AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Synthèse de vues pour l'initialisation de pose

THÈSE

présentée et soutenue publiquement le -

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Pierre Rolin

Composition du jury

Président :	El Mustapha Mouaddib, *professeur, Université de Picardie Jules Verne*
Rapporteurs :	Thierry Chateau, *professeur, Université Blaise Pascal* Eric Marchand, *professeur, Université de Rennes 1*
Examinateurs :	Marie-Odile Berger, *directrice de recherche, INRIA* Séverine Dubuisson, *maître de conférences, UPMC Sorbonne* Didier Galmiche, *professeur, Université de Lorraine* Frédéric Sur, *maître de conférences, Université de Lorraine*

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

Mis en page avec la classe thesul.

Remerciements

Je remercie en premier lieu les membres de mon jury de thèse pour l'intérêt qu'ils ont porté à mes travaux. Je remercie les rapporteurs Eric Marchand et Thierry Chateau pour le soin qu'ils ont accordé à la relecture de ma thèse et la pertinence de leurs rapports. Je remercie Didier Galmiche pour son suivi et ses conseils, depuis que je suis arrivé en master à Nancy. De façon générale, je remercie tous les examinateurs pour leurs questions qui ont fait de la soutenance un moment d'échange agréable et constructif.

Je remercie tout particulièrement mes encadrants de thèse, Marie-Odile et Frédéric, pour leur implication dans cette thèse, l'intérêt qu'ils ont porté à mes travaux et les nombreux conseils qu'ils m'ont donné tout au long de ces trois années. Ils m'ont permis de découvrir le monde de la recherche dans des conditions idéales et je leur en suis immensément reconnaissant.

Je remercie également tous les autres membres de l'équipe Magrit pour leur accueil chaleureux, Pierre-Frédéric, Gilles, Erwan et Brigitte, ainsi que tous les personnes avec qui j'ai partagé le bureau au cours de la thèse : Ahmed, Mathieu, Charlotte, Antoine, Cong et Vincent. Je remercie également l'équipe du département informatique de l'école des Mines de Nancy pour m'avoir permis de donner mes premiers cours dans des conditions exceptionnelles.

Mes remerciement vont également à ma famille, mes parents et mon frère en particulier, pour leur soutient moral et matériel pendant ces années de thèse, ainsi que pour leurs tentatives répétées pour comprendre ce sur quoi je travaille. Enfin je remercie mes amis pour toutes les soirées, vacances et autres réjouissances.

Table des matières

Chapitre 1
Initialisation de la pose d'une caméra

1.1 Calcul de pose . . . 2
1.1.1 Nature du modèle . . . 2
1.1.2 Calcul de pose . . . 4
1.1.3 Mise en correspondance par descripteurs . . . 5
1.2 Correspondances image-modèle . . . 6
1.2.1 Recherche de correspondances . . . 6
1.2.2 Filtrage des correspondances . . . 6
1.2.3 Mise en correspondance avec des modèles de grande taille . . . 7
1.3 Synthèse de vues pour la mise en correspondance . . . 7
1.3.1 Invariance limitée des descripteurs . . . 7
1.3.2 Descripteurs extraits de patchs rectifiés . . . 8
1.3.3 Transformations d'images . . . 8
1.3.4 Rendu à partir du modèle . . . 10
1.3.5 Difficultés de mise en œuvre . . . 10
1.4 Contributions . . . 10
1.4.1 Synthèse de patchs . . . 11
1.4.2 Mise en correspondance robuste . . . 12
1.4.3 Résultats expérimentaux . . . 12
1.4.4 Perspectives . . . 13

Chapitre 2
Synthèse de vues

2.1 Synthèse de vues sans modèle de scène . . . 15
2.1.1 Classification . . . 16

2.1.2 Mise en correspondance d'images 17
2.2 Synthèse de vues avec un modèle 18
2.2.1 Rendu 3D 18
2.2.2 Rectification 19
2.2.3 Simulation de patchs 20
2.3 Problèmes ouverts 21
2.3.1 Synthèse de vues pour un nuage de points non dense 21
2.3.2 Synthèse de vues pour l'initialisation de pose 21
2.3.3 Temps de calcul 21

Chapitre 3
Synthèse de patchs guidée par la géométrie

3.1 Synthèse de vue dans un monde localement plan 23
3.1.1 Transformations induites par un changement de point de vue 24
3.1.2 Approximation affine 25
3.2 Mise en œuvre 26
3.2.1 Construction du modèle 26
3.2.2 Ajout de descripteurs à partir de vues synthétiques 26
3.2.3 Estimation de la pose 27
3.3 Étude expérimentale 28
3.3.1 Protocole expérimental 28
3.3.2 Amélioration du calcul de pose dans les modèles enrichis 30
3.3.3 Comparaison des modèles affine et homographique 37
3.3.4 Ambiguïté due aux vues symétriques 37
3.4 Conclusion 40

Chapitre 4
Recherche efficace de correspondances

4.1 RANSAC et accélérations potentielles 44
4.1.1 RANSAC standard 44
4.1.2 Famille des méthodes RANSAC 45
4.2 PROSAC 49
4.3 Méthode proposée 51
4.3.1 Classement a priori des correspondances 52
4.3.2 Stratégie de tirage 53

4.3.3 Arrêt anticipé . . . 55
4.3.4 Paramètres de la méthode . . . 56
4.4 Résultats . . . 56
4.4.1 Classement des correspondances . . . 58
4.4.2 Temps de calcul . . . 60
4.5 Conclusion . . . 64

Chapitre 5
Synthèse de vues efficace

5.1 Segmentation du modèle en patchs plans . . . 66
5.2 Positionnement des points de vue virtuels . . . 69
5.2.1 Transition tilt . . . 70
5.2.2 Échantillonnage autour d'un patch . . . 71
5.3 Visibilité dans un nuage de points . . . 73
5.3.1 Visibilité à partir des caméras de construction . . . 75
5.3.2 Visibilité à partir des caméras virtuelles . . . 75
5.4 Transformation des vues . . . 78
5.5 Résultats . . . 80
5.6 Conclusion . . . 82

Chapitre 6
Résultats de calculs de pose dans des environnements variés

6.1 Protocole expérimental . . . 83
6.2 Expériences . . . 86
6.2.1 Séquence poster . . . 86
6.2.2 Séquence bureau . . . 88
6.2.3 Séquence livre . . . 90
6.2.4 Séquence pot . . . 92
6.2.5 Séquence tour . . . 94
6.2.6 Séquence place . . . 96
6.2.7 Séquence CAB . . . 98
6.3 Conclusion . . . 100

Chapitre 7
Conclusion et perspectives

7.1 Conclusion 101
7.2 Représentation compacte du modèle 102
7.2.1 Réduire une classe à quelques représentants 104
7.2.2 Réduction de dimension 104
7.2.3 Approximation par morceaux 105
7.3 Approche incrémentale 105
7.4 Synthèse à différentes profondeurs 106
7.5 Modèles non connexes 107
7.6 Deep learning 107

Bibliographie **109**

Chapitre 1

Initialisation de la pose d'une caméra

Ce chapitre introduit le problème de la localisation d'une caméra à partir d'une image seule et d'un modèle de scène. La localisation est un problème récurrent de la vision par ordinateur, avec des applications dans des domaines multiples tels que la robotique [Charmette et al., 2016], la réalité augmentée [Lowe, 1999] ou la reconnaissance d'objets [Collet et al., 2009]. On s'intéresse en particulier au problème d'initialisation de la pose, c'est-à-dire quand il n'y a pas d'information a priori sur la position de la caméra, par opposition au problème de suivi de pose, pour lequel la position de la caméra au pas de temps précédent est connue. Puisqu'on n'a pas d'a priori sur la position de la caméra, l'estimation de la pose s'appuie entièrement sur la recherche de correspondances entre l'image et le modèle de la scène. Cette mise en correspondance peut être mise en défaut lorsque l'image de la scène dont on cherche la pose présente de fortes différences avec les observations antérieures : des variations d'illumination, des différences d'échelle ou des forts changements de point de vue par exemple.

De nombreuses applications utilisent la localisation à partir d'images. En robotique la localisation est une tâche courante, que ce soit pour localiser un robot dans un environnement maîtrisé [Charmette et al., 2016] ou bien géo-localiser une voiture dans une ville [Garcia-Fidalgo and Ortiz, 2015]. Bien qu'il soit possible de s'appuyer sur une grande variété de capteurs, tels qu'un LiDAR ou un GPS par exemple, la vision reste un outil de choix pour la localisation. En effet le positionnement par GPS est largement moins précis que le positionnement par vision, et parfois impossible en milieu urbain à cause des problèmes de perte du signal GPS. Les capteurs utilisant des lasers sont généralement plus précis mais aussi plus coûteux à mettre en place, et leur fonctionnement peut être perturbé par certaines conditions extérieures. La réalité augmentée est un autre contexte qui nécessite la localisation de caméras à partir d'images [Lowe, 1999, Billinghurst et al., 2015]. L'objectif de la réalité augmentée est d'ajouter des éléments virtuels dans des images réelles. Lorsque ces éléments virtuels sont des objets 3D il est nécessaire de connaître la pose de la caméra pour les intégrer de façon réaliste car leur apparence dépend du point de vue sous lequel ils sont observés. Comme le problème consiste à intégrer des objets virtuels dans des images, il est naturel de chercher à obtenir la pose de la caméra directement à partir de ces images.

Dans la plupart de ces applications, le modèle de scène a été construit avant la phase de localisation, généralement par une autre personne que l'utilisateur qui cherche à se

localiser dans d'autres conditions d'acquisition. Par conséquent, la scène observée peut être très différente du modèle appris, à cause de variations de l'illumination, de l'appareil utilisé ou du point de vue par exemple. Comme la construction du modèle et son utilisation ne sont pas faites dans les mêmes conditions, il est nécessaire de proposer une méthode de calcul de pose qui soit robuste à ces changements de conditions. Dans cette thèse, parmi ces changements de conditions potentiels, on s'intéresse en particulier aux changements de point de vue.

Le calcul de pose à partir d'un modèle 3D utilise souvent l'appariement de points d'intérêt sur la base de descripteurs photométriques locaux. Pour faciliter l'appariement, ces descripteurs doivent être robustes à des changements d'illumination et à certaines transformations géométriques correspondant à des changements d'apparence locale de la scène. Pour assurer la qualité des correspondances entre l'image et le modèle, deux approches sont possibles : construire un descripteur de points d'intérêt suffisamment robuste pour que la mise en correspondance soit correcte, ou filtrer les correspondances. Ces deux approches sont utilisées conjointement par les méthodes de calcul de pose car indépendamment elles ne suffisent pas à assurer une mise en correspondance robuste : les descripteurs assurent la cohérence photométrique des correspondances et filtrer les correspondances permet de vérifier leur cohérence géométrique.

Ce chapitre présente les difficultés de mise en correspondance image-modèle, les solutions qui y ont été apportées dans la littérature et les limites de ces solutions. Il est organisé comme suit. La section 1.1 présente les méthodes de calcul de pose classiques, la section 1.2 discute le problème de la sélection robuste de correspondances correctes, la section 1.3 présente des méthodes utilisant la synthèse de vues pour améliorer la mise en correspondance image-modèle et la section 1.4 introduit les contributions qui ont été développées dans cette thèse.

1.1 Calcul de pose

Cette section décrit le processus de calcul de pose à partir d'un modèle non-structuré, à savoir la création du modèle, la recherche de correspondances et l'estimation de la pose elle-même.

1.1.1 Nature du modèle

À partir d'une collection d'images d'une scène on peut construire plusieurs types de modèles. Un premier type de modèle consiste en un ensemble de vues-clés souvent organisées dans une structure hiérarchique, chacune associée à une pose, comme dans [Rucklidge, 1995, Klein and Murray, 2008] par exemple. Ce type de modèle nécessite des observations relativement denses de la scène de sorte que les vues-clés puissent facilement être mises en correspondance avec de nouvelles vues. [Klein and Murray, 2008] proposent de sous-échantillonner et flouter les vues clés et d'utiliser les images résultantes comme descripteurs.

À partir d'une collection d'images il est également possible de construire un nuage de points par *structure from motion*. Le processus, décrit dans [Hartley and Zisserman, 2004,



FIGURE 1.1 – Un nuage de points reconstruit à partir d'un ensemble d'images. On cherche à déterminer la pose associée à l'image encadrée en rouge et dont le point de vue est sensiblement différent des images utilisées pour construire le modèle.

Agarwal et al., 2011], est le suivant. Pour chaque couple d'images possible, on recherche des correspondances entre les points de ces images. Les chaînes de points 2D ainsi mis en correspondances permettent d'estimer simultanément la position des points 3D et la pose des caméras correspondant aux différentes images, par ajustement de faisceaux. L'ensemble des descripteurs associés à chaque point est conservé dans le modèle final. Le modèle construit est donc un nuage de points, chaque point étant associé à un ensemble de descripteurs. Ces descripteurs sont ceux extraits dans les images de construction au niveau de l'image de ce point. La figure 1.1 illustre un modèle de ce type et les images qui ont servi à le construire.

Ce procédé correspond à la construction d'un modèle à partir d'un ensemble d'images données, mais il est également possible de construire itérativement le modèle à partir d'un flux d'images [Davison, 2003,Royer et al., 2007]. Construire le modèle itérativement facilite la mise en correspondance, puisque des images successives correspondent à des points de vue relativement proches, mais le procédé est moins robuste, puisque la reconstruction est susceptible de dériver. Dans le cadre de cette thèse la construction du modèle est vue comme une étape préliminaire en boite noire, on se limite donc à des modèles obtenus par les approches standard décrites ci-dessus. Comme construire le modèle itérativement peut potentiellement dégrader la qualité de la reconstruction on utilise des modèles reconstruit en une fois à partir d'une collection d'images.

Comme la construction du modèle utilise la mise en correspondance d'images, les zones bien texturées de la scène sont mieux reconstruites que les zones sans texture. Au delà des problèmes de textures, les points de vue des images de construction influent largement sur la qualité du modèle : s'ils sont trop éloignés les uns des autres on a peu de correspondances et donc peu de points reconstruits, s'il sont trop proches l'ajustement de faisceaux a une grande erreur de parallaxe. Du fait de ces problèmes la qualité des modèles obtenus peut varier considérablement, en particulier la densité des point reconstruits et la précision avec laquelle ils approximent la surface de la scène.

1.1.2 Calcul de pose

Il existe essentiellement deux familles de méthodes pour calculer une pose à partir d'une image et d'un modèle 3D. Elles se distinguent par l'approche utilisée pour trouver des correspondances entre l'image et le modèle : soit commencer par chercher des correspondances entre l'image requête et les images de construction, soit directement mettre en correspondance des points de l'image requête avec des points du modèle.

La première possibilité consiste à représenter le modèle par une collection d'images-clés de la scène, chacune associée à une pose [Robertson and Cipolla, 2004,Zhang and Kosecka, 2006]. Trouver la pose d'une nouvelle vue revient alors à chercher la ou les images-clés les plus proches pour en déduire la pose. Ce type de méthode est difficile à rendre robuste. Mettre en correspondance des images complètes de la scène peut en effet facilement devenir problématique lorsque l'apparence de la scène est modifiée localement, par exemple par la présence de nouveaux objets. Le manque éventuel d'images-clés peut également rendre la localisation difficile voire impossible. Cependant des méthodes sont développées pour dépasser ces difficultés. Dans le cadre de l'asservissement visuel, par exemple, [Crombez et al., 2015] proposent de modéliser une image par une mixture de gaussiennes, ce qui

permet d'élargir significativement les bassins de convergence. Cela permet d'estimer une pose correcte à partir d'une estimation initiale relativement lointaine.

Il est généralement plus simple de décrire de façon robuste de petites parties de la scène, d'où l'idée d'utiliser des descripteurs locaux pour représenter la scène plutôt que des vues complètes. Calculer une pose consiste dans ce cas à trouver des correspondances entre des points 2D de l'image et des points 3D du modèle de scène. Le problème consistant à déduire une pose de ces correspondances est appelé Perspective-n-Points ou PnP et de nombreuses approches ont été proposées pour le résoudre efficacement [Hesch and Roumeliotis, 2011, Lepetit et al., 2009]. Ces approches ne sont cependant pas robustes car les correspondances qu'elles utilisent doivent être précises, c'est-à-dire effectivement lier deux points qui représentent le même point réel [Lepetit and Fua, 2005]. Par conséquent elles sont généralement utilisées en combinaison avec un filtrage robuste des correspondances.

1.1.3 Mise en correspondance par descripteurs

Comme notre modèle de scène est un nuage de points, la recherche de correspondances s'appuie sur des descripteurs de points d'intérêt locaux pour mettre en correspondance l'image et le modèle. Deux problèmes se posent ici : identifier les points ou zones d'intérêt et leur associer un descripteur. La difficulté est de trouver des descripteurs qui sont à la fois caractéristiques du point d'intérêt tout en étant invariants à certaines transformations telles que des changements d'échelle ou d'illumination par exemple.

Le premier problème consiste à identifier des points caractéristiques. L'objectif de ces détecteurs est d'identifier des points qui peuvent être reconnus même après avoir été partiellement transformés. Les coins de l'image apparaissent naturellement comme de bons candidats. Un des premiers détecteurs de points d'intérêt est donc le détecteur de coins de Harris [Harris and Stephens, 1988]. Les points détectés ne sont cependant pas stables pour certaines transformations, les changements d'échelle en particulier. Pour obtenir cette invariance, [Lindeberg, 1994] proposent de rechercher des extrema locaux dans l'espace échelle d'une pyramide d'un Laplacien de Gaussienne. [Lowe, 1999] proposent d'approximer le Laplacien de Gaussienne par une Différence de Gaussienne. Les points détectés par cette dernière méthode sont effectivement invariants par changement d'échelle et par rotation dans une large mesure.

Pour pouvoir mettre en correspondance des points d'intérêt il faut leur associer un descripteur, c'est à dire un vecteur calculé sur un voisinage du point et qui en est caractéristique. Le descripteur SIFT présenté dans [Lowe, 2004] est un histogramme de gradients orientés, calculé à une échelle déterminée comme extremum de l'espace échelle et par rapport à l'orientation dominante localement du gradient. Cette échelle est donnée par le détecteur de point précédemment mentionné. Utilisé avec le détecteur décrit ci-dessus, ce descripteur est lui-même construit pour être invariant par changement d'échelle et rotation. Le temps de calcul associé peut être problématique dans le cadre d'applications temps réel, mais le descripteur SIFT est celui qui permet la meilleure qualité de mise en correspondance [Moreels and Perona, 2007, Heinly et al., 2012]. Dans le cadre de l'initialisation de la pose, dans la mesure où il n'est pas requis de faire les calculs en temps réel, le descripteur SIFT est adapté à notre application et c'est donc celui là qui est utilisé tout au long de cette thèse. On note que les réseaux de neurones convolutifs permettent

de définir de nouveaux type de descripteurs qui dépassent parfois les performances SIFT. Dans le cadre de l'estimation de pose de véhicule par exemple, [Chabot et al., 2015] proposent d'utiliser des réseaux de neurones convolutifs pour détecter des point d'intérêt et calculer des descripteurs. Ce type d'approche nécessite cependant une grande quantité d'images d'un type d'objet spécifique pour entraîner le réseau, ce qui la rend difficilement applicable à notre problème.

1.2 Correspondances image-modèle

Cette section présente les méthodes couramment utilisées pour rechercher des correspondances entre les points de l'image et ceux du modèle en s'appuyant sur les descripteurs qui leur sont associés.

1.2.1 Recherche de correspondances

Dans [Gordon and Lowe, 2006] le modèle de la scène est un ensemble de points 3D chacun associé à un ensemble de descripteurs SIFT. La recherche de correspondances image-modèle est faite par une recherche de plus proche voisin. Les points d'intérêt de l'image sont associés à des descripteurs et on cherche pour chacun le point du modèle le plus proche. La distance entre un point de l'image et un point du modèle est le minimum des distances euclidiennes entre le descripteur du point de l'image et un des descripteurs du point du modèle.

Plutôt que de faire la mise en correspondance au plus proche voisin comme décrit ci-dessus, il est possible d'utiliser une approche de type apprentissage. Les points 3D du modèle ne sont plus associés à une collection de descripteurs mais à une représentation plus compacte, comme dans [Irschara et al., 2009] par exemple. L'utilisation de ce type d'approche permet une meilleure généralisation, mais pose d'autres problèmes. Dans le cadre de la classification d'images, [Boiman et al., 2008] observent que regrouper les mots dans des vocabulaires visuels, par exemple, entraîne une perte d'information qui réduit le taux de classification correcte. De plus, conserver l'ensemble des descripteurs et faire des recherches au plus proches voisins est raisonnable dans certaines applications. Ces observations au sujet de la classification d'images s'appliquent bien au problème d'initialisation de pose, dans la mesure où le temps de calcul n'est pas critique et des descripteurs isolés peuvent s'avérer déterminants pour le calcul de la pose.

1.2.2 Filtrage des correspondances

Les correspondances obtenues peuvent présenter un taux d'erreur élevé. Il est donc nécessaire de filtrer les correspondances erronées pour pouvoir estimer une pose de façon robuste. L'idée est d'ajouter une contrainte géométrique pour ne retenir qu'un groupe de correspondances qui soient cohérentes. Les contraintes géométriques couramment utilisées viennent de la géométrie épipolaire [Kushnir and Shimshoni, 2012] ou d'un modèle de transformation homographique. Dans le cadre de l'initialisation de pose on cherche un ensemble de correspondances qui soit cohérent avec une certaine caméra.

Une méthode largement répandue pour identifier un ensemble de correspondances cohérentes vis-à-vis d'un modèle donné est RANSAC, introduite dans [Fischler and Bolles, 1981]. RANSAC est un processus de vote où chaque point de données est considéré cohérent avec un modèle donné ou non. On tire aléatoirement le minimum de points de données requis pour estimer un modèle, une pose de caméra dans le contexte du calcul de pose. Chaque point vote ensuite selon qu'il est cohérent ou non avec ce modèle d'après une métrique à définir pour chaque application. L'ensemble des points cohérents avec un modèle donné est son ensemble de consensus. On itère ce procédé jusqu'à ce qu'un critère d'arrêt soit atteint et le modèle avec le plus grand ensemble de consensus est retenu. Définir la métrique pour décider quels sont les points cohérents avec le modèle et définir le critère d'arrêt sont les principales difficultés de mise en œuvre.

Bien que RANSAC permette d'estimer un modèle à partir de données partiellement erronées, le temps de calcul est directement lié au taux de correspondances correctes. Lorsque celui-ci est faible, converger vers un modèle correct peut nécessiter un grand nombre d'itérations et dans certains cas il n'y a pas convergence. La vitesse de convergence est liée directement au taux de correspondances correctes.

1.2.3 Mise en correspondance avec des modèles de grande taille

Le problème de mise en correspondance image-modèle est un problème combinatoire, et la difficulté consiste à identifier les appariements corrects en un temps compatible avec l'application. Ce problème peut devenir critique lorsqu'on cherche à se localiser par rapport à un modèle de grande taille. C'est le cas dans [Li et al., 2012] où l'objectif est la géo-localisation ou [Schindler et al., 2007] pour la localisation en milieu urbain. Dans ce contexte, le modèle peut contenir des millions de descripteurs. Trouver les correspondances correctes est une tâche complexe d'une part car le temps de recherche augmente et d'autre part parce que les descripteurs deviennent moins discriminants. Ce problème est différent de celui posé par les changements de point de vue, mais dans les deux cas on cherche a identifier des correspondances correctes parmi un très grand nombre de correspondances. En ce sens les méthodes proposées pour améliorer la mise en correspondance avec des modèles de très grande taille sont adaptables à la recherche de correspondances en présence de forts changements de point de vue.

1.3 Synthèse de vues pour la mise en correspondance

Cette section illustre les difficultés de mise en correspondance d'images distantes et présente un certain nombre d'approches qui utilisent des descripteurs issus d'images synthétisées pour améliorer la robustesse de la mise en correspondances.

1.3.1 Invariance limitée des descripteurs

SIFT est largement utilisé dans de nombreuses applications en raison de son invariance aux rotations et changements d'échelles. Plus précisément, [Morel and Yu, 2011] montrent que SIFT est effectivement invariant aux rotations et changement d'échelle si

les conditions de Shannon-Nyquist sont respectées. En pratique, l'invariance aux rotations est généralement bien vérifiée, mais des changements d'échelle notables peuvent mettre l'invariance du descripteur en défaut.

En revanche, aucun descripteur n'est robuste aux changements de point de vue, au delà d'un certain déplacement [Moreels and Perona, 2007]. Un même point de la scène n'est donc pas reconnaissable entre deux images en se basant uniquement sur un descripteur. Cette difficulté à mettre les points en correspondance est illustrée par la figure 1.2 dans laquelle on peut voir que le nombre de correspondances entre deux images diminue à mesure que la distance entre les deux points de vue augmente.

1.3.2 Descripteurs extraits de patchs rectifiés

Dans le contexte de la mise en correspondance de modèles en milieu urbain, [Wu et al., 2008] proposent d'utiliser des descripteurs extraits d'images fronto-parallèles, c'est à dire des vues dont la direction d'observation est perpendiculaire à la surface de la scène. Puisqu'on ne possède pas nécessairement une image fronto-parallèle pour chaque partie de la scène, de telles vues sont synthétisées à partir de celles connues. Chaque point du modèle est donc associé à un unique descripteur, qui est le descripteur SIFT extrait à ce point dans l'image rectifiée du plan sur lequel il se trouve. La mise en correspondance de ces descripteurs est possible puisque pour un point donné de la scène on calcule toujours le même descripteur, peu importe le point de vue sous lequel il a été observé.

Cette approche est adaptée au contexte urbain qui se compose de larges parties planes et pour lesquelles la notion de vue fronto-parallèle est donc bien définie. Elle permet de mettre en correspondance des points 3D qui ont été reconstruits en utilisant des points de vue différents. En revanche, elle est limitée dans le cadre de la mise en correspondance image-modèle, car il faut pouvoir rectifier l'image avant d'en extraire des descripteurs.

1.3.3 Transformations d'images

Plutôt que de chercher à associer un unique descripteur à chaque point de la scène, bon nombre d'approches consistent à leur associer une collection de descripteurs. L'idée est d'avoir un ensemble de descripteurs chacun étant associé à une direction de vue particulière. Dans le cadre de la mise en correspondance d'images, [Morel and Yu, 2009] proposent d'appliquer un ensemble de transformations affines aux images puis à chercher des correspondances entre l'ensemble des images ainsi obtenues. De façon générale, ce type d'approche est utilisé pour l'apprentissage automatique pour augmenter artificiellement la taille des ensembles d'apprentissage. Dans [Paulin et al., 2014], dans le cadre de la reconnaissance d'objets, les images d'apprentissage sont transformées via une collection de transformations : rotations, symétries, transformations affines... L'ensemble des images originales et transformées est utilisée pour apprendre le classifieur. Les résultats obtenus en utilisant cette base de données augmentée sont considérablement meilleurs que ceux obtenus avec la base d'origine.

Utiliser des données de synthèse permet d'améliorer de façon générale les possibilités de mise en correspondance au delà de l'invariance des descripteurs. En revanche c'est un procédé coûteux en temps de calcul, pour la génération des vues de synthèse d'une part

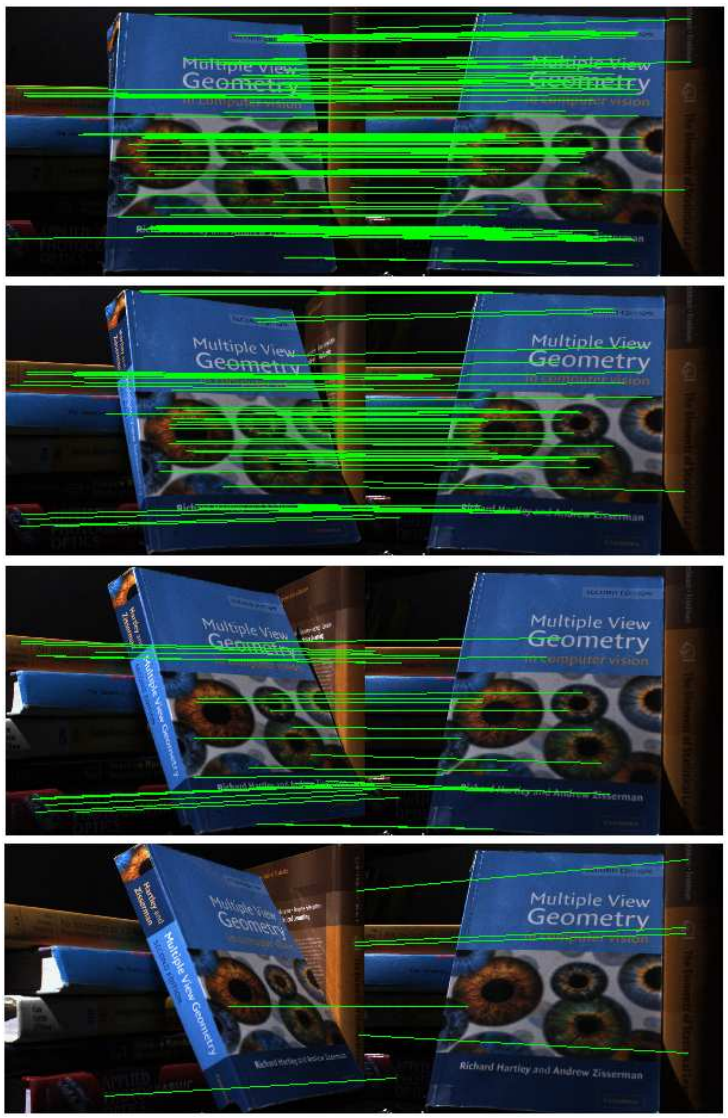


FIGURE 1.2 – Exemples de mise en correspondance de vues avec des points de vues différents. La vue de la colonne de droite est toujours la même, alors que celle de gauche est progressivement plus éloignée. Le nombre de correspondances chute considérablement avec le changement d'apparence de la couverture du livre, alors que les même objets restent visibles dans les images.

et pour la mise en correspondance d'autre part. En effet, le nombre de descripteurs à considérer lors de la mise en correspondance est potentiellement beaucoup plus important qu'en l'absence de descripteurs synthétiques.

1.3.4 Rendu à partir du modèle

Les méthodes précédentes utilisent des transformations 2D de vues existantes pour produire des vues synthétiques. Cependant il est possible de produire des vues synthétiques directement à partir du modèle de la scène. Par exemple, [Shan et al., 2014] proposent de générer des vues additionnelles à partir d'un modèle dense pour assister la localisation de drones. [Petit et al., 2012] proposent de synthétiser des contours à partir d'un modèle 3D pour permettre le suivi d'objet complexes à partir de mise en correspondance de contours.

Ce type de méthode permet de générer des vues synthétiques de l'ensemble de la scène, là où des méthodes basées sur des transformations d'images se limitent généralement à des patchs. En revanche elles nécessitent un modèle dense de la scène et sont donc peu utilisables avec des modèles reconstruits uniquement par SfM, qui n'ont pas une densité uniforme.

1.3.5 Difficultés de mise en œuvre

Les méthodes par synthèse de vues ont montré leur efficacité, mais se heurtent généralement à des problèmes de complexité : transformer des images est coûteux et ajouter des descripteurs ralentit la mise en correspondance. De plus les vues synthétisées peuvent ne pas correspondre à une situation effectivement observable. Dans ce cas les descripteurs extraits des vues synthétiques ne sont d'aucune aide pour la mise en correspondance et ont donc tendance à compliquer la tâche en ajoutant des fausses correspondances qui devront être éliminées par RANSAC.

Par ailleurs, beaucoup de méthodes de synthèse ont été proposées dans des contextes sans information sur la géométrie de la scène, comme c'est le cas pour la mise en correspondance image-image [Mishkin et al., 2015] ou la classification d'images [Paulin et al., 2014]. Dans ce cas, on ne peut pas garantir que les vues synthétisées ont effectivement un sens, les transformations effectuées ne correspondant pas à un changement de point de vue possible. Les vues synthétiques ne font alors qu'ajouter en complexité au problème.

1.4 Contributions

Cette section résume les différentes contributions apportées dans cette thèse au problème de mise en correspondance image-modèle et annonce le plan de la thèse. L'objectif global de ces contributions est de rendre le calcul de pose à partir d'un modèle SfM possible en présence de forts changements de point de vue en utilisant des descripteurs issus d'images synthétiques. Nous montrons que l'utilisation de patchs synthétiques permet le calcul de pose dans ces cas complexes et que la méthode proposée est efficace en temps de calcul. L'idée générale de notre méthode est illustrée dans la figure 1.3.

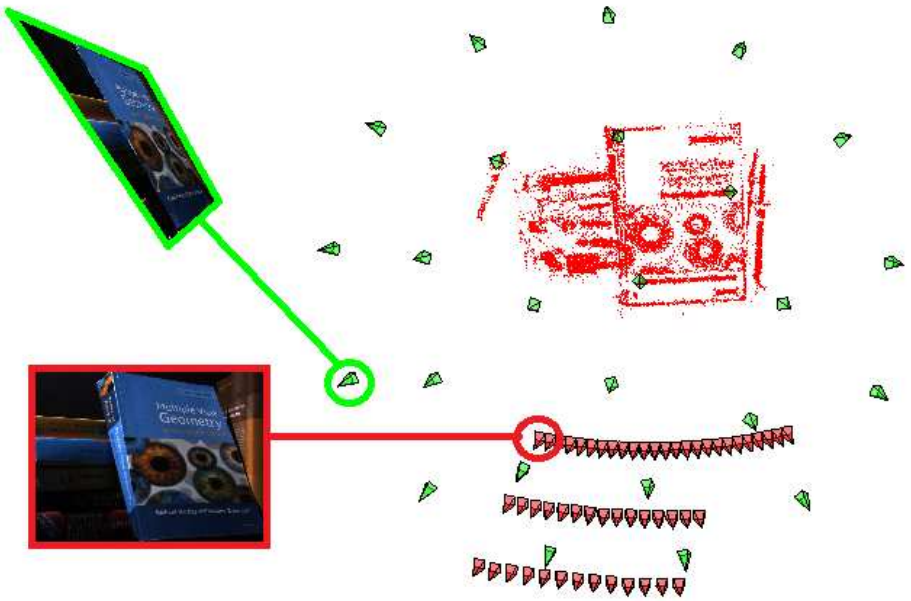


FIGURE 1.3 – La synthèse de patchs a pour objectif d'améliorer la couverture de la scène en termes d'observations. Les caméras en rouge sont celles qui ont servi à la construction du modèle. Elles sont toutes concentrées dans une faible région de l'espace et ne sont donc pas représentatives de l'ensemble des observations possibles. La synthèse de vues consiste à ajouter des points de vues virtuels, en vert, afin de densifier les observations de la scène.

1.4.1 Synthèse de patchs

Comme il n'existe pas de descripteur suffisamment robuste aux changements de point de vue nous proposons d'ajouter des descripteurs obtenus à partir de patchs simulés. L'approche proposée dans cette thèse consiste à simuler localement l'apparence de la scène sous certains points de vue puis à ajouter aux points du modèle des descripteurs extraits de ces simulations. La génération des patchs synthétiques est guidée par la géométrie de la scène. Plusieurs problèmes sont traités : comment faire des simulations réalistes ? Comment placer les points de vue à simuler ? Quelles parties de la scène synthétiser ? Comment rendre le procédé de simulation efficace ? La méthode proposée utilise une segmentation de la scène en plans, qui sont les éléments à simuler. Les caméras virtuelles pour lesquelles les simulations sont faites sont positionnées par rapport à ces plans. Chaque plan segmenté est donc associé à un ensemble de caméras virtuelles qui lui sont propres, et les patchs synthétiques sont les images de ce plan vu par cet ensemble de caméras. Ces contributions sont présentées en deux parties. Le chapitre 3 présente le principe général de la synthèse de vue. Les travaux présentés dans ce chapitre ont fait l'objet de plusieurs publications : [Rolin et al., 2014], [Rolin et al., 2015b] et [Rolin et al., 2015a]. Le chapitre 5 présente une approche plus générale de la synthèse de vue prenant en compte la géométrie de la scène et qui a fait l'objet d'une publication [Rolin et al., 2016].



FIGURE 1.4 – La projection de quelques contours de la scène permet d'illustrer la précision des poses calculées. L'image de gauche montre le cadre de référence. L'image du milieu montre les contours projetés avec des caméras estimées sans utiliser de vues de synthèse. L'image de droite montre les contours projetés avec des caméras estimées en utilisant de vues des synthèse. Les poses calculées présentent dans ce cas une variabilité bien plus faible, les contours projetés étant mieux localisés.

1.4.2 Mise en correspondance robuste

Le problème de mise en correspondance présent dans toutes les méthodes d'estimation de pose est amplifié par l'utilisation de descripteurs synthétiques. Pour cette raison, il est nécessaire de rendre la recherche de correspondances aussi efficace que possible. Une variante de RANSAC a été développée pour rendre cette recherche de correspondances efficace. Elle consiste a estimer un score à chaque correspondance puis à tirer préférentiellement des correspondances avec un haut score dans le même esprit que PROSAC [Chum and Matas, 2005]. Cette méthode de mise en correspondance permet de converger significativement plus rapidement vers une pose correcte que RANSAC standard ou d'autres variantes existantes. Ces travaux sont présentés dans le chapitre 4.

1.4.3 Résultats expérimentaux

Les expériences menées pour valider l'approche proposée se concentrent sur deux aspects : montrer que l'utilisation de patchs synthétiques permet de calculer des poses dans des situations difficiles et montrer que les temps de calculs associés sont raisonnables. Une expérience typique consiste à construire un modèle à partir d'une collection d'images par *structure from motion*, puis à essayer de localiser une image test par rapport à ce modèle. Cette image test est généralement prise avec un angle de vue éloigné de ceux des images utilisées pour la construction du modèle, afin de se placer dans une situation où la mise en correspondance n'est pas triviale. Ces expériences montrent que l'ajout de descripteurs issus de vues synthétiques permet d'estimer des poses dans des situations où c'était auparavant impossible. De façon générale l'utilisation de vues de synthèse améliore la précision des poses calculées, comme illustré dans la figure 1.4. En termes de temps de calcul, la

phase de synthèse en elle-même requiert environ autant de temps que la reconstruction du modèle, mais cette étape peut aussi être faite en avance et non pas au moment de l'estimation de la pose.

1.4.4 Perspectives

La synthèse de vue permet d'améliorer significativement le calcul de pose et s'applique à un grand nombre de situations. Plusieurs aspects de l'approche proposée peuvent être développés. Le chapitre 4 explique comment la mise en correspondance peut être faite de façon efficace avec un modèle de grande taille. Mais il est également possible de réduire directement la taille du modèle. Chaque point du modèle est représenté par une collection de descripteurs SIFT, mais il est possible de définir des représentations plus compactes. Une première idée est de choisir des éléments représentatifs pour chaque point, plutôt que de conserver l'ensemble des descripteurs, ce qui est proposé par exemple dans [Irschara et al., 2009]. Une autre possibilité est de réduire la dimension des descripteurs utilisés. D'autres points peuvent être pris en compte pour aller au delà de la méthode proposée : synthétiser des vues à une distance variable de la scène ou ajouter itérativement des vues de synthèse pendant le calcul de pose, par exemple. Ces perspectives sont développées dans le chapitre 7

Chapitre 2

Synthèse de vues

Ce chapitre présente les méthodes de synthèse de vues existantes, dans le contexte de la mise en correspondance de points. Les méthodes de mise en correspondance reposent sur des descripteurs de points, de zones, de contours... En utilisant ces descripteurs on espère pouvoir trouver un même point de la scène vu dans différentes images. Le descripteur doit donc rester relativement invariant d'une image à une autre. Certains changements de conditions laissent effectivement la plupart des descripteurs inchangés, mais pour d'autres il n'existe pas de descripteurs suffisamment invariant. Dans cette thèse on s'intéresse en particulier à la mise en correspondance de points dans des images prises avec des points de vue très différents.

Dans [Mikolajczyk and Schmid, 2005] les performances d'un ensemble de descripteurs photométriques classiques sont comparés en présence de différents type de transformations : changement de point de vue, changement d'illumination, changement d'échelle, rotation, flou et compression. Parmi ces transformations, les changements de point de vue sont les plus complexes à prendre en compte. Le taux de correspondances correctes diminue rapidement pour tous les descripteurs testés dès que le changement de point de vue dépasse 30 degrés [Moreels and Perona, 2007].

Comme des descripteurs issus d'images prises avec des points de vue différents ne peuvent pas être facilement mis en correspondance, on utilise des descripteurs extraits dans des images synthétiques. On ne définit pas un nouveau type de descripteur, mais on génère des images qui peuvent être mises en correspondance en utilisant les descripteurs existants. C'est le principe général utilisé par les méthodes de synthèse de vues. On peut distinguer principalement deux types de méthodes : celles qui n'utilisent pas de modèle de scène et celles qui en utilisent un.

2.1 Synthèse de vues sans modèle de scène

Dans un certain nombre d'applications de mise en correspondance on ne possède pas d'information sur la géométrie de la scène. C'est typiquement le cas pour la recherche de correspondances entre les points de plusieurs images, ou dans le cadre de la classification d'images. Malgré l'absence de modèle de scène, de nombreuses méthodes de synthèse de vues ont été développées dans ce contexte et ont montré leur utilité.

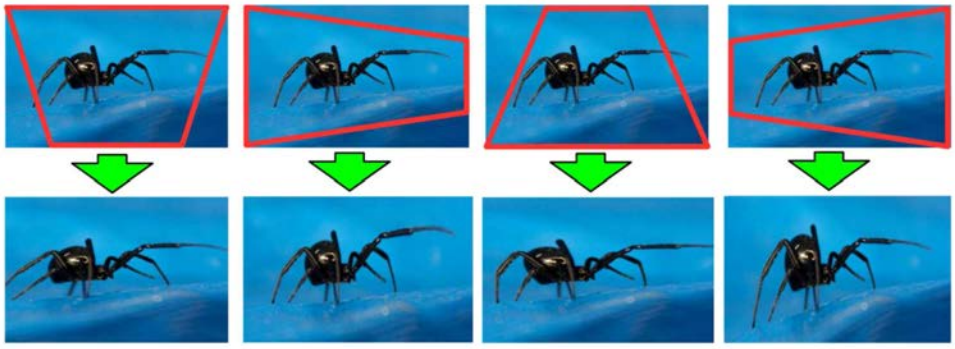


FIGURE 2.1 – Quelques exemples de vues synthétisées (en bas) à partir d'une image réelle (en haut). Les homographies appliquées ne sont pas définies par rapport à un modèle de scène mais les images obtenues sont de bonnes approximations de ce qu'on observerait avec un changement de point de vue réel (image extraite de [Paulin et al., 2014]).

2.1.1 Classification

La synthèse de vues est largement utilisée dans les applications de classification d'images. Un des problèmes récurrent dans l'apprentissage de classes est le manque de données. En effet, pour apprendre une classe d'images un grand nombre d'exemples est nécessaire et plus on cherche à décrire une classe précise plus la base d'apprentissage doit être grande.

Dans [Paulin et al., 2014] la synthèse de vues est utilisée dans le cadre de la classification d'objets. Pour augmenter la taille de l'ensemble d'apprentissage, un certains nombre de transformations sont appliquées aux images : symétries, découpages, homographies, homothéties, compression, rotations et compositions de ces transformations. Ces transformations correspondent à des changements de conditions auxquels les descripteurs sont peu robustes. La base d'images augmentée correspond à des observations de l'objet dans des conditions beaucoup plus variées que la base d'image initiales, et une nouvelle image a donc plus de chance d'être correctement affectée à la classe qui lui correspond. Les transformations appliquées ne s'appuient pas sur un modèle de l'objet, mais sont une suffisamment bonne approximation de ce qui peut être observé, comme illustré dans la Figure 2.1. Le temps de calcul est un problème récurrent des méthodes de synthèse, les auteurs proposent donc de sélectionner itérativement les transformations à appliquer aux images. De nouvelles vues de synthèse sont ajoutées tant qu'elles améliorent significativement le score de classification.

[Lepetit et al., 2005] utilisent des patchs synthétiques pour entraîner un arbre aléatoire (*randomized tree*) [Amit and Geman, 1997] dans le but de reconnaître des points d'intérêt. Les patchs sont générés autour de points d'intérêts par transformation affine, ajout de bruit et floutage, comme illustré dans la Figure 2.2.

Récemment les réseaux de neurones profonds ont permis d'améliorer significativement les scores de classification pour des tâches telles que la reconnaissance d'objet. Les performance des réseaux de neurones profond viennent avec un coût significatif en temps de calcul et le besoin de fournir une quantité considérable de données d'apprentissage,

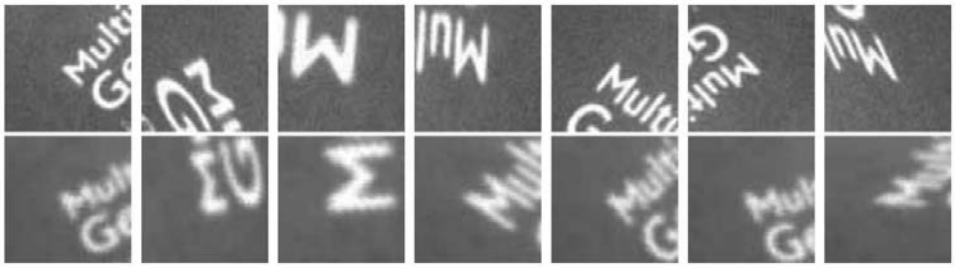


FIGURE 2.2 – Patchs synthétiques utilisés pour entraîner le classifieur dans [Lepetit et al., 2005]. Les patchs sont tous centrés autour d'un même point. Ils sont générés par un ensemble de transformations affines (première rangée) puis leur orientation est normalisée en s'appuyant sur un histogramme de gradient et le patch est flouté (seconde rangée) (image extraite de [Lepetit et al., 2005]).

typiquement de l'ordre de centaines de milliers d'images. Obtenir une telle quantité de données est souvent difficile voire impossible et l'augmentation de données par synthèse s'est donc imposée comme une étape classique de l'entraînement d'un réseau de neurones profond [Krizhevsky et al., 2012].

Il est également possible de générer des vues de synthèse directement à partir d'un réseau de neurones. [Zhou et al., 2016] proposent une architecture de réseau de neurones convolutifs pour la prédiction de l'apparence d'objets (chaises et voitures) et de scènes urbaines à partir de différents points de vue. Les vues produites par cette méthode sont visuellement proches de vues réelles, et, pour les objets, les changements de points de vue considérés sont assez importants. Cependant, une très grande quantité de données sont nécessaires pour entraîner ce type de réseau, et ce pour chaque objet ou type de scène qu'on souhaite générer.

2.1.2 Mise en correspondance d'images

La synthèse de vues a également été utilisée dans le cadre de la recherche de correspondances entre images. Dans ce cadre on suppose qu'on possède deux vues d'une même scène et on cherche les correspondances éventuelles entre ces deux vues.

[Morel and Yu, 2009] proposent une extension de SIFT robuste aux transformations affines. Les auteurs observent que l'effet d'un changement de point de vue sur un plan est une homographie et que cette transformation peut être approchée localement au premier ordre par une transformation affine. Comme les descripteurs SIFT sont calculés sur des patchs locaux, il est justifié d'utiliser cette approximation pour synthétiser des vues et en extraire des descripteurs. Ils développent donc la méthode de mise en correspondance suivante : pour chaque image à mettre en correspondance des vues de synthèse sont générées par transformations affines. Des descripteurs SIFT sont extraits dans l'ensemble des images produites et toutes les paires d'images sont mises en correspondance. Une sélection de transformation optimale est proposée qui couvre les changements de point de vue possibles jusqu'à un certain angle limite avec un minium de vues synthétiques,

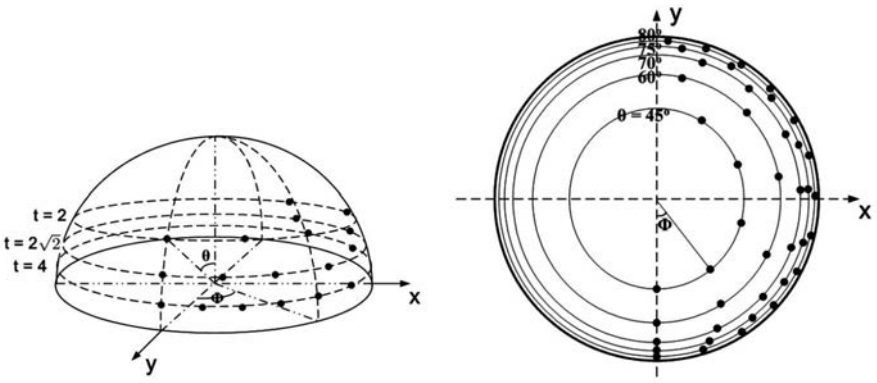


FIGURE 2.3 – Répartition des points de vues simulés par l'algorithme ASIFT (points noirs). Le point de vue de l'image d'origine est supposé fronto-parallèle ($\theta = 0$) (image extraite de [Morel and Yu, 2009]).

voir Figure 5.2. Pour éviter de générer systématiquement toutes les vues de synthèse les auteurs proposent de faire une première mise en correspondance à faible résolution pour identifier les directions de vue à simuler puis de refaire la mise en correspondance avec les images complètes en synthétisant uniquement les directions identifiées.

Dans [Mishkin et al., 2015] différents ensembles de transformations sont appliqués successivement aux images à mettre en correspondance. Les auteurs observent que le nombre de vue de synthèse à générer est dépendant de la difficulté de la mise en correspondance, et que pour la plupart des cas quelques vues synthétiques suffisent. Leur approche consiste à générer progressivement des vues de synthèse de plus en plus nombreuses jusqu'à ce que suffisamment de points soient mis en correspondance.

2.2 Synthèse de vues avec un modèle

Dans de nombreuses situations on possède un modèle de la scène. Il est alors possible de raffiner la synthèse de vues en exploitant ce modèle. Cette section présente différentes méthodes qui utilisent la synthèse de vues dans ce contexte.

2.2.1 Rendu 3D

Si on possède un modèle dense de la scène il est possible de produire des vues de synthèse directement par rendu 3D. Ce type de méthode permet de générer des vues synthétiques de l'ensemble de la scène, là où des méthodes basées sur des transformations d'images se limitent généralement à des patchs. En revanche elles nécessitent un modèle dense de la scène et sont donc peu utilisables avec des modèles reconstruits uniquement par SfM, qui n'ont pas une densité uniforme.

[Shan et al., 2014] proposent de générer des vues synthétiques à partir d'un modèle dense pour assister le positionnement de drones aériens par rapport à des modèles de

bâtiments au sol. Le modèle de scène a été reconstruit à partir du sol, il y a donc un fort changement de point de vue entre l'image à localiser et les images utilisées pour produire le modèle. Une pose approximative est supposée connue et une vue de synthèse est générée en projetant le modèle dense avec cette pose. La méthode est particulièrement adaptée à cette situation car la caméra à simuler est loin du modèle. Le manque de précision éventuel du modèle est donc relativement moins problématique, les détails de la scène n'étant de toute façon pas visibles à cette distance. La méthode permet d'estimer des poses pour des images présentant un fort changement de point de vue et est efficace, mais elle requiert un modèle dense de la scène et surtout une approximation de la position de la caméra.

[Torii et al., 2015] utilisent la synthèse de vues pour assister la localisation en milieu urbain. Le modèle de la scène consiste en une carte de profondeur associée à des vues panoramiques. Les vues de synthèse sont produites par *ray tracing*. A chaque pixel de la vue synthétique on associe un point 3D dans la scène à l'aide de la carte de profondeur. Ce point 3D est projeté dans la vue réelle la plus proche pour en déduire la couleur du pixel. Cette technique de synthèse produit des vues de la scène réaliste sous l'hypothèse qu'on possède une carte de profondeur raisonnable. Le placement des caméras virtuelles proposé, sur les nœuds d'une grille au niveau du sol, permet de générer des vues synthétiques couvrant les points de vue utiles mais est spécifique au contexte urbain.

[Petit et al., 2012] utilisent également le rendu 3D, dans le contexte du suivi d'objets complexes. Au lieu de rendre une image texturée, ils se limitent aux contours de l'objet à suivre, ce qui réduit significativement le coût en temps de calcul. Cette approche ne nécessite pas un modèle d'objet texturé, mais une carte de profondeur précise est tout de même requise pour synthétiser des contours raisonnables.

[Debevec et al., 1996] proposent une approche hybride de la synthèse d'image, dans laquelle un modèle structuré est construit à partir d'une séquence d'images. Cette approche est proposée pour des scènes urbaines, et l'hypothèse est faite que la scène peut être modélisée par un ensemble de primitives géométriques simples telles que des prismes droits, par exemple. Cette approche permet d'obtenir des rendus de vues synthétiques très réalistes, sans nécessiter de modèle préalablement construit. Cependant les conditions nécessaires à son fonctionnement restreignent son utilisation à des scènes urbaines.

2.2.2 Rectification

Utiliser des images rectifiées pour faire la mise en correspondance est une autre façon d'utiliser la synthèse de vues pour dépasser les problèmes liés aux changements de point de vue. L'idée est d'utiliser un point de vue de référence pour chaque point de la scène et de faire la mise en correspondance en utilisant les descripteurs obtenus à partir de ces points de vue.

Dans [Wu et al., 2008] par exemple les descripteurs sont extraits de vues fronto-parallèles. Les points d'intérêt sont caractérisés à la fois par leur position dans la scène, mais aussi par l'orientation locale de la surface (la direction du vecteur normal) sur laquelle ils se trouvent, ce qui permet de calculer un descripteur identique à partir de tout point de vue. Le principe est similaire à celui utilisé pour la détection de point SIFT pour lesquels les points d'intérêt sont caractérisés à la fois par leur position dans l'image mais également par l'échelle à laquelle ils sont extraits et par l'orientation locale du gradient,

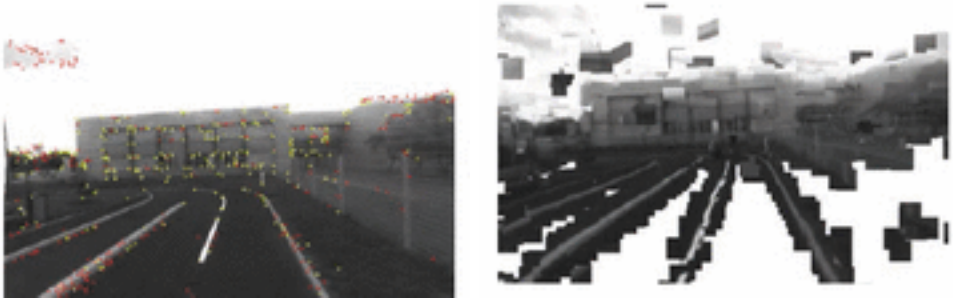


FIGURE 2.4 – Image réelle (à gauche) et image synthétique (à droite) obtenue en projetant un ensemble de patch par une pose proche de celle cherchée (image extraite de [Charmette et al., 2016]).

ce qui permet de calculer un descripteur invariant par similitude.

Comme ce type de méthode requiert la connaissance de l'orientation locale de la surface, le descripteur ne peut être calculé que si on connaît le lien entre l'image et le modèle. Par conséquent l'approche n'est utilisable que dans le cadre de la mise en correspondance de deux modèles, et pas pour la mise en correspondance image-modèle.

2.2.3 Simulation de patchs

Certaines méthodes de synthèse de vues produisent un ensemble de petites images, ou patchs, plutôt que des vues de la scène complète. Il est en effet généralement plus simple de simuler localement l'apparence de la scène. De plus, si l'objectif est d'extraire des descripteurs qui sont calculés sur un voisinage d'un point alors ces simulations locales sont suffisantes.

[Charmette et al., 2016] proposent une méthode permettant de suivre la position d'un robot dans un environnement préalablement reconstruit. La localisation s'appuie sur la continuité de la trajectoire du robot, de laquelle on peut déduire une approximation raisonnable de la position du robot. Comme on a une approximation de la pose, on peut générer une image synthétique de ce qu'on s'attend à observer qui est proche de ce qui est effectivement observé par le robot, ce qui est illustrée dans la Figure 2.4. La mise en correspondance de ces deux images est généralement aisée puisque l'image synthétique a été construite pour ressembler à l'image observée. Dans cette application le modèle se compose de patchs orientés, et générer une image synthétique consiste à projeter ces patchs en utilisant la pose approximative de la caméra.

[Savarese and Fei-Fei, 2008] utilisent la synthèse de vues pour simultanément identifier des objets et estimer leur pose. Le modèle de l'objet consiste en une collection d'observation réparties sur une sphère autour de l'objet, chaque observation étant associée à une direction de vue. Les vues de synthèse sont générées par interpolation à partir des vues connues. Le taux d'images correctement classifiées augmente significativement, en particulier lorsque le nombre d'images utilisées pour l'apprentissage est faible.

[Irschara et al., 2009] puis [Wendel et al., 2011] utilisent la synthèse de vues dans

le contexte de la localisation en milieu urbain. Des caméras virtuelles sont placées sur une grille régulière et utilisées pour produire des patchs synthétiques par transformations affines des vues existantes. Dans ces applications la synthèse de vue n'est pas utilisée pour permettre le calcul de pose à partir de points de vue extrêmes mais les résultats montrent que les poses calculées sont plus précises.

2.3 Problèmes ouverts

Cette section synthétise les problèmes communs présents dans les approches par synthèse de vues et expose les situations pour lesquelles ces méthodes sont peu adaptées.

2.3.1 Synthèse de vues pour un nuage de points non dense

Les méthodes de synthèse présentées ne sont pas applicables directement à un modèle de type nuage de points. Ce type de modèle est pourtant largement répandu puisque c'est typiquement ce qu'on obtient comme résultat d'une reconstruction par *structure from motion*. Le premier objectif de cette thèse est de proposer une méthode de synthèse de vues adaptée à ce type de modèle et de montrer que la synthèse de vues permet effectivement d'aider au calcul de pose lorsqu'on utilise ce type de modèle.

2.3.2 Synthèse de vues pour l'initialisation de pose

Le choix des vues à synthétiser est globalement peu abordé dans les méthodes présentées. Certaines utilisent une approximation de la pose cherchée pour assister la génération de vue synthétiques. Dans ce cas, on produit une unique vue synthétique en utilisant cette pose approximative, en espérant ainsi produire une image proche de celle à partir de laquelle on veut calculer une pose. D'autres méthodes prédéfinissent une série de transformations à appliquer aux images, généralement lorsqu'il n'y a pas de modèle de scène. Enfin, certaines méthodes proposent de positionner des caméras par rapport au modèle de la scène mais dans des contextes particuliers, comme la reconnaissance d'objet où le modèle est relativement petit et où on peut supposer les caméras réparties autour. Le problème consistant à déterminer les positions de caméras synthétiques à utiliser lorsqu'un modèle de scène est disponible reste largement ouvert dans un cadre général.

2.3.3 Temps de calcul

L'utilisation de vues de synthèse implique un coût supplémentaire significatif en temps de calcul, faire la synthèse de vues naïvement n'est pas réaliste dans une application pratique. La plupart des méthodes présentées apportent des solutions pour diminuer les temps de calcul : simuler uniquement une vue proche de la pose attendue [Charmette et al., 2016], simuler de façon incrémentale [Mishkin et al., 2015], faire une première exécution à faible résolution [Morel and Yu, 2009]. Ces accélérations sont spécifiques à certaines méthodes et ne peuvent pas être directement appliquées à celle développée dans

cette thèse. Développer des accélérations adaptées à la méthode proposée dans la thèse est donc également un point important.

Chapitre 3

Synthèse de patchs guidée par la géométrie

Ce chapitre discute la pertinence de l'utilisation de synthèse de vues dans le but d'améliorer la mise en correspondance image-modèle. Comme expliqué dans l'introduction, mettre en correspondance une image avec le modèle de la scène est complexe car on utilise uniquement des descripteurs photométriques. La mise en correspondance est donc limitée par l'invariance des descripteurs : on ne peut espérer trouver des correspondances qu'avec des images suffisamment proches de celles utilisées pour construire le modèle. Mais si le modèle avait été observé sous tous les angles de vue possibles, ou du moins si la densité des observations était suffisamment élevée, alors le problème ne se poserait pas. De cette observation est apparue l'idée de produire des vues synthétiques de la scène : si on ne possède pas suffisamment d'observations de la scène on peut les produire artificiellement. La méthode et les résultats présentés dans ce chapitre ont fait l'objet de deux publications : [Rolin et al., 2015b] et [Rolin et al., 2015a].

3.1 Synthèse de vue dans un monde localement plan

Cette section présente le principe général de synthèse de vue utilisé dans cette thèse. Nous supposons disposer d'un modèle d'une scène, constitué d'un nuage de points, et que chacun de ces points est associé à un ensemble de descripteurs SIFT provenant des vues réelles dans lesquelles il a été repéré. Nous supposons également que la scène est localement plane autour des points 3D, et que l'on a associé à chaque point le vecteur normal du plan sur lequel il se trouve. Étant donnée une vue réelle d'une zone plane autour d'un point 3D, comment synthétiser une vue de cette zone à partir d'une nouvelle position de caméra, afin d'en extraire un nouveau descripteur SIFT ?

Si on modélise les caméras comme des sténopés, deux vues d'un même plan sont liées par une homographie. Dans le modèle de caméras affines (lorsque la profondeur de la scène est faible devant la focale), les deux vues sont liées par une transformation affine. [Morel and Yu, 2009,Ozuysal et al., 2010] montrent que cette simplification est souvent suffisante. En effet, comme une transformation affine est une approximation au premier ordre d'une homographie, des transformations affines ou homographiques d'une petite zone de l'image

sont visuellement proches. Néanmoins les descripteurs SIFT sont souvent extraits sur des disques de plusieurs dizaines de pixels de rayon, pour lesquels l'approximation affine n'est plus valide dès que l'angle entre les vues est assez grand (plus grand que 30°).

3.1.1 Transformations induites par un changement de point de vue

Une façon courante de générer des vues synthétiques d'une scène consiste à transformer des images connues. On sait en effet décrire la transformation induite par un changement de point de vue au niveau d'un plan. Autrement dit, si I_1 et I_2 sont les images d'un même plan P vu par deux caméras C_1 et C_2, on peut calculer la transformation H telle que $I_2 = H(I_1)$ connaissant les matrices de projection des caméras et l'équation du plan.

Cette transformation peut être calculée de la façon suivante, comme détaillé dans [Hartley and Zisserman, 2004]. Soient deux caméras représentées par leurs matrices de projection $P_1 = K_1[R_1|T_1]$ et $P_2 = K_2[R_2|T_2]$ (où K_i est la matrice des paramètres intrinsèques et R_i, T_i déterminent la pose dans un repère commun, $i \in \{1, 2\}$). Considérons un plan de l'espace d'équation $n^T X + d = 0$ (où n est un vecteur normal au plan, d un paramètre réel, et X des coordonnées d'un point de l'espace). La transformation induite par le plan entre les deux caméras est alors l'homographie H donnée par l'équation homogène :

$$H = K_2(R - Tn^T/d)K_1^{-1} \tag{3.1}$$

où $R = R_2R_1^T$ et $T = -R_2(C_2 - C_1)$ (où le centre optique C_i vérifie $C_i = -R_i^T T_i$, $i \in \{1, 2\}$.)

Remarquons que dans le cas où les deux caméras partagent le même axe optique et que celui-ci porte le vecteur n, cette homographie se réduit à une similitude.

Si P_1 est la matrice de projection d'une caméra réelle, P_2 celle d'une caméra virtuelle, et I_1 et I_2 les images du plan dans ces deux caméras, alors $HI_1 = I_2$, soit :

$$K_2R_2(R_1^T + (C_2 - C_1)n^T/d))K_1^{-1}I_1 = I_2. \tag{3.2}$$

Rappelons que la matrice R_2 s'écrit $R_2 = R_Z(\kappa)R_Y(\phi)R_X(\omega)$ où (X, Y, Z) est un repère orthonormé tel que Z est l'axe optique de la caméra et (κ, ϕ, ω) sont les angles d'Euler associés. Les descripteurs SIFT étant supposés invariants par similitude (plane), on voit que toute rotation autour de l'axe optique ou tout changement de focale de la caméra 2 fournira les mêmes descripteurs. Donc la pose de la caméra virtuelle n'a besoin d'être fixée qu'à une rotation selon l'axe optique près, et la focale est arbitraire. Comme il l'a été souligné dans [Morel and Yu, 2011], ce raisonnement sur des images idéales continues reste valable pour des images discrètes sous réserve de respect de la condition de Shannon-Nyquist. Néanmoins la position de la caméra est ici importante (T_2 intervient dans (3.2)).

La donnée du plan, d'une pose de caméra réelle, et de la pose de la caméra virtuelle (à une rotation selon l'axe optique près) permet de synthétiser avec l'équation (3.2) une vue de laquelle nous allons extraire un descripteur SIFT.

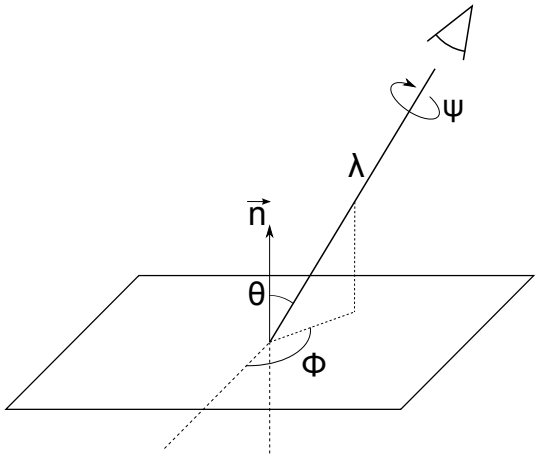


FIGURE 3.1 – Position d'une caméra affine par rapport à la normale d'un morceau de plan, avec les notations de l'équation (3.3) où $t = 1/\cos(\theta)$.

3.1.2 Approximation affine

Dans le cas de deux caméras affines, notons $(\lambda_i, \psi_i, t_i, \phi_i)$ les éléments caractéristique de la caméra $i \in \{1, 2\}$ dans un repère associé à un plan repéré par son vecteur normal n (figure 3.1). Les angles ϕ_i et θ_i sont respectivement la longitude et la latitude de l'axe optique de la caméra. Le paramètre $t_i = 1/\cos(\theta_i)$ est le tilt de la caméra. Le paramètre ψ_i correspond à la rotation de la caméra autour de son axe optique et λ_i au zoom. La transformation induite par le plan entre une vue fronto-parallèle de ce plan et la vue i est donnée par la transformation affine suivante [Morel and Yu, 2009, Ozuysal et al., 2010] :

$$A_i = \lambda_i \begin{pmatrix} \cos(\psi_i) & -\sin(\psi_i) \\ \sin(\psi_i) & \cos(\psi_i) \end{pmatrix} \begin{pmatrix} t_i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\phi_i) & -\sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) \end{pmatrix}. \tag{3.3}$$

Par composition, la transformation affine induite par le plan entre les deux caméras est :

$$A = A_2 A_1^{-1}. \tag{3.4}$$

Avec les mêmes notations que dans le cas des homographies, $AI_1 = I_2$ soit $A_1^{-1} I_1 = A_2^{-1} I_2$. L'invariance aux similitudes des descripteurs SIFT nous permet d'écrire que toutes les valeurs de $\psi_1, \psi_2, \lambda_1, \lambda_2$ fournissent les mêmes descripteurs SIFT, que l'on choisit donc arbitrairement à $\psi_1 = \psi_2 = 0, \lambda_1 = \lambda_2 = 1$.

Ainsi la donnée des positions relatives (t_i, ϕ_i) des caméras réelles et virtuelles par rapport à la normale à une partie plane de la scène permet de synthétiser une vue avec l'équation (3.4) de laquelle on extraira un descripteur SIFT.

3.2 Mise en œuvre

Cette section décrit un algorithme d'ajout de descripteurs par synthèse de vues. Un modèle non structuré est construit et les points associés à un ensemble de descripteurs SIFT et au vecteur normal au plan sous-jacent (section 3.2.1), puis des descripteurs associés à des vues de synthèse sont ajoutés (section 3.2.2). La pose d'une nouvelle vue peut ensuite être estimée à partir de ce modèle enrichi (section 3.2.3).

3.2.1 Construction du modèle

Le logiciel VisualSFM [Wu, 2011] est utilisé pour générer un ensemble de points $\mathcal{P}$ de la scène tridimensionnelle, chaque point étant associé à la classe des descripteurs SIFT extraits des images dans lesquelles il est vu. Le logiciel permet également de générer une reconstruction dense de la scène basée sur [Furukawa and Ponce, 2010]. Nous utilisons ce modèle dense pour générer en chaque point de $\mathcal{P}$ une estimation de la normale en considérant le plus petit vecteur propre d'une analyse en composantes principales des coordonnées de ses k-plus proches voisins [Hoppe et al., 1992]. La normale est orientée vers les caméras dans lesquelles le point considéré est repéré. Nous n'utilisons plus la reconstruction dense dans la suite.

3.2.2 Ajout de descripteurs à partir de vues synthétiques

Position des caméras virtuelles

La position des caméras virtuelles est choisie de manière à compléter les points de vue des caméras ayant permis de construire le modèle. Comme on l'a vu dans la section 3.1, le cas affine ne nécessite que de positionner les caméras sur une demie-sphère orientée par la normale considérée, alors que le cas homographique nécessiterait de préciser leur distance par rapport à la scène.

Dans cette étude préliminaire nous placerons les caméras virtuelles dans les mêmes positions dans les deux cas : il s'agit de vingt-cinq positions régulièrement réparties sur une demi-sphère s'appuyant sur un plan moyen de la scène, de rayon égal à la distance de la plus proche caméra à la scène, comme dans la figure 3.2 ; les caméras sont dirigées vers le barycentre de la scène. Nous simulons donc un grand nombre de directions d'observation de la scène, mais pas de variations de la distance de la caméra à la scène. Néanmoins, les expérience présentées dans la section 3.3.2 montrent que ces simulations sont suffisantes pour calculer des poses relativement éloignées des vues de reconstruction et des vues virtuelles.

Cet échantillonnage est arbitraire dans ce chapitre, mais sera défini en fonction de la géométrie de la scène et des points de vue utilisés pour construire le modèle dans le chapitre 5.

Choix de la vue utilisée pour la simulation et extraction d'un descripteur SIFT

Étant donné un point du modèle 3D (associé à des descripteurs venant de plusieurs vues réelles) et un point de vue à simuler, il faut également choisir à partir de quelle

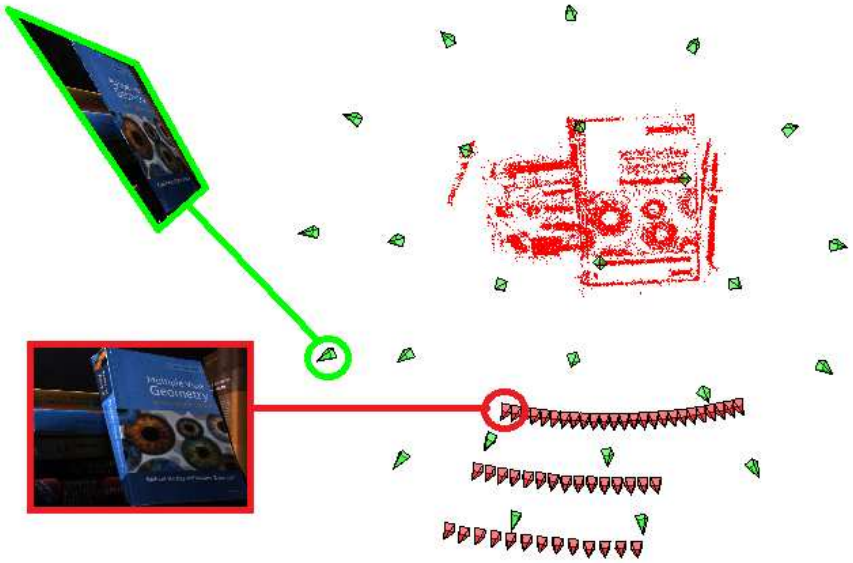


FIGURE 3.2 – Le modèle 3D de la scène (points rouge), les caméras ayant servi à le construire (en rouge pâle), une caméra éloignée dont on chercherait la pose (en cyan, entouré), et les caméras virtuelles (en vert), réparties ici sur une demi-sphère. Les caméras virtuelles permettent de générer de nouveaux descripteurs pour chaque point du modèle.

vue réelle réaliser la simulation. Parmi les vues dans lequel le point 3D est visible, la vue à partir de laquelle la simulation est réalisée est la plus proche angulairement du point de vue qu'on veut simuler, ce qui est un choix raisonnable pour limiter l'influence des spécularités.

L'image synthétique est une imagette de taille 100×100 pixels centrée sur un point du modèle, qui correspond à l'apparence de ce point observé à partir d'une caméra virtuelle. L'algorithme SIFT permet alors d'extraire des couples de points d'intérêt et descripteurs dans cette imagette. On ajoute alors à la liste des descripteurs de ce point 3D le descripteur extrait de l'imagette dont le point d'intérêt est le plus proche de la position théorique de la projection du point 3D, si cette distance est inférieure à 10 pixels. Ce seuil correspond à une distance de reprojection typique des points du modèle obtenu par SfM.

3.2.3 Estimation de la pose

Correspondances image/modèle

On commence par extraire les descripteurs SIFT de la nouvelle vue. La méthode de mise en correspondance utilisée est celle proposée dans [Gordon and Lowe, 2006]. Pour apparier un point d'intérêt p_1 de la nouvelle vue à un point 3D, on considère les distances d_1 et d_2 du descripteur SIFT de p_1 aux deux plus proches classes de descripteurs. Si d_1/d_2 est inférieur à un seuil λ on retient la correspondance. La recherche des plus proches voisins est accélérée comme dans [Gordon and Lowe, 2006] par une recherche approchée [Mount

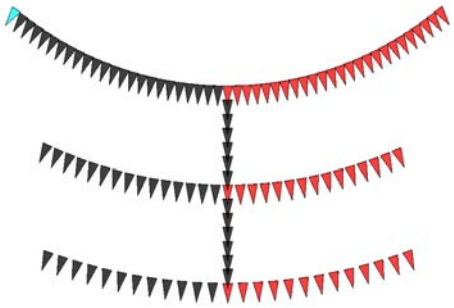


FIGURE 3.3 – Les positions des 119 caméras de la base *Robot Data Set*. En rouge les caméras servant à la reconstruction par SfM, en cyan la caméra de test.

and Arya, 2010].

Perspective-n-Points

Le calcul de pose se fait par une estimation robuste de type RANSAC [Fischler and Bolles, 1981] basée sur l'algorithme PnP proposé dans [Hesch and Roumeliotis, 2011]. Bien entendu, plus la proportion de correspondances correctes dans l'étape précédente est grande, plus le nombre d'itérations requises dans RANSAC peut être diminué.

3.3 Étude expérimentale

Les expériences suivantes montrent qu'en présence de fortes variations de direction de vue ou de profondeur la synthèse de vue améliore considérablement l'estimation de la pose. La pose peut être calculée dans des situations où une approche basée uniquement sur SIFT, telle que celle de [Gordon and Lowe, 2006], échoue. Plus généralement, pour un nombre fixé d'itérations de RANSAC, la pose est calculée avec plus de précision en utilisant les vues synthétiques. À la fin de cette section nous discutons les problèmes de temps de calcul et les améliorations envisageables.

3.3.1 Protocole expérimental

La méthode proposée est évaluée sur quatre séquences d'images : la séquence numéro 2 de la base *Robot Data Set* avec la première illumination proposée (la reconstruction de la scène est présentée dans la figure 3.2 et les positions des caméras utilisées dans la figure 3.3) et trois séquences personnelles, illustrées dans la figure 3.4. Ces séquences sont composées d'images de taille 1600×1200 pixels et les scènes associées sont globalement planes par morceaux et centrées objet.

Toutes les expériences utilisent le même protocole. Un modèle 3D de la scène est construit avec VisualSfM (section 3.2.1). La pose d'une vue test est calculée (section 3.2.3) dans trois scénarios : **S** où le modèle est la reconstruction obtenue par SfM sans synthèse de



FIGURE 3.4 – Images représentatives des quatre séquences. Livre vient de [Aanæs et al., 2012]. Les autres séquences sont personnelles.

vues, **A** où le modèle de **S** est enrichi par des descripteurs obtenus à partir de simulations affines (section 3.1.2), et **H** où le modèle de **S** est enrichi par des descripteurs obtenus à partir de simulations homographiques (section 3.1.1).

Pour comparer les trois scénarios, 100 poses sont calculées pour la même vue test dans chaque cas en utilisant le même nombre d'itérations de RANSAC. La variabilité de ces 100 poses est évaluée visuellement. Lorsque ces poses sont superposées, nous calculons également l'écart type (reporté dans les figures). L'échelle étant un paramètre libre de toute reconstruction SfM, les écart types sont exprimés en pourcentage de la distance à la scène. De plus, pour chaque expérience, des contours des objets de la scène sont reprojetés dans la vue test en utilisant les poses calculées.

Comme les taux d'*inliers* dans les correspondances image/modèle sont très variables d'une séquence à une autre (e.g., de 4 % à 23 % pour le scénario **S**), nous utilisons un nombre d'itérations de RANSAC différent pour chaque séquence. Cependant, pour rendre possible la comparaison de la variabilité, le même nombre d'itérations est utilisé pour les trois scénarios.

3.3.2 Amélioration du calcul de pose dans les modèles enrichis

Robustesse du calcul de pose aux changements de direction de vue

Nous montrons ici que la synthèse de vue améliore significativement la précision des poses calculées lorsque la vue test est éloignée des vues réelles et a donc un aspect très différent.

Nous présentons d'abord les résultats sur la séquence Livre (figure 3.2) pour laquelle la pose réelle de la vue test est connue. Il est donc possible de déterminer si une correspondance 2D/3D est correcte ou non, en reprojetant le point 3D en utilisant la pose de la vérité terrain. Si la distance de reprojection est inférieure à 20 pixels la correspondance est considérée correcte (ce seuil correspond à $\mu + 3\sigma$ avec μ et σ respectivement la moyenne et l'écart type de l'erreur de reprojection de l'étape SfM ; les images sont de taille 1600×1200 pixels). Dans cette expérience la proportion de correspondances correctes est de 23 % dans le scénario **S**, 30 % dans le scénario **A** et 37 % dans le scénario **H**.

La figure 3.5 montre la répartition des correspondances 2D/3D parmi les vues réelles et synthétiques dans le scénario **H**. Le point de vue qui contribue le plus au calcul de pose est virtuel et proche de la caméra test. Globalement, les vues synthétiques produisent 85 % de l'ensemble de consensus de RANSAC. Ces graphes illustrent la pertinence de l'approche proposée et l'augmentation du taux d'inliers obtenue grâce aux simulations.

Les résultats du calcul de pose sont illustrés dans les figures 3.6 ($N = 500$) et 3.7 ($N = 1\,000$). Les poses estimées sont visuellement plus précises dans les scénarios **A** et **H** que dans le scénario **S**. Avec 500 itérations dans RANSAC, le calcul de la pose échoue dans **S**, alors que les résultats sont corrects dans **H**. En augmentant le nombre d'itérations à 1 000, la variabilité de la pose n'est que légèrement réduite dans **S** alors que dans **H** toutes les poses calculées sont superposées.

Un phénomène remarquable se produit dans **A** (et dans une moindre mesure dans **S**). Dans cette expérience les poses calculées se répartissent en trois catégories : la plupart des poses sont proches du point de vue attendu, quelques unes sont totalement fausses et un groupe de poses erronées se trouve face à la couverture du livre. Cet ensemble d'erreurs est provoqué par un motif répété de la scène, à savoir l'œil de la couverture qui apparaît également sur la tranche du livre. La reprojection des bords de la couverture dans la figure 3.7 illustre bien le phénomène. Dans ce cas, les simulations homographiques produisent plus de correspondances en dehors de ce motif répété, ce qui permet d'obtenir des poses correctes dans **H**. L'influence des motifs répétés est discutée par exemple dans [Noury et al., 2010, Sur et al., 2013, Roberts et al., 2011].

Ces expériences ont été reproduites sur les séquences Poster et Bureau avec des résultats similaires, voir figures 3.8 et 3.9. Dans tous les cas présentés, la synthèse de vue améliore la précision de l'estimation de la pose, ce qui est illustré par la meilleure superposition des positions de caméra estimées ou des quadrilatères correspondant à la projection de contours 3D de la scène par les caméras estimées.

Robustesse du calcul de pose aux variations de distance par rapport à la scène

Comme expliqué dans la section 3.1, la simulation utilisant le modèle de caméra affine est indépendante de la distance du point de vue simulé à la scène. Bien que la simulation

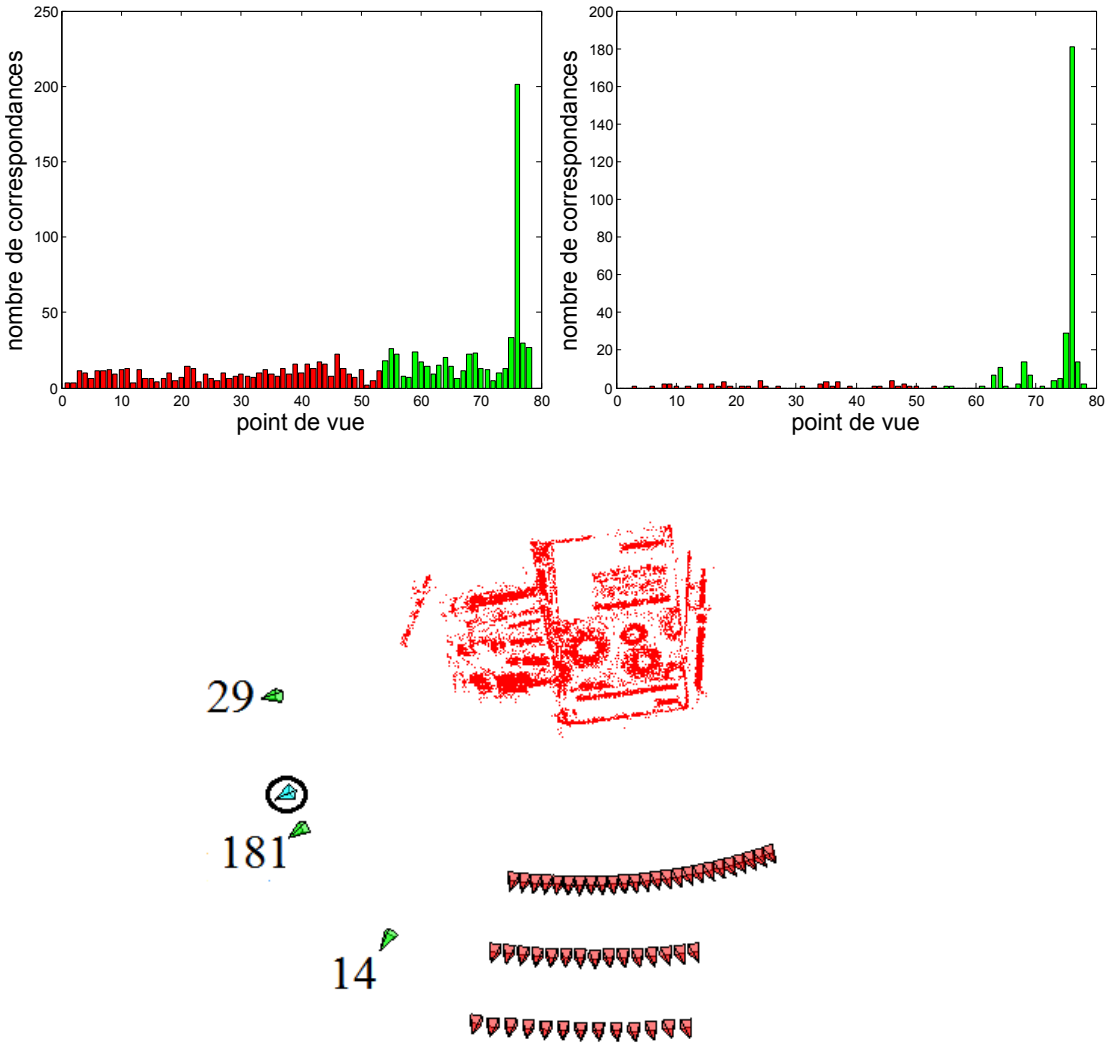


FIGURE 3.5 – Séquence Livre : nombre de correspondances associées à chaque point de vue (réel en rouge, virtuel en vert), pour l'ensemble des correspondances image/modèle (en haut à gauche) et dans l'ensemble de consensus de RANSAC (en haut à droite). Les points de vue contribuant le plus restent les mêmes, et sont proches de la pose cherchée. Les trois points de vue contribuant le plus et le nombre de correspondances associées sont montrés en bas.

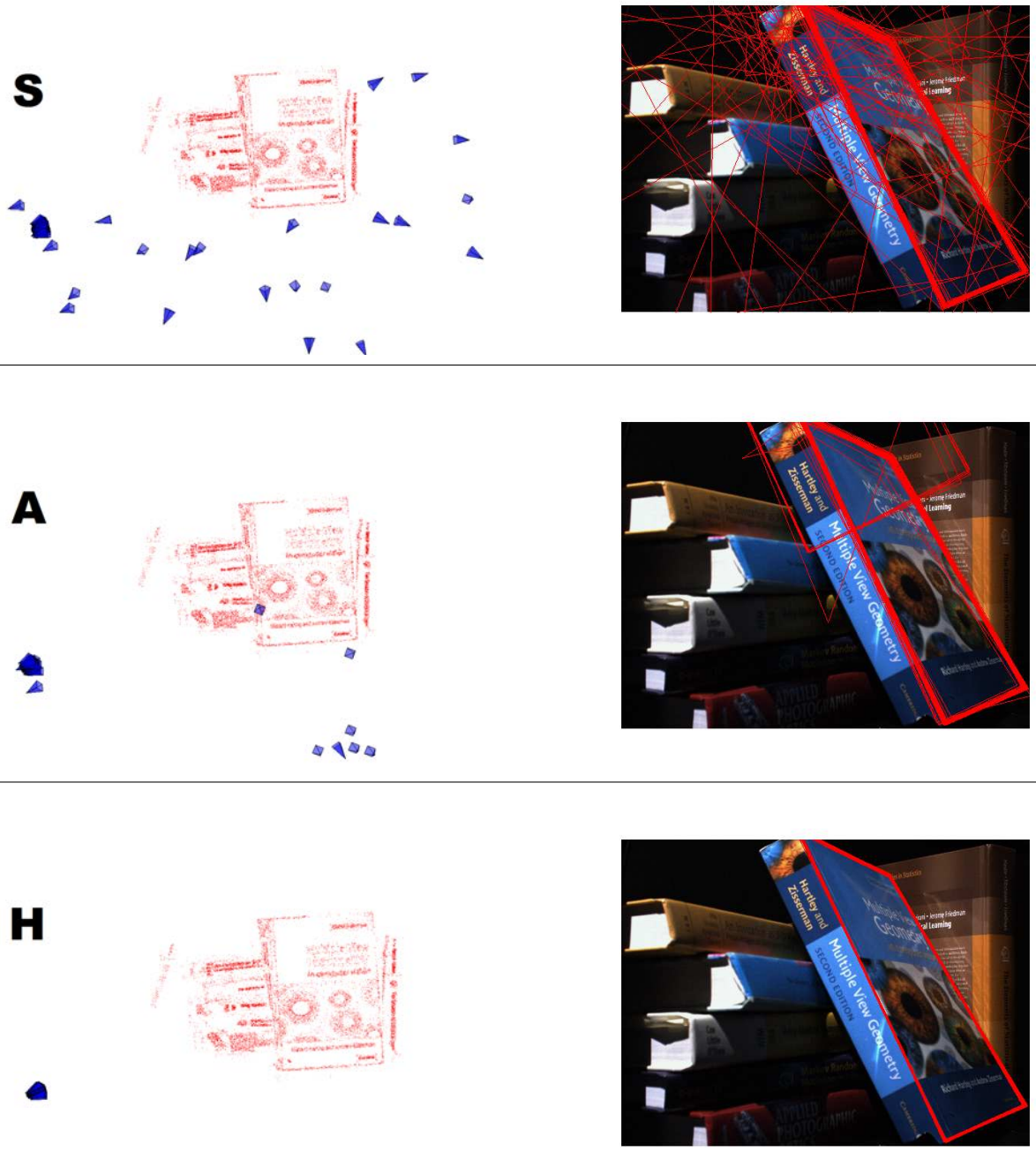


FIGURE 3.6 – Séquence Livre : 100 poses calculées avec N = 500 itérations de RANSAC, et la reprojection des bords de la couverture en utilisant ces 100 poses. Dans le scénario **H** l'écart type de la position de la caméra est 0,31 % de la distance à la scène.

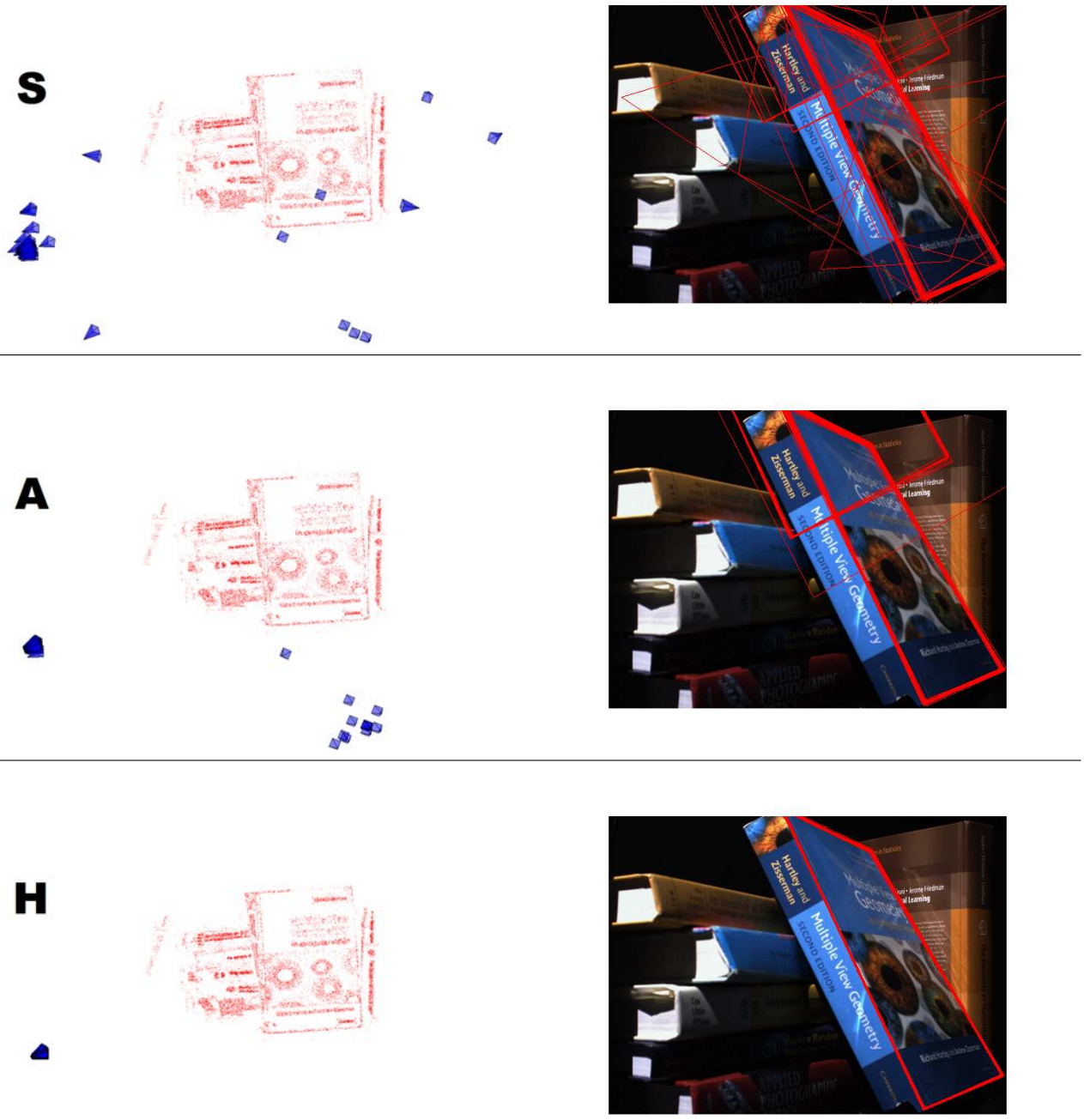


FIGURE 3.7 – Séquence Livre : 100 poses calculées avec N = 1 000 itérations de RANSAC, et la reprojection des bords de la couverture en utilisant ces 100 poses. Dans le scénario **H** l'écart type de la position de la caméra est 0,29 % de la distance à la scène.

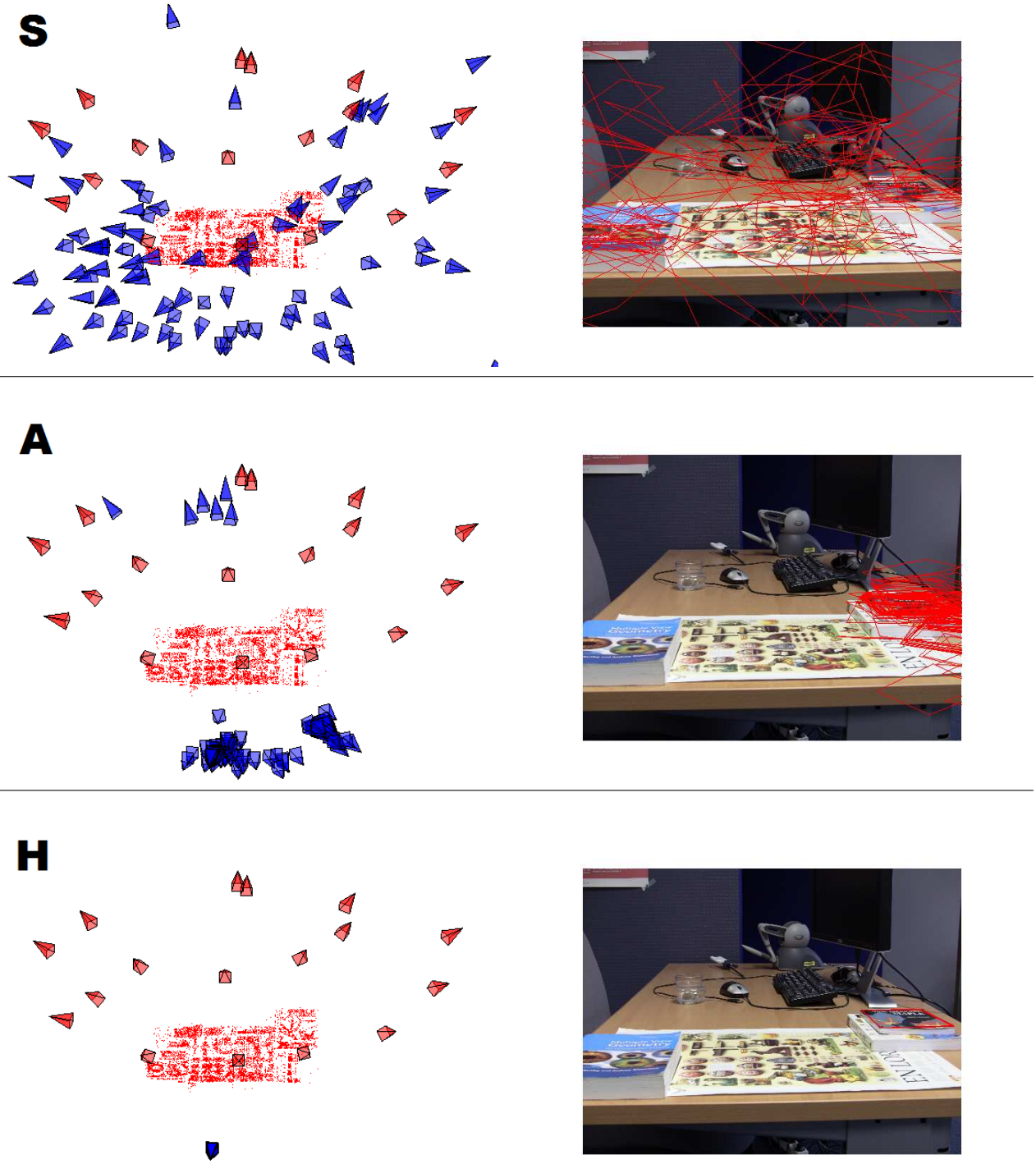


FIGURE 3.8 – Séquence Poster : 100 poses calculées avec N = 1 000 itérations de RANSAC, et la reprojection des bords du livre en utilisant ces 100 poses. Dans le scénario **H** l'écart type de la position de la caméra est 0,07 % de la distance à la scène.

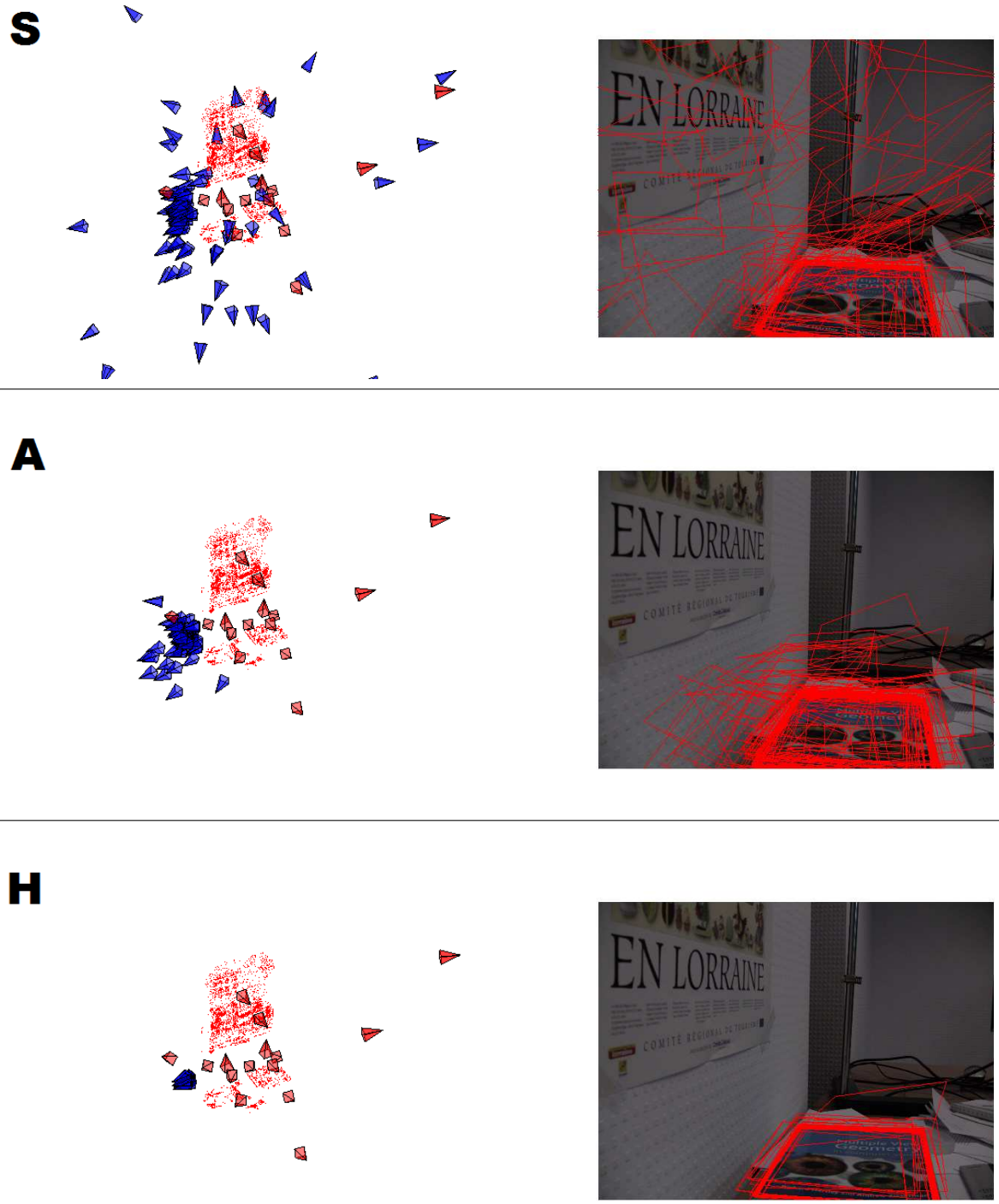


FIGURE 3.9 – Séquence Bureau : 100 poses calculées avec N = 5 000 itérations de RANSAC, et la reprojection des bords du livre de droite en utilisant ces 100 poses. Dans le scénario **H** l'écart type de la position de la caméra est 3,04 % de la distance à la scène.

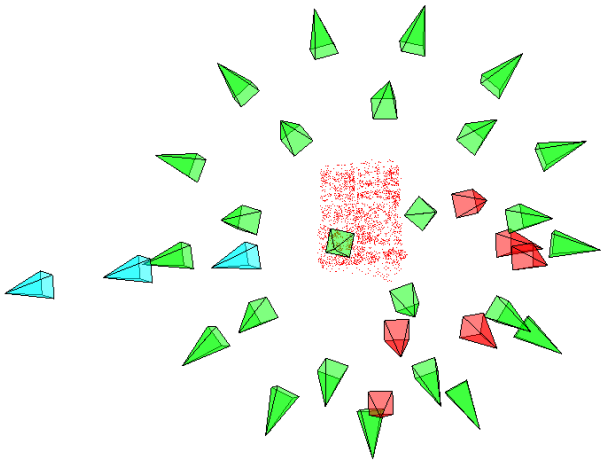


FIGURE 3.10 – Séquence Mur : position des caméras de reconstruction (rouge), des caméras virtuelles (vert) et des point de vue test (cyan) 1, 2 et 3.

par homographie dépende, elle, de cette distance, touts les points de vue simulés sont choisis à la même distance de la scène. L'objectif de cette expérience est de mettre en évidence l'influence de ce choix sur le calcul de pose lorsque le point de vue test est beaucoup plus éloigné de la scène que les points de vues utilisés pour la reconstruction.

Un modèle de la scène est construit à partir de 6 caméras orientées vers un poster (en rouge dans la figure 3.10). Cette scène a été choisie pour mettre en évidence l'apport de la synthèse de vue : nous avons besoin d'une caméra test non alignée avec l'axe optique des autres caméras et qui n'observe pas le poster en vue frontale, de telle sorte que la transformation résultante soit une homographie non réduite à une similarité.

Les vues de test sont donc prises avec un changement de direction de vue relativement faible mais de fortes variations de profondeur, voir figures 3.10 et 3.11. Le nombre d'itérations de RANSAC est $N = 300$ pour toutes ces expériences. Nous ne détaillons que les scénarios **S** et **H**, le scénario **A** produisant les même résultats que **S**. En effet le modèle de transformation affine ne prend pas en compte les transformations liées à un changement de profondeur.

La figure 3.12 montre les résultats dans le scénario **S**. On constate qu'une bonne estimation de la pose n'est possible que dans le cas où la vue test est proche des vues réelles, ce qui est le cas des vues 1 et 2. Par contre pour la vue 3 la précision est largement moindre. La figure 3.13 montre les résultats dans le scénario **H**. On constate que la pose est estimée avec précision dans l'ensemble des cas, les poses étant visuellement superposées.



(1) (2) (3)

FIGURE 3.11 – Séquence Mur : les trois vues de test utilisées pour évaluer la robustesse du calcul de pose par rapport à la distance à la scène.

3.3.3 Comparaison des modèles affine et homographique

Deux modèles de transformation ont été utilisés pour simuler les points de vues, homographique et affine. Dans toutes les expériences réalisées, les résultats sont meilleurs lorsqu'on utilise le modèle homographique, en terme de taux d'inliers parmi les correspondances image-modèle, de nombre de correspondances correctes et de précision des poses calculées.

L'utilisation de simulation affine est largement répandue dans les applications où il n'y a pas de modèle de scène et pour lesquelles définir des positions de caméras virtuelles n'a pas de sens bien défini. Si on a la possibilité de placer les caméras virtuelles dans le repère de la scène, comme c'est le cas dans notre contexte, il est préférable d'utiliser un modèle homographique plutôt qu'une approximation affine.

3.3.4 Ambiguïté due aux vues symétriques

Comme remarqué dans [Morel and Yu, 2009], avec un modèle de caméra affine un plan a la même apparence observé avec deux points de vue symétriques par rapport à la normale au plan (cf. figure 3.14), à une rotation d'image de 180 près. C'est ce qui justifie de ne simuler que par l'intermédiaire de caméras virtuelles situées sur un demi hémisphère dans l'algorithme ASIFT. Dans notre cas, la scène est composée de plusieurs plans et il n'y a donc pas de raison a priori pour se limiter à un demi hémisphère pour placer les points de vue virtuels. En effet, chaque plan possède un axe de symétrie, mais la scène dans son ensemble n'en n'a pas. Comme nos caméras virtuelles sont placées par rapport à la scène dans son ensemble cette symétrie ne doit a priori pas intervenir.

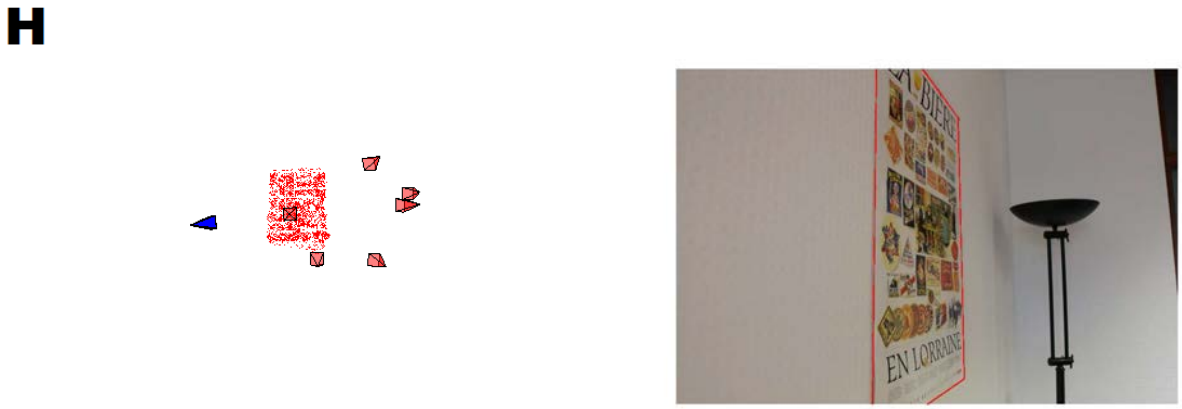
Cependant, dans certaines scènes dominées par un plan (Poster et Mur) on peut clairement observer l'influence de cette symétrie (figures 3.16 et 3.15).

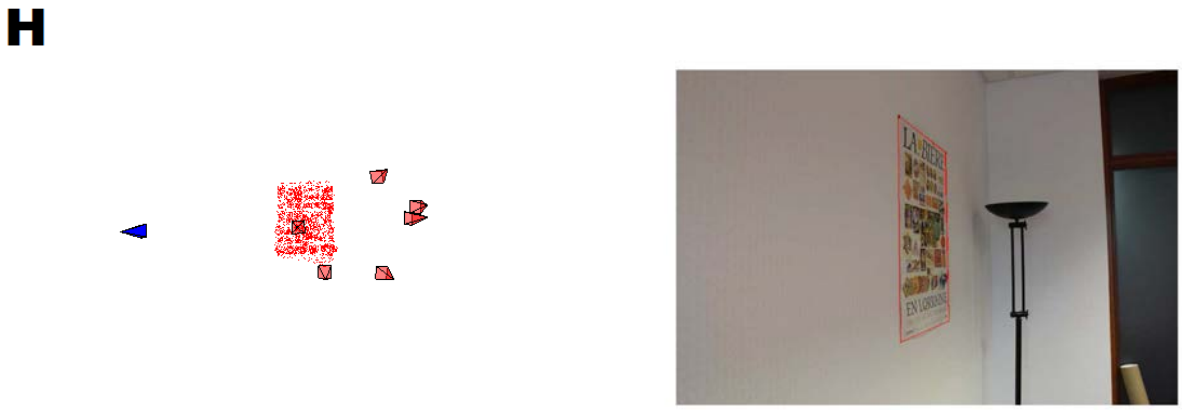
Dans la séquence Poster, on observe que les points de vue contribuant le plus à la mise en correspondance image/modèle sont un point de vue virtuel proche de la pose test et un autre point de vue virtuel symétrique du premier (figure 3.15).

Dans la séquence Mur, les points d'intérêts extraits de la vue test sont concentrés dans une faible portion de l'image. Ces points d'intérêts ont la même apparence avec la pose test correcte et la pose symétrique, et leur répartition ne permet plus de différencier les deux (figure 3.16).



FIGURE 3.12 – Séquence Mur : 100 calculs de pose avec N = 300 itération de RANSAC pour les trois vues de test (voir 3.11) dans le scénario **S**. De gauche à droite : les vues test 1 à 3. L'écart type de la position de la caméra est 2,14 % de la distance à la scène pour la vue 1 et 0,12 % pour la vue 2.





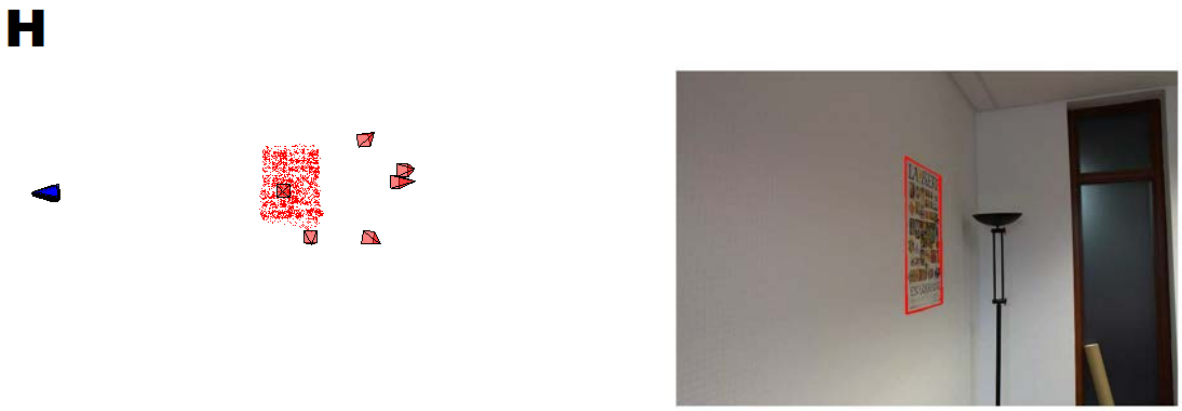


FIGURE 3.13 – Séquence Mur : 100 calculs de pose avec N = 300 itération de RANSAC pour les trois vues de test (voir 3.11) dans le scénario **H**. De gauche à droite : les vues test 1 à 3. L'écart type de la position de la caméra est 0,07 % de la distance à la scène pour la vue 1, 0,02 % pour la vue 2 et 0,28 % pour la vue 3.

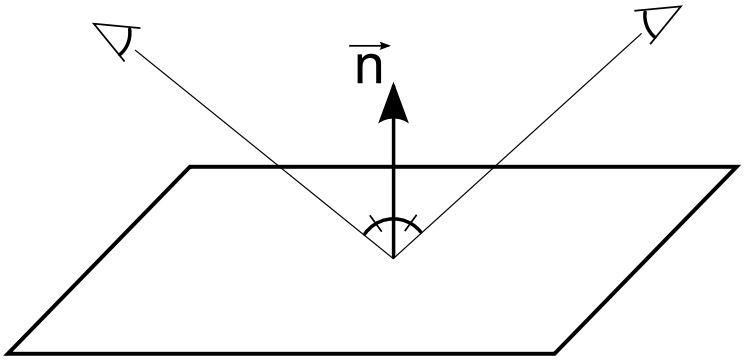


FIGURE 3.14 – Deux points de vue symétriques selon la normale $\vec{n}$ pour lesquels, avec le modèle affine, le plan a la même apparence à une rotation selon l'axe optique de 180 deg près.

Ces ambiguïtés se présentent donc dans des cas dégénérés où le modèle se comporte essentiellement comme un plan seul, soit parce que la scène est effectivement proche d'un plan soit parce que les seuls points utilisés dans la mise en correspondance sont sur un même plan. On peut également noter que ce phénomène se produit bien qu'on utilise un modèle de transformation homographique pour lequel la symétrie n'est normalement pas présente. Les effets de perspective sont en effet peu marqués à l'échelle des patchs synthétiques qui sont générés, et les patchs obtenus par des vues symétriques sont suffisamment semblables pour produire les même descripteurs.

3.4 Conclusion

Ce chapitre présente les principes de base de notre approche de synthèse de vues et apporte deux contributions principales.

Premièrement, on met en évidence la pertinence de la synthèse de vues dans le cadre du positionnement par rapport à un modèle SfM peu dense. En particulier, on montre que sous l'hypothèse que la scène est localement plane il est possible de générer des patchs réalistes à partir de position de caméras virtuelles. Les expériences montrent que l'utilisation de cette technique permet de calculer des poses dans des conditions où des méthodes standard échouent. L'utilité des descripteurs ajoutés au modèle apparaît clairement car les vues éloignées sont mises en correspondance presque exclusivement grâce à ces descripteurs.

Deuxièmement, les expériences montrent que le modèle sténopé pour les caméras, et donc homographique pour les transformations de plans, donne de meilleurs résultats que l'approximation affine.

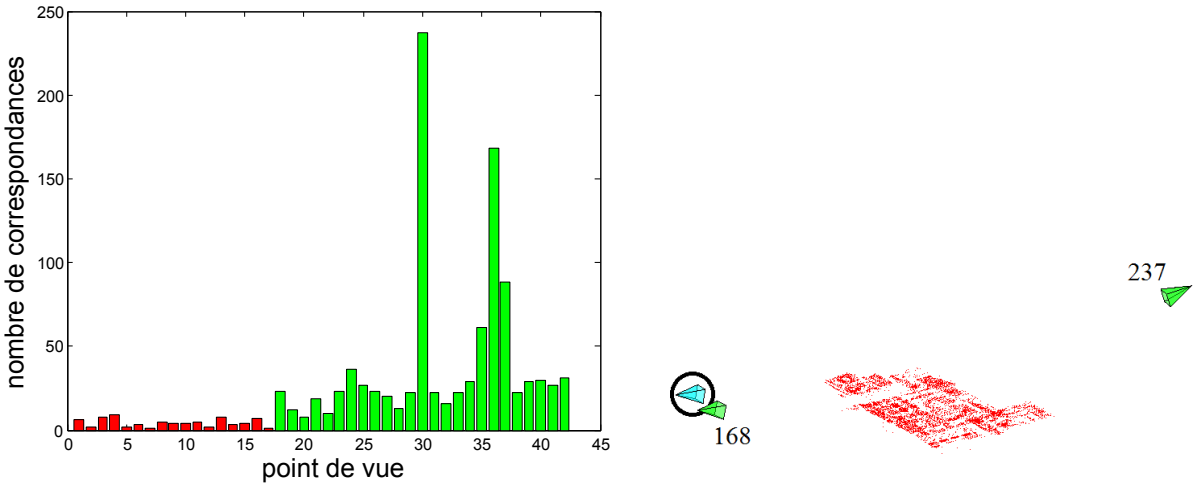


FIGURE 3.15 – Séquence Poster : nombre de correspondances associées à chaque point de vue (réel en rouge, virtuel en vert), pour l'ensemble des correspondances image/modèle (à droite). Les deux points de vue correspondant aux pics dans l'histogramme sont un point de vue proche de la pose test et le point de vue symétrique (à droite ;les contributions respectives des deux points de vue sont indiquées).

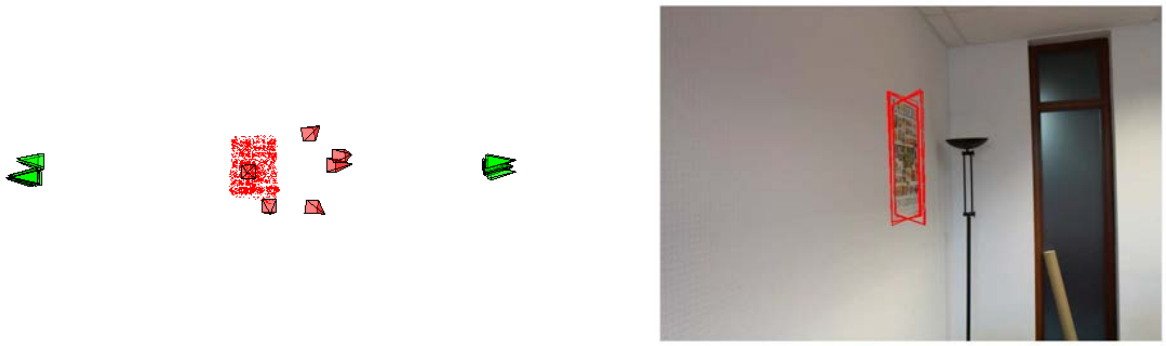


FIGURE 3.16 – Séquence Mur : 100 calculs de pose avec N = 200 itérations de RANSAC pour la vue 3. Deux groupes de poses sont calculés, qui correspondent respectivement à la pose test (gauche) et à la position symétrique (droite).

L'algorithme présenté ici peut cependant être amélioré sur plusieurs aspects. Le placement des points de vue de synthèse ne convient qu'à des scènes centrées sur un objet particulier. Le temps de calcul nécessaire pour générer les patchs synthétiques et en extraire des descripteurs est considérable, de l'ordre de plusieurs heures. La recherche de correspondance entre les descripteur de l'image et ceux du modèle est également très coûteuse en temps de calcul. La prise en compte de ces considération est l'objet du chapitre 5.

Chapitre 4

Recherche efficace de correspondances

Le calcul de la pose s'appuie sur des correspondances entre des points 3D dans le modèle de la scène et des points 2D dans l'image requête. Ces correspondances sont obtenues en deux temps : une recherche de correspondances candidates, typiquement en exploitant les descripteurs photométriques associés aux points, puis un filtrage de ces correspondances par un critère géométrique. Dans notre cas, le critère géométrique est la cohérence par rapport à une pose donnée et on utilise RANSAC pour déterminer un groupe de correspondances cohérentes avec une pose. RANSAC est une méthode robuste de recherche de modèle à partir de données bruitées mais ses performances, en terme de précision et de temps de calcul, peuvent varier fortement selon l'application. Dans le cadre de notre étude l'application de RANSAC se heurte principalement à deux difficultés.

Premièrement, le nombre de correspondances candidates peut être élevé, ce qui entraîne une augmentation du temps de calcul pour RANSAC. En effet, la méthode présentée dans le chapitre 3 augmente significativement le nombre de descripteurs présents dans le modèle. Comme les correspondances candidates sont obtenues en comparant les distances au plus proche et au deuxième plus proche voisin (voir 3.2.3), les points extraits dans l'image ont plus de chance de trouver un voisin parmi les points du modèle. Par conséquent, le nombre de correspondances candidates est augmenté car les points de l'image sont plus susceptibles de trouver un voisin proche parmi les points du modèle.

Deuxièmement, la proportion de correspondances correctes varie fortement d'une expérience à une autre. Dans les expériences qui ont été faites nous avons observé des valeurs allant de 5% à 60% pour cette proportion. Cette proportion dépend de la difficulté à mettre en correspondance les descripteurs de l'image avec ceux du modèle et elle est donc liée, par exemple, à la distance entre le point de vue de l'image requête et les points de vue utilisés pour construire le modèle, à la texture de la scène et à la présence de motifs répétés. Ces éléments varient d'une scène à une autre, et dépendent aussi du point de vue qu'on cherche à calculer, ce qui explique le taux de correspondances correctes particulièrement variable. Comme la vitesse de convergence de RANSAC est directement liée au taux de correspondances correctes, le temps nécessaire pour converger vers une pose correcte est également variable.

L'objectif de ce chapitre est de montrer qu'il est possible dans ces conditions de sélectionner un ensemble de correspondances correctes en un temps raisonnable pour une initialisation de pose. Une méthode de type RANSAC est proposée pour faire cette re-

cherche de correspondances, dans la lignée des méthodes de type *guided matching* qui consistent à utiliser préférentiellement certaines correspondances sur la base d'informations a priori.

4.1 RANSAC et accélérations potentielles

entrées: $\mathcal{E}$: un ensemble de correspondances
t_{stop} : le nombre d'itérations à faire
begin
  $t \leftarrow 0$;
  while $t < t_{stop}$ **do**
    $t \leftarrow t + 1$;
    tirer un échantillon s dans $\mathcal{E}^4$;
    calculer la pose $\mathcal{P}$ à partir de s;
    if $\#I(\mathcal{P}) > \#I(\mathcal{P}^*)$ **then**
      $\mathcal{P}^* \leftarrow \mathcal{P}$;
    end
  end
  calculer la pose $\mathcal{P}_{\text{final}}$ à partir de $I_N(\mathcal{P}^*)$;
end
sorties : la pose de la caméra $\mathcal{P}_{\text{final}}$

Algorithm 1: L'algorithme RANSAC standard dans le contexte du calcul de pose. $I(\mathcal{P})$ désigne l'ensemble de consensus associé à la pose $\mathcal{P}$.

4.1.1 RANSAC standard

RANSAC, pour *RANdom SAmple Consensus*, introduit par [Fischler and Bolles, 1981], est un procédé de vote permettant d'estimer un modèle à partir d'un ensemble de données partiellement erronées. Cette section présente cet algorithme, auquel on référera par la suite comme étant la version standard de RANSAC, et introduit les notations spécifiques utilisées dans ce chapitre.

L'algorithme prend en entrée un ensemble de données partiellement erronées $\mathcal{E}$. Soit un modèle $\mathcal{M}$ et une donnée $e \in \mathcal{E}$, on appelle $d(\mathcal{M}, e)$ la distance de e à $\mathcal{M}$, d étant une métrique spécifique à chaque problème. Dans le cadre du calcul de pose, par exemple, e est une correspondance image-modèle, $\mathcal{M}$ une pose et $d(\mathcal{M}, e)$ est l'erreur de reprojection. e est considérée *inlier*, c'est-à-dire cohérente avec le modèle, si $d(\mathcal{M}, e) < d^*$, d^* étant un seuil à fixer. Dans le cas contraire on appelle e un *outlier*. On appelle *ensemble de consensus* d'un modèle $\mathcal{M}$ l'ensemble des données cohérentes avec ce modèle. RANSAC fonctionne sur l'hypothèse que les données ont été générées par un modèle sous-jacent et que ce modèle est celui avec le plus grand ensemble de consensus.

L'algorithme fonctionne comme suit. On tire uniformément parmi l'ensemble des données le minimum de données requises pour estimer un modèle, cet ensemble de données

est appelé un *échantillon*. Un échantillon est considéré correct si tous ses éléments sont des inliers et incorrect sinon. Un modèle est calculé à partir de cet échantillon. Chaque donnée vote ensuite selon qu'elle est cohérente ou non avec le modèle estimé d'après une métrique à définir pour chaque application. L'ensemble des données cohérentes avec le modèle est son *ensemble de consensus*. On note $I(\mathcal{M})$ l'ensemble de consensus d'un modèle $\mathcal{M}$. On itère ce procédé jusqu'à ce qu'un critère d'arrêt soit atteint et le modèle avec le plus grand ensemble de consensus est retenu. L'algorithme est détaillé dans la figure 1. Plusieurs paramètres doivent être fixés : la métrique pour estimer la distance d'un point à un modèle, le seuil sur cette distance pour décider si un point est cohérent ou non avec un modèle et le nombre d'itérations à faire.

La vitesse de convergence de RANSAC est directement liée au taux d'inliers parmi l'ensemble des données. Soit τ le taux d'inliers et m le nombre d'éléments à tirer pour former un échantillon. À chaque itération de l'algorithme la probabilité de tirer un échantillon correct, c'est-à-dire un échantillon dont tous les éléments sont inliers, est de τ^m. Si on fixe le nombre total d'itérations à t_{max}, la probabilité de tirer au moins un échantillon correct est :

$$p = 1 - (1 - \tau^m)^{t_{max}}$$

Inversement, on peut estimer le nombre d'itérations à faire pour atteindre une certaine probabilité p^* de tirer au moins un échantillon correct :

$$t_{max} = \frac{\log(1 - p^*)}{\log(1 - \tau^m)} \tag{4.1}$$

RANSAC est utilisé pour sa robustesse et sa facilité d'implémentation, mais son temps de convergence dépend du taux d'inlier, et par extension le nombre d'itérations à faire pour avoir une forte probabilité de converger vers un modèle correct. Par exemple, pour $p^* = 0.95$ et $\tau = 0.05$ on a $t_{max} = 47932$ alors que pour $\tau = 0.6$ $t_{max} = 21$. Dans un contexte ou le taux d'inliers varie fortement d'une expérience à une autre, il n'est pas possible de fixer a priori ce nombre d'itérations. On pourrait le fixer de façon pessimiste, mais cela entraînerait des calculs inutiles dans les cas où le taux d'inliers est élevé.

4.1.2 Famille des méthodes RANSAC

À partir de l'algorithme standard ci-dessus toute une famille de méthodes a été développée, [Choi et al., 2009, Raguram et al., 2008] présentent les plus connues. Ces méthodes ont pour objectif d'améliorer certains aspects de l'algorithme standard, plus spécifiquement : améliorer la précision du modèle calculé, diminuer le temps de calcul ou augmenter la robustesse aux conditions expérimentales. La Figure 4.1 présente une synthèse de l'organisation des méthodes type RANSAC.

Précision Dans RANSAC standard le score d'un modèle est le nombre de données dont l'erreur par rapport à ce modèle est en dessous d'un seuil donné comme paramètre. Les méthodes destinées à améliorer la précision de RANSAC telles que MSAC [Torr and Zisserman, 2000] remplacent cette fonction de coût par un estimateur robuste. RANSAC standard fait également l'hypothèse qu'un modèle évalué à partir d'un échantillon correct

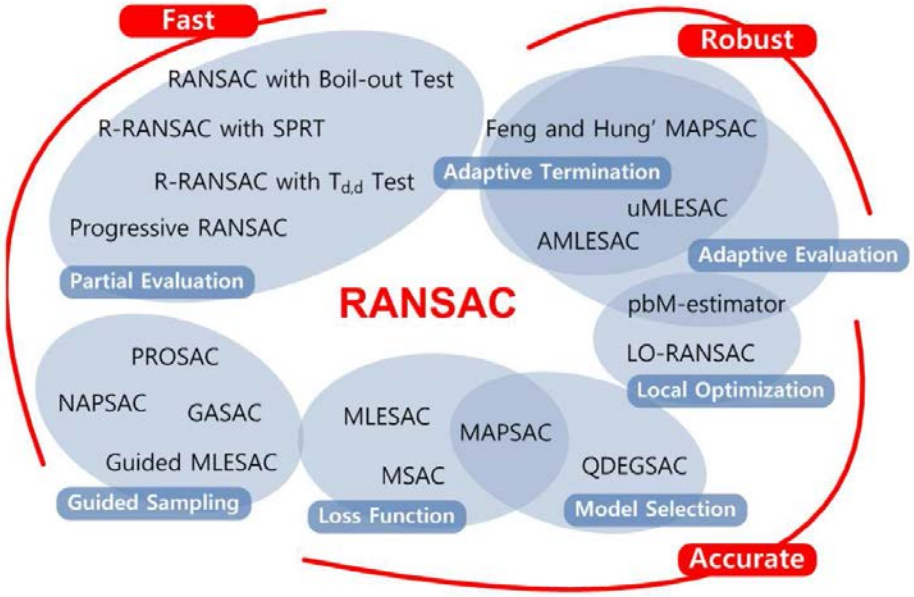


FIGURE 4.1 – Les méthodes type RANSAC, classées selon les aspects qu'elles améliorent par rapport à RANSAC standard (image extraite de [Choi et al., 2009]).

est cohérent avec tous les inliers, ce qui n'est pas nécessairement vérifié en pratique. LO-SAC introduit dans [Chum et al., 2003] optimise localement la solution en ré-estimant le modèle à partir de l'ensemble de consensus calculé. Ceci réduit la variance des modèles estimés et permet de converger plus rapidement vers une solution plus précise. Ces méthodes sont utiles lorsqu'on n'a pas de moyen de fixer le seuil d'acceptation des inliers de façon raisonnable. Dans notre situation, la métrique pour mesurer la cohérence d'une correspondance avec une pose est l'erreur de reprojection. Comme nous connaissons l'erreur de reprojection dans le SfM nous pouvons l'utiliser comme seuil pour décider si une correspondance est inlier ou outlier.

Robustesse RANSAC est particulièrement sensible à ses deux paramètres : le nombre d'itérations et le seuil d'acceptation des inliers. Les méthodes pour améliorer la robustesse de RANSAC consistent à estimer ou raffiner ces paramètres pendant l'exécution de l'algorithme afin qu'ils soient plus adaptés aux conditions expérimentales. Il est par exemple possible de fixer le nombre d'itérations dynamiquement, comme proposé par [Hartley and Zisserman, 2004]. Le taux d'inliers τ est estimé pendant l'exécution à partir du cardinal du plus grand ensemble de consensus rencontré :

$$\tau = \frac{\#I(P)}{N}$$

En supposant que les échantillons sont tirés indépendamment et que chaque correspondance a une probabilité τ d'être correcte, on peut montrer qu'un échantillon correct est

tiré avec probabilité p^* après t^* itérations avec :

$$t^* = \frac{\log(1-p^*)}{\log(1-\tau^4)} \tag{4.2}$$

En fixant la probabilité p^*, typiquement 0.95, on peut estimer pendant l'exécution de l'algorithme le nombre d'itérations nécessaires. Il est également possible de fixer dynamiquement le seuil d'acceptation des inliers, comme dans [Choi et al., 2009, Feng and Hung, 2003], en utilisant des distribution de probabilités paramétriques pour décrire les données. Comme expliqué précédemment, décider si une donnée est inlier n'est pas le problème principal pour notre application, car nous pouvons raisonnablement fixer a priori un seuil d'acceptation.

Temps de calcul Il existe essentiellement deux approches pour accélérer RANSAC : réduire le temps utilisé pour évaluer les échantillons ou réduire le nombre d'itérations. Réduire le temps d'évaluation des échantillons consiste généralement à ne pas l'évaluer sur l'ensemble des données, comme dans RANSAC avec tests $T_{d,d}$ [Nistér, 2005] ou R-RANSAC [Matas and Chum, 2004]. Réduire le nombre d'itérations est possible en orientant le tirage des échantillons en utilisant des probabilités a priori d'être inlier pour chaque donnée. C'est l'approche utilisé dans PROSAC [Chum and Matas, 2005] ainsi que dans [Botterill et al., 2009, Li et al., 2012]. Dans notre contexte l'objectif est de définir un algorithme aussi rapide que possible tout en restant robuste au taux d'inliers fortement variable. On s'intéresse donc à ces deux types de méthodes, qui sont discutées dans les sous-sections suivantes.

Évaluation partielle

Une première approche pour accélérer RANSAC consiste à écarter rapidement des échantillons incorrects en vérifiant leur cohérence sur un petit nombre de correspondances plutôt que sur l'ensemble complet.

[Nistér, 2005] propose de tester des groupes d'échantillons sur quelques données, au lieu de tester un échantillon sur l'ensemble des données. Pour chaque itération de RANSAC on suit la procédure suivante. On tire plusieurs échantillons et on estime un modèle pour chacun d'eux. Ces modèles sont testés par rapport à l'une des données. Les modèles avec les moins bon scores sont éliminés et on conserve les autres. On itère ce procédé jusqu'à conserver un seul modèle. Ce procédé de sélection d'un modèle s'appuie sur le fait qu'un échantillon a toujours moins de chances d'être correct qu'une donnée et qu'il est donc raisonnable d'écarter un échantillon sur la base d'un test avec une donnée.

[Matas and Chum, 2004] proposent de rejeter les modèles qui ne sont pas cohérents avec un petit groupe de données tirées aléatoirement. Si le modèle n'est pas rejeté on calcule son ensemble de consensus sur l'ensemble des données comme dans l'algorithme standard. Le nombre de données utilisées pour tester le modèle est un paramètre critique et dépend du taux d'inliers : plus celui-ci est élevé, plus il est raisonnable d'écarter un modèle sur la base de quelques tests.

[Svarm et al., 2014] introduit une méthode de rejet des outliers basé sur des considérations géométriques, sous l'hypothèse que l'orientation de la caméra est connue. Dans

un premier temps, une borne inférieure L sur le nombre d'inliers est estimée. Ensuite, pour chaque donnée e, on peut calculer une borne supérieure B_e sur le nombre d'inliers, grâce à l'orientation connue de la caméra. Si $B_e < L$ on peut rejeter définitivement la donnée. Cette approche diminue considérablement le nombre d'outliers dans les données et accélère donc significativement RANSAC. Cette méthode n'est pas utilisable dans notre contexte car on ne peut pas estimer a priori l'orientation de la caméra.

L'accélération obtenue par ce type d'approche dépend largement du taux d'inliers. En particulier, ce type de méthode est peu adapté aux situations où le taux d'inliers est faible puisqu'il faudrait de toute façon faire un grand nombre de tests avant de pouvoir écarter une hypothèse. Dans notre contexte le taux d'inliers peut varier significativement selon que l'image dont on cherche la pose est proche ou non des vues utilisées pour construire le modèle. En absence d'estimation raisonnable du taux d'inliers ce type de méthode est difficile à mettre en place.

Mise en correspondance guidée

Dans la version standard de RANSAC, lorsque les hypothèses sont sélectionnées, le tirage est uniforme parmi l'ensemble des données. Un certain nombre de travaux montrent que biaiser ce tirage, sur la base d'un a priori sur la qualité de chaque donnée, permet d'accélérer considérablement la convergence de l'algorithme.

[Botterill et al., 2009] proposent d'associer une probabilité d'être inlier à chaque correspondance puis d'actualiser cette probabilité au cours de l'exécution de l'algorithme. Ces probabilités sont utilisées pour biaiser le tirage des échantillons en faveur de données avec une haute probabilité d'être inlier. Cette approche nécessite un a priori sur la probabilité d'être inlier ou alternativement de faire des simulations pour l'estimer, ce qui est très coûteux en temps de calcul.

Dans le contexte du calcul de pose à grande échelle, [Li et al., 2012] utilisent la co-occurrence des correspondances correctes pour associer un score aux hypothèses tirées, en observant que les descripteurs des correspondances correctes apparaissent généralement ensemble dans les même images. Ce score porte sur les échantillons complets, c'est à dire des groupes de correspondances, contrairement aux méthodes précédentes qui associent un score individuellement à chaque correspondance.

Dans le contexte de la mise en correspondances image-image, [Chum and Matas, 2005] classent les correspondances en fonction d'un a priori et tirent les échantillons dans des sous-ensembles composés de correspondances ayant une forte probabilité a priori d'être inlier. Cette méthode est proche de celle que nous proposons et est détaillée dans la section suivante 4.2.

Ces méthodes permettent de trouver plus rapidement un échantillon correct que l'algorithme RANSAC standard. Dans le cadre de l'initialisation de pose, le taux d'inliers est potentiellement très faible et le nombre de tirages à faire pour trouver un échantillon correct est alors très long. Les méthodes accélérant cette recherche sont donc bien adaptées à notre problème et c'est ce type d'approche que nous développons.

4.2 PROSAC

Dans notre contexte nous avons de l'information a priori sur la qualité des données, qui sont les correspondances image-modèle. Cette section présente une méthode efficace lorsqu'on peut donner un a priori, PROSAC (*PROgressive SAmple Consensus*), dont la structure est proche de la méthode que nous proposons. PROSAC a été proposé par [Chum and Matas, 2005] et s'inscrit dans la lignée des algorithmes de mise en correspondances guidée. Il a été proposé dans le contexte de la mise en correspondance d'images, sans modèle 3D.

L'algorithme fonctionne comme suit. Dans un premier temps, un score est calculé pour chaque correspondance. Ce score donne un a priori sur le fait qu'une correspondance soit correcte ou non. Lorsqu'on tire des échantillons, on commence par les correspondances avec le plus haut score puis on ajoute progressivement des correspondances à l'ensemble dans lequel les échantillons sont tirés.

Dans PROSAC, le score associé à une correspondance utilise la distance entre les descripteurs utilisés pour obtenir cette correspondance. Soit I_1 et I_2 les images entre lesquelles on cherche des correspondances, soit f_1 et f_2 deux points d'intérêts respectivement dans I_1 et I_2. La distance $d(f_1, f_2)$ entre deux points d'intérêt est la distance euclidienne entre les descripteurs associés, SIFT dans le cadre des travaux présentés par [Chum and Matas, 2005]. On a une correspondance candidate entre f_1 et f_2 si f_2 est le plus proche voisin de f_1 dans I_2, au sens de la distance définie ci-dessus. Soit f_2' le second plus proche voisin de f_1 dans I_2. Le score associé à la correspondance entre f_1 et f_2 est :

$$s = \frac{d(f_1, f_2)}{d(f_1, f_2')}$$

On peut noter que ce score est généralement utilisé pour éliminer les correspondances lorsqu'il dépasse un certain seuil, typiquement 0.8 [Lowe, 1999]. Dans PROSAC toutes les correspondances sont conservées et ce score est utilisé comme prior sur la qualité des correspondances.

L'objectif de PROSAC est de tirer les mêmes échantillons que RANSAC standard mais dans un ordre différent. Soit $e_1, e_2, \ldots, e_N$ la suite des N correspondances candidates, ordonnées par scores croissants et $\mathcal{E}_n = \{e_1, \ldots, e_n\}$ pour tout $1 \leq n \leq N$ les ensembles imbriqués des n premières correspondances. On note m la taille des échantillons. On considère un ensemble de T_N échantillons tirés dans $\mathcal{E}$. PROSAC tire d'abord les échantillons de $\mathcal{E}_4$, puis $\mathcal{E}_5$, etc. La proportion d'échantillons tirés dans $\mathcal{E}_n$ est $\binom{n}{m} / \binom{N}{m}$, soit T_n échantillons :

$$T_n = T_N \frac{\binom{n}{m}}{\binom{N}{m}}$$

Comme T_n ne prend pas nécessairement de valeurs entières, la suite T_n' est introduite, définie par :

$$T'_m = 1 \text{ et } T'_{n+1} = T'_n + \lceil T_{n+1} - T_n \rceil$$

où $\lceil x \rceil$ est le plus petit entier supérieur ou égal à x. Le t-ième échantillon de PROSAC est tiré dans $\{e_{g(t)}\} \cup \mathcal{E}_{g(t)-1}$ où g est une fonction de croissance définie par $g(t) = \min\{n : T'_n \geq t\}$.

La première condition d'arrêt est la *non randomness* : l'ensemble de consensus trouvé doit avoir une faible probabilité, typiquement inférieure à 0.05, de consister en un ensemble de correspondances cohérente avec un même modèle par hasard. Cette condition est implémentée en imposant une taille minimum sur l'ensemble de consensus pour qu'une solution soit acceptable. On considère la distribution des cardinaux des ensembles de consensus des modèles incorrects, c'est à dire les modèles estimés à partir d'échantillons incorrects :

$$P_n(i) = \beta^i (1-\beta)^{m-i} \binom{n-m}{i-m}$$

où le paramètre β est comme la probabilité qu'une correspondance soit cohérente avec un modèle incorrect. Si on fixe une probabilité ψ que l'ensemble de consensus ne soit pas composé de correspondances cohérentes par hasard, on peut donner une borne inférieure sur la taille de l'ensemble de consensus :

$$I_n^{min} = min\{j : \sum_{i=j}^{n} P_n(i) < \psi\} \tag{4.3}$$

Les auteurs proposent d'estimer β par des considérations géométriques sans donner plus de précisions, il est généralement fixé à 0.01 dans les implémentations disponibles. Dans le cadre de l'estimation de pose donner une estimation raisonnable de β a priori n'est pas possible, étant donné qu'il dépend, par exemple, de la répartition dans l'espace des points reconstruits et de la pose qu'on cherche. Cela pose un problème car β a un impact direct sur le nombre d'itérations faites par l' algorithme. En effet, s'il est fixé à une valeur haute, tous les ensembles de consensus sont considéré comme étant obtenus par hasard et rejetés. C'est le cas si $\beta > 0.2$ dans la plupart de nos expériences. Dans ce cas l'algorithme tel qu'il est décrit ne termine pas, d'où la nécessité de fixer β de façon pessimiste. À l'inverse, s'il est fixé à une valeur trop faible, typiquement 0.01 pour nos expériences, tous les ensembles de consensus sont acceptables et le critère devient inutile. Il arrive alors qu'on accepte des modèles erronés. Dans notre contexte, $\beta = 0.1$ semble expérimentalement être une valeur raisonnable.

La deuxième condition d'arrêt est la *maximality* : on veut que la probabilité de tirer un échantillon correct soit supérieure à une borne p^* typiquement fixée à 0.95. Cette condition fixe une borne t^* sur le nombre d'itérations à faire à partir d'une estimation en ligne du taux d'inliers τ, comme dans l'équation 4.2 :

$$t^* = \frac{\log(1-p^*)}{\log(1-\tau^4)}$$

τ est uniquement évalué sur les n^* premières correspondances, où n^* est choisi pour minimiser t^* sous contrainte que la *non randomness* soit vérifiée, c'est-à-dire que $I_{n^*} > I_{n^*}^{min}$

avec les notations de l'équation 4.3. Cette dernière condition évite de sélectionner un n^* trop petit dans le cas où un modèle incorrect serait supporté par quelques correspondances par hasard. Le paramètre β de la *non randomness* intervient donc directement dans cette condition et c'est ce qui explique son influence directe sur le nombre d'itérations. La contrainte de *maximality* est vérifiée lorsque le nombre d'itérations faites est supérieur à t^*. Lorsque ces deux conditions sont remplies (*non randomness* et *maximality*) l'algorithme s'arrête.

entrées: $\mathcal{E}$: un ensemble de correspondances ordonnées
β : la probabilité qu'une correspondance soit cohérente avec un modèle incorrect
T_N : le nombre d'itérations avant que $n = N$ (N étant le nombre total de correspondances)
begin
  $t \leftarrow 0$;
  $n \leftarrow 4$;
  $n^* \leftarrow N$;
  while *les critères de* non randomness *et de* maximality *ne sont pas vérifiés* **do**
    $t \leftarrow t + 1$;
    if $t = T_n$ *et* $n < N$ **then**
      $n \leftarrow n + 1$;
    end
    tirer un échantillon s dans $\mathcal{E}_n$;
    calculer la pose $\mathcal{P}$ à partir de s;
    if $\#I(\mathcal{P}) > \#I(\mathcal{P}^*)$ **then**
      $\mathcal{P}^* \leftarrow \mathcal{P}$;
    end
    mettre à jour n^* si possible (voir texte)
  end
  calculer la pose $\mathcal{P}_{\text{final}}$ à partir de $I_N(\mathcal{P}^*)$;
end
sorties : la pose de la caméra $\mathcal{P}_{\text{final}}$

Algorithm 2: L'algorithme PROSAC. $\mathcal{E}_n$ désigne l'ensemble des n premières correspondances.

L'algorithme complet est synthétisé dans la figure 2. L'utilisation de PROSAC permet d'accélérer significativement la convergence en exploitant un a priori sur la qualité des données. Cette méthode est cependant dédiée à la mise en correspondance d'images et ne peut pas être utilisée comme telle pour le calcul de pose.

4.3 Méthode proposée

Nous proposons une méthode dans l'esprit de PROSAC qui consiste à tirer les échantillons dans des sous-ensembles imbriqués de plus en plus grands. Notre méthode reprend

les étapes clés de PROSAC : associer un score à chaque correspondance, définir un ensemble dans lequel tirer les correspondances à chaque étape et définir un critère d'arrêt permettant de capitaliser sur la découverte rapide d'un échantillon correct. Les contributions proposées redéfinissent ces étapes afin qu'elles s'intègrent dans le contexte du calcul de pose.

4.3.1 Classement a priori des correspondances

La première étape de notre approche consiste à ordonner les correspondances en fonction d'un critère de qualité. L'approche choisie consiste à calculer un score pour chaque correspondance puis à les classer en fonction de ce score. Comme ce score va être utilisé pour biaiser le tirage vers des correspondances avec un haut score il est important que les correspondances associées à un haut score soient effectivement inliers, quitte à avoir quelques inliers associés à un faible score. En revanche on veut éviter d'associer un haut score à des outliers, ce qui biaiserait le tirage en faveur d'échantillons incorrects.

Cette section décrit deux types d'approches existantes pour classer les correspondances : une basée sur la mise en correspondance des descripteurs et une autre utilisant la concentration des correspondances correctes parmi un faible nombre de vues. Nous proposons un classement des correspondance inspiré de cette dernière catégorie d'approches.

Information a priori basée sur les descripteurs photométriques

Il est possible d'utiliser la distance entre les descripteurs utilisés pour la mise en correspondance afin d'estimer la qualité d'une correspondance. Ce type d'approche est celle utilisée dans PROSAC, adaptée à notre problème de mise en correspondance image-modèle.

Ce score est défini pour des correspondances image-modèle obtenues par mise en correspondance au plus proche voisin à l'aide de descripteurs. Soit f un point d'intérêt d'une image et p le point du modèle le plus proche de f. Le point p, en temps que point 3D du modèle SfM, est associé à une collection de descripteur SIFT. La distance $dist(f, p)$ entre f et p est la plus petite distance entre le descripteur associé à f et un descripteur associé à p. Le score de la correspondance entre f et p est définit comme $\frac{\text{dist}(f,p)}{\text{dist}(f,p')}$ où p' est le second plus proche voisin de f. Les descripteurs sont classés par rapports de distance croissants. Les correspondances correctes sont en effet généralement bien séparées du second plus proche voisin, on a donc $\text{dist}(f, p) << \text{dist}(f, p')$. Ce score est généralement utilisé avec un seuil, typiquement 0.8, pour écarter les correspondances incorrectes [Lowe, 1999], y compris dans notre application lors de la mise en correspondance image-modèle.

[Li et al., 2010] observent que lorsque le nombre de descripteurs dans le modèle augmente le score basé sur le rapport des distances au plus proche et au second plus proche voisin devient moins pertinent car $\text{dist}(f, p')$ a plus de chances d'être proche de $\text{dist}(f, p)$. Ils proposent d'utiliser le rapport $\frac{\text{dist}(f,p)}{\text{dist}(f',p)}$ où f' est le second plus proche point d'intérêt de p dans l'image.

Co-occurrence

Dans le cadre du calcul de pose il est possible d'utiliser des critères géométriques pour évaluer la qualité des correspondances.

[Li et al., 2012] proposent une fonction de classement qui porte sur les échantillons, dans le cadre de la localisation à partir de très grand modèles (plusieurs millions de points 3D). Leur critère est basé sur la co-occurrence des correspondances correctes dans les images de construction du modèle. Le score d'un groupe de quatre correspondances est le nombre d'images dans lesquelles les quatre points 3D associés apparaissent ensemble. En pratique, il est inenvisageable de classer l'ensemble des $\binom{N}{4}$ échantillons possibles, d'autant plus que N est très grand dans ce contexte de localisation à grande échelle. Le tirage des échantillons est donc modifié pour approcher ce score : la première correspondance est tirée uniformément et les suivantes sont tirées selon leur co-occurrence avec les précédentes. Comme le problème traité est celui de la localisation à très grande échelle, la co-occurrence dans plusieurs vues est un bon indicateur de qualité.

Dans notre contexte ce type d'information est également pertinente, et nous proposons un critère de ce type pour classer les correspondances. Nous avons observé dans le chapitre 3 que la plupart des descripteurs utilisés dans les correspondances correctes viennent d'un petit sous-ensemble de vues : généralement moins de 3 vues produisent 90% des correspondances correctes. Le taux d'inliers est également significativement plus important parmi les correspondances obtenues à partir de ces vues que dans l'ensemble des correspondances. La figure 4.2 illustre cette répartition des correspondances correctes. Nous définissons un score, que nous désignerons dans la suite comme co-occurrence, pour chaque correspondance de la façon suivante. Chaque correspondance est associée à la vue v dans laquelle le descripteur modèle a été extrait. Le score est le nombre total de correspondances associées à v. Toutes les correspondances associées à la même vue ont donc le même score. Les correspondances sont classées par score décroissant, et le score est raffiné en ordonnant les correspondances avec la même co-occurrence selon le rapport du plus proche sur le second plus proche décrit précédemment.

Ces critères permettent un meilleur classement des correspondances lorsqu'ils sont applicables, comme le montrent les résultats présentés dans la section 4.4.1. En effet, la photométrie est déjà prise en compte lors de l'étape de mise en correspondance image-modèle qui utilise les distances aux deux plus proches descripteurs. Les critère géométriques décrit dans cette section ajoutent une information qui n'a pas du tout été prise en compte dans la mise en correspondance, ce qui explique qu'ils classent mieux les correspondances.

4.3.2 Stratégie de tirage

L'algorithme RANSAC standard utilise un tirage des correspondances uniforme parmi l'ensemble de toutes les candidates. Dans notre méthode le tirage reste uniforme mais est fait dans un sous-ensemble des correspondances candidates. Définir la stratégie de tirage consiste donc à définir dans quel sous-ensemble les correspondances sont tirées à chaque itération de l'algorithme.

Soit $e_1, e_2, \ldots, e_N$ une suite de N correspondances candidates, ordonnées par une des

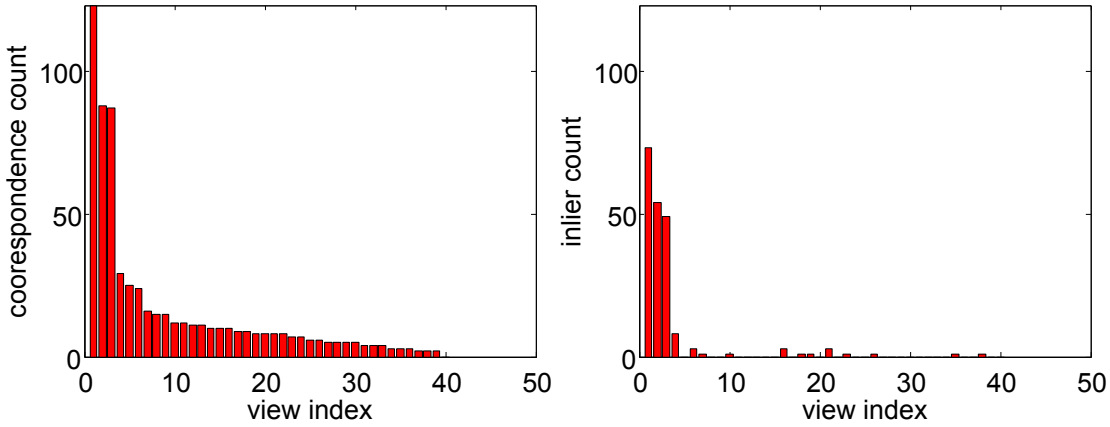


FIGURE 4.2 – Nombre de correspondances associées avec chaque point de vue après l'étape de mise en correspondance 2D/3D (gauche) et nombre d'inliers, par rapport à la vérité terrain, associés avec chaque point de vue (droite). Les points de vue ont été ordonnés par nombre de correspondances associées décroissante. On observe que les vues associées aux plus de correspondances sont également celles qui permettent d'obtenir les inliers.

fonctions précédemment décrites. Soit $\mathcal{E}_n = \{e_1, \ldots, e_n\}$ pour tout $1 \leq n \leq N$ les ensembles imbriqués des n premières correspondances. La stratégie de tirage consiste à tirer des échantillons dans $\mathcal{E}_4$, puis $\mathcal{E}_5$, etc. Pour un n donné, un échantillon consiste en la n-ième correspondance et trois autres tirées dans $\mathcal{E}_{n-1}$. Cela garantit que de nouveaux échantillons sont tirées à mesure que n augmente. Pour chaque n, t_n échantillons sont tirés dans $\mathcal{E}_n$. t_n est défini comme une proportion α de l'ensemble des échantillons possibles parmi n correspondances, c'est-à-dire :

$$t_n = \lceil \alpha \binom{n-1}{3} \rceil$$

où $\lceil x \rceil$ est le plus petit entier supérieur ou égal à x. Nous fixons expérimentalement α à 0.1.

Cette stratégie de tirage est essentiellement une reformulation de celle proposée dans PROSAC avec un paramétrage plus adapté à chaque expérience. Effectivement, dans chaque ensemble $\mathcal{E}_n$, PROSAC tire t_n^{PROSAC} échantillons :

$$\begin{aligned} t_n^{PROSAC} &= \lceil T_n - T_{n-1} \rceil \\ t_n^{PROSAC} &= \lceil \frac{T_N}{\binom{N}{4}} (\binom{n}{4} - \binom{n-1}{4}) \rceil \\ t_n^{PROSAC} &= \lceil \frac{T_N}{\binom{N}{4}} \binom{n-1}{3} \rceil \end{aligned}$$

Le paramètre T_N de PROSAC est donc le nombre d'échantillons tirés avant que n atteigne N alors que notre paramètre α est la proportion d'échantillons tirés avant que n atteigne N.

4.3.3 Arrêt anticipé

Comme le taux d'inliers varie fortement d'une expérience à une autre, de 5% à 60% dans nos expériences, un critère d'arrêt dynamique est nécessaire pour assurer la robustesse de l'algorithme. Nous utilisons un critère d'arrêt qui s'appuie sur une estimation en ligne du taux d'inliers dans le sous-ensemble $\mathcal{E}_n$.

Estimation du taux d'inliers sur $\mathcal{E}_n$

Il est possible d'estimer dynamiquement le taux d'inliers à partir de l'ensemble de consensus du meilleur modèle calculé, voire l'équation 4.2. Ce critère d'arrêt s'appuie sur une estimation du taux d'inliers faite sur l'ensemble des correspondances. Si on utilise ce critère, peu importe qu'on trouve un échantillon correct rapidement, le nombre d'itérations restera le même puisqu'il ne dépend que du taux d'inliers dans $\mathcal{E}$. En revanche $\mathcal{E}_n$ est justement un sous-ensemble de correspondances avec une haute probabilité d'être des inliers, il est donc raisonnable non seulement de tirer les échantillons dans cet ensemble mais aussi de les évaluer sur cet ensemble.

Nous proposons donc de restreindre l'évaluation du taux d'inliers à $\mathcal{E}_n$:

$$\tau_n = \frac{\#I_n(P)}{n} \tag{4.4}$$

où $I_n(P) = I(P) \cap \mathcal{E}_n$ est l'ensemble de consensus restreint à $\mathcal{E}_n$. Le nombre d'itérations est calculé comme précédemment. Comme les premières correspondances ont une probabilité d'être correctes élevée, τ_n est supérieur à τ et le nombre d'itérations requises sera plus faible qu'en utilisant le critère précédent.

Ce critère d'arrêt est le même que le critère de *maximality* définit dans PROSAC et se heurte à la même difficulté : comment estimer le taux d'inliers de façon fiable pour les faibles valeurs de n ?

Faibles valeurs de n

Nous avons maintenant définit un critère satisfaisant dans la mesure où la recherche de nouveaux échantillons peut effectivement s'arrêter plus tôt que dans RANSAC standard. Cependant, restreindre l'évaluation de τ_n à $\mathcal{E}_n$ pose un problème d'initialisation. En effet, ce critère est mal défini pour les faibles valeurs de n. Pour $n = 5$ par exemple, comme la pose est estimée à partir d'un échantillon de 4 correspondances il ne reste qu'une correspondance pour calculer un ensemble de consensus, et le taux estimé est donc soit 0 soit 1. Dans ce dernier cas l'algorithme terminerait immédiatement quelle que soit la probabilité p^* fixée, puisque si le taux d'inliers est 1 l'échantillon tiré est nécessairement correct.

Nous proposons d'estimer le taux d'inliers au moins sur les n_0 premières correspondances, c'est-à-dire $\mathcal{E}_{n_0}$, n_0 étant le nombre de correspondances associées à la vue ayant

produit le plus de correspondances. Le taux d'inliers est donc finalement calculé comme suit :

$$\tau_n = \begin{cases} \dfrac{\#I_{n_0}(P)}{n_0} \text{ si } n < n_0 \\ \dfrac{\#I_n(P)}{n} \text{ sinon} \end{cases}$$

Cette solution heuristique est justifiée par les observations faites dans la section 4.3.1. En effet, $\mathcal{E}_{n_0}$ est un sous ensemble de correspondances avec un taux d'inliers sensiblement supérieur au taux d'inliers parmi l'ensemble des données. Les résultats expérimentaux présentés dans la section 4.4.2 valident ce choix.

Dans PROSAC ce problème d'initialisation est également présent, et c'est la contrainte de *non randomness* qui permet de le résoudre. Cette contrainte impose effectivement une taille minimum pour accepter un ensemble de consensus. En pratique, comme le paramètre β associé à cette condition est difficile à fixer, voir section 4.2, leur condition revient essentiellement à imposer une taille fixe minimale sur l'ensemble de consensus.

4.3.4 Paramètres de la méthode

L'algorithme proposé est synthétisée dans la figure 3. Un des intérêts de notre méthode est la robustesse aux conditions expérimentales : le taux d'inliers n'a pas besoin d'être connu à l'avance et les autres paramètres peuvent être fixés a priori. Les paramètres de cet algorithme sont :

- la probabilité p^* de tirer au moins un échantillon correct, fixée à 0.95.
- le taux d'échantillons explorés α, fixé expérimentalement à 10%. Ce paramètre, à l'instar du paramètre équivalent T_N de PROSAC, ne semble affecter de façon significative le résultat des expériences. Cela est lié au fait que les correspondances sont bien classées par notre fonction de classement, on a donc une forte probabilité de tirer un échantillon correct tant que n ne croît pas trop rapidement.
- le seuil d'acceptation des inliers est fixé à l'erreur de reprojection des points dans le modèle SfM. Dans nos expériences cette erreur est de l'ordre de 10 pixels pour des images de taille 1600×1200 pixels.

4.4 Résultats

Cette section présente et compare des résultats de mise en correspondance et de calcul de pose obtenus en utilisant RANSAC standard, PROSAC et notre méthode. Le protocole expérimental est similaire à celui utilisé dans le chapitre 3 : un modèle SfM est construit à partir d'un ensemble d'images pour différentes scènes, illustrées dans la figure 4.3, puis on calcule une pose à partir d'une nouvelle image. Certaines expériences utilisent des modèles enrichis par simulation et sont notées *+sim*. L'évaluation porte sur le nombre d'itérations de RANSAC, le temps de calcul et la précision des poses obtenues.

entrées: $\mathcal{E}$: un ensemble de correspondances ordonnées
n_0 : le nombre de correspondances associées avec la vue la plus productive

begin
- $t \leftarrow 0$;
- $n \leftarrow m$;
- $t_{\text{stop}} \leftarrow t_{\max}$;
- **while** $t < t_{stop}$ **do**
  - $t \leftarrow t + 1$;
  - **if** $t > t_4 + ... + t_n$ **then**
    - $n \leftarrow n + 1$;
    - $\tau \leftarrow \dfrac{\#I_{\max(n,n_0)}(\mathcal{P})}{\max(n, n_0)}$;
    - $t_{\text{stop}} \leftarrow \dfrac{\log(1 - p_0)}{\log(1 - \tau^m)}$;
  - **end**
  - tirer un échantillon s dans $\mathcal{E}^3_{n-1} \times \{e_n\}$;
  - calculer la pose $\mathcal{P}$ à partir de s;
  - **if** $I_N(\mathcal{P}) > I_N(\mathcal{P}^*)$ **then**
    - $\mathcal{P}^* \leftarrow \mathcal{P}$;
    - $\tau \leftarrow \dfrac{\#I_{\max(n,n_0)}(\mathcal{P})}{\max(n, n_0)}$;
    - $t_{\text{stop}} \leftarrow \dfrac{\log(1 - p^*)}{\log(1 - \tau^m)}$;
  - **end**
- **end**
- calculer la pose $\mathcal{P}_{\text{final}}$ à partir de $I_N(\mathcal{P}^*)$;

end

sorties : la pose de la caméra $\mathcal{P}_{\text{final}}$

Algorithm 3: L'algorithme RANSAC proposé. Le paramètre d'arrêt t_{stop} est mis à jour lorsque n augmente ou qu'un ensemble de consensus plus grand est trouvé. L'algorithme stoppe dès que le nombre d'itérations courant t est supérieur à t_{stop}.



FIGURE 4.3 – Les scènes de tests utilisées, de gauche à droite et de haut en bas : livre (extraite de [Aanæs et al., 2012]), bière (extraite de [Jensen et al., 2014]), pot (extraite de [Jensen et al., 2014]) etCAB (extraite de [Cohen et al., 2012]).

4.4.1 Classement des correspondances

Nous présentons ici une évaluation des différentes fonctions de classement décrites dans la section 4.3.1. Comme notre méthode consiste à retreindre le tirage des échantillons à l'ensemble des n premières correspondances $\mathcal{E}_n$, et que la vitesse de convergence de RANSAC est liée au taux d'inliers, on souhaite que le taux d'inliers dans $\mathcal{E}_n$ soit élevé pour des faibles valeurs de n puis diminue à mesure que n augmente. Pour évaluer les fonctions de classement proposées on s'intéresse donc au taux d'inliers dans $\mathcal{E}_n$ fonction de n. Ce taux d'inliers est obtenu en utilisant la pose de la vérité terrain. Ces courbes 4.4 ne sont donc pas connues lors de l'exécution de l'algorithme. On peut noter que [Chum and Matas, 2005] utilisent une approximation de cette courbe pour déterminer leur critère d'arrêt, mais cette approximation utilise le modèle estimé par RANSAC.

La figure 4.4 montre que le classement selon la co-occurrence (courbe rouge) est toujours meilleur que les deux autres fonctions considérées. Le taux d'inliers parmi les n premières correspondances est supérieur pour cette fonction de classement. C'est en particulier visible pour les n_0 premières correspondances qui constituent l'ensemble sur lequel le taux d'inliers est évalué au début de l'algorithme. Dans le cas de l'expérience sur la base

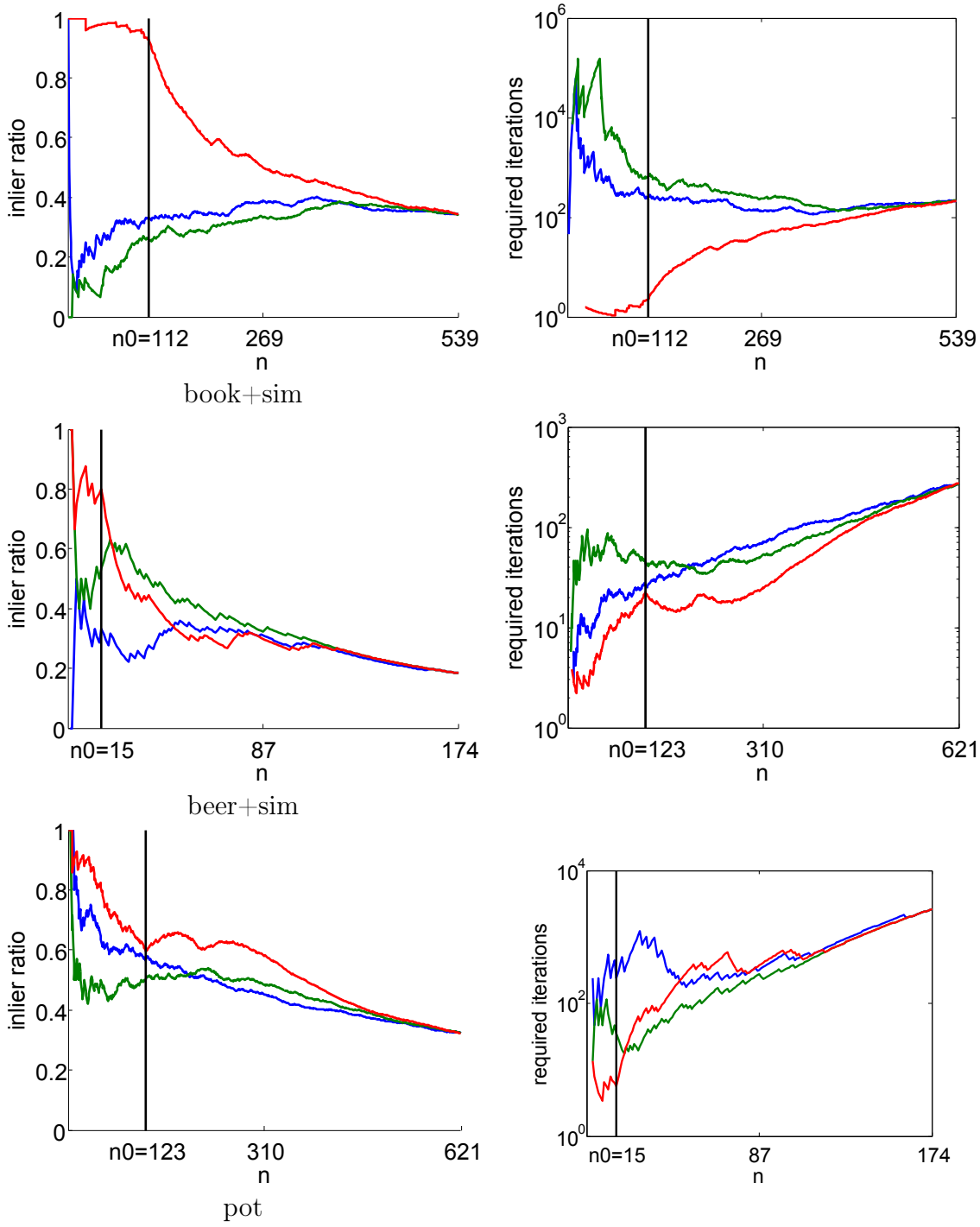


FIGURE 4.4 – Comparaison des trois fonctions de classement en utilisant le taux d'inliers de la vérité terrain : le rapport des distances au plus proche et au second plus proche voisins (bleu), le rapport inverse (vert) et le nombre de correspondances par vue (rouge). Les courbes de la première colonne représentent le taux d'inliers parmi les n premières correspondances. Les courbes de la seconde colonne montrent le nombre d'itérations requises pour obtenir une probabilité de convergence de 90%. La barre verticale indique le nombre n_0 de correspondances associées avec la vue ayant produit le plus de correspondances.

itérations	RANSAC	PROSAC	méthode proposée	vues utilisées
$book_1$+sim	234±36	413±15	6±0	1
$book_2$	85±3	73±3	5±1	1
$beer_1$+sim	340±100	361±0	22±1	2
$beer_2$	110±10	52±5	9±1	1
pot	2943±318	498±30	29±1	3
pot+sim	145±30	71±1	69±1	2
school	6867±581	625±193	103±0	3

TABLE 4.1 – Moyenne et écart-type du nombre d'itérations effectuées en utilisant notre fonction de classement basée sur la co-occurrence, sur 100 essais de calcul de pose. Pour RANSAC standard le critère d'arrêt donné par l'équation 4.2 est utilisé.

livre, avec simulations, on observe une augmentation significative du taux d'inliers dans $\mathcal{E}_{n_0}$, et par conséquent une réduction significative du nombre d'itérations nécessaires : 304 itérations sont nécessaires en utilisant le rapport des distances aux deux plus proches voisins, 98 sont nécessaires avec le rapport inverse et 6 avec la co-occurrence.

Dans les expériences, lorsqu'on utilise la co-occurrence, notre critère d'arrêt est souvent satisfait avant que n atteigne n_0. Dans ce cas, les échantillons ont uniquement été tirés parmi les correspondances associées à la vue associée au plus grand nombre de correspondances. Le nombre de vues utilisées augmente avec la complexité de la scène, une seule vue suffit pour la scène livre mais pas pour la scène CAB. Dans les expériences effectuées, les correspondances ont été tirées au plus parmi celles associées aux trois premières vues, voire la dernière colonne de la table 4.1. Cela indique que l'échantillonnage parmi les vues associées aux plus grand nombre de correspondances est raisonnable.

On observe également dans la figure 4.4 que le taux d'inliers diminue parmi les n_0 premières correspondances, lorsqu'on utilise la co-occurrence. Ordonner les correspondances associées à la même vue par le rapport des distances aux deux plus proches voisins est donc également justifié. Malgré la présence d'outliers au sein des n_0 premières correspondances le taux d'inliers parmi ces correspondances est le plus important lorsqu'on utilise la co-occurrence comme fonction de classement.

4.4.2 Temps de calcul

Dans cette section nous comparons la méthode proposée avec RANSAC standard et PROSAC, adapté à notre problème de calcul de pose. Notre méthode est paramétrée comme indiqué dans la section 4.3.4. Les paramètres de PROSAC T_N et β sont fixés à $T_N = 200,000$ et $\beta = 0.1$, comme suggéré par [Chum and Matas, 2005]. La valeur de β utilisée permet d'obtenir des résultats raisonnable avec PROSAC, c'est-à-dire que l'algorithme converge vers une pose proche de la pose correcte. La figure 4.3 illustre l'impact de β sur le nombre d'itérations. Les expériences portent sur le nombre d'itérations et le temps de calcul requis par chaque méthode pour converger vers une pose. Dans cette section nous discutons des expériences dans lesquelles les différentes méthodes convergent vers une pose proche de la vérité terrain.

temps (s)	RANSAC	PROSAC	méthode proposée
$book_1$+sim	1.37	1.62	0.02
$book_2$	0.34	0.30	0.02
$beer_1$+sim	2.33	2.40	0.14
$beer_2$	0.50	0.27	0.02
pot	15.67	1.94	0.04
school	25.48	2.4	0.38

TABLE 4.2 – Moyenne du temps de calcul d'une pose, évaluée sur 100 calculs de poses, en secondes.

beta	0.01	0.05	0.1	0.15
$book_2$	17 (echec)	41	73	73
$beer_2$	6 (echec)	23 (echec)	52	-
pot	12	40	498	-

TABLE 4.3 – Influence du paramètre β de PROSAC sur le nombre d'itérations. Dans les autres expériences β est fixé à 0.1. Lorsque le nombre d'itérations n'est pas indiqué, l'algorithme n'a pas convergé, aucun ensemble de consensus ne vérifiant la condition de *non randomness* paramétrée par β, voir section 4.2. Les expériences marquées *échec* indiquent que l'algorithme converge vers une pose incorrecte. Ces quelques exemples illustrent la difficulté du choix de β : la valeur de β pour laquelle les résultats sont les meilleurs varie d'une expérience à l'autre, sans qu'on puisse la fixer a priori.

Les tables 4.1 and 7.1 montrent respectivement le nombre d'itérations et les temps de calculs associés aux différentes méthodes. Les deux tables ne sont pas exactement proportionnelles car l'évaluation des ensembles de consensus dépend du nombre de données. Notre méthode converge jusqu'à 100 fois plus rapidement que RANSAC standard et jusqu'à 50 fois plus rapidement que PROSAC. L'écart-type du nombre d'itérations est également considérablement plus faible lorsque notre méthode est utilisée.

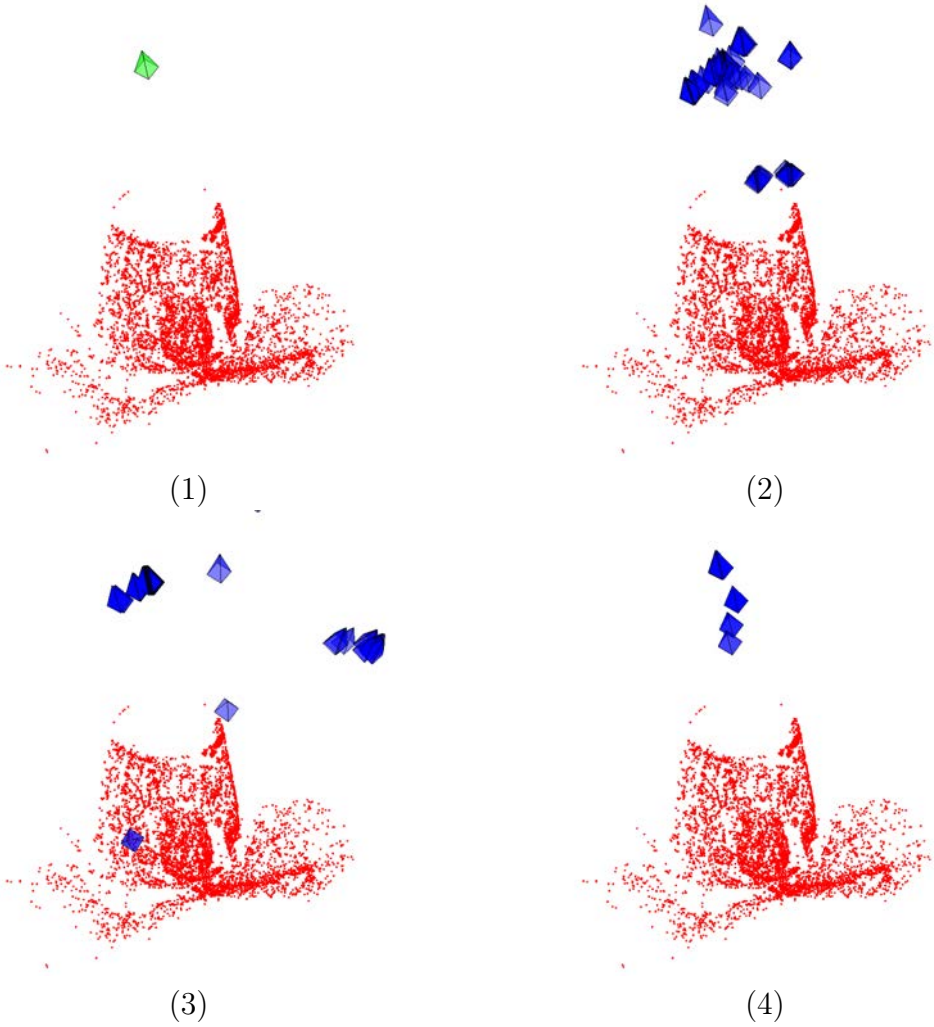
Les poses calculées en utilisant notre méthode sont proches de la vérité terrain, voire figure 4.5. Pour toutes les méthodes, la pose calculée varie d'une exécution à une autre à cause de la nature aléatoire de RANSAC et des variantes utilisées. Cependant, ces variations de la pose calculée sont largement amoindries avec notre méthode. En effet, notre processus de tirage tend à sélectionner systématiquement le même ensemble de consensus, et donc à calculer la même pose.

Dans l'expérience pot, sans synthèse de vues, figure 4.5, la mise en correspondances 2D/3D est particulièrement difficile à cause du fort changement de point vue. Dans cet expérience, le taux d'inliers est très faible (18%) mais même la version standard de RANSAC converge après un nombre suffisant d'itérations. Notre méthode converge considérablement plus rapidement dans ce cas car le classement des correspondances par notre méthode est particulièrement pertinent 4.4 et un échantillon correct est donc tiré presque dès les premières itérations.

L'expérience CAB présentée dans la figure 4.6 est difficile à cause d'une accumulation de difficultés telles que des motifs répétés, des spécularités des occlusions et de forts changements de point de vue. Cette expérience illustre un cas typique de changement de point de vue entre les vues de construction et l'image requête : les vues de construction viennent de la cour du bâtiment alors que la requête a été prise à partir de la rue. Le taux d'inliers est de 14% dans cette expérience, et notre méthode converge plus rapidement que RANSAC standard et PROSAC.

La méthode proposée est heuristique et dépend d'hypothèses fortes qui sont parfois mises en défaut. En particulier, le fait que $\mathcal{E}_{n_0}$ soit un ensemble de correspondances avec un très haut taux d'inliers est crucial. Les conditions d'échecs de notre méthode sont généralement les conditions dans lesquelles notre fonction de classement se trompe, c'est-à-dire attribue un haut score à des outliers. Cependant les erreurs surviennent généralement en raison de motifs répétés ou d'un taux d'inliers extrêmement faible ($< 10\%$), qui sont des conditions dans lesquelles toutes les méthodes échouent. Dans toutes les expériences, utiliser les heuristiques proposées dans notre méthode ne dégrade pas les performances par rapport à RANSAC standard, que ce soit en terme de vitesse de convergence ou de précision de la pose.

RANSAC standard est toujours plus lent que les deux autres méthodes puisqu'il n'utilise aucun a priori sur la qualité des correspondances et nécessite donc beaucoup plus d'itérations pour tirer un échantillon correct. Notre méthode est plus rapide que PROSAC pour deux raisons. Premièrement, la fonction de classement proposée est particulièrement adaptée au problème du calcul de pose. Deuxièmement, le critère d'arrêt que nous définissons stoppe l'algorithme très rapidement après avoir trouvé un échantillon correct, ce qui est idéal pour une approche type RANSAC. Cet arrêt rapide est possible car on a un ensemble de correspondances de haute qualité sur lequel s'appuyer, $\mathcal{E}_{n_0}$.



(1) (2)

(3) (4)

FIGURE 4.5 – 100 poses calculées sur la séquence pot avec RANSAC standard (1), PROSAC (2) et notre méthode (3). La figure (1) montre la vérité terrain. Dans cette expérience le taux d'inliers est de 18%. La variabilité des poses calculées est significativement plus faible avec notre méthode qu'avec les deux autres.



(1) (2) (3)

FIGURE 4.6 – Première rangée : 100 poses calculées sur la séquence CAB avec RANSAC standard (1), PROSAC (2) et notre méthode (3). Deuxième rangée : reprojection des contours de la façade. La variabilité de la position poses calculées est significativement plus faible avec notre méthode, bien que les trois méthodes aient une faible erreur de reprojection.

4.5 Conclusion

Ce chapitre présente une méthode de type RANSAC adaptée à notre problème, et permet de calculer rapidement une pose précise malgré un taux d'inliers variable et potentiellement très faible. La méthode proposée utilise des considérations géométriques sur les correspondances, à savoir que les correspondances correctes sont généralement issues d'un petit sous-ensemble de vues. Cette information géométrique complète la mise en correspondance purement photométrique et permet de donner une estimation fiable de la qualité d'une correspondance. Un échantillonnage progressif permet de rapidement trouver un modèle correct et la recherche s'arrête rapidement après avoir trouvé ce modèle. Les expériences montrent que notre méthode converge considérablement plus rapidement que l'algorithme RANSAC standard ainsi que PROSAC, tout en augmentant la précision des poses calculées.

Dans le contexte du calcul de pose à partir d'un modèle enrichi par synthèse de vue, comme expliqué dans le chapitre 3, cette méthode permet de calculer rapidement une pose malgré l'ajout d'un grand nombre de descripteurs au modèle. On peut donc utiliser la synthèse de vue pour améliorer considérablement le calcul de pose à partir de points de vue difficiles comme illustré dans le chapitre 3 sans pour autant augmenter le temps de calcul de la pose.

Chapitre 5

Synthèse de vues efficace

Le chapitre 3 présente un procédé de synthèse de vues et montre que celui-ci améliore significativement le calcul de pose. Cette méthode permet de calculer des poses à partir de points de vue éloignés des vues connues, dans des situations où un modèle SfM contenant seulement les descripteurs des vues de construction ne suffit pas. De façon générale, elle permet de calculer des poses plus précises. Cependant, cette approche ne fonctionne que si on exclut la possibilité de placer des caméras dans la scène et demande beaucoup de temps de calcul. Elle consiste à ajouter des points de vue virtuels sur une sphère autour de la scène, puis à produire un patch synthétique pour chaque paire composée d'un point du modèle et d'un point de vue virtuel. Un descripteur est extrait de chaque patch et ajouté à ceux du modèle. Plusieurs problèmes se posent avec cette approche : le positionnement des caméras n'a réellement de sens que si la scène est de petite taille, rien ne garantit que les points de vue virtuels ajoutés couvrent correctement les points de vues possibles, la synthèse est coûteuse en temps de calcul et enfin le nombre de descripteurs ajoutés rend la recherche de correspondances difficile. Le chapitre 4 montre que le problème de mise en correspondance peut être résolu par une approche de type échantillonnage progressif. Ce chapitre propose une nouvelle approche pour la synthèse de vues qui résout les problèmes restants, à savoir le positionnement des caméras virtuelles pour une scène quelconque et le temps de calcul. L'objectif est de présenter une nouvelle approche de synthèse de vue dont la mise en œuvre en pratique soit réaliste. Cette approche consiste en trois étapes : la segmentation du modèle en patchs plans, le positionnement des caméras par rapport à ces patchs et la génération des vues de synthèses.

L'approche présentée dans le chapitre 3 est limitée car elle considère le modèle uniquement comme un ensemble de points. Dans ce chapitre le modèle est vu comme un assemblage de patchs plans, en faisant l'hypothèse que la scène est plane par morceaux. Cette hypothèse est vérifiée dans la plupart des environnements, en particulier lorsque des objets et structures d'origine humaine sont présents. Considérer le modèle comme un assemblage de patchs offre plusieurs avantages. Premièrement, il est plus facile de raisonner sur la couverture en terme de points de vue pour un plan que pour une structure quelconque. Deuxièmement, comme nous synthétisons des vues en utilisant la transformation induite par un plan, il est possible de grouper tous les points d'un même patch lors de l'étape de synthèse. La figure 5.1 donne quelques exemples de scènes segmentées. La segmentation du modèle en patchs est expliquée dans la section 5.1.

Les caméras virtuelles sont positionnées par rapport aux patchs segmentés. Pour chaque patch on positionne un ensemble de caméras telles qu'elles couvrent les points de vue non présents parmi les caméras de reconstruction. Ce procédé produit donc un grand nombre de caméras virtuelles, de l'ordre de 1000 dans certaines de nos expériences, mais chaque caméra ne produit qu'une vue d'un patch spécifique. En comparaison, dans la section 3.2.2 du chapitre 3, 25 caméras étaient placées mais chacune produit un patch synthétique pour chaque point du modèle. La section 5.2.2 présente le positionnement des caméras virtuelles.

La synthèse des vues virtuelles est faite en groupant tous les points d'un même patch. Grouper les points de cette façon permet un gain de temps considérable par rapport à la méthode présentée dans la section 3.2.2 du chapitre 3 où un petit patch synthétique (100×100 pixels) est produit pour chaque point du modèle et pour chaque caméra virtuelle. Calculer tous ces patchs est long et redondant : si des points 3D sont proches les uns des autres les patchs associés se recouvrent partiellement, on synthétise alors plusieurs fois la même image. Grouper les points d'un même patch pour faire la synthèse élimine ces calculs redondants. En revanche considérer des patchs entraîne des problèmes de visibilité : comment s'assurer que le patch est visible à partir des points de vue utilisés lors de la synthèse ? La section 5.3 présente une solution aux problèmes de visibilité et la section 5.4 détaille la synthèse des vues.

Les résultats présentés dans la section 5.5 montrent que faire la synthèse de vue de cette façon demande à peu près le même temps de calcul que la construction du modèle, de l'ordre de quelques minutes. Ces temps sont à comparer aux heures de calculs nécessaires lorsque la synthèse est faite point par point. De plus ce procédé de synthèse à l'avantage d'être applicable à toute scène plane par morceaux. De plus, les résultats montrent que le calcul de pose est plus rapide lorsqu'on utilise la synthèse car les correspondances 2D/3D sont globalement plus précises. L'approche de la synthèse de vue telle qu'elle est développée dans ce chapitre a fait l'objet d'une publication [Rolin et al., 2016].

5.1 Segmentation du modèle en patchs plans

Identifier des patchs plans dans le modèle est utile pour deux raisons. Premièrement, il est possible de définir une répartition optimale des caméras virtuelles autour d'un patch plan, ce qui est l'objet de la section 5.2.2. Deuxièmement, notre méthode de synthèse utilise la transformation de l'image d'un plan induite par un changement de point de vue, comme expliqué en 3. Extraire des plans permet donc d'identifier quelles parties de la scène sont susceptibles d'apporter de nouveaux descripteurs utiles pour le mise en correspondance. Un grand nombres d'approches ont été proposées pour segmenter un nuage de points en plans. [Holz et al., 2011] proposent de calculer une normale en chaque point et de faire du clustering sur ces normales pour identifier les plans. Cette approche est conçue pour des nuages de points dont la densité est relativement uniforme, typiquement obtenus par un scanner laser, ce qui n'est pas le cas des modèles SfM que nous utilisons. [Schnabel et al., 2007] proposent d'utiliser une approche type RANSAC pour extraire des primitives géométriques dans un nuage de points. Cette approche ne nécessite pas que les points soient répartis uniformément sur la surface reconstruite pour

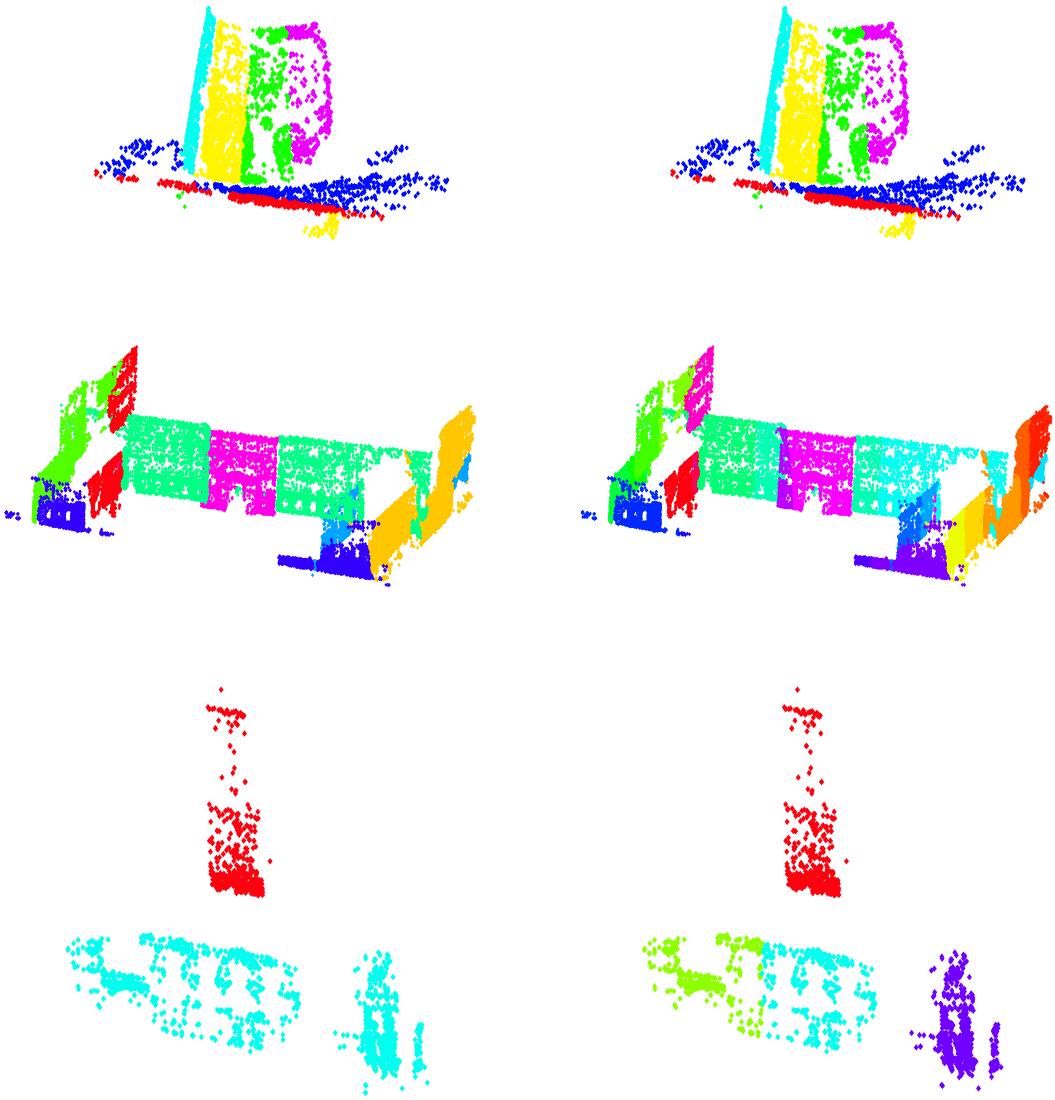


FIGURE 5.1 – Exemples de scènes segmentées par notre approche. La première colonne montre la segmentation par RANSAC et la deuxième colonne montre la segmentation additionnelle en patchs plans. La première rangée est une scène-objet et il n'est donc pas nécessaire de redécouper les plans après l'étape RANSAC. Les deux autres scènes sont des bâtiments et les plans sont découpés en patchs plus petits.

fonctionner et permet d'extraire plusieurs modèles. C'est ce type d'approche que nous mettons en œuvre pour segmenter le modèle. On note qu'un problème commun dans les méthodes de recherche de surfaces planes est d'identifier correctement les contours de cette surface, comme discuté par [Fleishman et al., 2005] par exemple. Dans notre contexte segmenter correctement le modèle à proximité des contours n'est pas utile car la synthèse de vue est généralement incorrecte à ces endroits. Nous pouvons donc nous contenter de détecter lorsqu'un point est proche d'un contour et l'éliminer, sans chercher à déterminer à quel plan il appartient.

Les plans sont extraits à partir du modèle SfM, en utilisant uniquement les coordonnées 3D. L'objectif est d'associer chaque point à un patch plan s'il se trouve sur une surface plane, ou à l'éliminer dans le cas contraire. La segmentation de la scène consiste en plusieurs étapes : une normale est calculée pour chaque point du modèle, les plans sont extraits par une approche type RANSAC puis ces plans sont découpés en patchs.

Pour chaque point p du modèle, on estime une normale à la surface de la scène au niveau de ce point. Cette normale est estimée comme dans [Hoppe et al., 1992] par analyse en composante principale des points dans un voisinage de p, ce voisinage étant une boule de rayon d centrée sur p. La normale est la composante associée à la plus faible valeur propre. Dans nos expériences d est fixé comme le double de la distance moyenne d'un point à son plus proche voisin. Comme le voisinage défini est une boule de rayon d centrée sur p, il est possible que p n'ait pas de voisins, ou très peu. Dans ce cas l'estimation de la normale est imprécise, voire impossible si p a moins de deux voisins. On considère que p est un point isolé s'il a moins de 5 voisins et on le retire alors du modèle.

Les plans de la scènes sont extraits par applications successives de RANSAC : un plan est extrait par RANSAC, les point associés sont retirés du modèle, puis on continue à chercher de nouveaux plans parmi les point restants. Un point appartient au plan si deux conditions sont remplies : la distance euclidienne entre le point et le plan est inférieure à t_d et l'angle entre la normale associée au point et celle du plan est inférieure à t_n. t_d est fixé comme la distance moyenne d'un point à son plus proche voisin. T_n est fixé à 10°. La recherche de nouveaux plans s'arrête lorsque le plan trouvé par RANSAC représente moins de 5% des points initiaux. Ce seuil de 5% fonctionne bien dans toutes nos expériences, dans lesquelles le nombre total de plans est de l'ordre de 10. Il serait nécessaire de le fixer de manière plus fine pour des scènes avec un grand nombre de plans, typiquement une scène urbaine avec plusieurs bâtiments reconstruits. Comme l'estimation des normales par analyse en composantes principales est imprécise à proximité des bords de la scène, les points proches des bords ne sont pas inclus dans les patchs segmentés, leurs normales ne s'alignant pas avec celle du plan. Nous utilisons cela à notre avantage : retirer ces points du modèle est bénéfique car la synthèse est généralement incorrecte à proximité des contours.

Les plans ainsi segmentés peuvent être trop grands pour être directement utilisés pour synthétiser de nouveaux patchs. En effet, on veut positionner des caméras virtuelles autour de chaque plan à une distance raisonnable. En l'absence d'indications supplémentaires liées à l'application, une distance raisonnable est celle entre les points du plan et les caméras qui les ont reconstruit. En effet, c'est à partir de ces caméras que les descripteurs utiles à la construction du modèle ont été extraits. Par ailleurs, si le le plan est très étendu la transformation des images par homographie est plus facilement sujette à des erreurs

d'échantillonnage. Si on veut contrôler la distance entre les caméras virtuelles et les points auxquels elles sont associées il faut redécouper les plans segmentés en patchs. Les plans extraits sont donc découpés en patchs plus petits de la façon suivante. Les patchs sont délimités par une grille dont les directions sont obtenues par ACP sur les points du plan et dont les cellules sont des carrés de côté c. Le paramètre c est choisi comme étant la distance moyenne entre les points du plan et les caméras qui les ont reconstruits. Ce paramètre n'est pas critique pour la méthode, il s'agit juste de donner une taille de patch qui soit cohérente avec une distance caméra-modèle typique de telle sorte que des descripteurs utiles puissent être extraits. L'algorithme de segmentation est synthétisé dans la figure 4 et les résultats sont illustrés dans 5.1.

entrées: $P3D$: un ensemble de points 3D
sorties : les patchs segmentés
begin
  estimer une normale pour chaque point de $P3D$;
  déterminer un ensemble de points $\mathcal{M}$ formant un plan par RANSAC à partir des points $P3D$;
  retirer les points de $\mathcal{M}$ de $P3D$;
  while $\#\mathcal{M} > 0.05 * \#P3D$ **do**
    déterminer un ensemble de points $\mathcal{M}$ formant un plan par RANSAC à partir des points $P3D$;
    retirer les points de $\mathcal{M}$ de $P3D$;
  end
  for *chaque plan segmenté* **do**
    $C \leftarrow$ l'ensemble des caméras utilisée pour reconstruire le plan;
    $c \leftarrow$ la distance moyenne des caméras de C aux points qu'elles ont reconstruit;
    découper p en patchs selon une grille de côté c;
  end
end

Algorithm 4: Pseudo-code pour la segmentation du nuage de points en patchs plans. On considère qu'un point appartient à un plan si la distance euclidienne entre les deux et l'angle entre leurs normales (estimée par ACP pour le point) sont tous les deux inférieurs à des seuils fixés. Le seuil sur la distance est fixé comme la distance moyenne d'un point à son plus proche voisin. Le seuil sur l'angle est fixé à 10°.

5.2 Positionnement des points de vue virtuels

Cette section explique comment les caméras virtuelles sont positionnées par rapport aux patchs segmentés. Les caméras ne sont pas positionnées par rapport à la scène complète, pour chaque patch segmenté on définit un ensemble de caméras virtuelles qui lui sont propres. Par rapport à ce qui est fait dans la section 3.2.2 chapitre 3, plus de caméras virtuelles sont ajoutées, mais chacune est utilisée pour synthétiser une unique vue.

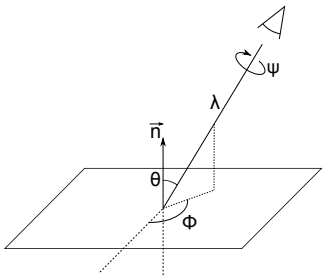


FIGURE 5.2 – Représentation géométrique des variables utilisées dans l'équation (3.3) avec $t = 1/\cos(\theta)$.

Le placement des caméras utilise le *transition tilt* introduit par [Yu and Morel, 2011]. Nous rappelons comment celui-ci est défini et expliquons pourquoi c'est une métrique pertinente dans notre contexte. Nous détaillons ensuite le placement des caméras virtuelles.

5.2.1 Transition tilt

Cette section donne la définition du *transition tilt* introduit par [Yu and Morel, 2011] que nous utilisons comme métrique pour évaluer la distance entre deux points de vue. Le *transition tilt* est défini entre deux caméras affines observant un plan $\mathcal{P}$. Bien que la section 3.3 du chapitre 3 montre qu'utiliser un modèle perspectif de caméra pour la synthèse permet d'obtenir des poses plus précises, nous utilisons le *transition tilt* en tant qu'heuristique pour placer les caméras virtuelles. On rappelle que la transformation affine entre la vue fronto-parallèle de $\mathcal{P}$ et son image par une caméra affine (λ, ψ, t, ϕ) peut se décomposer de la façon suivante :

$$A = \lambda \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \tag{5.1}$$

la figure 5.2 montre la représentation géométrique des paramètres de cette décomposition. Cette décomposition est valable pour toute transformation affine et donc pour la transformation induite par $\mathcal{P}$ entre deux caméras affines $(\lambda_1, \psi_1, t_1, \phi_1)$ et $(\lambda_2, \psi_2, t_2, \phi_2)$. On appelle A_1 et A_2 les transformations affines entre la vue fronto-parallèle et les images de $\mathcal{P}$ par les deux caméras. La transformation affine induite entre les deux caméras est donc $A_1 A_2^{-1}$ et se décompose de la même façon que dans l'équation 5.1. t est alors appelé *transition tilt* et dépend à la fois du changement de longitude $\phi_2 - \phi_1$ et des tilts t_1 et t_2 des deux caméras, comme expliqué par [Yu and Morel, 2011]. Comme t dépend également de l'écartement par rapport à la normale, il traduit mieux les changements d'apparence qu'une simple différence angulaire entre directions de vue. Expérimentalement, on constate que la mise en correspondance de deux vues est possible si et seulement si t est plus petit qu'un certain seuil, de l'ordre de $\sqrt{2}$. On peut donc estimer la couverture de quelques patchs en terme de points de vue : une direction de vue est couverte si son transition tilt avec la caméra la plus proche est inférieur à $\sqrt{2}$. Une représentation possible de cette

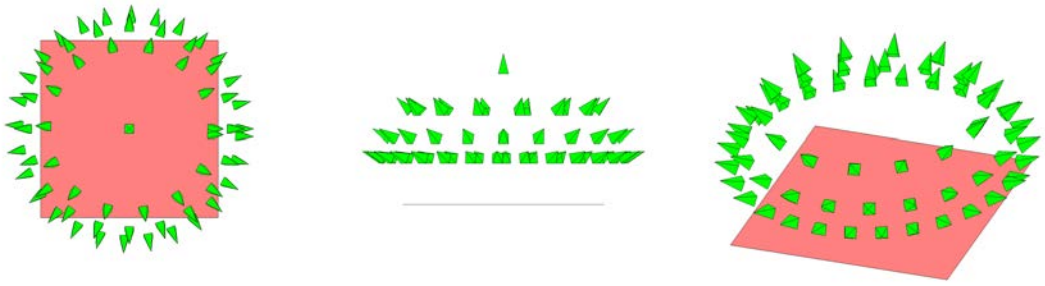


FIGURE 5.3 – L'échantillonnage des caméras virtuelles par rapport à un patch, défini à partir du *transition tilt*. L'échantillonnage est plus dense pour des points de vue éloignés de la vue fronto-parallèle car les changement d'apparence sont alors plus prononcés. l'écartement maximal par rapport à la normale (clairement visible sur la vue du milieu) est de l'ordre de 70°. Au delà l'image du plan est trop déformée pour pouvoir en extraire des descripteurs utiles.

couverture est proposée dans la figure 5.4.

5.2.2 Échantillonnage autour d'un patch

Dans l'équation 5.1, si $t = 1$ et $\phi = 0$, c'est à dire si les deux caméras ont la même latitude et longitude, alors A est une similitude. Si on considère les descripteurs SIFT invariants par similitude alors I_1 et I_2 produisent les mêmes descripteurs. Ceci justifie un échantillonnage des caméras selon leur latitude et longitude. Dans [Morel and Yu, 2009], la synthèse des vues virtuelles est faite avec un modèle affine et les caméras sont donc réparties sur un demi-hémisphère. En effet, les vues symétriques ($\theta_1 = \theta_2$ et $\phi_1 = \phi_2 + \pi$) sont identiques pour des caméras affines. On observe le même phénomène pour des caméras projectives si on synthétise juste un petit voisinage autour d'un point du modèle, ce qui est discuté dans la section 3.3.4 du chapitre 3. Comme nous utilisons le modèle de transformation homographique et que les patchs synthétisés ne sont pas réduit à des petits voisinages autour des points du modèle, l'échantillonnage que nous proposons couvre tout un hémisphère. Pour chaque patch, les caméras sont échantillonnées pour (t, ϕ) tels que $t = 2^{m/2}$ ($m \in \{1, 2, 3\}$) et $\phi = n\,72°/t$, de telle sorte que ϕ décrive $[0, 360°]$. Une caméra virtuelle est ajoutée uniquement s'il n'existe pas une caméra réelle à un transition tilt inférieur à $\sqrt{2}$, pour éviter de synthétiser des vues redondantes avec les observations existantes.

L'échantillonnage porte uniquement sur t et ϕ mais comme nous utilisons un modèle sténopé pour les caméras virtuelles, contrairement à [Morel and Yu, 2009], nous devons également fixer la position de ces caméras. Les caméras virtuelles sont positionnées de telle sorte que leur axe optique passe par le centre de gravité du patch. La distance entre le centre optique d'une caméra et le centre du patch est fixée comme $2m$, m étant la taille du patchs extrait (m est aussi la moyenne des distance entre les points du patch et les caméras qui l'ont reconstruit). La figure 5.3 illustre cet échantillonnage et la figure 5

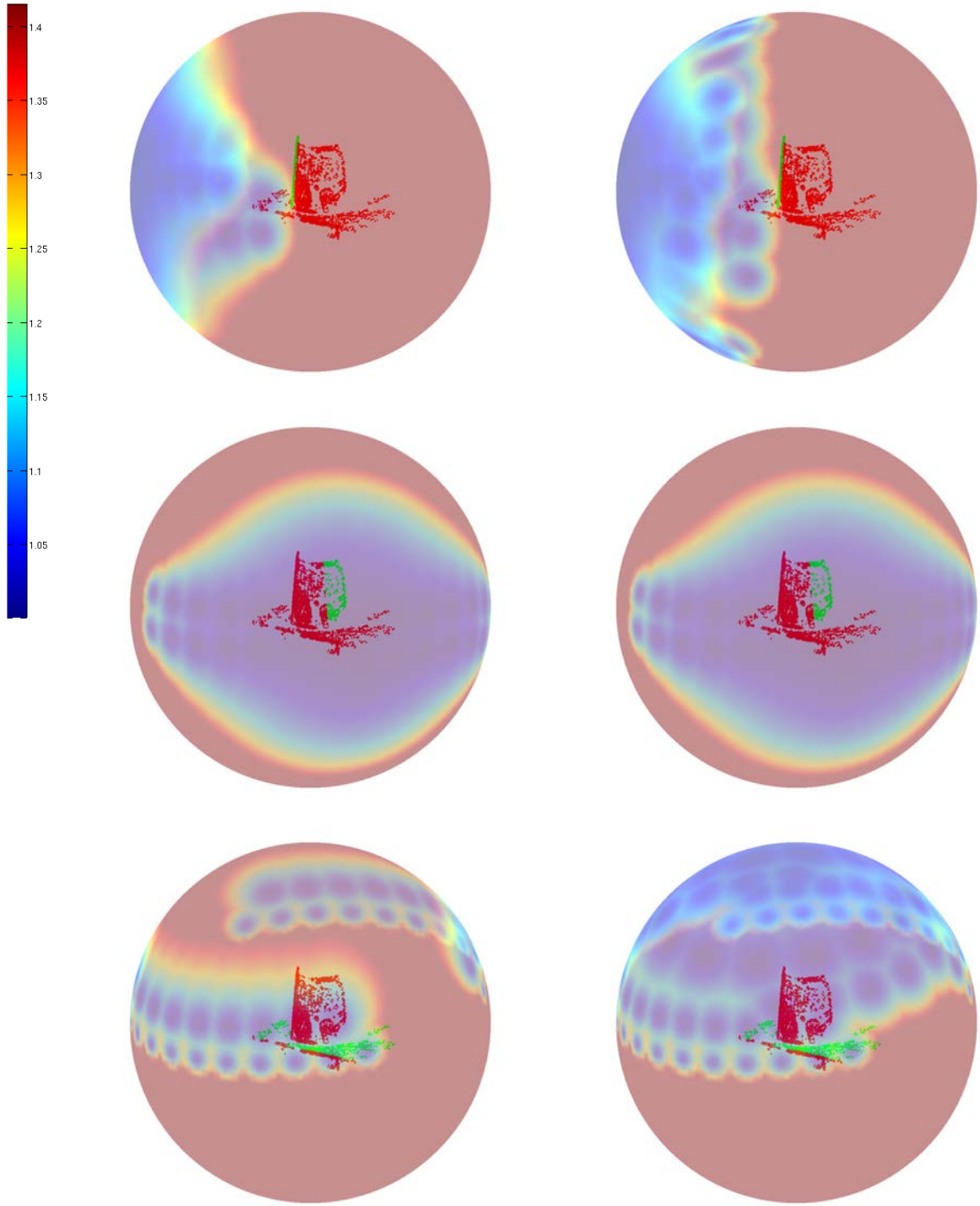


FIGURE 5.4 – Illustration de la couverture des points de vues pour certains patchs segmentés (points verts), sans caméras virtuelles (colonne de gauche) et avec caméras virtuelles (colonne de droite). La couleur de la sphère dépend du *transition tilt* avec le plus proche point de vue, rouge signifiant que la direction de vue est trop éloignée pour que la mise en correspondance soit possible ($t > \sqrt{2}$).

récapitule la méthode sous forme de pseudo-code.

entrées: $\mathcal{P}$ un patch segmenté
C_r : les caméras réelles ayant reconstruit $\mathcal{P}$
sorties : C_v : les caméras virtuelles associées à $\mathcal{P}$
begin
  $C_v \leftarrow \{\}$;
  for $m \in \{1, 2, 3\}$ **do**
    $t \leftarrow 2^{m/2}$;
    $\phi \leftarrow 0$;
    while $\phi < 360$ **do**
      $\phi \leftarrow \phi + 72/t$;
      ajouter une caméra (t, ϕ) à la liste C_v;
    end
  end
  for $(c_r, c_v) \in C_r \times C_v$ **do**
    if transition tilt$(c_r, c_v) < \sqrt{2}$ **then**
      retirer la caméra virtuelle c_v de C_v;
    end
  end
end

Algorithm 5: Pseudo-code pour l'ajout de caméra virtuelles à un patch. En pratique les paramètres (t, ϕ) des caméras sont pré calculés une fois pour l'ensemble de tous les patchs.

Pour un patch donné, au plus 66 caméras virtuelles sont définies, qui sont les caméras de la répartition optimale définie dans la section 5.2.2 et illustrée par la figure 5.3. La méthode que nous proposons peut donc définir un très grand nombre de caméras virtuelles, de l'ordre de 66 fois le nombre de patchs. Par exemple, pour la scène pot 222 caméras ont été ajoutées et pour la scène CAB 1032 caméras ont été ajoutées. Cependant, comme chacune de ces caméras est associée à un patch spécifique, chacune de ces caméras ne produit qu'une unique vue synthétique. Le nombre de simulation à faire est donc bien plus faible que dans le chapitre 3 où le nombre de simulations était de l'ordre du nombre de caméras multiplié par le nombre de points dans le modèle. L'échantillonnage des caméras dans une scène est illustré dans la figure 5.5.

5.3 Visibilité dans un nuage de points

Comme notre approche utilise des patchs et non pas des points isolés, des problèmes de visibilité supplémentaires apparaissent : à partir d'un point de vue donné un point est visible ou non alors qu'un patch est généralement partiellement visible. Cette section détaille comment ces problèmes sont pris en compte dans notre approche.

La visibilité dans un nuage de points est difficile à définir, puisque les points sont toujours visibles à moins qu'ils ne soient exactement alignés avec la position d'observation.

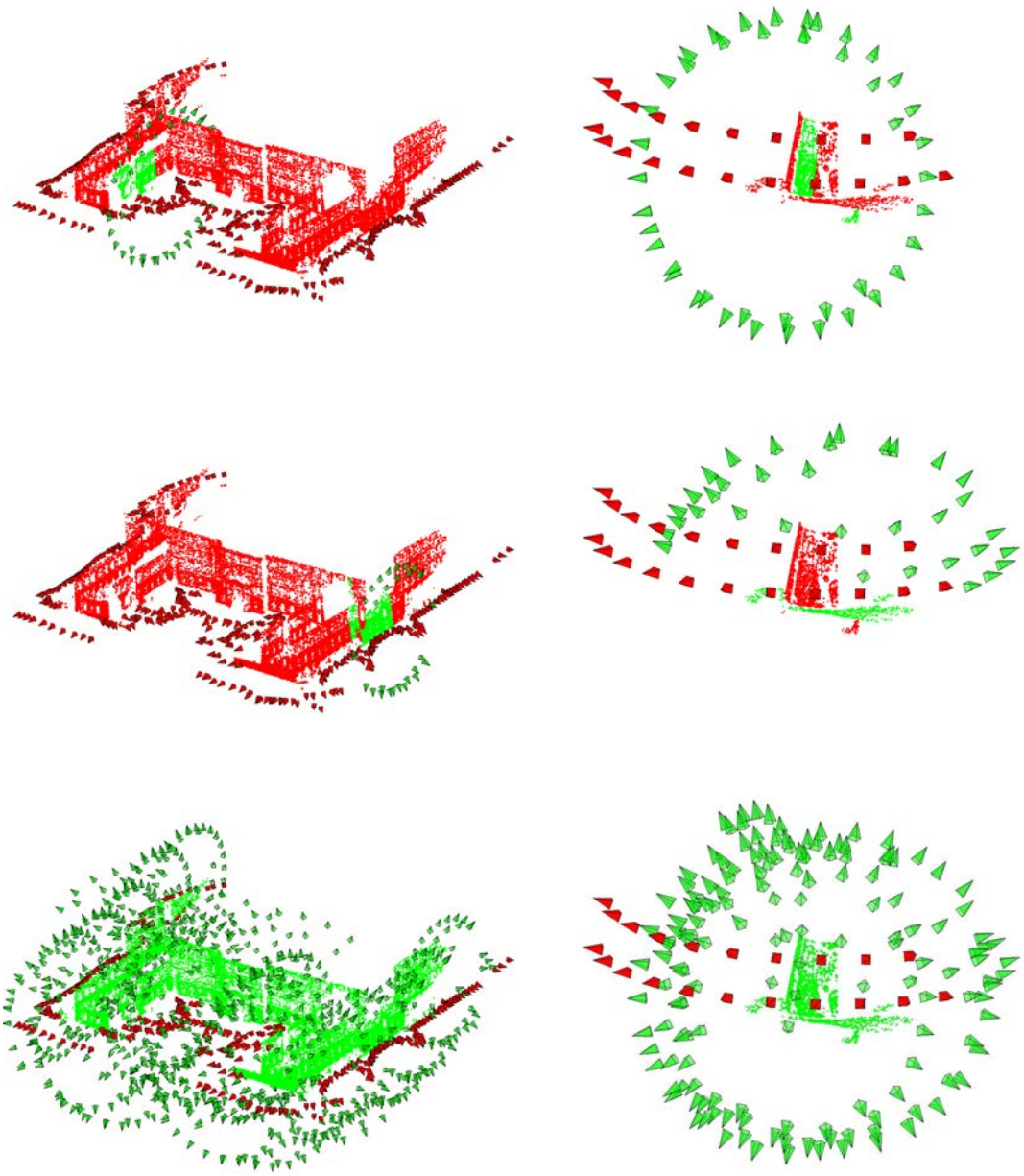


FIGURE 5.5 – Positionnement des caméras virtuelles (en vert) par rapport à des patchs (points verts). Les caméras rouges sont celles utilisées pour construire le modèle. Sur la dernière ligne toutes les caméras ajoutées à la scène sont représentées.

Lorsqu'on parle de visibilité dans un nuage de point, on suppose généralement que les points sont sur une surface sous-jacente et qu'un point est visible lorsque la surface sur laquelle il se trouve l'est. Dans le cas d'un modèle SfM par exemple les points se trouvent sur la surface reconstruite.

Il y a deux types de point de vue à partir desquels nous devons calculer la visibilité : les points de vue réels utilisés pour construire le modèle et les points de vue virtuels ajoutés. En effet, on veut synthétiser les patchs à partir de vues dans lesquelles ils sont visibles. Pour ces points de vue la visibilité peut être déduite à partir des données du modèle SfM, comme expliqué dans la section 5.3.1. De même, on veut s'assurer que les points du patchs sont visibles à partir des caméras virtuelles définies. La visibilité à partir des points de vue virtuels est expliquée dans la section 5.3.2 et utilise la méthode proposée par [Katz et al., 2007].

5.3.1 Visibilité à partir des caméras de construction

Cette section décrit la sélection des vues réelles utilisées pour générer les vues de synthèse. On s'intéresse à un patch composé d'un ensemble de point P qui ont été reconstruits à partir d'un ensemble de caméras C. Pour que les vues synthétisées soient utiles, il faut s'assurer que les points du patch segmenté sont visibles dans la vue réelle utilisée pour la synthèse. Rien ne garantit que tous les points du patch sont visibles dans une unique vue, il est donc parfois nécessaire de sélectionner plusieurs vues à transformer pour un patch et une caméra virtuelle donnés. Nous décrivons ici la méthode proposée pour sélectionner ces vues.

L'information de visibilité des points à partir des caméras réelles est accessible grâce au modèle SfM : un point est visible à partir d'une vue réelle si un descripteur extrait dans cette vue est associé à ce point. Seuls les points remplissant cette condition sont considérés comme visibles. Pour qu'un point soit visible à partir d'une caméra il est donc nécessaire mais pas suffisant que le point apparaisse dans l'image associée. Nous conservons ce critère sous cette forme car on veut non seulement que le points soit visible dans l'image, mais surtout qu'il soit possible d'extraire un descripteurs au niveau de ce point. À chaque caméra $c \in C$ est associé l'ensemble des points $P_c \subset P$ qui sont visibles depuis c.

Le problème consiste à trouver le plus petit sous-ensemble de caméra $C^* \subset C$ tel que tout point est visible depuis au moins une caméra de C^*. C^* est déterminé par une approche gloutonne. On commence par sélectionner la caméra c_1 à partir de laquelle le plus de points sont visibles, $c_1 = \text{argmax}_{c \in C}\{\#P_c\}$. On sélectionne ensuite la caméra c_2 à partir de laquelle le plus de points sont visibles parmi ceux qui ne sont pas visibles à partir de c_1, $c_2 = \text{argmax}_{c \in C}\{\#(P_c \setminus P_{c_1})\}$. La figure 5.6 montre pour quelques patchs les caméras sélectionnées et les points associés dans la même couleur.

5.3.2 Visibilité à partir des caméras virtuelles

Dans le cas des caméras virtuelles, les seules informations utilisables pour déterminer la visibilité des points du modèles sont leur position dans l'espace et la position des caméras virtuelles. Le problème est donc de décider quels points d'un nuage de points sont visibles à partir d'une position donnée. Dans ce contexte, un point est dit visible si la surface

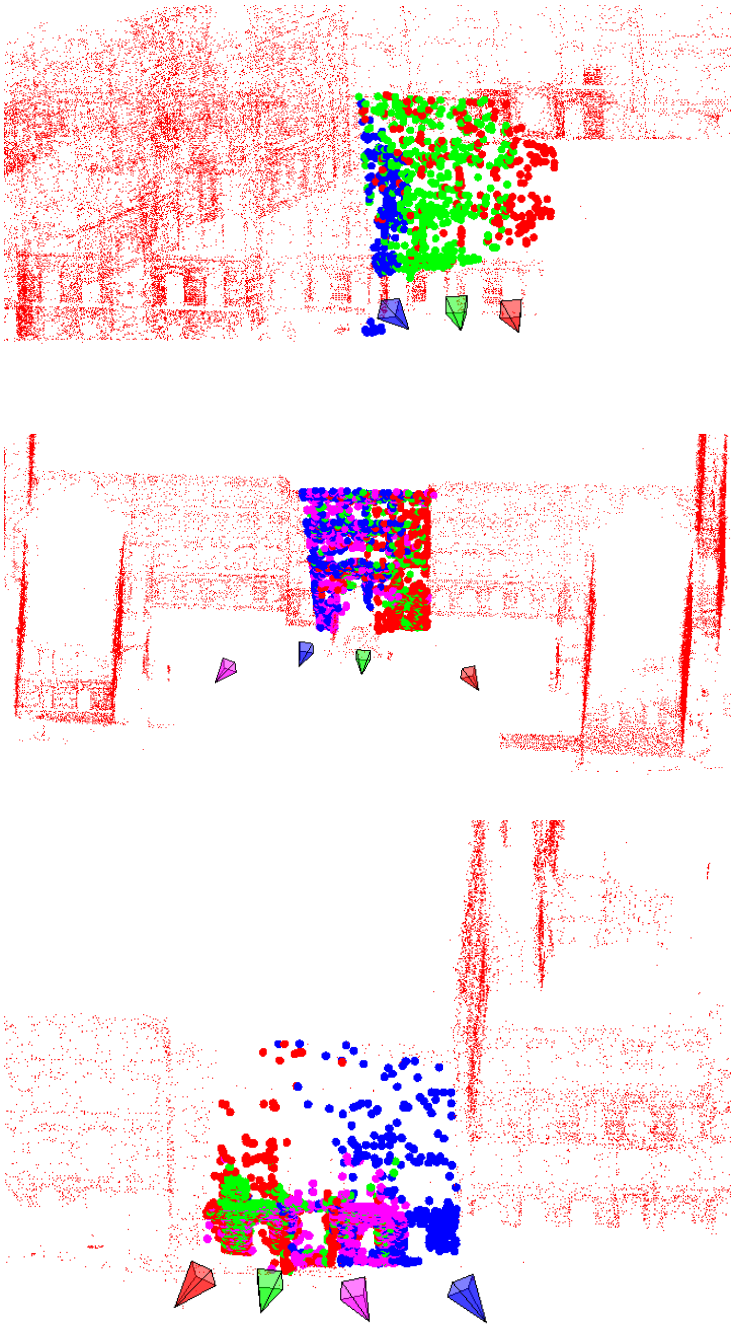


FIGURE 5.6 – Les caméras réelles utilisées pour synthétiser des vues, pour un patch donné. Les points sont représentés avec la couleur de la caméra à partir de laquelle ils sont visibles, avec la notion de visibilité définie dans la section 5.3.1.

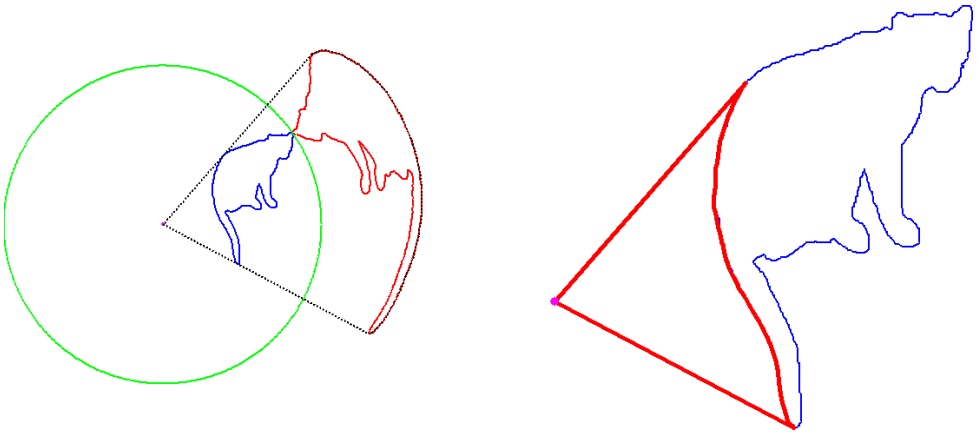


FIGURE 5.7 – L'élimination des points cachés, illustré dans le cas 2D. La figure de gauche illustre la transformation des points : les points bleus sont ceux du modèle, Les points rouges sont les points transformés. L 'enveloppe convexe est représentée en noir. La figure de droite montre l'antécédent de l'enveloppe convexe calculée, qui correspond aux points visibles (image extraite de [Katz et al., 2007]).

sur laquelle il se trouve est visible. La plupart des méthodes existantes [Amenta and Kil, 2004, Fleishman et al., 2005] pour traiter la visibilité dans un nuage de point consistent à calculer un maillage du modèle puis à déterminer la visibilité à partir de ce maillage. Reconstruire un maillage est un problème complexe, en particulier lorsque les points ne sont pas reconstruit précisément sur la surface, comme c'est le cas pour nos modèles SfM. Reconstruire une surface lisse devient alors difficile. De plus calculer le maillage est coûteux en temps de calcul. [Katz et al., 2007] proposent une méthode efficace et simple à mettre en œuvre pour déterminer quels points sont visibles dans ce contexte, sous la seule hypothèse que les points reconstruits se trouvent sur une surface. Cette hypothèse est raisonnable dans la mesure où nos points sont issus d'une reconstruction SfM, mais l'erreur de positionnement des points par rapport à la surface est sensiblement plus importante dans notre contexte que dans les travaux présentés par [Katz et al., 2007]. Cette section présente leur méthode et explique comment nous l'utilisons dans notre contexte.

La méthode proposé consiste en deux étape : les points du modèle sont transformés, puis on extrait l'enveloppe convexe de l'ensemble des points transformés. Soit E l'ensemble des points du modèle et C la position de la caméra. En supposant que le repère est centré en C, chaque point $p \in E$ est transformé par la transformation suivante :

$$f(p) = p + 2(R - ||p||)\frac{p}{||p||}$$

La figure 5.7 illustre cette transformation pour un exemple dans le plan. On considère l'ensemble des points transformés auxquels on ajoute la position C de la caméra $E' = \{f(p) : p \in E\}$. Les points de l'enveloppe convexe de $E' \cup \{C\}$ sont les points du modèle visibles à partir de C. Les auteurs montrent que cette méthode calcule effectivement les points visibles si les points se trouvent sur une surface sous-jacente et que leur densité est suffisamment élevée. La densité de points nécessaire dépend de la courbure de la surface.

Dans notre contexte, la densité des points n'est pas uniforme et ces résultats ne sont plus vérifiés, mais les expériences montrent qu'elle permet d'obtenir une approximation raisonable des points visibles, comme illustré dans la figure 5.8.

La méthode est simple à mettre en œuvre et est particulièrement efficace en temps de calcul. La complexité est en $O(n \log n)$ où n est le nombre de points dans le modèle ($O(n)$ pour la transformation, $O(n \log n)$ pour calculer l'enveloppe convexe). En pratique le temps de calcul utilisé pour cette étape est négligeable par rapport à celui utilisé pour l'ensemble de la synthèse.

L'approche utilise un unique paramètre R qui est le rayon de la sphère utilisée pour transformer les points. Intuitivement, plus R est grand, plus les points transformés sont proches de la sphère de rayon $2R$ et plus le nombre de points dans l'enveloppe convexe est important. Donc plus R est élevé plus on considère de points comme étant visibles. Dans notre contexte la densité des points reconstruits peut varier d'une partie de la scène à une autre. De plus les points reconstruits ne se trouvent pas précisément sur une surface en raison des erreurs de reconstruction, un point peut donc être masqué par un autre point de la même surface. Nous proposons de fixer empiriquement R en utilisant l'observation suivante : les points d'un patch plan, considéré indépendamment du reste de la scène, sont tous visibles à partir de n'importe quelle position. On considère l'ensemble P des points du modèles appartenant à un patch segmenté comme expliqué dans la section 5.1 et la position C d'une caméra virtuelle associée à ce patch. Tous les points du patch sont visibles depuis la position de la caméra puisque les seul points considérés se trouvent sur un unique plan. Cependant, en raison des erreurs de reconstruction, la méthode proposée par [Katz et al., 2007] peut détecter certains points comme étant masqués par d'autres. Nous choisissons R comme la plus petite valeur telle que tous les points sont considérés visibles à partir de C. De cette façon nous évitons les faux négatifs dus à la répartition des points autour du support du plan. Ce critère ne permet pas de déterminer finement la visibilité pour tous les points, ce qui est peu réaliste vu la précision des points reconstruits, mais permet d'éliminer les occultations les plus importantes. L'influence de$R$ est illustrée par la figure 5.8.

5.4 Transformation des vues

Cette section détaille la génération des patchs synthétiques. Comme des patchs plans ont été extraits et que chaque caméra virtuelle est affectée à un patch en particulier, il est possible de transformer les images des patchs complets, et pas uniquement un voisinage des points comme dans la section 3.2.2 du chapitre 3.

On considère un patch segmenté $\mathcal{P}$. On détermine quelles caméras réelles C_r doivent être utilisées pour la simulation, comme expliqué dans la section 5.3.1. On calcule les vues synthétiques à ajouter C_s. Un patch synthétique est généré pour chaque élément de $C_r \times C_s$.

Pour une paire caméra réelle / caméra synthétique $(c_r, c_s) \in C_r \times C_s$ le patch est généré de la façon suivante. On calcule l'homographie H induite par $\mathcal{P}$ entre c_r et c_s de la même façon que dans la section 3.1.1. On extrait de la vue associée à c_r la zone de l'image où P apparaît, déterminée comme étant la boite englobante des projections des

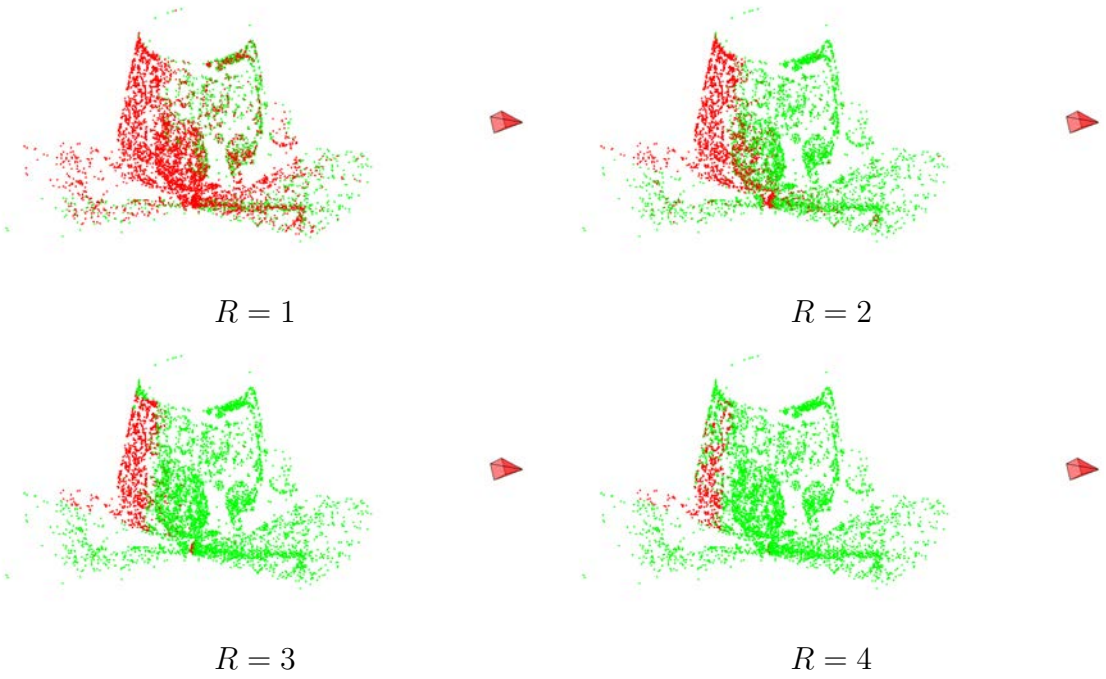


FIGURE 5.8 – Points visibles (vert) et points masqués (rouges) à partir de la caméra représentée, pour différente valeurs du paramètre R. Cette figure illustre les résultats de l'algorithme avec une valeur de R fixée pour l'ensemble du modèle, bien que dans notre implémentation une valeur différente soit utilisée pour chaque patch, comme expliqué dans le texte (les valeurs de R utilisées dans notre algorithme pour les différents patchs sont : $3.2, 3.4, 2.9, 3.1, 2.5, 2.1$). Pour $R = 1$ on voit bien que des points directement observés par la caméras ne sont pas considérés visibles, ce qui est dû à la répartition des points autour de la surface reconstruite. À l'inverse, pour $R = 4$ des points masqués par le pot sont marqués visibles.

points du patch dans l'image. Le patch synthétique est la transformation de cette image extraite par H.

L'algorithme correspondance est donné dans la figure 6 et l'ensemble du procédé de synthèse appliqué au modèle complet est résumé dans la figure 7.

entrées: $\mathcal{P}$: un patch
C_v : un ensemble de caméras virtuelles
sorties : $\mathcal{P}$ auquel on a ajouté de nouveaux descripteurs
begin
  Déterminer C_r l'ensemble des caméras réelles à utiliser pour la synthèse;
  for $c_v \in C_v$ **do**
    déterminer l'ensemble $\mathcal{P}^*$ des points de $\mathcal{P}$ visibles depuis c_v;
    for $c_r \in C_r$ **do**
      $I_r \leftarrow$ l'image de la scène par c_r;
      $I_r^* \leftarrow$ la région de I_r contenant les projections des points de $\mathcal{P}^*$;
      $H \leftarrow$ l'homographie induite par $\mathcal{P}$ entre c_r et c_v;
      $I_v \leftarrow HI_r^*$;
      ajouter aux points de $\mathcal{P}^*$ des descripteurs extraits de I_v;
    end
  end
end

Algorithm 6: Pseudo-code de la synthèse de vue pour un ensemble de caméras virtuelles C_v associées à un patch $\mathcal{P}$. Le calcul de l'homographie et l'ajout des descripteurs sont faits de la même façon que dans le chapitre 3.

entrées: un modèle SfM
soties : le modèle enrichi avec de nouveaux descripteurs
begin
  Segmenter le modèle en patchs plans (voir algorithme 4);
  for *chaque patch* $\mathcal{P}$ **do**
    calculer un ensemble de caméras virtuelles $C_\mathcal{P}$ (voir algorithme 5);
    calculer les nouveaux descripteurs pour le patch $\mathcal{P}$ et les caméras $C_\mathcal{P}$ (voir algorithme 6);
  end
end

Algorithm 7: Pseudo-code de la synthèse de vues pour le modèle complet.

5.5 Résultats

Cette section présente des résultats expérimentaux pour les temps de calculs de la synthèse de vue et du calcul de pose, avec différentes scènes. Nous nous concentrons ici sur les temps de calculs car c'est l'une des deux contributions principales de cette méthode

	poster	book	pot	tower	CAB
SfM	6min	11min	15min	5min	18min
synthèse	3min	6min	10min	4min	9min

TABLE 5.1 – Temps de calcul pour la construction du modèle et la synthèse de vues.

	poster	book	pot	tower	CAB
	modèle SfM				
nombre de descripteurs	47643	225207	32568	7774	324360
mise en correspondance	2.53s	3.15s	2.65s	1.48s	7.55s
calcul de pose	35.20	15.64s	10.17s	35.16s	8.29s
total	37.73	18.79	12.82	36.64	15.84
	modèle SfM + synthèse				
nombre de descripteurs	664848	887216	134484	85949	1523298
mise en correspondance	5.51s	4.60s	4.38s	3.72s	13.76s
calcul de pose	0.06s	0.80s	0.44s	0.38s	1.30s
total	5.57	5.40	4.82	4.10	15.06

TABLE 5.2 – Temps de calcul pour le calcul de pose. Le temps de calcul des poses est séparé en temps de recherche des correspondances 2D/3D et temps de calcul pour RANSAC-PnP.

de synthèse par rapport à celle présentée dans le chapitre 3, l'autre étant la possibilité d'appliquer la méthode à tout type de scène. Le chapitre 6 présente des résultats complets sur l'apport de la synthèse de vues en termes de précision des poses calculées.

On rappelle que pour une application typique, la synthèse de vue est une étape faite hors ligne, à la suite de la reconstruction du modèle. Pour cette raison, nous comparons le temps de calcul utilisé pour la synthèse avec celui utilisé pour reconstruire le modèle. Dans toutes les expériences, le temps de calcul utilisé pour la synthèse est plus court que celui utilisé pour la reconstruction du modèle, ce qui est raisonnable pour notre application. Pour comparaison, la méthode proposée dans le chapitre 3 nécessitait plusieurs heures pour calculer l'ensemble des nouveaux descripteurs à ajouter. Ces temps de calcul sont récapitulés dans la table 5.1.

Les expériences montrent également qu'utiliser la synthèse de vue réduit le temps de calcul pour le calcul de pose, bien qu'un grand nombre de descripteurs soient ajoutés au modèle. Plus précisément, la recherche de correspondances 2D/3D est plus longue mais le calcul de la pose par RANSAC-PnP est plus rapide. L'étape de recherche de correspondances demande plus de temps puisque la synthèse de vues augmente le nombre de descripteurs dans le modèle. Cette augmentation du temps de calcul est cependant limitée par l'utilisation d'un arbre KD pour faire cette recherche. En revanche, les correspondances sont de meilleure qualité grâce aux descripteurs ajoutés au modèle. Ce gain de temps dépend de la distance entre la caméra test et les caméras de reconstruction, c'est-à-dire de l'utilité des descripteurs ajoutés. On constate cependant que dans toutes les expériences le gain de temps sur l'étape de calcul de pose suffit à compenser le coût

additionnel lors de l'étape de recherche de correspondances. Ces temps de calcul sont donnés dans la table 5.2.

Ces expériences montrent que le coût global de la simulation est essentiellement nul : une étape supplémentaire hors ligne et le temps de calcul de pose reste similaire voire plus court. En échange de ce faible coût en temps de calcul la synthèse de vue permet de calculer des poses dans des situations complexes où un modèle SfM seul ne suffit pas.

5.6 Conclusion

Ce chapitre décrit un procédé de synthèse de vue complet reprenant les principes exposés dans le chapitre 3 mais de façon moins naïve. Deux contributions majeures sont apportées.

Premièrement, le procédé de synthèse dans son ensemble est applicable à tout type de scène sous l'hypothèse qu'il existe des parties planes. Cette hypothèse est généralement vérifiée, typiquement pour les scène d'intérieur ou en milieu urbain. Dans les approches existantes, par exemple [Irschara et al., 2009], [Torii et al., 2015] ou bien celle que nous proposons dans le chapitre 3, c'est le positionnement des caméras virtuelles qui est spécifique à un type de scène. La segmentation de la scène en patchs plans permet de simplifier le problème de placement des caméras virtuelles et de définir des positions raisonnables.

Deuxièmement, le temps de calcul nécessaire pour la synthèse est raisonnable, de l'ordre du temps nécessaire pour reconstruire le modèle. L'efficacité de l'approche est due à plusieurs facteurs. Les transformations d'images sont regroupées pour ne pas simuler des patchs se recouvrant partiellement. De plus seules des images virtuelles d'apparence sensiblement différente des vues réelles sont produites. Enfin, des contraintes de visibilités sont utilisées pour ne pas considérer les parties masquées de la scène lors de la synthèse.

En conclusion, les contributions apportées par cette nouvelles méthode de synthèse permettent de passer d'une méthode *proof of concept* fonctionnant sur des petits exemples à une méthode pouvant être mise en œuvre dans une situation réaliste.

Le chapitre 6 présente une série d'expériences qui mettent en évidence l'apport significatif de la synthèse de vues dans différents scénarios.

Chapitre 6

Résultats de calculs de pose dans des environnements variés

Ce chapitre rassemble les résultats expérimentaux obtenus au cours de la thèse. L'objectif de ces expériences est de montrer que l'approche de synthèse proposée permet de calculer des poses dans des cas de figure où le modèle SfM seul ne suffit pas et de façon général améliore la précision des poses de caméras calculées. Les expériences mettent en également en évidence l'impact de certains phénomènes remarquables, tels que des motifs répétés par exemple.

6.1 Protocole expérimental

Le protocole expérimental est le suivant. On commence par reconstruire un nuage de points 3D par SfM à partir d'une collection d'images. Un second modèle est obtenu en ajoutant des descripteurs par synthèse. Dans ces expériences, lorsque la synthèse de vue est utilisée, c'est la méthode décrite dans le chapitre 5 qui est utilisée. Nous considérons ces deux modèles dans les expériences : le modèle SfM et le modèle enrichi par synthèse de vues. Ces deux modèles se composent des mêmes points 3D, mais dans le second plus de descripteurs sont associés aux points du modèle. On suppose dans nos expériences que les les poses des points de vue utilisés pour construire le modèle sont correctement estimées lors de la reconstruction.

Les expériences consistent à calculer un ensemble de poses à partir d'un modèle et d'une image donnée. Pour chaque expérience discutée dans cette section, le nombre de poses calculées est de 100. On rappelle que les poses sont obtenues par RANSAC-PnP, ce qui explique la variabilité d'un calcul de pose à un autre. Nous utilisons dans cette section la version standard de RANSAC, avec estimation en ligne du taux d'inliers, comme expliqué dans la section 4.1 du chapitre 4. Pour cette raison, les poses ne sont pas aussi dispersées que dans certaines expériences présentées dans le chapitre 3 où RANSAC était stoppé après un nombre fixé d'itérations.

Pour calculer la pose les paramètres intrinsèques de l'appareil sont nécessaires. Pour certaines séquences d'images, livre, beer et pot, les paramètres sont donnés par les auteurs de la base d'images. Pour les séquences poster et bureau, qui sont des séquences per-

sonnelles, les paramètres de l'appareil ont été obtenus à l'aide de la *Calibration Toolbox* de MATLAB. Enfin, pour les séquences tour, piazza et CAB les paramètres intrinsèques utilisés sont ceux estimés lors de la reconstruction du modèle.

Les résultats sont montrés sous deux formes : une représentation 3D des poses calculées par rapport au nuage de point reconstruit et la reprojection de contours sélectionnés à la main. Ces contours sont obtenus en deux étapes. On calcule dans le repère de la scène les coordonnées des coins du contour. Pour ce faire les coins sont identifiés à la main dans les images de construction et leur position est calculée par triangulation. Cette triangulation est possible car les positions des caméras de construction sont connues via le modèle SfM. Nous projetons ensuite ces contours 3D dans l'image requête à partir des différentes poses calculées.

La figure 6.1 montre des images extraites des différentes séquences utilisées. La vérité terrain sur la pose de la caméra est disponible pour certaines de ces séquences d'images, à savoir livre, beer et pot. Dans les autres expériences, lorsque la vérité terrain n'est pas disponible, nous pouvons estimer une pose avec un très grand nombre d'itérations dans l'étape RANSAC-PnP, et en vérifiant visuellement que la pose obtenue est cohérente. Cette pose donne un point de comparaison pour calculer des erreurs de reprojection de points 3D dans les images, elle est utilisée dans ce seul but et n'intervient à aucun autre moment dans les calculs. Nous utilisons ces poses pour estimer le taux d'inliers parmi les correspondances image-modèles calculées dans chaque expérience. Une correspondance, composée d'un point 3D et d'un point 2D, est un inlier si le point 3D se projette à moins de 10 pixels du point 2D. Ce seuil de reprojection est le même que celui utilisé dans l'étape RANSAC-PnP.

Les expériences ont été réalisées sur un ordinateur portable équipé d'un processeur Intel CORE I7 et d'une carte graphique NVIDIA Quadro K610M, cette dernière étant uniquement utilisée lors du calcul des descripteurs SIFT. Dans nos expériences le logiciel VisualSfM [Wu, 2011] est utilisé pour la reconstruction du modèle. Le code utilisé pour la synthèse de vue et le calcul de la pose est écrit en MATLAB à l'exception de deux parties : le calcul des descripteurs SIFT est fait à l'aide de l'implémentation SiftGPU [Wu, 2007] et la recherche de correspondances utilise les arbres KD de la librairie ANN. Ces deux parties sont implémentées en C++.

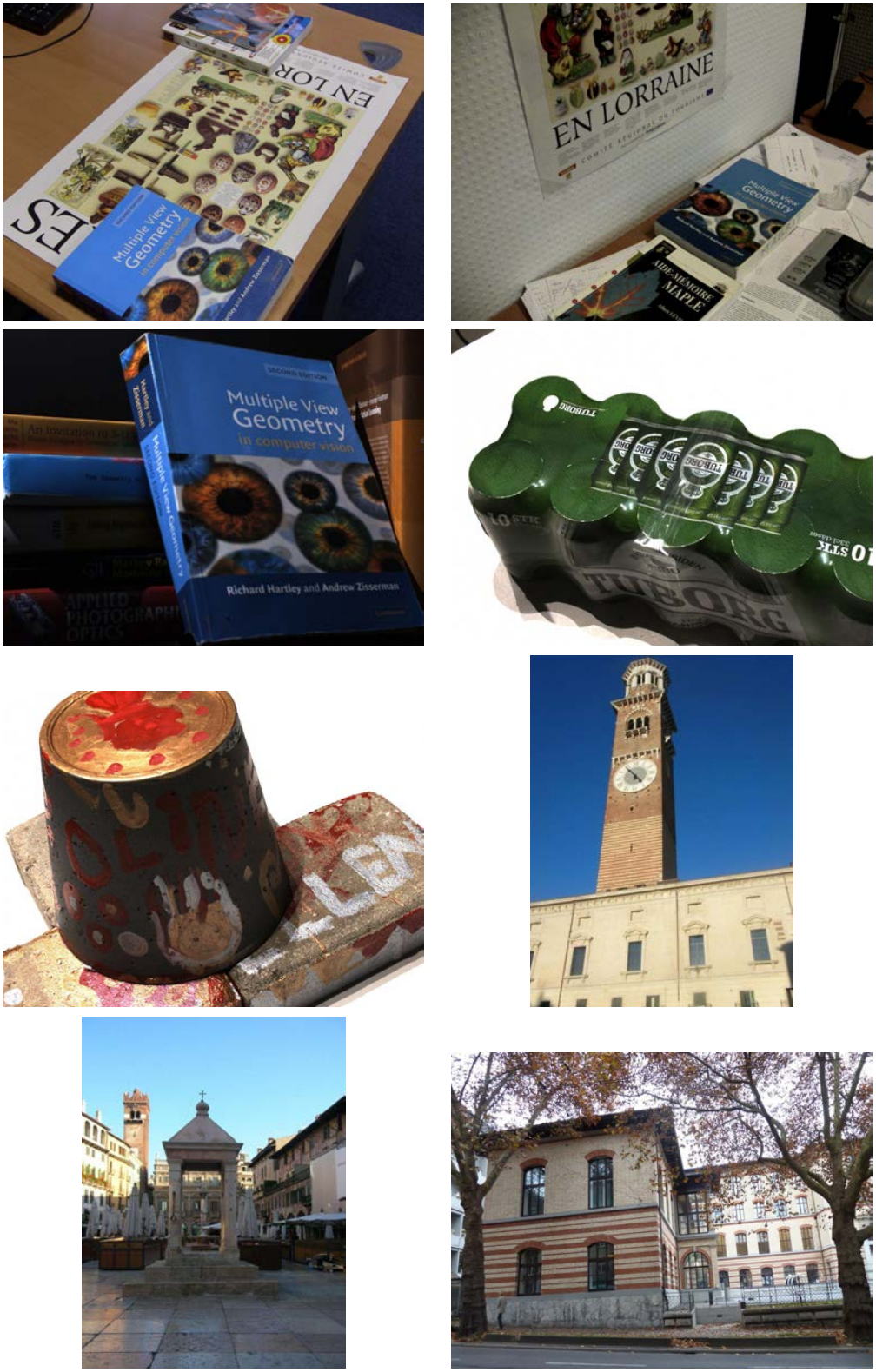


FIGURE 6.1 – Images extraites des différentes base de test présentées dans cette section. De gauche à droite et de haut en bas : poster, bureau, livre, beer, pot, tour, place et CAB.

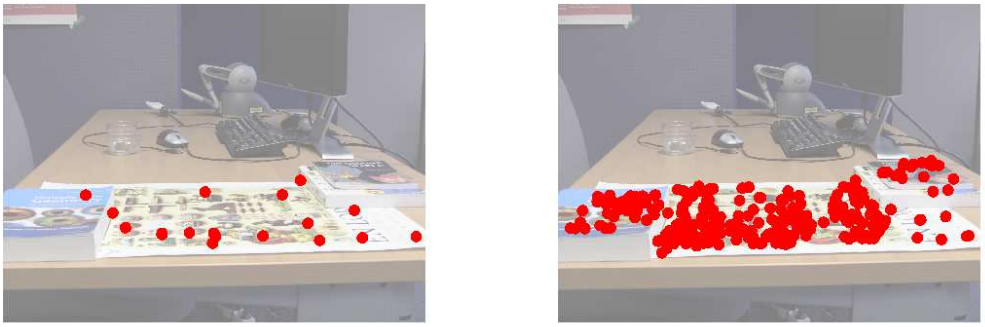


FIGURE 6.2 – Séquence poster, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2 Expériences

6.2.1 Séquence poster

La base de test poster, voir la figure 6.3, met en évidence l'utilité de la synthèse de vue avec une scène simple, réduite à un plan. Les caméras de construction observent ce plan sous une grande variété de directions modérément éloignées de la direction fronto-parallèle. La pose qu'on veut calculer est en revanche très éloignée de cette direction fronto-parallèle. Dans cette configuration l'utilisation de descripteurs obtenus par synthèse de vues augmente significativement le taux d'inliers parmi les correspondance image-modèle, de 7% à 70%. Avec 7% d'inliers, estimer une pose correcte est impossible, alors qu'avec 70% d'inliers RANSAC-PnP converge très rapidement vers une pose correcte, environ 10 itérations. La figure 6.2 montre la répartition des points correctement mis en correspondance avec et sans utilisation de synthèse de vues. On constate pour cette expérience une augmentation considérable du nombre de correspondances correctes, ainsi qu'une meilleure répartition de ces correspondances dans l'image. Ceci explique pourquoi le calcul de pose devient possible lorsqu'on ajoute les descripteurs obtenus par synthèse de vues.

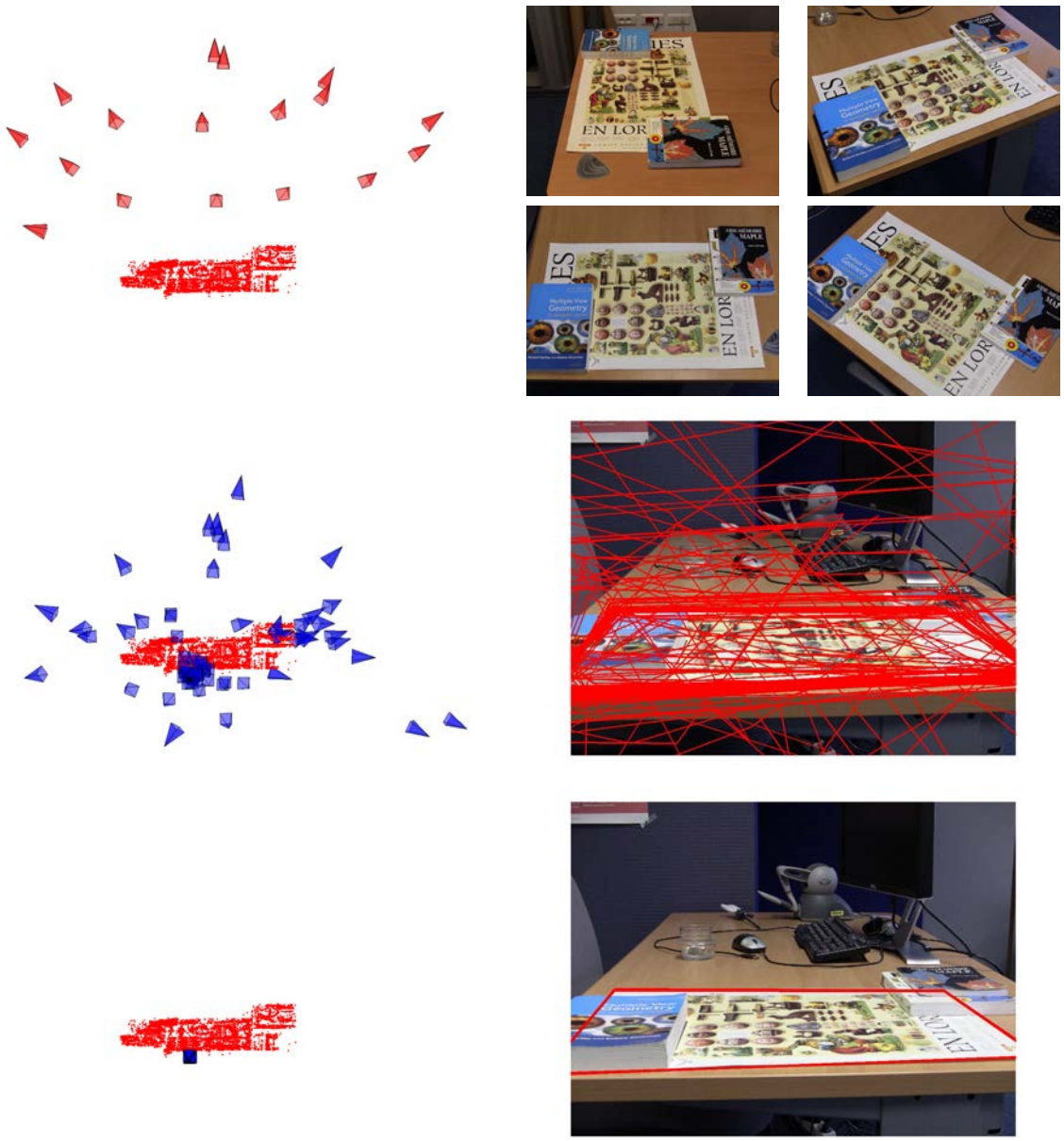


FIGURE 6.3 – Séquence poster. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.

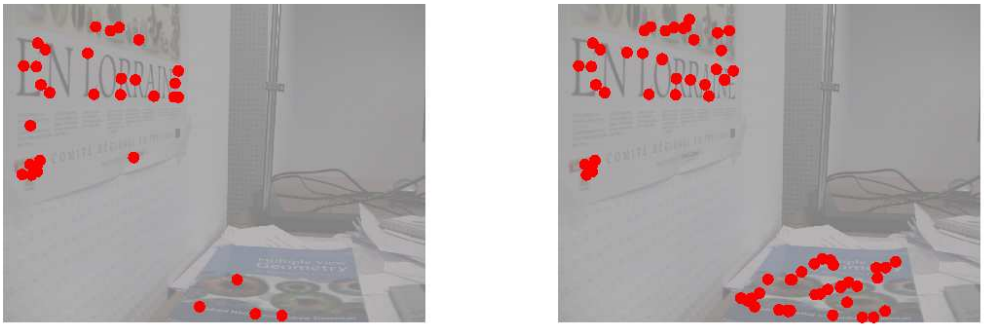


FIGURE 6.4 – Séquence bureau, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2.2 Séquence bureau

La scène bureau, voir la figure 6.5, est plus complexe que la scène poster et illustre l'utilité de la synthèse de vues dans une scène de plus grande taille. Dans ce cas, même si la scène n'est pas réduite à un plan, les points d'intérêt utiles au calcul de la pose se concentrent sur deux plans qui dominent la scène, à savoir le mur et la surface du bureau. La figure 6.4 montre que, en l'absence de synthèse de vues, l'essentiel des points correctement mis en correspondance se trouvent sur le mur, et très peu se trouvent sur le plan du bureau. Avoir l'ensemble des correspondance concentrées sur un seul plan observé avec une direction très éloignée de la normale ne permet pas d'estimer correctement une pose. Cela qui explique la dispersion des caméras observable dans la figure 6.5 lorsqu'on n'utilise pas la synthèse de vues. L'ajout de descripteurs par synthèse permet d'avoir un grand nombre de correspondances sur les deux surfaces, ce qui permet de calculer la pose. Dans cette base le taux d'inliers parmi les correspondances image-modèle varie de 11% à 18%.

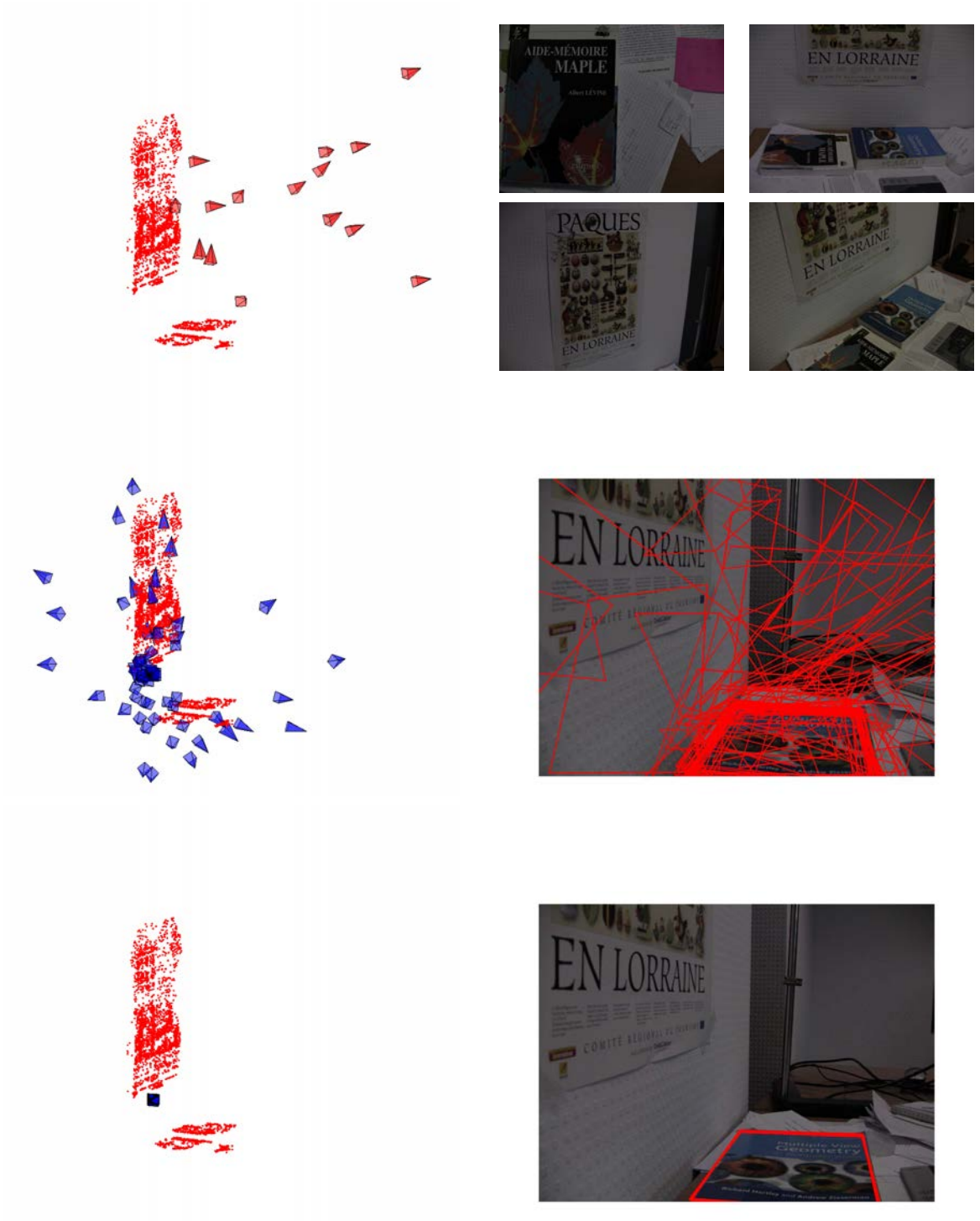


FIGURE 6.5 – Séquence bureau. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.

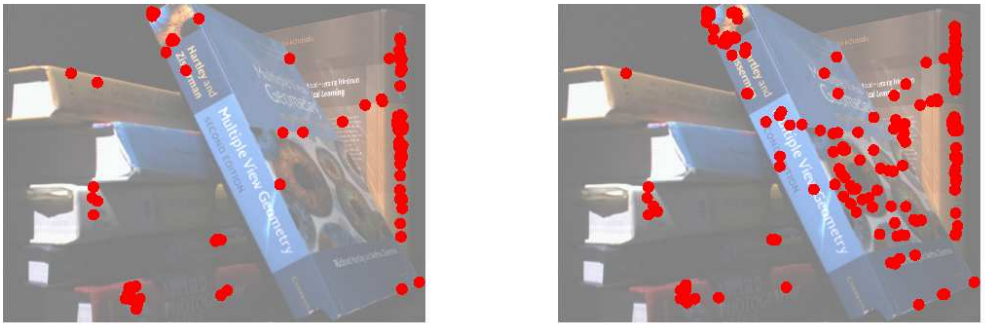


FIGURE 6.6 – Séquence book, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2.3 Séquence livre

L'expérience sur la base livre, voir la figure 6.7, met en évidence un problème de motif répété. La séquence d'image utilisée pour la reconstruction du modèle est dominée par un plan, la couverture du livre. La couverture est observée avec une direction proche de la direction fronto-parallèle dans les vues de construction. Dans la vue de test, en revanche, la couverture du livre est particulièrement déformée par rapport aux vues de construction, mais la tranche du livre est observée de façon fronto-parallèle. Comme un motif de la texture de la couverture se répète sur la tranche, on obtient un grand nombre de correspondances entre ces deux motifs. En l'absence de vue de synthèses, ce motif est le seul élément susceptible d'être mis en correspondance entre l'image requête et les vues de construction. Par conséquent, on observe des groupes de poses erronées bien distinct. Ces trois groupes de caméras observés sont : 29 caméras correctes et deux groupes de 15 et 56 caméras incorrectes. Les deux groupes de caméras erronées sont dus au même motif répété et correspondent aux deux positions symétriques possibles, comme cela avait été discuté dans la section 3.3.4 du chapitre 3. La figure 6.6 montre que l'essentiel des correspondances obtenues à l'aide de la synthèse de vues sont réparties sur la couverture du livre. Dans cette expérience la synthèse de vues permet d'augmenter le taux d'inliers parmi les correspondances image-modèle de 16% à 34%. De plus les nouvelles correspondances sont mieux réparties dans l'image, ce qui permet de lever l'ambiguïté provoquée par le motif répété.

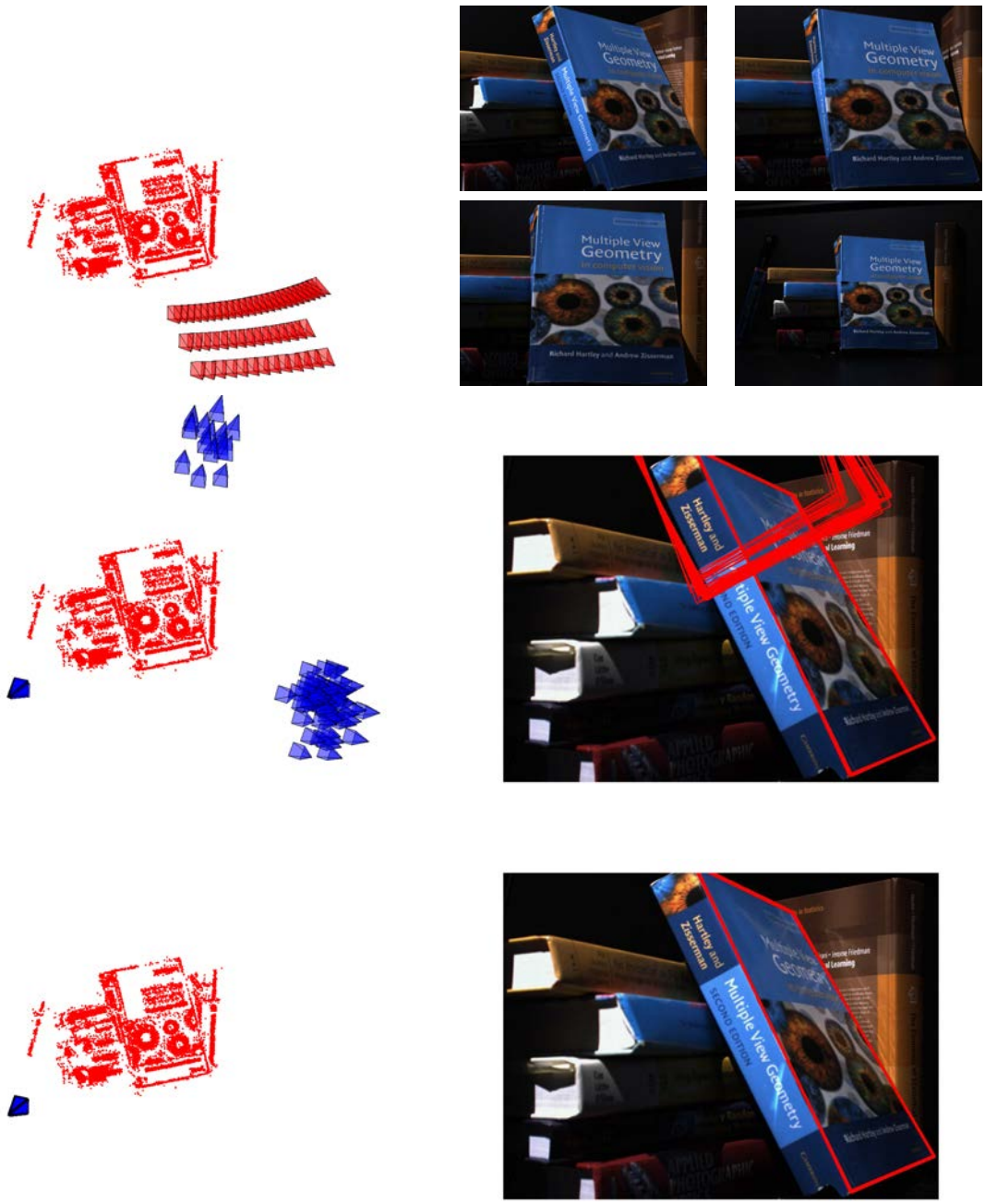


FIGURE 6.7 – Séquence livre. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.



FIGURE 6.8 – Séquence pot, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2.4 Séquence pot

L'expérience pot, voir la figure 6.9 est une scène composée de deux objets : une brique sur laquelle est posée un pot. Cette scène illustre les problèmes de visibilité dans une situation simple. La partie supérieure de la brique est identifiée comme un unique patch lors de la segmentation, et le pot posé dessus en masque une large partie lorsqu'on observe la scène sous certaines directions. Dans cette expérience, utiliser la contrainte de visibilité abordée dans la section 5.3.2 réduit le nombre de descripteurs dans le modèle enrichi de 229752 à 134484. Dans cette expérience, les seuls points correctement mis en correspondance en l'absence de synthèse de vues sont répartis sur l'arrête de la brique, voir figure 6.8. L'apparence de cette arrête est effectivement moins sensible aux changements de points de vue que les surfaces de la scènes. lorsqu'on ajoute des descripteurs par synthèse on obtient un grand nombre de correspondances réparties sur la brique. Dans cette expérience le taux d'inliers évolue de 18% à 40% entre le modèle SfM et le modèle enrichi par synthèse de vue, ce qui se traduit par une amélioration visible de la précision des poses estimées, comme illustré dans la figure 6.9.

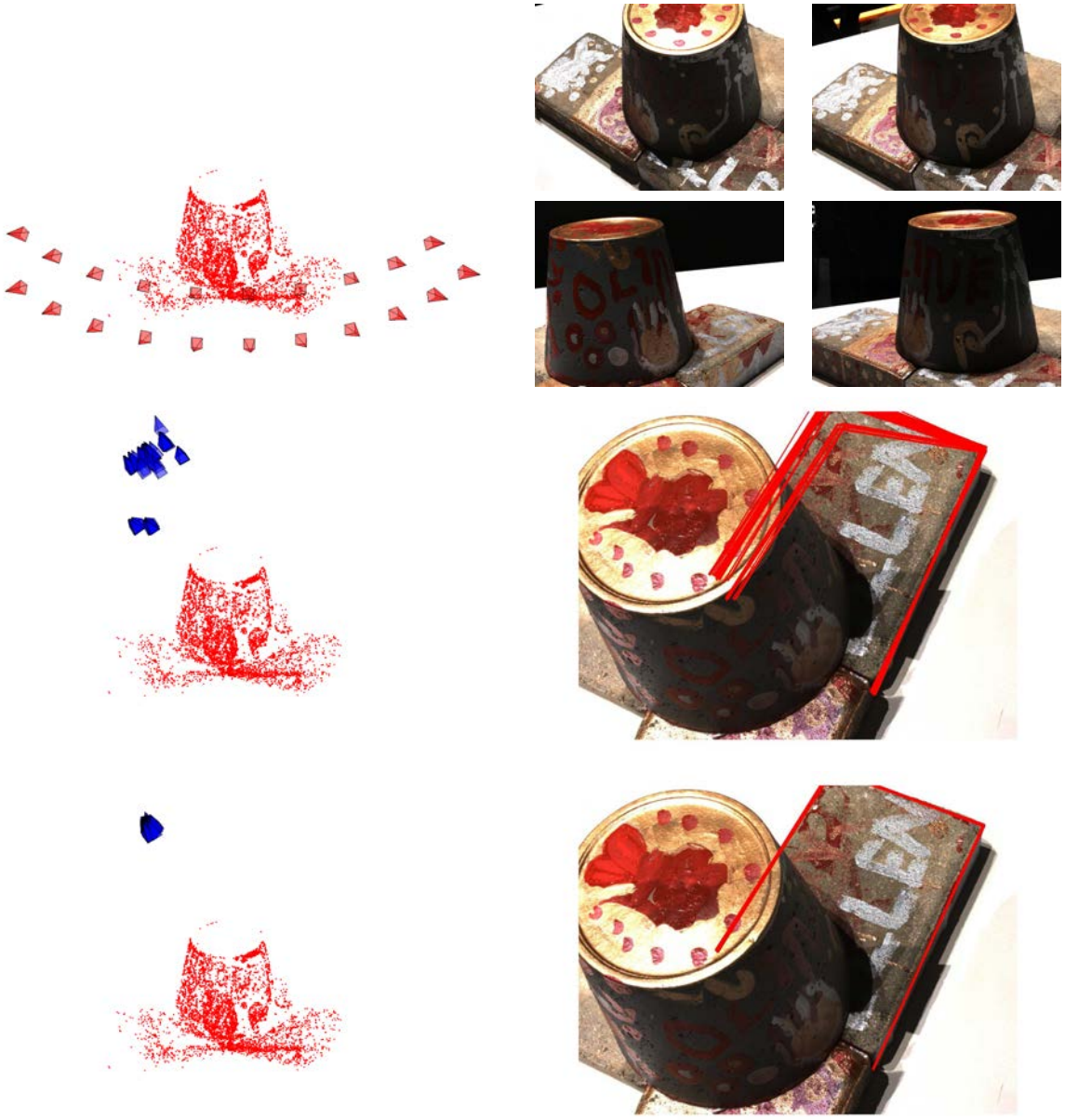


FIGURE 6.9 – Séquence pot. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.



FIGURE 6.10 – Séquence tour, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2.5 Séquence tour

L'expérience tour, voir la figure 6.11, utilise une scène d'extérieur simple, réduite à une façade. Les points de ce modèle sont répartis sur un unique plan, comme pour la base poster. En revanche le rapport entre la taille de la scène et la distance d'observation est très différent : dans poster on observe un poster à une distance de moins d'un mètre, dans tour on observe un bâtiment de plusieurs dizaine de mètres à une distance de quelques mètres. Par conséquent, comme expliqué dans la section 5.1 du chapitre 5 les points de la scènes sont répartis en plusieurs patchs, et les caméras virtuelles sont positionnées autour de ces patchs. La figure 6.10 montre que les correspondances 2D/3D sont plus nombreuses et mieux réparties dans l'image lorsqu'on utilise la synthèse de vue. On observe également que beaucoup de correspondances correctes sont situées sur la tour elle même, malgré sa texture répétitive. Dans cette expérience, le taux d'inliers évolue de 9% à 23%. Dans le premier cas obtenir une pose correcte est pratiquement impossible, comme l'indique la dispersion des poses calculées à partir du modèle sans synthèse. Dans le deuxième cas toutes les poses calculées sont proches de la pose de référence. Cette expérience montre que les conditions de l'exemple jouet poster (scène essentiellement plane, augmentation considérable du taux d'inliers) peuvent se présenter dans une acquisition plus standard, en extérieur.

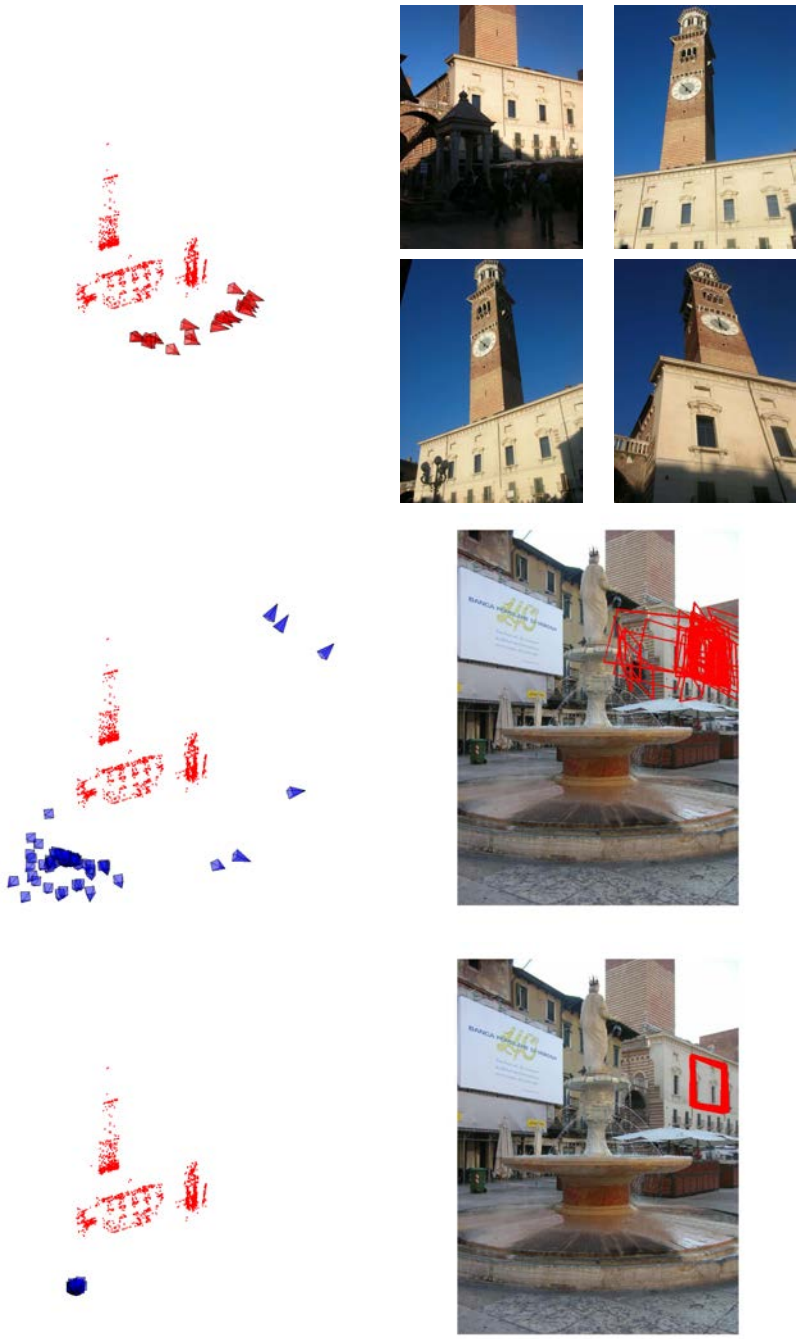


FIGURE 6.11 – Séquence tour. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.

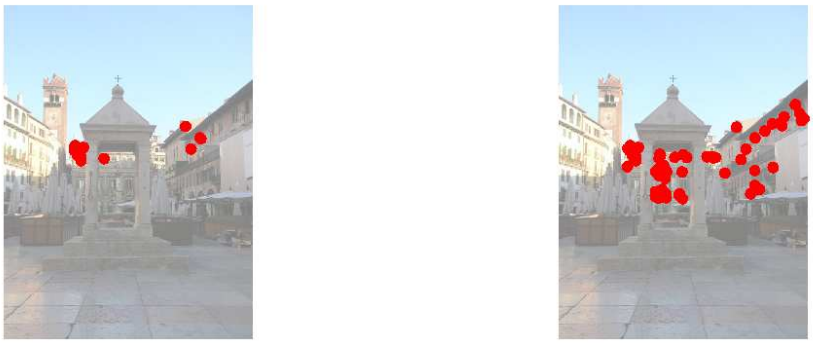


FIGURE 6.12 – Séquence place, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2.6 Séquence place

La base place, voir la figure 6.13, est un exemple de scène d'extérieur où placer des caméras autour de la scène n'aurait pas de sens. Dans cette séquence les caméras de construction sont situées dans la scène. La pose qu'on cherche à calculer est proche de certaines caméras de construction, mais leurs direction de vue sont très différentes. Cet exemple montre qu'il n'est pas nécessaire d'être dans des condition totalement différentes (fort changement de position et d'orientation), pour que la synthèse de vue améliore significativement le calcul de la pose. Dans ce cas, le calcul de pose est également difficile à cause d'une occultation, voir la vue test visible dans la figure 6.13. Cette sculpture qui bloque la vue n'a pas été reconstruite et ne fait pas partie du modèle SfM, elle n'est donc pas prise en compte par notre contrainte de visibilité. La façade partiellement masquée par la sculpture dans l'image de test, au fond de la place, est le seul élément de la scène qui a déjà été observé sous un angle comparable dans les vues de construction. Par conséquent, sans synthèse de vues, seuls quelques points sur cette façade sont utilisés pour calculer la pose, ce qui explique la grande variabilité de la pose calculée. Dans cette expérience, les correspondances utilisées pour calculer les poses dans la situation sans synthèse sont donc des inliers, mais leur répartition dans l'image ne permet pas de calculer une pose correcte. La figure 6.12 illustre cette amélioration de la répartition des correspondances. En particulier, on a sensiblement plus de correspondances sur le mur visible à droite de l'image, ce qui est normal puisque ce mur n'est pas observé sous cette direction dans les vues de construction. L'ajout de descripteurs par synthèse fait passer le taux d'inliers de 20% à 40% et permet le calcul d'une pose correcte.



FIGURE 6.13 – Séquence place. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.

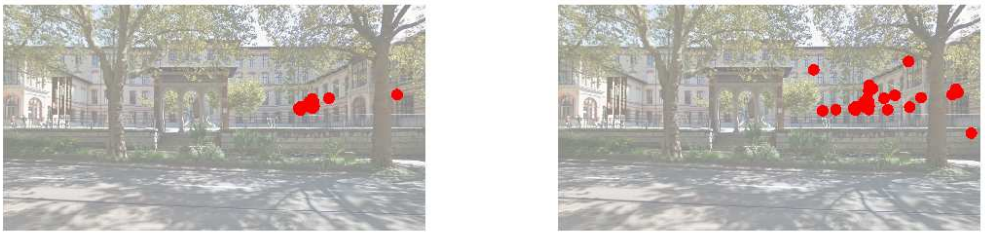


FIGURE 6.14 – Séquence CAB, répartition des points correctement mis en correspondance à l'issu de l'étape de mise en correspondance 2D/3D, sans synthèse de vues (gauche) et avec synthèse de vues (droite). L'image est celle à partir de laquelle on calcule la pose.

6.2.7 Séquence CAB

L'expérience CAB, voir la figure 6.15, donne un exemple de scène extérieure de taille plus importante, pour laquelle il est important de pouvoir placer des caméras virtuelles dans la scène et pas simplement autour. Ce modèle peut être considéré comme étant de très grande taille dans le contexte d'une application de localisation, dans la mesure où on possède généralement une estimation de la pose à une précision de quelques dizaines de mètres par GPS. Cette scène a été reconstruite à partir d'un ensemble d'images prises proches des façades du bâtiment, (disponibles à l'adresse *https ://cvg.ethz.ch/research/symmetries-in-sfm/*). L'image à partir de laquelle on calcule la pose a été obtenue via Google Street View, à partir de la rue. Dans cette expérience le taux d'inliers parmi les correspondances image-modèles augmente de 20% à 27% grâce à la synthèse de vue. De plus, les correspondances correctes sont toutes concentrées dans une petite partie de l'image dans le cas sans synthèse de vues alors qu'elles se répartissent sur plusieurs murs lorsqu'on ajoute les descripteurs obtenus par synthèse (figure 6.14). Dans cette expériences d'autres problèmes viennent s'ajouter aux changement de point de vue : les condition d'illumination sont très différentes entre l'acquisition et le test, l'appareil n'est pas le même et la scène est largement masquée par des éléments non reconstruits. Cette expérience montre que même dans ces conditions difficiles et dans un environnement de très grande taille, par rapport aux applications envisagées, la méthode proposée améliore significativement les résultats du calcul de pose.

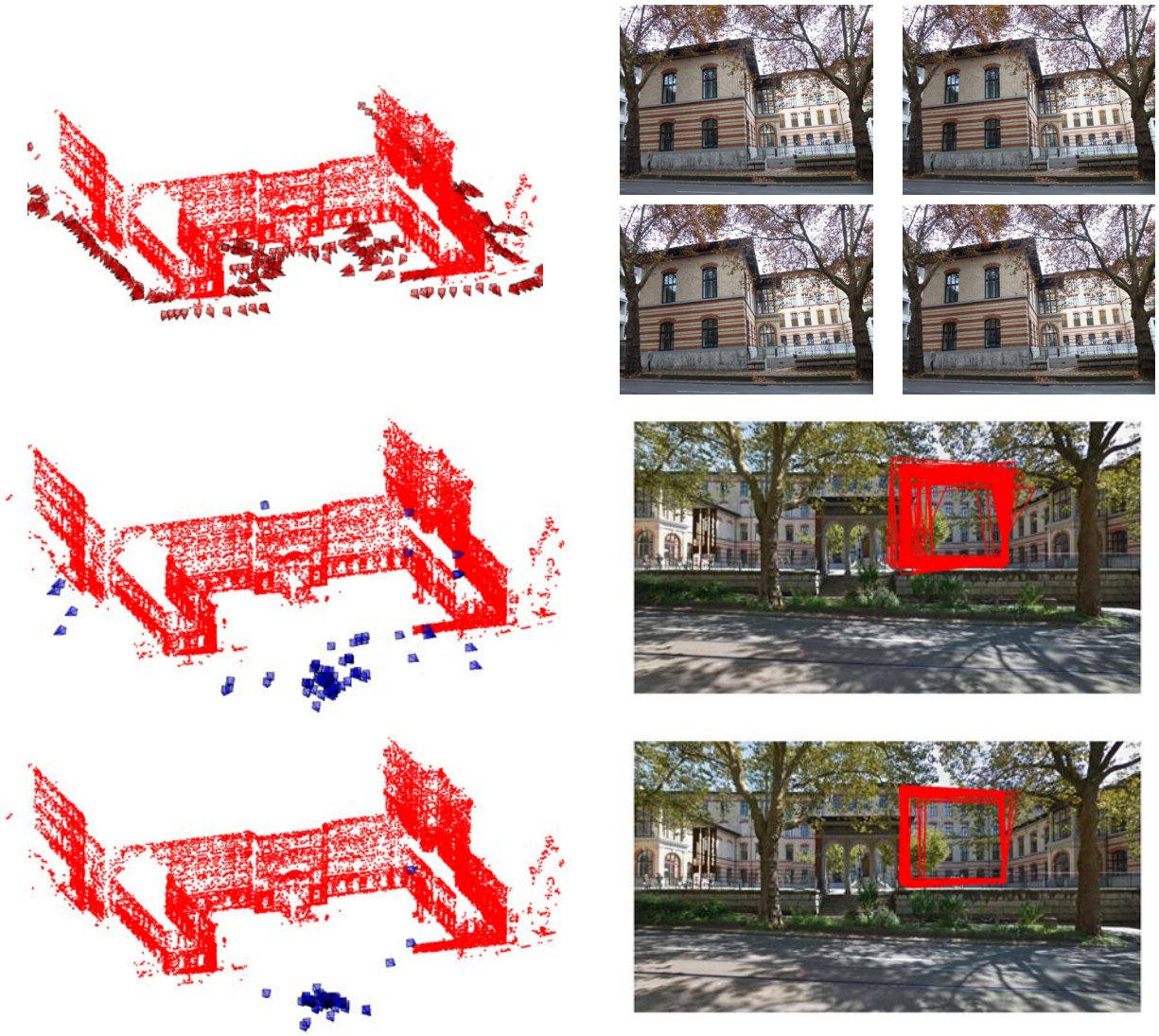


FIGURE 6.15 – Séquence CAB. La première rangée montre la position des caméra de construction par rapport au modèle et un échantillon des images utilisées pour la reconstruction. La deuxième et la troisième rangée montrent les positions de caméras et les reprojections des contours pour respectivement 100 poses calculées à partir du modèle SfM et 100 poses calculées à partir du modèle enrichi par synthèse de vues.

	poster	bureau	book	pot	tower	place	CAB
SfM	47.0	134.1	731.4	6.9	12.1	4.5	26.3
SfM + synthèse	0.9	3.1	2.0	3.4	3.3	3.6	16.1

TABLE 6.1 – Médiane de l'erreur de reprojection des contours extraits dans l'image requête, pour les 100 poses calculées dans chaque expériences. L'erreur très élevée pour l'expérience book vient d'une erreur systématique sur l'évaluation de la pose dues à un motif répété.

6.3 Conclusion

Les expériences passent en revue plusieurs situations dans lesquelles la méthode proposée améliore la qualité des poses calculées : des environnement simples composés de quelques objets, des scènes présentant des problèmes particuliers, des scènes extérieures de grande taille... Les résultats obtenus montrent que, dans toutes ces situations, la synthèse de vue améliore significativement la précision des poses calculées, bien que toutes ces expériences aient été réalisées en laissant RANSAC estimer en ligne le nombre d'itérations à faire. Cela signifie que non seulement les poses calculées sont plus précises lorsqu'on ajoute des descripteurs par synthèse mais également que le calcul de pose est plus rapide, comme cela avait été illustré dans le chapitre 5. De plus, dans un certaines expériences, comme poster ou tour, le calcul de pose est rendu possible par l'utilisation de synthèse de vues. Dans toutes les expériences, le taux d'inliers augmente significativement : de 7% à 70% dans le cas le plus extrême (poster), de 20% à 27% dans le cas où l'écart est le plus faible (cab). Dans tous les cas, la répartition des correspondances correctes dans l'image est visiblement améliorée par l'utilisation de synthèse de vue, ce qui, en plus du nombre de correspondances correctes, explique l'amélioration des poses calculées. La table 6.1 donne les erreurs de reprojection des coins des contours dans les différentes expériences.

Les poses calculées avec l'aide de descripteurs synthétiques sont plus précises pour différentes raisons illustrées dans les expériences proposées. De façon générale, le taux d'inliers parmi les correspondances image-modèle augmente suffisamment pour que l'étape RANSAC-PnP converge significativement plus rapidement. L'augmentation du nombre de correspondances correctes permet également de distinguer la pose correcte d'artefacts dus à des motifs répétés. Enfin, les descripteurs synthétiques permettent d'avoir des correspondances mieux réparties dans l'image, ce qui réduit la variabilité des poses estimées.

Chapitre 7

Conclusion et perspectives

7.1 Conclusion

Dans cette thèse nous avons abordé le problème du calcul d'une pose à partir d'une image seule et d'un modèle SfM.

Le chapitre 3 présente un procédé de synthèse consistant à simuler localement l'apparence de la scène autour de chaque point 3D du modèle. Les vues de synthèse sont obtenues par transformation 2D des vues existantes de la scène. Deux modèles de transformation, correspondant à deux modèles de caméra, sont envisagés : le modèle affine et le modèle homographique. La méthode de synthèse produit des petits patchs de 100 pixels de côtés centrés sur les points du modèle et correspondant à des points de vue virtuels répartis sur un hémisphère centré sur la scène. Les expériences menées permettent de montrer que ce type de synthèse de vue permet de calculer des poses à partir d'images éloignées des vues de construction. On observe aussi que dans ce contexte le modèle homographique est en tout point préférable : les images produites sont plus proches de la réalité sans entraîner de coût supplémentaire en temps de calcul

Le chapitre 4 étudie la sélection, par un algorithme de type RANSAC, des correspondances image-modèle utilisées pour calculer la pose. L'accent est mis sur l'efficacité en terme de temps de calcul de cette sélection. Nous proposons une approche d'échantillonnage progressif, qui s'appuie sur un classement des correspondances selon leur qualité. La qualité des correspondances est estimée à partir de leur co-occurence dans les différents points de vue observant la scène : les correspondances correctes sont généralement associées à un ou deux points de vue, qui sont les points de vue proches de la pose cherchée. Les résultats de ce chapitre montrent que l'étape de sélection des correspondances correctes peut être faite en quelques dixièmes de seconde, à comparer aux dizaines de secondes nécessaires lorsqu'on utilise une approche RANSAC standard.

Le chapitre 5 propose une approche de la synthèse de vue applicable dans une large variété de scènes. Nous montrons que, en décomposant le modèle de la scène en morceaux de plans, nous pouvons à la fois placer les caméras virtuelles de façon optimale mais également produire des vues de synthèse ces morceaux de plans plutôt que pour un voisinage de chaque point. Les intérêts de cette nouvelle approche sont multiples. Premièrement, les positionnement des points de vue virtuels est défini pour toute scène qui

peut se décomposer en morceaux de plans. Deuxièmement, comme les points de vue sont placés par rapport à des morceaux de plan, il est possible de les placer de façon optimale, à la manière de ASIFT [Yu and Morel, 2011]. Enfin, en faisant des synthèse par patch plutôt que par point, et en prenant en compte la visibilité de la scène, le temps de calcul des vues de synthèse est réduit à quelques minutes dans notre implémentation MATLAB. Les résultats présentés dans le chapitre 6 montrent que ce procédé de synthèse de vues améliore significativement les possibilités de calcul de pose dans une grande variété d'environnements.

Ce chapitre présente plusieurs perspectives pour continuer ces travaux. Le problème en suspens le plus important est la génération de correspondances 2D/3D candidates pour le calcul de pose. En effet, le chapitre 4 se concentre sur la sélection d'un ensemble de correspondances correctes à partir d'un ensemble de candidates, mais nous n'avons pas développé de méthode propre à notre problème pour la génération de cet ensemble de correspondance candidates. Nous proposons également deux améliorations possibles pour notre approche de synthèse. Premièrement, nous considérons la possibilité de placer des caméras virtuelles à des distances variables de la scène, ce qui permettrait d'augmenter encore plus la couverture de la scène en terme de points de vue. Deuxièmement, il est possible de raffiner itérativement la pose calculée en ajoutant des vues de synthèse au moment du calcul de pose. Les sections suivantes présentent ces différentes perspectives.

7.2 Représentation compacte du modèle

La méthode de synthèse de vues développée dans la thèse augmente de façon significative le nombre de descripteurs SIFT dans le modèle. Le chapitre 4 montre qu'il est possible de calculer une pose par RANSAC-PnP à partir de modèles avec un grand nombre de descripteurs en un temps raisonnable pour notre application. Cependant, l'étape de recherche de correspondances 2D/3D candidates est généralement coûteuse en temps de calcul en raison de la taille du modèle. On s'intéresse ici à des représentations alternatives du modèle de la scène qui permettraient d'accélérer la recherche des correspondances 2D/3D.

On rappelle que le modèle de la scène que nous utilisons est composé de points 3D, chacun associé à un ensemble de descripteurs SIFT. Dans cette section on s'intéresse à ces ensembles de descripteurs et comment les représenter pour accélérer la recherche de correspondances image-modèle. Les correspondances image-modèle sont obtenues par recherche au plus proche voisin entre les descripteurs de l'image et ceux du modèle.

Les idées proposées dans cette section viennent de résultats indiquant que les ensembles de descripteurs associés aux points du modèle possèdent une structure sous-jacente. Dans les modèles que nous utilisons les classes de descripteurs sont des ensembles de descripteurs SIFT, c'est-à-dire des vecteurs de dimension 128. Proposer une représentation visuelle de ces ensembles de descripteurs est difficile, mais on peut par exemple projeter ces ensembles dans l'espace défini par les premières composantes d'une analyse en composante principales. Les représentations obtenues, voir figure 7.1, laissent clairement apparaître une structure au sein de l'ensemble de descripteur.

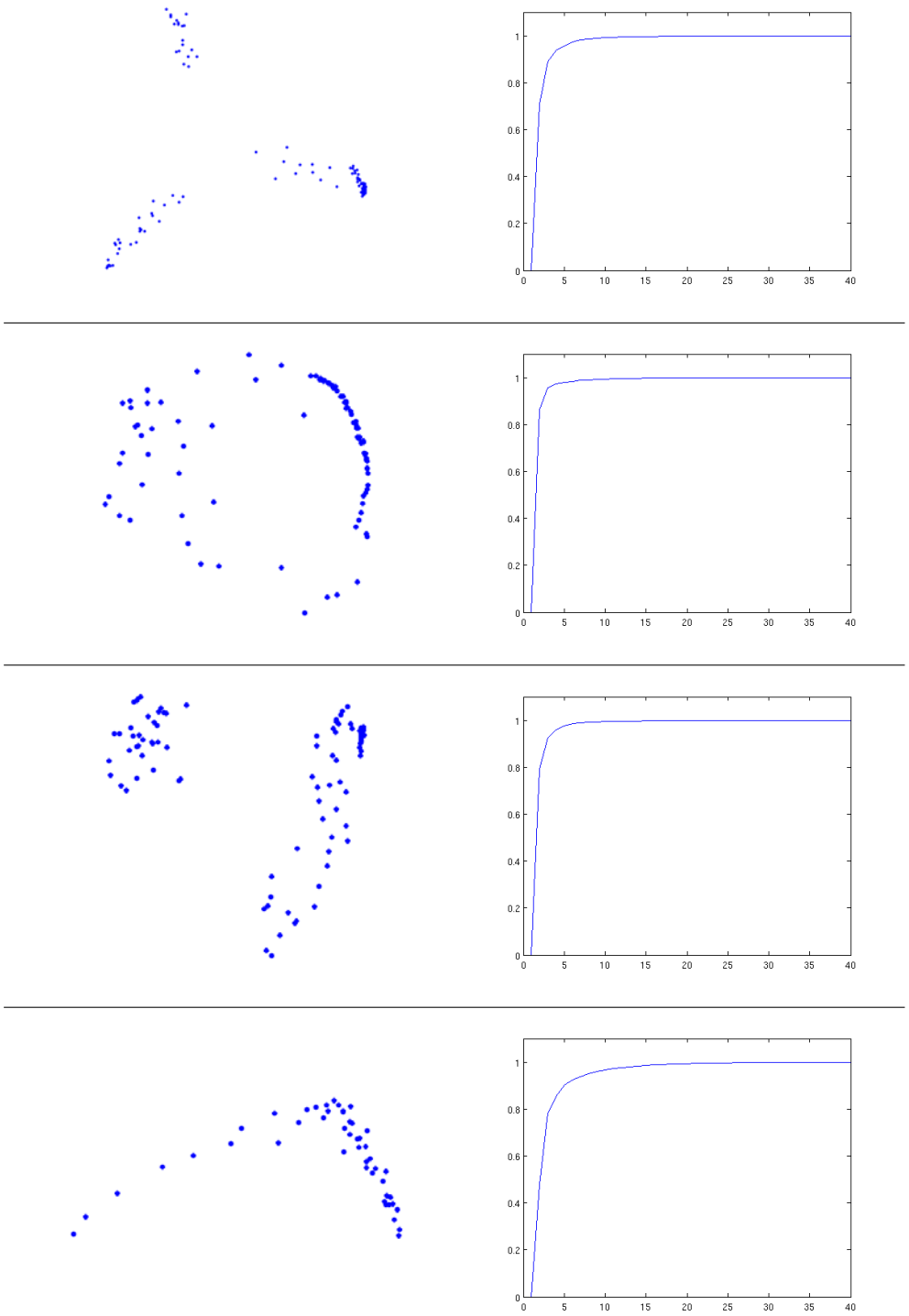


FIGURE 7.1 – Exemples d'ensembles de descripteurs associés à un point 3D du modèle de la scène. La colonne de gauche représente les descripteurs projeté dans l'espace défini par les deux premières composantes de l'analyse en composantes principales. La courbes de la colonne de droite montrent la part d'inertie expliquée par les n premières valeurs propres. On observe que les 5 premières dimensions suffisent à expliquer 90% de l'inertie.

7.2.1 Réduire une classe à quelques représentants

Pour réduire le nombre de descripteurs associés à chaque points 3D du modèle une possibilité est de choisir un nombre restreint de représentants. [Irschara et al., 2009] proposent par exemple d'utiliser une classification par mean shift sur chaque classe de descripteurs et de ne retenir que les centres obtenus. Cette approche fonctionne bien lorsque les descripteurs forment des groupes denses et localisés. C'est typiquement le cas pour les observations de surfaces planes avec une direction proche de la normale car l'apparence de la scène change peu entre de telles vues. Par conséquent elles produisent un large groupe de descripteurs similaires. A l'inverse, les observations éloignées de la direction normale produisent des descripteurs différents de toutes les autres vues. Ces considérations sont à mettre en relation avec la finesse de l'échantillonnage des caméras qui observent la scène. Avec l'échantillonage que nous utilisons, tel qu'il est décrit dans la section 5.2.2 du chapitre 5 on constate que des descripteurs isolés sont utiles pour la mise en correspondance.

La présence de descripteurs redondants dans le modèle justifie l'utilisation de ce type de méthode. Il faut cependant déterminer une façon de la mettre en œuvre sans perdre les bénéfices apportés par la synthèse, qui produit des descripteurs différents de ceux présents dans le modèle et qui sont donc susceptibles d'être mis de côtés par une méthode de type mean shift.

7.2.2 Réduction de dimension

Une autre idée pour diminuer la taille du modèle consiste à réduire la dimension des descripteurs utilisés. On peut, par exemple, utiliser les premières composantes d'une analyse en composantes principales. [Valenzuela et al., 2012] suggèrent que, pour des descripteurs SIFT, il est possible de se limiter à 70 dimensions au lieu de 128 sans affecter significativement la mise en correspondance. Dans des expériences menées sur nos données nous constatons effectivement qu'il est possible de réduire la dimension des descripteurs SIFT du modèle à environ 70 sans affecter le résultat de la mise en correspondance.

LPP (*Locality Preserving Projection* [Niyogi, 2004]) est une autre approche de réduction de dimension. La figure 7.2 propose une comparaison visuelle des résultats d'une ACP et de LPP. Cette technique vise à produire, pour un ensemble de points donnés, une base de faible dimension dans laquelle les distances relatives entre les points sont globalement préservés. Puisque l'objectif est de faire une recherche au plus proche voisin, cette approche semble naturelle. Cependant, dans les expériences que nous avons faites, utiliser LPP au lieu d'une ACP ne permet pas de réduire plus la dimension des descripteurs sans dégrader la qualité de la mise en correspondance.

Ces approches par réduction de dimension sont faciles à mettre en œuvre et ne dégradent pas la mise en correspondance. Cependant, elles n'entraînent pas un gain de temps significatif, puisque seul le temps de calcul des distances entre descripteurs est affecté. La complexité d'un calcul de distance entre des vecteurs de dimension n est de l'ordre de $O(n)$, on peut donc espérer au mieux un facteur d'accélération du même ordre que la réduction de dimension, soit 2. La table 7.1 donne les temps de calcul pour la recherche de correspondances lorsqu'on utilise une ACP ou LPP pour réduire la dimension de descripteurs. La dimension des vecteurs et le temps de calcul sont effectivement divisés

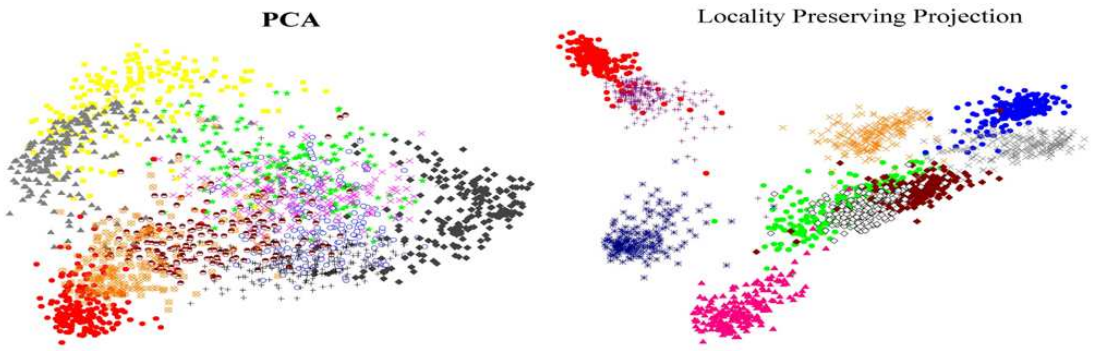


FIGURE 7.2 – Comparaison entre ACP et LPP, image extraite de [Niyogi, 2004]. Les données sont des imagettes représentant des chiffres, chaque couleur correspondant à un chiffre. Les deux images sont les projections de ces données en deux dimensions, la base étant déterminée par ACP à gauche et par LPP à droite. On observe que LPP conserve des clusters bien séparés correspondants à chaque chiffre, contrairement à l'ACP.

	poster	book	pot
pas de réduction	3.32s (128)	3.00s (128)	4.40s (128)
ACP	1.51s (70)	1.49s (67)	2.45s (72)
LPP	1.32s (64)	1.53s (70)	2.58s (73)

TABLE 7.1 – Temps de recherche des correspondances 2D/3D en utilisant différentes techniques de réduction de dimension, et nombre de dimensions restantes après réduction. Dans toutes ces expériences le résultat de la mise en correspondance n'est pas affectée par la réduction de dimension, c'est-à-dire qu'on obtient les même correspondances dans les trois lignes.

par environ 2. Pour utiliser efficacement la réduction de dimension il faut donc trouver une approche qui permet de réduire d'avantage la dimension que les méthodes standard existantes, pour notre application particulière.

7.2.3 Approximation par morceaux

Pour représenter de façon compacte les ensembles de descripteurs associés à chaque point on peut les approximer par un ensemble de parties de faible dimension.

Une approche simple consiste à utiliser une approximation linéaire par morceaux : l'ensemble de descripteurs est segmenté et chaque partie est représentée par un sous-espace de faible dimension.

7.3 Approche incrémentale

Dans la méthode de calcul de pose proposée, la pose est calculée en une seule fois : on recherche des correspondances images-modèle puis on calcule la pose par RANSAC-PnP.

Dans un certain nombre de situations, en l'absence de synthèse de vues, on constate que la pose calculée est proche de la pose correcte. Il est certainement possible d'utiliser cette estimation initiale pour ne synthétiser que des vues proches de cette première solution pour ensuite raffiner la pose calculée.

Utiliser une telle approche demande de résoudre plusieurs problèmes. Lorsque la pose est systématiquement fausse, par exemple à cause d'un motif répété, utiliser une approche de synthèse de vues incrémentale ne permettrai pas de corriger le problème. Pour mettre en œuvre ce type de méthode il faut pouvoir estimer la confiance qu'on a dans l'estimation de pose initiale. Pour ce faire on peut, par exemple, considérer la répartition dans l'image et dans le modèle des points utilisés pour calculer la pose. En effet, dans le cas de poses erronées les points sont souvent concentrés sur une petite zone de l'image, comme c'est par exemple le cas dans les expériences livre et place présentées dans le chapitre 6. Par ailleurs, ajouter une étape de synthèse au moment du calcul de la pose peut augmenter significativement le temps de calcul, comme c'est le cas dans ASIFT par exemple. Il faut donc limiter le temps utilisé pour transformer les vues et le nombre de vues synthétisées pour qu'une approche incrémentale soit profitable.

7.4 Synthèse à différentes profondeurs

Dans les méthodes que nous proposons pour mettre en œuvre le principe de synthèse de vues les caméras virtuelles sont toujours placées à une distance fixe. Dans le chapitre 3 la distance est fixée par rapport au centre de gravité du nuage de point, dans le chapitre 5 la distance est fixée pour chaque patch, par rapport au centre du patch. Faire la synthèse à distance fixée est justifié par la robustesse des descripteurs SIFT aux changements d'échelle. En effet, comme les descripteurs sont extrait localement, une variation de la distance scène-caméra se traduit par un changement d'échelle au niveau de l'image. Certaines de nos expériences, comme celle présentée dans la section 3.3.2 du chapitre 3 par exemple, suggèrent que pour certaines applications l'invariance de SIFT est effectivement suffisante. La robustesse des descripteurs SIFT aux changements d'échelle est cependant limitée en pratique [Morel and Yu, 2011], en particulier à cause de l'échantillonnage des images.

Il existe donc un intervalle de distances scène-caméra dans lesquelles la mise en correspondance est possible et en dehors de laquelle la qualité de la mise en correspondance diminue significativement. Il faudrait pouvoir évaluer cet intervalle de façon simple, afin de pouvoir, le cas échéant, ajouter de nouveaux points de vue virtuels. La génération de vues de synthèse à différentes profondeurs pose de nouveaux problèmes. Par exemple, comment synthétiser les détails d'une surface tels qu'il apparaissent dans une vue proche à partir de vues lointaines ? Une solution à ce problème serait d'utiliser des approche de type super résolution en exploitant les différentes observations d'un même point de la scène.

7.5 Modèles non connexes

Il arrive que le modèle SfM soit constitué de plusieurs parties disjointes. Cela se produit typiquement pour des scènes d'intérieur avec plusieurs pièces reconstruites. Si la séquence a été obtenue, par exemple, à partir d'un robot naviguant entre les pièces, la mise en correspondance entre les images de la séquence est généralement difficile au moment du passage d'une pièce à une autre. De façon générale, une discontinuité dans la séquence d'images entraîne la reconstruction de plusieurs modèles séparés, c'est-à-dire plusieurs nuages de points avec des repères différents. Cette situation est relativement courante et surtout nous ne pouvons pas garantir qu'elle ne se présente pas, puisqu'on se place dans une situation ou l'utilisateur cherchant à se localiser n'est pas forcément l'utilisateur qui a reconstruit le modèle. Jusqu'à présent nous avons toujours considéré des modèles de scène à une composante, c'est à dire un nuage de point exprimé dans un seul repère. Nous devons donc nous assurer que les principes développés continuent de s'appliquer dans la situation où le modèle de la scène comporte plusieurs composantes.

La méthode proposée fonctionne partiellement dans une telle situation : on peut traiter chaque composante du modèle comme un modèle indépendant et l'enrichir par synthèse de vues, et on peut également calculer une pose pour chaque composante. Cependant, il faut proposer un critère pour décider dans quelle composante se trouve la pose correcte. Il y a également des expériences à faire pour vérifier que le placement de points de vue virtuels est correct lorsque plusieurs composantes sont reconstruites, en particulier ce qui concerne la prise en compte de la visibilité. Par ailleurs, on pourrait utiliser les résultats de la synthèse pour tenter de reconnecter le modèle. En effet, la fragmentation en sous-modèle est parfois due à de rapides changements de points de vue, dont les effets sont normalement limités par l'utilisation de vues de synthèse.

7.6 Deep learning

Les méthodes utilisant le deep learning, ou apprentissage profond, ont obtenu des résultats spectaculaires pour différents problèmes de la vision par ordinateur : classification, segmentation,... Des résultats ont récemment été obtenus avec ces méthodes dans deux aspects liés au problème étudié dans cette thèse : le calcul de descripteurs robustes et le calcul de pose.

[Yi et al., 2016] montrent qu'il est possible d'apprendre à extraire des points d'intérêt et calculer des descripteurs. Les expériences présentées montrent que ces descripteurs appris, appelés LIFT, sont plus robustes que les descripteurs SIFT à certaines transformations, en particulier les changements d'illumination et les changements de point de vue. La robustesse aux changements de point de vue n'est cependant pas suffisante pour significativement améliorer le calcule de pose dans notre contexte. Par ailleurs il est difficile d'estimer à quel point les performances de mise en correspondance de ces descripteurs appris se généralisent et quels types d'images les mettent en défaut.

[Kendall et al., 2015] montrent qu'il est possible de calculer une pose de bout en bout en utilisant un réseau de neurones. Les résultats sont prometteurs, dans la mesure où les poses calculées sont proches de la vérité terrain. La précision des poses calculées reste

cependant inférieure à celle des méthodes classiques présentées en introduction.

Les méthodes utilisant le *deep learning* ne sont donc pas, à ce jour, compétitives pour résoudre les problèmes d'initialisation de pose en présence de fort changement de point de vue. En revanche ces résultats montrent qu'une approche de type *deep learning* pourrait fonctionner pour le calcul de pose dans un avenir proche, et il faudra alors s'y comparer. Il serait également intéressant de combiner ces approches avec de la synthèse de vues telle qu'elle a été proposée dans la thèse, par exemple pour faire de l'augmentation de données.

Bibliographie

[Aanæs et al., 2012] Aanæs, H., Dahl, A., and Pedersen, K. (2012). Interesting interest points. *International Journal of Computer Vision*, 97 :18–35.

[Agarwal et al., 2011] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. *Communications of the ACM*, 54 :105–112.

[Amenta and Kil, 2004] Amenta, N. and Kil, Y. J. (2004). Defining point-set surfaces. In *ACM Transactions on Graphics*, volume 23, pages 264–270.

[Amit and Geman, 1997] Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural computation*, 9 :1545–1588.

[Billinghurst et al., 2015] Billinghurst, M., Clark, A., and Lee, G. (2015). A survey of augmented reality. *Foundations and Trends on Human-Computer Interaction*, 8 :73–272.

[Boiman et al., 2008] Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of Nearest-Neighbor based image classification. In *Conference on Computer Vision and Pattern Recognition*.

[Botterill et al., 2009] Botterill, T., Mills, S., and Green, R. (2009). New conditional sampling strategies for speeded-up RANSAC. In *British Machine Vision Conference*, pages 1–11.

[Chabot et al., 2015] Chabot, F., Chaouch, M., Rabarisoa, J., Chateau, T., and Teulière, C. (2015). Détection de pose de véhicule pour la reconnaissance de marque et modèle. In *Journées francophones des jeunes chercheurs en vision par ordinateur*.

[Charmette et al., 2016] Charmette, B., Royer, E., and Chausse, F. (2016). Vision-based robot localization based on the efficient matching of planar features. *Machine Vision and Applications*.

[Choi et al., 2009] Choi, S., Kim, T., and Wonpil, Y. (2009). Performance evaluation of RANSAC family. In *British Machine Vision Conference*.

[Chum and Matas, 2005] Chum, O. and Matas, J. (2005). Matching with PROSAC - progressive sample consensus. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 220–226.

[Chum et al., 2003] Chum, O., Matas, J., and Kittler, J. (2003). Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, pages 236–243.

[Cohen et al., 2012] Cohen, A., Zach, C., Sinha, S., and Pollefeys, M. (2012). Discovering and exploiting 3D symmetries in structure from motion. pages 1514–1521.

[Collet et al., 2009] Collet, A., Berenson, D., Srinivasa, S., and Ferguson, D. (2009). Object recognition and full pose registration from a single image for robotic manipulation. In *International Conference on Robotics and Automation*, pages 48–55.

[Crombez et al., 2015] Crombez, N., Caron, G., and Mouaddib, E. (2015). Photometric gaussian mixtures based visual servoing. In *International Conference on Intelligent Robots and Systems*, pages 5486–5491.

[Davison, 2003] Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, pages 1403–1410.

[Debevec et al., 1996] Debevec, P. E., Taylor, C. J., and Malik, J. (1996). Modeling and rendering architecture from photographs : A hybrid geometry-and image-based approach. In *SIGGRAPH*, pages 11–20.

[Feng and Hung, 2003] Feng, C. and Hung, Y. (2003). A robust method for estimating the fundamental matrix. In *DICTA*, pages 633–642.

[Fischler and Bolles, 1981] Fischler, M. and Bolles, R. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 :381–395.

[Fleishman et al., 2005] Fleishman, S., Cohen-Or, d., and Silva, C. (2005). Robust moving least-squares fitting with sharp features. In *ACM transactions on graphics*, volume 24, pages 544–552.

[Furukawa and Ponce, 2010] Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 :1362–1376.

[Garcia-Fidalgo and Ortiz, 2015] Garcia-Fidalgo, E. and Ortiz, A. (2015). Vision-based topological mapping and localization methods : A survey. *Robotics and Autonomous Systems*, 64 :1–20.

[Gordon and Lowe, 2006] Gordon, I. and Lowe, D. (2006). What and where : 3D object recognition with accurate pose. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A., editors, *Toward Category-Level Object Recognition*, Lecture Notes in Computer Science, pages 67–82.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Second edition.

[Heinly et al., 2012] Heinly, J., Dunn, E., and Frahm, J.-M. (2012). Comparative evaluation of binary features. In *European Conference on Computer Vision*, pages 759–773.

[Hesch and Roumeliotis, 2011] Hesch, J. and Roumeliotis, S. (2011). A direct least-squares (DLS) method for PnP. In *International Conference on Computer Vision*, pages 383–390.

[Holz et al., 2011] Holz, D., Holzer, S., Rusu, R. B., and Behnke, S. (2011). *Real-time plane segmentation using RGB-D cameras*.

[Hoppe et al., 1992] Hoppe, H., DeRose, T., Duchamp, T., J.McDonald, and Stuetzle, W. (1992). Surface reconstruction from unorganized points. In *SIGGRAPH*, volume 26, pages 71–78.

[Irschara et al., 2009] Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 2599–2606.

[Jensen et al., 2014] Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., and s, H. A. (2014). Large scale multi-view stereopsis evaluation. In *Conference on Computer Vision and Pattern Recognition*.

[Katz et al., 2007] Katz, S., Tal, A., and Basri, R. (2007). Direct visibility of point sets. *ACM Transactions on Graphics*, 26 :24.

[Kendall et al., 2015] Kendall, A., Grimes, M., and Cipolla, R. (2015). Posenet : A convolutional network for real-time 6-dof camera relocalization. In *The IEEE International Conference on Computer Vision (ICCV)*.

[Klein and Murray, 2008] Klein, G. and Murray, D. (2008). Improving the agility of keyframe-based slam. In *European Conference on Computer Vision*, pages 802–815.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kushnir and Shimshoni, 2012] Kushnir, M. and Shimshoni, I. (2012). Epipolar geometry estimation for urban scenes with repetitive structures. In *Asian Conference on Computer Vision*, pages 163–176.

[Lepetit and Fua, 2005] Lepetit, V. and Fua, P. (2005). *Monocular model-based 3D tracking of rigid objects*.

[Lepetit et al., 2005] Lepetit, V., Lagger, P., and Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 775–781.

[Lepetit et al., 2009] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP : An Accurate $O(n)$ Solution to the PnP Problem. *International Journal of Computer Vision*, 81 :155–166.

[Li et al., 2012] Li, Y., Noah, S., Huttenlocher, D., and Fua, P. (2012). Worldwide pose estimation using 3D point clouds. In *European Conference on Computer Vision*, volume 7572, pages 15–29.

[Li et al., 2010] Li, Y., Snavely, N., and Huttenlocher, D. (2010). Location recognition using prioritized feature matching. In *European Conference on Computer Vision*, volume 6312, pages 791–804.

[Lindeberg, 1994] Lindeberg, T. (1994). Scale-space theory : A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21 :225–270.

[Lowe, 1999] Lowe, D. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–.

[Lowe, 2004] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 :91–110.

[Matas and Chum, 2004] Matas, J. and Chum, O. (2004). Randomized RANSAC with $t_{d,d}$ test. *Image and Vision Computing*, 22 :837–842.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 :1615–1630.

[Mishkin et al., 2015] Mishkin, D., Matas, J., and Perdoch, M. (2015). Mods : Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 141 :81 – 93.

[Moreels and Perona, 2007] Moreels, P. and Perona, P. (2007). Evaluation of features detectors and descriptors based on 3D objects. *International Journal of Computer Vision*, 73 :263–284.

[Morel and Yu, 2009] Morel, J.-M. and Yu, G. (2009). ASIFT : A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2 :438–469.

[Morel and Yu, 2011] Morel, J.-M. and Yu, G. (2011). Is SIFT scale invariant ? *AIMS Inverse Problems and Imaging*, 5 :115–136.

[Mount and Arya, 2010] Mount, D. and Arya, S. (2010). ANN : A library for approximate nearest neighbor searching.

[Nistér, 2005] Nistér, D. (2005). Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16 :321–329.

[Niyogi, 2004] Niyogi, X. (2004). Locality preserving projections. In *Neural information processing systems*, volume 16, page 153.

[Noury et al., 2010] Noury, N., Sur, F., and Berger, M.-O. (2010). How to overcome perceptual aliasing in ASIFT ? In *International Symposium on Visual Computing*, pages 231–242.

[Ozuysal et al., 2010] Ozuysal, M., Calonder, M., Lepetit, V., and Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32 :448–461.

[Paulin et al., 2014] Paulin, M., Revaud, J., Harchaoui, Z., Perronnin, F., and Schmid, C. (2014). Transformation Pursuit for Image Classification. In *Conference on Computer Vision and Pattern Recognition*.

[Petit et al., 2012] Petit, A., Marchand, E., and Kanani, K. (2012). Tracking complex targets for space rendezvous and debris removal applications. In *International Conference on Intelligent Robots and Systems*, pages 4483–4488.

[Raguram et al., 2008] Raguram, R., Frahm, J.-M., and Pollefeys, M. (2008). A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision*, pages 500–513.

[Roberts et al., 2011] Roberts, R., Sinha, S., Szeliski, R., and Steedly, D. (2011). Structure from motion for scenes with large duplicate structures. In *Conference on Computer Vision and Pattern Recognition*, pages 3137–3144.

[Robertson and Cipolla, 2004] Robertson, D. and Cipolla, R. (2004). An image-based system for urban navigation. In *British Machine Vision Conference*, pages 1–10.

[Rolin et al., 2014] Rolin, P., Berger, M.-O., and Sur, F. (2014). Simulation de point de vue pour la localisation d'une caméra à partir d'un modèle non structuré. In *Reconnaissance de Formes et Intelligence Artificielle*.

[Rolin et al., 2015a] Rolin, P., Berger, M.-O., and Sur, F. (2015a). Simulation de point de vue pour la mise en correspondance et la localisation. *Traitement du Signal*, 32 :169–194.

[Rolin et al., 2015b] Rolin, P., Berger, M.-O., and Sur, F. (2015b). Viewpoint simulation for camera pose estimation from an unstructured scene model. In *International Conference on Robotics and Automation*, pages 6320–6327.

[Rolin et al., 2016] Rolin, P., Berger, M.-O., and Sur, F. (2016). Enhancing pose estimation through efficient patch synthesis. In *British Machine Vision Conference*.

[Royer et al., 2007] Royer, E., Lhuillier, M., Dhome, M., and Lavest, J.-M. (2007). Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74 :237–260.

[Rucklidge, 1995] Rucklidge, W. J. (1995). Locating objects using the hausdorff distance. In *International Conference on Computer Vision*, pages 457–464.

[Savarese and Fei-Fei, 2008] Savarese, S. and Fei-Fei, L. (2008). View synthesis for recognizing unseen poses of object classes. In *European Conference on Computer Vision*, pages 602–615.

[Schindler et al., 2007] Schindler, G., Brown, M., and Szeliski, R. (2007). City-scale location recognition. In *Conference on Computer Vision and Pattern Recognition*.

[Schnabel et al., 2007] Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226.

[Shan et al., 2014] Shan, Q., Wu, C., Curless, B., Furukawa, Y., Hernandez, C., and Seitz, S. M. (2014). Accurate geo-registration by ground-to-aerial image matching. In *International Conference on 3D Vision*, pages 525–532.

[Sur et al., 2013] Sur, F., Noury, N., and Berger, M.-O. (2013). An a contrario model for matching interest points under geometric and photometric constraints. *SIAM Journal on Imaging Sciences*, 6 :1956–1978.

[Svarm et al., 2014] Svarm, L., Enqvist, O., Oskarsson, M., and Kahl, F. (2014). Accurate localization and pose estimation for large 3D models. In *Conference on Computer Vision and Pattern Recognition*, pages 532–539.

[Torii et al., 2015] Torii, A., Arandjelović, R., Sivic, J., Okutomi, M., and Pajdla, T. (2015). 24/7 place recognition by view synthesis. In *Conference on Computer Vision and Pattern Recognition*, pages 1808–1817.

[Torr and Zisserman, 2000] Torr, P. and Zisserman, A. (2000). Mlesac : A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78 :138–156.

[Valenzuela et al., 2012] Valenzuela, R. E. G., Schwartz, W. R., and Pedrini, H. (2012). Dimensionality reduction through PCA over SIFT and SURF descriptors. In *International Conference on Cybernetic Intelligent Systems*, pages 58–63.

[Wendel et al., 2011] Wendel, A., Irschara, A., and Bischof, H. (2011). Natural landmark-based monocular localization for mavs. In *International Conference on Robotics and Automation*, pages 5792–5799.

[Wu, 2007] Wu, C. (2007). SiftGPU : A GPU implementation of scale invariant feature transform (SIFT).

[Wu, 2011] Wu, C. (2011). VisualSFM : A visual structure from motion system.

[Wu et al., 2008] Wu, C., Clipp, B., Li, X., Frahm, J.-M., and Pollefeys, M. (2008). 3D model matching with viewpoint-invariant patches (VIP). *Conference on Computer Vision and Pattern Recognition*.

[Yi et al., 2016] Yi, K., Trulls, E., Lepetit, V., and Fua, P. (2016). Lift : Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483.

[Yu and Morel, 2011] Yu, G. and Morel, J.-M. (2011). ASIFT : An algorithm for fully affine invariant comparison. *Image Processing On Line*.

[Zhang and Kosecka, 2006] Zhang, W. and Kosecka, J. (2006). Image based localization in urban environments. In *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 33–40.

[Zhou et al., 2016] Zhou, T., Tulsiani, S., Sun, W., Malik, J., and Efros, A. (2016). View synthesis by appearance flow. *Computing Research Repository*.

Résumé

La localisation est un problème récurrent de la vision par ordinateur, avec des applications dans des domaines multiples tels que la robotique ou la réalité augmentée. Dans cette thèse on considère en particulier le problème d'initialisation de la pose, c'est-à-dire la localisation sans information a priori sur la position de la caméra. Nous nous intéressons à la localisation à partir d'une image monoculaire et d'un nuage de points reconstruit à partir d'une séquence d'images. Puisque nous n'avons pas d'a priori sur la position de la caméra, l'estimation de la pose s'appuie sur la recherche de correspondances entre des points de l'image et des points du modèle de la scène. Cette mise en correspondance est difficile en raison de sa combinatoire élevée. Elle peut être mise en défaut lorsque l'image dont on cherche la pose est très différente de celles ayant servi à la construction du modèle, en particulier en présence de forts changements de point de vue.

Cette thèse développe une approche permettant la mise en correspondance image-modèle dans ces situations complexes. Elle consiste à synthétiser localement l'apparence de la scène à partir de points de vue virtuels puis à ajouter au modèle des descripteurs extraits des images synthétisées. Comme le modèle de scène est un nuage de points, la synthèse n'est pas faite par rendu 3D mais utilise des transformations 2D locales des observations connues de la scène. Les contributions suivantes sont apportées. Nous étudions différents modèles de transformation possibles et montrons que la synthèse par homographie est la plus adaptée pour ce type d'application. Nous définissons une méthode de positionnement des points de vue virtuels par rapport à une segmentation de la scène en patchs plans. Nous assurons l'efficacité de l'approche proposée en ne synthétisant que des vues utiles : elles sont éloignées de celles existantes et elles ne se recouvrent pas. Nous vérifions également que la scène est visible à partir des points des vue virtuels pour ne pas produire des vues aberrantes à cause d'occultations. Enfin, nous proposons une méthode de recherche de correspondances image-modèle qui est à la fois rapide et robuste. Cette méthode exploite la répartition non-uniforme des correspondances correctes dans le modèle, ce qui permet de guider leur recherche. Les résultats expérimentaux montrent que la méthode proposée permet de calculer des poses dans des configurations défavorables où les approches standard échouent. De façon générale la précision des poses obtenues augmente significativement lorsque la synthèse de vue est utilisée. Enfin nous montrons que, en facilitant la mise en correspondance image-modèle, cette méthode accélère le calcul de pose.

Mots-clés: Calcul de pose, mise en correspondance, synthèse de vues

Abstract

Localisation is a central problem of computer vision which has numerous applications such as robotics or augmented reality. In this thesis we consider the problem of pose initialisation, which is pose computation without prior knowledge on the camera position. We are interested in pose computation from a single image and a point cloud that has been reconstructed from a set of images. As we do not have prior knowledge on the camera position, pose estimation entirely rely on finding correspondences between the image and the model. The search for these correspondences is a difficult problem because of its high combinatorial complexity. It can fail if the image is very different from the ones we used to construct the model, in particular when there is a large viewpoint change between them.

This thesis proposes an approach to make matching possible in such difficult scenarios. It consists in synthesising locally the appearance of the scene from virtual viewpoints and add descriptors extracted from these synthetic views to the model. Because the scene model is a point cloud, the synthesis is not a 3D rendering but a local 2D transform of existing observations of the scene. The following contributions have been proposed. We study different transform models and show that homographic transformations are the best suited for this application. We define a method to position the virtual viewpoints with respect to a planar segmentation of the scene model. We ensure time efficiency by only synthesising useful views, i.e. views that are far from the existing one and don't overlap. Furthermore we verify that the synthesized surface is visible from the virtual viewpoint to avoid producing aberrant views due to occlusions. Finally, we propose a robust and time efficient method to research image-model correspondences. It uses geometric cues in a guided matching framework to efficiently identify sets of correct correspondences. Experimental results show that the proposed approach makes possible pose computation in situation where standard methods fail. In general the precision and repeatability of computed poses is significantly improved by the use of view synthesis. We also show that it also reduce the pose computation times by making image-model matching easier.

Keywords: Pose computation, matching, view synthesis