une approche plus globale et plus spécifique (par exemple pour distinguer une porte d'une fenêtre). Les formes particulières des éléments architecturaux ne sont pas non plus garanties même si cela peut être renforcé en introduisant l'inférence de contours comme dans le chapitre précédent. Ces erreurs pouvant perturber le recalage, on a donc tout intérêt à avoir la segmentation sémantique la plus propre possible.

Il existe des méthodes qui cherchent à améliorer la segmentation sémantique en profitant de la régularité très spécifique des façades. Parmi celles-ci certaines sont « Bottom-Up » mais il y aussi des approches « Top-Down ». Parmi les approches « Bottom-Up » qui tirent partie de la structure particulière des bâtiments, Yang et al. [117] utilisent TILT [120] pour rectifier la façade ce qui permet de récupérer une texture de façade nettoyée des occultations en sous-produit. En effet dans TILT, l'image courante I supposée transformée par τ est modélisée par une matrice de rang faible I_r polluée par un bruit épars E, $I \circ \tau = I_r + E$. L'optimisation vise alors à trouver toutes ces composantes en minimisant le problème relaxé sous-contrainte : $\text{argmin}_{I_r,E,\tau} \|I_r\|_\star + \gamma \|E\|_1$. De cette image régularisée I_r sont extraits des descripteurs bas-niveau (HOG, textons) qui sont classifiés par une forêt aléatoire. Cette classification initiale est ensuite partitionnée par une heuristique de division/fusion en plusieurs blocs de matrice de rang 1 (Fig. 4.1). Comme pour TILT, l'approximation de rang 1 par bloc est la solution d'un problème de minimisation par Multiplicateur de Lagrange Augmenté. Le même genre de couple descripteur bas-niveau/arbre de décision est utilisé dans [41] pour la classification initiale. La résultat de la classification est raffiné itérativement en ajoutant à l'image des descripteurs *auto-context* calculés sur l'état précédent de la segmentation. Ces descripteurs cherchent à renforcer les formes rectangulaires ou les répétitions verticales et horizontales le long de la façade.

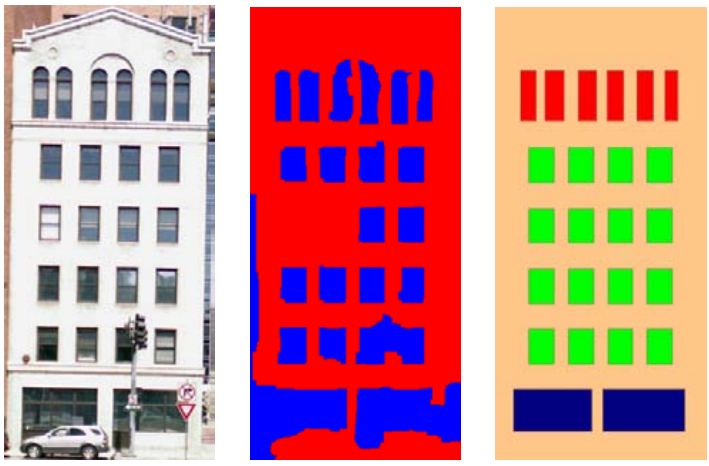

FIGURE 4.1 – A gauche une façade, au milieu sa segmentation sémantique inférée, à droite cette même segmentation sémantique régularisée par approximation de rang 1. Image issue de [117].

Les approches « Top-Down » visent à proposer un modèle génératif des ces *a priori* architecturaux. Il s'agit de décomposer une segmentation par une série de lois structurelles paramétriques appelée une grammaire de forme (*Shape Grammar*) souvent représentée par un arbre (Fig. 4.2). La très grande organisation structurelle des façades qui trouve son origine dans des règles ar-

chitecturales en fait un bon candidat de ce genre de modèle. Teboul et al. [107] utilisent des techniques d'apprentissage par renforcement comme le *Q-learning* sur un modèle de processus de décision markovien pour construire l'arbre optimal de segmentation. Dans [64] Kozinski et al. utilisent un formalisme voisin et définissent la grammaire comme un partionnement hiérarchique en grilles. Cela permet de gérer les alignements 2D de structures sur la façade plus efficacement. Le problème d'optimisation de l'image selon cette grammaire est formulé sous la forme d'un champ de Markov sur une grille de pixels 4-connectés résolu par Maximum *A Posteriori*.

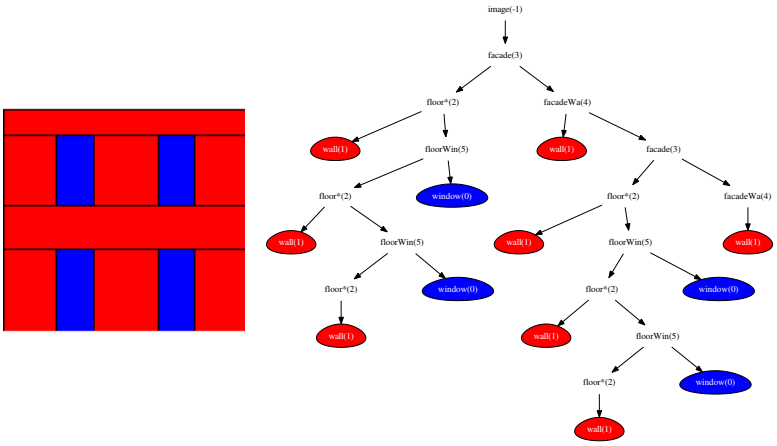

FIGURE 4.2 – Description d'une grammaire de façade représentée par un arbre. Image issue de [107].

[81][88] sont des méthodes hybrides qui cherchent à améliorer le bas-niveau par une approche « Bottom-Up » pour simplifier la résolution « Top Down » que ce soit par la résolution d'un champ aléatoire conditionnel (*Conditional Random Field*) qui utilise la détection d'objets dans le premier cas ou par l'exploitation de la répétition et de la symétrie des façades à différents niveaux du processus markovien dans le second.

Dans ces travaux de segmentation spécifiques aux façades, la régularisation est conditionnée par le fait que seule la façade est présente dans l'image. Cette hypothèse n'est pas directement compatible avec le problème du recalage qui cherche justement à trouver la transformation entre une façade de référence et une façade contenue dans une image de plus large contexte. De plus le surcoût de complexité de ces méthodes est très important ce qui disqualifie ces améliorations possibles de la sémantique pour le recalage en temps réel. En fait on peut remarquer que les problèmes de segmentation sémantique et de recalage sont liés. Certes une meilleure segmentation sémantique rend plus précis le recalage qui l'utiliserait. Mais dans un schéma itératif, lorsque le recalage est proche de la solution optimale, celui-ci permet aussi de lever l'ambiguité sur certaines classes. Ainsi si une porte a été faussement étiquetée fenêtre, la segmentation idéale de la référence peut aider à corriger cette erreur. On souhaite tirer profit de cette observation en résolvant conjointement les deux problèmes. Nous proposons dans un premier temps un moyen d'initialiser les deux tâches. Puis nous introduisons le modèle bayésien qui les lie. Les détails de l'inférence par Espérance-Maximisation sont ensuite décrits avant de discuter les résultats tant

qualitativement que quantitativement sur différentes bases de tests.

4.2 Initialisation du recalage et de la segmentation sémantique

L'initialisation nécessaire à l'algorithme d'Espérance-Maximisation repose sur les chapitres précédents. Elle peut être décomposée en 4 étapes et on y fera référence sous la notation $t = t_0$:

1. les 3 points de fuite de Manhattan sont calculés dans l'image (cf. chapitre 2)
2. l'image est rectifiée de manière à apparaitre comme étant vue frontalement (plusieurs rectifications sont possibles)
3. les façades sont détectées et identifiées dans les images rectifiées (cf. chapitre 3)
4. la segmentation sémantique ainsi que le recalage sont initialisées à partir des boites englobantes des façades reconnues

4.2.1 Initialisation du recalage par détection

Comme l'image a été préalablement rectifiée en connaissant les paramètres de calibration intrinsèques de la caméra, les seuls paramètres nécessaires pour recaler la façade de référence sur l'image courante sont un paramètre d'échelle s (l' *aspect-ratio* d'image est conservé) et deux paramètres de translation (t_x, t_y). La reconnaissance de façade permet de sélectionner la façade de référence à recaler parmi une large base de références. De plus, grâce à la détection de façades, on peut estimer une première initialisation des paramètres de recalage $\left(s^{(t_0)}, t_x^{(t_0)}, t_y^{(t_0)}\right)$ en résolvant par moindres carrés le système linéaire qui traduit la mise en correspondance des 4 coins de la détection avec les 4 coins de la référence :

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ h & 1 & 0 \\ w & 0 & 1 \end{pmatrix} \begin{pmatrix} s^{(t_0)} \\ t_x^{(t_0)} \\ t_y^{(t_0)} \end{pmatrix} = \begin{pmatrix} x_{min} \\ y_{min} \\ x_{max} \\ y_{max} \end{pmatrix} \tag{4.1}$$

(x_{min}, y_{min}) et (x_{max}, y_{max}) représentent respectivement les coordonnées du coin supérieur gauche et du coin supérieur droit de la boite englobante détectée. (h, w) sont les dimensions de la façade de référence qui doit être recalée.

4.2.2 Initialisation de la segmentation sémantique

Dans le chapitre précédent, on a vu que la détection de façades utilisait des indices basés sur la segmentation sémantique. On choisit cette première estimation de la segmentation sémantique comme *a priori* pour notre méthode conjointe. Si le réseau SegNet avait été modifié dans le chapitre précédent de manière à avoir des segments sémantiques plus rectangulaires, il reste sensible au changement d'échelle (Fig. 4.4). Pour améliorer cette segmentation initiale, on adapte l'échelle de l'image à celle de l'image de référence transformée par le recalage initialisé à l'étape précédente $I_{ref} \circ T^{(t_0)}$. Comme les dimensions d'entrée de réseaux sont fixés à (360×480), ce

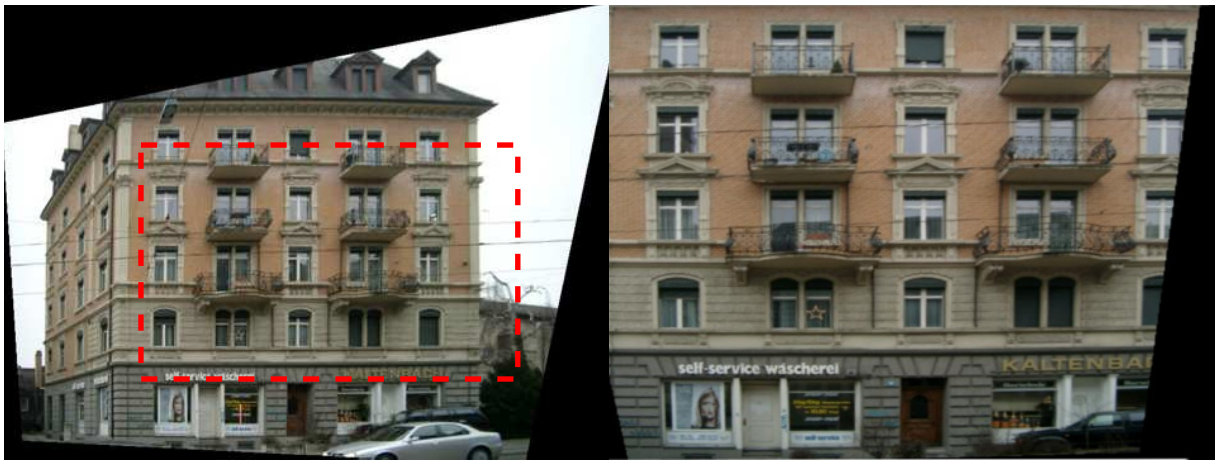

FIGURE 4.3 – Le recalage initial de l'image de référence I_{ref} (à droite) en surimpression sur l'image cible I (à gauche).

changement d'échelle se traduit par un zoom dans la région de l'image qui correspond aux frontières de la référence transformée $I_{|(t_x^{(t_0)},t_y^{(t_0)},s^{(t_0)}h+t_x^{(t_0)},s^{(t_0)}w+t_y^{(t_0)})}$. En pratique, le paramètre d'échelle $s^{(t_0)}$ de l'initialisation est augmenté d'une constante pour éviter que la région d'intérêt ne soit rognée (typiquement $s^{(t_0)} + 0.4$).

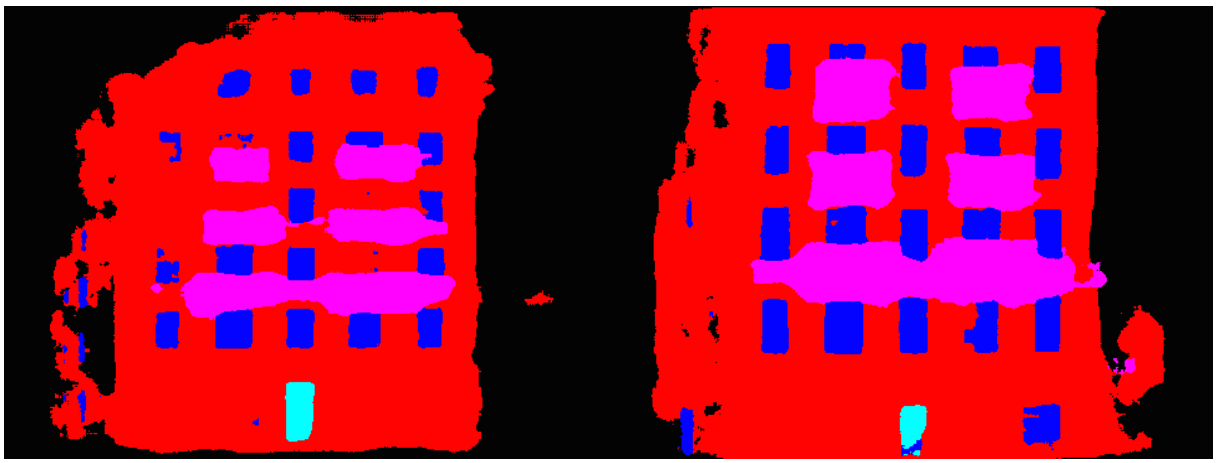

FIGURE 4.4 – Segmentation sémantique initiale de l'image cible I avant (à gauche) et après changement d'échelle (à droite).

4.3 Résolution jointe des problèmes de recalage et segmentation sémantique

4.3.1 Formulation bayésienne

On souhaite recaler l'image de référence reconnue I_{ref} sur l'image cible I dans laquelle la façade a été détectée selon la transformation T. On veut simultanément améliorer la qualité de la segmentation sémantique. On note $L = \{l_j\}_{1 \leq j \leq K}$ les différentes étiquettes de la segmentation sémantique qui sont caractéristiques de l'architecture des façades comme "fenêtre", "porte" ou "balcon". Les autres étiquettes inférées par le réseau du chapitre précédent sont abandonnées (i.e. "façade","ciel","route","fond"). Les images cible et de référence sont toutes deux considérées comme des ensembles de points 2D. Soit $X = \{X_i\}_{1 \leq i \leq N}$ un ensemble de N points de données $X_i = (x_i, y_i)$ de l'image cible I. Ces points sont les coordonnées des pixels i de l'image cible I qui ont une probabilité raisonnable d'être une des étiquettes retenues $P(l_j|i, I) \geq 0.01$ (Fig. 4.6). Cette probabilité $P(l_j|i, I)$ est le score de la dernière couche du CNN utilisé pour la segmentation sémantique.

L'ensemble de points X_{ref} de l'image de référence I_{ref} est modélisé par un mélange de distributions Gaussiennes L_p (notées $\mathcal{N}_p$ cf. Eq. 4.4) pour chaque étiquette l_j : $\left(\pi_{k_j}, \mu_{k_j}, \Sigma_{k_j}\right)_{1 \leq k_j \leq m_j}$. Ces distributions sont particulièrement bien adaptées aux composantes architecturales des façades car la boule unité de la norme L_p $\|M\|_{p,\Sigma}^p = \frac{m_x^p}{\Sigma_{xx}} + \frac{m_y^p}{\Sigma_{yy}}$ est grossièrement rectangulaire pour les grandes valeurs de p (Fig. 4.5).

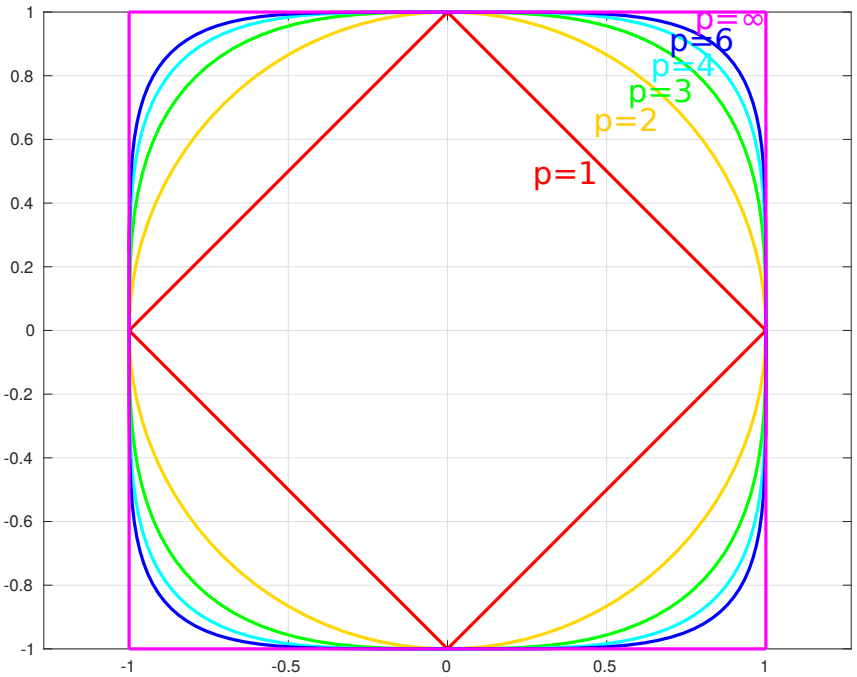

FIGURE 4.5 – Frontière de la boule-unité selon la norme L_p ($\|x\|_p = 1$) pour différentes valeurs de p.

Le but est d'estimer la transformation géométrique $T(\Theta)$ de paramètre d'état $\Theta = (t_x, t_y, s)$ qui recale ces Gaussiennes L_p sur les points de données observés X de l'image cible I. De plus, l'assignation d'un point de donnée X_i à une Gaussienne L_p transformée, renforcée par la probabilité de segmentation *a priori* $P(l_j|, i, I)$ peut être vue comme une segmentation *a posteriori*. En faisant l'hypothèse que les données observées sont indépendantes et en prenant le logarithme, la distribution *a posteriori* peut être maximisée pour trouver Θ :

$$\begin{aligned} \Theta^\star &= \underset{\Theta}{\operatorname{argmax}} \ln P(X|\Theta, I)P(\Theta) \\ &= \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \ln P(X_i|\Theta, I) + \ln P(\Theta) \end{aligned} \tag{4.2}$$

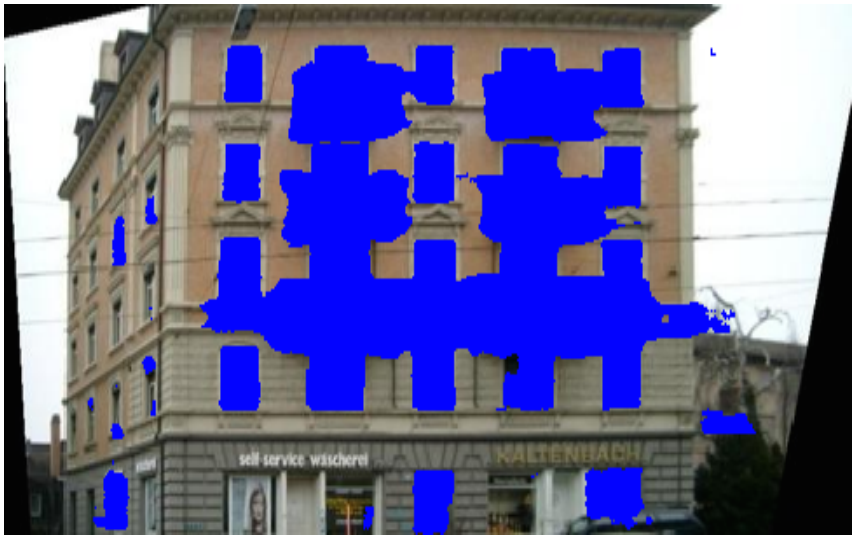

FIGURE 4.6 – Points de données X de l'image cible. Seuls les points des pixels i qui sont probablement $(P(l_j|i, I) \geq 0.01)$ des éléments architecturaux caractéristiques des façades sont considérés.

En utilisant la formule des probabilités totales, on peut introduire la probabilité $P(X_i|l_j, \Theta, I)$ qui est modélisée par un mélange de Gaussiennes L_p transformées (Eq. 4.4), et $P(l_j|i, \Theta, I)$ qui peut être vue comme une probabilité de segmentation *a priori* :

$$\begin{aligned} P(X_i|\Theta, I) = &\sum_{j=1}^{K} P(X_i|l_j, \Theta, I)P(l_j|i, \Theta, I) \\ &+ P(X_i|o, \Theta, I)P(o|i, \Theta, I) \end{aligned} \tag{4.3}$$

$\alpha = P(o|i, \Theta, I)$ est le taux d'anomalies (*outliers*) et on choisit une distribution spatiale uniforme $\mathcal{U}[H, W]$ pour modéliser les anomalies $P(X_i|o, I) = \frac{1}{HW}$ avec H, W les dimensions de l'image cible. En pratique, le taux d'anomalie est initialisé à $\alpha = 0.25\left(1 - \frac{s^2hw}{HW}\right)$ avec h, w les

dimensions de l'image de référence.

$$
\begin{aligned}
P(X_i|l_j,\Theta,I) &= \sum_{k_j=1}^{m_j} \pi_{k_j} \mathcal{N}_p\left(X_i|T\mu_{k_j}, s^p\Sigma_{k_j}\right) \\
&= \sum_{k_j=1}^{m_j} \pi_{k_j} \frac{\exp\left(-\left\|X_i - T\mu_{k_j}\right\|^p_{p,s^p\Sigma_{k_j}}\right)}{4/p^2\Gamma(1/p)^2|s^p\Sigma_{k_j}|}
\end{aligned}
\tag{4.4}
$$

Pour modéliser correctement la forme des composants de façade tout en impactant peu le temps de calcul on choisit $p = 4$. Le nombre de Gaussiennes L_p et leurs paramètres sont déterminés dans l'image de référence I_{ref} (Fig. 4.7). Premièrement, on suppose qu'une segmentation sémantique parfaite de l'image de référence de la façade détectée est disponible. Ensuite, pour chaque étiquette, on y extrait les composantes connexes et une Gaussienne L_p y est ajustée pour chacune d'elles. Cet ajustement est fait en deux temps. D'abord on initialise les paramètres comme s'il s'agissait d'une Gaussienne classique ($p = 2$). Le centre d'une Gaussienne L_p, μ_{k_j} est alors initialisé à la moyenne des coordonnées pixels de la composante connexe dont elle est issue. Comme l'image est rectifiée et les formes des composantes connexes généralement rectangulaires, les axes des Gaussiennes L_p sont alignés avec les axes du repère image. La covariance $\Sigma_{k_j} = \text{diag}\left(\sigma_x^{p/2}, \sigma_y^{p/2}\right)$ est donc considérée diagonale et est initialisée à partir des variances verticales et horizontales (respectivement σ_x et σ_y). Ces deux paramètres $\left(\mu_{k_j}, \Sigma_{k_j}\right)$ sont ensuite rafinés en minimisant l'erreur entre la composante connexe et une véritable distribution gaussienne L_p par un algorithme de Gauss-Newton. Les poids du mélange $\left(\pi_{k_j}\right)_{1\leq j\leq K, 1\leq k_j\leq m_j}$ sont initialisés tels que π_{k_j} est le ratio du nombre de points X_{ref} de la composante connexe k_j sur le nombre total de points X_{ref} dans l'image de référence I_{ref}. Ils sont ensuite normalisés $\sum_{j,k_j} \pi_{k_j} = 1$.

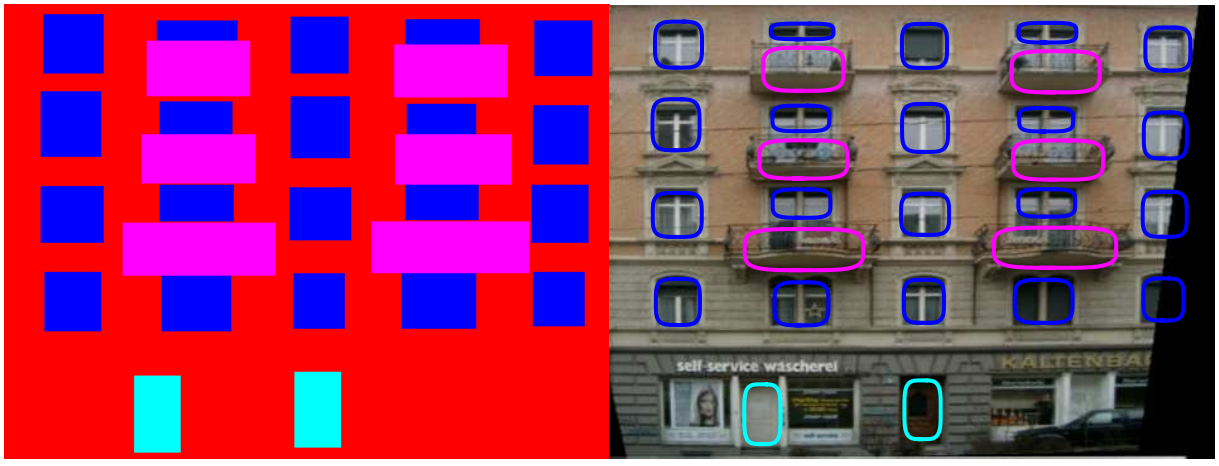

FIGURE 4.7 – Verité-terrain de la segmentation sémantique de l'image de référence I_{ref} (à gauche) et le mélange de Gaussiennes L_p qui le modélise (à droite).

Pour être plus robuste aux occultations, on choisit de laisser les poids du mélange libres

d'évoluer au cours de l'inférence mais on suppose qu'ils suivent une certaine distribution *a priori*. Ce compromis permet de maintenir leurs valeurs proches de leurs valeurs initiales et ainsi éviter une totale ignorance des données au profit des anomalies. On peut en fait ajouter les poids du mélange aux paramètres $\Theta = \left(\{\pi_{k_j}\}_{1\leq j\leq K\,,1\leq k_j\leq m_j}, \alpha, t_x, t_y, s\right)$ sans changer l'équation Eq. 4.2. Nous ne considérons aucun *a priori* sur les paramètres de la transformation (t_x, t_y, s) mais on choisit une distribution de Dirichlet comme *a priori* pour les poids du mélange π_{k_j} :

$$P(\Theta) = \mathcal{D}ir\left(\pi_{k_j}|\alpha_{k_j}\right)_{1\leq j\leq K\,,1\leq k_j\leq m_j} \propto \prod_{j,k_j} \pi_{k_j}^{\alpha_{k_j}-1} \tag{4.5}$$

Gauvain et al. [42] ont montré que cette distribution de Dirichlet est un candidat efficace pour les mélanges de distributions car il permet d'établir des formules analytiques aux solutions des équations de l'Espérance-Maximisation. $\left(\alpha_{k_j}\right)_{1\leq j\leq K\,,1\leq k_j\leq m_j}$ sont fixés aux mêmes valeurs que les poids initiaux du mélange $\alpha_{k_j} = \pi_{k_j}^{(t_0)}$.

4.3.2 Résolution par Espérance-Maximisation

Ce problème de Maximum *A Posteriori* (MAP Eq. 4.2) peut être résolu dans un cadre d'Espérance-Maximisation.
On définit les variables latentes $Z = \left\{z_{i,j,k_j} \in \{0,1\}, z_{i,o} \in \{0,1\}\right\}_{1\leq i\leq N\,,1\leq j\leq K\,,1\leq k_j\leq m_j}$ telles que z_{i,j,k_j} assigne un point X_i à une gaussienne $L_p\left(T\mu_{k_j}, s^p\Sigma_{k_j}\right)$ de l'étiquette l_j et $z_{i,o}$ assigne X_i à la classe d'anomalie supplémentaire o. L'algorithme d'Espérance-Maximisation cherche à trouver la solution iterativement en alternant entre le calcul de l'espérance (par rapport à Z) de la log-vraisemblance complétée $Q(\Theta|\Theta^{(t)})$ conditionnellement à X et aux paramètres courants $\Theta^{(t)}$, et la recherche des paramètres Θ qui maximise cette quantité :

$$\begin{aligned}
Q\left(\Theta|\Theta^{(t)}\right) &= \mathbb{E}_{Z|X,\Theta^{(t)}} \ln P(X, Z|\Theta) \\
&= \sum_{Z} P(Z|X,\Theta^{(t)}) \ln P(X, Z|\Theta) \\
&= \sum_{i,j}\sum_{k_j} \beta_{i,j,k_j}\left(\ln \pi_{k_j} + \ln P(l_j|i,\Theta,I)\right) \\
&+ \sum_{i,j}\sum_{k_j} \beta_{i,j,k_j} \ln \mathcal{N}_p\left(X_i|T\mu_{k_j}, s^p\Sigma_{k_j}\right) \\
&+ \sum_{i} \gamma_i \ln \frac{\alpha}{HW}
\end{aligned} \tag{4.6}$$

avec $\beta_{i,j,k_j} = \mathbb{E}\left(z_{i,j,k_j}|X,\Theta^{(t)}\right)$ et $\gamma_i = \mathbb{E}\left(z_{i,o}|X,\Theta^{(t)}\right)$
Ainsi l'algorithme d'Espérance-Maximisation itère entre ces deux étapes :

— **E-Step :** calcule β_{i,j,k_j} et γ_i

— **M-Step :** $\Theta^{(t+1)} = \text{argmax}_\Theta\, Q\left(\Theta|\Theta^{(t)}\right) + \ln P(\Theta)$

L'étape **E-Step** peut être vue comme le calcul de la probabilité d'assignation de chaque point de donnée observé X_i à une Gaussienne $L_p\left(T\mu_{k_j}, s^p\Sigma_{k_j}\right)$ de l'étiquette l_j connaissant les paramètres $\Theta^{(t)} = \left(\{\pi_{k_j}\}_{1\leq j\leq K, 1\leq k_j\leq m_j}^{(t)}, \alpha^{(t)}, t_x^{(t)}, t_y^{(t)}, s^{(t)}\right)$. En utilisant la règle de Bayes et en notant $\lambda = \frac{\alpha}{HW}$, on peut écrire :

$$\begin{aligned}\beta_{i,j,k_j} &= \mathbb{E}\left(z_{i,j,k_j}|X,\Theta^{(t)}\right)\\ &= \frac{\pi_{k_j}\mathcal{N}_p\left(X_i|T\mu_{k_j}, s^p\Sigma_{k_j}\right)P(l_j|i,\Theta,I)}{\sum_{j',k'}\pi_{k'_{j'}}\mathcal{N}_p\left(X_i|T\mu_{k'_{j'}}, s^p\Sigma_{k'_{j'}}\right)P(l_{j'}|i,\Theta,I)+\lambda}\end{aligned} \tag{4.7}$$

$$\begin{aligned}\gamma_i &= \mathbb{E}\left(z_{i,o}|X,\Theta^{(t)}\right)\\ &= \frac{\lambda}{\sum_{j',k'}\pi_{k'_{j'}}\mathcal{N}_p\left(X_i|T\mu_{k'_{j'}}, s^p\Sigma_{k'_{j'}}\right)P(l_{j'}|i,\Theta,I)+\lambda}\end{aligned} \tag{4.8}$$

Dans l'étape **M-Step** nous visons à maximiser $R = Q\left(\Theta|\Theta^{(t)}\right) + \ln P(\Theta)$ connaissant les assignations $\beta_{i,j,k}$ et γ_i. En développant les expressions des distributions des équations 4.4 et 4.5 et en ignorant les termes constants, R peut être réécrit $\tilde{R}$:

$$\begin{aligned}\tilde{R} = &-\sum_{i,j,k_j}\beta_{i,j,k_j}\left(\ln|s^p\Sigma_{j,k_j}| + \left\|X_i - T\mu_{k_j}\right\|^p_{p,s^p\Sigma_{j,k_j}} - \ln P(l_j|i,\Theta,I)\right)\\ &+\sum_{i,j,k_j}\beta_{i,j,k_j}\ln\pi_{k_j} + \sum_i\gamma_i\ln\lambda + \sum_{j,k_j}\left(\alpha_{k_j}-1\right)\ln\pi_{k_j}\end{aligned} \tag{4.9}$$

Des dérivées partielles $\frac{\partial\tilde{R}}{\partial t_x} = \frac{\partial\tilde{R}}{\partial t_y} = \frac{\partial\tilde{R}}{\partial s} = 0$ on peut dériver un système polynomial qui ne peut être résolu de manière analytique pour $p = 4$. Notre stratégie de résolution est similaire à l'approche utilisée pour l'initialisation des paramètres du mélange sur la sémantique de référence. Premièrement nous résolvons analytiquement le système polynomial avec $p = 2$ qui correspond à un mélange de Gaussiennes classique :

$$\begin{cases} a_1 + a_2 s + a_3 t_x + a_4 t_y + a_5 t_x s + a_6 t_y s + a_7 t_x^2 + a_8 t_y^2 + a_9 s^2 + a_{10} s^3 = 0\\ \frac{a_3}{2} + a_7 t_x + a_5 s - a_{11} s^2 = 0\\ \frac{a_4}{2} + a_8 t_x + a_6 s - a_{12} s^2 = 0\end{cases} \tag{4.10}$$

La résolution d'un tel système revient à résoudre une équation quartique en s :

$$\begin{cases} \left(4a_{11}^2 a_8 + 4a_{12}^2 a_7\right) s^4 + \left(4a_{10}a_7a_8 - 4a_{11}a_5a_8 - 4a_{12}a_6a_7\right) s^3 + 4a_7a_8a_9 s^2 \\ + \left(4a_2a_7a_8 - 2a_3a_5a_8 - 2a_4a_6a_7\right) s + 4a_1a_7a_8 - a_3^2a_8 - a_4^2a_7 = 0 \\ t_x = \frac{2a_{11}s^2 - 2a_5 s - a_3}{2a_7} \\ t_y = \frac{2a_{12}s^2 - 2a_6 s - a_4}{2a_8} \end{cases} \tag{4.11}$$

avec

$$\begin{aligned} a_1 &= -\sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{x_i^2}{\sigma_{k_j,x}} + \frac{y_i^2}{\sigma_{k_j,y}} \right) \\ a_2 &= \sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{x_i \mu_{k_j,x}}{\sigma_{k_j,x}} + \frac{y_i \mu_{k_j,y}}{\sigma_{k_j,y}} \right) \\ a_3 &= 2\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{x_i}{\sigma_{k_j,x}} \quad a_4 = 2\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{y_i}{\sigma_{k_j,y}} \\ a_5 &= -\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x}}{\sigma_{k_j,x}} \quad a_6 = -\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y}}{\sigma_{k_j,y}} \\ a_7 &= -\sum_{i,j,k_j} \beta_{i,j,k_j} / \sigma_{k_j,x} \quad a_8 = -\sum_{i,j,k_j} \beta_{i,j,k_j} / \sigma_{k_j,y} \\ a_9 &= 2\sum_{i,j,k_j} \beta_{i,j,k_j} \quad a_{10} = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{2P(l_j|i,I)} \frac{\partial P(l_j|i,I)}{\partial s} \\ a_{11} &= -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{2P(l_j|i,I)} \frac{\partial P(l_j|i,I)}{\partial t_x} \quad a_{12} = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{2P(l_j|i,I)} \frac{\partial P(l_j|i,I)}{\partial t_y} \end{aligned} \tag{4.12}$$

Puis nous raffinons le résultat en minimisant $J = \left\| \frac{\partial \tilde{R}}{\partial t_x} \right\|^2 + \left\| \frac{\partial \tilde{R}}{\partial t_y} \right\|^2 + \left\| \frac{\partial \tilde{R}}{\partial s} \right\|^2$ pour $p = 4$ par descente de gradient. Comme J est polynomiale, le gradient comme le Hessien peuvent être calculés rapidement en utilisant leur expression polynomiale dans l'algorithme de Gauss-Newton (cf. Annexe B.2). La convergence est atteinte après quelques itérations et on met à jour les paramètres de la transformation $\left(t_x^{(t+1)}, t_y^{(t+1)}, s^{(t+1)}\right)$. La mise-à-jour des poids π_{k_j} et le taux d'anomalies α suivent les formules établies par [42] :

$$\pi_{k_j}^{(t+1)} = \frac{\sum_i \beta_{i,j,k_j} + \alpha_{k_j} - 1}{\sum_{i,k'_{j'}} \beta_{i,j,k_{j'}} + \sum_{k'_{j'}} \left(\alpha_{k'_{j'}} - 1\right)} \tag{4.13}$$

$$\alpha^{(t+1)} = \frac{\sum_i \gamma_i}{\sum_{i,k'_{j'}} \beta_{i,j,k_{j'}} + \sum_{k'_{j'}} \left(\alpha_{k'_{j'}} - 1\right)} \tag{4.14}$$

4.3.3 Discussions sur les cas de résolution

La distribution $P(l_j|\Theta, I)$ peut être identifiée à l' *a priori* de la segmentation sémantique de l'image cible et en ce sens est indépendante de la transformation T. Cela permet de simplifier les formules précédentes (cf. Eq. 4.11) qui font intervenir les dérivées partielles $\frac{\partial P(l_j|\Theta, I^{(t)})}{\partial s}$, $\frac{\partial P(l_j|\Theta, I^{(t)})}{\partial t_x}$. Leur simplification permet d'alléger le système polynomial quartique de l'étape **M-**

Step avec $p = 2$ (cf. Eq. 4.9) qui se réduit alors à une équation quadratique en s et deux équations linéaires en t_x et t_y qui se résout facilement :

$$\begin{cases} s = \frac{-2a_2a_7a_8+a_3a_5a_8+a_4a_6a_7 \pm \sqrt{\Delta}}{4a_9a_7a_8} \\ t_x = \frac{-a_3-2a_5s}{2a_7} \\ t_y = \frac{-a_4-2a_6s}{2a_8} \end{cases} \tag{4.15}$$

avec

$$\begin{aligned} \Delta = &-16a_1a_7^2a_8^2a_9 + 4a_2^2a_7^2a_8^2 - 4a_2a_3a_5a_7a_8^2 - 4a_2a_4a_6a_7^2a_8 \\ &+ a_3^2a_5^2a_8^2 + 4a_3^2a_7a_8^2a_9 + 2a_3a_4a_5a_6a_7a_8 + a_4^2a_6^2a_7^2 + 4a_4^2a_7^2a_8a_9 \end{aligned} \tag{4.16}$$

Malgré la robustesse intrinsèque des réseaux convolutionels aux translations, l'inférence de la segmentation sémantique par le CNN reste sensible aux changements d'échelle. Aussi on peut vouloir tirer profit du recalage courant pour améliorer la distribution $P(l_j|\Theta, I)$ à chaque itération de l'EM. On a alors développé une variante du problème dans laquelle on infére une nouvelle segmentation sémantique plus précise autour de chaque recalage. On distingue pour cela l'image cible de départ $I = I^{(t_0)}$ et l'image cible au temps t donnée par $I^{(t)} = I_{|(t_x,t_y,sh+t_x,sw+t_y)}$, la sous-image de I restreinte à la zone de recalage courant définie par la frontière de $I_{ref} \circ T^{(t)}$. Ainsi $P(l_j|\Theta, I^{(t)})$ dépend de T et les dérivées partielles $\frac{\partial P(l_j|\Theta,I^{(t)})}{\partial s}$, $\frac{\partial P(l_j|\Theta,I^{(t)})}{\partial t_x}$ et $\frac{\partial P(l_j|\Theta,I^{(t)})}{\partial t_y}$ doivent être prises en compte dans la résolution du système polynomial (Eq. 4.11) qui bénéficie néanmoins toujours d'une solution analytique. Ces dérivées partielles peuvent être calculées par différences finies du CNN selon les paramètres de transformation T :

$$\frac{\partial P(l_j|\Theta, I^{(t)})}{\partial s} \approx \frac{P(l_j|I_{|(t_x,t_y,(s+\epsilon)h+t_x,(s+\epsilon)w+t_y)}, s, t_x, t_y) - P(l_j|I_{|(t_x,t_y,(s-\epsilon)h+t_x,(s-\epsilon)w+t_y)}, s, t_x, t_y)}{2\epsilon} \tag{4.17}$$

$$\frac{\partial P(l_j|\Theta, I^{(t)})}{\partial t_x} \approx \frac{P(l_j|I_{|((t_x+\epsilon,t_y,sh+t_x+\epsilon,sw+t_y)}, s, t_x, t_y) - P(l_j|I_{|(t_x-\epsilon,t_y,sh+t_x-\epsilon,sw+t_y)}, s, t_x, t_y)}{2\epsilon} \tag{4.18}$$

$$\frac{\partial P(l_j|\Theta, I^{(t)})}{\partial t_y} \approx \frac{P(l_j|I_{|(t_x,t_y+\epsilon,sh+t_x,sw+t_y+\epsilon)}, s, t_x, t_y) - P(l_j|I_{|(t_x,t_y-\epsilon,sh+t_x,sw+t_y-\epsilon)}, s, t_x, t_y)}{2\epsilon} \tag{4.19}$$

Ces dérivées partielles pourraient également être calculées plus efficacement par passage de l'image et de sa Jacobienne dans un nouveau réseau de neurones convolutionnels dont les couches peuvent être déduites du réseau original par dérivations en chaine (*Chain-Rule*). Quoiqu'il en soit cette nouvelle formulation du problème nécessite plusieurs passages dans un CNN à chaque itération de l'EM ce qui est excessivement coûteux. Aussi si ce renforcement en ligne de la segmentation peut mieux gérer les cas de fort changement d'échelle, on lui préfèrera en pratique sa version simplifiée. De plus la stabilité intrinsèque du réseau aux petites transformations et le zoom initial font que le gain de la méthode est limité par rapport à sa version efficace (voir Fig. 4.12).

4.4 Détails d'implémentation

4.4.1 Efficacité de la méthode

Contrairement à la plupart des approches EM, dans notre méthode, les paramètres des Gaussiennes L_p sont tous fixés sauf les poids du mélange. En effet ces distributions modélisent les composantes sémantiques de la façade de référence et ce que l'on cherche c'est la transformation T. Cette représentation compacte d'une façade explique en partie l'efficacité de notre algorithme. La complexité pour une itération t de l'algorithme EM dans le cas simplifié est $O\left(NK \max_j m_j\right)$ et la parallélisation est facile pour l'étape **E-Step** car les calculs des β_{i,j,k_j} sont indépendants.

Le nombre de Gaussiennes L_p est de l'ordre du nombre de fenêtres. Il varie typiquement entre 2 et 30 pour des façades de style européen. Le nombre de points de données N est plus difficile à estimer dans l'absolu. Si on suppose que l'image cible est composée complètement de façades adjacentes et que la surface de mur entre deux fenêtres est aussi large que la surface des fenêtres elles-mêmes on peut approximer $N \approx 0.25HW$. Dans nos données de test cette approximation est valide et le nombre moyen de points est de $\hat{N} = 31072$ par image.

L'efficacité de la méthode est aussi une conséquence de la résolution analytique partielle. Il est d'ailleurs important de noter que les itérations internes de l'algorithme de Gauss-Newton de l'étape **M-Step** sont négligeables dans l'ensemble de l'EM. Elles sont en effet typiquement de l'ordre d'une vingtaine mais ne dépendent pas du nombre de points N (cf. annexe B.2). Cela permet à notre schéma d'optimisation du système polynomial d'être dans notre cas particulier plus rapide que des méthodes générales par prolongement homotopique.

L'implémentation est en Matlab avec l'algorithme EM écrit en C++. Le temps de calcul moyen pour une itération t est de 0.023 seconde sur un processeur I7-3520M. Ce temps pourrait être drastiquement réduit par une implémentation GPU qui pourrait profiter des multiples points de parallélisation possibles de l'algorithme.

4.4.2 Schéma multi-résolutions

En pratique le recalage ne requiert pas qu'un point de donnée X_i soit échantillonné à chaque pixel. Partant de ce constat, notre implémentation utilise un schéma multi-résolutions à plusieurs niveaux. On construit une représentation multi-résolutions de l'ensemble des points de données X par sous-échantillonnages successifs de facteur 2. L'algorithme EM est d'abord exécuté sur la version de plus basse résolution jusqu'à convergence $\left\|\Theta^{(t+1)} - \Theta^{(t)}\right\| \leq \epsilon$ puis exécuté sur le niveau suivant de plus haute résolution de X à partir des paramètres $\Theta^{(t)}$ estimés précédemment.

Pour les images de test de taille (360×480) pixels, on utilise deux niveaux de résolution. Si le nombre d'itérations avant convergence de l'algorithme EM dépend fortement de la qualité de l'initialisation, il est en moyenne de seulement 6 pour le niveau de basse résolution. 2 itérations supplémentaires sont typiquement nécessaires pour converger sur le niveau supérieur (Fig 4.8). Ainsi le temps moyen de calcul de l'ensemble de l'algorithme EM est de 0.121 seconde.

Cette accélération a son importance car pour éviter les problèmes de convergence de l'algorithme EM vers un minimum local, on utilise en pratique plusieurs initialisations. La méthode est donc appliquée non seulement sur la façade détectée mais également sur les 20 premières

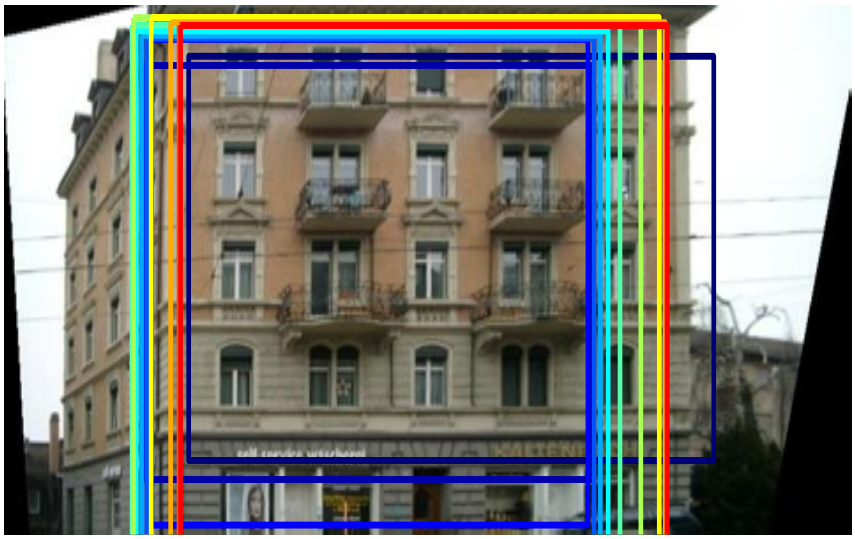

FIGURE 4.8 – Les recalages successifs de la référence pour chaque itération de l'algorithme EM sont représentés en couleurs selon la palette jet de Matlab. La première itération est en bleue et la dernière en rouge.

propositions de façade de la méthode du chapitre précédent, triées selon leur métrique siamoise à la référence I_{ref}. De manière évidente ces exécutions multiples de la méthode sont parallélisables. La solution finale est celle de plus forte log-vraisemblance R.

4.5 Résultats et limites

4.5.1 Jeux de données utilisés

Nous avons testé notre méthode sur 3 jeux de données différents. Le premier est VarCity 3D [1] et s'il ne présente pas de très grosses difficultés en terme de recalage, il permet de valider statistiquement notre méthode. Il consiste en 401 images en vue piétonne des bâtiments le long de la Rue Monge à Paris. Un étiquetage sémantique manuel et une reconstruction 3D de la scène sont disponibles de même que les paramètres de caméra des différentes images. Les points de vue de ces images sont tous très proches d'une vue frontale et les façades recouvrent l'essentiel des images. En conséquence, les changements d'échelle qui paramétrisent la transformation par rapport à la référence sont mineurs mais les translations peuvent être importantes avec également de larges zones de façades non visibles (Fig. 4.9 à gauche).

Le second jeu de données est constitué des 100 premiers bâtiments de la Zurich Buildings Database (ZuBuD) avec 5 points de vue différents par bâtiment. Parmi ces scènes on ne garde que celles qui ont été correctement reconstruites par *Structure From Motion* (SFM) via Visual SFM [2] en les contrôlant chacune visuellement. La diversité de points de vue (Fig. 4.9 au milieu) de ce jeu de données offre une plus large gamme d'échelles que pour VarCity ainsi que des occultations.

1. https ://varcity.ethz.ch/3dchallenge
2. http ://ccwu.me/vsfm

Le dernier jeu de tests dénommé « NancyLights » vise à montrer la robustesse de la méthode proposée aux changements d'illuminations (Fig. 4.9 à droite). Il consiste en deux acquisitions longues (*time-lapse*) de la même façade sous le même point de vue pendant le lever et le coucher de soleil pour un total de 56 images.

FIGURE 4.9 – Exemples d'images issues des différents jeux de données. De gauche à droite : les deux premières illustrent les translations importantes sur VarCity 3D, les deux suivantes les changements de points de vue dans ZuBuD et les deux dernières les changements d'illuminations de NancyLights

Pour chaque bâtiment dans les 3 bases de données, nous choisissons la façade de référence comme la façade en vue frontale qui est la plus complète possible et avec le moins d'occultations. La référence est segmentée sémantiquement manuellement selon les 3 étiquettes "fenêtre", "porte" et "balcon" (Fig. 4.7). Le recalage de la référence est transféré à toutes les images où cette même façade est visible en utilisant les informations géométriques disponibles par le modèle 3D issu de l'algorithme SFM. Cela constitue la vérité-terrain des recalages pour chaque image de la base.

4.5.2 Résultats quantitatifs

On compare notre méthode à la fois à des approches de recalage basées sur la mise en correspondance de points mais aussi à des approches denses basées sur les images entières. Le recalage est systématiquement fait entre l'image cible rectifiée I et l'image de référence I_{ref}. Dans la catégorie des méthodes de recalage denses, nous nous comparons à la détection brute du chapitre précédent, à la minimisation de la norme L_2 entre intensités d'images $\text{argmin}_T \|I - Iref \circ T\|_2$ par descente de gradient, à la maximisation de l'Information Mutuelle [82][104] $\text{argmax}_T\, MI\,(I, Iref \circ T)$, à la corrélation de phase [92]. Pour les méthodes d'optimisation, les mêmes initialisations sont utilisées que pour notre méthode. La méthode par mise en correspondance de points utilisée est une approche basée sur le descripteur SIFT dans un cadre RANSAC [75]. On extrait les descripteurs SIFT [3] dans l'image rectifiée avec une orientation fixe car la transformation T ne concerne pas les rotations. 2 paires de descripteurs SIFT mis en correspondance en utilisant le critère de Lowe [75] sont utilisées pour générer des hypothèses de transformation pour l'estimation par RANSAC. La comparaison est faite dans le référentiel de l'image en calculant les histogrammes normalisés cumulés des erreurs en translation et en échelle. Pour ZuBuD et VarCity 3D, le modèle acquis par SFM permet de montrer aussi l'impact de l'erreur de recalage 2D sur la translation de la pose de la caméra (Tab. 4.1 et Fig. 4.18).

3. http ://www.vlfeat.org

VarCity 3D

Les bons résultats sur VarCity 3D (Fig. 4.10) montrent que notre méthode peut gérer des cas de translations importantes grâce au support infini des Gaussiennes L_p. Même quand ce phénomène coïncide avec la présence de motifs répétés, les initialisations multiples qui exploitent ces répétitions et symétries ainsi que la régularisation du MAP aident à estimer un recalage correct. Au contraire, ces situations sont le défaut majeur des méthodes d'optimisation basées sur l'image qui sont facilement attirées vers un optimum local au moindre contour (Fig. 4.11). Malgré tout, notre méthode peut également échouer dans quelques uns de ces cas lorsqu'un manque de composantes architecturales discriminantes, comme une porte, lui fait préférer un mauvais alignement d'étage ou de fenêtres (Fig. 4.24 à gauche). SIFT résout presque tous ces cas précis en profitant d'autres éléments visuels de la façade.

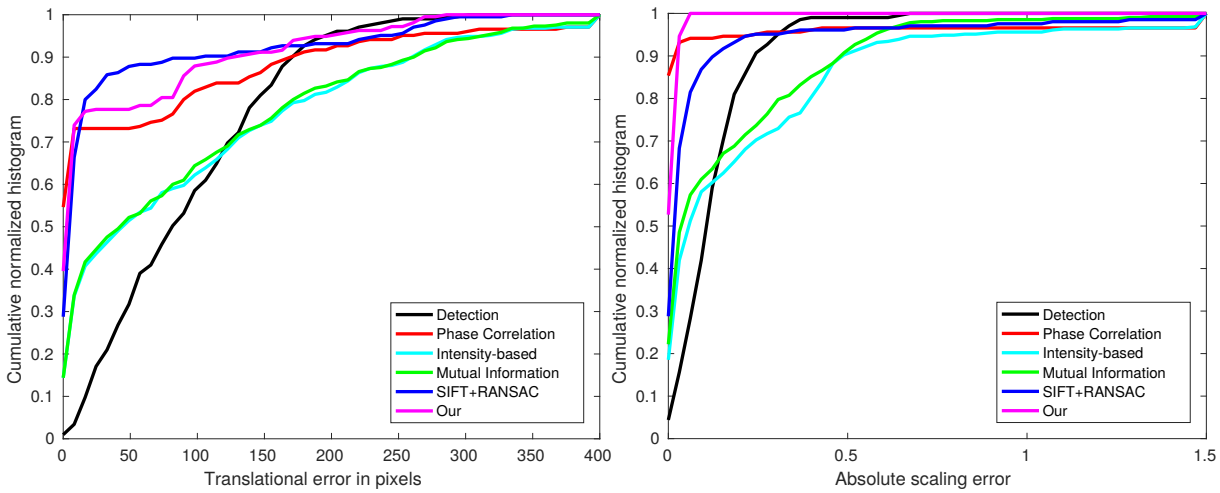

FIGURE 4.10 – Erreurs de recalage 2D sur Varcity 3D

Zurich Buildings Database

Le jeu de tests ZuBuD met l'accent sur d'autres situations complexes car la diversité de points de vues provoque des changements d'apparence dans les images rectifiées notamment en échelle. Les façades sont généralement des objets peu texturés et les artefacts de faible résolution dus à la rectification n'arrangent rien. Dans ces conditions, peu de descripteurs SIFT sont extraits et ils se ressemblent tous ce qui peut être source de mauvais recalage (Fig. 4.13). Comme le recalage est borné à la seule façade, il peut échouer là où un algorithme SFM classique peut exploiter des éléments visuels du contexte. D'un autre côté, notre approche peut bénéficier d'une détection initiale plutôt décente (Fig. 4.12). La segmentation sémantique pouvant se montrer sensible au changement d'échelle, on montre également que la version avec inférence sémantique en ligne de notre méthode (*Our with Jacobian* dans la figure 4.12) améliore un peu les résultats au prix néanmoins d'un surcoût de complexité prohibitif.

Les occultations sont une autre conséquence de la diversité de points de vue sur ce jeu de tests. La mise-à-jour des poids du mélange pendant l'EM permet à notre méthode d'être robuste

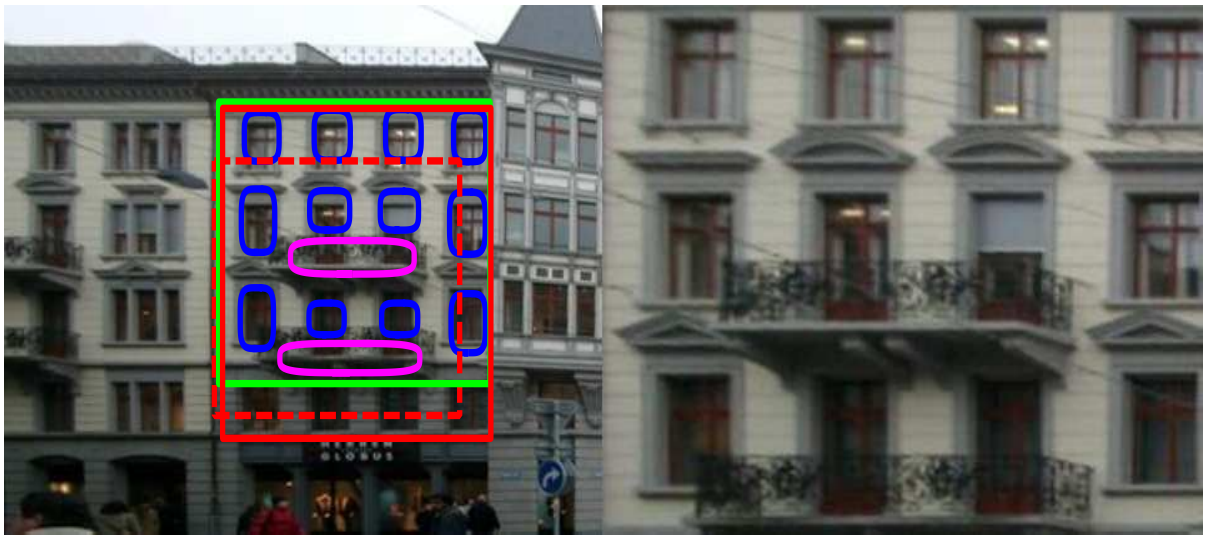

FIGURE 4.11 – Piégée dans un minimum local, la méthode de minimisation de norme L_2 basée sur l'intensité (en rouge) échoue à estimer le recalage tandis que notre méthode (en vert) réussit. Le recalage initial est en tiret et le final en trait plein.

à ceux-ci ainsi qu'aux parties cachées (Fig. 4.17) car la valeur de π_{k_j} peut décroitre si un élément n'est pas visible. Jouant comme critère de régularisation, la distribution *a priori* de Dirichlet sur les poids du mélange évite également la complète ignorance des données en gardant les poids proches de leur valeur initiale α_{k_j} (Fig. 4.14).

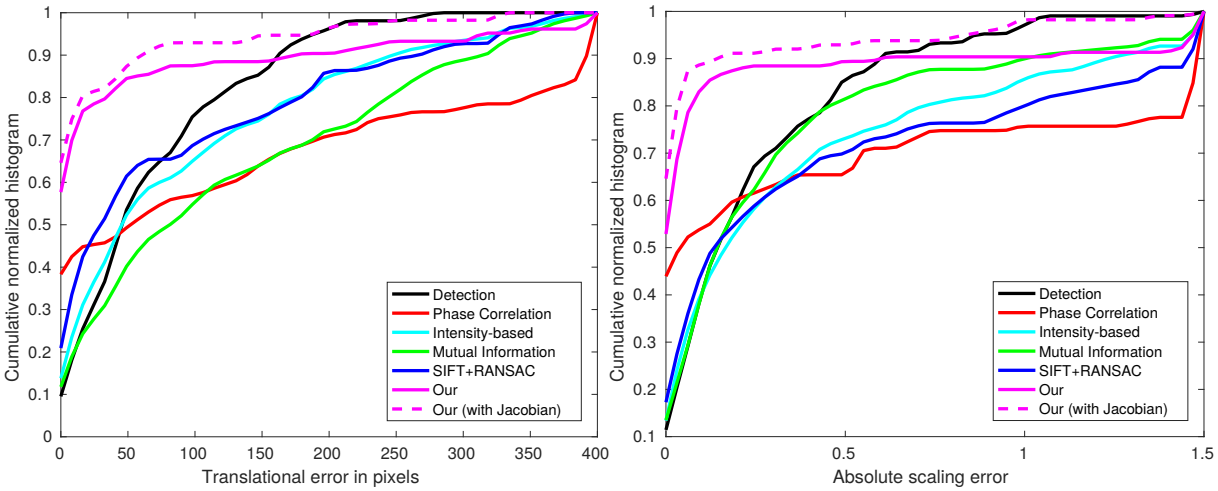

FIGURE 4.12 – Erreurs de recalage 2D sur ZuBuD

NancyLights

L'apparence visuelle des façades peut beaucoup changer : l'aspect des fenêtres varie avec les réflexions du soleil et la présence de volets, l'orientation des balcons dépend du point de

FIGURE 4.13 – Le recalage par SIFT/RANSAC (en rouge) échoue à cause de la symétrie de la façade, tandis que notre méthode réussit (en vert). Le recalage initial est en tiret et le final en trait plein.

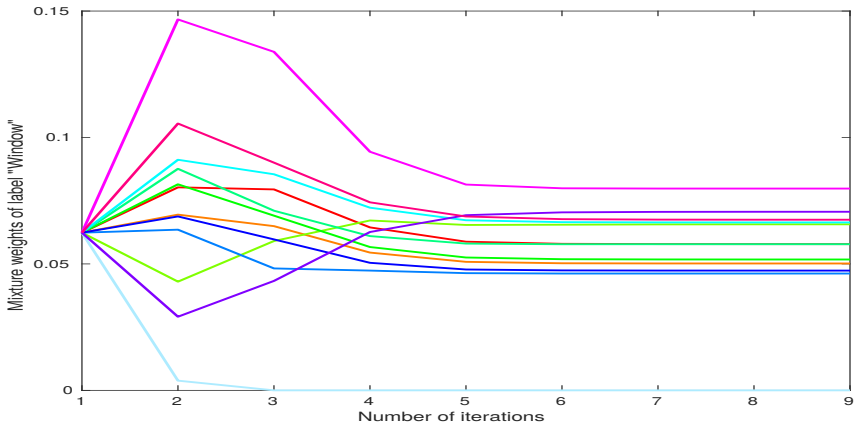

FIGURE 4.14 – Evolution des poids du mélange de l'étiquette "fenêtre" en fonction des itérations sur la facade de l'avant-dernière ligne de la figure 4.23. L'occultation de la fenêtre en bas à gauche fait chuter le poids correpondant (en bleu clair). La distribution de Dirichlet maintient les autres proches de leur valeur initiale malgré la mauvaise segmentation.

vue. Si cette observation est déjà vraie sur ZuBuD, elle l'est encore plus sur le dernier jeu de donnée « NancyLights » où la robustesse aux changements d'illumination est évaluée (Fig. 4.15). En se reposant sur la segmentation sémantique, notre méthode se concentre sur la structure géométrique de la façade alors que les changements d'apparence sont encodés dans le réseau de neurones. L'invariance en illumination du CNN est particulièrement bonne même pour des changements extrêmes de conditions lumineuses (jour/nuit) qui font échouer les autres méthodes (Fig. 4.16).

Même lorsque la distribution de segmentation *a priori* $P(l_j|i, I)$ n'est pas mise-à-jour pendant

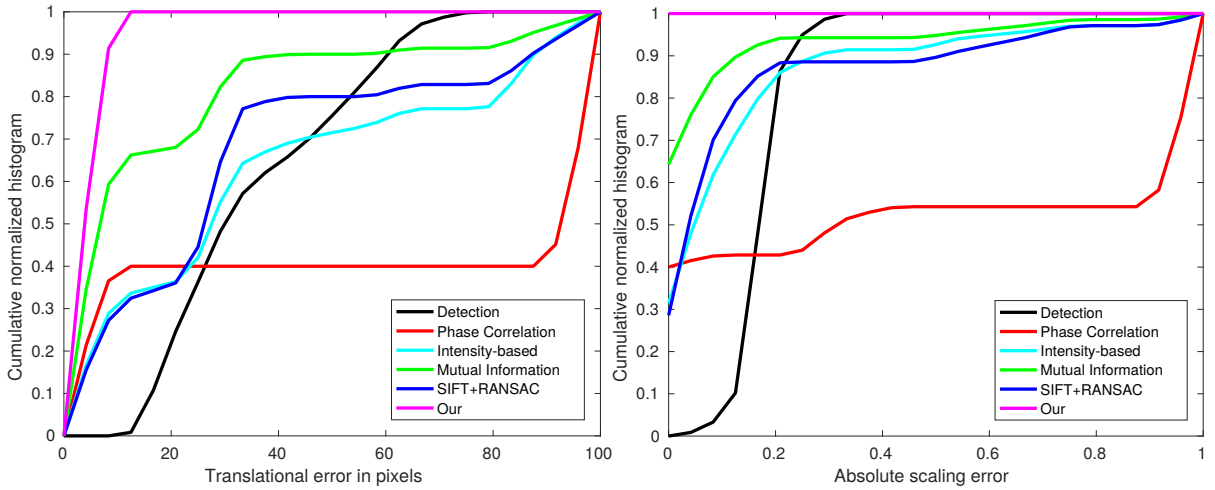

FIGURE 4.15 – Erreurs de recalage 2D sur NancyLight

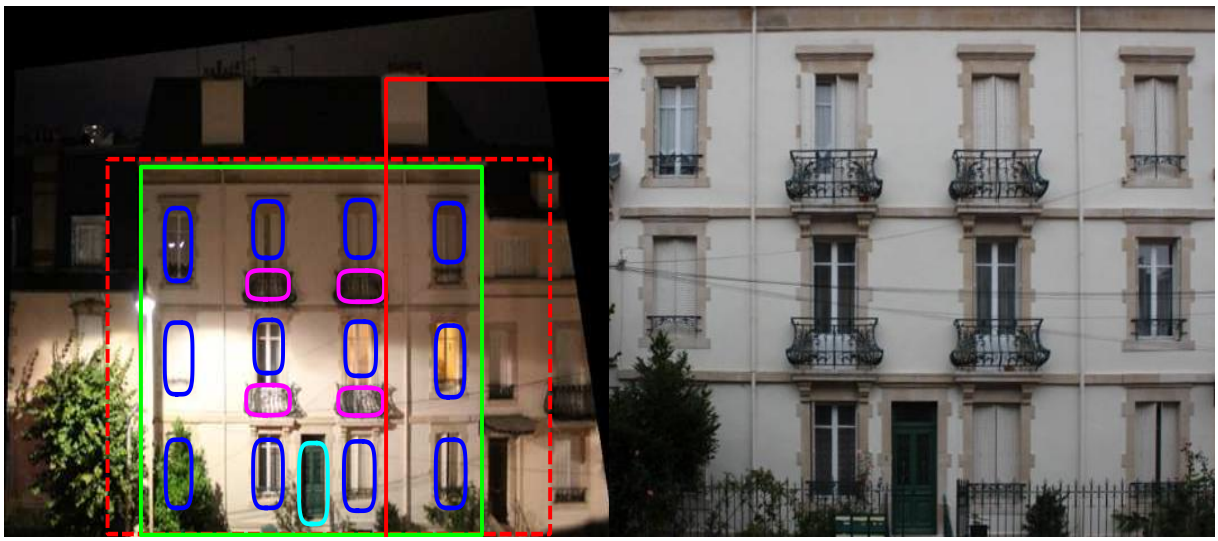

FIGURE 4.16 – Le recalage par corrélation de phase (en rouge) échoue à cause du changement d'illumination important, tandis que notre approche (en vert) réussit. Le recalage initial est en tiret et le final en trait plein.

	Sift	PhCorr	LstSq	InfMut	Nous
VarCity 3D	0.04	0.02	0.37	0.35	0.03
ZuBuD	0.22	0.67	0.33	0.44	0.07

TABLE 4.1 – Erreur médiane relative de la translation 3D de la pose de caméra (relativement à la distance à la façade). "Sift" renvoie à l'approche par mise en correspondance via SIFT+RANSAC, "PhCorr" renvoie à l'approche par corrélation de phase, "LstSq" à la minimisation par moindres carrés (L_2) et "InfMut" à la maximisation de l'Information Mutuelle.

l'EM (cas simplifié), les étiquettes l_j des points de données peuvent changer d'une itération à

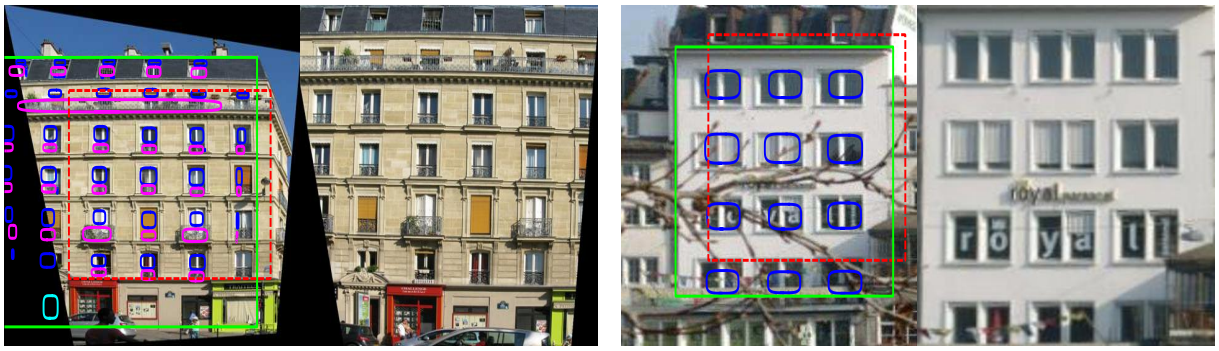

FIGURE 4.17 – Exemples de recalages réussis (en vert) malgré les parties de la référence non visibles (à gauche) ou les occultations (à droite). Le recalage initial est en tiret et le final en trait plein.

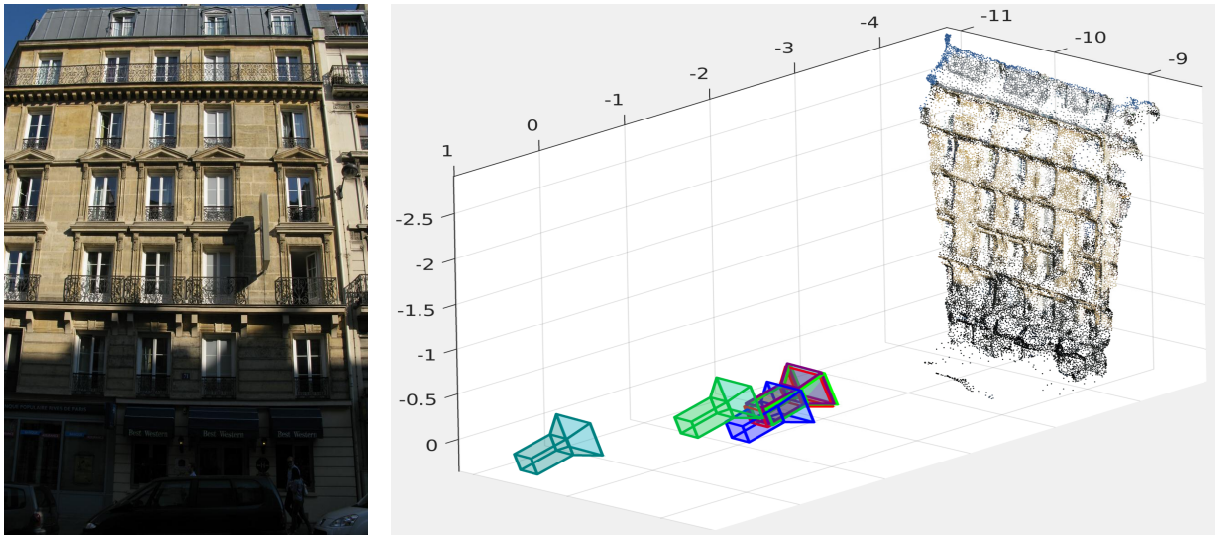

FIGURE 4.18 – Exemple d'image de VarCity 3D (à gauche) et le modèle acquis par SFM associé (à droite). La vérité-terrain de la pose de caméra est en vert clair. Les poses inférées des approches par SIFT+RANSAC, corrélation de phase, moindres carrés, information mutuelle et par notre méthode sont respectivement en bleu, rouge, cyan, vert foncé, et magenta. Notre solutions est superposée à la vérité-terrain.

l'autre (Fig. 4.19). En effet si les classifications incorrectes sont courantes pour les éléments visuellement très proches comme les "portes" et les "fenêtres", l' *a priori* de segmentation peut être renforcé par l'influence de Gaussiennes L_p pendant le recalage. Finalement on peut transférer la probabilité *a posteriori* des points de données X pour mettre à jour à l'*a priori* initial ce qui l'améliore nettement au niveau de ces éléments ambigus (Fig. 4.20).

4.5.3 Résultats qualitatifs

On présente des exemples de résultats sur les trois bases (VarCity 3D, ZuBuD et NancyLights) pour notre méthodes. La colonne de gauche montre l'image de référence I_{ref}. La colonne du milieu

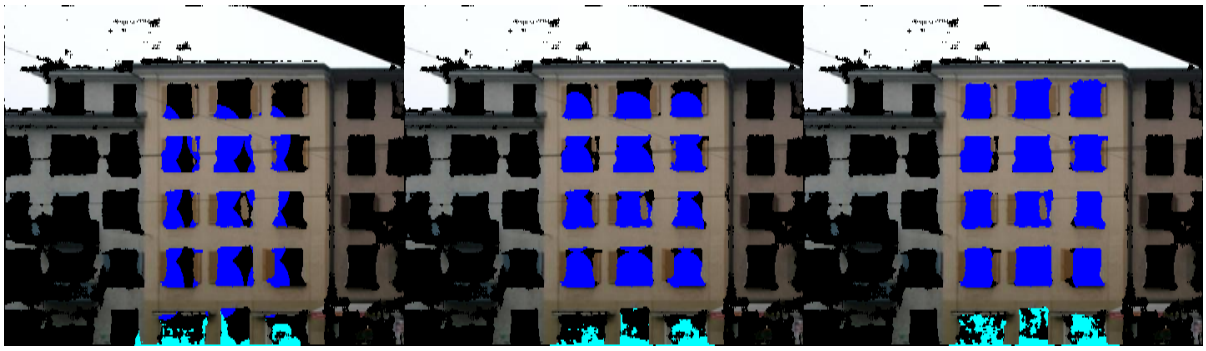

FIGURE 4.19 – Evolution de la segmentation sémantique pendant l'EM sur les 3 premières itérations. Les portes du rez-de-chaussée sont progressivement classifiées en même temps qu'elles aident à guider le recalage.

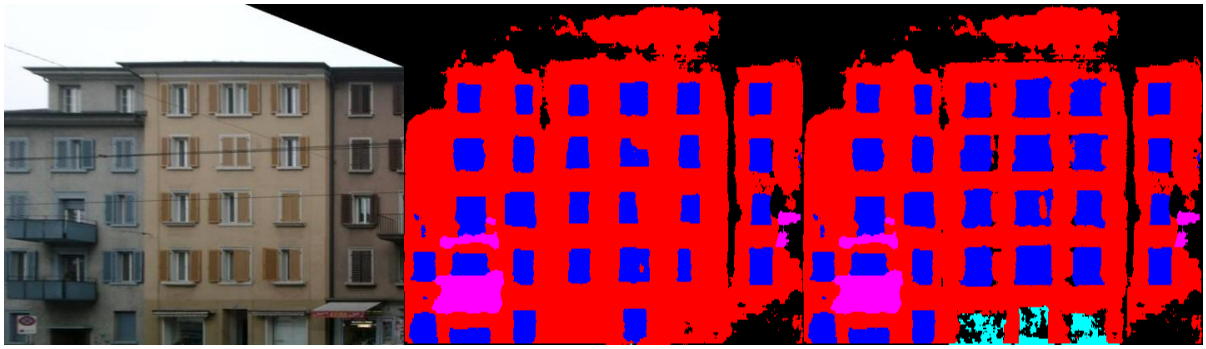

FIGURE 4.20 – De gauche à droite : l'image cible I avec le bâtiment orange comme référence, la segmentation sémantique *a priori* $P(l_j|i, I)$, et la segmentation sémantique *a posteriori* après le recalage. Les 3 portes qui étaient faussement classifiées "façade" et "fenêtre" dans la segmentation *a priori* sont finalement correctement étiquetés.

montre le résultat du recalage sur l'image cible I avec les Gaussiennes L_p étiquetés recalées. Enfin la colonne de droite montre la segmentation *a posteriori*.

4.5.4 Limites de la méthode

Utiliser une vérité-terrain de la segmentation sémantique de référence peut être vu comme une limite de la méthode pour des applications réelles en réalité augmentée ou en robotique. Même si ce genre de données pourraient être déduites des plans de construction de bâtiments ils sont en pratique encore souvent faits manuellement. Cependant comme on l'a rappelé dans l'état de l'art, l'inférence du réseau SegNet pourrait être post-traitée en introduisant de l'information de régularisation basé sur des règles architecturales comme dans [107] ou [81]. Ces méthodes nécessitent les limites exactes de la façade et sont généralement lentes mais seraient finalement bien adaptées pour fournir des segmentations sémantiques de référence propres hors-ligne.

Notre approche est bien adaptée aux images qui ont une structure éparse en éléments caractéristiques comme les façades mais ne peut être généralisée à tous les types d'image. En effet

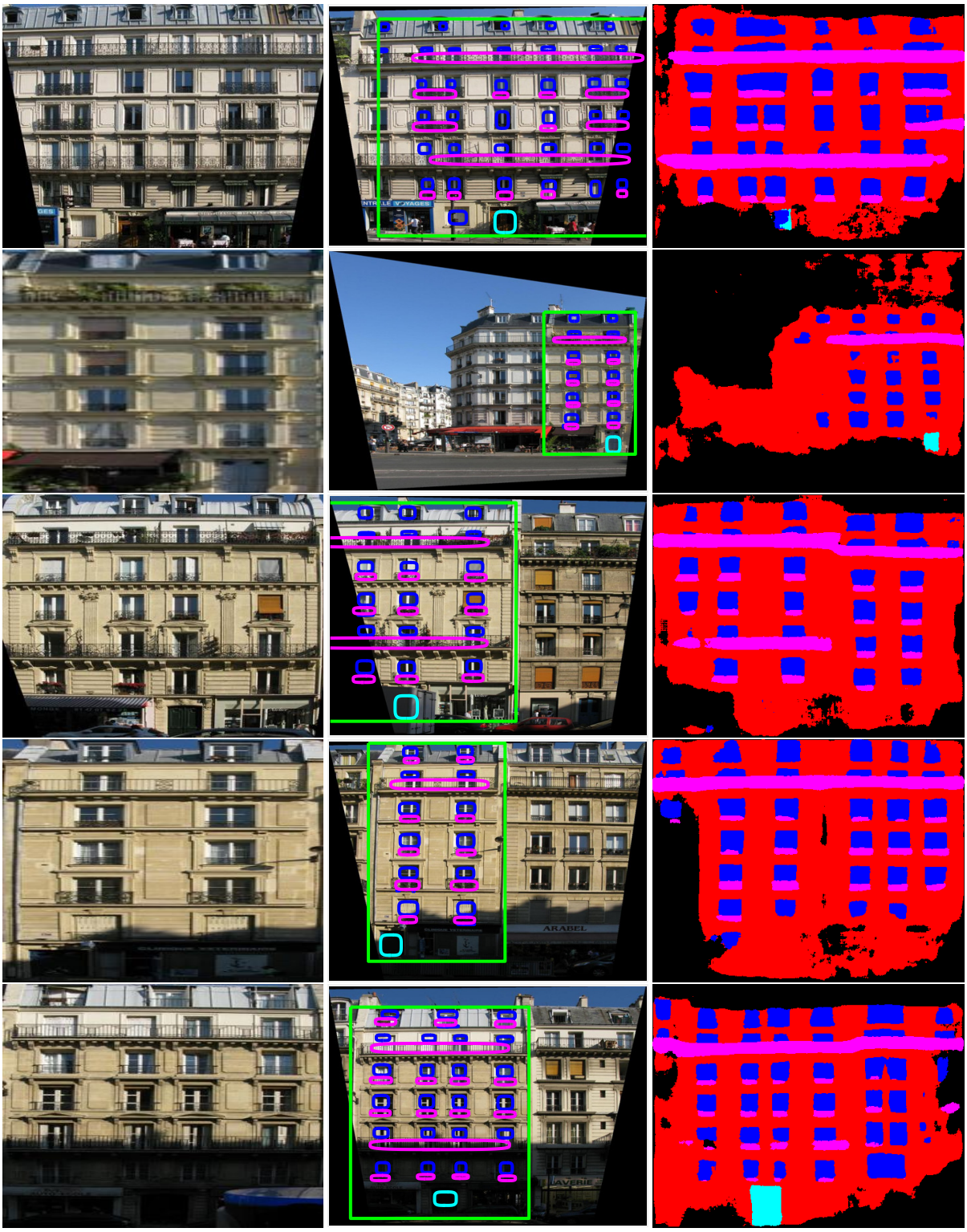

FIGURE 4.21 – Résultats qualitatifs sur VarCity 3D. A gauche la façade de référence, au milieu le résultat du recalage par notre méthode, à droite la segmentation *a posteriori*.

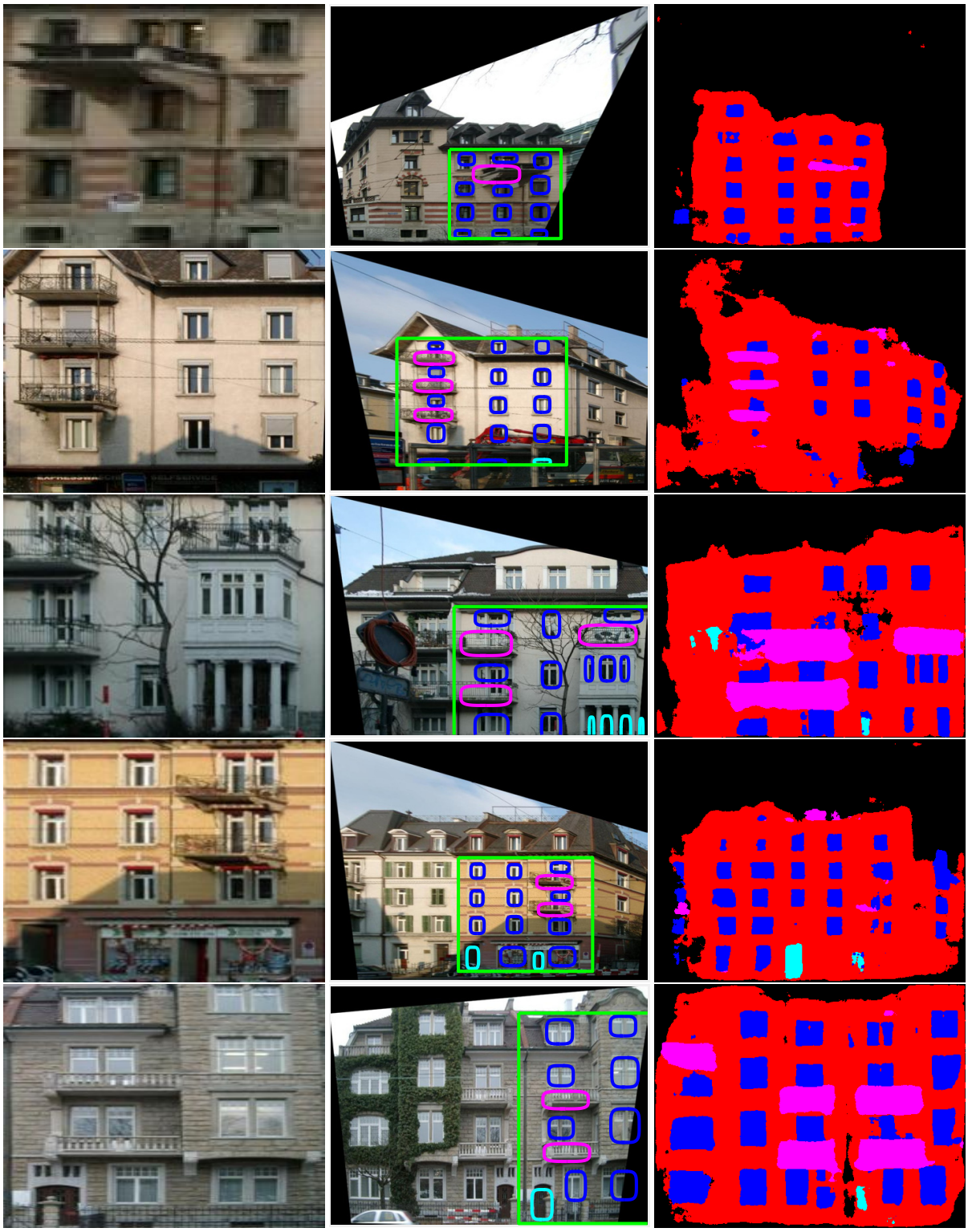

FIGURE 4.22 – Résultats qualitatifs sur ZuBuD. A gauche la facade de référence, au milieu le résultat du recalage par notre méthode, à droite la segmentation *a posteriori*.

FIGURE 4.23 – Résultats qualitatifs sur NancyLights. A gauche la facade de référence, au milieu le résultat du recalage par notre méthode, à droite la segmentation *a posteriori*.

l'utilisation de distributions spatiales spécifiques pour modéliser les façades (Gaussiennes L_p pour les composants de façades et distribution uniforme pour les anomalies) réduit les transferts à d'autres cas d'utilisation où une approche jointe aurait été intéressante comme l'imagerie médicale par exemple. Notre algorithme rencontre d'ailleurs des difficultés dans les situations où cette structuration est faible. Dans le cas où les points de données sont proches d'une répartition dense et où il y a une faible diversité dans les étiquettes, notre méthode tend alors à échouer si l'initalisation n'est pas assez proche de la solution. Les points de données peuvent alors être tous classés comme anomalies ou appartenant à une seule Gaussienne L_p géante (Fig. 4.24 à droite).

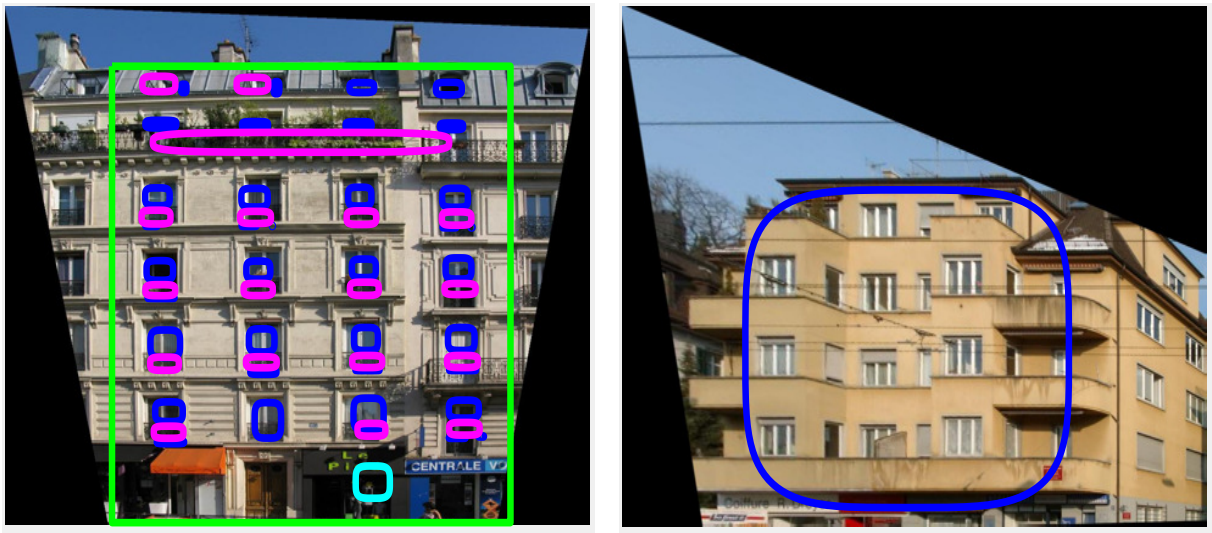

FIGURE 4.24 – Exemple d'erreurs de recalage de notre méthode. A gauche le résultat de notre méthode est décalé d'une colonne de fenêtre ne pouvant s'aligner avec les fenêtres de droite partiellement visibles. A droite la présence du seul élément sémantique dense "fenêtre" conjugué à une géométrie non conforme à l'hypothèse de façade planaire a conduit notre méthode à finir dans un minima local.

4.6 Conclusion

Nous avons présenté un modèle bayésien qui permet de résoudre conjointement les problèmes de recalage et de segmentation sémantique. La méthode proposée est efficace et robuste aux problèmes d'occultations, de répétitions et de changements d'illumination tout en améliorant la qualité de la segmentation sémantique. On a également présenté une variante de notre méthode qui autorise une inférence en ligne de la segmentation sémantique pour augmenter la robustesse au changement d'échelle. La résolution actuelle par différences finies de cette version du problème est très limitante en cela qu'elle fait exploser le temps de calcul. La structure en couches différentiables du réseau de neurones convolutionnels permet d'imaginer un réseau dérivé qui résoudrait ces équations plus efficacement en un passage de l'image et de sa Jacobienne.

Conclusion

Contributions

Dans cette thèse nous avons abordé le problème de la localisation en milieux urbains. Inférer un positionnement précis en ville est important dans nombre d'applications de réalité augmentée ou de robotique mobile mais les capteurs GPS et les centrales inertielles ont cependant une fiabilité limitée dans ces environnements. La vision par ordinateur offre une solution alternative intéressante en estimant la pose d'une caméra relativement à un modèle de la ville. Ainsi nous supposons que le système de localisation dispose d'une caméra qui prend des vues piétonnes de scènes urbaines.

Le modèle de ville que l'on propose est une alternative aux modèles purement géométriques basés sur l'empreinte au sol des bâtiments et aux modèles *features-based* issus d'algorithmes de type *Structure From Motion* trop demandeurs en ressources. Il consiste en un ensemble discret de rectangles géoréférencés modélisant les façades auxquels sont associés des descripteurs compacts mêlant information photométrique et structure sémantique. Ce modèle léger est bien adapté aux contraintes de temps réels inhérentes aux applications de réalité augmentée.

Le problème de localisation globale peut alors se décomposer en deux étapes. Dans un premier temps nous cherchons à reconnaitre les façades visibles de l'image par rapport aux références du modèle. Puis nous inférons la pose de la caméra par rapport aux façades détectées. Cette pose relative combinée aux données géoréférencées du modèle permet alors de calculer un positionnement précis rapidement.

La méthode originale développée dans cette thèse s'inscrit dans ce contexte et vise à dépasser les limitations actuelles du positionnement en terme d'efficacité et de robustesse aux occultations ainsi qu'aux changements de points de vue et d'illumination. Pour cela, l'idée principale est de profiter des progrès récents de l'apprentissage profond par réseaux de neurones convolutionnels pour extraire de l'information de haut-niveau sur laquelle on peut baser des modèles géométriques. Notre approche est donc mixte *Bottom-Up/ Top-Down* et se décompose en trois étapes clés.

Nous proposons tout d'abord une méthode d'estimation de la rotation de la pose de caméra. Les 3 points de fuite principaux des images en milieux urbains, dits points de fuite de Manhattan, sont détectés grâce à un réseau de neurones convolutionnels (CNN) qui fait à la fois une estimation de ces points de fuite mais aussi une segmentation de l'image relativement à eux. Une seconde étape de raffinement utilise ces informations et les segments de l'image dans une formulation

bayésienne pour estimer efficacement et plus précisément ces points. L'estimation de la rotation de la caméra permet de rectifier les images et ainsi de s'affranchir des effets de perspectives pour la recherche de la translation.

Une deuxième contribution est une méthode de détection de façades dans ces images rectifiées qui reconnait également ces façades parmi une base de bâtiments connus afin d'estimer une translation grossière. Dans un souci d'efficacité, nous avons proposé une série d'indices basés sur des caractéristiques spécifiques aux façades (répétitions, symétrie, sémantique) qui permettent de sélectionner rapidement des candidats façades potentiels. Ensuite ceux-ci sont classifiés en façade ou non selon un nouveau descripteur CNN contextuel. Enfin la mise en correspondance des façades détectées avec les références est opérée par une recherche au plus proche voisin relativement à une métrique apprise sur ces descripteurs.

Finalement nous proposons une méthode de raffinement de l'estimation de la translation qui repose sur la segmentation sémantique de l'image par un CNN pour sa robustesse aux changements d'illuminations et aux petites déformations. La façade étant identifiée à l'étape précédente, on adopte une méthode basée modèle par recalage. Comme les problèmes de recalage et de segmentation sont liés, on propose un modèle bayésien qui permet de résoudre conjointement ces deux problèmes. Ce traitement conjoint améliore les résultats de recalage et de segmentation tout en restant efficace en terme de temps de calcul.

Perspectives

Ces trois étapes ont été évaluées et validées statistiquement et qualitativement sur des grands jeux de données de la communauté (York Urban, ZuBuD, ...). Il serait maintenant intéressant de pouvoir évaluer notre méthode de localisation en ville dans sa globalité et sur une base de tests dont les dimensions sont compatibles avec le contexte applicatif. En effet, si les résultats sont encourageants sur ZuBuD, il est difficile de se prononcer sur le passage à l'échelle d'une ville entière sans limitation de la zone par GPS. La capacité de discrimination du descripteur CNN introduit au chapitre 3 est-elle suffisante pour reconnaitre n'importe quelle façade d'une ville ? Si ce n'est pas le cas, elle pourrait être améliorée en exploitant davantage le graphe de structure sémantique des façades du chapitre 3, tout en gagnant en robustesse aux occultations. Celui-ci pourrait ainsi participer à la classification et à l'identification en utilisant par exemple les travaux sur l'apprentissage profond dans des espaces structurés (*Geometric Deep Learning*) [87]. Une autre piste pour améliorer la reconnaissance serait d'intégrer l'information de co-occurrences de façades dans l'image car pour l'instant la reconnaissance comme la pose est faite indépendamment pour chaque façade détectée. Le modèle pourrait alors intégrer des liens topologiques entres façades résumés en graphes.

Mais avant cela il est primordial, pour pouvoir statuer sur le passage à l'échelle de la reconnaissance, de disposer d'une base de tests conséquente. Si une telle base de test n'est pas directement disponible, elle pourrait théoriquement être construite en fusionnant les données géométriques (empreintes au sol et hauteur des bâtiments) de OpenStreetMap et les images piétonnes de Google Street View. Nos premiers essais ont montré que le recalage à partir des données GPS est de qualité variable. L'imprécision en translation peut ainsi décaler fortement

les limites des bâtiments sur l'image ce qui est problématique pour extraire les descripteurs de façades qui constituent en partie notre modèle. Des bases comme TorontoCity [115] ou les données IGN récentes du projet iTowns semblent plus pertinentes pour établir une telle base de tests à l'échelle d'une ville.

Des données plus précises et plus nombreuses bénéficieraient également à l'apprentissage notamment dans le second chapitre. Dans ce même chapitre, plutôt que de se reposer sur des droites uniquement issues de segments de l'image, les *feature maps* du CNN pourraient être utilisées pour caractériser des répétitions linéaires de motifs. On pourrait alors imaginer un apprentissage de bout en bout de l'extraction de primitives à la résolution de l'EM qui peut être convertis en réseaux récurrents différentiables [47]. Un tel apprentissage de bout en bout pourrait également profiter au chapitre 3 pour mieux gérer les interactions entre les paramètres de proposition, de classification et de reconnaissance.

Si l'on a apporté un certain soin à l'implémentation des différentes étapes pour garantir des temps d'exécution raisonnables dans le contexte de la réalité augmentée, plusieurs améliorations pourraient être apportés en plus du portage en C de l'ensemble de la méthode. De nombreuses parties du code pourraient ainsi être parallélisées notamment les algorithmes EM. Le goulot d'étranglement du calcul devrait alors être l'inférence des CNN dont l'architecture VGG est gourmande en ressources. Des architectures réduites pour appareils mobiles ont été récemment proposées qui pourraient fortement réduire le temps d'inférence [53]. Si actuellement la variante du recalage avec inférence de la segmentation en ligne par le CNN est très lente, elle pourrait bénéficier d'une telle architecture en plus d'être calculée à partir d'un réseau dérivé utilisant le Jacobien de l'image plutôt que par différences finies. Il serait alors envisageable d'utiliser notre méthode pour du suivi temporel.

Annexe A

Annexe : Partitionnement optimal de l'espace couleur

L'indice de contraste de couleur est défini comme la différence de la distribution de couleurs à l'intérieur du rectangle considéré et celle dans une région entourant ce rectangle [2]. Pour renforcer cet indice sur les façades, on choisit de biaiser cette distribution en partitionnant non uniformément l'espace couleur LAB. Les partitions de cet espace couleur sont codées par des *Octree* qui subdivisent la boîte initiale ($-100 \leq L \leq 100, -100 \leq A \leq 100, -100 \leq B \leq 100$), et on cherche le meilleur *Octree* $T^\star$ de sorte à maximiser la séparabilité entre les distributions de valeur d'indice $P(s_{color}|\text{façade})$ pour les façades et $P(s_{color}|\text{non-façade})$ pour les non-façades (Ch. 2 Fig. 3.9, à gauche). Cette séparabilité est mesurée par J :

$$J = \sum_{s_{color}} |P(s_{color}|\text{façade}) - P(s_{color}|\text{non-façade})| \tag{A.1}$$

L'optimisation est faite par recuit simulé sur les *Octree* avec des contraintes de profondeur maximale D_{max} et de nombre maximal de feuilles L_{max} pour ne pas dépasser les 256 cases. On définit une mutation $T' = \text{mutate}(T)$ de l'arbre courant T en suivant la procédure suivante :

- Tous les noeuds n_i de l'arbre T sont pondérés par leur profondeur normalisée : $w_i = \frac{\text{depth}(n_i)}{\sum_{n_j \in T} \text{depth}(n_j)}$.
- On choisit un noeud n_i selon les probabilités discrètes w_i.
- Si $\text{depth}(n_i) = D_{max}$ ou si le nombre de feuilles est déjà maximal L_{max}, on fusionne l'arbre au niveau du noeud père de n_i.
- Sinon on fait une modification de l'arbre T au niveau du noeud n_i selon la règle suivante. On a 75% de chance de subdiviser le noeud n_i en 8 noeuds fils et 25% de chance de fusionner son noeud père.

Selon l'algorithme de recuit simulé, cette mutation T' est acceptée si $-J(T') < -J(T)$ ou si $x < \exp^{\frac{J(T')-J(T)}{\text{Temp}}}$ avec x une valeur tirée aléatoirement entre 0 et 1 uniformément. On recommence alors la procédure en baissant la température Temp jusqu'à stabilisation de la solution.

Annexe B

Annexe : résolution du système polynomial $p = 4$

B.1 Système polynomial pour $p = 4$

En notant $B_s = \frac{\partial \tilde{R}}{\partial s}$, $B_{t_x} = \frac{\partial \tilde{R}}{\partial t_x}$ et $B_{t_y} = \frac{\partial \tilde{R}}{\partial t_y}$, le système polynomial complet avec $p = 4$ s'écrit :

$$\begin{cases} B_s\left(s, t_x, t_y\right) = 0 \\ B_{t_x}\left(s, t_x, t_y\right) = 0 \\ B_{t_y}\left(s, t_x, t_y\right) = 0 \end{cases} \tag{B.1}$$

$$\begin{cases} b_0 + b_1 s + b_2 t_x + b_3 t_y + b_4 s t_x + b_5 s t_y + b_6 t_x^2 + b_7 t_y^2 + b_8 s^2 + a_9 s^3 \\ + b_{10} t_x^3 + b_{11} t_y^3 + b_{12} s^2 t_x + b_{13} s^2 t_y + b_{14} s t_x^2 + b_{15} s t_y^2 \\ + b_{16} s t_x^3 + b_{17} s t_y^3 + b_{18} s^3 t_x + b_{19} s^3 t_y + b_{20} s^2 t_x^2 + b_{21} s^2 t_y^2 \\ + b_{22} t_x^4 + b_{23} t_y^4 + b_{24} s^4 + b_{25} s^5 = 0 \\ b_{18} s^3 + b_{20} t_x s^2 + \frac{b_{12}}{2} s^2 + b_{16} s t_x^2 + \frac{2b_{14}}{3} s t_x + \frac{b_4}{3} s \\ b_{22} t_x^3 + \frac{3b_{10}}{4} t_x^2 + \frac{b_6}{2} t_x + \frac{b_2}{4} + b_{26} s^4 = 0 \\ b_{19} s^3 + b_{21} t_y s^2 + \frac{b_{13}}{2} s^2 + b_{17} s t_y^2 + \frac{2b_{15}}{3} s t_y + \frac{b_5}{3} s \\ b_{23} t_y^3 + \frac{3b_{11}}{4} t_y^2 + \frac{b_7}{2} t_y + \frac{b_3}{4} + b_{27} s^4 = 0 \end{cases} \tag{B.2}$$

avec

$$
\begin{aligned}
b_0 &= -\sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{x_i^4}{\sigma_{k_j,x}} + \frac{y_i^4}{\sigma_{k_j,y}} \right) \quad b_1 = 3 \sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{\mu_{k_j,x} x_i^3}{\sigma_{k_j,x}} + \frac{\mu_{k_j,y} y_i^3}{\sigma_{k_j,y}} \right) \\
b_2 &= 4 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{x_i^3}{\sigma_{k_j,x}} \quad b_3 = 4 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{y_i^3}{\sigma_{k_j,y}} \\
b_4 &= -9 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x} x_i^2}{\sigma_{k_j,x}} \quad b_5 = -9 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x} y_i^2}{\sigma_{k_j,y}} \\
b_6 &= -6 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{x_i^2}{\sigma_{k_j,x}} \quad b_7 = -6 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{y_i^2}{\sigma_{k_j,y}} \\
b_8 &= -3 \sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{\mu_{k_j,x}^2 x_i^2}{\sigma_{k_j,x}} + \frac{\mu_{k_j,y}^2 y_i^2}{\sigma_{k_j,y}} \right) \quad b_9 = \sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{\mu_{k_j,x}^3 x_i}{\sigma_{k_j,x}} + \frac{\mu_{k_j,x}^3 y_i}{\sigma_{k_j,y}} \right) \\
b_{10} &= 4 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{x_i}{\sigma_{k_j,x}} \quad b_{11} = 4 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{y_i}{\sigma_{k_j,y}} \\
b_{12} &= 6 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x}^2 x_i}{\sigma_{k_j,x}} \quad b_{13} = 6 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y}^2 y_i}{\sigma_{k_j,y}} \\
b_{14} &= 9 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x} x_i}{\sigma_{k_j,x}} \quad b_{15} = 9 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y} y_i}{\sigma_{k_j,y}} \\
b_{16} &= -3 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x}}{\sigma_{k_j,x}} \quad b_{17} = -3 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y}}{\sigma_{k_j,y}} \\
b_{18} &= -\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x}^3}{\sigma_{k_j,x}} \quad b_{19} = -\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y}^3}{\sigma_{k_j,y}} \\
b_{20} &= -3 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x}^2}{\sigma_{k_j,x}} \quad b_{21} = -3 \sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y}^2}{\sigma_{k_j,y}} \\
b_{22} &= -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{\sigma_{k_j,x}} \quad b_{23} = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{\sigma_{k_j,y}} \\
b_{24} &= -2 \sum_{i,j,k_j} \beta_{i,j,k_j} \quad b_{25} = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{4P(l_j|i,I)} \frac{\partial P(l_j|i,I)}{\partial s} \\
b_{26} &= -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{4P(l_j|i,I)} \frac{\partial P(l_j|i,I)}{\partial t_x} \quad b_{27} = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{4P(l_j|i,I)} \frac{\partial P(l_j|i,I)}{\partial t_y}
\end{aligned}
\tag{B.3}
$$

B.2 Gradient et hessien pour $p = 4$

On résout numériquement le système précédent (Eq. B.2) par une approche itérative de Gauss-Newton. Soit $J = B_s^2 + B_{t_x}^2 + B_{t_y}^2$, on en calcul le gradient g et le Hessien H :

$$
g = 2 \begin{pmatrix} B_s \frac{\partial B_s}{\partial s} + B_{t_x} \frac{\partial B_{t_x}}{\partial s} + B_{t_y} \frac{\partial B_{t_y}}{\partial s} \\ B_s \frac{\partial B_s}{\partial t_x} + B_{t_x} \frac{\partial B_{t_x}}{\partial t_x} + B_{t_y} \frac{\partial B_{t_y}}{\partial t_x} \\ B_s \frac{\partial B_s}{\partial t_y} + B_{t_x} \frac{\partial B_{t_x}}{\partial t_y} + B_{t_y} \frac{\partial B_{t_y}}{\partial t_y} \end{pmatrix}
\tag{B.4}
$$

$$H_{i,j} = 2\left(\frac{\partial B_s}{\partial \theta_i}\frac{\partial B_s}{\partial \theta_j} + B_s\frac{\partial^2 B_s}{\partial \theta_i \partial \theta_j} + \frac{\partial B_{t_x}}{\partial \theta_i}\frac{\partial B_{t_x}}{\partial \theta_j} + B_{t_x}\frac{\partial^2 B_{t_x}}{\partial \theta_i \partial \theta_j} + \frac{\partial B_{t_y}}{\partial \theta_i}\frac{\partial B_{t_y}}{\partial \theta_j} + B_{t_y}\frac{\partial^2 B_{t_y}}{\partial \theta_i \partial \theta_j}\right) \quad \text{(B.5)}$$

avec

$$\begin{aligned}\frac{\partial B_s}{\partial s} = {} & b_1 + b_4 t_x + b_5 t_y + 2s b_8 + 3a_9 s^2 + 2b_{12} s t_x + 2b_{13} s t_y + b_{14} t_x^2 + b_{15} t_y^2 + b_{16} t_x^3 + b_{17} t_y^3 \\ & + 3b_{18} s^2 t_x + 3b_{19} s^2 t_y + 2b_{20} s t_x^2 + 2b_{21} s t_y^2 + 4b_{24} s^3 + 5b_{25} s^4\end{aligned} \quad \text{(B.6)}$$

$$\frac{\partial B_s}{\partial t_x} = b_2 t_x + b_4 s + 2b_6 t_x + 3b_{10} t_x^2 + b_{12} s^2 + 2b_{14} s t_x + 3b_{16} s t_x^2 + b_{18} s^3 + 2b_{20} s^2 t_x + 4b_{22} t_x^3 \quad \text{(B.7)}$$

$$\frac{\partial B_s}{\partial t_y} = b_3 t_y + b_5 s + 2b_7 t_y + 3b_{11} t_y^2 + b_{13} s^2 + 2b_{15} s t_y + 3b_{17} s t_y^2 + b_{19} s^3 + 2b_{21} s^2 t_y + 4b_{23} t_y^3 \quad \text{(B.8)}$$

$$\frac{\partial B_{t_x}}{\partial s} = 3b_{18} s^2 + 2b_{20} t_x s + b_{12} s + b_{16} t_x^2 + \frac{2b_{14}}{3} t_x + \frac{b_4}{3} + 4b_{26} s^3 \quad \text{(B.9)}$$

$$\frac{\partial B_{t_x}}{\partial t_x} = b_{20} s^2 + 2b_{16} s t_x + \frac{2b_{14}}{2} t_x + 3b_{22} t_x^2 + \frac{3b_{10}}{2} t_x + \frac{a_6}{2} \quad \text{(B.10)}$$

$$\frac{\partial B_{t_x}}{\partial t_y} = 0 \quad \text{(B.11)}$$

$$\frac{\partial B_{t_y}}{\partial s} = 3b_{19} s^2 + 2b_{21} t_y s + b_{13} s + b_{17} t_y^2 + \frac{2b_{15}}{3} t_y + \frac{b_5}{3} + 4b_{27} s^3 \quad \text{(B.12)}$$

$$\frac{\partial B_{t_y}}{\partial t_x} = 0 \quad \text{(B.13)}$$

$$\frac{\partial B_{t_y}}{\partial t_y} = b_{21} s^2 + 2b_{17} s t_y + \frac{2b_{15}}{2} t_x + 3b_{23} t_y^2 + \frac{3b_{11}}{2} t_y + \frac{a_7}{2} \quad \text{(B.14)}$$

$$\frac{\partial^2 B_s}{\partial s^2} = 20b_{25} s^3 + 6b_{18} s t_x + 6b_{19} s t_y + 2b_{20} t_x^2 + 2b_{21} t_y^2 + 12b_{24} s^2 + 2b_{12} t_x + 2b_{13} t_y + 6b_9 s + 2b_8 \quad \text{(B.15)}$$

$$\frac{\partial^2 B_s}{\partial s \partial t_x} = 3b_{16} t_x^2 + 3b_{18} s^2 + 4b_{20} s t_x + 2b_{12} s + 2b_{14} t_x + b_4 \quad \text{(B.16)}$$

$$\frac{\partial^2 B_s}{\partial s \partial t_y} = 3b_{17} t_y^2 + 3b_{19} s^2 + 4b_{21} s t_y + 2b_{13} s + 2b_{15} t_y + b_5 \quad \text{(B.17)}$$

$$\frac{\partial^2 B_s}{\partial t_x^2} = 6b_{16} s t_x + 2b_{20} s^2 + 12b_{22} t_x^2 + 6b_{10} t_x + 2b_{14} s + 2b_6 \quad \text{(B.18)}$$

$$\frac{\partial^2 B_s}{\partial t_x \partial t_y} = 0 \quad \text{(B.19)}$$

$$\frac{\partial^2 B_s}{\partial t_y^2} = 6b_{17}st_y + 2b_{21}s^2 + 12b_{23}t_y^2 + 6b_{11}t_y + 2b_{15}s + 2b_7 \tag{B.20}$$

$$\frac{\partial^2 B_{t_x}}{\partial s^2} = 12b_{26}s^2 + 6b_{18}s + 2b_{20}t_x + b_{12} \tag{B.21}$$

$$\frac{\partial^2 B_{t_x}}{\partial s \partial t_x} = 2b_{20}s + 2b_{16}t_x + \frac{2b_{14}}{3} \tag{B.22}$$

$$\frac{\partial^2 B_{t_x}}{\partial s \partial t_y} = 0 \tag{B.23}$$

$$\frac{\partial^2 B_{t_x}}{\partial t_x^2} = 2b_{16}s + 6b_{22}t_x + \frac{3b_{10}}{2} \tag{B.24}$$

$$\frac{\partial^2 B_{t_x}}{\partial t_x \partial t_y} = 0 \tag{B.25}$$

$$\frac{\partial^2 B_{t_x}}{\partial t_y^2} = 0 \tag{B.26}$$

$$\frac{\partial^2 B_{t_y}}{\partial s^2} = 12b_{27}s^2 + 6b_{19}s + 2b_{21}t_y + b_{13} \tag{B.27}$$

$$\frac{\partial^2 B_{t_y}}{\partial s \partial t_x} = 0 \tag{B.28}$$

$$\frac{\partial^2 B_{t_y}}{\partial s \partial t_y} = 2b_{21}s + 2b_{17}t_y + \frac{2b_{15}}{3} \tag{B.29}$$

$$\frac{\partial^2 B_{t_y}}{\partial t_x^2} = 0 \tag{B.30}$$

$$\frac{\partial^2 B_{t_y}}{\partial t_x \partial t_y} = 0 \tag{B.31}$$

$$\frac{\partial^2 B_{t_y}}{\partial t_y^2} = 2b_{17}s + 6b_{23}t_y + \frac{3b_{11}}{2} \tag{B.32}$$

Publications personnelles

[FBS15] Antoine Fond, Marie-Odile Berger et Gilles Simon, « Prior-based Facade Rectification for AR in Urban Environment », *in Proceedings of the IEEE International Symposium on Mixed and Augmented Reality workshop on Urban Augmented Reality*, p. 94–99, 2015.

[FBS16] Antoine Fond, Marie-Odile Berger et Gilles Simon, « Génération d'hypothèses de façades utilisant des critères contextuels et structurels », *in Reconnaissance des Formes et Intelligence Artificielle*, 2016.

[FBS17] Antoine Fond, Marie-Odile Berger et Gilles Simon, « Facade Proposals for Urban Augmented Reality », *in IEEE International Symposium on Mixed and Augmented Reality*, oct. 2017.

[SFB16] Gilles Simon, Antoine Fond et Marie-Odile Berger, « A Simple and Effective Method to Detect Orthogonal Vanishing Points in Uncalibrated Images of Man-Made Environments », *in Proceedings of Eurographics*, 2016.

Bibliographie

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an Object ? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11) :2189–2202, 2012.

[3] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view : Capturing the world at street level. *Computer*, 43(6) :32–38, 2010.

[4] Matthew E Antone and Seth Teller. Automatic recovery of relative camera rotations for urban scenes. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 282–289. IEEE, 2000.

[5] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[6] Anil Armagan, Martin Hirzer, Peter M Roth, and Vincent Lepetit. Accurate camera registration in urban environments using high-level feature matching. In *British Machine Vision Conference*, 2017.

[7] Anil Armagan, Martin Hirzer, Peter M Roth, and Vincent Lepetit. Learning to align semantic segmentation and 2.5 d maps for geolocalization. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[8] Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. Instant Outdoor Localization and SLAM Initialization from 2.5D Maps. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 1309–1318, Fukuoka, Japan, 2015.

[9] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. SegNet : A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling. *CoRR*, abs/1505.07293, 2015.

[10] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[11] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep) :1345–1382, 2005.

[12] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited : 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5965–5974, 2016.

[13] Olga Barinova, Victor Lempitsky, Elena Tretiak, and Pushmeet Kohli. Geometric image parsing in man-made environments. *Computer Vision–ECCV 2010*, pages 57–70, 2010.

[14] Stephen T Barnard. Interpreting perspective images. *Artificial intelligence*, 21(4) :435–462, 1983.

[15] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf : Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[16] Jean-Charles Bazin and Marc Pollefeys. 3-line ransac for orthogonal vanishing point detection. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4282–4287. IEEE, 2012.

[17] Jean-Charles Bazin, Yongduek Seo, Cédric Demonceaux, Pascal Vasseur, Katsushi Ikeuchi, Inso Kweon, and Marc Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 638–645. IEEE, 2012.

[18] Selim Benhimane and Ezio Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 943–948. IEEE, 2004.

[19] Jean-Pascal Burochin, Bruno Vallet, Mathieu Brédif, Clément Mallet, Thomas Brosset, and Nicolas Paparoditis. Detecting blind building façades from highly overlapping wide angle aerial imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96 :193–209, 2014.

[20] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief : Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

[21] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the Devil in the Details : Delving Deep into Convolutional Nets. In *Proceedings of the British Machine Vision Conference*, Nottingam, England, 2014.

[22] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv :1606.00915*, 2016.

[23] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–546, San Diego, USA, 2005.

[24] Hang Chu, Shenlong Wang, Raquel Urtasun, and Sanja Fidler. HouseCraft : Building Houses from Rental Ads and Street Views. In *Proceedings of the European Conference on Computer Vision*, pages 500–516, Amsterdam, The Netherlands, 2016.

[25] Daniel Conrad and Guilherme N DeSouza. Homography-based ground plane detection for mobile robot navigation using a modified em algorithm. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 910–915. IEEE, 2010.

[26] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, number 1-22 in 1, pages 1–2. Prague, 2004.

[27] Mark Cummins and Paul Newman. Fab-map : Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6) :647–665, 2008.

[28] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[29] Amaury Dame and Eric Marchand. Accurate real-time tracking using mutual information. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 47–56. IEEE, 2010.

[30] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[31] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*, pages 197–210. Springer, 2008.

[32] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv :1606.03798*, 2016.

[33] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[34] Georgios D Evangelidis and Radu Horaud. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[35] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8) :1915–1929, 2013.

[36] Antoine Fond, Marie-Odile Berger, and Gilles Simon. Prior-based Facade Rectification for AR in Urban Environment. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality workshop on Urban Augmented Reality*, pages 94–99, Fukuoka, Japan, 2015.

[37] Antoine Fond, Marie-Odile Berger, and Gilles Simon. Facade Proposals for Urban Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality*, Nantes, France, October 2017.

[38] Jean-Sébastien Franco and Edmond Boyer. Learning temporally consistent rigidities. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1241–1248. IEEE, 2011.

[39] Björn Fröhlich, Erik Rodner, and Joachim Denzler. A Fast Approach for Pixelwise Labeling of Facade Images. In *Proceedings of the International Conference on Pattern Recognition*, pages 3029–3032, Istanbul, Turkey, 2010.

[40] Kunihiko Fukushima and Sei Miyake. Neocognitron : A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[41] Raghudeep Gadde, Varun Jampani, Renaud Marlet, and Peter Gehler. Efficient 2D and 3D Facade Segmentation using Auto-Context. *CoRR*, abs/1606.06437, 2016.

[42] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2) :291–298, 1994.

[43] Bernard Ghanem, Ali Thabet, Juan Carlos Niebles, and Fabian Caba Heilbron. Robust manhattan frame estimation from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3772–3780, 2015.

[44] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[45] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[47] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in neural information processing systems*, 2017.

[48] Gregory D Hager and Peter N Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE transactions on pattern analysis and machine intelligence*, 20(10) :1025–1039, 1998.

[49] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *Proceedings of the European Conference on Computer Vision*, pages 346–361, Zurich, Switzerland, 2014.

[51] Radu Horaud, Florence Forbes, Manuel Yguel, Guillaume Dewaele, and Jian Zhang. Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3) :587–602, 2011.

[52] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How Good are Detection Proposals, really ? In *Proceedings of the British Machine Vision Conference*, Nottingham, England, 2014.

[53] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.

[54] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1) :106–154, 1962.

[55] Du Q Huynh. Metrics for 3d rotations : Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2) :155–164, 2009.

[56] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4) :107, 2017.

[57] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. *arXiv preprint arXiv :1408.5093*, 2014.

[58] Frédéric Jurie and Michel Dhome. Real time robust template matching. In *BMVC*, pages 1–10, 2002.

[59] Jayashree Karlekar, Steven Zhiying Zhou, Weiquan Lu, Loh Zhi Chang, Yuta Nakayama, and Daniel Hii. Positioning, tracking and mapping for outdoor augmentation. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 175–184, Seoul, Korea, 2010.

[60] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet : A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[61] Jeongtae Kim and Jeffrey A Fessler. Intensity-based image registration using robust correlation coefficients. *IEEE transactions on medical imaging*, 23(11) :1430–1444, 2004.

[62] Nikolay Kobyshev, Hayko Riemenschneider, and Luc Van Gool. Matching features correctly through semantic understanding. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 472–479. IEEE, 2014.

[63] Jana Košecká and Wei Zhang. Extraction, Matching, and Pose Recovery Based on Dominant Rectangular Structures. *Computer Vision and Image Understanding*, 100(3) :274–293, 2005.

[64] Mateusz Kozinski, Raghudeep Gadde, Sergey Zagoruyko, Guillaume Obozinski, and Renaud Marlet. A mrf shape prior for facade parsing with occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2015.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q.

Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[67] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[68] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[69] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv :1609.04802*, 2016.

[70] José Lezama, Rafael Grompone von Gioi, Gregory Randall, and Jean-Michel Morel. Finding vanishing points via point alignments in image primal and dual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 509–515, 2014.

[71] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[72] Fei Liu and Stefan Seipel. Detection of Facade Regions in Street View Images from Split-and-Merge of Perspective Patches. *Journal of Image and Graphics*, 2(1) :8–14, 2014.

[73] Jingchen Liu and Yanxi Liu. Local Regularity-Driven City-Scale Facade Detection from Aerial Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3778–3785, Columbus, USA, 2014.

[74] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[75] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, November 2004.

[76] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition : A survey. *IEEE Transactions on Robotics*, 32(1) :1–19, 2016.

[77] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 647–679. Vancouver, BC, Canada, 1981.

[78] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10) :1331–1398, 2012.

[79] Stéphane Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065) :20150203, 2016.

[80] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality : a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12) :2633–2651, 2016.

[81] Andelo Martinovic, Markus Mathias, Julien Weissenberg, and Luc J. Van Gool. A Three-Layered Approach to Facade Parsing. In *Proceedings of the European Conference on Computer Vision*, pages 416–429, Florence, Italy, 2012.

[82] David Mattes, David R Haynor, Hubert Vesselle, Thomas K Lewellen, and William Eubank. Nonrigid multimodality image registration. *Medical imaging*, 4322(1) :1609–1620, 2001.

[83] Branislav Micusík, Horst Wildenauer, and Jana Kosecka. Detection and Matching of Rectilinear Structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, Anchorage, USA, 2008.

[84] Michael J Milford and Gordon F Wyeth. Seqslam : Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

[85] Faraz M Mirzaei and Stergios I Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2454–2461. IEEE, 2011.

[86] Mahesh Mohan, Dorian Gálvez-López, Claire Monteleoni, and Gabe Sibley. Environment selection and hierarchical place recognition. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5487–5494. IEEE, 2015.

[87] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. *arXiv preprint arXiv :1611.08402*, 2016.

[88] David Ok, Mateusz Kozinski, Renaud Marlet, and Nikos Paragios. High-level bottom-up cues for top-down parsing of facade images. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 128–135. IEEE, 2012.

[89] Aude Oliva and Antonio Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3) :145–175, 2001.

[90] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. *Computer Vision–ECCV 2010*, pages 143–156, 2010.

[91] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images : a survey. *IEEE transactions on medical imaging*, 22(8) :986–1004, 2003.

[92] B Srinivasa Reddy and Biswanath N Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8) :1266–1271, 1996.

[93] Gerhard Reitmayr and Tom Drummond. Going out : Robust Model-based Tracking for Outdoor Augmented Reality. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 109–118, Santa Barbara, USA, 2006.

[94] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Convolutional neural network architecture for geometric matching. *arXiv preprint arXiv :1703.05593*, 2017.

[95] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[96] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443, 2006.

[97] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb : An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[98] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3) :211–252, 2015.

[99] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[100] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1) :1–10, 1966.

[101] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A Simple and Effective Method to Detect Orthogonal Vanishing Points in Uncalibrated Images of Man-Made Environments. In *Proceedings of Eurographics*, Lisbon, Portugal, 2016.

[102] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[103] Josef Sivic and Andrew Zisserman. Video google : A text retrieval approach to object matching in videos. In *Computer Vision (ICCV), 2003 IEEE International Conference on*, page 1470. IEEE, 2003.

[104] Rahunathan Smriti, D Stredney, P Schmalbrock, and BD Clymer. Image registration using rigid registration and maximization of mutual information. In *MMVR13. The 13th Annual Medicine Meets Virtual Reality Conference, Long Beach, CA*, page 74, 2005.

[105] Niko Suenderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place Recognition with ConvNet Landmarks : Viewpoint-Robust, Condition-Robust, Training-Free. In *Proceedings of Robotics : Science and Systems*, Rome, Italy, 2015.

[106] Jean-Philippe Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1250–1257. IEEE, 2009.

[107] Olivier Teboul, Iasonas Kokkinos, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Parsing facades with shape grammars and reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7) :1744–1756, 2013.

[108] Carl Toft, Carl Olsson, Fredrik Kahl, Daniele De Gregorio, Tommaso Cavallari, Luigi Di Stefano, Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, et al. Long-term 3d localization and pose from semantic labellings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 650–659, 2017.

[109] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 883–890, 2013.

[110] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2) :154, 2013.

[111] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv preprint arXiv :1711.10925*, 2017.

[112] Paul Viola and Michael J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2) :137–154, 2004.

[113] Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2) :137–154, 1997.

[114] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd : A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4) :722–732, 2010.

[115] Shenlong Wang, Min Bai, Gellert Mattyus, Hang Chu, Wenjie Luo, Bin Yang, Justin Liang, Joel Cheverie, Sanja Fidler, and Raquel Urtasun. Torontocity : Seeing the world with a million eyes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3017, 2017.

[116] Yiliang Xu, Sangmin Oh, and Anthony Hoogs. A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1376–1383, 2013.

[117] Chao Yang, Tian Han, Long Quan, and Chiew-Lan Tai. Parsing facade with rank-one approximation. In *CVPR*, pages 1720–1727, 2012.

[118] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift : Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[119] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5657–5665, 2016.

[120] Zhengdong Zhang, Arvind Ganesh, Xiao Liang, and Yi Ma. TILT : Transform invariant low-rank textures. *International Journal of Computer Vision*, 99(1) :1–24, Aug 2012.

[121] Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann LeCun. Stacked what-where auto-encoders. *arXiv preprint arXiv :1506.02351*, 2015.

[122] C. Lawrence Zitnick and Piotr Dollár. Edge Boxes : Locating Object Proposals from Edges. In *Proceedings of the European Conference on Computer Vision*, pages 391–405, Zurich, Switzerland, 2014.

[123] Siavash Zokai and George Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Transactions on Image Processing*, 14(10) :1422–1434, 2005.

Résumé

Dans cette thèse on aborde le problème de la localisation en milieux urbains. Inférer un positionnement précis en ville est important dans nombre d'applications comme la réalité augmentée ou la robotique mobile. Or les systèmes basés sur des capteurs inertiels (IMU) sont sujets à des dérives importantes et les données GPS peuvent souffrir d'un effet de vallée qui limite leur précision.

Une solution naturelle est de s'appuyer le calcul de pose de caméra en vision par ordinateur. On remarque que les bâtiments sont les repères visuels principaux de l'humain mais aussi des objets d'intérêt pour les applications de réalité augmentée. On cherche donc à partir d'une seule image à calculer la pose de la caméra par rapport à une base de données de bâtiments références connus.

On décompose le problème en deux parties : trouver les références visibles dans l'image courante (reconnaissance de lieux) et calculer la pose de la caméra par rapport à eux. Les approches classiques de ces deux sous-problèmes sont mises en difficultés dans les environnements urbains à cause des forts effets perspectives, des répétitions fréquentes et de la similarité visuelle entre façades.

Si des approches spécifiques à ces environnements ont été développés qui exploitent la grande régularité structurelle de tels milieux, elles souffrent encore d'un certain nombre de limitations autant pour la détection et la reconnaissance de façades que pour le calcul de pose par recalage de modèle.

La méthode originale développée dans cette thèse s'inscrit dans ces approches spécifiques et vise à dépasser ces limitations en terme d'efficacité et de robustesse aux occultations, aux changements de points de vue et d'illumination. Pour cela, l'idée principale est de profiter des progrès récents de l'apprentissage profond par réseaux de neurones convolutionnels pour extraire de l'information de haut-niveau sur laquelle on peut baser des modèles géométriques. Notre approche est donc mixte *Bottom-Up/Top-Down* et se décompose en trois étapes clés.

Nous proposons tout d'abord une méthode d'estimation de la rotation de la pose de caméra. Les 3 points de fuite principaux des images en milieux urbains, dits points de fuite de Manhattan sont détectés grâce à un réseau de neurones convolutionnels (CNN) qui fait à la fois une estimation de ces points de fuite mais aussi une segmentation de l'image relativement à eux. Une second étape de raffinement utilise ces informations et les segments de l'image dans une formulation bayésienne pour estimer efficacement et plus précisément ces points. L'estimation de la rotation de la caméra permet de rectifier les images et ainsi s'affranchir des effets de perspectives pour la recherche de la translation.

Dans une seconde contribution, nous visons ainsi à détecter les façades dans ces images rectifiées et à les reconnaître parmi une base de bâtiments connus afin d'estimer une translation grossière. Dans un soucis d'efficacité, on a proposé une série d'indices basés sur des caractéristiques spécifiques aux façades (répétitions, symétrie, sémantique) qui permettent de sélectionner rapidement des candidats façades potentiels. Ensuite ceux-ci sont classifiés en façade ou non selon un nouveau descripteur CNN contextuel. Enfin la mise en correspondance des façades détectées avec les références est opérée par un recherche au plus proche voisin relativement à une métrique apprise sur ces descripteurs.

Finalement nous proposons une méthode de raffinement de l'estimation de la translation qui repose sur la segmentation sémantique de l'image par un CNN pour sa robustesse aux changements d'illuminations et aux petites déformations. La façade étant identifiée à l'étape précédente, on adopte une méthode basée modèle par recalage. Comme les problèmes de recalage et de segmentation sont liés, on propose un modèle bayésien qui permet de résoudre conjointement ces deux problèmes. Ce traitement conjoint améliore les résultats de recalage et de segmentation tout en restant efficace en terme de temps de calcul.

Ces trois parties ont fait l'objet de validations sur des jeux de données conséquents de la communauté. Les résultats montrent que notre approche est rapide et plus robuste aux changements de conditions de prise de vue que les méthodes précédentes.

Mots-clés : Vision par ordinateur, Apprentissage automatique, Réseaux de neurones, Modèles Bayésiens, Détection d'objets, Reconnaissance de lieux, Points de fuite

Abstract

This thesis addresses the problem of localization in urban areas. Inferring accurate positioning in the city is important in many applications such as augmented reality or mobile robotics. However, systems based on inertial sensors (IMUs) are subject to significant drifts and GPS data can suffer from a valley effect that limits their accuracy.

A natural solution is to rely on the camera pose estimation in computer vision. We notice that buildings are the main visual landmarks of human beings but also objects of interest for augmented reality applications. We therefore aim to compute the camera pose relatively to a database of known reference buildings from a single image.

The problem is twofold : find the visible references in the current image (place recognition) and compute the camera pose relatively to them. Conventional approaches to these two sub-problems are challenged in urban environments due to strong perspective effects, frequent repetitions and visual similarity between facades.

While specific approaches to these environments have been developed that exploit the high structural regularity of such environments, they still suffer from a number of limitations in terms of detection and recognition of facades as well as pose computation through model registration.

The original method developed in this thesis is part of these specific approaches and aims to overcome these limitations in terms of effectiveness and robustness to clutter and changes of viewpoints and illumination. For do so, the main idea is to take advantage of recent advances in deep learning by convolutional neural networks to extract high-level information on which geometric models can be based. Our approach is thus mixed Bottom-Up/Top-Down and is divided into three key stages.

We first propose a method to estimate the rotation of the camera pose. The 3 main vanishing points of the image of urban environnement, known as Manhattan vanishing points, are detected by a convolutional neural network (CNN) that estimates both these vanishing points and the image segmentation relative to them. A second refinement step uses this information and image segmentation in a Bayesian model to estimate these points effectively and more accurately. By estimating the camera's rotation, the images can be rectified and thus free from perspective effects to find the translation.

In a second contribution, we aim to detect the facades in these rectified images to recognize them among a database of known buildings and estimate a rough translation. For the sake of efficiency, a series of cues based on facade specific characteristics (repetitions, symmetry, semantics) have been proposed to enable the fast selection of facade proposals. Then they are classified as facade or non-facade according to a new contextual CNN descriptor. Finally, the matching of the detected facades to the references is done by a nearest neighbor search using a metric learned on these descriptors.

Eventually we propose a method to refine the estimation of the translation relying on the semantic segmentation infered by a CNN for its robustness to changes of illumination ans small deformations. If we can already estimate a rough translation from these detected facades, we choose to refine this result by relying on the semantic segmentation of the image inferred from a CNN for its robustness to changes of illuminations and small deformations. Since the facade is identified in the previous step, we adopt a model-based approach by registration. Since the problems of registration and segmentation are linked, a Bayesian model is proposed which enables both problems to be jointly solved. This joint processing improves the results of registration and segmentation while remaining efficient in terms of computation time.

These three parts have been validated on consistent community data sets. The results show that our approach is fast and more robust to changes in shooting conditions than previous methods.

Keywords : Computer vision, Machine learning, Neural networks, Bayesian models, Objects detection, Place recognition, Vanishing points