Positionnement visuel dans un monde d'objets

Vincent Gaudillière

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Positionnement visuel dans un monde d'objets

THÈSE

présentée et soutenue publiquement le 24 juin 2020

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Vincent Gaudillière

Composition du jury

Rapporteurs :	Michel Dhome	Directeur de Recherche CNRS – Institut Pascal
	Cédric Demonceaux	Professeur des Universités – Université de Bourgogne
Examinateurs :	Gabriela Csurka	Directrice de recherche – Naver Labs Europe
	Sylvain Contassot-Vivier	Professeur des Universités – Université de Lorraine
Encadrants :	Marie-Odile Berger	Directrice de Recherche Inria – Inria Nancy - Grand Est
	Gilles Simon	Maître de Conférences HDR – Université de Lorraine

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

Mis en page avec la classe thesul.

En géométrie, il n'y a pas de chemin réservé aux rois.
Euclide

La géométrie n'est pas vraie, elle est avantageuse.
Henri Poincaré

Remerciements

Je tiens tout d'abord à remercier ma directrice de thèse, Marie-Odile Berger, pour son exigence et son engagement, ainsi que mon co-encadrant, Gilles Simon, pour son expertise et son enthousiasme. Qu'il soit aussi remercié pour son aide précieuse, sa gentillesse, et plus généralement pour le plaisir que j'ai eu à travailler à ses côtés.

J'adresse tous mes remerciements à Monsieur Cédric Demonceaux, Professeur à l'Université de Bourgogne, ainsi qu'à Monsieur Michel Dhome, Directeur de Recherche CNRS à l'Institut Pascal, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse.

Je remercie également Madame Gabriela Csurka, Directrice de Recherche à Naver Labs, qui a accepté de faire partie de mon jury de thèse, et Monsieur Sylvain Contassot-Vivier, Professeur à l'Université de Lorraine, qui a en outre accepté de le présider.

Je tiens à remercier ma famille pour son soutien, et plus particulièrement mes parents qui m'ont appris à être curieux et à toujours chercher les réponses à mes questions.

Je voudrais également remercier l'ensemble des professeurs dont j'ai croisé la route pendant mon parcours. Certains ont souhaité que j'aille à mon rythme et m'ont accordé leur confiance. Qu'ils en soient particulièrement remerciés.

Travailler et évoluer au sein de l'équipe MAGRIT aura été pour moi une très belle expérience. Je tiens à remercier en particulier Dasha, pour m'avoir fait (re)découvrir de nombreuses perles culturelles d'ici ou d'ailleurs, Pierre-Frédéric, pour m'avoir initié au floorball et m'avoir si souvent invité chez lui ou servi de chauffeur, Erwan, pour sa sagesse et son objectivité toute scientifique à propos de la Bretagne, Fabien, pour sa disponibilité et son humour inimitable (que certains qualifieraient d'"exacerbé"), Frédéric, pour le regard bienveillant et intéressé qu'il porte sur les travaux des autres, et Brigitte, pour sa bonne humeur à toute épreuve. Je n'oublie bien sûr pas tous ceux avec qui j'ai partagé un bout de chemin et dont certains le continueront après moi : Pierre, Cong, Antoine, Juan, Thomas Mouzon, Florent, Paul, Clémence, Simo, Anastasiia, Thomas Mangin, Hugo, Amandine et Matthieu.

Je remercie également l'ensemble des personnels du centre Inria Nancy - Grand Est, et en particulier les assistantes de l'équipe (Emmanuelle et Isabelle), ainsi que le personnel de restauration (Caroline, Isabelle, Floriane, Tarek et les autres). Je pense aussi aux doctorants d'autres équipes avec qui j'ai partagé cette aventure, en particulier Théo, Daniel et Virgile.

Je tiens enfin à remercier le Dr. Lucie Valdenaire pour m'avoir permis de préparer et de réaliser cette soutenance dans des conditions de sérénité et de confiance inimaginables il y a encore quelques mois.

Sommaire

Introduction **1**

1 Contexte : la Réalité Augmentée en milieu industriel 1

1.1 La Réalité Augmentée 1

1.2 Applications possibles dans un contexte industriel 2

1.3 Le projet DGA/DGE RAPID EVORA 3

2 Périmètre et objectifs de la thèse 4

2.1 Spécificités des milieux considérés 4

2.2 Intérêts d'une approche basée image 5

2.3 Positionnement visuel dans un monde d'objets 5

3 Contributions et organisation du manuscrit 6

3.1 Publications et brevet associés 7

1 Reconnaissance visuelle de lieux **9**

1.1 Formulation générale du problème 9

1.2 Définition d'un lieu en contexte industriel 10

1.3 Etat de l'art 11

1.3.1 Traitement de l'information visuelle 13

1.3.2 Représentation de l'environnement 14

1.3.3 Génération de décision 15

1.4 Sélection de lieux candidats par approche mixte 16

1.5 Reclassement des lieux candidats par mise en correspondance locale sous contrainte géométrique 19

1.5.1 Géométries de correspondance 19

1.5.2 Estimation des géométries planes et épipolaire 21

1.6 Résultats expérimentaux 27

1.6.1 Etape 1 : Sélection des lieux candidats par approche mixte 27

1.6.2 Etape 2 : Reclassement des lieux candidats par mise en correspondance locale sous contrainte géométrique 29

1.7 Conclusion 34

2 Estimation de pose de caméra à partir d'objets : état de l'art et choix de modélisation **35**

2.1 Estimation de pose de caméra à partir d'objets : état de l'art 36
2.1.1 Approches basées sur des correspondances entre points 36
2.1.2 Approches basées sur des modèles d'objets 38
2.2 Estimation de pose de caméra à partir de correspondances conique - quadrique . 40
2.2.1 Rappels sur les ellipses et ellipsoïdes 40
2.2.2 Réprésentation des ellipses et ellipsoïdes en coordonnées homogènes 41
2.2.3 Equation de projection d'une quadrique 43
2.2.4 Linéarisation de l'équation 44
2.2.5 Détermination de la matrice de projection de la caméra 45
2.2.6 Limites de la méthode 46
2.3 L'Equation d'Alignement des Cônes 46
2.3.1 Rappels sur les cônes du second degré 46
2.3.2 Le cône de projection 47
2.3.3 Le cône de rétroprojection 48
2.3.4 L'Equation d'Alignement des Cônes 49
2.3.5 Signification des termes de l'équation 49
2.3.6 Lien avec un problème aux valeurs propres généralisé 50
2.4 Conclusion 50

3 Estimation de pose de caméra à partir de paires ellipse - ellipsoïde : synthèse des contributions **53**

3.1 Problème à un ellipsoïde 53
3.1.1 Stratégie de résolution 53
3.1.2 Formulation 54
3.1.3 Synthèse des résultats 56
3.2 Problème à N ellipsoïdes 56
3.2.1 Détermination de la position de la caméra connaissant son orientation . . 56
3.2.2 Estimation de la pose complète de la caméra 56

4 Découplage orientation - position et application à la relocalisation de caméra **59**

4.1 Eléments mathématiques 60
4.1.1 Discriminant et relations entre coefficients et racines d'un polynôme de degré 3 60
4.1.2 Décomposition d'une matrice en éléments propres 60

4.1.3 Problème aux valeurs propres généralisé . . . 61
4.2 Découplage entre orientation et position . . . 62
4.2.1 Formulations équivalente du problème . . . 62
4.2.2 Expression de la position connaissant l'orientation . . . 63
4.3 Méthode robuste d'estimation de la pose . . . 67
4.3.1 Construction du modèle . . . 67
4.3.2 Procédure de type RANSAC pour l'estimation de la position . . . 68
4.3.3 Estimation de l'orientation . . . 68
4.4 Résultats . . . 70
4.4.1 Précision de la pose avec un objet . . . 70
4.4.2 Scénarios réels . . . 73
4.4.3 Discussion . . . 79
4.5 Conclusion . . . 80

5 Détermination de l'ensemble des caméras satisfaisant une correspondance ellipse - ellipsoïde 83
5.1 Changement de variables scalaires . . . 83
5.2 Cas de l'ellipsoïde non dégénéré . . . 85
5.3 Cas du sphéroïde . . . 89
5.3.1 Cas du cône elliptique non dégénéré . . . 90
5.3.2 Cas du cône de révolution . . . 97
5.4 Cas de la sphère . . . 99
5.5 Etude de la sensibilité au bruit . . . 103
5.6 Conclusion . . . 105

6 Estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde 107
6.1 Estimation de pose à partir de deux paires ellipse - ellipsoïde . . . 107
6.1.1 Résumé de la méthode . . . 108
6.1.2 Orientation de la caméra . . . 108
6.1.3 Position de la caméra . . . 113
6.1.4 Algorithme d'estimation de la pose . . . 113
6.2 Estimation robuste de la pose et affinement . . . 114
6.2.1 Procédure de type RANSAC pour l'estimation de la pose . . . 114
6.2.2 Affinement de la pose . . . 114
6.3 Expériences et évaluation . . . 115
6.3.1 Expériences sur la base T-LESS . . . 115

6.3.2 Expériences sur la base Freiburg 120
6.4 Conclusion 121

Conclusion **123**

Annexes **125**

A Exemples de résultats de notre méthode de sélection des lieux candidats par approche mixte. **125**

B Théorème 2 : code Maple **129**

C Théorème 3 : code Maple **131**

D Etude complète des configurations du Résultat 7. **133**

Bibliographie **137**

Résumé **145**

Abstract **145**

Table des figures

1 Exemples de solutions de Réalité Augmentée présentes sur le marché. 2
2 Visuel publicitaire de l'entreprise Daqri, promettant de rendre plus rapide et ergonomique l'obtention d'informations techniques relatives à un équipement industriel, grâce à l'utilisation d'un dispositif de Réalité Augmentée. 3
3 Applications possibles de la Réalité Augmentée en contexte industriel. 4
4 Principales spécificités visuelles des environnements industriels. 5

1.1 Représentation schématique d'un système de reconnaissance visuelle de lieux. . . 10
1.2 Illustration des difficultés à définir, et à associer, les concepts de *lieu* et de *salle* en milieu industriel, en raison de la disparité des situations rencontrées. 11
1.3 Exemple de fonctionnement attendu pour un système de Réalité Augmentée dans l'Usine d'Electricité de Metz. 12
1.4 Exemples de sous-images extraites puis mises en correspondance automatiquement sur la base de descripteurs CNN. 14
1.5 Schéma de principe de la mesure de similarité entre images basée sur les similarités entre sous-parties de ces images. 17
1.6 Exemples de régions mises en correspondance automatiquement par notre méthode (images du sous-marin HMS Ocelot). 18
1.7 Illustration du comportement des deux mesures de similarité utilisées dans notre méthode, en présence d'un fort changement de point de vue. 19
1.8 Représentation schématique de la géométrie épipolaire. 19
1.9 Représentation schématique de la géométrie plane. 20
1.10 Représentation schématique de notre méthode de mise en correspondance robuste des indices locaux. 22
1.11 Vue d'ensemble de notre méthode d'estimation des homographies locales. 23
1.12 Illustration de la difficulté à mettre en correspondance des segments sur la base de leurs descripteurs, en environnement industriel. 25
1.13 Illustration de l'apport de notre méthode de mise en correspondance locale sous contrainte géométrique. 28
1.14 Comparaison des performances de récupération d'image de nos deux mesures de similarités, en fonction de deux descripteurs CNN différents. 29
1.15 Exemples d'images de test accompagnées des 5 meilleures images récupérées après application de notre méthode de sélection des lieux candidats. 30
1.16 Erreurs moyennes de différentes méthodes d'estimation de la géométrie épipolaire. 31
1.17 Exemple d'images de test pour lesquelles les inliers définis par ORSA sont plus nombreux parmi nos correspondances que parmi les correspondances entre LIFT. 32

1.18 Enveloppes convexes des inliers points et segments des homographies estimées par notre méthode, après l'étape de fusion, pour deux paires d'images de test. 33
1.19 Illustration du calcul de pose de caméra à partir d'indices locaux. 34

2.1 Exemple d'image peu adaptée aux méthodes de mise en correspondance locale, et dans laquelle les objets détectés semblent pouvoir être utilisés pour estimer la pose de la caméra. 37
2.2 Fonctions de répartition des erreurs en orientation et en position sur les poses de caméras estimées par PnP à partir des centres des boîtes détectées. 38
2.3 Reconstruction *SfM* d'une scène d'objets modélisés par des ellipsoïdes, avec modèle de caméra orthographique. 39
2.4 Reconstruction perspective d'un ellipsoïde à partir de trois ellipses projetées. . . 40
2.5 Illustration du plan image, du centre de projection, de l'ellipsoïde, et de son ellipse projetée. 48

3.1 Illustration du problème de détermination des caméras satisfaisant une correspondance ellipse - ellipsoïde. 54
3.2 Aperçu des solutions du problème de détermination des caméras satisfaisant une correspondance ellipse - ellipsoïde. 57

4.1 Illustration sur le RGB-D TUM Dataset de la situation dans laquelle un objet reçoit deux étiquettes différentes de la part du détecteur d'objets. 68
4.2 Les points de fuite de Manhattan sont des indices robustes pour calculer l'orientation de la caméra. 69
4.3 Fonctions de répartition des erreurs de reprojection moyennes et des erreurs en position des caméras obtenues par notre méthode sur la base d'images LINEMOD. 72
4.4 Erreurs de reprojection moyennes pour notre méthode sur la base de données LINEMOD complète, en fonction de l'erreur de détection sur les ellipses. 73
4.5 Illustration des erreurs sur les ellipses inscrites dans les boîtes de détection, par rapport aux ellipses projetées avec la vérité terrain de la caméra. 74
4.6 Cas d'échec de la détection des points de fuite de Manhattan. 75
4.7 Cas d'échec de notre méthode. 76
4.8 Résultats de relocalisation de caméras. 77
4.9 Illustration de la robustesse de notre méthode sur différentes images de la base de données RGB-D TUM (séquence *fr2/desk*). 81
4.10 Illustration de la robustesse de notre méthode sur différentes images de la base de données RGB-D TUM (séquence *fr3/long_office*). 82

5.1 Illustration des huit cônes de rétroprojection tangents à l'ellipsoïde, pour une valeur de σ fixée. 87
5.2 Illustration des deux caméras compatibles avec chaque cône de rétroprojection tangent à l'ellipsoïde. 89
5.3 Lieux des centres et des sommets des axes principaux des ellipsoïdes tangents au cône de rétroprojection. 92
5.4 Illustration des deux ellipsoïdes de révolution tangents au cône de rétroprojection. 92
5.5 Lieux des centres de caméras dans le cas d'un ellipsoïde de révolution. 96
5.6 Effet du bruit de détection des ellipses sur les lieux des caméras solutions. 104

6.1 Illustrations des ellipsoïdes, ellipses projetées, plan image, centre de la caméra, ainsi que des bases vectorielles associées au monde et à la caméra. 109
6.2 Illustration du biais sur les ellipses inscrites dans les boîtes de détection par rapport à la vérité terrain. 115
6.3 T-LESS : Erreurs moyennes (avec écart-type) en position et en orientation avant et après l'étape d'affinement. 117
6.4 Illustration de l'effet d'une optimisation de la pose basée sur des indices locaux (contours). 118
6.5 Freiburg : Fonctions de distribution des erreurs en orientation et en position en comparaison avec PnP. 121

Liste des tableaux

1.1 Performances moyennes de différentes méthodes d'estimation de la géométrie épipolaire, en termes de nombre d'inliers et de taux d'inliers. 31

2.1 Présentation des différents types d'ellipsoïdes. 42
2.2 Présentation des différents types de cônes du second degré. 47

4.1 Comparaison de notre méthode de relocalisation de caméra avec la méthode d'estimation de pose d'objets la plus précise actuellement sur la base d'images LINEMOD. 71
4.2 Erreurs sur la base de données RGB-D TUM en terme de position et d'orientation de caméras. 78

5.1 Configurations possibles du problème. 93
5.2 Signe des $P_i(x)$ dans le cas $S_1 < D_1 < S_2 < D_2$. 94
5.3 Signe des $P_i(x)$ dans le cas $S_1 < S_2 < D_1 < D_2$. 94
5.4 Effet du bruit de détection des ellipses sur l'estimation de la caméra. 105

6.1 Erreurs angulaires moyennes ($\pm$ écart-type) dues aux approximations, sur les images utilisées pour les tests. 108
6.2 T-LESS : Erreurs médianes ($\pm$ écart-type), sur l'ensemble des caméras, de notre méthode d'estimation de pose de type RANSAC. 116
6.3 T-LESS : Erreurs, initiales et finales, moyennes ($\pm$ écart-type) sur l'angle θ_1 (en °) en fonction de l'hypothèse initiale. 119
6.4 T-LESS : Comparaison avec CorNet sur la Scène 08 / Objet 20. 120

D.1 Configurations possibles du problème. 134
D.2 Signe de d. 134
D.3 Signe de k_1. 134
D.4 Signe de k_2. 134
D.5 Signe de k_3. 134
D.6 Ordre des racines de $P_i(x)$ ($i = 1$ ou $i = 2$). 135
D.7 Ordre des racines de $P_3(x)$. 135
D.8 Comparaison des racines de $P_1(x)$ et $P_2(x)$. 135
D.9 Signe des $P_i(x)$ dans la Configuration #1. 135
D.10 Signe des $P_i(x)$ dans la Configuration #5. 136

Introduction

Les travaux présentés dans ce mémoire s'inscrivent dans le domaine de la vision par ordinateur, cette branche de l'informatique *« dont le principal but est de permettre à une machine d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition »* [1]. Parmi les applications les plus visibles des recherches en vision par ordinateur, on peut citer les véhicules autonomes [2] qui sont déjà testés en conditions quasi réelles dans plusieurs villes du monde, l'annotation automatique d'images utilisée par un réseau social comme Facebook [3], ou encore la Réalité Augmentée récemment mise en lumière à travers le service de navigation Google Live View [4]. Au cours de ce travail de recherche, nous nous sommes intéressés aux problèmes posés par le déploiement de la Réalité Augmentée dans un contexte industriel, et plus particulièrement aux défis que des environnements industriels de grande taille (usines, centrales, navires) représentent en termes d'analyse et de traitement des images. Dans ce chapitre introductif, nous exposons le contexte général de la thèse, le périmètre des recherches, ainsi que les objectifs à remplir. Nous résumons ensuite les principales contributions apportées par ces travaux, et présentons l'organisation du mémoire.

1 Contexte : la Réalité Augmentée en milieu industriel

Si la Réalité Augmentée s'invite déjà dans différents secteurs d'activités (ex. : culture [5], divertissement [6], défense [7]), c'est probablement dans l'industrie que ses effets sont le plus attendus. Dans cette section, nous définissons précisément la notion de Réalité Augmentée, puis nous soulignons les intérêts possibles d'une telle technologie appliquée à un contexte industriel, avant de présenter le projet dans le cadre duquel cette thèse a été financée.

1.1 La Réalité Augmentée

La Réalité Augmentée peut être définie comme *« la superposition de la réalité et d'éléments (sons, images 2D, 3D, vidéos, etc.) calculés par un système informatique en temps réel »* [8]. Cette définition très large permet de regrouper sous le même terme toute action visant à *augmenter* la

1. https://fr.wikipedia.org/wiki/Vision_par_ordinateur
2. https://eng.uber.com/improving-transportation-artificial-intelligence/
3. https://engineering.fb.com/ml-applications/building-scalable-systems-to-understand-content/
4. https://www.usine-digitale.fr/article/google-maps-etend-sa-fonctionnalite-de-navigation-gps-en-realite-augmentee-a-plus-de-smartphones
5. https://www.lesechos.fr/2017/10/quand-la-realite-augmentee-sinvite-au-musee-184735
6. https://www.lavoixdunord.fr/602329/article/2019-06-21/apres-pokemon-go-le-nouveau-jeu-harry-potter-wizards-unite-sort-ce-vendredi
7. https://www.defense.gouv.fr/actualites/articles/fid-2018-raft-vers-une-realite-augmentee-pour-les-fantassins
8. https://fr.wikipedia.org/wiki/Réalité_augmentée

(a) Google Live View[11]

(b) OpenSight[12]

FIGURE 1 – Exemples de solutions de Réalité Augmentée présentes sur le marché : (a) système de navigation sur smartphone, (b) système d'aide à l'intervention chirurgicale basé sur les lunettes Microsoft Hololens.

réalité vécue par un observateur par l'ajout d'informations voulues comme pertinentes pour lui, et visant à faciliter ou améliorer son expérience. En pratique, ce terme désigne le plus souvent l'ajout d'éléments visuels, soit dans le champ de vision d'un observateur par l'intermédiaire de lunettes spécifiques (ex. : Microsoft Hololens [9], Magic Leap One [10]), soit sur un écran à travers lequel l'observateur voit la réalité (généralement un smartphone ou une tablette). La Figure 1 illustre le type d'informations qui peuvent être ajoutées pour augmenter la réalité, ainsi que les deux modes de visualisation de cette information.

L'ajout d'information visuelle au sein d'une image ou du champ visuel d'un observateur ne peut bien sûr pas se faire au hasard. En effet, il est nécessaire de connaître le point de vue (position) et la direction d'observation (orientation) de l'observateur par rapport à son environnement, afin d'être en mesure d'ajouter la bonne information au bon endroit. Cette information de pose (position + orientation) peut être obtenue à partir de capteurs embarqués dans le dispositif de prise de vue, de balises placées au préalable dans l'environnement, par traitement de l'image captée par le dispositif, ou encore par une combinaison de ces différentes méthodes. Dans le cadre de cette thèse, nous nous intéressons aux solutions basées uniquement sur l'analyse d'image, en raison de spécificités liées au contexte applicatif et détaillées dans la partie 2.1 de cette introduction.

1.2 Applications possibles dans un contexte industriel

Les effets potentiellement bénéfiques de l'utilisation de la Réalité Augmentée dans l'industrie sont très attendus, et beaucoup y voient une des briques technologiques principales de l'*Usine du Futur* [13].

La Réalité Augmentée visant à rendre plus facilement accessibles et plus pertinentes les informations reçues par son utilisateur (voir Figure 2), ses applications possibles dans l'industrie sont nombreuses : aide à la fabrication, aide à la maintenance, documentation, formation, sécurité, etc. Un ouvrier pourrait ainsi suivre en temps réel les étapes de fabrication d'un produit sans

9. https://www.microsoft.com/fr-fr/hololens
10. https://www.magicleap.com/magic-leap-one
11. https://www.blog.google/products/maps/take-your-next-destination-google-maps/
12. https://www.opensight.health/
13. https://fr.wikipedia.org/wiki/Industrie_4.0#Réalité_augmentée

FIGURE 2 – Visuel publicitaire de l'entreprise Daqri, promettant de rendre plus rapide et ergonomique l'obtention d'informations techniques relatives à un équipement industriel, grâce à l'utilisation d'un dispositif de Réalité Augmentée.

avoir à consulter le manuel correspondant (Figure 3a), un rondier pourrait avoir facilement accès aux valeurs nominales des capteurs qu'il inspecte (Figure 3b), un formateur pourrait guider à distance son apprenti (Figure 3c), ou encore un opérateur d'une grande installation scientifique pourrait visualiser le niveau de radioactivité au sein de son environnement de travail (Figure 3d).

L'industrie représente ainsi un des champs d'applications les plus prometteurs pour la Réalité Augmentée, même si de nombreuses difficultés techniques restent à surmonter.

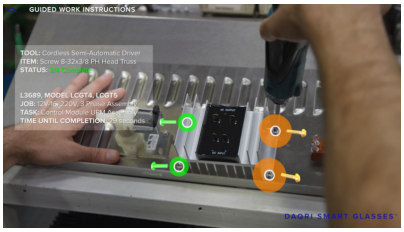
1.3 Le projet DGA/DGE RAPID EVORA

La thèse dont les travaux sont résumés dans ce mémoire a été financée dans le cadre du projet DGA/DGE RAPID [17] EVORA, réunissant l'entreprise SBS Interactive [18] et le centre de recherche Inria Nancy - Grand Est [19] (équipe-projet MAGRIT).

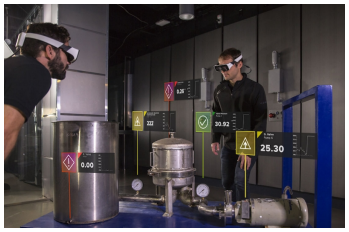
Le projet vise à permettre l'utilisation de la Réalité Augmentée dans des environnements industriels de grande taille. Pour cela, des données de prise de vues ont notamment été collectées au cours de la thèse dans l'Usine d'Electricité de Metz (UEM) [20] grâce au concours de l'entreprise. Ces données, ainsi que des échanges avec l'entreprise, nous ont permis d'identifier les principales difficultés inhérentes à l'application de techniques de vision par ordinateur dans des environnements industriels.

Certaines des méthodes présentées dans la suite du mémoire ont ainsi pu faire l'objet de tests utilisant les images de l'UEM. Pour le reste, des tests ont été conduits sur des bases de données publiques spécifiques à certains problèmes (ex. : environnements répétitifs, scènes intégrant des objets facilement reconnus par les détecteurs d'objets courants).

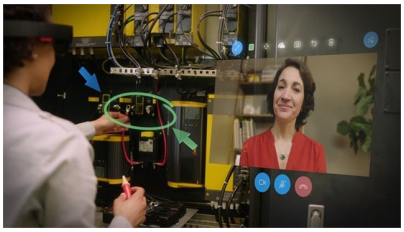
14. https://daqri.com/worksense/
15. https://azure.microsoft.com/fr-fr/blog/augmented-reality-becomes-mainstream-in-manufacturing-changes-the-face-of-the-industry/
16. https://cds.cern.ch/journal/CERNBulletin/2016/07/News%20Articles/2130668?ln=fr
17. https://www.defense.gouv.fr/aid/deposer-vos-projets/subventions/rapid
18. https://www.sbs-interactive.fr/
19. https://www.inria.fr/
20. https://www.uem-metz.fr/



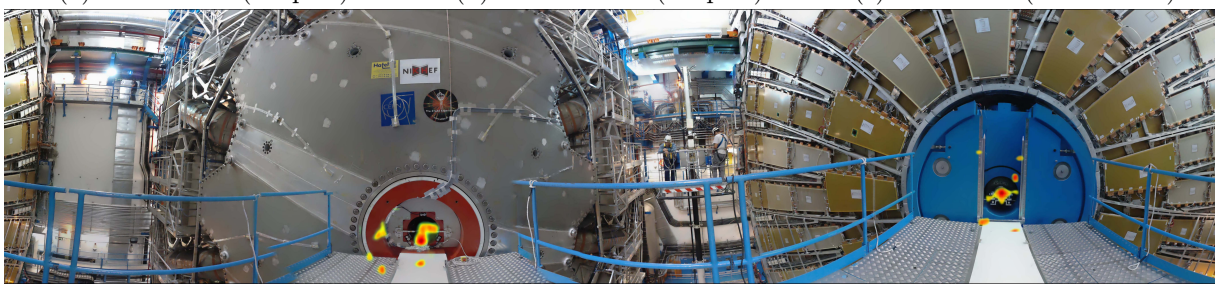
(a) Fabrication (Daqri[14])



(b) Maintenance (Daqri[14])



(c) Formation (Microsoft[15])



(d) Sécurité (Projet EDUSAFE au *Centre Européen de la Recherche Nucléaire*[16])

FIGURE 3 – Applications possibles de la Réalité Augmentée en contexte industriel.

2 Périmètre et objectifs de la thèse

Si l'objectif général du projet est de permettre l'utilisation de la Réalité Augmentée dans des environnements industriels de grande taille, seules les solutions basées sur l'analyse d'image et n'impliquant pas de modification de l'environnement (ex. : pose de marqueurs) ont fait l'objet de recherches lors de cette thèse.

2.1 Spécificités des milieux considérés

Les environnements industriels se caractérisent par leur grande diversité (fonction, taille, équipements, mode d'éclairage...). Il est cependant possible de trouver des similitudes partagées par une majorité d'entre eux, afin d'identifier les principaux problèmes à résoudre dans le cadre de ce travail de recherche.

Les environnements considérés (usines, centrales, navires) ont pour caractéristique première d'être des environnements intérieurs, ce qui les rend imperméables à l'utilisation de systèmes de positionnement par satellite (ex. : GPS[21], Galileo[22]). D'autre part, nous considérons des environnements de grande taille, éventuellement composés de plusieurs lieux (voir la définition d'un lieu au chapitre 1, partie 1.2).

Par ailleurs, les principales caractéristiques visuelles communes à ces environnements, illustrées en Figure 4, sont :

— la présence de surfaces spéculaires (ex. : matériaux métalliques) : Figure 4a,

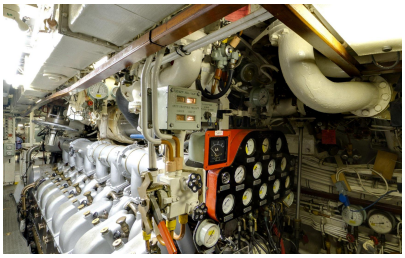
— la répétition de motifs à différentes échelles : Figures 4b,

— la présence de larges zones peu ou pas texturées : Figure 4c,

— la présence de contours d'occultation (ex. : tuyaux) : Figure 4d,

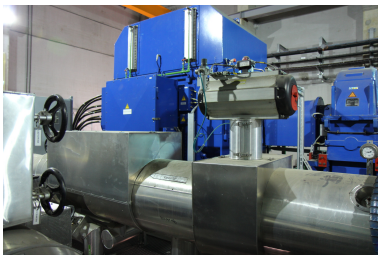
21. https://www.gps.gov/
22. https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system

(a) Spécularités[23]



(b) Motifs répétitifs[24]

(c) Contours d'occultation[25]



(d) Régions peu texturées[23]

(e) Plans verticaux[23]

(f) Contours rectilignes[23]

FIGURE 4 – Principales spécificités visuelles des environnements industriels.

— la présence de plans verticaux : Figure 4e,

— la présence de contours rectilignes : Figure 4f.

Toutes ces spécificités peuvent représenter des défis à surmonter pour un système de vision par ordinateur opérant dans un tel environnement, mais leur identification peut également permettre de développer une solution adaptée.

2.2 Intérêts d'une approche basée image

Dans cette thèse, nous nous intéressons aux solutions de Réalité Augmentée utilisant l'analyse d'image uniquement, et n'impliquant pas de modification de l'environnement (ex. : ajout de marqueurs). En effet, les systèmes de localisation basés sur l'émission/réception d'ondes physiques (magnétiques, acoustiques, optiques, wifi...) ou sur la détection de marqueurs dans les images ne sont pas adaptés aux milieux considérés car ils limitent les interactions avec l'environnement et ont une faible portée.

De plus, l'utilisation d'une caméra classique (sans information de profondeur) comme seul capteur permet d'envisager une solution de Réalité Augmentée à moindre coût.

Notre approche se veut donc robuste, mais également suffisamment générale pour pouvoir être appliquée à n'importe quel environnement de ce type.

2.3 Positionnement visuel dans un monde d'objets

Le sujet de la thèse couvre trois grandes thématiques : la reconnaissance de lieux, la reconnaissance d'objets, et la Réalité Augmentée.

Pour fonctionner dans ces environnements complexes, notre système doit en premier lieu être capable de reconnaître le lieu dans lequel se trouve l'observateur, puis, dans un second

23. Image acquise à l'UEM.
24. Image issue de la visite Google Street View du sous-marin HMS Ocelot.
25. Image prise au sein du centre Inria Nancy - Grand Est.

temps, d'estimer de manière suffisamment précise sa position et son orientation pour permettre l'affichage des informations souhaitées. Pour aborder ces problématiques, nous nous sommes tout d'abord intéressés à des méthodes classiques de vision par ordinateur, et avons relevé les limites associées à leur application en contexte industriel. Par la suite, nous avons travaillé sur l'introduction d'information sémantique dans la résolution de ces différents problèmes.

En ce qui concerne la reconnaissance de lieux (chapitre 1), la mesure de similarité entre images, ainsi que leur mise en correspondance locale, est guidée par le calcul de correspondances de niveau objet entre régions d'intérêt des deux images. Pour calculer la pose de la caméra, nous tirons ensuite profit des objets d'intérêt présents dans la scène (chapitres 2 à 6), en utilisant pour cela une modélisation sous forme d'ellipsoïdes qui permet notamment de réduire le problème d'estimation de pose de caméra à un problème d'estimation de son orientation uniquement.

3 Contributions et organisation du manuscrit

Le chapitre 1 de ce manuscrit est principalement consacré au problème de la reconnaissance de lieux. Si le calcul de pose de caméras à partir d'indices locaux est brièvement évoqué à la fin de ce chapitre, nous mettons en avant l'intérêt que représente la prise en considération des objets présents dans la scène pour définir une solution mieux adaptée aux environnements considérés (chapitre 2). Nous nous interrogeons également, dans le deuxième chapitre, sur le choix de modélisation des objets le plus pertinent, et retenons la modélisation sous forme d'ellipsoïde. Les différentes contributions apportées au problème d'estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde sont résumées au chapitre 3. Plus spécifiquement, nous avons montré que le paradigme de modélisation ellipse - ellipsoïde induit un découplage entre l'orientation et la position de la caméra, au sens où la position est entièrement déterminée par l'orientation (chapitre 4). Nous avons ensuite étudié théoriquement le problème d'estimation de pose de caméra à partir d'une seule correspondance ellipse - ellipsoïde (chapitre 5). Enfin, nous avons proposé une méthode d'estimation de pose de caméra fonctionnant à partir de deux détections d'objets dans l'image (chapitre 6).

Les principales contributions apportées par ces travaux se résument ainsi :

- une méthode de reconnaissance de lieux en deux étapes, conjuguant à la fois rapidité de traitement et précision des résultats (chapitre 1),
- une méthode d'estimation de la géométrie épipolaire guidée par des correspondances semi-globales, et tirant profit des différents indices visuels locaux pouvant être extraits des images considérées (chapitre 1),
- des contributions théoriques à la résolution du problème d'estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde : la mise en évidence de l'existence d'un découplage entre l'orientation et la position de la caméra (chapitre 4), la détermination de l'ensemble des solutions dans le cas d'une seule correspondance, en fonction du type d'ellipsoïde considéré (chapitre 5), ainsi qu'une approximation analytique de l'orientation de la caméra pouvant être calculée à partir de deux correspondances (chapitre 6),
- ces résultats sont à la base de méthodes robustes, proposées aux chapitres 4 et 6, permettant la résolution du problème d'estimation de pose de caméra à partir d'objets modélisés par des ellipsoïdes : une méthode de calcul de la position de la caméra connaissant son orientation et un ensemble de correspondances possibles entre objets détectés dans l'image et objets présents dans le modèle, ainsi qu'une méthode permettant d'estimer la pose de la caméra à partir de deux détections d'objets et d'un ensemble de correspondances possibles avec les objets du modèle.

3.1 Publications et brevet associés

Certains travaux menés dans le cadre de cette thèse ont fait l'objet de publications dans des conférences nationales et internationales, dans un journal international, ou sous forme de rapport technique. Les autres vont faire l'objet d'une soumission très prochainement. La liste des articles publiés ou en cours de soumission est indiquée ci-après :

[GSB20c] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Camera pose estimation from ellipse - ellipsoid correspondences.* International Journal of Computer Vision. **(En cours de soumission)**.

[GSB20b] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Perspective-2-Ellipsoid : bridging the gap between object detections and 6-DoF camera pose.* IROS 2020 - 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2020, Las Vegas, USA.

[GSB20a] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Perspective-2-Ellipsoid : bridging the gap between object detections and 6-DoF camera pose.* IEEE Robotics and Automation Letters, 2020.

[GSB19d] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Camera relocalization with ellipsoidal abstraction of objects.* ISMAR 2019 - 18th IEEE International Symposium on Mixed and Augmented Reality, Oct 2019, Beijing, China.

[GSB19c] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Camera pose estimation with semantic 3D model.* IROS 2019 - 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nov 2019, Macao, China.

[GSB19b] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Estimation de pose de caméra à partir de correspondances ellipse-ellipsoïde.* ORASIS 2019 - Journées francophones des jeunes chercheurs en vision par ordinateur, May 2019, Saint-Dié-des-Vosges, France.

[GSB19a] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Perspective-12-Quadric : An analytical solution to the camera pose estimation problem from conic-quadric correspondences.* Technical report, 2019.

[GSB18b] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Region-based epipolar and planar geometry estimation in low-textured environment.* ICIP 2018 - 25th IEEE International Conference on Image Processing, Oct 2018, Athens, Greece.

[GSB18a] Vincent Gaudillière, Gilles Simon and Marie-Odile Berger. *Estimation des géométries planaire et épipolaire en environnement faiblement texturé basée sur la mise en correspondance de régions.* RFIAP 2018 - Congrès Reconnaissance des Formes, Image, Apprentissage et Perception, Jun 2018, Marne-la-Vallée, France.

Les travaux présentés au chapitre 4 ont également fait l'objet d'un dépôt de brevet :

[BGS19] Marie-Odile Berger, Vincent Gaudillière et Gilles Simon. *Dispositif de traitement de prise de vue.* France, N°de brevet : FR1905535.

Chapitre 1

Reconnaissance visuelle de lieux

Sommaire

1.1 Formulation générale du problème 9
1.2 Définition d'un lieu en contexte industriel 10
1.3 Etat de l'art 11
1.3.1 Traitement de l'information visuelle 13
1.3.2 Représentation de l'environnement 14
1.3.3 Génération de décision 15
1.4 Sélection de lieux candidats par approche mixte 16
1.5 Reclassement des lieux candidats par mise en correspondance locale sous contrainte géométrique 19
1.5.1 Géométries de correspondance 19
1.5.2 Estimation des géométries planes et épipolaire 21
1.6 Résultats expérimentaux 27
1.6.1 Etape 1 : Sélection des lieux candidats par approche mixte 27
1.6.2 Etape 2 : Reclassement des lieux candidats par mise en correspondance locale sous contrainte géométrique 29
1.7 Conclusion 34

La reconnaissance visuelle de lieux (*Visual Place Recognition* en anglais), est un problème important en vision par ordinateur. Dans le cadre de cette thèse, il représente la première tâche à exécuter par le système, pour lui permettre de se localiser grossièrement dans l'environnement. C'est pourquoi on parlera indifféremment dans la suite de *localisation*. Dans ce chapitre, nous commençons par présenter une formulation générale du problème (partie 1.1), avant de nous intéresser à la définition d'un lieu en environnement industriel (partie 1.2). La partie 1.3 présentera ensuite l'état de l'art associé. La solution que nous avons retenue est enfin décrite dans les parties 1.4 et 1.5, puis évaluée dans la partie 1.6. Les travaux décrits dans la partie 1.5 ont fait l'objet de la publication [GSB18b] (cf page 7).

1.1 Formulation générale du problème

Une formulation générale du problème de reconnaissance visuelle de lieux est donnée dans [LSN+16] : *« étant donnée une image d'un lieu, un humain, animal, ou robot est-il capable de décider si cette image correspond à un lieu qu'il a déjà vu ? »*. C'est un problème qui s'étend donc bien au-delà du seul domaine informatique, mais nous nous intéressons ici seulement à la

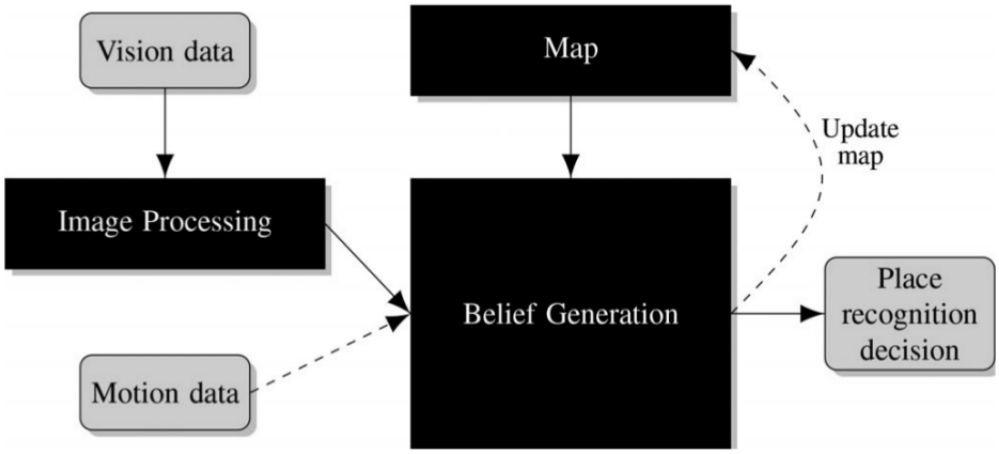


FIGURE 1.1 – Représentation schématique d'un système de reconnaissance visuelle de lieux (tirée de [LSN+16]).

reconnaissance de lieux par un ordinateur. Une formulation plus adaptée serait donc : *Comment un algorithme peut-il décider si l'image qu'il reçoit en entrée correspond à un lieu connu ?* Cette formulation, bien que plus restrictive, soulève néanmoins trois questions :

(i) Comment traiter l'image pour en extraire l'information utile ?

(ii) Comment construire une représentation de l'environnement à laquelle l'algorithme pourra se référer pour décider si le lieu est connu ou inconnu ?

(iii) Comment générer une telle décision ?

La figure 1.1 illustre schématiquement l'organisation et le fonctionnement d'un système de reconnaissance visuelle de lieux. Un tel système reçoit une information visuelle en entrée (typiquement une image), et éventuellement une information de mouvement, puis génère une décision quant au lieu représenté par l'information visuelle. Pour cela, le système est composé de trois modules principaux : le module de traitement de l'image d'entrée, qui permet d'extraire l'information pertinente de l'image, le module de cartographie, appelé également carte, qui contient la représentation que le système a de son environnement, et le module de génération de décision, qui compare l'information issue de l'image d'entrée à la carte de l'environnement pour rendre sa décision. Le système peut éventuellement mettre à jour sa carte au cours de l'opération. Il est à noter que le choix de l'architecture de chacun des trois modules est une réponse aux questions soulevées précédemment : (i) module de traitement de l'image d'entrée, (ii) module de cartographie, et (iii) module de génération de décision.

1.2 Définition d'un lieu en contexte industriel

La définition de *lieu* est cruciale dans le développement d'un tel système, et dépend grandement du contexte applicatif. Dans une maison ou un appartement, il peut être naturel de considérer qu'un lieu correspond à une pièce par exemple. Dans un environnement industriel, les choses semblent moins évidentes. En effet, si dans un sous-marin le concept de pièce ou de salle peut garder un sens, il le perd lorsque l'application a pour cadre certaines usines ou certains entrepôts de grande taille constitués essentiellement de quatre murs et d'un toit. D'autre part,

(a) Usine constituée d'une seule salle[1]

(b) Salle de la turbine de l'UEM

(c) Couloir du sous-marin HMS Ocelot

(d) Salle du sonar du sous-marin HMS Ocelot

FIGURE 1.2 – Illustration des difficultés à définir, et à associer, les concepts de *lieu* et de *salle* en milieu industriel, en raison de la disparité des situations rencontrées.

dans des espaces où chaque centimètre carré est optimisé (cf. un sous-marin à nouveau), une salle peut parfois prendre la forme d'un couloir ou d'un renfoncement. La figure 1.2 illustre les difficultés à définir, et à associer, les concepts de *lieu* et de *salle* en environnement industriel.

Il convient donc d'opter pour une définition plus fonctionnelle du concept de lieu. En effet, le but final étant l'utilisation d'un système de Réalité Augmentée, les applications d'un tel système concerneront soit l'aide à l'intervention sur un dispositif particulier (ex. : vanne, capteur, moteur), soit l'aide au déplacement entre deux dispositifs de ce type (dans le cadre d'une ronde par exemple). **Il apparaît donc pertinent de définir un lieu comme une zone d'interaction possible avec un dispositif ou objet d'intérêt**. Les reconnaissances d'objets et de lieux sont ainsi étroitement liées. La figure 1.3 illustre le fonctionnement attendu d'un système de Réalité Augmentée dans l'Usine d'Electricité de Metz : le système reconnaît qu'il est dans la zone d'interaction autour d'un manomètre et d'autres objets d'intérêt marqués en rouge. Un positionnement plus fin, qui correspond à la seconde fonctionnalité attendue de notre système, est présenté en bas de la figure.

1.3 Etat de l'art

La reconnaissance de lieux est un problème qui a fait l'objet de nombreuses recherches, et dont les principales difficultés résident dans le fait que l'apparence de ces lieux peut changer (selon le point de vue d'observation ou les conditions d'illumination par exemple), que l'environnement

1. Image de l'entreprise RDS

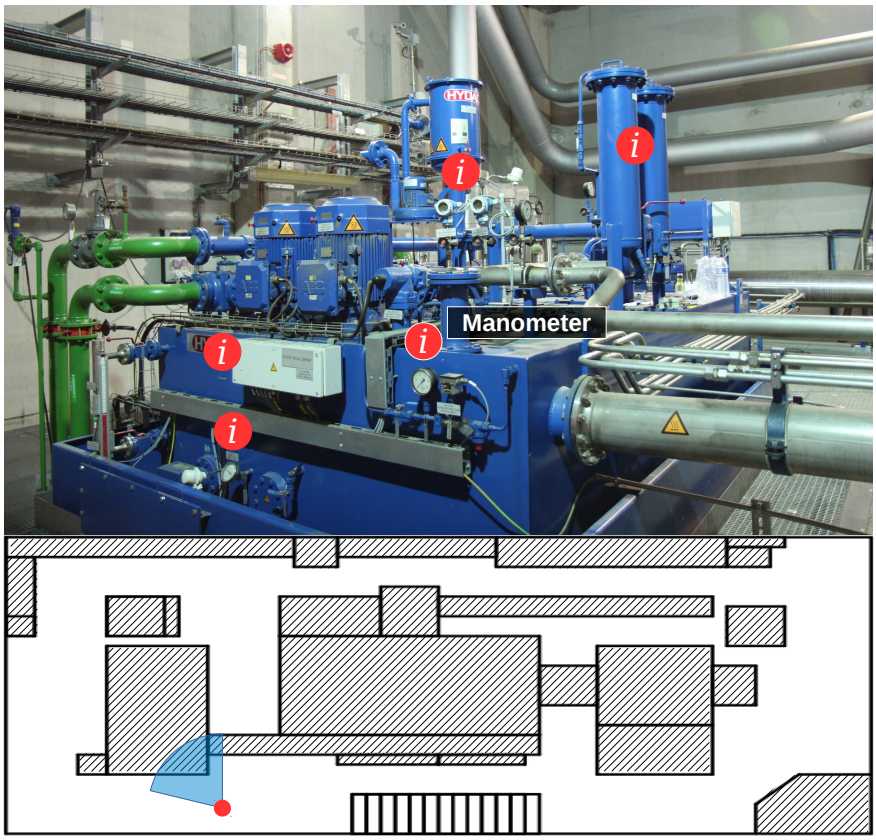


FIGURE 1.3 – Exemple de fonctionnement attendu pour un système de Réalité Augmentée dans l'Usine d'Electricité de Metz.

peut subir des modifications, ou que des environnements différents peuvent être ressemblant (*aliasing perceptif*). Nous détaillons dans cette partie certains travaux importants en les reliant, en fonction de leurs apports, aux trois modules principaux des systèmes de reconnaissance visuelle de lieux (cf. figure 1.1).

1.3.1 Traitement de l'information visuelle

Les méthodes d'analyse et de traitement de l'image d'entrée peuvent être regroupées en trois grandes catégories.

Approches locales

Les approches locales visent à extraire l'information présente à une échelle pixellique dans l'image. Pour cela, elles commencent par y détecter des points d'intérêt, puis à leur associer à chacun un descripteur, en s'appuyant sur des méthodes telles que SIFT [Low04], SURF [BTG06], FAST [RD06], BRIEF [CLSF10], ORB [RRKB11], ou encore LIFT [YTLF16]. Dans un souci de passage à l'échelle (des centaines de descripteurs locaux sont extraits de chaque image connue de l'environnement), l'espace des descripteurs est souvent quantifié en un nombre fixe de *mots visuels*, en utilisant par exemple un partitionnement en k-moyennes. Chaque nouvelle image inconnue de l'environnement peut alors être décrite par un descripteur global contenant le nombre d'occurences de chacun des mots visuels du dictionnaire (construit préalablement), ce descripteur pouvant ensuite être comparé à ceux des images de référence afin d'en mesurer les similarités. On parle dans ce cas d'approche par *sacs de mots visuels* [CDF$^+$04, SZ03].

Le principal intérêt des méthodes locales est qu'elles s'appuient sur des indices présentant une bonne invariance aux changements de points de vue et d'échelle, leur assurant elles-mêmes une certaine robustesse vis-à-vis de ces perturbations. Toutefois, les approches par sacs de mots visuels reposent sur l'hypothèse que les occurrences des différents mots dans une image sont indépendantes. Or la présence d'indices répétés - qui est un problème fréquent dans les environnements considérés - entre en contradiction avec cette hypothèse, entrainant une dégradation des performances [JDS09]. De plus, certains indices sont présents dans des lieux différents de l'environnement, perdant ainsi tout pouvoir discriminant et influençant négativement les performances de localisation. Pour contourner ces difficultés, Knopp *et al.* [KSP10] ont proposé d'identifier les régions des images de référence contenant des indices non discriminants, au sens où ces indices sont semblables à des indices présents dans des images de lieux éloignés, puis à les ignorer au moment de construire le dictionnaire. La principale limitation de cette méthode est qu'elle nécessite de connaître la localisation relative des images de référence (fournie par des données GPS par exemple). Dans [TSPO13, TSOP15], Torii *et al.* proposent eux de construire un graphe des indices locaux extraits d'une image, dont les arêtes témoignent à la fois d'une proximité dans l'image et d'une similarité des descripteurs, afin d'identifier les régions contenant des motifs répétitifs. L'importance donnée, dans la construction du descripteur de l'image, aux indices répétés est alors majorée, afin de ne pas perturber la reconnaissance.

Outre les approches par sacs de mots visuels, d'autres méthodes permettent d'agréger les descripteurs locaux pour obtenir un descripteur global de l'image, comme par exemple les vecteurs de Fischer [JH98, PSM10], ou les VLAD (pour *Vector of Locally Aggregated Descriptors*, ou vecteur de descripteurs agrégés localement) [JDSP10].

Si les méthodes utilisant des indices locaux présentent une certaine robustesse aux changements de point de vue, elles sont extrêmement sensibles aux changements de conditions d'acquisition : conditions météorologiques en extérieur (ex. : neige/pluie/soleil), conditions d'illumination

FIGURE 1.4 – Exemples de sous-images extraites puis mises en correspondance automatiquement sur la base de descripteurs CNN : une couleur par correspondance (tirée de [SSJ+15]).

(ex. : jour/nuit, éclairage artificiel, ombres), voire nature du dispositif de prise de vue (utilisation d'appareils photos différents).

Approches globales

L'autre grand type d'approche pour décrire les images arrivant en entrée du système est l'approche globale. Ici, l'image n'est plus décrite à partir de l'ensemble des indices locaux extraits de celle-ci, mais directement dans sa globalité. Un des principaux descripteurs globaux fut Gist [OT01]. L'arrivée des réseaux de neurones convolutifs (CNN, pour *Convolutional Neural Networks*) a permis d'obtenir des descripteurs globaux très performants qui ont pris le pas, ces dernières années, sur les approches locales dans les tâches de reconnaissance [ZYT18]. Ces réseaux peuvent s'inspirer des approches locales comme VLAD en en proposant une version *apprise* [NYD15, AGT+16, AGT+18], ou être simplement des réseaux entraînés à des tâches génériques (ex. : classification) dont une couche cachée est utilisée comme descripteur [KSH12, SZ15, SLJ+15, HZRS16].

Les approches globales présentent l'avantage d'être robustes aux changements de conditions évoqués précédemment. A l'inverse, les descripteurs globaux sont peu invariants aux changements de point de vue, et ces méthodes possèdent donc en résumé les avantages et inconvénients opposés à ceux des approches locales.

Approches mixtes

Les approches mixtes, en proposant un compromis entre les techniques locales et globales, visent à bénéficier simultanément des avantages de ces deux types d'approches [LSN+16]. Dans [SSJ+15], Sünderhauf *et al.* proposent d'associer des descripteurs CNN à des sous-images extraites automatiquement de l'image globale. Ensuite, chaque descripteur issu d'une image est associé à son plus proche voisin parmi les descripteurs issus de l'autre image, et seules les correspondances mutuelles sont conservées. En faisant cela, des sous-parties de chaque image sont mises en correspondance automatiquement, puis les images globales sont comparées sur la base du nombre de correspondances entre sous-images et de la similarité entre les descripteurs associés. La figure 1.4 présente des exemples de telles correspondances semi-globales.

1.3.2 Représentation de l'environnement

Pour reconnaître un lieu, le système a besoin de se référer à une carte de l'environnement. Ici, le mot *carte* est entendu au sens large, et ne désigne pas nécessairement un plan contenant de l'information métrique. Une taxonomie des différents types de carte et de leur niveau d'abstraction est présentée dans [LSN+16] (Table 1).

La forme de carte la plus simple est un ensemble d'images ne contenant pas d'information de position. Il est également possible d'y ajouter de l'information topologique (ex. : les transitions possibles entre différents lieux sont connues), et/ou métriques (ex. : les distances entre et/ou à l'intérieur des lieux sont connues).

Stumm *et al.* ont proposé de représenter l'environnement sous la forme d'un graphe de covisibilité des indices visuels SIFT [SML13, SMLC15, SML16b, SML$^+$16a]. Les noeuds du réseau sont constitués des indices visuels extraits des images (et éventuellement suivis d'une image à l'autre) [SML13, SMLC15], ou des mots visuels du dictionnaire associés à ces indices [SML16b, SML$^+$16a]. Les arêtes du graphe relient les indices/mots visibles dans une même image. Des *positions virtuelles* correspondant à l'image inconnue sont alors retrouvées sous la forme de cliques dans le graphe, puis évaluées dans un cadre Bayésien. Cette méthode, bien qu'obtenant des performances significatives, nécessite d'avoir acquis au préalable une séquence continue d'images de l'environnement pour construire le graphe de covisibilité, ce qui limite sa mise en oeuvre.

D'autres méthodes proposent de modéliser la base d'images de référence sous la forme d'un graphe : graphe représentant la similarité entre images [CS15], graphe de covisibilité d'indices semi-globaux décrits par un descripteur CNN [CCB$^+$17], ou graphe multicouche contenant différentes relations géométriques (proximité, parallélisme, colinéarité, coplanarité) entre différents types d'indices locaux (points d'intérêt, segments de droites, plans) [LSLL12]. Ces différentes méthodes sont principalement limitées par le coût important associé à la recherche d'un sous-graphe correspondant à l'image inconnue, au moment de la reconnaissance.

Dans notre travail, la reconnaissance de lieux est distincte du calcul précis de point de vue : une fois l'opérateur face à un dispositif reconnu, il convient ensuite d'estimer précisément sa position et son orientation afin d'afficher les informations pertinentes dans l'image. Il n'est donc pas nécessaire de reposer sur une représentation continue de l'environnement pour la tâche de reconnaissance de lieux. En conséquence, nous faisons le choix de représenter l'environnement sous la forme d'un simple ensemble d'images de référence, sans information de position. Les images peuvent éventuellement être regroupées en sous-ensembles disjoints correspondant à des lieux clairement distincts (salles). Cette représentation peu compacte de l'environnement pourrait éventuellement nécessiter quelques adaptations lorsque les environnements considérés sont de très grande taille, mais elle s'avère efficace dans le cas des environnements testés au cours de nos expériences (quelques centaines d'images forment typiquement nos bases de référence).

1.3.3 Génération de décision

La reconnaissance de lieux nécessite une prise de décision de la part du système concernant le lieu reconnu dans l'image d'entrée. Une première méthode consiste à entrainer un classifieur (ex. : machine à vecteurs de support, ou *SVM* [BGV92]), pour associer un lieu à chaque image d'entrée. Dans [GOSP13, GSOP16], Gronat *et al.* proposent une variante qui consiste à entrainer un classifieur par lieu, en utilisant pour cela les nombreuses images ne montrant pas ce lieu, plutôt que les quelques images du même lieu. La principale limitation de ces méthodes réside dans le volume d'images nécessaires à l'entrainement du classifieur, et dans sa capacité ou non à se généraliser à des points de vue inconnus.

Dans notre travail, nous ne considérons pas un ensemble discret de lieux, mais souhaitons simplement être en mesure de retrouver, parmi les images de référence, une image montrant le même lieu que l'image d'entrée. Notre problème se résume donc à une tâche de récupération d'image (*image retrieval* en anglais), dans laquelle il s'agit de mesurer une similarité entre l'image inconnue et chacune des images de référence, afin de sélectionner la (ou les) plus proche(s) au sens de cette mesure de similarité.

Mise en correspondance locale

Les images de référence considérées comme similaires à l'image inconnue peuvent être erronées. Pour confirmer ou infirmer cette décision, [KSP10] tente d'estimer une transformation géométrique entre l'image de référence et l'image inconnue à partir de correspondances entre indices locaux. En pratique, des correspondances potentielles entre indices locaux des deux images sont calculées sur la base d'une distance entre leurs descripteurs [Low04], puis la consistance géométrique de ces correspondances est évaluée : si la méthode parvient à estimer un modèle de géométrie épipolaire ou d'homographie (scènes planes) entre les deux images, alors ces dernières ont de grandes chances de représenter le même lieu vu depuis deux points de vue différents.

Le problème de la mise en correspondance géométrique entre deux vues a été étudié de manière extensive au cours des dernières décennies. Cependant, les méthodes basées sur les points d'intérêt fonctionnent dans des environnements bien texturés, mais échouent le plus souvent dans le cas inverse. De plus, le ratio de Lowe, qui demande, pour qu'une correspondance entre deux indices locaux soit retenue, que le rapport des distances au plus proche voisin et au deuxième plus proche voisin soit suffisamment éloigné de 1 [Low04], conduit à ne pas inclure les motifs répétés dans l'ensemble initial de correspondances. Les segments sont très présents en milieux industriels et peuvent apparaître comme des indices appropriés, même si leur mise en correspondance reste un problème très difficile. Plusieurs stratégies de mise en correspondance ont été étudiées par le passé, et la plupart d'entre elles consistent à estimer itérativement un modèle géométrique cohérent avec un ensemble de correspondances supposées (algorithmes de type RANSAC). La mise en correspondance initiale des segments peut se faire en étudiant la similarité de leurs voisinages [ZK13, BFG05, SMM16b]. Cependant, les segments présents dans les images d'environnements industriels correspondent souvent à des arêtes d'objets tridimensionels. Autrement dit, leurs voisinages sont souvent constitués d'une partie pauvre en information (couleur unie par exemple), et d'une autre dont le contenu dépend du point de vue d'observation (fond de scène). Le même argument peut aussi être opposé aux méthodes dans lesquelles les segments sont décrits par les points d'intérêt présents dans leurs voisinages [FWH10, LSLL12]. A l'inverse, il est possible de créer l'ensemble de correspondances potentielles en se basant sur des critères géométriques [KWK09, FWH12, JGF+16], mais ces méthodes centrées sur les invariants restent hautement sensibles au bruit, et se traduisent souvent par une explosion combinatoire. Enfin, la mise en correspondance de groupes de segments peut permettre de compenser le manque d'information associée à chaque segment, mais là encore au prix d'une combinatoire élevée [WNY09, LSFP15].

1.4 Sélection de lieux candidats par approche mixte

Suivant l'exemple de [KSP10], nous avons fait le choix d'une récupération d'image en deux parties. La première partie, détaillée ci-après, est elle-même décomposée en deux sous-parties. Les images sont d'abord décrites par un unique descripteur CNN global, puis les images de référence sont classées selon la similarité entre leur descripteur et celui de l'image inconnue. Les N_2 meilleures images (Top N_2) sont alors conservées, puis reclassées à l'aide d'une seconde mesure de similarité plus robuste, avant de ne conserver que le Top N_1 ($N_1 < N_2$).

Cette deuxième mesure de similarité est largement inspirée du travail de Sünderhauf *et al.* [SSJ+15]. La figure 1.5 présente la méthode de manière schématique. En pratique, N régions d'intérêt sont extraites, après analyse des contours présents dans l'image [ZD14] et sous la forme de boîtes rectangulaires, de chacune des deux images I_1 et I_2 à comparer. A chacune de ces sous-images est ensuite associé un descripteur CNN. Puis, chaque descripteur est mis en correspondance avec son plus proche voisin (PPV) parmi les descripteurs issus de l'autre image, cette

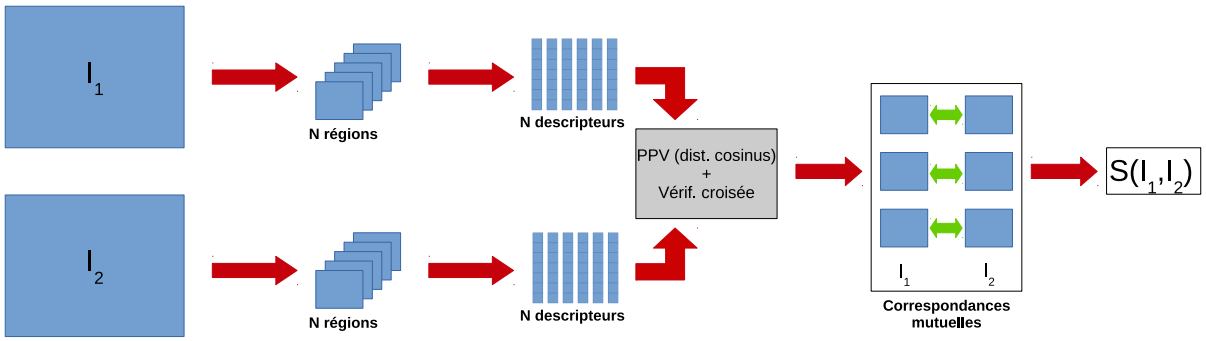


FIGURE 1.5 – Schéma de principe de la mesure de similarité entre images basée sur les similarités entre sous-parties de ces images (inspirée de [SSJ+15]).

similarité étant mesurée par le cosinus de l'angle θ entre les deux descripteurs, disons A et B :

$$cos(\theta) = \frac{A \cdot B}{||A||.||B||} \tag{1.1}$$

Une étape de vérification croisée est ensuite réalisée, c'est-à-dire que seules les correspondances mutuelles entre sous-images sont conservées (lorque le plus proche voisin du plus proche voisin d'un descripteur est ce descripteur lui-même). Puis la similarité entre images est calculée comme la somme des similarités entre les sous-images en correspondance mutuelle, divisée par le nombre de sous-images extraites :

$$S(I_1, I_2) = \frac{1}{N} \sum_i cos(\theta_i) \tag{1.2}$$

où les θ_i sont les angles entre les descripteurs de sous-images en correspondance mutuelle.

Cette seconde mesure se veut plus robuste que la première, mais également un peu plus coûteuse en temps de calcul, et c'est pour cette raison que nous ne l'appliquons qu'entre l'image inconnue et les N_2 meilleures images de référence (au sens du descripteur global). En effet, la similarité entre descripteurs est calculée à l'aide d'un simple produit scalaire. En revanche, le calcul du descripteur se fait par le passage de l'image (ou sous-image) dans un réseau de neurones convolutifs. Si certains réseaux permettent, en un seul passage, d'obtenir le descripteur associé à n'importe quelle sous-partie de l'image [Gir15], d'autres nécessitent autant de passages qu'il y a de sous-images à décrire [KSH12, SZ15]. Il est à noter que le calcul du (ou des) descripteur(s) ne se fait, à l'utilisation, que pour l'image inconnue, puisque ceux associés aux images de référence sont calculés en amont. Au cours de cette deuxième sous-étape, les N_2 images sont reclassées au sens de cette nouvelle mesure de similarité, puis seules les N_1 meilleures sont conservées ($N_1 < N_2$).

Les performances de notre méthode de mise en correspondance automatique de régions sont illustrées en figure 1.6. Elle montre que notre méthode permet de générer des correspondances semi-globales entre images, même dans des environnements complexes et en présence de forts changements de point de vue.

La figure 1.7 illustre finalement la plus grande robustesse aux changements de point de vue de la mesure de similarité basée boîtes (1.7c) sur la mesure de similarité globale (1.7b).

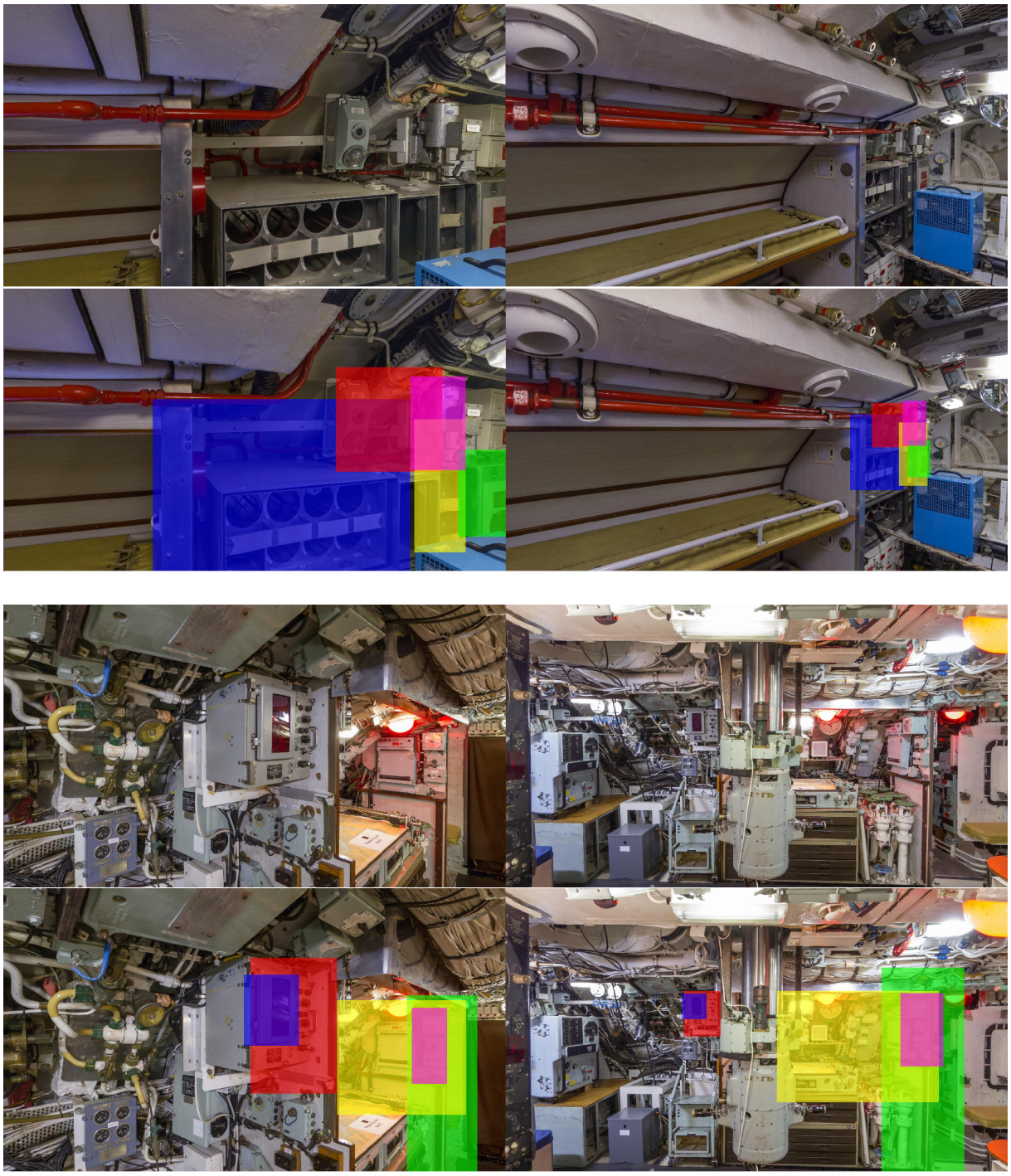


FIGURE 1.6 – Exemples de régions mises en correspondance automatiquement par notre méthode : une couleur par correspondance (images du sous-marin HMS Ocelot).

(a) Image inconnue. (b) Meilleure image (simil. $n°1$). (c) Meilleure image (simil. $n°2$).

FIGURE 1.7 – Illustration du comportement des deux mesures de similarité utilisées dans notre méthode, en présence d'un fort changement de point de vue. La mesure de similarité globale ($n°1$) capture la structure générale de l'image, tandis que la mesure de similarité basée boîtes ($n°2$) permet de tenir compte des régions communes aux deux images.

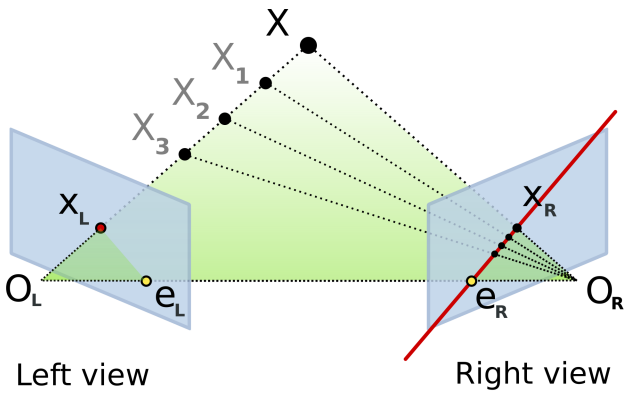


FIGURE 1.8 – Représentation schématique de la géométrie épipolaire[2].

1.5 Reclassement des lieux candidats par mise en correspondance locale sous contrainte géométrique

Une fois le Top N_1 des images de référence constitué, nous avons mis en place une étape de vérification géométrique de la pertinence des images retenues. En pratique, nous estimons une transformation géométrique directement à partir de correspondances entre indices locaux, sans avoir besoin d'un modèle de la scène, ni de connaissances a priori sur les paramètres des caméras. L'estimation d'une transformation géométrique liant le contenu des deux images permet de conclure sur la pertinence de l'image récupérée.

1.5.1 Géométries de correspondance

Dans ce travail, nous nous intéressons à deux types de géométries de correspondance : la géométrie épipolaire, qui ne nécessite aucune hypothèse sur les caméras ni sur l'architecture de la scène, et l'homographie, dans le cas où la scène considérée est plane.

Géométrie épipolaire

Si les images montrent bien le même lieu depuis deux points de vue différents, alors il existe un modèle de géométrie épipolaire liant les pixels des deux images. La figure 1.8 illustre son

2. Image issue de Wikipédia.

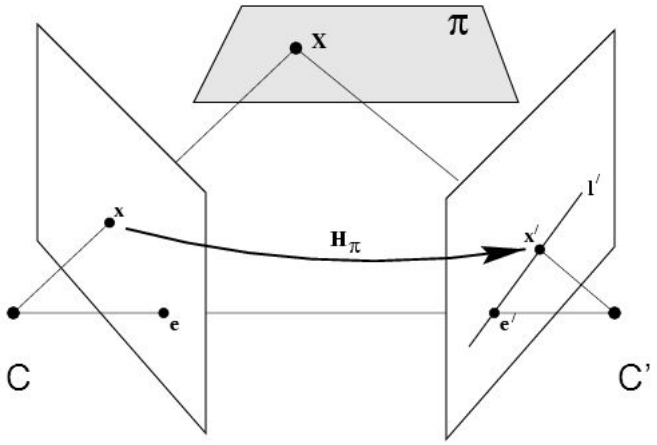


FIGURE 1.9 – Représentation schématique de la géométrie plane.

principe : considérant deux caméras de centres respectifs O_L et O_R, puis considérant un point x_L dans la première image, il est possible de reconstruire le rayon 3D auquel appartient le point X dont x_L est la projection. Ce rayon 3D se projette ensuite dans la deuxième image sous la forme d'une droite 2D, à laquelle appartient nécessairement la projection x_R de X. Connaissant x_L et la géométrie épipolaire liant les deux images, il est donc possible de construire la droite à laquelle appartient x_R, sans avoir besoin de connaître la position du point X sur le rayon issu de x_L (X, ou X_1, X_2, X_3,...). Le lien entre les coordonnées du point x_L et les coordonnées du point x_R est donné par la *matrice fondamentale* F (de taille 3×3 : les points sont représentés par leurs coordonnées homogènes) :

$$x_R^\top F x_L = 0 \tag{1.3}$$

En pratique, connaissant au moins huit correspondances entre points des deux images, il est possible de calculer la matrice fondamentale [MMM16].

Géométrie plane (homographie)

La géométrie épipolaire ne nécessite donc aucune connaissance a priori sur la scène pour être calculée. Sous l'hypothèse que la scène est plane (ou qu'une partie d'elle l'est), il existe alors également une homographie liant les projections, dans les deux images, des points de ce plan (voir figure 1.9). Considérant un plan 3D π et un point X appartenant à ce plan, alors les projections x et x' de X dans les deux images sont liées par une homographie H_π, et cette homographie est la même pour tous les points X du plan :

$$x' = H_\pi x \tag{1.4}$$

Là encore, H_π est de taille 3×3, et les points sont écrits en coordonnées homogènes. En pratique, H_π peut être calculée à partir de quatre correspondances [MMM12].

Estimation du modèle (RANSAC)

Pour estimer un modèle de géométrie épipolaire ou d'homographie entre deux images, la plupart des méthodes existantes s'appuie sur des correspondances potentielles entre indices locaux des deux images (ex. : points d'intérêt, segments de droite). Or ces correspondances potentielles,

déterminées sur la base d'une similarité entre descripteurs, ne sont pas toutes correctes. Pour séparer les correspondances aberrantes (*outliers*) des correspondances pertinentes (*inliers*), et ainsi estimer précisément la transformation géométrique entre les deux images, un algorithme robuste de type RANSAC [FB81] est presque toujours utilisé.

Cet algorithme itératif consiste à choisir au hasard un nombre de correspondances potentielles égal au nombre minimal de correspondances nécessaires au calcul des paramètres de la transformation (8 pour la matrice fondamentale, 4 pour une homographie), puis à estimer un modèle géométrique à partir de ces correspondances, et enfin à compter le nombre de correspondances potentielles en accord avec cette transformation (ensemble de consensus). Après un certain nombre d'itérations, le modèle qui obtient le plus grand ensemble de consensus est considéré comme le modèle correct.

Le nombre N_{iter} d'itérations nécessaires pour assurer, avec une probabilité p, qu'au moins un ensemble constitué uniquement d'inliers a été tiré au sort dépend du nombre minimal de correspondances nécessaires pour estimer un modèle. Pour estimer une homographie par exemple, N_{iter} est donné par la formule :

$$N_{iter} = \frac{log(1-p)}{log(1-w^4)} \tag{1.5}$$

où w est le taux supposé de correspondances pertinentes parmi l'ensemble des correspondances potentielles.

1.5.2 Estimation des géométries planes et épipolaire

Dans notre travail, nous tirons profit de plusieurs propriétés globales des environnements industriels pour guider la mise en correspondance :

— Premièrement, les environnements industriels sont souvent constitués d'un nombre important de plans verticaux. Partant du constat que les informations à propos des plans existant dans une image sont d'une certaine manière contenues dans les points de fuite, la manière dont notre méthode tire profit de ces derniers est double. Tout d'abord, nous n'utilisons que les segments associés à des points de fuite, et ces associations sont ensuite utilisées pour contraindre la mise en correspondance des segments. Ensuite, les points de fuite sont utilisés comme primitives pour estimer les homographies locales, ce qui permet d'une part de réduire la combinatoire, et d'autre part de limiter le recours aux points d'intérêt visuels.

— Deuxièmement, nous utilisons des correspondances semi-globales pour guider la mise en correspondance des indices locaux. En pratique, les correspondances entre régions sont utilisées comme hypothèses de correspondances entre plans verticaux, entre lesquels nous tentons d'estimer des homographies. De cette façon, la mise en correspondance des segments est davantage contrainte, ce qui permet de contourner les difficultés inhérentes aux environnements industriels sans compromettre le temps de calcul.

Etant données deux vues différentes d'une même scène, notre algorithme de traitement se décompose de la manière suivante : (1) Des correspondances entre régions des deux images sont calculées. (2) Les homographies locales qui peuvent exister entre les régions mises en correspondance sont détectées, puis fusionnées pour identifier les plans verticaux présents. (3) Les correspondances entre segments générées par les homographies sont utilisées pour améliorer l'estimation de la géométrie épipolaire. Le principe de la méthode est présenté en figure 1.10.

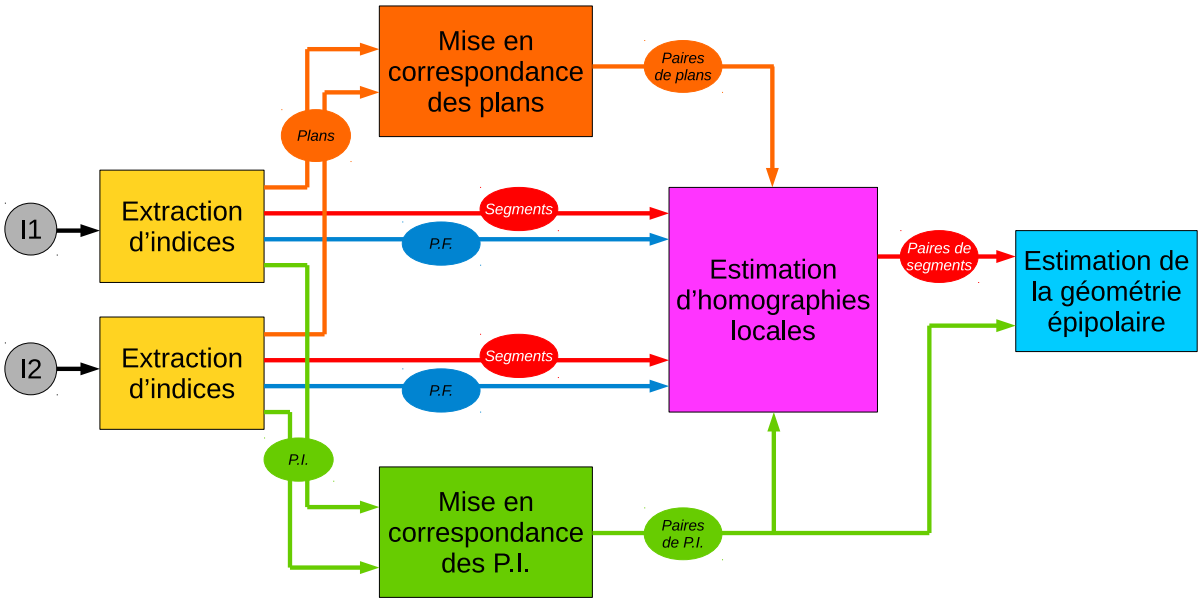


FIGURE 1.10 – Représentation schématique de notre méthode de mise en correspondance robuste des indices locaux.

Extraction et mise en correspondance des régions

Le première étape de notre méthode consiste à générer des hypothèses de correspondances entre régions, dans le but de calculer par la suite des homographies locales. Pour cela, nous réutilisons les correspondances semi-globales déterminées lors du calcul de la deuxième mesure de similarité (voir partie 1.4). La méthode *Edge Boxes* [ZD14], qui détecte des accumulations de contours, est utilisée pour générer les régions d'intérêt (que nous considérons ensuite comme hypothèses de plans verticaux) car elle apparaît particulièrement adaptée aux images d'environnements industriels.

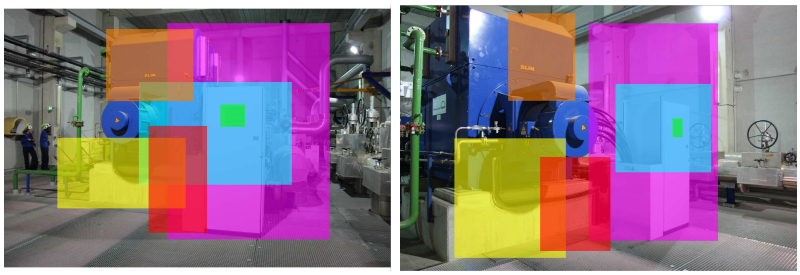
Estimation des homographies locales

Une fois les sous-images mises en correspondance, notre méthode a pour objectif de détecter les homographies locales entre régions en correspondance. Pour assurer des estimations efficaces, nous avons développé un algorithme de type RANSAC dans lequel les hypothèses de modèle sont d'abord générées à partir des points d'intérêt visuels et des points de fuite, avant que ces mêmes hypothèses ne soit validées sur les points d'intérêt et les segments. Ce schéma nous permet de contourner les difficultés relatives aux images peu texturées (faible densité de points d'intérêt visuels et difficile mise en correspondance des segments), tout en tirant profit de l'abondance de segments et de points de fuite inhérente aux environnements industriels. Notre méthode est illustrée en figure 1.11.

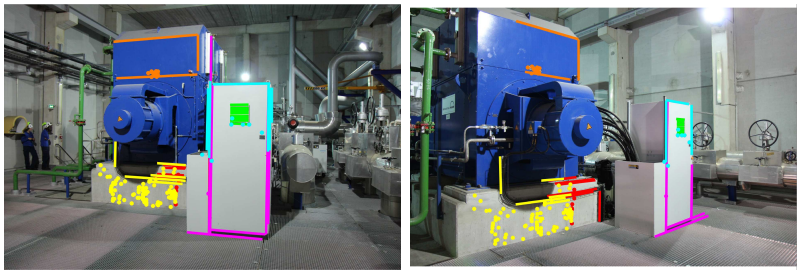
Les points de fuite (PF) sont détectés dans les deux images en utilisant la méthode décrite dans [SFB16]. Cette méthode détecte le zénith (point de fuite vertical) ainsi que les points de fuite horizontaux présents dans les images, chacun de ces derniers étant donc associé à un plan vertical. Les zéniths sont ensuite directement mis en correspondance entre les deux images.

Les segments sont ensuite extraits par LSD [vGJMR12]. Les segments associés aux PF préala-

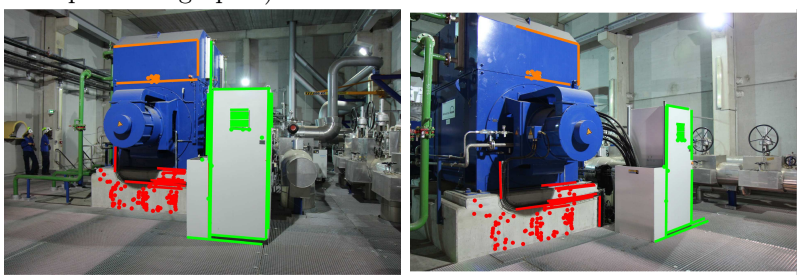
(a) Deux vues de l'UEM.



(b) Exemple de régions mises en correspondance.



(c) Correspondances pertinentes (points et segments) des homographies estimées entre régions en correspondance (une couleur par homographie).



(d) Correspondances pertinentes après fusion des homographies.

(e) Trois premiers plans obtenus avec un algorithme multi-RANSAC 4 points.

FIGURE 1.11 – Vue d'ensemble de notre méthode d'estimation des homographies locales.

blement extraits sont conservés, tandis que les autres sont abandonnés. L'association segment-PF se fait en considérant un seuil sur l'angle entre la droite à laquelle le segment appartient et la droite reliant le milieu du segment au PF. Ainsi, le segment est associé au PF si et seulement si cet angle est inférieur au seuil considéré.

Les points d'intérêt (PI) sont enfin détectés en utilisant une méthode appelée LIFT [YTLF16], qui implémente sous forme de réseaux de neurones tout le pipeline d'extraction, d'estimation d'orientation, et de description des points d'intérêt, et qui présente des performances supérieures à SIFT [Low04] dans les tests que nous avons menés sur des images d'environnements industriels. Les correspondances entre PI des deux images sont ensuite obtenues en appliquant le ratio de Lowe dans les deux sens, puis en ne gardant que les correspondances mutuelles, afin de maximiser le taux d'inliers au sein de l'ensemble de correspondances initial E_0.

En raison de la spécificité des environnements considérés, les régions extraites puis mises en correspondance lors de la première étape peuvent être considérées comme des hypothèses de plans verticaux. Cependant, il existe de nombreux recoupements entre les différentes boîtes englobantes d'une même image, donc tester l'ensemble des paires de boîtes pour y trouver des homographies ferait exploser inutilement le temps de calcul. Pour remédier à cela, les paires de boîtes sont d'abord classées grossièrement des plus petites vers les plus grandes, en considérant comme critère pour chaque paire l'aire moyenne des deux boîtes. L'idée principale est ensuite de traiter les paires de boîtes dans cet ordre, tout en tenant à jour deux cartes des pixels déjà visités (une par image). Ces cartes ont la même résolution que les images originales, et chaque pixel y prend la valeur 1 si une boîte le contenant a déjà été testée, 0 sinon. Une paire de boîtes est testée si et seulement si moins de 50% des pixels de chaque boîte ont déjà été visités.

Le traitement d'une paire de régions consiste en premier lieu à sélectionner les indices contenus dans chaque région (PI, segments, et PF associés aux segments), puis à appliquer une étape de fusion des segments dans le but de corriger la sur-segmentation induite par LSD. En pratique, les segments à fusionner sont déterminés en utilisant des contraintes tangentielle et normale similaires à celles présentées dans [JGF+16]. Plus précisément, les segments associés au même PF sont comparés deux à deux, et les segments qui vérifient ces deux contraintes, i.e. qui sont suffisamment alignés, sont fusionnés. Finalement, seuls les segments les plus longs (i.e. dont la longueur est supérieure à un certain pourcentage du plus grand segment associé au même PF) sont conservés, ce qui permet de ne garder que les segments les plus significatifs et de réduire la combinatoire associée à leur mise en correspondance.

En raison des difficultés liées à la mise en correspondance des segments (voir figure 1.12), les hypothèses de RANSAC sont générées uniquement à partir des PI et des PF. A chaque itération, nous tirons au hasard une paire de PF horizontaux, deux paires de PI parmi E_0, et la paire de zéniths. Les 4 paires de points définissent ainsi une homographie induite par un plan vertical. Contrairement à la méthode présentée dans [SMM16b], les PI sont ici directement utilisés comme primitives pour la définition des modèles. Ce choix nous permet de réduire drastiquement la combinatoire ainsi que le recours aux PI visuels, en limitant à simplement deux le nombre de correspondances entre PI nécessaires pour définir une homographie.

Pour déterminer le nombre d'itérations de RANSAC, nous utilisons une version modifiée de l'algorithme adaptatif présenté dans [HZ04] (Section 4.7.1). Etant donné N_{total}^{PFh} le nombre de paires possibles entre PF horizontaux des deux images, la probabilité de tirer la paire correcte à n'importe quelle itération est : $w_{PFh} = 1/N_{total}^{PFh}$. Etant donné N_{total}^{p} le nombre de correspondances entre PI contenues dans E_0, la probabilité de tirer une paire correcte est $w_p = N_{inliers}^{p}/N_{total}^{p}$, avec $N_{inliers}^{p}$ le nombre de paires pertinentes contenues dans E_0 à l'ité-

FIGURE 1.12 – Illustration de la difficulté à mettre en correspondance des segments sur la base de leurs descripteurs, en environnement industriel (segments extraits par [SMM16a], puis mis en correspondance par [ZK13]).

ration courante. Le nombre N_{iter} d'itérations nécessaires pour assurer, avec une probabilité p, qu'au moins un ensemble de quatre primitives correctes a été tiré au sort est alors donné par la formule :

$$N_{iter} = \frac{log(1-p)}{log(1 - w_{PFh}.w_p^2)} \tag{1.6}$$

A chaque itération, N_{iter} est mis à jour à partir de la valeur courante de w_p, et l'algorithme s'arrête si l'indice de l'itération courante est plus grand que N_{iter}. Nous appliquons également un seuil maximal au nombre d'itérations, qui est égal au nombre de combinaisons différentes qu'il est possible de tirer. Cela permet de stopper prématurément la recherche d'homographie dans le cas où il n'en existe pas.

Si on suppose une configuration avec deux PF horizontaux par image, 40% d'inliers points, et 1% d'inliers segments (nous n'utilisons pas de contrainte forte pour réduire les correspondances possibles entre segments), notre méthode nécessite 113 itérations, tandis que la méthode 4 points (voir formule 1.5) en nécessite 178, et la méthode 4 segments 4.6×10^8. Si la paire de régions considérée est incorrecte, le gain de performance entre notre méthode et la méthode 4 points est beaucoup plus important, du fait de la plus faible combinatoire.

La validation est basée sur les PI et les segments. Les inliers PI sont déterminés de manière classique, tandis que toutes les paires de segments possibles (entre les segments associés aux zéniths d'une part, et entre les segments associés aux deux PF horizontaux tirés au sort d'autre part) sont testées. Les paires segment transféré/segment qui satisfont à la fois les contraintes tangentielle et normale présentées dans [JGF+16] sont retenues. Pour éviter les configurations dégénérées, un segment de l'image originale ne peut être mis en correspondance qu'avec un seul segment de l'image cible (le plus proche au sens de la contrainte normale). A partir de là, le nombre de segments de l'image cible impliqués dans des paires qui satisfont au modèle d'homographie calculé ($N^s_{inliers}$) est ajouté au nombre de PI inliers, définissant ainsi un score pour le modèle H : $Score(H) = N^p_{inliers} + N^s_{inliers}$. Comme seuls les segments les plus significatifs ont été conservés, les correspondances entre segments ont de grandes chances d'être de même importance. De plus, le fait d'ajouter les contributions des points et des segments est justifié par le fait que cela permet de se reposer sur n'importe lequel de ces indices lorsque l'autre se fait rare.

Les sous-images mises en correspondance pendant la première étape peuvent ne pas être liées par une homographie, soit parce qu'elles ne contiennent pas d'objets plans, soit parce que la correspondance est incorrecte. Ainsi, pour limiter la présence de fausses homographies, on ne considère que les homographies qui possèdent plus de 10 correspondances pertinentes.

A ce stade, plusieurs homographies locales peuvent lier différentes parties d'un même plan physique. On applique donc une étape de fusion qui consiste, pour chaque homographie, à tester les correspondances pertinentes des autres homographies. Si plus de 50% des correspondances entre PI pertinentes selon une homographie A sont aussi pertinentes selon une homographie B, et si la même performance est atteinte pour les segments, alors A et B sont fusionnées.

Estimation de la géométrie épipolaire

Les paires de segments contenues dans les ensembles de consensus des homographies calculées à l'étape précédente peuvent désormais être utilisées pour améliorer l'estimation de la géométrie épipolaire. L'idée principale, inspirée par [BFG05], est d'ajouter les intersections de segments à l'ensemble initial de correspondances entre PI E_0, puis d'utiliser ce nouvel ensemble de correspondances en entrée d'un algorithme d'estimation de la géométrie épipolaire [MMM16], que nous

appellerons dans la suite ORSA. Cette méthode a la particularité de s'appuyer sur une approche a contrario pour distinguer les inliers des outliers, ce qui lui permet d'être fiable même en cas de fort taux d'outliers parmi les correspondances potentielles.

Dans notre méthode, les segments sont d'abord convertis en droites dans chacune des images. Pour chaque homographie trouvée à l'étape précédente, les intersections entre droites verticales (celles générées par les segments associés au zénith) et droites horizontales (celles générées par les segments associés au point de fuite horizontal) sont calculées. Ensuite, comme plusieurs segments peuvent générer approximativement la même droite (à cause de l'étape de fusion des homographies), et comme des correspondances quasi-identiques répétées peuvent perturber le bon fonctionnement de l'algorithme d'estimation de la géométrie épipolaire, nous avons fait le choix de diviser les images selon une grille régulière, et de ne garder au plus qu'un point d'intersection par case de la grille. Finalement, ces nouvelles correspondances entre points d'intersection (déduites des correspondances entre segments) sont ajoutées à l'ensemble initial de correspondances entre PI E_0.

La figure 1.13 illustre l'apport de notre méthode en termes de nombre de correspondances compatibles avec le modèle de géométrie épipolaire estimé, et en termes de répartition des inliers dans l'image, ce qui permet d'assurer une plus grande précision du modèle.

1.6 Résultats expérimentaux

Les différentes étapes de la méthode décrite précédemment ont été évaluées, et les résultats sont présentés ci-dessous. Les résultats correspondant à la méthode décrite dans la partie 1.4 sont présentés dans la partie 1.6.1, et ceux correspondant à la méthode décrite dans la partie 1.5 le sont dans la partie 1.6.2.

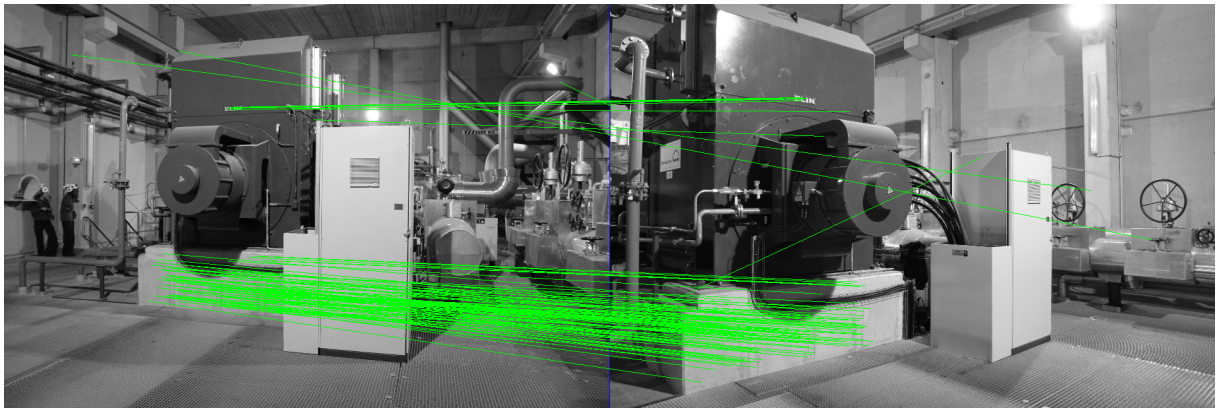
1.6.1 Etape 1 : Sélection des lieux candidats par approche mixte

La figure 1.14 compare les performances de nos deux mesures de similarités, basée sur un descripteur global pour la première et basée sur les similarités entre boîtes pour la seconde, sur une base d'images du sous-marin HMS Ocelot [3]. La base est constituée de 533 images de référence, divisées en 10 salles définies manuellement, et de 103 images de test, elles aussi réparties dans les 10 salles. Pour différentes valeurs de N, les courbes montrent le pourcentage d'images de test pour lesquelles au moins une image correcte est présente dans le Top N des images de référence récupérées (*Précision*). La figure montre que la mesure de similarité basée boîtes permet d'atteindre une précision de reconnaissance plus importante que la mesure basée sur les descripteurs globaux, quel que soit le descripteur CNN utilisé : CaffeNet [JSD+14], ou VGGNet [SZ15].

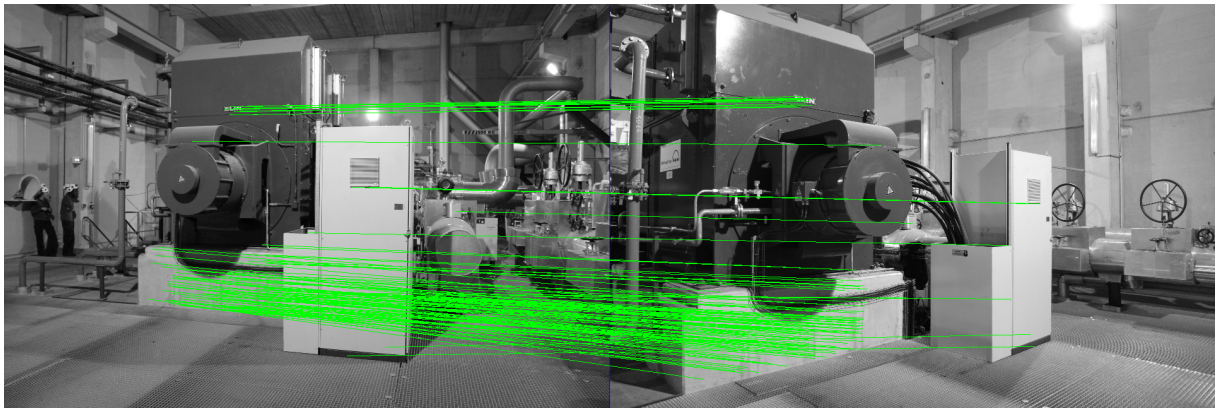
La figure 1.15 montre un exemple des 5 meilleures images de référence récupérées après application de notre méthode de sélection des lieux candidats ($N_2 = 20$, $N_1 = 5$), sur des images de l'Usine d'Electricité de Metz. La base de référence est constituée de 216 images, et la base de test de 68 images. Plus d'exemples de résultats sont présentés en Annexe A.

Sur les 68 images de test, la meilleure image retrouvée (parmi les 216 images de référence) est correcte dans 91% des cas (62 images/68). Le Top 5 des images retrouvées contient au moins une image correcte dans 99% des cas (67/68). Une seule image de test n'a pas de résultat positif parmi les 5 meilleures images retrouvées (voir exemple en bas de la figure 1.15).

3. Images issues de la visite Google Street View du sous-marin HMS Ocelot.



(a) Inliers à partir de correspondances potentielles SIFT [Low04].



(b) Inliers à partir de correspondances potentielles LIFT [YTLF16].

(c) Inliers à partir de correspondances potentielles obtenues par notre méthode.

FIGURE 1.13 – Illustration de l'apport de notre méthode de mise en correspondance locale sous contrainte géométrique : elle permet d'obtenir un modèle de géométrie épipolaire compatible avec des correspondances plus nombreuses et mieux réparties dans l'image.

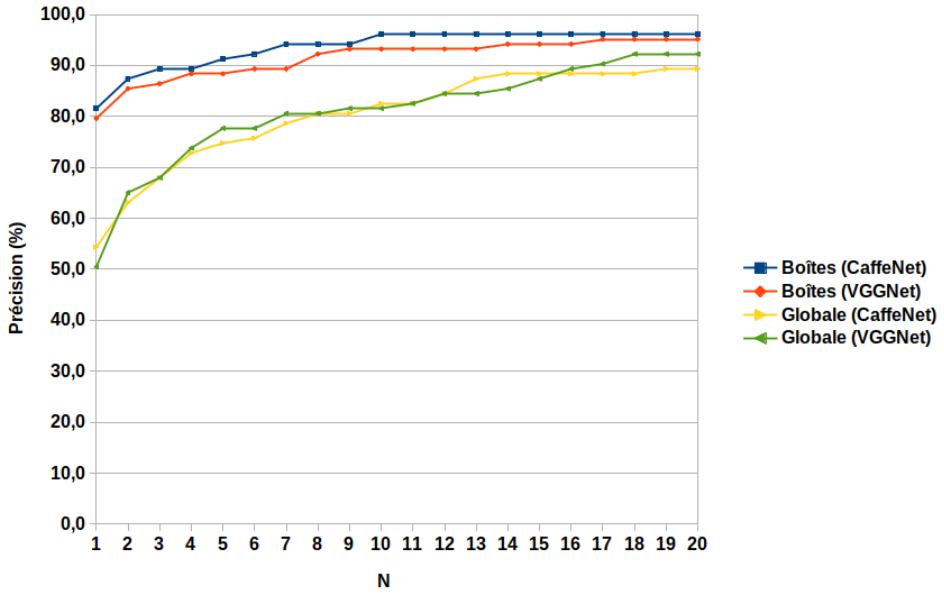


FIGURE 1.14 – Comparaison des performances de récupération d'image de nos deux mesures de similarités, en fonction de deux descripteurs CNN différents.

1.6.2 Etape 2 : Reclassement des lieux candidats par mise en correspondance locale sous contrainte géométrique

Les expériences que nous avons menées l'ont été sur deux ensembles d'images. Le premier est constitué de 46 paires d'images prises dans l'Usine d'Electricité de Metz (UEM), tandis que le second est formé de 14 paires d'images d'environnements urbains tirées de la base présentée dans [ZWA+16] (ces environnements présentent des caractéristiques communes avec les environnements industriels : présence de motifs répétés et de plans verticaux notamment). Pour chaque paire d'images, la vérité terrain est constituée d'une vingtaine de paires de points placés à la main. Ces points ont été sélectionnés avec l'objectif de couvrir chaque image le plus largement possible, tout en assurant une certaine homogénéité de leur distribution. Les images industrielles ont une résolution de 1280×1920 pixels, tandis que les images urbaines ont une résolution de 640×640. Il convient de noter que la méthode présentée dans [SMM16b] échoue à détecter des correspondances correctes entre segments dans la plupart des exemples considérés. Or ces correspondances sont nécessaires pour calculer la rotation entre les deux caméras. En leur absence, la méthode repose alors uniquement sur les correspondances entre PI pour estimer la matrice essentielle, ce qui n'est pas pertinent dans les environnements considérés ici.

Dans la suite, notre méthode est comparée à la méthode classique qui consiste à (i) extraire et mettre en correspondance les PI des deux images en appliquant le ratio de Lowe dans les deux sens avant de ne garder que les correspondances mutuelles (ii) calculer la géométrie épipolaire à partir de ces correspondances en utilisant ORSA. SIFT et LIFT ont été utilisées comme méthode d'extraction de PI dans nos comparaisons. Les géométries épipolaires estimées ont finalement été comparées en termes d'inliers (nombre et taux), et de précision (erreur vis-à-vis de la vérité terrain).

Le précision des matrices fondamentales estimées est évaluée en Fig. 1.16. Pour chaque paire d'images, l'écart entre les points de la vérité terrain et les droites épipolaires estimées a été mesuré par l'intermédiaire de l'erreur quadratique moyenne (RMSE) et de l'erreur maximale.

FIGURE 1.15 – Trois exemples d'images de test (sur fond bleu) accompagnées des 5 meilleures images récupérées après application de notre méthode de sélection des lieux candidats (la meilleure est au milieu de la première ligne, puis de gauche à droite et de haut en bas, jusqu'à la cinquième à droite de la deuxième ligne). Les images jugées correctes sont sur fond vert, celles incorrectes sur fond rouge.

	Nb d'inliers moyen		Taux d'inliers moyen (%)	
	indus.	urban	indus.	urban
SIFT	198.6	102.8	30.02	52.3
LIFT	207.4	100.3	42.24	58.4
Nous	**260.7**	**126.0**	**45.84**	**62.44**

TABLE 1.1 – Performances moyennes de différentes méthodes d'estimation de la géométrie épipolaire, en termes de nombre d'inliers et de taux d'inliers.

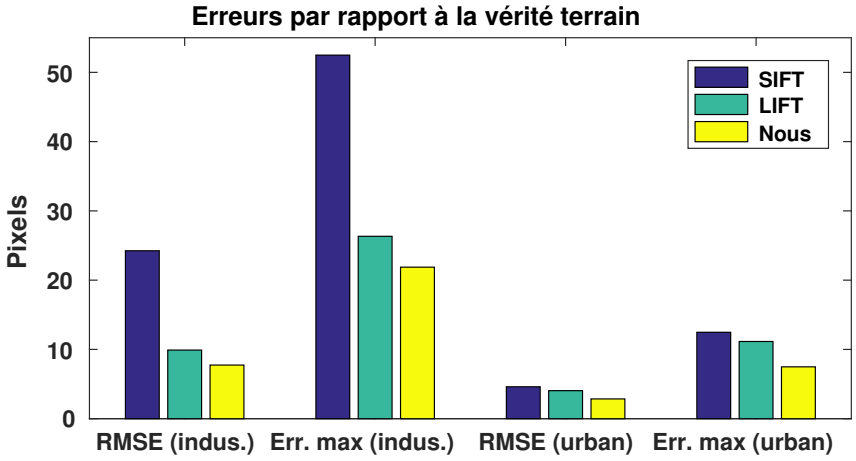


FIGURE 1.16 – Erreurs moyennes de différentes méthodes d'estimation de la géométrie épipolaire sur les bases d'images industrielles et urbaines (désignées par *indus.* et *urban*), par rapport à la vérité terrain.

Ces mesures ont ensuite été moyennées sur l'ensemble des images de chaque base de données. En moyenne, notre méthode présente la plus grande précision, quel que soit le critère d'erreur considéré.

Le tableau 1.1 présente la qualité des modèles estimés en termes d'inliers (ici encore, les résultats ont été moyennés sur les ensembles d'images). Notre méthode présente le plus grand nombre d'inliers, ainsi que le meilleur taux d'inliers, ce qui peut permettre d'améliorer la qualité des résultats d'une éventuelle étape suivante de reconstruction ou de calcul de pose. La Fig. 1.17 illustre le plus grand nombre d'inliers obtenu grâce à notre méthode. Par ailleurs, il convient de noter qu'il existe 6 paires d'images industrielles pour lesquelles ORSA échoue à estimer un modèle à partir des correspondances entre PI SIFT, alors que cette même méthode y parvient avec nos correspondances.

La Fig. 1.11 illustre la capacité de notre méthode à détecter et mettre en correspondance les plans verticaux présents dans les images (figure 1.11d), tandis qu'une méthode multi-RANSAC basée sur les correspondances entre PI ne parvient pas à séparer les différents plans physiques (cf inliers orange et verts), ni à en saisir les contours (cf inliers rouges et verts) (figure 1.11e). La Fig. 1.18 montre d'autres exemples de plans détectés par notre méthode (enveloppes convexes des inliers points et segments) dans des images de test d'environnements industriel et urbain.

Ces expériences montrent que notre méthode permet d'améliorer l'estimation de la géométrie épipolaire dans des environnements plans par morceaux complexes, et ce même en l'absence d'un contenu visuel riche.

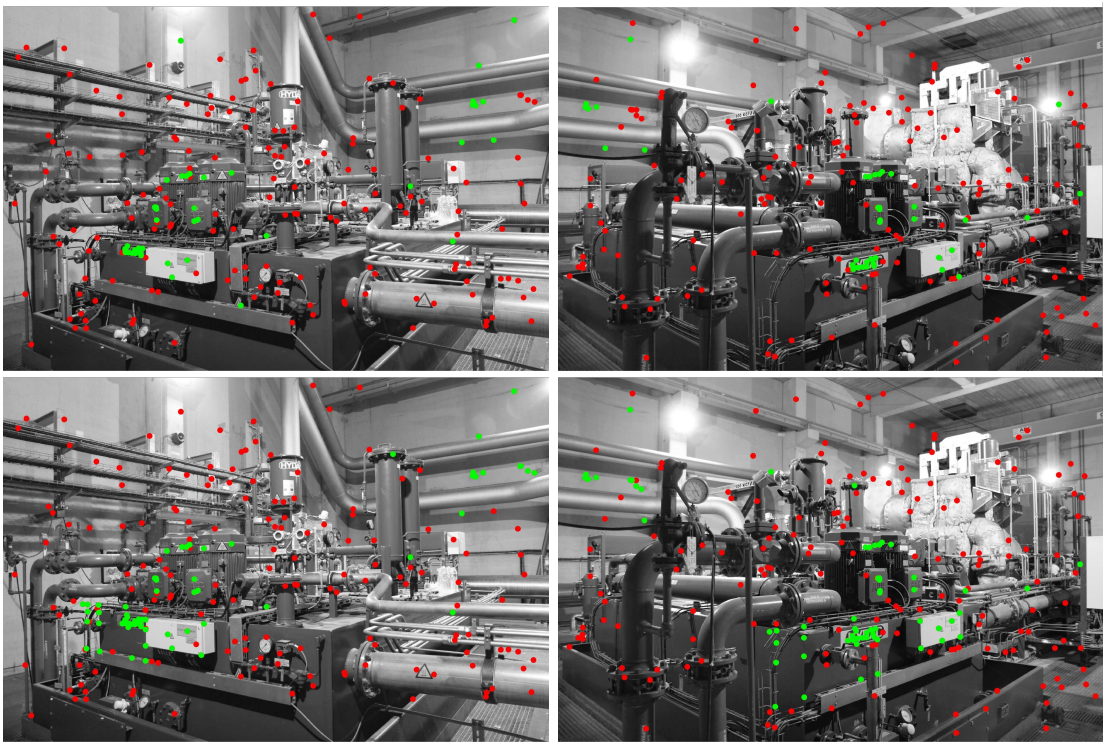


FIGURE 1.17 – Exemple d'images de test pour lesquelles les inliers définis par ORSA (en vert) sont plus nombreux parmi nos correspondances (Ligne 2) que parmi les correspondances entre LIFT (Ligne 1).

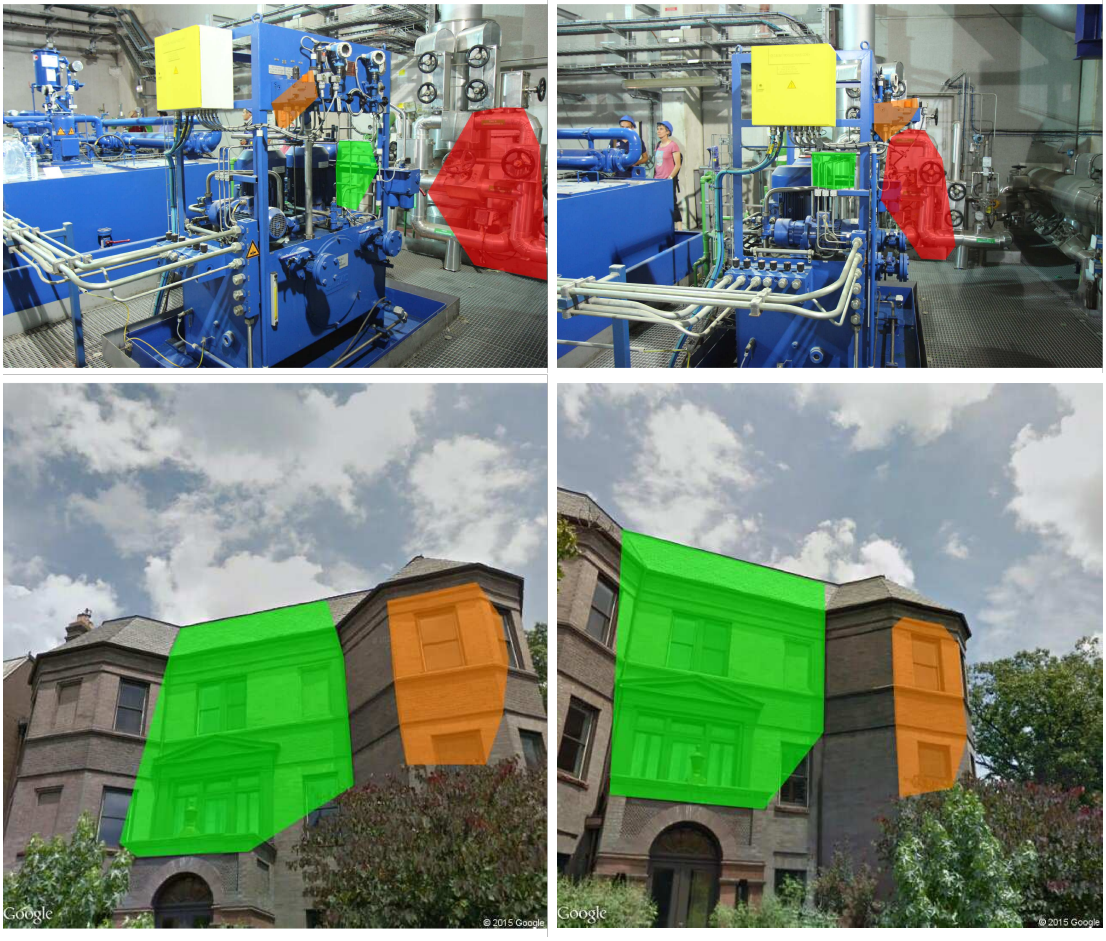


FIGURE 1.18 – Enveloppes convexes des inliers points et segments des homographies estimées par notre méthode, après l'étape de fusion, pour deux paires d'images de test (une couleur par homographie).

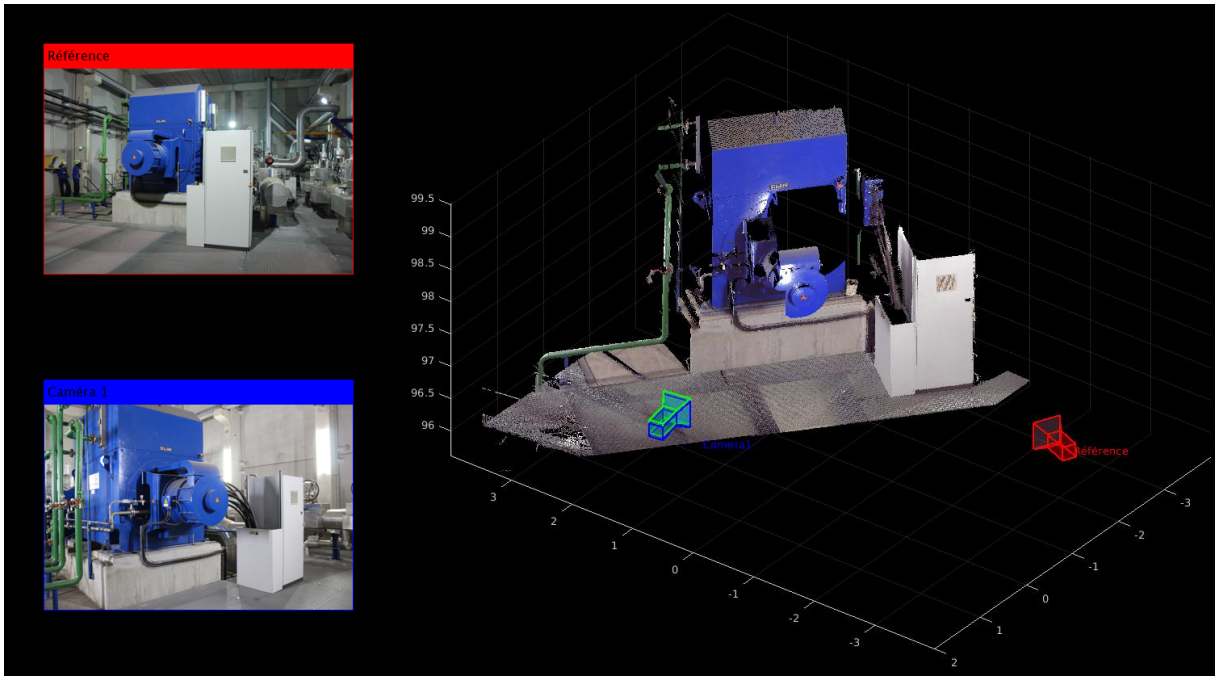


FIGURE 1.19 – Illustration du calcul de pose de caméra à partir d'indices locaux, par PnP [HR11]. L'image de référence et la caméra associée sont représentées en rouge, tandis que l'image inconnue et la caméra estimée à partir de nos correspondances locales sont représentées en bleu. La vérité terrain pour la caméra inconnue est présentée en vert.

1.7 Conclusion

Dans ce chapitre, nous avons présenté une méthode de reconnaissance de lieux sous la forme d'une méthode de récupération d'images en deux parties. Dans la première, deux mesures de similarité différentes - la première plus rapide mais moins fiable, et la deuxième moins rapide mais plus précise - permettent de classer les images de référence, potentiellement très nombreuses, de la plus ressemblante à l'image de test, à la moins ressemblante. Dans la deuxième partie, une méthode d'estimation d'un modèle de géométrie épipolaire entre l'image inconnue et les meilleures images de référence permet de décider définitivement de la pertinence des images récupérées, tout en mettant en correspondance localement les deux images. Cette méthode a été pensée pour répondre aux défis que représentent les environnements industriels en termes d'analyse et de traitement des images.

Notre méthode de mise en correspondance locale permet d'atteindre une certaine robustesse, notamment vis-à-vis du manque de texture dans les images. Cependant, et même si la mise en correspondance d'indices locaux peut être vue comme un préalable au calcul de pose de caméra (voir figure 1.19), nous nous intéressons plutôt dans la suite de ce mémoire à des cas dans lesquels l'utilisation d'indices locaux est encore plus difficile, voire impossible. Dans ces cas extrêmes, l'exploitation d'indices de plus haut niveau (objets), porteurs d'une information sémantique importante, devient nécessaire.

Chapitre 2

Estimation de pose de caméra à partir d'objets : état de l'art et choix de modélisation

Sommaire

2.1 Estimation de pose de caméra à partir d'objets : état de l'art . . . 36
2.1.1 Approches basées sur des correspondances entre points 36
2.1.2 Approches basées sur des modèles d'objets 38
2.2 Estimation de pose de caméra à partir de correspondances conique - quadrique 40
2.2.1 Rappels sur les ellipses et ellipsoïdes 40
2.2.2 Réprésentation des ellipses et ellipsoïdes en coordonnées homogènes . . 41
2.2.3 Equation de projection d'une quadrique 43
2.2.4 Linéarisation de l'équation 44
2.2.5 Détermination de la matrice de projection de la caméra 45
2.2.6 Limites de la méthode 46
2.3 L'Equation d'Alignement des Cônes 46
2.3.1 Rappels sur les cônes du second degré 46
2.3.2 Le cône de projection 47
2.3.3 Le cône de rétroprojection 48
2.3.4 L'Equation d'Alignement des Cônes 49
2.3.5 Signification des termes de l'équation 49
2.3.6 Lien avec un problème aux valeurs propres généralisé 50
2.4 Conclusion 50

Une fois l'utilisateur grossièrement localisé dans son environnement, notre système doit être en mesure de calculer sa position et son orientation par rapport à ce même environnement, pour ensuite intégrer dans l'image des informations pertinentes. Nous avons vu au chapitre précédent une méthode de mise en correspondance locale d'images qui peut permettre un calcul précis de pose de caméra, à condition de posséder un modèle 3D dense de l'environnement. Si cette méthode permet de rendre plus robuste la mise en correspondance des indices locaux, tout en limitant le recours aux points d'intérêt, elle ne s'en affranchit pas totalement et ne peut donc pas être mise en oeuvre dans les conditions les plus difficiles [SMT+18]. En effet, la répétition de motifs similaires en différents lieux rend les descripteurs locaux peu discriminants, et la taille

des environnements considérés entraine une explosion combinatoire lors du processus de mise en correspondance. Par ailleurs, le manque de texture dans les images peut être tel qu'il ne permet pas l'extraction d'indices pertinents en nombre suffisant, ou qu'il ne permet pas d'avoir une répartition spatiale de ces indices suffisante pour rendre le calcul fiable. Le calcul de pose basé sur l'utilisation d'indices locaux et de modèles 3D denses est donc peu adapté aux cas des environnemnents industriels que nous considérons. C'est pourquoi nous nous intéressons, dans la suite de ce mémoire, à l'utilisation d'indices de haut niveau, peu nombreux, discriminants, et dont la reconnaissance est peu dépendante du point de vue d'observation : les objets. La figure 2.1 illustre l'intuition selon laquelle les détections d'objets présents dans une image peuvent permettre d'estimer la pose de caméra associée, alors même que l'image semble mal adaptée aux méthodes basées sur les indices locaux : présence de larges zones non texturées (murs, table, chaise) et de motifs répétés (radiateur, store, parquet, clavier de l'ordinateur).

Dans ce chapitre, nous passons en revue un certain nombre de travaux qui tirent profit de la détection d'objets d'intérêt présents dans la scène pour l'estimation de pose de caméra (partie 2.1). Nous montrons notamment l'intérêt des méthodes basées sur la modélisation des objets par une forme géométrique simple qui permettent un calcul approché de la pose sans connaissance a priori, et nous mettons l'accent sur la modélisation par ellipsoïde (objet 3D) et ellipse (détection 2D). En étudiant le formalisme utilisé par les méthodes existantes pour décrire les ellipses (respectivement ellipsoïdes), nous mettons en évidence le fait qu'il ne permet pas toujours d'exploiter au maximum les spécificités de ces objets mathématiques en les considérant de manière trop générale, en l'occurence comme n'importe quels éléments de la famille des coniques (quadriques). Cette observation nous permet de présenter un regard critique sur une méthode d'estimation de pose de caméra, basée sur l'utilisation de correspondances conique - quadrique, que nous proposons (partie 2.2). Dans la partie 2.3, nous présentons une équation reliant un ellipsoïde et son ellipse projetée, qui sera reprise dans la suite de ce manuscrit pour construire des méthodes d'estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde.

2.1 Estimation de pose de caméra à partir d'objets : état de l'art

De nombreux auteurs ont remarqué que le calcul de pose de caméra à partir d'indices locaux était peu fiable en présence de forts changements de point de vue, et souvent mis en échec dans les environnements ambigus ou répétitifs [BADP17]. Pour surmonter ces difficultés, l'utilisation des objets a été investiguée.

2.1.1 Approches basées sur des correspondances entre points

Plusieurs méthodes [BS11, BBCS12, BADP17] utilisent les détections d'objets dans les images pour aider au calcul de pose de caméras et à la reconstruction de la scène (méthodes de SLAM : *Simultaneous Localization And Mapping*). Si ces méthodes parviennent à tirer profit de l'information apportée par les objets pour améliorer la précision des poses de caméras, elles restent toutefois dépendantes des points d'intérêt. Dans notre travail, nous nous intéressons plutôt à l'estimation de poses de caméras à partir d'objets seuls, et supposons que le modèle de scène est connu a priori, comme c'est le plus souvent le cas en Réalité Augmentée [MUS16].

Dans [ORL18], Oberweger *et al.* proposent d'estimer la pose d'un unique objet, qu'ils approximent en 3D par un parallélépipède rectangle, en entrainant un CNN à détecter les projections des sommets du parallélépipède dans l'image. Cette méthode permet ensuite de calculer la pose

1. https://en.wikipedia.org/wiki/Object_detection

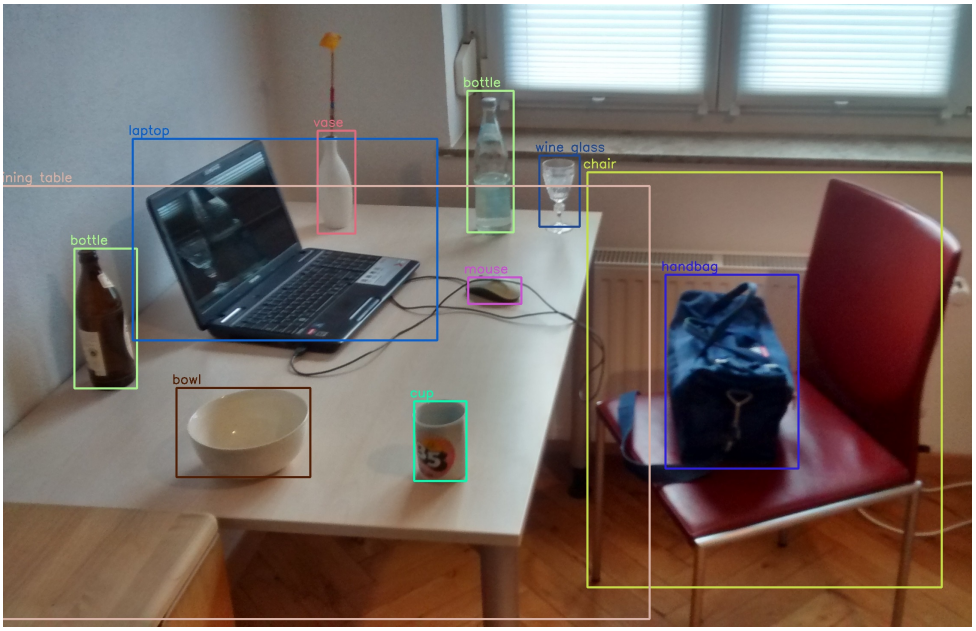


FIGURE 2.1 – Exemple d'image peu adaptée aux méthodes de mise en correspondance locale, et dans laquelle les objets détectés semblent pouvoir être utilisés pour estimer la pose de la caméra (objets détectés avec YOLOv3 [RF18])[1].

de l'objet par P*n*P. Crivellaro *et al.* [CRV+15, CRV+18] proposent eux, pour plus de robustesse, de décomposer l'objet en parties, et de calculer la pose de chaque partie à partir de points de contrôle virtuels détectés là-encore par un CNN. La même idée de points d'intérêt sémantiques détectés par CNN est présente dans [PZC+17], cette fois associée à un modèle de forme déformable pour calculer la pose. Dans CorNet, *et al.* prédit les poses d'objets similaires à ceux ayant été vus pendant l'entraînement, en détectant leurs coins dans l'image et en inférant leurs poses, puis en les mettant en correspondance avec les coins du modèle 3D CAO. Ces méthodes présentent toutes de très bonnes performances en termes de pose estimées, mais nécessitent un entraînement spécifique ainsi que des modèles 3D très précis pour générer des vues synthétiques des objets, ce qui limite leur mise en oeuvre. D'autre part, elles ne permettent pas de lever les ambiguïtés dues à la symétrie de certains objets.

Pour estimer la pose d'une caméra à partir de plusieurs objets, il convient tout d'abord de détecter les objets dans l'image. Pour cela, la plupart des détecteurs d'objet actuels (YOLO [RDGF16], Faster R-CNN [RHGS15], SSD [LAE+16]) présentent leurs résultats sous forme de boîtes englobantes rectangulaires alignées avec les axes de l'image. Une première approche naïve consisterait alors à assimiler les objets 3D à leurs centres de gravité, et à considérer les centres des boîtes détectées comme projections de ces points 3D, pour ensuite calculer la pose de la caméra par une méthode classique de P*n*P. La figure 2.2 montre les résultats d'expériences menées sur la base de données *Freiburg_2/desk* du TUM RGB-D Dataset [SEE+12]. Les projections d'objets sont détectées dans l'image par YOLOv3 [RF18], puis les objets 3D auxquels les détections peuvent correspondre sont déterminés sur la base d'une compatibilité d'étiquettes. Un algorithme RANSAC P*n*P [HR11] est ensuite appliqué à l'ensemble des correspondances possibles entre centres des boîtes 2D et centres de gravité des objets 3D pour déterminer la pose de la caméra. Entre 3 et 8 objets du modèle sont détectés dans chaque image, qui sont de taille 640×480 pixels, et les projections des caméras au sol couvrent grossièrement un carré de 4m de côté. Une des

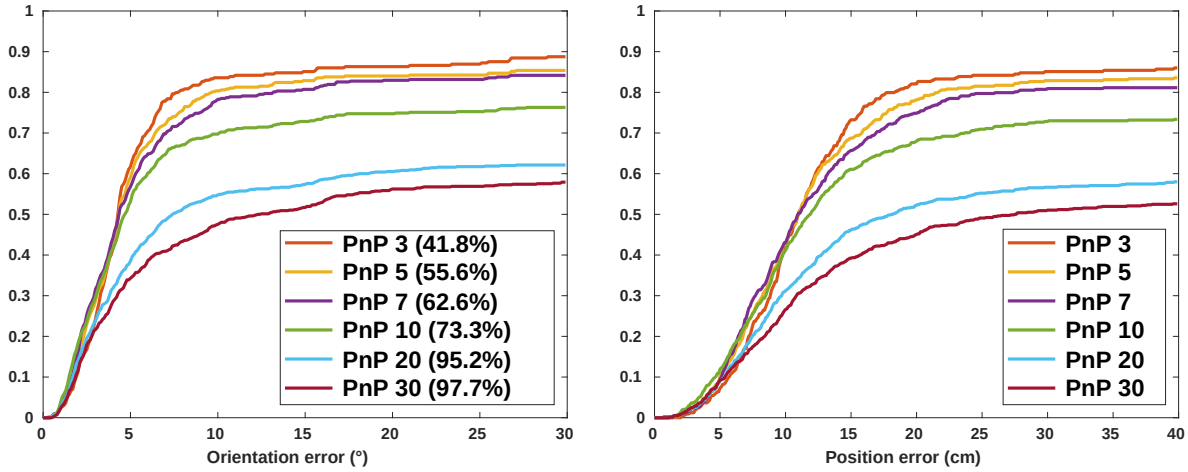


FIGURE 2.2 – Fonctions de répartition des erreurs en orientation (gauche) et en position (droite) sur les poses de caméras estimées par PnP à partir des centres des boîtes détectées. *PnP X* désigne le seuil de distance en pixels X utilisé pour séparer les inliers des outliers. Les pourcentages entre parenthèses désignent la proportion d'images pour lesquelles la méthode a été en mesure d'estimer une pose (au moins 4 inliers).

difficultés de cette méthode réside dans le choix du seuil de distance utilisé pour séparer les inliers des outliers. Les résultats montrent qu'une petite valeur de ce seuil empêche de calculer une pose (moins de 4 inliers) pour la plupart des images (41.8% d'images traitées dans le cas où le seuil est fixé à 3 pixels), mais donne des erreurs inférieures à 10° en orientation et 20 cm en position dans environ 80% des cas traités, tandis qu'une valeur de seuil plus grande permet de traiter plus d'images (97.7% des images lorsque le seuil est fixé à 30 pixels) mais dégrade la précision des poses estimées (seulement 45% d'erreurs inférieures à 10° et 20 cm). D'autre part, cette méthode nécessite de détecter au minimum 4 objets avec une étiquette correcte pour pouvoir estimer une pose, ce qui n'est pas toujours le cas en pratique. Ces expériences sont reprises dans le chapitre 6.

La réduction à un simple point des boîtes rectangulaires détectées entraine nécessairement une perte d'information qui est préjudiciable au calcul de pose. De plus, les positions des détections d'objets dans l'image sont incertaines, et le fait de réduire ces détections à leur centre ne permet pas de diminuer l'influence de cette incertitude. Il convient donc de conserver une représentation globale de chaque objet, même si la modélisation d'une forme complexe par une forme plus simple ne peut permettre d'envisager qu'une résolution approchée du problème d'estimation de pose de caméra. Cela reste toutefois très utile puisqu'il est ensuite possible d'affiner cette pose par de nombreuses approches existantes.

2.1.2 Approches basées sur des modèles d'objets

Afin de tirer profit de toute l'information apportée par les boîtes de détection, Li *et al.* [LMD17, LXMD18, LMD19] proposent de considérer les boîtes 2D comme des approximations, assez grossières, des projections de boîtes englobantes 3D (parallélépipèdes rectangles), et d'utiliser un certain nombre d'informations contextuelles sur la scène pour retrouver la pose relative de la caméra. Une des limitations de cette méthode vient du fait que la projection d'un parallélépipède rectangle sur un plan n'est en général pas un rectangle, d'où l'introduction d'une

nouvelle incertitude qui s'ajoute à celle due à la simplification de la forme des objets. En outre, les paramètres du polygone projeté ne peuvent être déduits formellement des paramètres du parallélépipède et de ceux de la projection. En effet, l'identification des arêtes définissant la projection d'un parallélépipède dépend de la pose de la caméra et ne peut être exprimée formellement. Autrement dit, il est impossible d'écrire une équation liant les modèles 3D et 2D des objets, et a fortiori il est impossible d'obtenir une expression formelle de la pose de la caméra.

La modélisation des objets 2D par des ellipses inscrites dans les boîtes de détection a permis à Crocco *et al.* de proposer une solution analytique au problème de reconstruction par *Structure from Motion (SfM)* de la scène sous la forme d'un ensemble d'ellipsoïdes [CRD16]. Si cette méthode présente l'avantage de s'appuyer sur des modélisations 2D et 3D compatibles (la projection d'un ellipsoïde sur un plan est toujours une ellipse), elle est cependant limitée au cas des projections orthographiques (voir figure 2.3), de même que son extension intégrant des modèles CAO génériques des objets pour contraindre la forme des ellipsoïdes reconstruits [GBRD17].

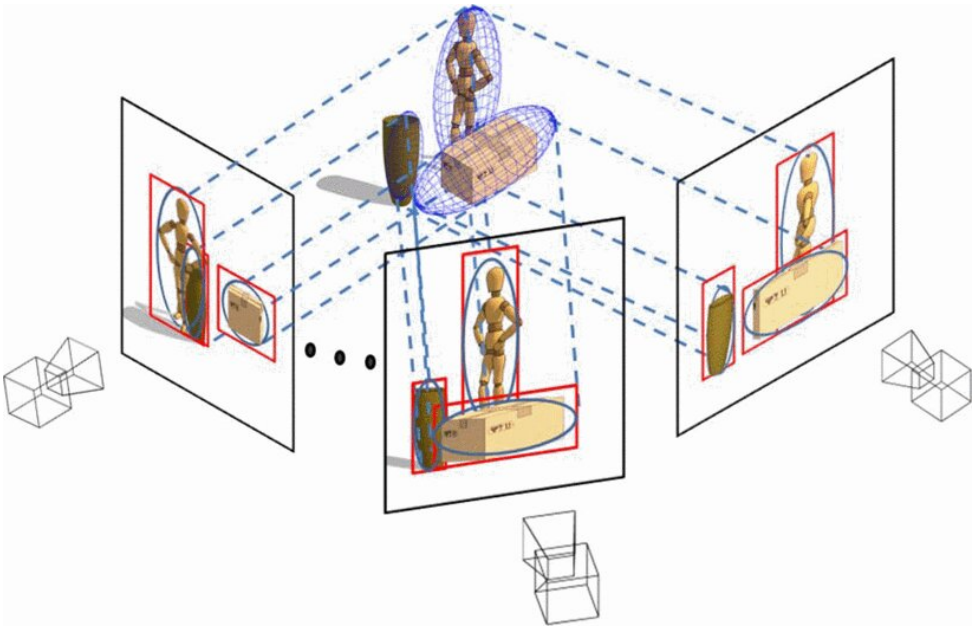


FIGURE 2.3 – Reconstruction *SfM* d'une scène d'objets modélisés par des ellipsoïdes, avec modèle de caméra orthographique (tirée de [CRD16]).

Dans [NMS19], Nicholson *et al.* ont introduit une méthode de SLAM sémantique permettant, là encore, de construire l'ensemble des ellipsoïdes 3D tout en calculant les poses des caméras, mais cette fois en prenant en compte un modèle de caméra perspective. Cette solution, qui est numérique, consiste à minimiser une erreur de reprojection géométrique des ellipsoïdes en fonction des six paramètres de pose de la caméra et des paramètres des ellipsoïdes, mais nécessite pour cela de connaître une estimée de cette pose obtenue par odométrie.

Par ailleurs, Rubino *et al.* [RCD18] ont proposé une solution analytique pour construire un tel modèle sémantique de scène à partir de seulement trois caméras perspectives calibrées (voir figure 2.4). Dans notre travail, nous nous sommes inspirés de cette méthode et l'avons adaptée pour résoudre le problème inverse qui consiste à estimer la pose d'une caméra à partir de correspondances entre ellipses et ellipsoïdes (voir partie 2.2).

La modélisation d'un objet 3D sous forme d'ellipsoïde, associée à la modélisation de sa projection sous forme d'ellipse, permet donc d'aboutir à un lien plus naturel entre les différents modèles.

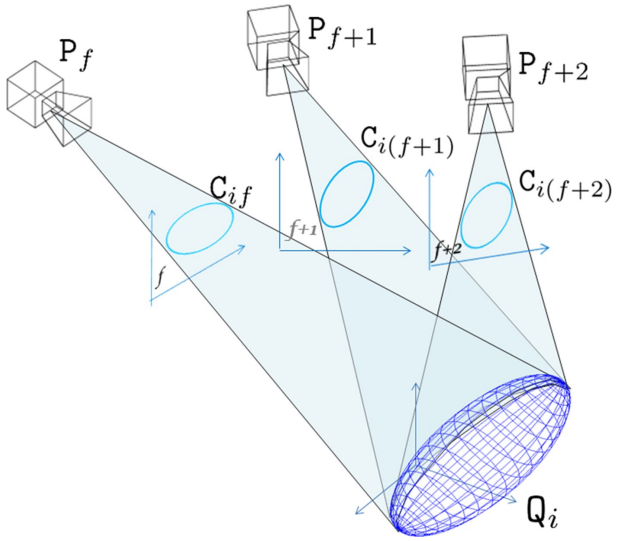


FIGURE 2.4 – Reconstruction perspective d'un ellipsoïde à partir de trois ellipses projetées (tirée de [RCD18]).

Plus précisément, il est possible d'obtenir formellement les paramètres de l'ellipse projetée en fonction des paramètres de l'ellipsoïde et de ceux de la projection, ce qui permet d'envisager l'obtention d'une expression formelle des paramètres de la projection (paramètres de la caméra) à partir de correspondances entre ellipses et ellipsoïdes. Un exemple d'une telle expression est présentée dans la partie suivante.

2.2 Estimation de pose de caméra à partir de correspondances conique - quadrique

Les travaux présentés précédemment et utilisant le paradigme de modélisation ellipse - ellipsoide [CRD16, GBRD17, RCD18, NMS19] reposent tous sur une représentation en coordonnées homogènes de ces éléments géométriques [HZ04]. Dans cette partie, nous proposons une méthode d'estimation de pose de caméra basée sur ce même formalisme.

Les ellipses et les ellipsoïdes sont des éléments de la famille des coniques (ellipses, paraboles, hyperboles) et des quadriques (ellipsoïdes, hyperboloïdes, paraboloïdes, cônes, cylindres). Dans cette partie, nous présentons brièvement les ellipsoïdes et ellipses (partie 2.2.1), puis leur représentation en coordonnées homogènes (partie 2.2.2). Ensuite nous rappelons l'équation de projection d'une quadrique en coordonnées homogènes (partie 2.2.3). Dans la partie 2.2.4, nous reprenons l'observation faite dans [CRD16, GBRD17, RCD18] selon laquelle cette équation peut être linéarisée, ce qui nous permet d'introduire une résolution par *Direct Linear Transformation* [Sut74] dont le but est d'obtenir les éléments de la matrice de projection de la caméra (partie 2.2.5). Les limites de cette méthode sont discutées dans la partie 2.2.6.

2.2.1 Rappels sur les ellipses et ellipsoïdes

Un ellipsoïde peut être entièrement caractérisé par son centre $\boldsymbol{O} \in \mathbb{R}^3$ et sa forme quadratique associée, représentée par la matrice $A \in \mathbb{R}^{3\times3}$, de sorte qu'il est défini par l'équation

$$(\boldsymbol{X} - \boldsymbol{O})^\top A(\boldsymbol{X} - \boldsymbol{O}) = 1$$

dans laquelle $\boldsymbol{X} \in \mathbb{R}^3$ est un point appartenant à l'ellipsoïde, et A est une matrice **symétrique définie positive**. Autrement dit, elle vérifie

$$\forall \boldsymbol{X} \in \mathbb{R}^3, \qquad \boldsymbol{X}^\top A \boldsymbol{X} \geq 0$$

avec égalité si et seulement si $\boldsymbol{X} = \boldsymbol{0}$ (vecteur nul).

La matrice A est symétrique réelle, donc diagonalisable dans une base orthonormée. Ses éléments propres caractérisent la taille et l'orientation de l'ellipsoïde. En effet, si a, b, c sont les longueurs de ses axes principaux, et si son orientation est donnée par la matrice de rotation $R \in SO_3(\mathbb{R})$, alors la matrice A s'écrit :

$$A = R \begin{pmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{pmatrix} R^\top$$

Les valeurs propres de A $\left(\lambda_{A,1} = \frac{1}{a^2}, \lambda_{A,2} = \frac{1}{b^2}, \lambda_{A,3} = \frac{1}{c^2}\right)$ découlent directement des longueurs des axes principaux de l'ellipsoïde, et les vecteurs propres de A, qui sont orthogonaux entre eux, donnent l'orientation de l'ellipsoïde.

Dans le cas général, les trois valeurs propres de A sont différentes : on parle alors d'ellipsoïde non dégénéré. Un tel ellipsoïde possède trois plans de symétrie orthogonaux, qui sont les plans passant par son centre et de normales les vecteurs propres de A. Lorsque A ne possède que deux valeurs propres distinctes, alors l'ellipsoïde possède un axe de révolution et un plan de symétrie orthogonal à cet axe, et on parle d'ellipsoïde de révolution, ou sphéroïde. Enfin, lorsque les trois valeurs propres sont identiques, l'ellipsoïde est une sphère. Ces trois types d'ellipsoïdes sont présentés dans le tableau 2.1.

Les ellipses obéissent également à une équation de la forme

$$(\boldsymbol{X} - \boldsymbol{O})^\top A (\boldsymbol{X} - \boldsymbol{O}) = 1 \tag{2.1}$$

dans laquelle $\boldsymbol{X} = (x, y)^\top \in \mathbb{R}^2$ est un point quelconque de l'ellipse, $\boldsymbol{O} \in \mathbb{R}^2$ est son centre, et $A \in \mathbb{R}^{2\times 2}$ est symétrique réelle définie positive.

2.2.2 Réprésentation des ellipses et ellipsoïdes en coordonnées homogènes

Développer l'équation de l'ellipse (2.1) permet de la réécrire sous la forme

$$\boldsymbol{X}^\top P \boldsymbol{X} + \boldsymbol{Q} \cdot \boldsymbol{X} + r = 0$$

où

$$\begin{cases} P = A \\ \boldsymbol{Q} = -2A\boldsymbol{O} \\ r = \boldsymbol{O}^\top A \boldsymbol{O} - 1 \end{cases}$$

En posant ensuite

$$P = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix} \qquad \boldsymbol{Q} = \begin{pmatrix} d \\ e \end{pmatrix} \qquad r = f$$

et en développant à nouveau les calculs, on aboutit à l'équation cartésienne de l'ellipse :

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

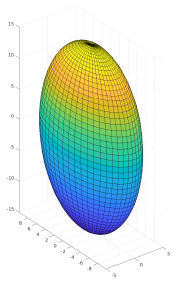
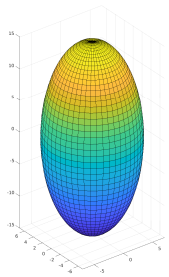
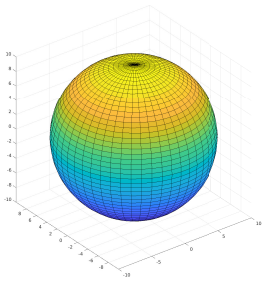
	Ellipsoïde non dégénéré	Ellipsoïde de révolution	Sphère
Exemple			
Longueurs des axes principaux	a, b, c	a, b	r
Valeurs propres de A	$\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{c^2}$	$\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{b^2}$	$\frac{1}{r^2}, \frac{1}{r^2}, \frac{1}{r^2}$
Signes des valeurs propres	$+, +, +$	$+, +, +$	$+, +, +$
Polynôme caractéristique de A	$P_A(x) = (x-a)(x-b)(x-c)$	$P_A(x) = (x-a)(x-b)^2$	$P_A(x) = (x-r)^3$
Polynôme minimal de A	$\pi_A(x) = (x-a)(x-b)(x-c)$	$\pi_A(x) = (x-a)(x-b)$	$\pi_A(x) = (x-r)$

TABLE 2.1 – Présentation des différents types d'ellipsoïdes.

Or ce type d'équation est le même pour tous types de coniques, et seules certaines relations entre les coefficients de l'équation permettent de les distinguer.

Par ailleurs, cette équation peut être transformée en une équation homogène en effectuant les substitutions $x \mapsto x_1/x_3$ et $y \mapsto x_2/x_3$ [HZ04] (page 30) :

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$

Finalement, l'équation homogène peut être réécrite sous une forme matricielle :

$$\boldsymbol{x}^\top C\boldsymbol{x} = 0$$

en posant

$$C = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix}$$

et c'est cette équation qui est utilisée pour définir une ellipse en coordonnées homogènes.

De la même façon, un ellipsoïde est défini, comme toute quadrique, par l'équation

$$\boldsymbol{x}^\top Q\boldsymbol{x} = 0$$

où $x \in \mathbb{R}^4$ est le vecteur homogène associé à un point 3D appartenant à la quadrique, et où $Q \in \mathbb{R}^{4\times4}$ est une matrice symétrique réelle.

2.2.3 Equation de projection d'une quadrique

En coordonnées homogènes, une conique possède 5 degrés de liberté représentés par les 6 éléments de la partie triangulaire inférieure de sa matrice C, excepté un pour l'échelle, et sa conique duale est définie par la matrice $C^* = adj(C)$. De la même façon, une quadrique possède 9 degrés de liberté, et sa quadrique duale est définie par la matrice $Q^* = adj(Q)$. Les matrices duales permettent de caractériser la tangence. Plus précisément, les droites $\boldsymbol{l}$ tangentes à la conique sont celles qui vérifient

$$\boldsymbol{l}^\top C^*\boldsymbol{l} = 0$$

et les plans $\boldsymbol{\pi}$ tangents à la quadrique sont ceux qui vérifient

$$\boldsymbol{\pi}^\top Q^*\boldsymbol{\pi} = 0$$

Par ailleurs, la matrice de projection associée à une caméra s'écrit sous la forme

$$P = K[R|t] = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

où R et T représentent la rotation et la translation entre le repère du modèle et le repère associé à la caméra, et où K représente les paramètres intrinsèques de la caméra.

R. Hartley et A. Zisserman [HZ04] (page 201) ont montré que l'action projective de la caméra sur la quadrique obéit à l'équation

$$C^* = PQ^*P^\top \tag{2.2}$$

En effet, les droites $\boldsymbol{l}$ tangentes à la conique ($\boldsymbol{l}^\top C^*\boldsymbol{l} = 0$) sont rétroprojetées sur les plans $\boldsymbol{\pi} = P^\top\boldsymbol{l}$ qui sont eux-mêmes tangents à la quadrique ($\boldsymbol{\pi}^\top Q^*\boldsymbol{\pi} = 0$), d'où, pour chaque droite on a :

$$\begin{aligned} \boldsymbol{\pi}^\top Q^*\boldsymbol{\pi} &= \boldsymbol{l}^\top PQ^*P^\top\boldsymbol{l} \\ &= \boldsymbol{l}^\top C^*\boldsymbol{l} = 0 \end{aligned}$$

Comme ceci est vrai pour toutes les droites tangentes à la conique, l'équation (2.2) est prouvée.

2.2.4 Linéarisation de l'équation

L'équation de projection (2.2) n'est pas linéaire par rapport aux éléments de P, mais Crocco *et al.* [CRD16, GBRD17, RCD18] ont montré qu'elle est linéaire par rapport à des produits des éléments de P.

Pour cela, définissons l'opérateur $vec()$ qui concatène tous les éléments d'une matrice quelconque en un vecteur, et l'opérateur $vech()$ qui concatène les éléments de la partie triangulaire inférieure d'une matrice symétrique. Puis définissons

$$v^* = vech(Q^*) \qquad \text{et} \qquad c^* = vech(C^*)$$

Il est alors possible de réarranger les produits des éléments de P et $P^\top$ en une matrice $G \in \mathbb{R}^{6\times 10}$ comme suit :

$$G = D(P \otimes P)E$$

où $\otimes$ symbolise le produit de Kronecker, et où les matrices $D \in \mathbb{R}^{6\times 9}$ et $E \in \mathbb{R}^{16\times 10}$, sont définies de telle sorte que

$$\forall X \in \mathbb{R}^{3\times 3},\ vech(X) = Dvec(X)$$

et

$$\forall Y \in \mathbb{R}^{4\times 4},\ vec(Y) = Evech(Y)$$

D et E s'écrivent donc

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{et} \qquad E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice G s'exprime alors en fonction des coefficients de P :

$$G = \begin{bmatrix} p_{11}^2 & 2p_{11}p_{12} & p_{12}^2 & 2p_{11}p_{13} & 2p_{12}p_{13} & p_{13}^2 \\ p_{11}p_{21} & p_{11}p_{22}+p_{12}p_{21} & p_{12}p_{22} & p_{11}p_{23}+p_{13}p_{21} & p_{12}p_{23}+p_{13}p_{22} & p_{13}p_{23} \\ p_{21}^2 & 2p_{21}p_{22} & p_{22}^2 & 2p_{21}p_{23} & 2p_{22}p_{23} & p_{23}^2 \\ p_{11}p_{31} & p_{11}p_{32}+p_{12}p_{31} & p_{12}p_{32} & p_{11}p_{33}+p_{13}p_{31} & p_{12}p_{33}+p_{13}p_{32} & p_{13}p_{33} \\ p_{21}p_{31} & p_{21}p_{32}+p_{22}p_{31} & p_{22}p_{32} & p_{21}p_{33}+p_{23}p_{31} & p_{22}p_{33}+p_{23}p_{32} & p_{23}p_{33} \\ p_{31}^2 & 2p_{31}p_{32} & p_{32}^2 & 2p_{31}p_{33} & 2p_{32}p_{33} & p_{33}^2 \end{bmatrix}$$

$$\begin{bmatrix} 2p_{11}p_{14} & 2p_{12}p_{14} & 2p_{13}p_{14} & p_{14}^2 \\ p_{11}p_{24}+p_{14}p_{21} & p_{12}p_{24}+p_{14}p_{22} & p_{13}p_{24}+p_{14}p_{23} & p_{14}p_{24} \\ 2p_{21}p_{24} & 2p_{22}p_{24} & 2p_{23}p_{24} & p_{24}^2 \\ p_{11}p_{34}+p_{14}p_{31} & p_{12}p_{34}+p_{14}p_{32} & p_{13}p_{34}+p_{14}p_{33} & p_{14}p_{34} \\ p_{21}p_{34}+p_{24}p_{31} & p_{22}p_{34}+p_{24}p_{32} & p_{23}p_{34}+p_{24}p_{33} & p_{24}p_{34} \\ 2p_{31}p_{34} & 2p_{32}p_{34} & 2p_{33}p_{34} & p_{34}^2 \end{bmatrix}$$

En utilisant la matrice G, il est possible de réécrire l'équation de projection d'une quadrique (2.2) sous une forme linéaire :

$$\beta c^* = G v^* \tag{2.3}$$

où β est un scalaire non nul.

2.2.5 Détermination de la matrice de projection de la caméra

Pour retrouver les éléments de la matrice P analytiquement à partir d'un ensemble de paires conique - quadrique connues, nous passons par l'intermédiaire de la matrice G. En effet, un système d'équations (2.3) d'inconnue G

$$\beta c_i^* = G v_i^* \qquad \text{où } 1 \leq i \leq N$$

peut être résolu en utilisant la méthode DLT (*Direct Linear Transformation*) [Sut74], *i. e.* peut être réécrit sous la forme :

$$0 = Bg,$$

où $g = vec(G) \in \mathbb{R}^{60}$, et $B \in \mathbb{R}^{15n \times 60}$ est la concaténation des matrices $b_i \in \mathbb{R}^{15 \times 60}$:

$$b_i = \begin{bmatrix} c_{2i}\mathbf{v_i}^{*\top} & -c_{1i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & c_{3i}\mathbf{v_i}^{*\top} & -c_{2i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & c_{4i}\mathbf{v_i}^{*\top} & -c_{3i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & c_{5i}\mathbf{v_i}^{*\top} & -c_{4i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & c_{6i}\mathbf{v_i}^{*\top} & -c_{5i}\mathbf{v_i}^{*\top} \\ -c_{3i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & c_{1i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & -c_{4i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & c_{2i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & -c_{5i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & c_{3i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & -c_{6i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & c_{4i}\mathbf{v_i}^{*\top} \\ c_{4i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & -c_{1i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & c_{5i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & -c_{2i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & c_{6i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & -c_{4i}\mathbf{v_i}^{*\top} \\ -c_{5i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & c_{1i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top \\ \mathbf{0_{10}}^\top & -c_{6i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & c_{2i}\mathbf{v_i}^{*\top} \\ c_{6i}\mathbf{v_i}^{*\top} & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & \mathbf{0_{10}}^\top & -c_{1i}\mathbf{v_i}^{*\top} \end{bmatrix},$$

avec :

$$\mathbf{c_i^*}^\top = \begin{bmatrix} c_{1i} & c_{2i} & c_{3i} & c_{4i} & c_{5i} & c_{6i} \end{bmatrix},$$
$$\mathbf{v_i^*}^\top = \begin{bmatrix} v_{1i} & v_{2i} & v_{3i} & v_{4i} & v_{5i} & v_{6i} & v_{7i} & v_{8i} & v_{9i} & v_{10i} \end{bmatrix},$$
$$\mathbf{0_{10}}^\top = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Chaque paire de conique - quadrique fournit donc 15 équations linéaires homogènes d'inconnues les éléments de G. Cependant, on peut observer que seulement 5 d'entre elles sont linéairement indépendantes. Ainsi, 12 paires de correspondances sont nécessaires pour résoudre le système. La meilleure solution au sens des moindres carrés peut être obtenue en choisissant g comme vecteur singulier associé à la plus petite valeur singulière de B. Finalement, il est possible de retrouver les éléments de la matrice de projection P à partir de ceux de G, même si, en pratique, la redondance des éléments de P dans les éléments de G rend difficile cette opération.

2.2.6 Limites de la méthode

La méthode décrite précédemment nécessite la connaissance de 12 correspondances pour calculer la pose de la caméra. D'une part, il est assez rare de détecter autant d'objets dans une image, et, d'autre part, on peut observer que les méthodes reposant sur des correspondances point - point ne requièrent que 4 correspondances. Cette solution a donc un faible intérêt pratique.

Nous pouvons mettre en avant deux explications à ce constat. Premièrement, cette méthode ne prend pas en compte une éventuelle connaissance des paramètres intrinsèques de la caméra K (souvent supposée en calcul de pose). Deuxièmement, la résolution proposée n'intègre pas les contraintes permettant de distinguer les ellipses et les ellipsoïdes des autres coniques et quadriques, et ce manque de spécificité engendre nécessairement une perte d'information préjudiciable à une résolution efficace du problème.

L'équation (2.2) a toutefois le mérite de montrer qu'il existe une relation formelle entre une quadrique et sa conique projetée, impliquant les paramètres de la projection. Cette même démonstration a été faite par David Eberly dans le cas plus spécifique des ellipsoïdes [Ebe07]. Il montre en effet qu'une ellipse est la projection d'un ellipsoïde si et seulement si ces deux entités vérifient une certaine équation, que nous présentons dans la partie suivante. Cette propriété est importante puisqu'elle permet d'envisager une résolution formelle du problème d'estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde.

2.3 L'Equation d'Alignement des Cônes

L'Equation d'Alignement des Cônes [Ebe07] traduit le fait qu'un ellipsoïde se projette dans une ellipse si et seulement si le cône dont le sommet est le centre de projection et qui passe par les points de l'ellipse est tangent à l'ellipsoïde. Nous la présentons dans la suite de cette partie, juste après quelques rappels sur les cônes.

2.3.1 Rappels sur les cônes du second degré

Un cône du second degré peut être entièrement caractérisé par son sommet $\boldsymbol{E} \in \mathbb{R}^3$ et sa forme quadratique associée, représentée par la matrice $B \in \mathbb{R}^{3\times3}$, de sorte qu'il est défini par l'équation

$$(\boldsymbol{X} - \boldsymbol{E})^\top B(\boldsymbol{X} - \boldsymbol{E}) = 0 \tag{2.4}$$

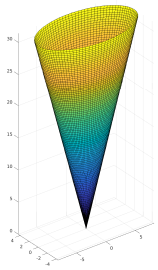
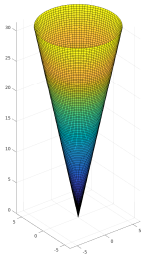
	Cône elliptique non dégénéré	Cône de révolution
Exemple		
Valeurs propres de B	$\lambda_1, \lambda_2, \lambda_3$	$\lambda_1, \lambda_2, \lambda_2$
Signes des valeurs propres	$-,+,+$ ou $+,-,-$	$-,+,+$ ou $+,-,-$
Polynôme caractéristique de B	$P_B(x) = (x-\lambda_1)(x-\lambda_2)(x-\lambda_3)$	$P_B(x) = (x-\lambda_1)(x-\lambda_2)^2$
Polynôme minimal de B	$\pi_B(x) = (x-\lambda_1)(x-\lambda_2)(x-\lambda_3)$	$\pi_B(x) = (x-\lambda_1)(x-\lambda_2)$

TABLE 2.2 – Présentation des différents types de cônes du second degré.

dans laquelle $\boldsymbol{X} \in \mathbb{R}^3$ est un point appartenant au cône, et la matrice B est une matrice symétrique réelle inversible, de signature (2,1) ou (1,2), *i. e.* elle possède une valeur propre d'un signe et deux valeurs propres de l'autre signe.

La matrice B caractérise les proportions et l'orientation du cône. Une fois décomposée en valeurs et vecteurs propres (qui sont orthogonaux entre eux), la matrice B s'écrit :

$$B = R' \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} R'^{\top}$$

où $R' \in SO_3(\mathbb{R})$ représente l'orientation du cône.

Dans le cas général, les deux valeurs propres de B de même signe sont différentes : on parle alors de cône elliptique non dégénéré. Un tel cône possède trois plans de symétrie orthogonaux, qui sont les plans passant par son sommet et de normales les vecteurs propres de B. Lorsque les deux valeurs propres de même signe sont égales, alors le cône possède un axe de révolution et un plan de symétrie orthogonal à cet axe, et on parle de cône de révolution. Ces deux types de cônes sont présentés dans le tableau 2.2 (les illustrations représentent des demi-cônes).

2.3.2 Le cône de projection

Suivant les notations introduites dans [Ebe07] et présentées dans la figure 2.5, nous considérons un ellipsoïde de centre $\boldsymbol{C}$ et de forme quadratique A défini par l'équation

$$(\boldsymbol{X} - \boldsymbol{C})^{\top} A (\boldsymbol{X} - \boldsymbol{C}) = 1 \tag{2.5}$$

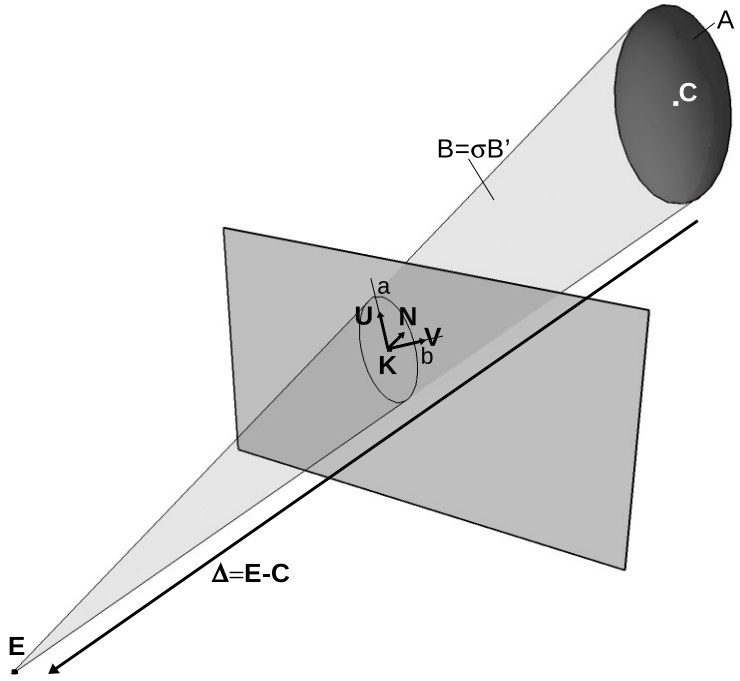


FIGURE 2.5 – Illustration du plan image, du centre de projection, de l'ellipsoïde, et de son ellipse projetée.

Etant donnés un centre de projection $\boldsymbol{E}$ et un plan de projection de normale $\boldsymbol{N}$ auquel $\boldsymbol{E}$ n'appartient pas, la projection de l'ellipsoïde sur ce plan est une ellipse de centre $\boldsymbol{K}$ et de demi-axes de longueurs a et b. Les directions principales de l'ellipse sont représentées par les vecteurs unitaires $\boldsymbol{U}$ et $\boldsymbol{V}$, de telle sorte que le triplet $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{N}\}$ forme une base orthonormale de l'espace.

On définit le cône de projection comme étant le cône de sommet $\boldsymbol{E}$ tangent à l'ellipsoïde. Toujours d'après [Ebe07], en notant $\boldsymbol{\Delta} = \boldsymbol{E} - \boldsymbol{C}$ le vecteur reliant le centre de l'ellipsoïde $\boldsymbol{C}$ au centre de projection $\boldsymbol{E}$, le cône de projection est caractérisé par la matrice

$$B = A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A - (\boldsymbol{\Delta}^\top A\boldsymbol{\Delta} - 1)A$$

de sorte que les points $\boldsymbol{X}$ du cône de projection sont ceux qui satisfont l'équation

$$(\boldsymbol{X} - \boldsymbol{E})^\top B(\boldsymbol{X} - \boldsymbol{E}) = 0.$$

2.3.3 Le cône de rétroprojection

On définit le cône de rétroprojection comme le cône elliptique généré par l'ensemble des droites passant par $\boldsymbol{E}$ et par un point de l'ellipse. On dit que l'ellipse est une directrice du cône. Eberly montre que ce cône est caractérisé par une matrice B', définie de la manière suivante

$$B' = P^\top MP - Q$$

avec

$$\begin{aligned} M &= \boldsymbol{U}\boldsymbol{U}^\top/a^2 + \boldsymbol{V}\boldsymbol{V}^\top/b^2 \\ \mathbf{W} &= \boldsymbol{N}/(\boldsymbol{N}\cdot(\boldsymbol{K}-\boldsymbol{E})) \\ P &= I - (\boldsymbol{K}-\boldsymbol{E})\mathbf{W}^\top \\ Q &= \mathbf{W}\mathbf{W}^\top \end{aligned}$$

Là encore, les points $\boldsymbol{X}$ du cône de rétroprojection sont ceux qui vérifient l'équation

$$(\boldsymbol{X}-\boldsymbol{E})^\top B'(\boldsymbol{X}-\boldsymbol{E}) = 0.$$

2.3.4 L'Equation d'Alignement des Cônes

Dès lors, étant donnés un ellipsoïde, une projection centrale (centre et plan de projection), ainsi qu'une ellipse inscrite dans le plan de projection, l'ellipse est l'image projetée de l'ellipsoïde si et seulement si les cônes de projection et de rétroprojection sont confondus, *i. e.* si et seulement s'il existe un scalaire non nul σ tel que $B = \sigma B'$ [Ebe07]. Soit, en remplaçant B par son expression dépendant des paramètres de l'ellipsoïde :

$$A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A + \mu A = \sigma B' \tag{2.6}$$

avec

$$\mu \stackrel{def}{=} 1 - \boldsymbol{\Delta}^\top A \boldsymbol{\Delta}$$

Cette équation, que nous appelons *Equation d'Alignement des Cônes*, traduit ainsi la tangence entre le cône de rétroprojection et l'ellipsoïde.

2.3.5 Signification des termes de l'équation

L'Equation d'Alignement des Cônes est une équation matricielle, indépendante de la base vectorielle dans laquelle les éléments sont exprimés.

Dans cette équation, l'ellipse projetée est représentée par le cône de rétroprojection, caractérisé par la matrice B', tandis que l'ellipsoïde est caractérisé par la matrice A. Dans notre travail, pour estimer la pose d'une caméra à partir d'un ellipsoïde et de son ellipse projetée, nous supposons que l'ellipse est connue dans l'image et que les paramètres intrinsèques de la caméra le sont également. Ainsi, nous connaissons l'expression de la matrice B' dans une certaine base associée à la caméra. Par ailleurs, nous supposons que l'ellipsoïde constitue le modèle de la scène, par rapport auquel nous cherchons à retrouver la pose de la caméra. Ainsi, nous connaissons l'expression de la matrice A dans une autre base, associée cette fois au modèle. Ces expressions permettent de fixer la forme du cône et la taille de l'ellipsoïde (valeurs propres des matrices B' et A). Les expressions de ces matrices dans une certaine base donnent l'orientation de l'élément correspondant par rapport à cette même base. Une fois les matrices A et B' exprimées dans une base commune, le couple $\{A, B'\}$ traduit donc l'orientation relative entre le cône et l'ellipsoïde. Leur position relative est, elle, représentée par le vecteur $\boldsymbol{\Delta}$. Enfin, σ est une inconnue scalaire additionnelle qui n'a pas de signification évidente. Résoudre l'équation (2.6) consiste donc à déterminer les scalaires σ solutions et les expressions, dans une base commune, des matrices et vecteurs $(A, B', \boldsymbol{\Delta})$ solutions.

En remplaçant $\boldsymbol{X}$ par $\boldsymbol{E}$ dans l'équation de l'ellipsoïde (2.5), il vient que la variable secondaire μ, qui dérive de A et $\boldsymbol{\Delta}$, caractérise la configuration du centre de projection $\boldsymbol{E}$ par rapport à l'ellipsoïde. Plus précisément, les configurations possibles sont :

- $\mu < 0$: $\boldsymbol{E}$ est situé à l'extérieur de l'ellipsoïde,
- $\mu = 0$: $\boldsymbol{E}$ appartient à l'ellipsoïde,
- $\mu > 0$: $\boldsymbol{E}$ est situé à l'intérieur de l'ellipsoïde.

Pour aboutir à l'équation (2.6), Eberly suppose que $\boldsymbol{E}$ est situé à l'extérieur de l'ellipsoïde, et nous retrouvons d'une autre manière, dans le chapitre 6, que $\mu < 0$.

2.3.6 Lien avec un problème aux valeurs propres généralisé

Dans [Ebe07], David Eberly a montré qu'il existe un lien entre l'Equation d'Alignement des Cônes (2.6) et un problème aux valeurs propres généralisé [GVL96].

Résultat 1. *Si un quadruplet $(A, B', \boldsymbol{\Delta}, \sigma)$ vérifie l'Equation d'Alignement des Cônes* (2.6)*, alors il vérifie aussi l'équation* (2.7)*.*

$$A\boldsymbol{\Delta} = \sigma B'\boldsymbol{\Delta} \tag{2.7}$$

σ est donc une valeur propre généralisée du couple $\{A, B'\}$, et $\boldsymbol{\Delta}$ est un vecteur propre généralisé associé à σ.

Démonstration. En multipliant à droite l'équation (2.6) par $\boldsymbol{\Delta}$, on obtient

$$(A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A - (\boldsymbol{\Delta}^\top A\boldsymbol{\Delta} - 1)A)\boldsymbol{\Delta} = \sigma B'\boldsymbol{\Delta}$$

soit

$$A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A\boldsymbol{\Delta} - \boldsymbol{\Delta}^\top A\boldsymbol{\Delta}A\boldsymbol{\Delta} + A\boldsymbol{\Delta} = \sigma B'\boldsymbol{\Delta}$$

Or, $\boldsymbol{\Delta}^\top A\boldsymbol{\Delta}$ étant un scalaire, le terme de gauche s'écrit :

$$\begin{aligned} A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A\boldsymbol{\Delta} - \boldsymbol{\Delta}^\top A\boldsymbol{\Delta}A\boldsymbol{\Delta} + A\boldsymbol{\Delta} &= A\boldsymbol{\Delta}(\boldsymbol{\Delta}^\top A\boldsymbol{\Delta}) - \boldsymbol{\Delta}^\top A\boldsymbol{\Delta}A\boldsymbol{\Delta} + A\boldsymbol{\Delta} \\ &= (\boldsymbol{\Delta}^\top A\boldsymbol{\Delta})A\boldsymbol{\Delta} - \boldsymbol{\Delta}^\top A\boldsymbol{\Delta}A\boldsymbol{\Delta} + A\boldsymbol{\Delta} \\ &= A\boldsymbol{\Delta} \end{aligned}$$

Finalement, $A\boldsymbol{\Delta} = \sigma B'\boldsymbol{\Delta}$. □

Nous avons travaillé à la résolution de l'Equation d'Alignement des Cônes, et les contributions que nous avons apportées sont résumées dans le chapitre 3.

2.4 Conclusion

Les objets d'intérêt présents dans un environnement industriel peuvent être vus comme des indices de haut niveau, porteurs d'une information sémantique et géométrique importante. Ils peuvent être, de plus, en nombre très limité en comparaison avec des indices locaux par exemple, alors qu'ils possèdent un fort pouvoir discriminant, et qu'ils peuvent être détectés selon une large variété de points de vue. Ils représentent donc des indices de choix pour estimer la pose d'une caméra dans un tel environnement.

Nous avons vu, de plus, que le choix du mode de représentation des objets est crucial dans le développement d'une telle méthode. Un choix pertinent consiste à modéliser les objets sous forme d'ellipses (projections) et d'ellipsoïdes (objets 3D). En effet, ce paradigme de modélisation permet d'obtenir une expression formelle liant l'ellipse et l'ellipsoïde, et nous verrons dans la

suite qu'il permet également d'obtenir une expression formelle de la pose de la caméra. De plus, puisqu'il est possible d'associer une ellipse à une boîte englobante rectangulaire, ce choix de modélisation ne nécessite pas d'entrainement spécifique à un objet, mais peut être utilisé en association avec n'importe quel détecteur d'objet générique [RDGF16, RHGS15, LAE+16].

Le reste de ce mémoire est donc consacré à l'estimation de pose de caméra à partir de correspondances entre ellipses et ellipsoïdes. Le chapitre 3 résume nos contributions à la résolution de ce problème. Le chapitre 4 présente une méthode d'estimation de pose de caméra dans laquelle la position de la caméra est calculée à partir de son orientation et d'une unique correspondance. Le chapitre 5 s'intéresse principalement à la détermination théorique de l'ensemble des caméras satisfaisant une correspondance ellipse - ellipsoïde, et montre que ces résultats sont difficiles à exploiter en pratique. Enfin, le chapitre 6 s'intéresse au calcul de la pose complète de la caméra à partir de deux correspondances ellipse - ellipsoïde.

Chapitre 3

Estimation de pose de caméra à partir de paires ellipse - ellipsoïde : synthèse des contributions

Sommaire

3.1 Problème à un ellipsoïde 53
3.1.1 Stratégie de résolution 53
3.1.2 Formulation 54
3.1.3 Synthèse des résultats 56
3.2 Problème à N ellipsoïdes 56
3.2.1 Détermination de la position de la caméra connaissant son orientation . 56
3.2.2 Estimation de la pose complète de la caméra 56

Nous proposons dans ce chapitre une synthèse des contributions apportées au problème d'estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde, afin de faciliter la lecture de la suite du manuscrit, et en particulier des développements techniques.

Pour aborder ce problème, nous nous sommes tout d'abord intéressés au cas où le modèle de scène est constitué d'un unique ellipsoïde. Etant donné sa projection dans une image prise depuis un point de vue inconnu, nous avons cherché à déterminer théoriquement l'ensemble des poses de caméras permettant d'aboutir à cette projection. La formulation du problème et la synthèse de nos contributions sont présentées dans la partie 3.1. Nous avons ensuite cherché à exploiter ces résultats pour résoudre le problème à N ellipsoïdes. Nos contributions sont cette fois résumées dans la partie 3.2.

3.1 Problème à un ellipsoïde

3.1.1 Stratégie de résolution

Nous considérons un repère global, dit *repère monde*, dans lequel les paramètres de l'ellipsoïde sont connus, et un repère local attaché à la caméra, dans lequel l'ellipse est entièrement déterminée. L'ellipse est la projection de l'ellipsoïde si et seulement si le cône de sommet le centre optique de la caméra et de directrice l'ellipse est tangent à l'ellipsoïde [HZ04, Ebe99, Ebe07]. Or les paramètres de ce cône sont connus dans le repère caméra, puisqu'ils dépendent uniquement de l'ellipse (directrice du cône) et des paramètres de la projection (paramètres intrinsèques de la

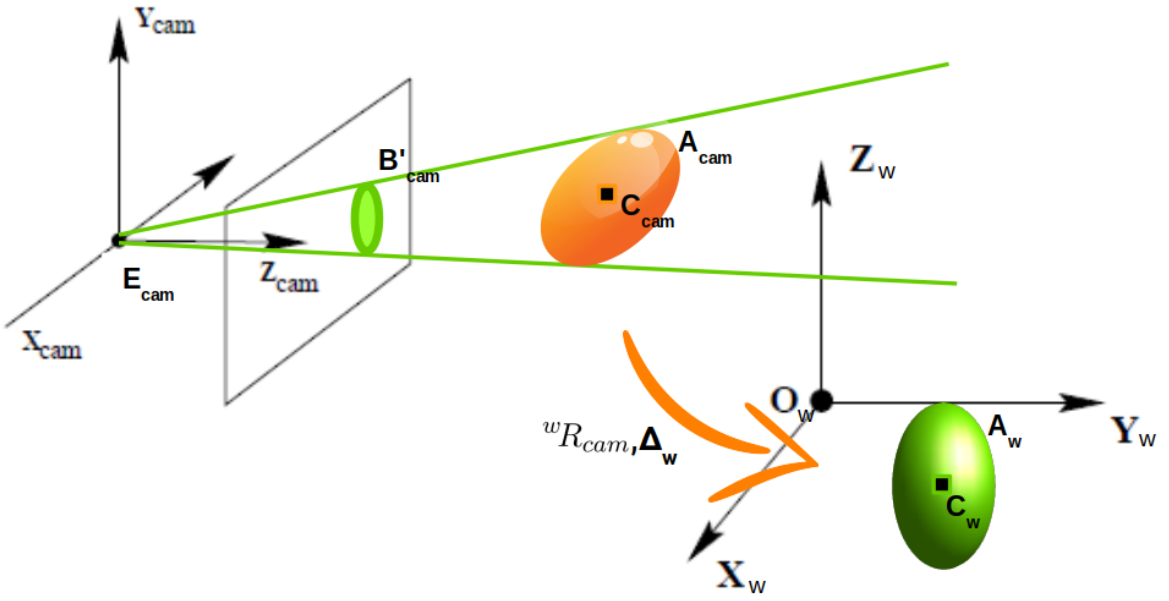


FIGURE 3.1 – Illustration du problème de détermination des caméras satisfaisant une correspondance ellipse - ellipsoïde : nous commençons par déterminer les ellipsoïdes tangents au cône dans le repère caméra, puis en déduisant les transformations possibles entre le repère caméra et le repère monde.

caméra). Pour résoudre le problème, une solution, illustrée en figure 3.1, consiste à déterminer l'ensemble des ellipsoïdes de taille fixée (correspondant à celle du modèle) tangents au cône dans le repère caméra, puis à déterminer l'ensemble des déplacements 3D rigides permettant de passer des ellipsoïdes exprimés dans le repère caméra à l'ellipsoïde exprimé dans le repère monde. Faisant cela, nous déterminons l'ensemble des transformations possibles entre le repère monde et le repère caméra, c'est-à-dire l'ensemble des poses des caméras solutions. Cependant, nous montrons qu'il est parfois préférable, selon le type d'ellipsoïde considéré, de déterminer plutôt l'ensemble des cônes de forme fixée (correspondant à celle du cône dans le repère caméra) tangents à l'ellipsoïde, donnant ainsi l'ensemble des positions de caméras, avant d'en déduire leurs orientations.

La détermination des ellipsoïdes tangents au cône ou des cônes tangents à l'ellipsoïde se fait par résolution de l'Equation d'Alignement des Cônes (2.6), présentée dans le chapitre 2 (partie 2.3), qui traduit cette tangence.

3.1.2 Formulation

N. B. : Toutes les bases vectorielles considérées sont orthonormées directes, de sorte que n'importe quelle matrice de passage ${}^jR_i$ d'une base $\mathscr{B}_i$ vers une base $\mathscr{B}_j$ est une matrice de rotation. On appellera, par ailleurs, M_i l'expression d'une matrice M dans une base $\mathscr{B}_i$, et $\boldsymbol{X}_i$ l'expression d'un vecteur $\boldsymbol{X}$ dans la même base, de sorte que $M_j = {}^jR_i M_i {}^jR_i^\top$ et $\boldsymbol{X}_j = {}^jR_i \boldsymbol{X}_i$. De plus, on notera $\boldsymbol{P_i}$ l'expression d'un point $\boldsymbol{P}$ dans un repère $\mathscr{R}_i$.

Le problème est illustré en figure 3.1. On considère un repère global $\mathscr{R}_w$, dit *repère monde*, constitué d'une origine O et d'une base vectorielle $\mathscr{B}_w$, ainsi qu'un repère local attaché à la caméra : $\mathscr{R}_{cam} = (\boldsymbol{E}; \mathscr{B}_{cam})$, dont l'origine E est le centre optique de la caméra. Déterminer la

pose de la caméra consiste alors à déterminer sa position dans le repère monde E_w ainsi que son orientation par rapport à ce même repère ${}^wR_{cam}$.

On considère par ailleurs l'ellipsoïde de centre $\boldsymbol{C}$ et de forme quadratique A, dont on suppose que l'on connait les expressions $\boldsymbol{C}_w$ et A_w dans la base monde (modèle de scène). Parallèlement, on considère le cône de sommet $\boldsymbol{E}$ et de directrice l'ellipse inscrite dans l'image, et on appelle B' sa forme quadratique, de telle sorte que l'on connait leurs expressions $\boldsymbol{E}_{cam}$ et B'_{cam} dans la base caméra. Enfin, on appelle $\boldsymbol{\Delta} = \boldsymbol{E} - \boldsymbol{C}$ le vecteur reliant le centre de l'ellipsoïde $\boldsymbol{C}$ au centre de la caméra $\boldsymbol{E}$.

Notre stratégie consiste alors à déterminer soit l'ensemble des ellipsoïdes $(\boldsymbol{C}_{cam}, A_{cam})$ tangents au cône $(\boldsymbol{E}_{cam}, B'_{cam})$ dans le repère caméra, soit l'ensemble des cônes $(\boldsymbol{E}_w, B'_w)$ tangents à l'ellipsoïde $(\boldsymbol{C}_w, A_w)$ dans le repère monde. En pratique, la détermination des positions $\boldsymbol{C}_{cam}$ ou $\boldsymbol{E}_w$ se fait par l'intermédiaire des vecteurs $\boldsymbol{\Delta}$ ($\boldsymbol{\Delta}_{cam}$ ou $\boldsymbol{\Delta}_w$), ce qui est possible car une des extrémités du vecteur est systématiquement connue. Cette détermination se fait par la résolution de l'Equation d'Alignement des Cônes (2.6), grâce à la connaissance des valeurs propres de A et B'.

Dans la majorité des cas rencontrés en pratique (ellipsoïde non dégénéré, ou sphéroïde dont l'axe de révolution n'est pas aligné avec celui du cône), il est possible de déterminer l'ensemble des ellipsoïdes $(\boldsymbol{\Delta}_{cam}, A_{cam})$ tangents au cône de rétroprojection, reproduisant ainsi la stratégie proposée par [WP10]. Les orientations des caméras solutions sont alors les matrices de rotations ${}^wR_{cam}$ qui vérifient l'égalité

$$A_{cam} = {}^wR_{cam}^{\top} A_w {}^wR_{cam}$$

Si l'ellipsoïde est non dégénéré, alors l'égalité précédente est vérifiée par seulement huit matrices de changement de base (l'image de chaque vecteur propre est définie au signe près), qui se réduit à quatre en se limitant aux matrices de rotation. Si, à l'inverse, l'ellipsoïde est dégénéré, alors il existe une infinité d'orientations de caméras ${}^wR_{cam}$ compatibles avec chaque paire (A_{cam}, A_w). Les positions de caméras $\boldsymbol{E}_w$ viennent ensuite par les opérations

$$\boldsymbol{\Delta}_w = {}^wR_{cam}\boldsymbol{\Delta}_{cam}$$

puis

$$\boldsymbol{E}_w = \boldsymbol{C}_w + \boldsymbol{\Delta}_w.$$

Dans le cas où l'ellipsoïde est non dégénéré, il est possible de déterminer l'ensemble des cônes $(\boldsymbol{\Delta}_w, B'_w)$ tangents à ce dernier. Les orientations des caméras solutions sont alors les matrices de changement de base qui vérifient l'égalité

$$B'_w = {}^wR_{cam} B'_{cam} {}^wR_{cam}^{\top}$$

Comme le cône de rétroprojection est lui aussi non dégénéré dans ce cas, B' possède trois valeurs propres distinctes. Il n'existe alors que deux matrices de rotation ${}^wR_{cam}$ qui permettent de vérifier l'égalité précédente tout en respectant la contrainte de chiralité (ellipsoïde situé face à la caméra). Les positions des caméras solutions sont ensuite déduites par

$$\boldsymbol{E}_w = \boldsymbol{C}_w + \boldsymbol{\Delta}_w.$$

La principale différence entre les deux méthodes vient des matrices qu'on cherche à décomposer en éléments propres. Dans le premier cas, les valeurs propres de la matrice A sont données par le modèle, tandis que dans le second cas (B'), elles dépendent de l'ellipse détectée dans l'image et sont donc potentiellement bruitées. Ceci peut éventuellement entrainer une robustesse plus faible en pratique.

3.1.3 Synthèse des résultats

Nos contributions à la résolution du problème à un ellipsoïde sont doubles :

1. nous avons montré qu'il existe un découplage entre les orientation et position relatives de l'ellipsoïde et du cône de rétroprojection, ou, plus précisément, que leur position est entièrement déterminée par leur orientation. Ce découplage nous a permis de montrer que la position d'une caméra peut être déduite de son orientation et d'une unique correspondance ellipse - ellipsoïde (voir chapitre 4) ;
2. nous avons déterminé l'ensemble des solutions de l'Equation d'Alignement des Cônes, et en avons déduit l'ensemble des poses possibles. Nous avons pour cela distingué les différents types d'ellipsoïdes. Nous avons montré, en particulier, qu'il existe une infinité d'ellipsoïdes non dégénérés de taille fixée tangents à un cône donné, alors que, comme il a déjà été démontré dans [WP10], il n'existe que deux sphéroïdes de taille fixée vérifiant cette propriété (voir figures 3.2a et 3.2b). Dans le premier cas, l'infinité des ellipsoïdes tangents au cône (ou réciproquement l'infinité des cônes tangents à l'ellipsoïde) explique l'infinité des caméras solutions (cf. figure 3.2c), et en offre une paramétrisation. Dans le second cas, c'est l'infinité des matrices de changement de base permettant de passer d'un ellipsoïde dans le repère caméra à l'ellipsoïde dans le repère monde qui explique l'infinité des caméras solutions (cf. figure 3.2d). Les développements amenant à ces résultats sont présentés au chapitre 5.

3.2 Problème à N ellipsoïdes

Nous nous sommes ensuite intéressés au problème d'estimation de pose de caméra à partir de N correspondances ellipse - ellipsoïde ($N > 1$). Nous avons développé deux méthodes dont les principaux intérêts sont présentés ci-après.

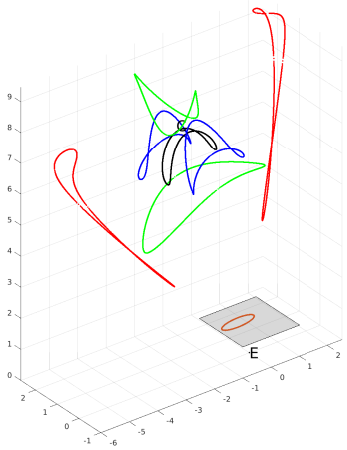
3.2.1 Détermination de la position de la caméra connaissant son orientation

Le découplage entre orientation et position nous a permis de développer une méthode d'estimation de pose de caméra dans laquelle la position de la caméra est calculée à partir de son orientation et d'au moins une correspondance ellipse - ellipsoïde. Nous avons, de plus, proposé une manière robuste de traiter les multiples appariements possibles entre les objets détectés dans l'image et les objets présents dans le modèle 3D de la scène. Dans ce travail, nous avons obtenu une précision importante de la méthode malgré le fait que les ellipses considérées, qui sont inscrites dans les boîtes de détection, ne correspondent en général pas précisément aux projections des ellipsoïdes dans l'image. Cette méthode est présentée au chapitre 4.

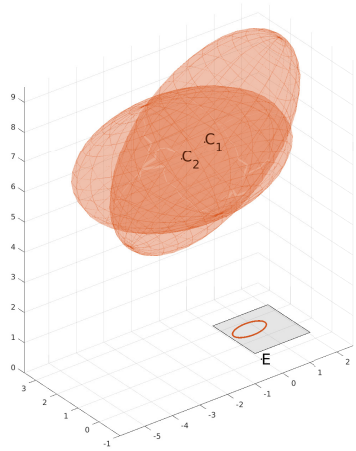
3.2.2 Estimation de la pose complète de la caméra

Nous avons, dans un premier temps, tenté d'exploiter la paramétrisation des solutions du problème à un ellipsoïde, en déterminant, pour chaque correspondance 2D - 3D, les lieux des caméras solutions, avant de les intersecter. Cette méthode, si elle fonctionne en théorie, se révèle extrêmement sensible aux perturbations rencontrées sur des cas réels, et s'avère ainsi difficilement exploitable en pratique (voir chapitre 5).

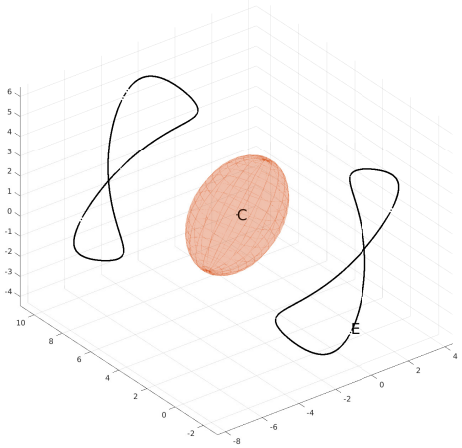
Pour palier à ce problème, nous avons également développé une méthode permettant d'estimer l'orientation de la caméra à partir d'au moins deux correspondances ellipse - ellipsoïde, qui est basée sur une hypothèse simplificatrice quasiment toujours vérifiée en pratique. Cette méthode tire également profit du découplage orientation - position pour en déduire la position de la



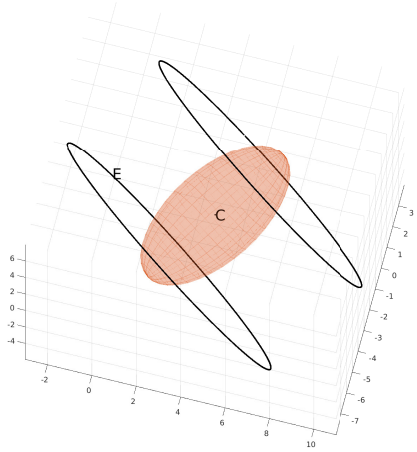
(a) Lieux des centres (noir) et des sommets des axes principaux (rouge, vert, bleu) des ellipsoïdes tangents au cône de rétroprojection.



(b) Sphéroïdes tangents au cône de rétroprojection.



(c) Lieu des sommets des cônes tangents à l'ellipsoïde, *i. e.* lieu des centres de caméras.



(d) Lieu des sommets des cônes tangents au sphéroïde, *i. e.* lieu des centres de caméras.

FIGURE 3.2 – Aperçu des solutions du problème de détermination des caméras satisfaisant une correspondance ellipse - ellipsoïde, lorsque l'ellipsoïde est non dégénéré (colonne de gauche), et lorsqu'il possède un axe de révolution (colonne de droite).

caméra. Là encore, un mécanisme permet de gérer les multiples hypothèses d'appariement entre les ellipses et les ellipsoïdes, tout en estimant la pose complète de la caméra. La principale source d'imprécision provient de l'utilisation des ellipses inscrites dans le boîtes comme approximation des projections d'ellipsoïdes. Cette méthode est présentée au chapitre 6.

Chapitre 4

Découplage orientation - position et application à la relocalisation de caméra

Sommaire

4.1 Eléments mathématiques 60
4.1.1 Discriminant et relations entre coefficients et racines d'un polynôme de degré 3 60
4.1.2 Décomposition d'une matrice en éléments propres 60
4.1.3 Problème aux valeurs propres généralisé 61
4.2 Découplage entre orientation et position 62
4.2.1 Formulations équivalente du problème 62
4.2.2 Expression de la position connaissant l'orientation 63
4.3 Méthode robuste d'estimation de la pose 67
4.3.1 Construction du modèle 67
4.3.2 Procédure de type RANSAC pour l'estimation de la position 68
4.3.3 Estimation de l'orientation 68
4.4 Résultats 70
4.4.1 Précision de la pose avec un objet 70
4.4.2 Scénarios réels 73
4.4.3 Discussion 79
4.5 Conclusion 80

Dans ce chapitre, nous reprenons le problème d'estimation de pose de caméra à partir d'une unique correspondance ellipse - ellipsoïde, formulé au chapitre précédent, et montrons que la position de la caméra est entièrement déterminée par son orientation. Après avoir présenté certains éléments mathématiques utiles à la compréhension des développements (partie 4.1), nous mettons en évidence ce résultat théorique (partie 4.2), puis en tirons une méthode robuste d'estimation de pose de caméra opérant à partir des détections d'objets dans l'image (partie 6.2). Cette méthode est finalement évaluée sur des bases d'images standard pour la communauté (partie 4.4). Les travaux décrits dans ce chapitre ont fait l'objet des publications [GSB19c] et [GSB19d] (cf page 7).

4.1 Eléments mathématiques

4.1.1 Discriminant et relations entre coefficients et racines d'un polynôme de degré 3

On considère le polynôme de degré 3 $P(x) = ax^3 + bx^2 + cx + d$, avec $(a, b, c, d) \in \mathbb{R}^4$. Le discriminant du polynôme P est donné par la formule :

$$D = 18abcd - 4b^3d + b^2c^2 - 4ac^3 - 27a^2d^2$$

Le lien entre les racines du polynôme et le signe de son discriminant est le suivant :

- $D > 0$: P possède trois racines réelles disctinctes,
- $D = 0$: P possède une ou deux racines réelles disctinctes,
- $D < 0$: P possède trois racines distinctes, dont une réelle et deux complexes conjuguées.

Si on appelle x_1, x_2, x_3 les racines du polynôme, alors elles vérifient les relations suivantes :

$$\begin{cases} x_1 + x_2 + x_3 = -\frac{b}{a} \\ x_1x_2 + x_1x_3 + x_2x_3 = \frac{c}{a} \\ x_1x_2x_3 = -\frac{d}{a} \end{cases}$$

4.1.2 Décomposition d'une matrice en éléments propres

Dans ce travail, nous ne nous intéressons qu'à la diagonalisation des matrices réelles sur le corps $\mathbb{R}$ des réels, donc, sauf mention contraire, les éléments que nous manipulons (coefficients de vecteurs, de matrice et de polynômes, ainsi que valeurs propres et racines de polynômes) sont réels. Nous rappelons ci dessous un ensemble de résultats d'algèbre linéaire qui seront utilisés dans ce travail.

Soit $M \in \mathbb{R}^{n\times n}$ une matrice carrée de taille n représentant un endomorphisme.

Valeurs propres et vecteurs propres

Un scalaire λ est une valeur propre de M s'il existe un vecteur $\boldsymbol{X}$ non nul tel que $M\boldsymbol{X} = \lambda\boldsymbol{X}$. $\boldsymbol{X}$ est appelé vecteur propre associé à λ.

Les valeurs propres de M sont donc les scalaires λ tels que $Ker(\lambda I_n - M) \neq \{\mathbf{0}\}$, ce qui est équivalent à $det(\lambda I_n - M) = 0$.

Le polynôme

$$P_M(x) = det(xI_n - M)$$

où I_n est la matrice identité de taille n est appelé polynôme caractéristique de M. Il est ainsi de degré n et ses racines sont les valeurs propres de M.

Polynômes annulateurs

D'après le théorème de Cayley-Hamilton, tout endomorphisme annule son polynôme caractéristique. Donc $P_M(M) = 0_n$, où 0_n est la matrice nulle de taille n.

Le polynôme minimal, $\pi_M(x)$, est défini comme l'unique polynôme qui annule M et divise tous les polynômes annulateurs de M. Le polynome minimal divise donc le polynôme caractéristique, et on peut montrer que ses racines sont les valeurs propres de l'endomorphisme.

Lien avec la diagonalisation des matrices

Une matrice est diagonalisable s'il existe une base dans laquelle sa représentation est une matrice diagonale. Une matrice n'est diagonalisable que si la dimension de chaque sous-espace propre E_{λ_i} est égale à la multiplicité m_i de la valeur propre λ_i.

Une condition nécessaire pour que la matrice M soit diagonalisable est que son polynôme caractéristique soit scindé, *i. e.* qu'il s'écrive comme produit de polynômes de degré 1. Dans ce cas, le polynôme caractéristique s'écrit :

$$P_M(x) = \prod_{\lambda_i} (x - \lambda_i)^{m_i}$$

Une matrice est diagonalisable si et seulement si son polynôme minimal est scindé à racines simples. Il s'écrit alors

$$\pi_M(x) = \prod_{\lambda_i} (x - \lambda_i)$$

Cas des matrices symétriques

Toute matrice symétrique à coefficients réels est diagonalisable à l'aide d'une matrice de passage orthogonale et ses sous-espaces propres sont orthogonaux.

4.1.3 Problème aux valeurs propres généralisé

Soit $M, N \in \mathbb{R}^{n \times n}$ deux matrices réelles carrées. Le problème aux valeurs propres généralisé relatif au couple de matrices $\{M, N\}$ consiste à trouver l'ensemble des scalaires $\lambda \in \mathbb{R}$ et des vecteurs $\boldsymbol{X} \in \mathbb{R}^n$ qui vérifient

$$M\boldsymbol{X} = \lambda N\boldsymbol{X}$$

De tels scalaires λ sont appelés les valeurs propres généralisées du couple $\{M, N\}$, et les vecteurs $\boldsymbol{X}$ associés leurs vecteurs propres généralisés.

Si la matrice N est inversible, le problème se rapporte à un problème aux valeurs propres classique :

$$N^{-1}M\boldsymbol{X} = \lambda\boldsymbol{X}$$

Si, de plus, la matrice M est également inversible, alors la matrice $N^{-1}M$ l'est aussi, et donc les valeurs propres λ sont non nulles.

Lorsque les valeurs propres généralisées λ du couple $\{M, N\}$ sont non nulles, il apparaît de manière évidente que les scalaires $1/\lambda$ sont les valeurs propres généralisées du couple $\{N, M\}$ associées aux mêmes vecteurs propres généralisés :

$$N\boldsymbol{X} = \frac{1}{\lambda} M\boldsymbol{X}$$

Il est, par ailleurs, possible de définir le polynôme caractéristique du couple de la manière suivante :

$$P_{\{M,N\}}(x) = det(M - xN)$$

Les racines de ce polynôme sont ainsi les valeurs propres généralisées du couple $\{M, N\}$.

Lorsque la matrice M est symétrique et que la matrice N est symétrique définie positive, alors le couple $\{M, N\}$ est dit *défini symétrique*. Un tel couple possède les propriétés suivantes [GVL96] :

1. Les valeurs propres généralisées du couple sont réelles,
2. Chaque sous-espace propre généralisé est de dimension égale à la multiplicité de la valeur propre associée,
3. Les vecteurs propres généralisés forment une base de $\mathbb{R}^3$, et ceux associés à des valeurs propres différentes sont N-orthogonaux.

D'autre part, les valeurs propres généralisées λ d'un couple défini symétrique vérifient l'inégalité suivante [ANT08] (théorème 3)

$$\forall \boldsymbol{X} \in \mathbb{R}^3 \backslash \{\mathbf{0}\}, \;\; \min \lambda \leq \frac{\boldsymbol{X}^\top M \boldsymbol{X}}{\boldsymbol{X}^\top N \boldsymbol{X}} \leq \max \lambda$$

4.2 Découplage entre orientation et position

Nous considérons le problème à un ellipsoïde (chapitre 3, partie 3.1). Nous cherchons donc à déterminer les ellipsoïdes de taille fixée tangents au cône de rétroprojection ou les cônes de forme fixée tangents à l'ellipsoïde. Dans cette partie, nous montrons que leur position relative est en fait déterminée par leur orientation relative.

4.2.1 Formulations équivalente du problème

Tout d'abord, nous montrons que l'Equation d'Alignement des Cônes (2.6) peut être reformulée d'au moins deux manières différentes, qui montrent bien la *symétrie* des rôles joués par l'ellipsoïde et le cône.

Théorème 1. *Les équations* (2.6)*,* (4.1) *et* (4.2) *sont équivalentes.*

$$A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A + \mu A = \sigma B' \tag{2.6}$$

$$\boldsymbol{\Delta}\boldsymbol{\Delta}^\top = A^{-1} - \frac{\mu}{\sigma} B'^{-1} \tag{4.1}$$

$$B'\boldsymbol{\Delta}\boldsymbol{\Delta}^\top B' - \frac{1}{\sigma} B' = -\frac{\mu}{\sigma^2} A \tag{4.2}$$

avec $\mu = 1 - \boldsymbol{\Delta}^\top A \boldsymbol{\Delta}$.

Démonstration. Nous montrons l'équivalence entre les équations (2.6) et (4.1). Le même type de raisonnement peut être utilisé pour montrer l'équivalence avec l'équation (4.2).

$\boxed{\Longrightarrow}$ Supposons que l'équation (2.6) soit vérifiée. Alors, d'après le résultat 1, on a

$$A\boldsymbol{\Delta} = \sigma B' \boldsymbol{\Delta}$$

En injectant ce résultat dans l'équation (2.6) on obtient

$$\sigma B' \boldsymbol{\Delta}\boldsymbol{\Delta}^\top A = \sigma B' - \mu A$$

Puis, en multipliant à gauche par B'^{-1}

$$\sigma \boldsymbol{\Delta}\boldsymbol{\Delta}^\top A = \sigma I - \mu B'^{-1} A$$

et en multipliant ensuite à droite par A^{-1}

$$\sigma \boldsymbol{\Delta}\boldsymbol{\Delta}^\top = \sigma A^{-1} - \mu B'^{-1}$$

Finalement ($\sigma \neq 0$ par définition)

$$\boldsymbol{\Delta}\boldsymbol{\Delta}^\top = A^{-1} - \frac{\mu}{\sigma} B'^{-1}$$

$\boxed{\Longrightarrow}$ Supposons maintenant que l'équation (4.1) soit vérifiée. Multiplier (4.1) à droite par A donne

$$\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A = I - \frac{\mu}{\sigma} B'^{-1} A$$

Puis, en multipliant à gauche par $\sigma B'$

$$\sigma B' \boldsymbol{\Delta}\boldsymbol{\Delta}^\top A = \sigma B' - \mu A \tag{4.3}$$

Si on multiplie ensuite à droite par $\boldsymbol{\Delta}$ on obtient

$$\mu A \boldsymbol{\Delta} = \sigma B' \boldsymbol{\Delta} - \sigma B' \boldsymbol{\Delta}\boldsymbol{\Delta}^\top A \boldsymbol{\Delta}$$

D'où, puisque $\boldsymbol{\Delta}^\top A \boldsymbol{\Delta}$ est un scalaire

$$\begin{aligned} \mu A \boldsymbol{\Delta} &= \sigma B' \boldsymbol{\Delta} - \sigma B' \boldsymbol{\Delta} (\boldsymbol{\Delta}^\top A \boldsymbol{\Delta}) \\ &= \sigma B' \boldsymbol{\Delta} - \sigma (\boldsymbol{\Delta}^\top A \boldsymbol{\Delta}) B' \boldsymbol{\Delta} \\ &= \sigma (1 - \boldsymbol{\Delta}^\top A \boldsymbol{\Delta}) B' \boldsymbol{\Delta} \\ &= \sigma \mu B' \boldsymbol{\Delta} \end{aligned}$$

On retrouve alors l'équation (2.7) en divisant par μ ($\mu \neq 0$ car le centre de projection $\boldsymbol{E}$ n'appartient pas à l'ellipsoïde : cf Partie 2.3.5) :

$$A \boldsymbol{\Delta} = \sigma B' \boldsymbol{\Delta}$$

On peut alors injecter ce résultat dans (4.3) pour obtenir

$$A \boldsymbol{\Delta}\boldsymbol{\Delta}^\top A + \mu A = \sigma B'$$

□

L'une ou l'autre de ces équations équivalentes pourra être utilisée dans la suite du manuscrit pour traduire le problème à résoudre.

4.2.2 Expression de la position connaissant l'orientation

L'équation matricielle (2.6) est indépendante de la base vectorielle dans laquelle les éléments sont exprimés, et peut donc être résolue dans n'importe quelle base. Lorsque les matrices A et B' sont exprimées dans une base commune, le couple $\{A, B'\}$ caractérise l'orientation relative entre le cône et l'ellipsoïde. Dans cette partie, nous montrons que l'expression des inconnues σ, μ et $\boldsymbol{\Delta}$ est entièrement déterminée par celle du couple $\{A, B'\}$. Ainsi, la position relative entre le cône et l'ellipsoïde est déductible de leur orientation relative.

Résultat 2. *Si le couple $\{A, B'\}$ est solution de l'équation* (2.6)*, alors il possède exactement deux valeurs propres généralisées distinctes, qui sont non nulles et de signes opposés. En notant σ_1 la valeur propre de multiplicité 1 et σ_2 celle de multiplicité 2, les autres inconnues de l'équation* (2.6) *sont données par*

$$\sigma = \sigma_1$$
$$\mu = \frac{\sigma_1}{\sigma_2}$$
$$\boldsymbol{\Delta} = \pm\sqrt{k^2}\boldsymbol{\delta}, \text{ avec } \begin{cases} \boldsymbol{\delta} \text{ un vecteur propre généralisé du couple } \{A, B'\} \text{ associé à } \sigma \\ ||\boldsymbol{\delta}|| = 1 \\ k^2 = tr(A^{-1}) - \frac{1}{\sigma_2} tr(B'^{-1}) \end{cases}$$

avec une seule des deux valeurs de $\boldsymbol{\Delta}$ qui permet de définir un ellipsoïde situé face à la caméra.

Démonstration. Valeurs propres généralisées de $\{A, B'\}$ Nous allons montrer que le couple $\{A, B'\}$ possède exactement deux valeurs propres généralisées distinctes, qui sont non nulles et de signes opposés.

Puisque les deux matrices sont inversibles, la matrice $B'^{-1}A$ l'est également, donc ses valeurs propres, qui sont confondues avec **les valeurs propres généralisées de $\{A, B'\}$, sont non nulles**.

Nous allons maintenant montrer qu'elles sont au nombre de deux, et de signes opposés. Supposons qu'il existe un vecteur $\mathbf{\Delta}$ et un scalaire non nul σ tels que le quadruplet $(A, B', \mathbf{\Delta}, \sigma)$ vérifie l'équation (2.6). Alors, d'après le résultat 1, ce même quadruplet vérifie aussi l'équation (2.7) :

$$A\mathbf{\Delta} = \sigma B'\mathbf{\Delta}$$

Trouver les paires $(\sigma, \mathbf{\Delta})$ qui satisfont cette seconde équation consiste donc à trouver les valeurs et vecteurs propres généralisés du couple $\{A, B'\}$. En particulier, les valeurs propres généralisées sont les racines du polynôme caractéristique $P_{\{A,B'\}}(x) = det(A - xB')$, qui sont confondues avec celles du polynôme caractéristique de la matrice $B'^{-1}A$. Or, on peut remarquer que le polynôme

$$Q(x) = \mu x^2 - (\mu + 1)\sigma x + \sigma^2$$

est un polynôme annulateur de $B'^{-1}A$. En effet, en injectant (2.7) dans (2.6), on obtient :

$$\sigma^2 B'\mathbf{\Delta}\mathbf{\Delta}^\top B' + \mu A = \sigma B'$$

On peut alors déduire l'expression suivante pour A :

$$A = \frac{\sigma}{\mu}(B' - \sigma B'\mathbf{\Delta}\mathbf{\Delta}^\top B')$$

Puis, en multipliant l'équation précédente à gauche par B'^{-1},

$$B'^{-1}A = \frac{\sigma}{\mu}(I - \sigma\mathbf{\Delta}\mathbf{\Delta}^\top B')$$

Elever cette expression au carré conduit à

$$\begin{aligned}
(B'^{-1}A)^2 &= \frac{\sigma^2}{\mu^2}(I - \sigma\mathbf{\Delta}\mathbf{\Delta}^\top B')^2 \\
&= \frac{\sigma^2}{\mu^2}(I - 2\sigma\mathbf{\Delta}\mathbf{\Delta}^\top B' + \sigma^2\mathbf{\Delta}\mathbf{\Delta}^\top B'\mathbf{\Delta}\mathbf{\Delta}^\top B') \\
&= \frac{\sigma^2}{\mu^2}(I - 2\sigma\mathbf{\Delta}\mathbf{\Delta}^\top B' + \sigma^2\mathbf{\Delta}(\mathbf{\Delta}^\top B'\mathbf{\Delta})\mathbf{\Delta}^\top B') \\
&= \frac{\sigma^2}{\mu^2}(I - 2\sigma\mathbf{\Delta}\mathbf{\Delta}^\top B' + \sigma^2(\mathbf{\Delta}^\top B'\mathbf{\Delta})\mathbf{\Delta}\mathbf{\Delta}^\top B') \\
&= \frac{\sigma^2}{\mu^2}(I - \sigma(2 - \sigma\mathbf{\Delta}^\top B'\mathbf{\Delta})\mathbf{\Delta}\mathbf{\Delta}^\top B')
\end{aligned}$$

Puis, observant que $\mu = 1 - \mathbf{\Delta}^\top A\mathbf{\Delta} = 1 - \sigma\mathbf{\Delta}^\top B'\mathbf{\Delta}$

$$\begin{aligned}
(B'^{-1}A)^2 &= \frac{\sigma^2}{\mu^2}(I - \sigma(\mu + 1)\mathbf{\Delta}\mathbf{\Delta}^\top B') \\
&= \frac{\sigma^2}{\mu^2}((\mu + 1)(I - \sigma\mathbf{\Delta}\mathbf{\Delta}^\top B') - \mu I) \\
&= \frac{\sigma}{\mu}(\mu + 1)B'^{-1}A - \frac{\sigma^2}{\mu}I
\end{aligned}$$

Finalement, on a bien

$$\mu(B'^{-1}A)^2 = \sigma(\mu+1)B'^{-1}A - \sigma^2 I \tag{4.4}$$

i. e.

$$Q(B'^{-1}A) = 0$$

où 0 est ici la matrice nulle.

Le polynôme minimal $\pi_{B'^{-1}A}(x)$ divisant tout polynôme annulateur de $B'^{-1}A$, on en déduit qu'il est au plus de degré 2, donc que la matrice $B'^{-1}A$ ne possède qu'au plus deux valeurs propres distinctes. Il en est de même, a fortiori, pour les couples $\{A, B'\}$ et $\{B', A\}$.

Puisque le couple $\{B', A\}$ est défini symétrique, chacun de ses sous-espaces propres généralisés est de dimension égale à la multiplicité de la valeur propre associée, d'où, s'il ne possédait qu'une seule valeur propre de multiplicité 3 (disons $1/\sigma_0$), alors on aurait $dim(Ker(B' - \frac{1}{\sigma_0}A)) = 3$, c'est-à-dire $B' = \frac{1}{\sigma_0}A$, ce qui est impossible car A représente un ellipsoïde tandis que B' représente un cône. Donc **le couple $\{\boldsymbol{A}, \boldsymbol{B'}\}$ possède exactement deux valeurs propres généralisées distinctes.**

Appelons ensuite σ_1 (multiplicité 1) et σ_2 (multiplicité 2) ces deux valeurs propres. Puisque $\frac{1}{\sigma_1}$ et $\frac{1}{\sigma_2}$ sont les valeurs propres généralisées du couple $\{B', A\}$, on peut écrire, d'après [ANT08] (théorème 3)

$$\forall \boldsymbol{X} \in \mathbb{R}^3 \backslash \{\boldsymbol{0}\}, \quad \min\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}\right) \leq \frac{\boldsymbol{X}^\top B' \boldsymbol{X}}{\boldsymbol{X}^\top A \boldsymbol{X}} \leq \max\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}\right)$$

Si σ_1 et σ_2 étaient de même signe, alors $\forall \boldsymbol{X} \in \mathbb{R}^3 \backslash \{\boldsymbol{0}\}$, $\boldsymbol{X}^\top B' \boldsymbol{X}$ serait de ce signe (car $\boldsymbol{X}^\top A \boldsymbol{X} > 0$). Or B' n'est ni définie positive ni définie négative (cône), donc c'est impossible. Ainsi, on en déduit que **les deux valeurs propres généralisées sont de signes opposés.**

Valeur de σ Nous allons maintenant montrer que σ est nécessairement égal à σ_1, en montrant qu'il ne peut pas être égal à σ_2.

Raisonnons par l'absurde, et supposons qu'il existe $k \in \mathbb{R}^*$ tel que $(A, B', \sigma_2, k\boldsymbol{\delta}_2)$ vérifie l'équation (2.6), avec $\boldsymbol{\delta}_2$ un vecteur propre généralisé de $\{A, B'\}$ associé à σ_2. On a alors, en injectant ces inconnues dans l'équation (2.6) :

$$B' - \frac{1}{\sigma_2}A = MA \tag{4.5}$$

avec

$$M = \frac{k^2}{\sigma_2}(A\boldsymbol{\delta}_2\boldsymbol{\delta}_2^\top - \boldsymbol{\delta}_2^\top A\boldsymbol{\delta}_2 I)$$

Puisque $\frac{1}{\sigma_2}$ est une valeur propre généralisée du couple $\{B', A\}$ qui est défini symétrique, on a

$$dim(Ker(B' - \frac{1}{\sigma_2}A)) = 2$$

d'où, vu (4.5) et puisque A est inversible,

$$dim(Ker(MA)) = dim(Ker(M)) = 2$$

Cependant, en définissant $\boldsymbol{\delta}_2^\perp = \{\boldsymbol{X} \in \mathbb{R}^3 / \boldsymbol{X} \perp \boldsymbol{\delta}_2\}$ le sous-espace de dimension 2 orthogonal

à $\boldsymbol{\delta}_2$, on observe que

$$\begin{aligned}\forall \boldsymbol{X} \in \boldsymbol{\delta}_2^{\perp},\, M\boldsymbol{X} &= \frac{k^2}{\sigma_2} A\boldsymbol{\delta}_2\boldsymbol{\delta}_2^{\top}\boldsymbol{X} - \frac{k^2}{\sigma_2}\boldsymbol{\delta}_2^{\top} A\boldsymbol{\delta}_2\boldsymbol{X} \\ &= \frac{k^2}{\sigma_2} A\boldsymbol{\delta}_2(\boldsymbol{\delta}_2 \cdot \boldsymbol{X}) - \frac{k^2}{\sigma_2}\boldsymbol{\delta}_2^{\top} A\boldsymbol{\delta}_2\boldsymbol{X} \\ &= -\frac{k^2}{\sigma_2}\boldsymbol{\delta}_2^{\top} A\boldsymbol{\delta}_2\boldsymbol{X}\end{aligned}$$

Comme A est définie positive, $\boldsymbol{\delta}_2^{\top} A\boldsymbol{\delta}_2 > 0$, donc il vient que

$$\forall \boldsymbol{X} \in \boldsymbol{\delta}_2^{\perp}\backslash\{\mathbf{0}\},\, M\boldsymbol{X} \neq \mathbf{0}$$

Ainsi, les sous-espaces de $\mathbb{R}^3$ $\boldsymbol{\delta}_2^{\perp}$ et $Ker(M)$ sont en somme directe (*i. e.* $\boldsymbol{\delta}_2^{\perp} \cap Ker(M) = \{\mathbf{0}\}$). Donc le sous-espace résultant de cette somme est de dimension

$$\begin{aligned}dim(\boldsymbol{\delta}_2^{\perp} \oplus Ker(M)) &= dim(\boldsymbol{\delta}_2^{\perp}) + dim(Ker(M)) \\ &= 2 + 2 \\ &= 4\end{aligned}$$

Or c'est impossible puisque $dim(\mathbb{R}^3) = 3$.

Ainsi, les triplets $(A, \sigma_2, k\boldsymbol{\delta}_2)$ ne peuvent pas être solutions de l'équation (2.6), donc **les seules solutions possibles s'écrivent nécessairement $(\boldsymbol{A, \sigma_1, k\delta_1})$, avec $\boldsymbol{k} \in \mathbb{R}^*$, et $\boldsymbol{\delta}_1$ un vecteur propre généralisé de $\{A, B'\}$ associé à σ_1.**

Valeur de μ La trace de la matrice $B'^{-1}A$ s'écrit

$$tr(B'^{-1}A) = \sigma_1 + 2\sigma_2$$

et, en élevant la matrice au carré,

$$tr((B'^{-1}A)^2) = \sigma_1^2 + 2\sigma_2^2$$

Sachant que $tr(I) = 3$ et que $\sigma = \sigma_1$, on obtient alors, en appliquant $tr()$ au polynôme (4.4),

$$\mu(\sigma_1^2 + 2\sigma_2^2) - \sigma_1(\mu + 1)(\sigma_1 + 2\sigma_2) + 3\sigma_1^2 = 0$$

ce qui est équivalent à

$$\begin{aligned}&\mu\sigma_1^2 + 2\mu\sigma_2^2 - \mu\sigma 1^2 - 2\mu\sigma_1\sigma_2 - \sigma_1^2 - 2\sigma_1\sigma_2 + 3\sigma_1^2 = 0 \\ \iff &2(\mu\sigma_2^2 - \mu\sigma_1\sigma_2 + \sigma_1^2 - \sigma_1\sigma_2) = 0 \\ \iff &\mu\sigma_2(\sigma_2 - \sigma_1) = \sigma_1(\sigma_2 - \sigma_1) \\ \iff &\mu = \frac{\sigma_1}{\sigma_2}\end{aligned}$$

On en déduit que $\mu < 0$, puisque les valeurs propres sont de signes opposés (ce qui permet de retrouver que le centre de projection $\boldsymbol{E}$ est situé à l'extérieur de l'ellipsoïde).

Valeur de $\boldsymbol{\Delta}$ En sachant que $\sigma = \sigma_1$, $\mu = \frac{\sigma_1}{\sigma_2}$, et $\boldsymbol{\Delta} = k\boldsymbol{\delta}_1$, l'équation (4.1) (équivalente à l'équation (2.6) d'après le théorème 1) s'écrit

$$k^2\boldsymbol{\delta}_1\boldsymbol{\delta}_1^{\top} = A^{-1} - \frac{1}{\sigma_2}B'^{-1}$$

D'où, en appliquant $tr()$ et en choisissant $\boldsymbol{\delta}_1$ tel que $\|\boldsymbol{\delta}_1\| = 1$,

$$k^2||\boldsymbol{\delta}_1||^2 = k^2 = tr(A^{-1}) - \frac{1}{\sigma_2}tr(B'^{-1})$$

Les deux vecteurs $\pm\sqrt{k^2}\boldsymbol{\delta}$ définissent des centres d'ellipsoïdes $\boldsymbol{C}$ symétriques par rapport au centre de la caméra $\boldsymbol{E}$, donc le seul qui vérifie la contrainte de chiralité (ellipsoïde situé face à la caméra) est celui dont la coordonnée selon le vecteur $\boldsymbol{N}$ est négative. □

Le résultat 2 montre que les inconnues σ, μ et $\boldsymbol{\Delta}$ s'expriment de manière unique à partir de A et B'. Il existe donc un découplage entre l'orientation et la position de l'ellipsoïde ou du cône, et, a fortiori, entre l'orientation et la position de la caméra par rapport au repère monde. Le problème de détermination de sa pose complète (6 degrés de liberté) peut alors se limiter au problème de détermination de son orientation seulement (3 degrés de liberté). Plus précisément, si l'orientation ${}^wR_{cam}$ de la caméra est connue, alors il est possible d'exprimer la matrice B' dans la base monde :

$$B'_w = {}^wR_{cam}B'_{cam}{}^wR_{cam}^\top$$

puis d'en déduire σ et $\boldsymbol{\Delta}_w$ à partir du couple $\{A_w, B'_w\}$ en utilisant le résultat 2. Enfin, la position de la caméra $\boldsymbol{E}_w$ est calculée à partir de la position de l'ellipsoïde $\boldsymbol{C}_w$ et du vecteur $\boldsymbol{\Delta}_w$:

$$\boldsymbol{E}_w = \boldsymbol{C}_w + \boldsymbol{\Delta}_w$$

4.3 Méthode robuste d'estimation de la pose

Nous allons maintenant appliquer les résultats théoriques précédents pour construire une méthode d'estimation de pose de caméra à partir de N détections d'objets modélisés par des ellipses (2D) / ellipsoïdes (3D), dans laquelle la position de la caméra est déduite de son orientation.

Une liste d'associations possibles entre objets 2D et 3D, basée sur les compatibilités d'étiquettes, est l'entrée de notre procédure de pose. En raison d'erreurs de reconnaissance, certaines d'entre elles peuvent être erronées. De plus, comme les systèmes de reconnaissance sont principalement appris sur des catégories d'objets, il peut y avoir des ambiguïtés dans le choix de l'objet physique correspondant à la détection. L'apparition fréquente d'objets répétés dans des environnements artificiels est une autre source de fausses associations de données. Dans les travaux précédents [NMS19], l'association était réalisée manuellement pour éviter ce problème. Au contraire, nous proposons une procédure de type RANSAC dédiée à l'association robuste d'objets 2D - 3D dans le but d'éliminer automatiquement les fausses associations.

4.3.1 Construction du modèle

Notre système de relocalisation de caméra nécessite la connaissance d'un modèle 3D léger composé d'abstractions ellipsoïdales d'objets d'intérêt présents dans la scène. Pour reconstruire chaque objet, éventuellement séparément, quelques images avec des poses connues (3 au minimum [RCD18]) couvrant la plus large gamme possible de points de vue est nécessaire. Ces poses peuvent par exemple être obtenues grâce à des capteurs placés sur la caméra. Dans ces images, les objets sont automatiquement détectés (sous forme de boîtes rectangulaires) et étiquetés à l'aide d'un algorithme de détection d'objets (*e.g.* [HGDG17, LAE+16, RF18]), puis l'association entre les détections 2D des différentes images est effectuée manuellement. Pour chaque objet, les ellipses inscrites dans les boîtes détectées sont considérées comme des projections 2D d'un ellipsoïde 3D sous-jacent, reconstruit à l'aide de [RCD18]. Les étiquettes sont automatiquement

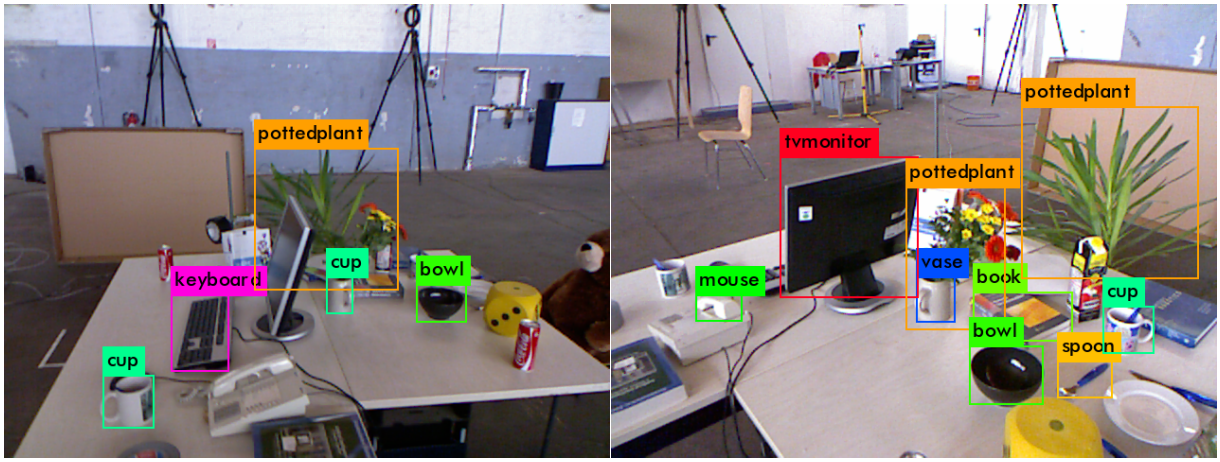


FIGURE 4.1 – Illustration sur le RGB-D TUM Dataset [SEE+12] de la situation dans laquelle un objet (la tasse contenant des fleurs au centre des images) reçoit deux étiquettes différentes de la part du détecteur d'objets YOLOv3 [RF18] : *tasse* (image de gauche) et *vase* (droite). Ces images illustrent également le fait qu'il existe différents objets possédant la même étiquette (*e.g.* les trois tasses, ou les deux plantes), et que certaines détections d'objets peuvent être erronées (cf. le téléphone confondu avec une souris d'ordinateur dans l'image de droite).

transférées des détections 2D aux ellipsoïdes 3D, ce qui conduit parfois à des situations où un seul objet est décrit par plusieurs étiquettes, et où différents objets sont décrits par la même étiquette (voir figure 4.1).

4.3.2 Procédure de type RANSAC pour l'estimation de la position

A partir d'un ensemble de détections d'objets et d'une estimation de l'orientation de la caméra (voir 6.1.2 pour plus de détails sur le calcul de l'orientation), notre méthode consiste à résoudre conjointement les problèmes d'association de données et d'estimation de la position de la caméra.

En pratique, toute association possible entre les objets détectés et les ellipsoïdes 3D est déterminée sur la base d'une compatibilité d'étiquettes. Comme la pose peut être calculée à partir d'une unique correspondance ellipse - ellipsoïde, une pose est calculée pour chaque paire possible. Un ensemble de consensus est construit au niveau des ellipsoïdes : pour chacune de ces poses potentielles, nous reprojetons tous les ellipsoïdes 3D dans l'image, et considérons les paires ellipse - ellipsoïde comme des inliers lorsque leurs étiquettes sont compatibles et que l'*intersection sur l'union* (IoU) entre les boîtes détectées et reprojetées est supérieur à un certain seuil (0,5 dans nos expériences). Lorsqu'un objet 3D est reprojeté sur plusieurs détections 2D, seul celui qui a le plus grand IoU est pris en compte. Lorsque deux configurations conduisent au même nombre d'inliers, celle qui a la plus grande somme de scores IoU est sélectionnée.

4.3.3 Estimation de l'orientation

Dans notre méthode, nous supposons que l'orientation de la caméra est connue. Le but de cette partie est de décrire brièvement comment cette matrice peut être obtenue en pratique, et comment notre méthode peut fournir un mécanisme permettant de lever l'ambiguïté inhérente à certaines méthodes de détermination de l'orientation. Les unités de mesure inertielle (IMU) sont des dispositifs électroniques dont l'orientation est mesurée à l'aide d'une combinaison d'accéléro-

FIGURE 4.2 – Les points de fuite de Manhattan sont des indices robustes pour calculer l'orientation de la caméra. Cette figure montre quelques exemples de résultats dans des scènes industrielles complexes. Ils ont été obtenus en utilisant la méthode décrite dans [SFB18]. Les points de fuite sont représentés par les segments de droites (une couleur par point de fuite) et les directions du repère de Manhattan sont montrées en rouge, vert, bleu.

mètres et de gyroscopes, parfois aussi de magnétomètres. Des IMU sont contenues dans presque tous les smartphones et les tablettes, et peuvent facilement être utilisées pour fournir la matrice d'orientation de la caméra requise par notre méthode. Par ailleurs, cette matrice peut aussi être obtenue directement à partir des images grâce à des méthodes basées sur la détection de points de fuite (PF) [SFB16, SFB18, ZWJ16]. En effet, la détection de PF présente maintenant des performances relativement bonnes dans différents types d'environnements, comme les environnements intérieurs ou industriels. Pour les expériences présentées dans la partie 4.4.2, nous utilisons la méthode décrite dans [SFB18], puisque son intérêt a été démontré dans différents types d'environnements industriels.

En utilisant les PF, l'orientation de la caméra peut être obtenue par rapport à un repère aligné sur trois directions orthogonales de la scène qui correspondent à trois PF spécifiques (qualifiés de Manhattan). Les images d'environnements créés par l'homme (y compris industriels) contiennent souvent un tel triplet de PF orthogonaux (voir par exemple la figure 4.2). Cette idée de calculer l'orientation de la caméra à partir des points de fuite de Manhattan a été proposée il y de nombreuses années [KZ02], mais nous abordons ici certaines questions qui ne sont souvent pas prises en compte dans la littérature alors qu'elles sont rencontrées dans la pratique : comment fixer la rotation entre le repère de Manhattan et le repère dans lequel les ellipsoïdes sont exprimés,

et comment traiter les problèmes d'échanges d'axes ou de symétries entre ces deux repères.

Calcul de la rotation entre le repère des ellipsoïdes et celui de Manhattan

Les ellipsoïdes sont exprimés dans le repère monde, dont l'orientation ne correspond pas nécessairement aux directions de Manhattan de la scène. Pour cette raison, nous devons calculer la rotation ${}^{m}R_{w}$ entre le repère monde (noté w) et le repère de Manhattan (noté m), et utiliser ${}^{cam}R_{w} = {}^{cam}R_{m}\,{}^{m}R_{w}$, où ${}^{cam}R_{m}$ est la matrice de rotation entre le repère de Manhattan et celui de la caméra (obtenue en utilisant [SFB18]). Pour faire cela, nous détectons les PF de Manhattan dans N images pour lesquelles les rotations monde-caméra ${}^{cam}R_{w}{}^{(i)}$ sont connues. Cela fournit N matrices ${}^{m}R_{w}{}^{(i)} = {}^{cam}R_{m}{}^{(i)\top}\,{}^{cam}R_{w}{}^{(i)}$, qui doivent toutes être les mêmes si les données ne sont pas bruitées. Pour une plus grande précision, nous calculons la moyenne de ces matrices de rotations

$$\overline{{}^{m}R_{w}} = \sum_{i=1}^{N} \frac{{}^{m}R_{w}{}^{(i)}}{N}$$

Comme cette moyenne n'est pas nécessairement elle-même une matrice de rotation, nous utilisons sa projection orthogonale ${}^{m}R_{w}$ sur le groupe spécial orthogonal $SO_3(\mathbb{R})$, donnée par :

$${}^{m}R_{w} = \overline{{}^{m}R_{w}}U\,\mathrm{diag}\left(\frac{1}{\sqrt{\Lambda_1}}, \frac{1}{\sqrt{\Lambda_2}}, \frac{s}{\sqrt{\Lambda_3}}\right)U^{\top},$$

où $\Lambda_1 \geq \Lambda_2 \geq \Lambda_3 \geq 0$ sont les valeurs propres de $M = \overline{{}^{m}R_{w}}^{\top}\overline{{}^{m}R_{w}}$, $U^{\top}\mathrm{diag}(\Lambda_1, \Lambda_2, \Lambda_3)U$ est la décomposition en valeurs singulières (SVD) de M et s est le déterminant de $\overline{{}^{m}R_{w}}$[Moa02].

Gestion des problèmes d'échanges d'axes et de symétries

Selon la position de la caméra par rapport à la scène, les axes X et Y du repère de Manhattan calculé peuvent être échangés ou réfléchis par rapport à ceux attachés à la scène (et utilisés pour calculer la matrice ${}^{m}R_{w}$). Pour traiter ce cas, nous considérons, dans la procédure de type RANSAC décrite dans la partie 4.3.2, chacun des quatre cas possibles pour la matrice de rotation Manhattan-caméra ${}^{cam}R_{m}$: $({}^{cam}R_{m:,1}\ {}^{cam}R_{m:,2}\ {}^{cam}R_{m:,3})$, $(-{}^{cam}R_{m:,1}\ -{}^{cam}R_{m:,2}\ {}^{cam}R_{m:,3})$, $({}^{cam}R_{m:,2}\ -{}^{cam}R_{m:,1}\ {}^{cam}R_{m:,3})$, $(-{}^{cam}R_{m:,2}\ {}^{cam}R_{m:,1}\ {}^{cam}R_{m:,3})$, et conservons celle qui maximise l'ensemble de consensus tel qu'expliqué dans la partie 4.3.2.

4.4 Résultats

4.4.1 Précision de la pose avec un objet

La base d'images LINEMOD

Nous évaluons tout d'abord notre méthode d'estimation de la position de la caméra à partir de la détection d'un objet dans l'image sur le jeu de données standard LINEMOD [HLI+12]. Cet ensemble d'images est conçu pour étalonner les algorithmes d'estimation de la pose 6D d'un objet, et plusieurs mesures de précision sont couramment utilisées : erreur de reprojection, score IoU, mesure ADD... (voir par exemple [TSF18] pour plus de détails). Cependant, notre méthode sans entrainement, basée sur la modélisation ellipsoïdale des objets 3D et la modélisation elliptique de leurs projections 2D, est conçue pour une relocalisation grossière de la caméra au lieu d'une estimation précise de la pose.

Méthode	Tekin *et al.* [TSF18]	Notre méthode					
Perturbation	-	0°	1°	2°	0°	1°	2°
Seuil	5 pixels	5 pixels			20 pixels		
ape	92.10	**94.69**	**94.77**	**94.69**	100	100	100
benchvise	**95.06**	12.07	12.16	11.56	93.1	93.1	93.0
bowl	-	**79.31**	**79.05**	**78.46**	100	100	100
cam	**93.24**	49.61	49.44	49.35	100	100	100
can	**97.44**	58.99	58.38	57.77	100	100	100
cat	**97.41**	81.84	81.93	81.58	100	100	100
cup	-	**61.60**	**61.68**	**61.09**	100	100	100
driller	**79.41**	8.79	8.61	8.08	98.4	98.2	98.5
duck	94.65	**94.85**	**94.93**	**94.68**	100	100	100
eggbox	90.33	**97.26**	**97.18**	**96.59**	100	100	99.9
glue	**96.53**	18.96	18.79	18.19	100	100	100
holepuncher	**92.86**	91.41	91.33	91.25	100	100	100
iron	**82.94**	37.11	36.66	35.30	100	100	100
lamp	**76.87**	13.67	12.73	12.39	100	100	99.9
phone	**86.07**	17.85	17.69	17.27	100	99.8	99.9
Moyenne	**90.37**	54.53	54.36	53.88	99.4	99.4	99.4

TABLE 4.1 – Comparaison de notre méthode de relocalisation de caméra avec la méthode d'estimation de pose d'objets la plus précise actuellement [TSF18] sur la base d'images LINEMOD. Nous reportons les pourcentages de poses correctement estimées. Pour notre méthode, nous reportons les résultats en fonction de trois niveaux de perturbations différents appliqués au orientations de caméras, en ° par angle d'Euler. Les chiffres **en gras** indiquent la meilleure méthode au regard de la métrique avec seuil à 5 pixels. Même si notre méthode ne requiert pas d'entrainement spécifique à un objet, elle donne des résultats relativement précis, et surpasse même la référence sur trois objets. De plus, notre méthode apparaît robuste à la perturbation appliquée sur l'orientation de la caméra.

Détails techniques et résultats

La plupart des méthodes de détection d'objets donnent leurs résultats sous forme de boîtes englobantes rectangulaires alignées avec les axes de l'image (YOLO [RDGF16], Faster R-CNN [RHGS15], SSD [LAE+16]). Pour simuler ce comportement, nous commençons par projeter le modèle CAO de l'objet dans l'image en utilisant la vérité terrain de la matrice de projection, et calculons ensuite la boîte englobante des points 2D obtenus. L'ellipse inscrite dans la boîte est finalement utilisée pour modéliser l'objet projeté, comme suggéré dans [CRD16, RCD18].

Nous choisissons au hasard 50 images par objet (la base de données contient 15 objets, avec approximativement 1200 images pour chaque) pour construire leur modèle ellipsoïdal en utilisant [RCD18]. Toutes les autres images sont utilisées pour les tests. Durant ces derniers, nous ajoutons un bruit uniforme inférieur à un certain seuil (0° (aucune perturbation), 1°, puis 2°) à chacun des 3 angles d'Euler (AE) représentant l'orientation correcte de la caméra, dans le but de simuler les mesures obtenues par utilisation de capteurs. L'erreur totale sur l'orientation de la caméra peut atteindre 2° dans le premier cas (1°/AE), et 4.5° dans le second (2°/AE).

La première mesure utilisée pour évaluer notre méthode est l'erreur de reprojection des points du modèle. Habituellement, les poses estimées sont considérées comme correctes lorsque l'erreur de reprojection moyenne est inférieure à un seuil donné en pixels (généralement 5). Le tableau 4.1 présente nos résultats sur les 15 objets LINEMOD en comparaison avec la méthode d'estimation de pose d'objets qui présente les meilleures performances actuellement [TSF18]. Il est important de noter que les objectifs des deux méthodes ne sont pas identiques. En effet, l'objectif de la référence est d'estimer avec précision la pose de la caméra en se basant sur un entrainement

spécifique à un objet, alors que notre méthode générique vise à effectuer une relocalisation grossière de la caméra à partir du ou des objets présents dans la scène, et s'appuie donc sur une modélisation parfois grossière des objets sous forme d'ellipsoïdes. Malgré cela, notre méthode est assez précise (la quasi-totalité des images présente une erreur de reprojection moyenne inférieure à 20 pixels), et est même plus précise que la référence sur 23% des objets (*eggbox*, *duck*, et *ape*). De plus, les résultats montrent que notre méthode est robuste à la perturbation appliquée sur les orientations de la caméra, puisque les performances ne présentent pas de diminution significative lorsque le niveau de bruit augmente. Des résultats plus détaillés de notre méthode sont fournis dans la figure 4.3 (colonne de gauche), pour un niveau de bruit sur l'orientation égal à 1°.

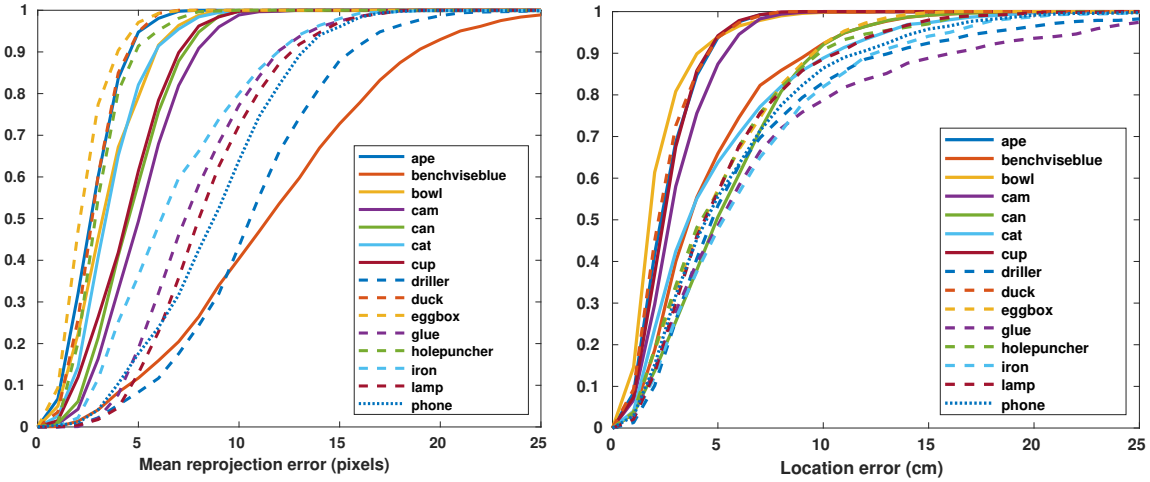


FIGURE 4.3 – Fonctions de répartition des erreurs de reprojection moyennes (en pixels, à gauche) et des erreurs en position des caméras obtenues par notre méthode (en cm, à droite) sur la base d'images LINEMOD.

La seconde métrique utilisée pour évaluer notre méthode est l'erreur de pose en 3D. En partant d'une orientation bruitée obtenue en ajoutant un bruit d'amplitude 1° sur chaque angle d'Euler représentant l'orientation correcte de la caméra, l'erreur totale sur l'orientation de la caméra ne dépasse pas 2°. Les précisions en termes de positions sont présentées dans la figure 4.3 (colonne de droite). Le diamètre maximal des objets varie de 10 cm à 30 cm, et la distance moyenne entre les caméras et les objets est d'environ 92 cm. Si l'on considère les 5 premiers objets (*bowl*, *duck*, *ape*, *cam*, *cup*), la distance entre les positions estimées des caméras et la vérité terrain est toujours inférieure à 9 cm. Dans le pire des cas (*glue*), cette distance ne dépasse pas 20 cm pour 90% des caméras. Même si notre méthode peut calculer la position de la caméra en se basant sur la détection d'un seul objet, elle est conçue pour tirer parti d'autres objets présents dans l'image. Ce pire niveau de précision ne serait donc atteint que dans des configurations très difficiles.

Interprétation des résultats

Afin de fournir une analyse plus approfondie des résultats précédents, nous étudions l'effet de la modélisation ellipsoïdale de divers objets sur les performances finales de relocalisation de la caméra. En effet, les résultats présentés dans la partie précédente montrent des différences notables selon l'objet considéré. Notre méthode repose sur la détection d'une ellipse virtuelle considérée comme une projection du modèle d'objet 3D (ellipsoïde). Par conséquent, elle dépend

fortement de notre capacité à détecter des ellipses similaires à la projection du modèle au moyen de la vérité terrain de la matrice de projection. Pour quantifier l'écart entre les détections réelles et attendues, nous définissons une erreur de détection comme la distance moyenne entre les 4 sommets (extrémités des axes principaux) de l'ellipse détectée et leurs points les plus proches sur l'ellipse projetée avec la vérité terrain de la matrice de projection de la caméra. La figure 4.4 montre la corrélation entre l'erreur de reprojection moyenne sur l'ensemble du jeu de données LINEMOD (illustrant nos performances de relocalisation) et l'erreur de détection initiale. La figure 4.5 illustre plus concrètement ce phénomène sur les objets fournissant les meilleurs (*eggbox*) et pires (*driller*) résultats.

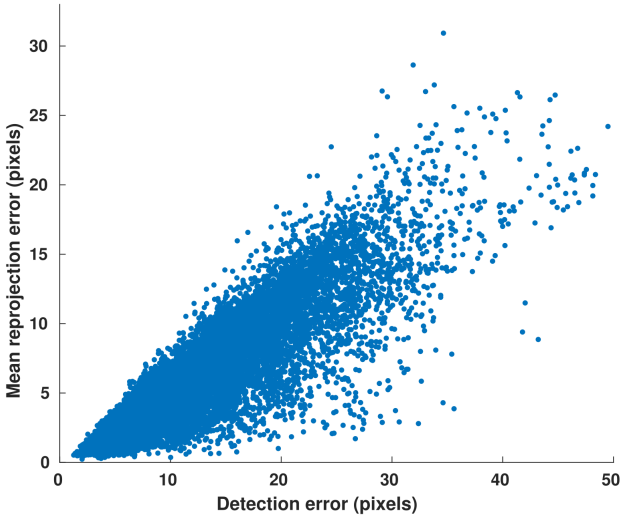


FIGURE 4.4 – Erreurs de reprojection moyennes (en pixels) pour notre méthode sur la base de données LINEMOD complète, en fonction de l'erreur de détection sur les ellipses (en pixels). Cette figure montre la corrélation entre nos performances de relocalisation et l'erreur initiale sur la détection de l'ellipse.

4.4.2 Scénarios réels

La base d'images RGB-D TUM

Nous évaluons maintenant la robustesse de la méthode de relocalisation de caméra sur le jeu de données standard RGB-D TUM [SEE+12]. Même si ce jeu de données a été créé à l'origine pour étalonner les algorithmes de SLAM, ses séquences contenant des objets communs répétés et des occlusions le rendent pertinent pour l'évaluation des performances de notre algorithme.

Nous utilisons deux séquences pour les tests : *fr2/desk* et *fr3/long_office*. Elles représentent toutes deux des environnements de bureau avec des objets répétés tels que des ordinateurs, des livres, des tasses ou des bouteilles. Elles sont composées d'environ 2700 images prises par une personne debout effectuant une boucle autour d'un bureau central. Les scènes et les trajectoires des opérateurs sont approximativement contenues dans un carré de 4 mètres de côté pour *fr2/desk*, et de 5 mètres de côté pour *fr3/long_office*.

La première scène (*fr2/desk*) est composée de 16 objets regroupés sous 11 étiquettes (jusqu'à 3 objets avec la même étiquette). Au total, 104 images ont été utilisées pour construire le modèle, et les 2861 restantes pour les tests. La deuxième scène (*fr3/long_office*) est composée de 28 objets

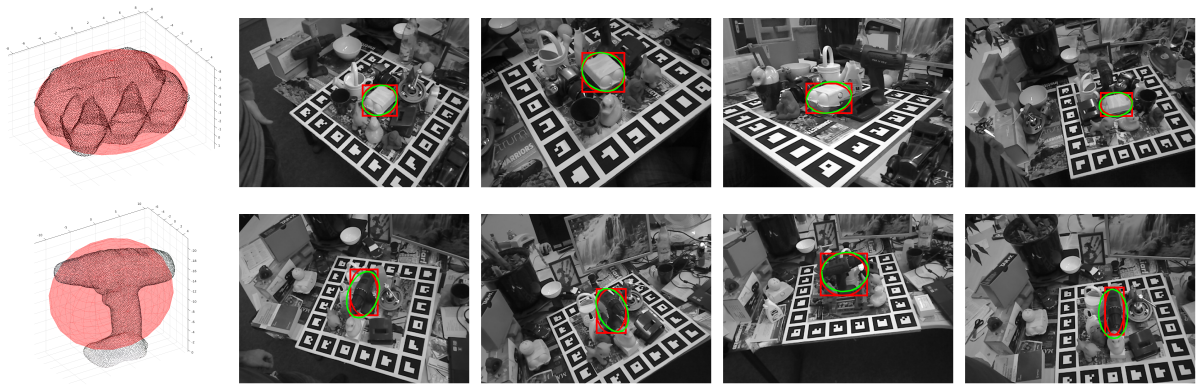


FIGURE 4.5 – Illustration des erreurs sur les ellipses inscrites dans les boîtes de détection (rouge), par rapport aux ellipses projetées avec la vérité terrain de la caméra (vert). Cette erreur est bien plus importante sur l'objet LINEMOD *driller* (ligne 2), qui ne peut pas être modélisé précisément par un ellipsoïde, que sur l'objet *eggbox* (ligne 1), qui est relativement bien approximé par son ellipsoïde reconstruit. La première colonne montre les nuages de points 3D fournies dans la base de données (en noir) et les ellipsoïdes reconstruits correspondants (en rouge).

et 9 étiquettes différentes (jusqu'à 10 objets avec la même étiquette). Parmi les 2559 images de la séquence, 71 ont été utilisées pour reconstruire les ellipsoïdes.

Dans nos expériences, seules les images RVB sont utilisées (aucune information de profondeur). La vérité terrain des poses a été obtenue par l'utilisation de capteurs associés à la caméra, mais certaines caméras sont fournies sans vérité terrain (705 dans *fr2/desk* et 2 dans *fr3/long_office*). Les paramètres intrinsèques des caméras sont connus.

Estimation de l'orientation

Notre méthode est évaluée en utilisant des orientations simulant des mesures par IMU ainsi que des orientations calculées à partir de la détection des points de fuite.

Les simulations d'IMU sont obtenues en ajoutant un bruit uniforme contenu entre $-1°$ et $+1°$ à chaque angle d'Euler représentant la vérité terrain de l'orientation de la caméra, ce qui conduit à une erreur totale d'au plus $2°$.

Les points de fuite de Manhattan ont été obtenus en utilisant la procédure décrite dans la partie 6.1.2. Afin de calculer la rotation entre le repère monde et le repère de Manhattan ${}^{m}R_{w}$, nous avons considéré les mêmes images que celles utilisées pour reconstruire les ellipsoïdes, et avons conservé celles qui nous ont permis de détecter le repère de Manhattan (voir partie 6.1.2). Au final, deux images étaient utilisables sur *fr2/desk* et quatre sur *fr3/long_office*. Nous avons obtenu, respectivement :

$$ {}^{m}R_{w} = \left(\begin{array}{c|c|c} 0.9994 & \text{-}0.0141 & 0.0212 \\ 0.0141 & 0.9996 & \text{-}0.0202 \\ \text{-}0.0202 & 0.0208 & 0.9993 \end{array}\right), {}^{m}R_{w} = \left(\begin{array}{c|c|c} 1.0000 & \text{-}0.0033 & 0.0062 \\ 0.0034 & 0.9999 & \text{-}0.0126 \\ \text{-}0.0060 & 0.0125 & 0.9998 \end{array}\right). $$

Ces matrices sont très proches de la matrice identité, ce qui signifie que le repère monde était déjà aligné avec le repère de Manhattan (défini par les bords des bureaux) lorsque nous avons utilisé la vérité terrain des poses pour reconstruire les ellipsoïdes.

fr2/desk et *fr3/long_office* sont en fait des séquences difficiles en termes de détection des PF. La dernière ligne de la figure 4.8 montre la vérité terrain des trajectoires suivies par la caméra

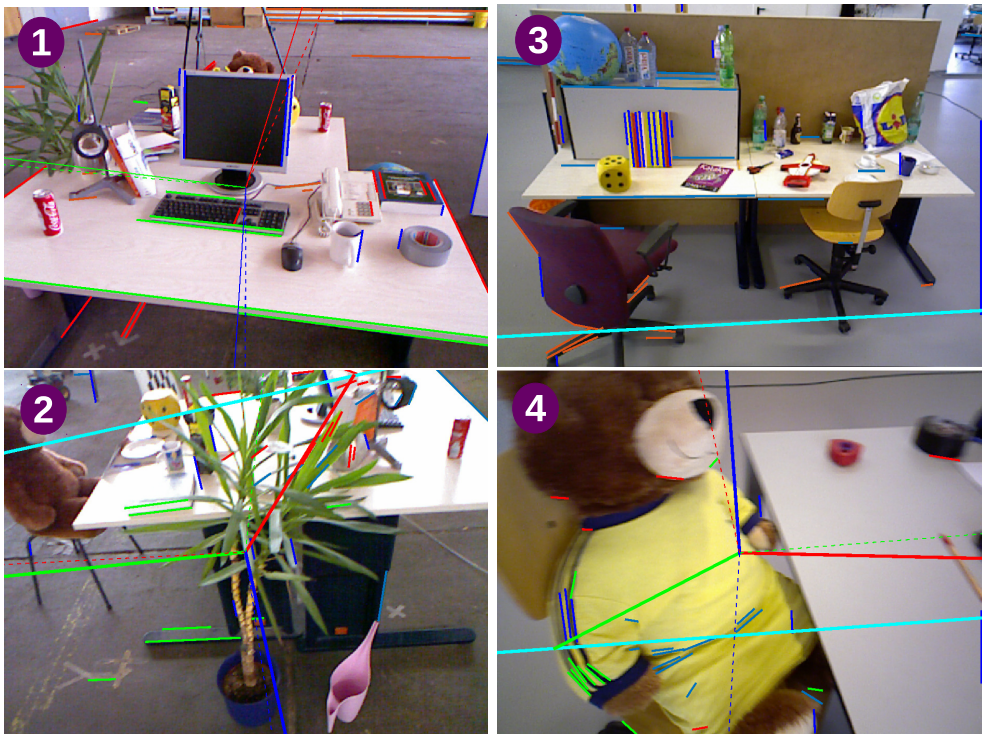


FIGURE 4.6 – Cas d'échec de la détection des points de fuite de Manhattan. Le repère de Manhattan calculé est illustré par les lignes continues en rouge, vert, bleu, la vérité terrain en lignes pointillées.

(projection des positions de la caméra sur le plan du sol, tracée en noir) dans ces deux séquences. Les parties des trajectoires où les PF de Manhattan n'ont pas pu être obtenus sont indiquées par des courbes violettes, et des cas typiques de défaillances sont présentés dans la figure 4.6, en utilisant les mêmes numéros que pour les parties des trajectoires. Dans la partie 1, les défaillances ou les inexactitudes sont souvent dues au fait que l'écran, qui est légèrement incliné, occupe une grande partie de l'image, ce qui induit en erreur la détection du zénith. Dans la partie 2, une plante occulte une grande partie de la scène, ce qui perturbe la détection des PF horizontaux. Dans la partie 3, le livre vertical génère plusieurs segments de droites qui correspondent à des droites parallèles quasi verticales dans la scène, ce qui, là encore, entraine une erreur sur la détection du zénith. De plus, les pieds de chaises sont en forme d'étoile, ce qui génère plusieurs segments de droites qui se rejoignent au centre des pieds, et produit de faux points de fuite. Dans la partie 4, l'ours en peluche occupe une grande partie de l'image. En dehors de ces cas difficiles, on retrouve régulièrement le repère de Manhattan tout au long des trajectoires.

Détection et mise en correspondance

Dans nos expériences, le détecteur d'objets utilisé est YOLOv3 [RF18]. Globalement, la procédure décrite dans la section 4.3.2 comporte suffisamment d'inliers pour calculer de manière robuste la pose, et un seul inlier peut même suffire. Certaines parties des séquences présentent toutefois des difficultés particulières que nous décrivons maintenant. Ces parties sont indiquées par des courbes orange dans la figure 4.8 (première ligne) et illustrées par des images typiques dans la figure 4.7, en utilisant les mêmes indices dans les deux figures.

Dans *fr2/desk*, certaines parties de la séquence présentent très peu (ou même aucun) objet

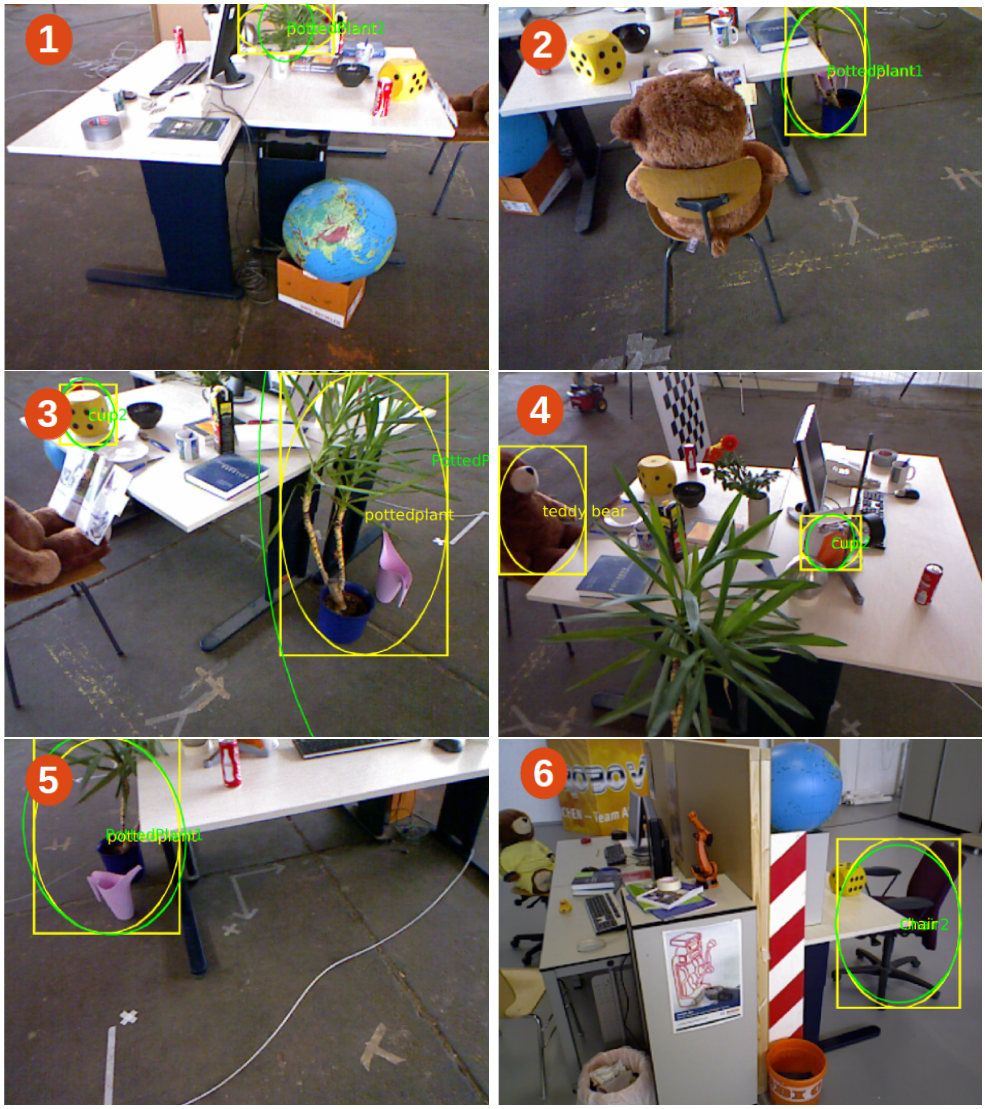


FIGURE 4.7 – Cas d'échec de notre méthode. (1, 2, 5) : un seul objet tronqué est détecté, aboutissant à des tailles de projections sous-évaluées. (3,4) : seulement deux détections dont une est fausse (*cup*), perturbant notre algorithme. (6) : une seule détection avec une étiquette ambigue (*chair*). Dans ce dernier cas, notre algorithme associe la détection à la chaise $n°2$ du modèle plutôt qu'à la chaise $n°1$.

entièrement visibles. En pratique, les images qui ne contiennent pas de détections, ou dont les étiquettes détectées ne sont pas présentes dans le modèle, sont ignorées. Dans d'autres cas, les très rares détections correctes peuvent souffrir d'occultations importantes et/ou être tronquées (voir indices 1, 2 et 5), et/ou être couplées à de fausses détections (voir indices 3 et 4), entraînant un échec de relocalisation de la caméra.

Dans *fr3/long_ office*, la partie la plus problématique est dénotée (6). Il s'agit d'images dans lesquelles une seule chaise est détectée, alors que trois chaises différentes sont présentes dans le modèle. Considérant une détection unique avec une ambiguïté d'étiquette, et sans aucune information géométrique précise sur les objets reconstruits (seulement des ellipsoïdes grossiers), notre méthode échoue parfois à associer la détection à l'objet auquel elle correspond.

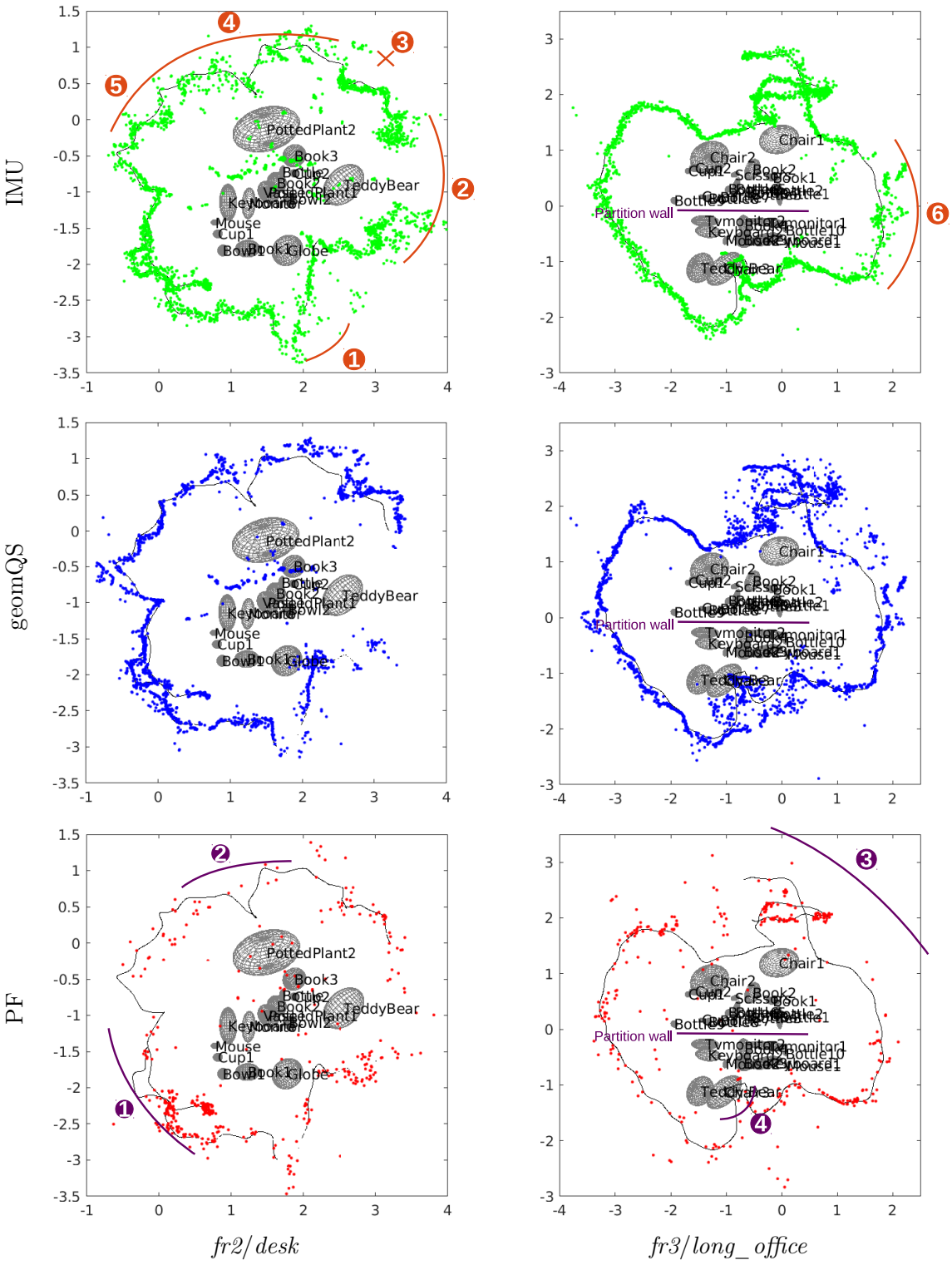


FIGURE 4.8 – Résultats de relocalisation de caméras avec notre méthode et les orientations IMU (ligne 1), l'optimisation à 6 degrés de liberté (ligne 2) et notre méthode avec les orientations obtenues par les PF (ligne 3). La vérité terrain des positions de caméras (projetées sur le plan du sol) est représentée par la ligne noire, les positions estimées par les points colorés. Première colonne : *fr2/desk* ; deuxième colonne : *fr3/long_office*.

Séquence		*fr2/ desk*			*fr3/ long_ office*		
Méthode		IMU	geomQS	PF	IMU	geomQS	PF
Position	moyenne	0.26	0.31	0.39	0.11	0.25	0.93
	écart-type	0.46	0.46	0.79	0.22	0.32	1.56
	médiane	0.08	0.12	0.10	0.06	0.13	0.11
Orientation	moyenne	*0.94*	1.41	15.6	*0.94*	4.24	39.0
	écart-type	*0.32*	3.71	43.5	*0.31*	8.28	71.6
	médiane	*0.93*	0.04	2.47	*0.93*	0.04	1.74

TABLE 4.2 – Erreurs sur la base de données RGB-D TUM en terme de position (en m) et d'orientation (en °) de caméras. Les chiffres en *italique* désignent des données synthétiques (obtenues par ajout de bruit à la vérité terrain).

Résultats quantitatifs

La relocalisation est évaluée sur *fr2/ desk* et *fr3/ long_ office* en utilisant les orientations simulant l'IMU et celles issues des PF. À titre de comparaison, nous avons également calculé les six paramètres de la pose en minimisant itérativement l'erreur de reprojection géométrique introduite dans [NMS19] et définie dans l'équation (4.6) :

$$X^*, Q^* = \text{argmin}_{X,Q} \sum_i ||f(\mathbf{x}_i, \mathbf{u}_i) \ominus \mathbf{x}_{i+1}||^2_{\sum_i} + \sum_{ij} ||\mathbf{b}_{ij} - \beta_{(\mathbf{x}_i, \mathbf{q}_j)}||^2_{\Lambda_{ij}}. \tag{4.6}$$

Dans la méthode de SLAM citée, la configuration maximale a posteriori des poses de caméra X^* et des quadriques duales Q^* est obtenue en résolvant le problème aux moindres carrés (4.6). Le premier terme reflète l'attachement aux données d'odométrie et le second terme est l'erreur de reprojection géométrique entre les ellipsoïdes projetés et les boîtes de détection. Plus précisément, l'erreur géométrique est définie comme la somme des distances de Mahalanobis au carré, avec une covariance Λ_{ij}, entre les sommets des boîtes englobantes des ellipsoïdes reprojetés ($\beta_{(\mathbf{x}_i, \mathbf{q}_j)}$) et les sommets des boîtes détectées ($\mathbf{b}_{ij}$). Dans nos tests, nous ne considérons pas l'attache aux données d'odométrie, et nous remplaçons la distance de Mahalanobis par la distance euclidienne, puisque nous traitons chaque image indépendamment des autres. De plus, les ellipsoïdes $\mathbf{q}_j$ sont fixés (modèle). Nous avons utilisé l'algorithme Levenberg-Marquardt pour effectuer l'optimisation et la vérité terrain pour initialiser les paramètres de pose. Par souci d'équité, nous avons utilisé les correspondances inliers déterminées par notre méthode.

Le tableau 4.2 indique les erreurs en position et en orientation moyennes et médianes obtenues par les trois méthodes sur les deux séquences. Toutes les positions calculées sont montrées après projection sur le plan du sol dans la figure 4.8. Dans ces figures, la méthode avec des orientations simulées est appelée *IMU*, la procédure basée sur les PF est nommée *PF* et la procédure minimisant l'erreur géométrique est appelée *geomQS*.

Orientations simulant un IMU Les poses de caméra obtenues avec les orientations simulées sont relativement proches de la trajectoire de la vérité terrain dans les deux séquences (cf. figure 4.8 (première ligne) et vidéo [1]), bien que le nuage des positions de caméras soit légèrement plus chaotique dans *fr2/ desk*. Cela est dû au fait que cette séquence contient des images plus difficiles en termes de détections, comme expliqué dans la partie précédente. Dans *fr3/ long_ office*, seule une partie de la séquence est particulièrement difficile à gérer, du fait qu'une seule chaise est

1. Vidéo : https://members.loria.fr/VGaudilliere/files/ISMAR19.mp4

détectée dans les images alors qu'il en existe plusieurs dans le modèle. Malgré ces difficultés, notre méthode présente d'excellentes performances de relocalisation : l'erreur médiane sur la position de la caméra est de 6 cm sur *fr3/long_office* et 8 cm sur *fr2/desk*.

Les figures 4.9 et 4.10 montrent qualitativement la robustesse de la méthode sur des configurations favorables (beaucoup d'objets détectés : fig. 4.9 - ligne 1, fig. 4.10 - ligne 1) et défavorables : motifs répétés (fig. 4.10 - lignes 1,4,5), détection d'objet unique (fig. 4.10 - ligne 2), occultations (fig. 4.9 - lignes 1,2,3), bruit de mouvement (fig. 4.10 - ligne 3). Ces figures illustrent également la robustesse de la méthode par rapport à différentes distances et orientations relatives entre les caméras et la scène.

Orientations obtenues à partir des PF Lorsque l'on utilise les PF de Manhattan pour calculer l'orientation de la caméra, on peut constater que le nuage des positions de caméras est beaucoup plus clairsemé que lorsque l'on utilise les orientations simulées (figure 4.8, dernière ligne). Bien entendu, cette approche ne peut réussir que si la détection d'objet et la détection des PF de Manhattan réussissent toutes deux, et hérite donc de tous les types de problèmes qui peuvent survenir pour chaque procédure. C'est pourquoi les erreurs moyennes indiquées dans le tableau 4.2 sont élevées, et en particulier pour *fr3/long_office*. Toutefois, les erreurs médianes (10 cm / 2,47° pour *fr2/desk*, 11 cm / 1,74° pour *fr3/long_office*) sont acceptables pour une tâche de relocalisation.

Optimisation à six degrés de liberté de l'erreur géométrique Les positions de caméras obtenues par optimisation, en fonction des six paramètres de pose, de l'erreur de reprojection géométrique sont relativement instables (figure 4.8, ligne du milieu) : la moitié du temps, la pose calculée est très proche de la vérité terrain, mais l'autre moitié du temps, elle en est éloignée. Ceci peut également être déduit des erreurs moyennes en position et en orientation présentées dans le tableau 4.2, qui sont importantes, alors que les erreurs médianes sont très faibles. Nous analysons ce résultat comme suit. Premièrement, l'optimisation à six degrés de liberté peut compenser les imprécisions sur les ellipses (inscrites dans les boîtes détectées), alors que dans les autres méthodes l'orientation est fixée. Deuxièmement, l'erreur géométrique tend à favoriser les ellipses plus grandes, au détriment des petits objets comme l'illustre, par exemple, la figure 4.9 - ligne 1. Cela confirme que cette méthode nécessite une attache aux données d'odométrie (premier terme de l'équation (4.6)). Il faut également noter que, pour cette raison, nous préférons ne pas faire suivre notre solution analytique par une optimisation non linéaire de la pose comme cela se fait couramment en calcul de pose.

4.4.3 Discussion

Dans cette partie, nous examinons les avantages et les inconvénients de notre méthode de relocalisation basée sur les objets par rapport aux méthodes bas-niveau basées sur les indices locaux. Par nature, notre méthode se concentre sur les caractéristiques sémantiques pertinentes des images de la scène, ce qui nous amène à traiter des modèles 3D légers qui ne représentent que la sémantique de la scène et omettent les informations inutiles. En outre, elle opère au niveau des objets, qui sont intrinsèquement plus discriminants que les indices locaux dans les tâches de compréhension de la scène et de localisation, et sont également plus robustes aux changements de point de vue ou d'illumination [LSN+16].

À l'opposé, les méthodes bas-niveau ont l'avantage d'être indépendantes du contexte, en ce sens qu'elles peuvent être appliquées sur des images de n'importe quel environnement, dès lors que ces images sont suffisamment texturées. Au contraire, notre méthode est dépendante de

l'algorithme de détection d'objets utilisé, et ne peut donc traiter que les classes d'objets apprises par le détecteur. L'application de notre méthode sur un nouvel environnement requiert donc d'entraîner le détecteur sur les objets principalement rencontrés dans ce type d'environnement.

Enfin, les modèles basés sur les indices locaux et sur les objets ont chacun leurs propres limites. Dans le contexte d'applications intérieures ou industrielles, de grandes parties de la scène ne sont pas texturées. Avec les techniques de Structure from Motion, les indices locaux peuvent être regroupés dans de petites régions des images, ce qui entraîne une estimation de la pose instable. En outre, les indices locaux répétés sont le plus souvent simplement rejetés en raison de leur ambiguïté et n'apparaissent pas dans le modèle de scène final [SSS08, HCH10]. Les modèles basés sur les objets permettent de prendre en compte des zones plus larges et non texturées de la scène pour le calcul de la pose, ce qui permet une plus grande robustesse, bien qu'au détriment de la précision due à l'approximation des objets en tant qu'ellipsoïdes. Une méthode itérative d'amélioration de la pose peut ensuite être utilisée.

4.5 Conclusion

Dans ce chapitre, nous avons exploré les moyens de réaliser une relocalisation au niveau des objets. Nous profitons des progrès réalisés dans la détection d'objets qui nous permettent de générer des correspondances 2D - 3D entre les objets détectés dans les images et approximés par une ellipse, et les objets 3D représentés par des ellipsoïdes. En supposant qu'une estimation de l'orientation de la caméra est disponible, nous avons proposé une méthode analytique pour calculer la position de la caméra à partir d'une correspondance ellipse - ellipsoïde. Un algorithme de type RANSAC opérant au niveau des objets est proposé pour gérer les associations de données erronées, soit en raison d'une fausse détection, soit en raison de la présence d'objets répétés dans la scène. Les expériences menées ont prouvé l'efficacité de la méthode même lorsqu'un petit nombre d'objets est détecté. Comme le montrent les expériences, les informations d'orientation fournies par l'IMU ou la détection des points de fuite s'avèrent suffisantes pour la relocalisation. Des méthodes itératives précises basées modèle peuvent ensuite être utilisées à partir de cette première estimation pour affiner la pose.

Cette méthode présente de nombreux avantages. En considérant le calcul de pose au niveau des objets, nous évitons les problèmes courants dus aux motifs répétés rencontrés par les méthodes basées sur les indices locaux, en particulier dans les environnements industriels. De plus, la combinatoire des correspondances ellipse - ellipsoïde est relativement faible, ce qui ouvre la voie à une relocalisation efficace dans les grands environnements, où seuls les objets les plus importants sont intégrés dans le modèle.

Dans la partie suivante, nous reprenons le problème théorique à un ellipsoïde, et essayons d'en tirer une méthode pour calculer, de plus, l'orientation de la caméra.

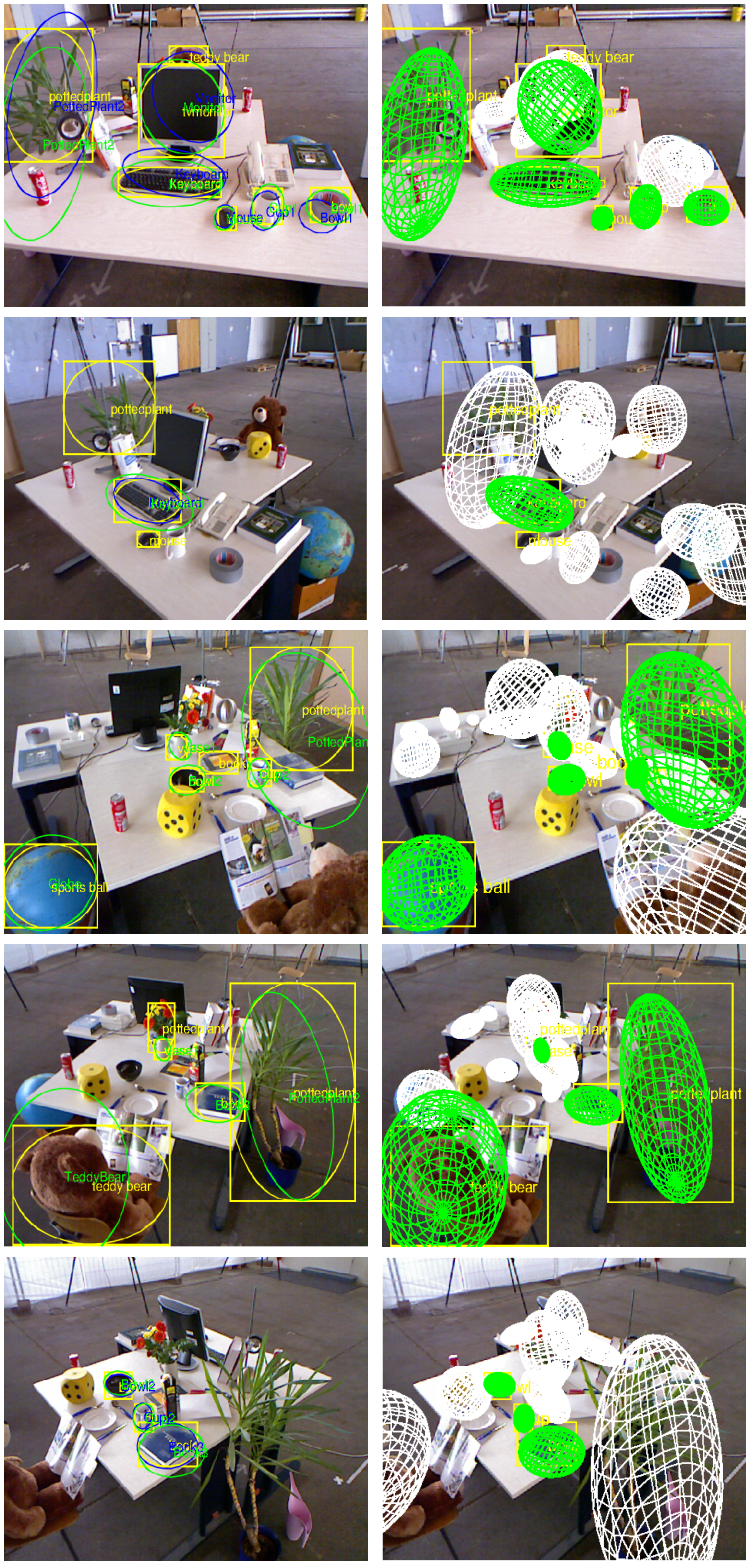


FIGURE 4.9 – Illustration de la robustesse de notre méthode sur différentes images de la base de données RGB-D TUM (séquence *fr2/desk*). Les boîtes de détection et les ellipses inscrites sont présentées en jaune, avec leurs étiquettes. Les reprojections des ellipsoïdes inliers sont présentées en vert, tandis que les autres sont en blanc. Les objets ne peuvent pas être classifiés en tant qu'inliers si leurs projections ne sont pas détectées dans l'image. Dans la première colonne, les ellipses reprojetées en utilisant la pose estimée par *geomQS* sont présentées en bleu.

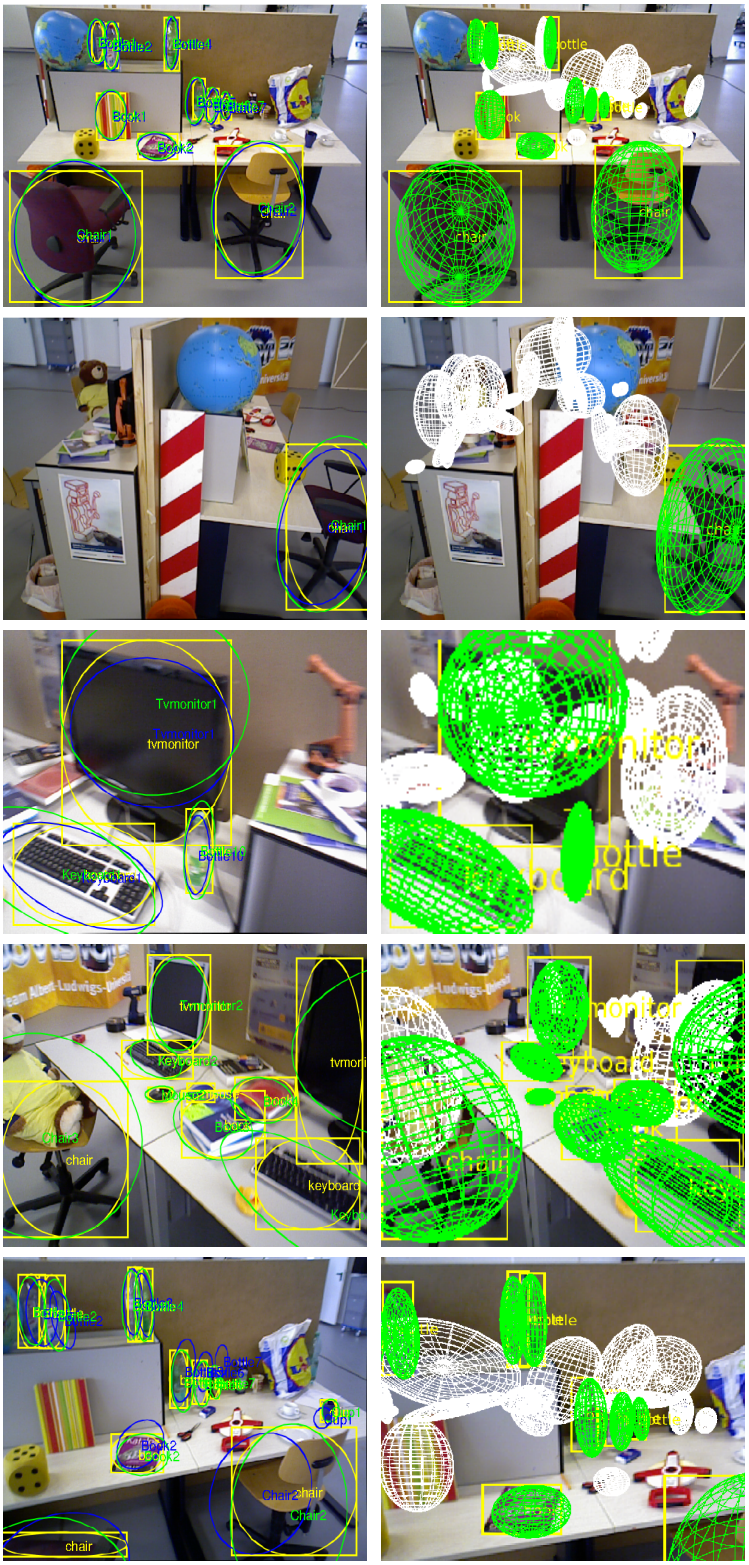


FIGURE 4.10 – Illustration de la robustesse de notre méthode sur différentes images de la base de données RGB-D TUM (séquence *fr3/long_office*). Les boîtes de détection et les ellipses inscrites sont présentées en jaune, avec leurs étiquettes. Les reprojections des ellipsoïdes inliers sont présentées en vert, tandis que les autres sont en blanc. Les objets ne peuvent pas être classifiés en tant qu'inliers si leurs projections ne sont pas détectées dans l'image. Dans la première colonne, les ellipses reprojetées en utilisant la pose estimée par *geomQS* sont présentées en bleu.

Chapitre 5

Détermination de l'ensemble des caméras satisfaisant une correspondance ellipse - ellipsoïde

Sommaire

5.1 Changement de variables scalaires 83
5.2 Cas de l'ellipsoïde non dégénéré 85
5.3 Cas du sphéroïde 89
5.3.1 Cas du cône elliptique non dégénéré 90
5.3.2 Cas du cône de révolution 97
5.4 Cas de la sphère 99
5.5 Etude de la sensibilité au bruit 103
5.6 Conclusion 105

Nous avons vu, au chapitre précédent, que le paradigme de modélisation ellipse - ellipsoïde induit un découplage entre orientation et position, qui permet de réduire le problème d'estimation de pose de caméra à l'estimation de son orientation seulement. Nous en avons tiré une méthode permettant de calculer la position de la caméra à partir de son orientation et d'au moins une détection d'objet dans l'image. Notre but est maintenant de développer une méthode pour calculer la pose complète de la caméra à partir d'un ensemble de correspondances entre ellipses et ellipsoïdes. Dans ce chapitre, nous revenons au problème théorique de détermination des caméras satisfaisant une unique correspondance, et déterminons l'ensemble de ses solutions. Plus précisément, nous développons des solutions différentes selon le type d'ellipsoïde considéré (parties 5.2, 5.3 et 5.4). Dans tous les cas, nous nous intéressons tout d'abord aux solutions scalaires σ de l'équation (2.6), puis nous en déduisons les expressions des autres inconnues. Faisant cela, nous montrons que le problème de détermination des poses de caméras satisfaisant une correspondance ellipse - ellipsoïde est un problème à un seul degré de liberté. Nous étudions ensuite la sensibilité de la méthode au bruit de détection sur les ellipses, dans le cas des ellipsoïdes non dégénérés (partie 5.5).

5.1 Changement de variables scalaires

Dans cette partie, nous introduisons une variable scalaire $m = \sqrt[3]{\mu}$ qui va nous servir à paramétrer le problème. Dans le résultat 3, nous montrons qu'il existe une relation bijective entre

m et chacune des deux valeurs propres généralisées du couple $\{A, B'\}$, ce qui nous permettra d'effectuer des changements de variables qui simplifieront les développements suivants. Nous mettrons alors en évidence le fait qu'il existe une infinité de valeurs possibles pour l'inconnue scalaire σ dans le cas d'un ellipsoïde non dégénéré (partie 5.2), chacune de ces valeurs fournissant un nombre fini de caméras solutions. A l'inverse, nous montrerons qu'il n'en existe qu'une dans le cas d'un sphéroïde ou d'une sphère (parties 5.3 et 5.4), mais qu'une infinité de caméras solutions y est attachée.

Résultat 3. *Supposons que le quadruplet* $(A, B', \mathbf{\Delta}, \sigma)$ *vérifie l'équation* (2.6). *En notant* $d = \sqrt[3]{\frac{det(A)}{det(B')}}$, *et* σ_i *la valeur propre généralisée de multiplicité* i *du couple* $\{A, B'\}$, *alors*

$$\sigma_1 = dm^2 \qquad et \qquad \sigma_2 = \frac{d}{m}.$$

Puisque $m < 0$, *on a également*

$$m = -\sqrt{\frac{\sigma_1}{d}} = \frac{d}{\sigma_2}.$$

Démonstration. Supposons que l'équation (2.6) soit vérifiée. D'après le résultat 2 (page 63), nous avons

$$\sigma_2 = \frac{\sigma_1}{\mu}$$

Puisque

$$\sigma_1 \sigma_2^2 = det(B'^{-1} A)$$

il vient

$$\frac{\sigma_1^3}{\mu^2} = \frac{det(A)}{det(B')}$$

ce qui est équivalent à

$$\sigma_1^3 = d^3 m^6$$

soit

$$\sigma_1 = dm^2$$

Puisque $\sigma_2 = \frac{\sigma_1}{\mu}$, nous avons également

$$\sigma_2 = \frac{\sigma_1}{m^3} = \frac{dm^2}{m^3} = \frac{d}{m}$$

□

Pour un même problème (taille de l'ellipsoïde et forme du cône fixées), il peut y avoir plusieurs quadruplets $(A, B', \mathbf{\Delta}, \sigma)$ qui vérifient l'équation (2.6). En particulier, le scalaire σ, et donc a fortiori m, peut prendre plusieurs valeurs. En revanche, le scalaire d est fixé, puisqu'il ne dépend que des valeurs propres des matrices A et B', qui sont elles-mêmes fixées.

Dans la suite, nous déterminons l'ensemble des solutions de l'équation dans le cas d'un ellipsoïde non dégénéré, puis dans le cas d'un ellipsoïde de révolution, et enfin dans le cas d'une sphère.

5.2 Cas de l'ellipsoïde non dégénéré

A priori, le scalaire m peut prendre toute valeur appartenant à l'intervalle $]-\infty;0[$ (car $\mu < 0$). Nous allons montrer que son domaine de définition est en fait plus restreint.

Lorsque l'ellipsoïde possède trois axes de longueurs différentes (ellipsoïde non dégénéré), le théorème 2 fournit une caractérisation des scalaires m qui correspondent à des solutions σ de l'équation (2.6).

Théorème 2. *Lorsque l'ellipsoïde est non dégénéré, le scalaire $\sigma = dm^2$ est solution de l'équation (2.6) si et seulement si les trois éléments du vecteur $M_A^{-1}V(m)$ sont simultanément positifs :*

$$M_A = \begin{pmatrix} 1 & 1 & 1 \\ \lambda_{A,1} & \lambda_{A,2} & \lambda_{A,3} \\ \lambda_{A,1}^2 & \lambda_{A,2}^2 & \lambda_{A,3}^2 \end{pmatrix}$$

$$V(m) = \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m \\ 1 - m^3 \\ tr(B')dm^2 - tr(A)m^3 \end{pmatrix}$$

Démonstration. $\boxed{\Longrightarrow}$ Supposons que l'équation (2.6) soit satisfaite :

$$A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A + \mu A = \sigma B' \tag{2.6}$$

Alors, l'équation équivalente (4.1) l'est aussi :

$$\boldsymbol{\Delta}\boldsymbol{\Delta}^\top = A^{-1} - \frac{\mu}{\sigma}B'^{-1} \tag{4.1}$$

Puisque la trace d'un produit de matrices ne dépend pas de l'ordre, on a

$$\begin{aligned} tr(A\boldsymbol{\Delta}\boldsymbol{\Delta}^\top A) &= tr((A\boldsymbol{\Delta})(\boldsymbol{\Delta}^\top A)) \\ &= tr((\boldsymbol{\Delta}^\top A)(A\boldsymbol{\Delta})) \\ &= tr(\boldsymbol{\Delta}^\top A^2 \boldsymbol{\Delta}) \\ &= \boldsymbol{\Delta}^\top A^2 \boldsymbol{\Delta} \qquad \text{(scalaire)} \end{aligned}$$

et, de la même façon,

$$tr(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top) = \boldsymbol{\Delta}^\top \boldsymbol{\Delta}$$

Comme, de plus, $\mu = 1 - \boldsymbol{\Delta}^\top A \boldsymbol{\Delta}$, appliquer $tr()$ aux équations (4.1) et (2.6) permet d'aboutir au système suivant :

$$\begin{cases} \boldsymbol{\Delta}^\top \boldsymbol{\Delta} = tr(A^{-1}) - \frac{\mu}{\sigma} tr(B'^{-1}) \\ \boldsymbol{\Delta}^\top A \boldsymbol{\Delta} = 1 - \mu \\ \boldsymbol{\Delta}^\top A^2 \boldsymbol{\Delta} = \sigma tr(B') - \mu tr(A) \end{cases}$$

Le système peut être réécrit en remplaçant μ et σ par leurs expressions en fonction de m :

$$\begin{cases} \boldsymbol{\Delta}^\top \boldsymbol{\Delta} = tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m \\ \boldsymbol{\Delta}^\top A \boldsymbol{\Delta} = 1 - m^3 \\ \boldsymbol{\Delta}^\top A^2 \boldsymbol{\Delta} = tr(B')dm^2 - tr(A)m^3 \end{cases} \tag{5.1}$$

Ces équations métriques sont indépendantes de la base considérée, donc en considérant une base dans laquelle A est diagonale, et en notant $(\mathbf{\Delta}_{ell,x}, \mathbf{\Delta}_{ell,y}, \mathbf{\Delta}_{ell,z})^\top$ l'expression de $\mathbf{\Delta}$ dans cette base, nous pouvons réécrire le système précédent sous la forme d'une équation matricielle :

$$\begin{pmatrix} 1 & 1 & 1 \\ \lambda_{A,1} & \lambda_{A,2} & \lambda_{A,3} \\ \lambda_{A,1}^2 & \lambda_{A,2}^2 & \lambda_{A,3}^2 \end{pmatrix} \begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 \\ \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d} m \\ 1 - m^3 \\ tr(B')dm^2 - tr(A)m^3 \end{pmatrix}$$

i. e.

$$M_A \begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 \\ \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = V(m)$$

Puisque les valeurs propres de A sont toutes différentes, la matrice de Vandermonde M_A est inversible, d'où il est possible d'écrire

$$\begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 \\ \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = M_A^{-1} V(m) \tag{5.2}$$

Les éléments du terme de gauche sont bien tous positifs, d'où le résultat.

$\boxed{\Longleftarrow}$ Supposons maintenant que les trois éléments du vecteur $M_A^{-1}V(m)$ soient positifs. Soit $\mathbf{\Delta}_{ell} = (\mathbf{\Delta}_{ell,x}, \mathbf{\Delta}_{ell,y}, \mathbf{\Delta}_{ell,z})^\top$ un vecteur tel que

$$\begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 \\ \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = M_A^{-1} V(m)$$

Une telle définition est possible car les trois éléments du vecteur sont positifs.

On peut alors démontrer, à l'aide d'un logiciel de calcul formel (le code Maple correspondant est fourni en Annexe B), que, quel que soient les choix de signes faits pour $\mathbf{\Delta}_{ell}$, la matrice

$$\frac{\sigma}{\mu}(A_{ell}^{-1} - \mathbf{\Delta}_{ell}\mathbf{\Delta}_{ell}^\top) = \frac{d}{m}(A_{ell}^{-1} - \mathbf{\Delta}_{ell}\mathbf{\Delta}_{ell}^\top)$$

possède les mêmes valeurs propres avec les mêmes multiplicités que B'^{-1}. Comme ces deux matrices sont diagonalisables car symétriques, cela veut dire qu'elles sont semblables, et donc que l'équation (4.1), équivalente à (2.6), est vérifiée.

□

Dans les trois résultats qui suivent, nous allons montrer qu'à partir de σ, il est possible de déterminer les expressions des vecteurs $\mathbf{\Delta}$ et des matrices B' compatibles avec la matrice A exprimée dans la base monde (résultat 4 page 86). Puis nous en déduirons les positions (résultat 5 page 88), et enfin les orientations (résultat 6 page 88), des caméras solutions.

Résultat 4. *Lorsque l'ellipsoïde est non dégénéré, il existe, pour chaque valeur de σ, huit cônes de rétroprojection $(\boldsymbol{E}_w, B'_w)$ tangents à l'ellipsoïde. Ces cônes sont symétriques par rapport aux trois plans principaux de l'ellipsoïde (voir figure 5.1).*

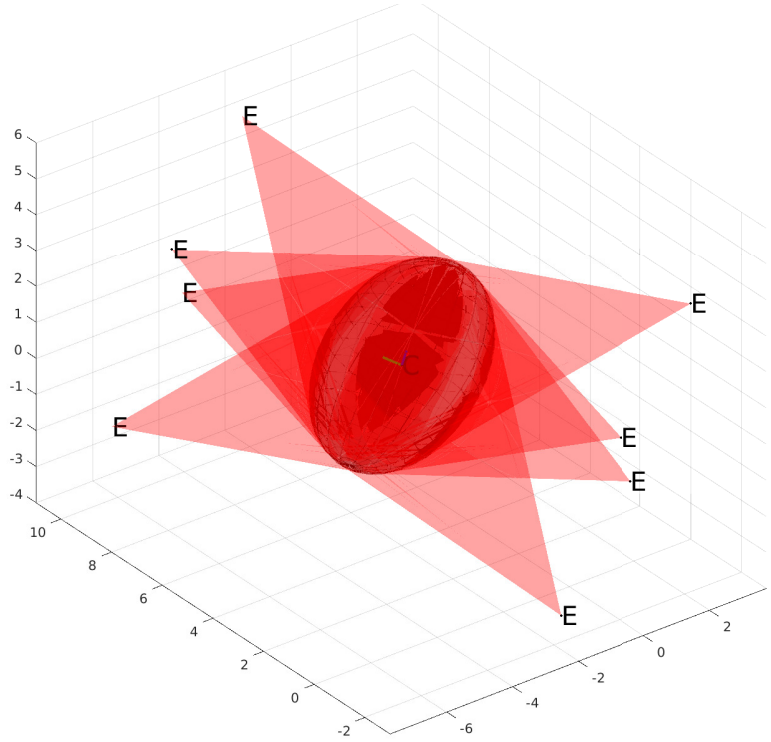


FIGURE 5.1 – Illustration des huit cônes de rétroprojection tangents à l'ellipsoïde, pour une valeur de σ fixée.

Démonstration. Dans le cas où l'ellipsoïde est non dégénéré, le théorème 2 fournit une caractérisation nécessaire et suffisante des scalaires σ pour qu'ils soient solutions de l'Equation d'Alignement des Cônes. De plus, sa démonstration met en évidence le fait que les vecteurs $\mathbf{\Delta}$ sont les vecteurs qui s'expriment dans la base propre de A sous la forme

$$\mathbf{\Delta}_{ell} = \begin{pmatrix} \pm\sqrt{\mathbf{\Delta}^2_{ell,x}} \\ \pm\sqrt{\mathbf{\Delta}^2_{ell,y}} \\ \pm\sqrt{\mathbf{\Delta}^2_{ell,z}} \end{pmatrix} \tag{5.3}$$

où

$$\begin{pmatrix} \mathbf{\Delta}^2_{ell,x} \\ \mathbf{\Delta}^2_{ell,y} \\ \mathbf{\Delta}^2_{ell,z} \end{pmatrix} = M_A^{-1} V(m)$$

Il y a donc huit vecteurs $\mathbf{\Delta}_{ell}$ solutions associés à un même σ, et ils sont symétriques par rapport aux trois plans principaux de l'ellipsoïde.

D'autre part, la matrice A s'exprime dans cette base

$$A_{ell} = \begin{pmatrix} \lambda_{A,1} & 0 & 0 \\ 0 & \lambda_{A,2} & 0 \\ 0 & 0 & \lambda_{A,3} \end{pmatrix}.$$

Puis l'équation (2.6) nous fournit l'expression de B' :

$$B'_{ell} = \frac{1}{dm^2} A_{ell} \mathbf{\Delta}_{ell} \mathbf{\Delta}_{ell}^\top A_{ell} + \frac{m}{d} A_{ell} \tag{5.4}$$

Il est alors possible d'en déduire les expressions de $\boldsymbol{\Delta}$ et B' dans la base monde :

$$\boldsymbol{\Delta}_w = {}^wR_{ell}\boldsymbol{\Delta}_{ell} \tag{5.5}$$

$$B'_w = {}^wR_{ell}B'_{ell}{}^wR_{ell}^\top \tag{5.6}$$

puis on retrouve les positions des sommets des cônes :

$$\boldsymbol{E}_w = \boldsymbol{C}_w + \boldsymbol{\Delta}_w \tag{5.7}$$

□

Résultat 5. *Lorsque l'ellipsoïde est non dégénéré, le cône de rétroprojection est un cône elliptique non dégénéré.*

Démonstration. Raisonnons par l'absurde et supposons que le cône de rétroprojection possède un axe de révolution.

Supposons également que le centre de l'ellipsoïde n'appartienne pas à cet axe. Puisque l'ellipsoïde est tangent au cône, tout nouvel ellipsoïde obtenu par rotation du premier autour de l'axe de révolution doit toujours être tangent au cône, donc être solution de l'équation (2.6). Or, dans ce cas, le lieu des centres de ces ellipsoïdes serait un cercle situé dans un plan orthogonal à l'axe et dont le centre appartiendrait à cet axe. Tous ces points seraient donc à égale distance du sommet du cône. Autrement dit, il existerait une infinité de $\boldsymbol{\Delta}$ solutions pour un même σ, puisque si on applique $tr()$ à l'équation (4.1), on constate qu'à une valeur de $||\boldsymbol{\Delta}||$ correspond une unique valeur de σ. Or cela entre en contradiction avec le résultat 4 (équation (5.3)), page 86). Donc le centre de l'ellipsoïde doit appartenir à l'axe de révolution du cône.

Si le centre de l'ellipsoïde appartenait à l'axe de révolution du cône, alors le vecteur $\boldsymbol{\Delta}$ serait colinéaire à cet axe, donc serait un vecteur propre de B', et a fortiori de A d'après l'équation (2.7). Or dans ce cas, les symétries du couple cône - ellipsoïde imposeraient que l'ellipse de tangence (intersection de l'ellipsoïde et du plan polaire issu de E [Wyl08]) appartienne à un plan orthogonal à l'axe de révolution du cône, qui est aussi un axe principal de l'ellipsoïde. Ainsi, cet ellipse de tangence devrait à la fois être un cercle (section du cône de révolution par un plan orthogonal à son axe) et une ellipse non dégénérée (section de l'ellipsoïde non dégénéré par un plan parallèle à un de ses plans principaux), ce qui est impossible. Donc le cône ne peut pas posséder d'axe de révolution. □

Résultat 6. *Lorsque l'ellipsoïde est non dégénéré, chaque cône de rétroprojection tangent à l'ellipsoïde permet de définir deux caméras solutions (cf. figure 5.2).*

Démonstration. Chaque cône $(\boldsymbol{E}_w, B'_w)$ permet de définir une position de caméras (sommet du cône), et les orientations ${}^wR_{cam}$ de ces caméras vérifient :

$$B'_w = {}^wR_{cam}B'_c{}^wR_{cam}^\top$$

Il est possible d'obtenir ${}^wR_{cam}$ en la décomposant sous la forme d'un produit de matrices de rotation vers la base propre du cône (base orthonormée dans laquelle B' est diagonale) :

$${}^wR_{cam} = {}^wR_{cone}{}^{cam}R_{cone}^\top$$

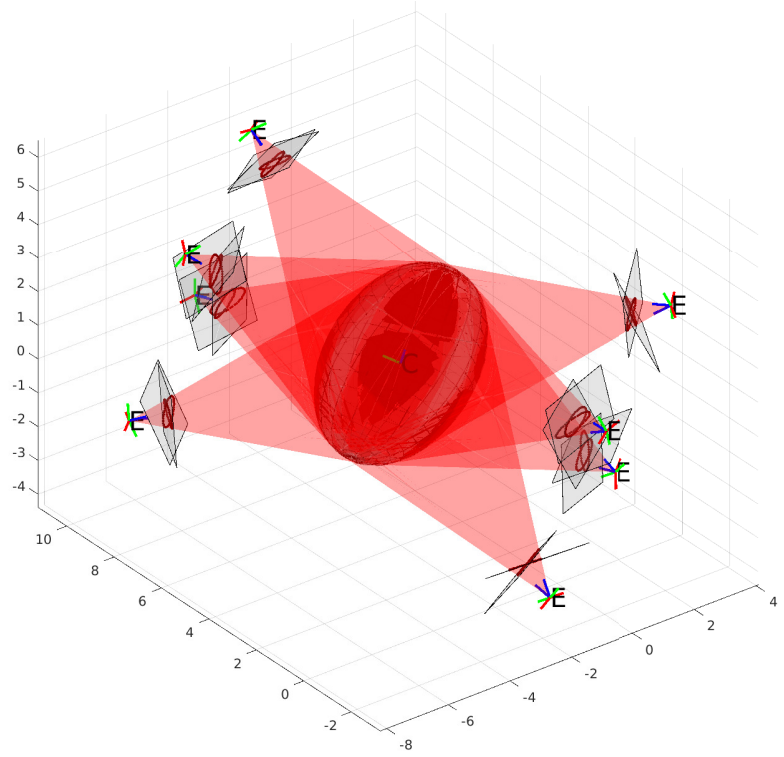


FIGURE 5.2 – Illustration des deux caméras compatibles avec chaque cône de rétroprojection tangent à l'ellipsoïde.

Puisque le cône est non dégénéré, les droites vectorielles propres de B' sont définies sans ambiguïté. En fixant arbitrairement le sens des vecteurs propres de B'_w par exemple, il reste ensuite quatre façons de choisir le sens des vecteurs propres de B'_{cam} pour que la matrice de passage ${}^{cam}R_{cone}$ soit une matrice de rotation. Or, sur les quatre orientations ${}^{w}R_{cam}$ qui en résultent, seulement deux permettent d'aboutir à un ellipsoïde situé en face de la caméra. □

En conclusion, et comme présenté dans la figure 5.2, un ellipsoïde non dégénéré induit un ensemble de solutions scalaires σ (caractérisées dans le théorème 2), et à chaque σ correspondent 16 poses de caméras, comprenant huit positions différentes (deux caméras par position).

5.3 Cas du sphéroïde

Lorsque l'ellipsoïde possède un axe de révolution, nous utilisons une démarche différente puisque le théorème 2 n'est plus valable (la matrice de Vandermonde constituée des valeurs propres de A n'est plus inversible). Nous déterminons cette fois l'ensemble des sphéroïdes tangents au cône de rétroprojection. Il convient de considérer deux cas, selon le caractère dégénéré ou non du cône de rétroprojection. Il est à noter que la détermination des sphéroïdes tangents au cône de rétroprojection a déjà été traitée dans [WP10], en utilisant une paramétrisation différente du problème. Les auteurs montrent notamment que dans le cas général (cône non dégénéré), il existe seulement deux sphéroïdes tangents au cône, et nous retrouvons ce résultat ci-après.

5.3.1 Cas du cône elliptique non dégénéré

De la même manière que pour le théorème 2, nous montrons qu'il est possible de caractériser les scalaires σ solutions de l'Equation d'Alignement des Cônes en fonction des valeurs propres du cône B', lorsque ce dernier est non dégénéré.

Théorème 3. *Lorsque le cône de rétroprojection est non dégénéré, le scalaire $\sigma = dm^2$ est solution de l'équation (2.6) si et seulement si les trois éléments du vecteur $M_{B'}^{-1}V'(m)$ sont simultanément positifs :*

$$M_{B'} = \begin{pmatrix} 1 & 1 & 1 \\ \lambda_{B',1} & \lambda_{B',2} & \lambda_{B',3} \\ \lambda_{B',1}^2 & \lambda_{B',2}^2 & \lambda_{B',3}^2 \end{pmatrix}$$

$$V'(m) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/dm^2 & 0 \\ 0 & 0 & 1/d^2m^4 \end{pmatrix} V(m)$$

Démonstration. $\boxed{\Longrightarrow}$ Supposons que l'Equation (2.6) soit satisfaite. Reprenons le système (5.1) introduit dans la démonstration du théorème 2 :

$$\begin{cases} \boldsymbol{\Delta}^\top \boldsymbol{\Delta} = tr(A^{-1}) - \frac{tr(B'^{-1})}{d} m \\ \boldsymbol{\Delta}^\top A \boldsymbol{\Delta} = 1 - m^3 \\ \boldsymbol{\Delta}^\top A^2 \boldsymbol{\Delta} = tr(B')dm^2 - tr(A)m^3 \end{cases}$$

Puisque $A\boldsymbol{\Delta} = \sigma B'\boldsymbol{\Delta}$, il est possible d'écrire

$$\begin{pmatrix} \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \\ \boldsymbol{\Delta}^\top A \boldsymbol{\Delta} \\ \boldsymbol{\Delta}^\top A^2 \boldsymbol{\Delta} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \\ \boldsymbol{\Delta}^\top B' \boldsymbol{\Delta} \\ \boldsymbol{\Delta}^\top B'^2 \boldsymbol{\Delta} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & dm^2 & 0 \\ 0 & 0 & d^2m^4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \\ \boldsymbol{\Delta}^\top B' \boldsymbol{\Delta} \\ \boldsymbol{\Delta}^\top B'^2 \boldsymbol{\Delta} \end{pmatrix}$$

En considérant une base vectorielle propre de B' (dans laquelle B' est diagonale), et en notant $(\boldsymbol{\Delta}_{cone,x}, \boldsymbol{\Delta}_{cone,y}, \boldsymbol{\Delta}_{cone,z})^\top$ l'expression de $\boldsymbol{\Delta}$ dans cette base, nous pouvons écrire l'équation matricielle suivante

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & dm^2 & 0 \\ 0 & 0 & d^2m^4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ \lambda_{B',1} & \lambda_{B',2} & \lambda_{B'3} \\ \lambda_{B',1}^2 & \lambda_{B',2}^2 & \lambda_{B'3}^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta}_{cone,x}^2 \\ \boldsymbol{\Delta}_{cone,y}^2 \\ \boldsymbol{\Delta}_{cone,z}^2 \end{pmatrix} = \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d} m \\ 1 - m^3 \\ tr(B')dm^2 - tr(A)m^3 \end{pmatrix}$$

i. e.

$$\begin{pmatrix} 1 & 1 & 1 \\ \lambda_{B',1} & \lambda_{B',2} & \lambda_{B'3} \\ \lambda_{B',1}^2 & \lambda_{B',2}^2 & \lambda_{B'3}^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta}_{cone,x}^2 \\ \boldsymbol{\Delta}_{cone,y}^2 \\ \boldsymbol{\Delta}_{cone,z}^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/dm^2 & 0 \\ 0 & 0 & 1/d^2m^4 \end{pmatrix} \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d} m \\ 1 - m^3 \\ tr(B')dm^2 - tr(A)m^3 \end{pmatrix}$$

soit

$$M_{B'} \begin{pmatrix} \boldsymbol{\Delta}_{cone,x}^2 \\ \boldsymbol{\Delta}_{cone,y}^2 \\ \boldsymbol{\Delta}_{cone,z}^2 \end{pmatrix} = V'(m)$$

Puisque les valeurs propres de B' sont toutes différentes, la matrice de Vandermonde $M_{B'}$ est inversible, d'où il est possible d'écrire

$$\begin{pmatrix} \boldsymbol{\Delta}^2_{cone,x} \\ \boldsymbol{\Delta}^2_{cone,y} \\ \boldsymbol{\Delta}^2_{cone,z} \end{pmatrix} = M_{B'}^{-1} V'(m)$$

Les éléments du terme de gauche sont bien tous positifs, d'où le résultat.

$\boxed{\Longleftarrow}$ Supposons maintenant que les trois éléments du vecteur $M_{B'}^{-1}V'(m)$ soient positifs. Soit $\boldsymbol{\Delta}_{cone} = (\boldsymbol{\Delta}_{cone,x}, \boldsymbol{\Delta}_{cone,y}, \boldsymbol{\Delta}_{cone,z})^\top$ un vecteur tel que

$$\begin{pmatrix} \boldsymbol{\Delta}^2_{cone,x} \\ \boldsymbol{\Delta}^2_{cone,y} \\ \boldsymbol{\Delta}^2_{cone,z} \end{pmatrix} = M_{B'}^{-1} V'(m)$$

Une telle définition est possible car les trois éléments du vecteur sont positifs.

On peut alors démontrer, à l'aide d'un logiciel de calcul formel (le code Maple correspondant est fourni en Annexe C), que, quel que soit le choix fait pour $\boldsymbol{\Delta}_{cone}$, la matrice

$$\boldsymbol{\Delta}_{cone}\boldsymbol{\Delta}_{cone}^\top + \frac{\mu}{\sigma} B'^{-1}_{cone} = \boldsymbol{\Delta}_{cone}\boldsymbol{\Delta}_{cone}^\top + \frac{m}{d} B'^{-1}_{cone}$$

possède les mêmes valeurs propres avec les mêmes multiplicités que A^{-1}. Comme ces deux matrices sont diagonalisables car symétriques, cela veut dire qu'elles sont semblables, et donc que l'équation (4.1), équivalente à (2.6), est vérifiée. □

Conséquence du théorème 3 : *Le théorème précédent est également valable dans le cas d'un ellipsoïde non dégénéré, puisque nous avons montré que le cône est dans ce cas également non dégénéré (résultat 5, page 88). Il permet alors de reconstruire les ellipsoïdes dans le repère caméra (cf. figure 5.3 et vidéo*[1]*).*

Nous allons montrer (résultat 7) que, dans le cas du sphéroïde associé à un cône non dégénéré, l'ensemble des scalaires σ solutions est réduit à un seul élément.

Considérons $(\lambda_{B',1}, \lambda_{B',2}, \lambda_{B',3})$ les valeurs propres de B', avec $\lambda_{B',1}$ et $\lambda_{B',2}$ du même signe (opposé à celui de $\lambda_{B',3}$), et faisons l'hypothèse, quitte à échanger les rôles, que $|\lambda_{B'1}| > |\lambda_{B'2}|$.

Résultat 7. *Lorsque l'ellipsoïde est un sphéroïde et que le cône de rétroprojection est non dégénéré, il existe un unique scalaire σ solution de l'Equation d'Alignement des Cônes :*

$$\sigma = \begin{cases} \frac{\lambda_{A,simple}\lambda_{B',1}}{\lambda_{B',2}\lambda_{B',3}} & si\ \lambda_{A,simple} < \lambda_{A,double} \\ \frac{\lambda_{A,simple}\lambda_{B',2}}{\lambda_{B',1}\lambda_{B',3}} & si\ \lambda_{A,simple} > \lambda_{A,double} \end{cases} \tag{5.8}$$

Cette valeur de σ permet de définir deux sphéroïdes tangents au cône, qui sont symétriques par rapport à un des plans principaux de ce dernier (voir figure 5.4).

Démonstration. D'après le théorème 3, un scalaire $\sigma = dm^2$ est solution de l'équation (2.6) si et seulement si les trois éléments du vecteur suivant sont simultanément positifs :

$$\begin{pmatrix} \boldsymbol{\Delta}^2_{cone,x}(m) \\ \boldsymbol{\Delta}^2_{cone,y}(m) \\ \boldsymbol{\Delta}^2_{cone,z}(m) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \lambda_{B',1} & \lambda_{B',2} & \lambda_{B',3} \\ \lambda^2_{B',1} & \lambda^2_{B',2} & \lambda^2_{B',3} \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/dm^2 & 0 \\ 0 & 0 & 1/d^2m^4 \end{pmatrix} \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d} m \\ 1 - m^3 \\ tr(B')dm^2 - tr(A)m^3 \end{pmatrix}$$

1. Vidéo : https://members.loria.fr/VGaudilliere/files/Ellipsoid.mp4

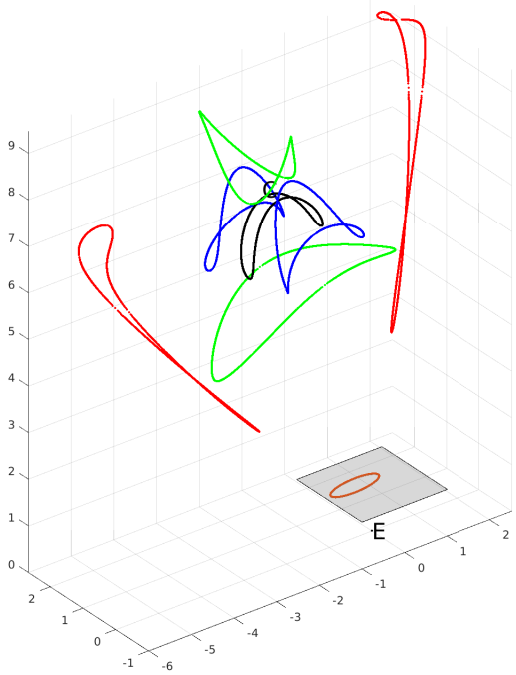


FIGURE 5.3 – Lieux des centres (noir) et des sommets des axes principaux (rouge, vert, bleu) des ellipsoïdes tangents au cône de rétroprojection. La vidéo de la construction de cette figure est disponible.

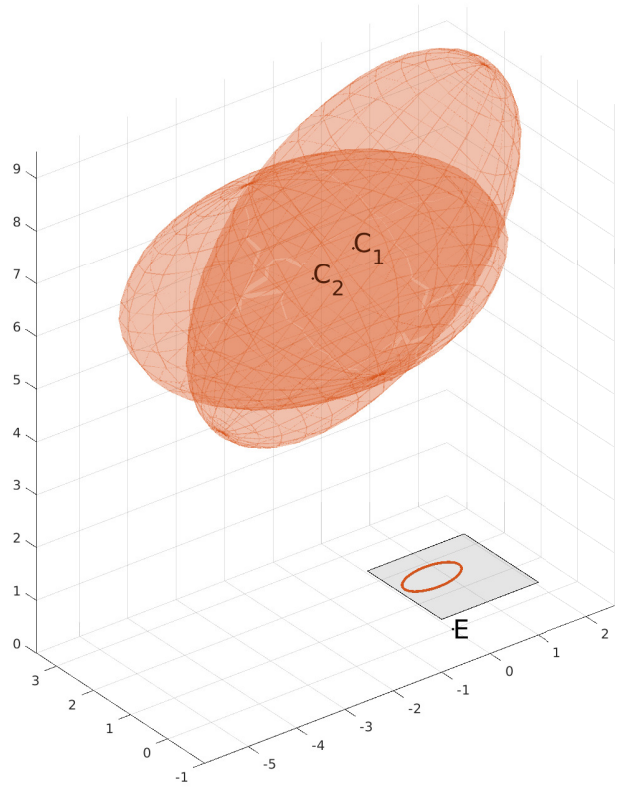


FIGURE 5.4 – Illustration des deux ellipsoïdes de révolution tangents au cône de rétroprojection.

Or, en développant le terme de droite, on obtient le système suivant

$$\begin{cases} \mathbf{\Delta}^2_{cone,x}(m) = \frac{1}{m^2} P_1(m) \\ \mathbf{\Delta}^2_{cone,y}(m) = \frac{1}{m^2} P_2(m) \\ \mathbf{\Delta}^2_{cone,z}(m) = \frac{1}{m^2} P_3(m) \end{cases}$$

où

$$\begin{cases} P_1(x) = k_1 \left(x - \frac{\lambda_{B',1}}{\lambda_{A,simple}} d\right) \left(x - \frac{\lambda_{B',1}}{\lambda_{A,double}} d\right)^2 \\ P_2(x) = k_2 \left(x - \frac{\lambda_{B',2}}{\lambda_{A,simple}} d\right) \left(x - \frac{\lambda_{B',2}}{\lambda_{A,double}} d\right)^2 \\ P_3(x) = k_3 \left(x - \frac{\lambda_{B',3}}{\lambda_{A,simple}} d\right) \left(x - \frac{\lambda_{B',3}}{\lambda_{A,double}} d\right)^2 \end{cases}$$

avec

$$\begin{cases} k_1 = \frac{-\lambda_{B',2}\lambda_{B',3}}{\lambda_{B',1}(\lambda_{B',1}-\lambda_{B',2})(\lambda_{B',1}-\lambda_{B',3})d} \\ k_2 = \frac{-\lambda_{B',1}\lambda_{B',3}}{\lambda_{B',2}(\lambda_{B',2}-\lambda_{B',1})(\lambda_{B',2}-\lambda_{B',3})d} \\ k_3 = \frac{-\lambda_{B',1}\lambda_{B',2}}{\lambda_{B',3}(\lambda_{B',3}-\lambda_{B',1})(\lambda_{B',3}-\lambda_{B',2})d} \end{cases}$$

Le lieu des solutions scalaires m est le sous-ensemble de $\mathbb{R}$ sur lequel $P_1(x)$, $P_2(x)$ et $P_3(x)$ sont simultanément positifs. L'étude des variations des $P_i(x)$ va démontrer que, dans tous les cas, ce lieu se réduit à un seul élément.

Pour réaliser cette étude de signes, nous choisissons $\lambda_{B',1}$ et $\lambda_{B',2}$ tels que $|\lambda_{B',1}| > |\lambda_{B',2}|$, puis identifions 4 configurations possibles en fonction des valeurs propres de A et B' (voir tableau 5.1).

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	#1	#2
$\lambda_{A,simple} > \lambda_{A,double}$	#3	#4

Table 5.1 – Configurations possibles du problème.

Dans un soucis de simplification, nous ne traitons ici que la configuration #1. L'ensemble des configurations est traitée en annexe D.

Les signes des valeurs propres de B' imposent que

$$d = \sqrt[3]{\frac{\lambda_{A,simple}\lambda^2_{A,double}}{\lambda_{B',1}\lambda_{B',2}\lambda_{B',3}}} > 0$$

puis que

$$k_1 < 0 \ , \ k_2 > 0 \ , \ k_3 < 0$$

Appelons S_i la racine simple de $P_i(x)$, et D_i sa racine double, de telle sorte que

$$S_i = \frac{\lambda_{B',i}}{\lambda_{A,simple}} d \qquad \text{et} \qquad D_i = \frac{\lambda_{B',i}}{\lambda_{A,double}} d.$$

Les signes des $\lambda_{B',i}$ et le fait que $\lambda_{A,simple} < \lambda_{A,double}$ imposent que

$$\begin{aligned} S_1 < D_1 < 0 \\ S_2 < D_2 < 0 \\ 0 < D_3 < S_3 \end{aligned}$$

On peut ainsi constater que les racines de $P_1(x)$ et $P_2(x)$ sont négatives, tandis que celles de $P_3(x)$ sont positives. Comme $k_3 < 0$, on a

$$\forall x \leq 0,\ P_3(x) > 0.$$

Intéressons nous maintenant aux signes de $P_1(x)$ et $P_2(x)$ pour déterminer le lieu des valeurs possibles de m. Puisque $\lambda_{B',1} < \lambda_{B',2}$, les racines des deux premiers polynômes vérifient

$$\begin{aligned} S_1 < S_2 \\ D_1 < D_2 \end{aligned}$$

Ainsi, il est possible de distinguer deux cas concernant l'ordre des racines :

$$S_1 < D_1 < S_2 < D_2$$

ou

$$S_1 < S_2 < D_1 < D_2$$

Dans le premier cas, les variations des différents polynômes sont présentées dans le tableau 5.2.

x	$-\infty$		S_1		D_1		S_2		D_2		0
$P_1(x)$		$+$	0	-	0			-			
$P_2(x)$			-				0	$+$	0	$+$	
$P_3(x)$						$+$					

TABLE 5.2 – Signe des $P_i(x)$ dans le cas $S_1 < D_1 < S_2 < D_2$.

Nous constatons que les trois polynômes ne sont jamais simultanément positifs, donc ce cas est impossible. Dans le second cas, en revanche, il existe une valeur pour laquelle les trois polynômes sont positifs : D_1 (cf. tableau 5.3).

x	$-\infty$		S_1	S_2	D_1	D_2		0
$P_1(x)$		$+$	0	-	0	-		
$P_2(x)$		-		0	$+$	0	$+$	
$P_3(x)$					$+$			

TABLE 5.3 – Signe des $P_i(x)$ dans le cas $S_1 < S_2 < D_1 < D_2$.

D'où l'unique σ solution est donné par

$$\begin{aligned}\sigma &= dD_1^2 \\ &= d^3\left(\frac{\lambda_{B',1}}{\lambda_{A,double}}\right)^2 \\ &= \frac{\lambda_{A,simple}\lambda_{A,double}^2}{\lambda_{B',1}\lambda_{B',2}\lambda_{B',3}}\left(\frac{\lambda_{B',1}}{\lambda_{A,double}}\right)^2 \\ &= \frac{\lambda_{A,simple}\lambda_{B',1}}{\lambda_{B',2}\lambda_{B',3}}\end{aligned}$$

Par ailleurs, puisque D_1 est une racine de $P_1(x)$, les vecteurs $\boldsymbol{\Delta}_{cone}$ exprimés dans la base propre du cône vérifient :

$$\begin{cases} \boldsymbol{\Delta}_{cone,x}^2 = 0 \\ \boldsymbol{\Delta}_{cone,y}^2 = \frac{1}{D_1^2}P_2(D_1) \\ \boldsymbol{\Delta}_{cone,z}^2 = \frac{1}{D_1^2}P_3(D_1) \end{cases}$$

Puisque le signe de $\boldsymbol{\Delta}_{cone,z}$ est donné par la contrainte de chiralité (ellipsoïde situé face à la caméra), il reste deux expressions possibles pour $\boldsymbol{\Delta}_{cone}$. Les deux vecteurs résultants sont symétriques par rapport au plan principal du cône dont la normale est le vecteur propre associé à la valeur propre $\lambda_{B',2}$:

$$\boldsymbol{\Delta}_{cone} = \begin{pmatrix} 0 \\ \pm\sqrt{\frac{1}{D_1^2}P_2(D_1)} \\ \sqrt{\frac{1}{D_1^2}P_3(D_1)} \end{pmatrix} \quad \text{ou bien} \quad \boldsymbol{\Delta}_{cone} = \begin{pmatrix} 0 \\ \pm\sqrt{\frac{1}{D_1^2}P_2(D_1)} \\ -\sqrt{\frac{1}{D_1^2}P_3(D_1)} \end{pmatrix} \tag{5.9}$$

Sachant que la matrice B' s'écrit dans sa base propre

$$B'_{cone} = \begin{pmatrix} \lambda_{B',1} & 0 & 0 \\ 0 & \lambda_{B',2} & 0 \\ 0 & 0 & \lambda_{B',3} \end{pmatrix}$$

l'équation (4.2) nous fournit les expressions de A dans cette même base :

$$A_{cone} = \frac{\sigma}{\mu}B'_{cone} - \frac{\sigma^2}{\mu}B'_{cone}\boldsymbol{\Delta}_{cone}\boldsymbol{\Delta}_{cone}^\top B'_{cone}$$

soit, en utilisant le fait que $\mu = m^3$ et que $m = -\sqrt{\frac{\sigma}{d}}$,

$$A_{cone} = -\frac{d\sqrt{d}}{\sqrt{\sigma}}B'_{cone} + d\sqrt{d\sigma}B'_{cone}\boldsymbol{\Delta}_{cone}\boldsymbol{\Delta}_{cone}^\top B'_{cone} \tag{5.10}$$

Il est alors possible d'en déduire les expressions de $\boldsymbol{\Delta}$ et A dans la base caméra :

$$\begin{aligned}\boldsymbol{\Delta}_{cam} &= {}^{cam}R_{cone}\boldsymbol{\Delta}_{cone} \\ A_{cam} &= {}^{cam}R_{cone}A_{cone}{}^{cam}R_{cone}^\top\end{aligned}$$

puis on retrouve les positions des centres des ellipsoïdes :

$$\boldsymbol{C}_{cam} = \boldsymbol{E}_{cam} - \boldsymbol{\Delta}_{cam}$$

□

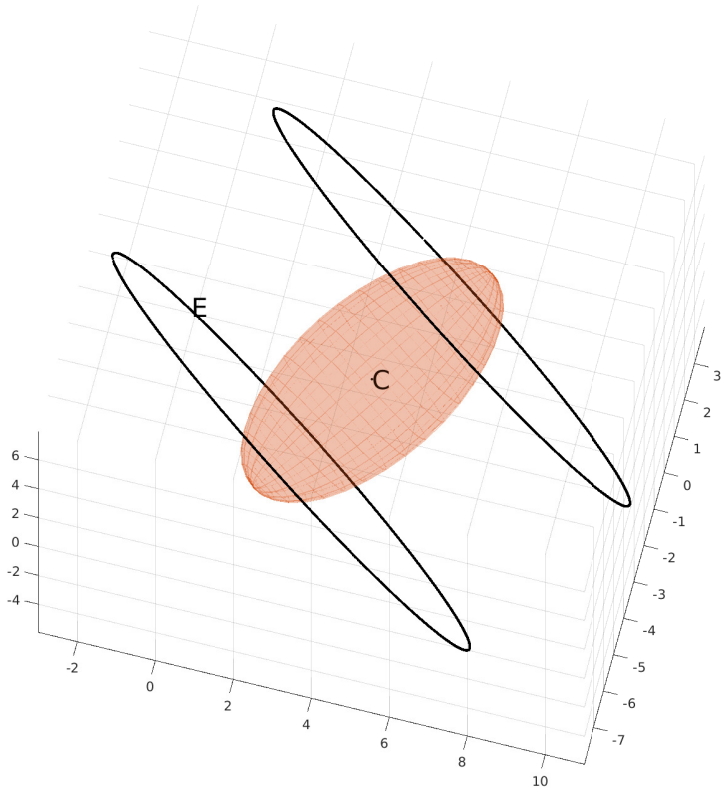


FIGURE 5.5 – Lieux des centres de caméras dans le cas d'un ellipsoïde de révolution.

Maintenant que les sphéroïdes solutions ont été déterminés dans le repère caméra, il est possible d'en déduire les poses de caméras solutions.

Résultat 8. *Lorsque l'ellipsoïde est un sphéroïde et que le cône de rétroprojection est non dégénéré, il existe une infinité de caméras solutions, et toutes ses caméras sont situées à égale distance du centre du sphéroïde (cf. figure 5.5).*

Démonstration. Comme il n'existe qu'un seul scalaire σ solution, les vecteurs $\mathbf{\Delta}$ ont tous la même norme, *i. e.* les caméras sont situées à égale distance du centre de l'ellipsoïde.

Par ailleurs, les orientations ${}^{w}R_{cam}$ de ces caméras vérifient :

$$A_{cam} = {}^{w}R_{cam}^{\top} A_w {}^{w}R_{cam}$$

Il est possible d'obtenir ${}^{w}R_{cam}$ en la décomposant sous la forme d'un produit de matrice de rotations vers une base propre de l'ellipsoïde :

$${}^{w}R_{cam} = {}^{w}R_{ell} {}^{cam}R_{ell}^{\top}$$

Puisque l'ellipsoïde est de révolution, il existe une ambiguïté sur le choix de deux des vecteurs propres de A. En fixant arbitrairement les vecteurs propres de A_w par exemple, il reste ensuite deux façons de choisir l'un des vecteurs propres de A_c, et une infinité de façons de choisir les deux autres. □

5.3.2 Cas du cône de révolution

Lorsque le cône B' possède exactement deux valeurs propres distinctes (cône de révolution), disons $\lambda_{B',simple}$ et $\lambda_{B',double}$, nous allons montrer qu'il n'existe qu'un unique sphéroïde tangent à celui-ci (résultat 9).

Résultat 9. *Lorsque l'ellipsoïde est un sphéroïde et que le cône de rétroprojection est de révolution, il existe un unique scalaire σ solution de l'Equation d'Alignement des Cônes :*

$$\sigma = \frac{\lambda_{A,simple}}{\lambda_{B',simple}} \tag{5.11}$$

Cette valeur de σ permet de définir un unique sphéroïde tangent au cône, qui a son axe de révolution confondu avec celui du cône. La distance du sommet du cône au centre du sphéroïde est donnée par :

$$||\mathbf{\Delta}|| = \sqrt{\frac{1}{\lambda_{A,simple}}\left(1 - \frac{\lambda_{A,simple}\lambda_{B',double}}{\lambda_{A,double}\lambda_{B',simple}}\right)} \tag{5.12}$$

Démonstration. Puisque A ne possède que deux valeurs propres distinctes, son polynôme minimal s'écrit

$$\begin{aligned}\pi_A(x) &= (x - \lambda_{A,simple})(x - \lambda_{A,double}) \\ &= x^2 - (\lambda_{A,simple} + \lambda_{A,double})x + \lambda_{A,simple}\lambda_{A,double}\end{aligned}$$

Puisque A est solution de son polynôme minimal, il vient

$$A^2 = (\lambda_{A,simple} + \lambda_{A,double})A - \lambda_{A,simple}\lambda_{A,double}I$$

d'où, en multipliant à gauche par $\mathbf{\Delta}^\top$ et à droite par $\mathbf{\Delta}$

$$\mathbf{\Delta}^\top A^2 \mathbf{\Delta} = (\lambda_{A,simple} + \lambda_{A,double})\mathbf{\Delta}^\top A \mathbf{\Delta} - \lambda_{A,simple}\lambda_{A,double}\mathbf{\Delta}^\top \mathbf{\Delta} \tag{5.13}$$

En injectant les expressions de $\mathbf{\Delta}^\top A^2 \mathbf{\Delta}$, $\mathbf{\Delta}^\top A \mathbf{\Delta}$ et $\mathbf{\Delta}^\top \mathbf{\Delta}$ issues de (5.1) dans (5.13), on obtient

$$tr(B')dm^2 - tr(A)m^3 = (\lambda_{A,simple} + \lambda_{A,double})(1 - m^3) - \lambda_{A,simple}\lambda_{A,double}\left(tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m\right)$$

En observant que

$$tr(A) = \lambda_{A,simple} + 2\lambda_{A,double}$$

et que

$$tr(A^{-1}) = \frac{1}{\lambda_{A,simple}} + \frac{2}{\lambda_{A,double}}$$

on obtient, après quelques développements,

$$m^3 - \frac{tr(B')d}{\lambda_{A,double}}m^2 + \frac{\lambda_{A,simple}tr(B'^{-1})}{d}m - \frac{\lambda_{A,simple}}{\lambda_{A,double}} = 0$$

Ainsi, on constate que m est une racine du polynôme

$$P_{spheroid}(x) = x^3 - \frac{tr(B')d}{\lambda_{A,double}}x^2 + \frac{\lambda_{A,simple}tr(B'^{-1})}{d}x - \frac{\lambda_{A,simple}}{\lambda_{A,double}}$$

Puis, en développant $tr(B')$ et $tr(B'^{-1})$, on peut observer que ce polynôme se factorise sous la forme

$$P_{spheroid}(x) = \left(x - \frac{\lambda_{B',simple}}{\lambda_{A,double}}d\right)\left(x - \frac{\lambda_{B',double}}{\lambda_{A,double}}d\right)^2$$

Or d est du signe de $\lambda_{B',simple}$:

$$d = \sqrt[3]{\frac{det(A)}{det(B')}} = \sqrt[3]{\frac{\lambda_{A,simple}\lambda_{A,double}^2}{\lambda_{B',simple}\lambda_{B',double}^2}}$$

Donc les signes des racines de $P_{spheroid}(x)$ sont :

$$\frac{\lambda_{B',simple}}{\lambda_{A,double}}d > 0 \qquad \text{et} \qquad \frac{\lambda_{B',double}}{\lambda_{A,double}}d < 0$$

Comme $m < 0$, il ne peut prendre que la deuxième valeur, et σ est alors donné par :

$$\begin{aligned}
\sigma &= d\left(\frac{\lambda_{B',double}}{\lambda_{A,double}}d\right)^2 \\
&= d^3\left(\frac{\lambda_{B',double}}{\lambda_{A,double}}\right)^2 \\
&= \frac{\lambda_{A,simple}\lambda_{A,double}^2}{\lambda_{B',simple}\lambda_{B',double}^2}\left(\frac{\lambda_{B',double}}{\lambda_{A,double}}\right)^2 \\
&= \frac{\lambda_{A,simple}}{\lambda_{B',simple}}
\end{aligned}$$

Intéressons nous maintenant à $\mathbf{\Delta}$, et reprenons les deux premières équations du système 5.1 :

$$\begin{cases} \mathbf{\Delta}^\top\mathbf{\Delta} = tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m \\ \mathbf{\Delta}^\top A\mathbf{\Delta} = 1 - m^3 \end{cases}$$

En considérant une base vectorielle dans laquelle A est diagonale, et en notant $(\mathbf{\Delta}_{ell,x}, \mathbf{\Delta}_{ell,y}, \mathbf{\Delta}_{ell,z})^\top$ l'expression de $\mathbf{\Delta}$ dans cette base, nous pouvons réécrire le système précédent sous la forme d'une équation matricielle :

$$\begin{pmatrix} 1 & 1 \\ \lambda_{A,simple} & \lambda_{A,double} \end{pmatrix}\begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 + \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m \\ 1 - m^3 \end{pmatrix}$$

La matrice de Vandermonde est inversible car $\lambda_{A,simple} \neq \lambda_{A,double}$, d'où

$$\begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 + \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \lambda_{A,simple} & \lambda_{A,double} \end{pmatrix}^{-1}\begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m \\ 1 - m^3 \end{pmatrix}$$

En développant le terme de droite, on aboutit finalement à

$$\begin{pmatrix} \mathbf{\Delta}_{ell,x}^2 \\ \mathbf{\Delta}_{ell,y}^2 + \mathbf{\Delta}_{ell,z}^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\lambda_{A,simple}}\left(1 - \frac{\lambda_{A,simple}\lambda_{B',double}}{\lambda_{A,double}\lambda_{B',simple}}\right) \\ 0 \end{pmatrix}$$

Ainsi,

$$\boldsymbol{\Delta}_{ell} = \begin{pmatrix} \boldsymbol{\Delta}_{ell,x} \\ \boldsymbol{\Delta}_{ell,y} \\ \boldsymbol{\Delta}_{ell,z} \end{pmatrix} = \begin{pmatrix} \pm\sqrt{\frac{1}{\lambda_{A,simple}}\left(1 - \frac{\lambda_{A,simple}\lambda_{B',double}}{\lambda_{A,double}\lambda_{B',simple}}\right)} \\ 0 \\ 0 \end{pmatrix} \tag{5.14}$$

$\boldsymbol{\Delta}$ est donc un vecteur propre de l'ellipsoïde A associé à la valeur propre $\lambda_{A,simple}$, donc est colinéaire à l'axe de révolution de l'ellipsoïde.

L'équation (2.7) $A\boldsymbol{\Delta} = \sigma B'\boldsymbol{\Delta}$ impose alors que $\boldsymbol{\Delta}$ soit également un vecteur propre du cône B', et nécessairement celui portant l'axe de révolution du cône, car sinon le centre de l'ellipsoïde serait situé hors du cône. Ainsi, les axes de révolution du cône et de l'ellipsoïde sont confondus, et la norme de $\boldsymbol{\Delta}$ est donnée par :

$$||\boldsymbol{\Delta}|| = |\boldsymbol{\Delta}_{ell,x}| = \sqrt{\frac{1}{\lambda_{A,simple}}\left(1 - \frac{\lambda_{A,simple}\lambda_{B',double}}{\lambda_{A,double}\lambda_{B',simple}}\right)}.$$

Il est ensuite possible d'en déduire A_{cone} par l'équation (5.10) (page 95), puis A_{cam} et $\boldsymbol{C}_{cam}$ (orientation et centre du sphéroïde déterminé dans le repère caméra). □

Maintenant que le sphéroïde solution a été déterminé dans le repère caméra, il est possible d'en déduire les poses de caméras solutions.

Résultat 10. *Lorsque l'ellipsoïde est un sphéroïde et que le cône de rétroprojection est de révolution, il existe deux positions de caméras solutions, qui sont situées sur l'axe de révolution du sphéroïde et à égale distance de son centre, ainsi qu'une infinité d'orientations solutions.*

Démonstration. Le résultat précédent a montré que les vecteurs $\boldsymbol{\Delta}$ solutions de l'équation 2.6 s'expriment, dans une base propre à l'ellipsoïde (dont le premier vecteur est celui qui porte l'axe de révolution du sphéroïde), sous la forme :

$$\boldsymbol{\Delta}_{ell} = \begin{pmatrix} \boldsymbol{\Delta}_{ell,x} \\ \boldsymbol{\Delta}_{ell,y} \\ \boldsymbol{\Delta}_{ell,z} \end{pmatrix} = \begin{pmatrix} \pm\sqrt{\frac{1}{\lambda_{A,simple}}\left(1 - \frac{\lambda_{A,simple}\lambda_{B',double}}{\lambda_{A,double}\lambda_{B',simple}}\right)} \\ 0 \\ 0 \end{pmatrix} \tag{5.14}$$

Par symétrie, les deux valeurs possibles pour $\boldsymbol{\Delta}_{ell}$ sont solutions.

Par ailleurs, les orientations ${}^{w}R_{cam}$ de ces caméras vérifient :

$$A_{cam} = {}^{w}R_{cam}^{\top} A_{w} {}^{w}R_{cam}$$

De la même façon que dans le cas du sphéroïde associé à un cône non dégénéré (résultat 8, page 96), nous concluons sur l'infinité des orientations de caméras possibles. □

5.4 Cas de la sphère

Lorsque l'ellipsoïde est une sphère, la matrice A s'exprime de la même façon dans toutes les bases de l'espace :

$$A = \lambda_{A,triple} I \tag{5.15}$$

où $\frac{1}{\sqrt{\lambda_{A,triple}}}$ est le rayon de la sphère. Dans ce cas, nous allons montrer qu'il n'existe qu'une unique sphère (définie par sa position) tangente au cône, qui possède nécessairement un axe de

révolution. Par symétrie, nous allons montrer que le lieu des caméras est également une sphère, dont le centre est confondu avec celui de l'objet.

Résultat 11. *Lorsque l'ellipsoïde est une sphère, le cône de rétroprojection est de révolution. De plus, il existe un unique scalaire σ solution de l'Equation d'Alignement des Cônes :*

$$\sigma = \frac{\lambda_{A,triple}}{\lambda_{B',simple}} \tag{5.16}$$

Cette valeur de σ permet de définir une unique sphère tangente au cône, et dont le centre appartient à l'axe de révolution de ce dernier. La distance du sommet du cône au centre de la sphère est donnée par :

$$||\mathbf{\Delta}|| = \sqrt{\frac{1}{\lambda_{A,triple}}\left(1 - \frac{\lambda_{B',double}}{\lambda_{B',simple}}\right)} \tag{5.17}$$

Démonstration. Puisque

$$A^{-1} = \frac{1}{\lambda_{A,triple}} I$$

l'équation (4.1), équivalente à (2.6), s'écrit

$$\mathbf{\Delta}\mathbf{\Delta}^\top = \frac{1}{\lambda_{A,triple}} I - \frac{\mu}{\sigma} B'^{-1} \tag{5.18}$$

Soit

$$B'^{-1} = -\frac{\sigma}{\mu}\left(\mathbf{\Delta}\mathbf{\Delta}^\top - \frac{1}{\lambda_{A,triple}} I\right)$$

En élevant cette égalité au carré, on obtient

$$\begin{aligned}
(B'^{-1})^2 &= \frac{\sigma^2}{\mu^2}\left(\mathbf{\Delta}\mathbf{\Delta}^\top\mathbf{\Delta}\mathbf{\Delta}^\top - \frac{2}{\lambda_{A,triple}}\mathbf{\Delta}\mathbf{\Delta}^\top + \frac{1}{\lambda^2_{A,triple}} I\right) \\
&= \frac{\sigma^2}{\mu^2}\left(\mathbf{\Delta}(\mathbf{\Delta}^\top\mathbf{\Delta})\mathbf{\Delta}^\top - \frac{2}{\lambda_{A,triple}}\mathbf{\Delta}\mathbf{\Delta}^\top + \frac{1}{\lambda^2_{A,triple}} I\right) \\
&= \frac{\sigma^2}{\mu^2}\left((\mathbf{\Delta}^\top\mathbf{\Delta})\mathbf{\Delta}\mathbf{\Delta}^\top - \frac{2}{\lambda_{A,triple}}\mathbf{\Delta}\mathbf{\Delta}^\top + \frac{1}{\lambda^2_{A,triple}} I\right) \\
&= \frac{\sigma^2}{\mu^2}\left(||\mathbf{\Delta}||^2\mathbf{\Delta}\mathbf{\Delta}^\top - \frac{2}{\lambda_{A,triple}}\mathbf{\Delta}\mathbf{\Delta}^\top + \frac{1}{\lambda^2_{A,triple}} I\right)
\end{aligned}$$

On peut alors remplacer $\mathbf{\Delta}\mathbf{\Delta}^\top$ par son expression en fonction de B'^{-1} et I (équation (5.18)) pour s'apercevoir, sans avoir à développer les calculs plus avant, que $(B'^{-1})^2$ s'exprime comme combinaison linéaire de B'^{-1} et I. Ceci revient à dire qu'il existe un polynôme annulateur de B'^{-1} de degré 2, donc que la matrice B'^{-1}, et a fortiori la matrice B', ne possède qu'au plus deux valeurs propres distinctes. De plus, la nature de B' impose que ses valeurs propres soient non nulles, et que l'une d'elles soit du signe opposé à celui des deux autres. Donc B' ne peut pas posséder une seule valeur propre, d'où elle en possède exactement deux.

Appelons $\lambda_{B',simple}$ et $\lambda_{B',double}$ ses valeurs propres. En multipliant l'égalité $A = \lambda_{A,triple} I$ à gauche par $\mathbf{\Delta}^\top$ et à droite par $\mathbf{\Delta}$, on obtient

$$\mathbf{\Delta}^\top A \mathbf{\Delta} = \lambda_{A,triple} \mathbf{\Delta}^\top \mathbf{\Delta}$$

En remplaçant ensuite $\mathbf{\Delta}^\top A \mathbf{\Delta}$ et $\mathbf{\Delta}^\top \mathbf{\Delta}$ par leurs expressions en fonction de m (système (5.1), page 85), on obtient

$$1 - m^3 = \lambda_{A,triple} \left(tr(A^{-1}) - \frac{tr(B'^{-1})}{d} m \right)$$

Or $tr(A^{-1}) = \frac{3}{\lambda_{A,triple}}$ et $tr(B'^{-1}) = \frac{1}{\lambda_{B',simple}} + \frac{2}{\lambda_{B',double}}$, d'où l'égalité qui précède est équivalente à

$$\begin{aligned} 1 - m^3 &= \lambda_{A,triple} \left(\frac{3}{\lambda_{A,triple}} - \frac{1}{d} \left(\frac{1}{\lambda_{B',simple}} + \frac{2}{\lambda_{B',double}} \right) m \right) \\ &= 3 - \frac{\lambda_{A,triple}}{d} \frac{\lambda_{B',double} + 2\lambda_{B',simple}}{\lambda_{B',simple}\lambda_{B',double}} m \end{aligned}$$

On peut par ailleurs observer que

$$d = \sqrt[3]{\frac{det(A)}{det(B')}} = \frac{\lambda_{A,triple}}{\lambda_{B',simple}^{1/3} \lambda_{B',double}^{2/3}}$$

D'où, en injectant cela dans l'égalité précédente

$$\begin{aligned} 0 &= m^3 - \lambda_{A,triple} \frac{\lambda_{B',simple}^{1/3} \lambda_{B',double}^{2/3}}{\lambda_{A,triple}} \frac{\lambda_{B',double} + 2\lambda_{B',simple}}{\lambda_{B',simple}\lambda_{B',double}} m + 2 \\ &= m^3 - \frac{\lambda_{B',double} + 2\lambda_{B',simple}}{\lambda_{B',simple}^{2/3} \lambda_{B',double}^{1/3}} m + 2 \\ &= m^3 - \left(\frac{\lambda_{B',double}^{2/3}}{\lambda_{B',simple}^{2/3}} + 2 \frac{\lambda_{B',simple}^{1/3}}{\lambda_{B',double}^{1/3}} \right) m + 2 \end{aligned}$$

En posant

$$R = \sqrt[3]{\frac{\lambda_{B',double}}{\lambda_{B',simple}}}$$

l'égalité précédente signifie que m est une racine du polynôme

$$P_{sphere}(x) = x^3 - \left(R^2 + \frac{2}{R} \right) x + 2$$

Or il apparaît que R est une racine évidente de ce polynôme :

$$P_{sphere}(x) = (x - R) \left(x^2 + Rx - \frac{2}{R} \right)$$

Même s'il n'est pas évident d'obtenir une expression formelle des deux autres racines (le signe du discriminant du facteur de degré 2 dépend de la valeur de R), disons x_1 et x_2, on peut utiliser

les relations entre coefficients et racines du polynôme $P_{sphere}(x)$, rappelés dans le chapitre 4 partie 4.1, pour obtenir les égalités suivantes :

$$\begin{cases} x_1 + x_2 + R = 0 \\ x_1 x_2 R = -2 \end{cases}$$

Si les racines x_1 et x_2 sont complexes conjuguées, alors R est la seule valeur possible pour m. Si elles sont réelles, alors, puisque $R < 0$ (les valeurs propres de B' sont de signes opposés), la deuxième égalité impose que x_1 et x_2 soient de même signe, et la première égalité impose alors qu'elles soient de signe positif. Or on a déjà vu que $m < 0$, donc nécessairement

$$m = R$$

La valeur de σ correspondante est :

$$\begin{aligned} \sigma &= dR^2 \\ &= \frac{\lambda_{A,triple}}{\lambda_{B',simple}^{1/3}\lambda_{B',double}^{2/3}} \frac{\lambda_{B',double}^{2/3}}{\lambda_{B',simple}^{2/3}} \\ &= \frac{\lambda_{A,triple}}{\lambda_{B',simple}} \end{aligned}$$

En appliquant $tr()$ à l'équation (5.18) (page 100), on obtient la valeur de $||\mathbf{\Delta}||^2$:

$$||\mathbf{\Delta}||^2 = \frac{3}{\lambda_{A,triple}} - \frac{\mu}{\sigma}\left(\frac{1}{\lambda_{B',simple}} + \frac{2}{\lambda_{B',double}}\right)$$

Or

$$\sigma = \frac{\lambda_{A,triple}}{\lambda_{B',simple}}$$

et

$$\mu = m^3 = \frac{\lambda_{B',double}}{\lambda_{B',simple}}$$

d'où

$$\begin{aligned} ||\mathbf{\Delta}||^2 &= \frac{3}{\lambda_{A,triple}} - \frac{\lambda_{B',double}}{\lambda_{A,triple}}\left(\frac{1}{\lambda_{B',simple}} + \frac{2}{\lambda_{B',double}}\right) \\ &= \frac{3}{\lambda_{A,triple}} - \frac{\lambda_{B',double}}{\lambda_{B',simple}}\frac{1}{\lambda_{A,triple}} - \frac{2}{\lambda_{A,triple}} \\ &= \left(1 - \frac{\lambda_{B',double}}{\lambda_{B',simple}}\right)\frac{1}{\lambda_{A,triple}} \end{aligned}$$

□

Résultat 12. *Lorsque l'ellipsoïde est une sphère, il existe une infinité de caméras solutions, et le lieu des positions de caméras forme une sphère de rayon $||\mathbf{\Delta}||$ autour du centre de l'ellipsoïde.*

Démonstration. Comme il n'existe qu'un seul scalaire σ solution, les vecteurs $\mathbf{\Delta}$ ont tous la même norme, *i. e.* les caméras solutions sont situées à égale distance du centre de l'ellipsoïde.

Par symétrie, nous pouvons conclure que toutes les points situées à une distance $||\mathbf{\Delta}||$ du centre de la sphère sont des positions de caméras solutions. □

5.5 Etude de la sensibilité au bruit

Dans ce chapitre, nous avons déterminé théoriquement l'ensemble des caméras vérifiant une correspondance ellipse - ellipsoïde. Nous proposons ici une étude de la sensibilité de la méthode au bruit de détection sur les ellipses, dans le cas d'ellipsoïdes non dégénéré (méthode présentée dans la partie 5.2).

Nous avons montré que le lieu des positions de caméras solutions est une courbe paramétrée par un seul paramètre. Nous considérons maintenant deux ellipsoïdes définis de telle sorte que leurs lieux respectifs se coupent en un unique point (l'union des deux ellipsoïdes ne présente aucune symétrie). Nous avons ensuite étudié la robustesse de notre méthode de détermination des solutions vis-à-vis du bruit de détection sur les ellipses.

Pour cela, nous avons mené des tests sur la base d'images T-LESS [HHO+17] (la taille des images est de 2560×1920 pixels). Nous avons considéré deux objets dans une dizaine d'images, et les avons reconstruits sous forme d'ellipsoïdes en utilisant la méthode présentée dans [RCD18]. Puis nous avons projeté ces ellipsoïdes dans des nouvelles images en utilisant la matrice de projection correcte. La figure 5.6 montre les résultats sur une image représentative de la base. Les ellipses obtenues sont présentées en vert (colonne 1). Ensuite, nous avons tiré 6 points au hasard sur ces ellipses, et leur avons appliqué un déplacement de 1, 5 puis 10 pixels dans des directions tirées au hasard. Nous appelons cette perturbation le *bruit de détection*. Nous avons ensuite interpolé ces nouveaux points pour obtenir des ellipses bruitées (ellipses rouges). Nous avons finalement déterminé les lieux de solutions issues des ellipses correctes (couleurs foncées) et bruitées (couleurs claires) en utilisant d'abord le théorème 2 (page 85), puis les résultats 4 (page 86) et 6 (page 88).

Puisque les lieux $\mathscr{L}_1$ et $\mathscr{L}_2$ issus des ellipses bruitées ne se coupent pas nécessairement dans l'espace, nous définissons comme *intersection* de ces lieux le centre du segment le plus court reliant les points de chaque lieu. Plus précisément, la distance entre lieux est donnée par :

$$d(\mathscr{L}_1, \mathscr{L}_2) = \min_{M \in \mathscr{L}_1} \min_{N \in \mathscr{L}_2} d(M, N)$$

Les extrémités de ce segment sont définies par :

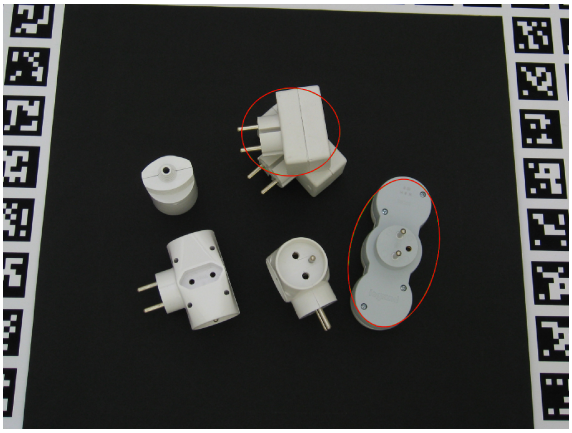
$$E_{\mathscr{L}_1} = \arg\min_{M \in \mathscr{L}_1} \min_{N \in \mathscr{L}_2} d(M, N)$$
$$E_{\mathscr{L}_2} = \arg\min_{N \in \mathscr{L}_2} \min_{M \in \mathscr{L}_1} d(M, N)$$

L'intersection des lieux $\mathscr{L}_1$ et $\mathscr{L}_2$ est définie comme le milieu du segment :

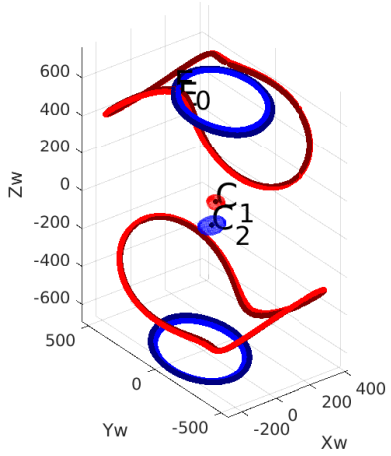
$$E = \frac{1}{2}\left(E_{\mathscr{L}_1} + E_{\mathscr{L}_2}\right)$$

La vérité terrain de la position de la caméra est nommée E_0. Les points E et E_0 sont présentés dans la figure 5.6 (colonne 2). Dans le tableau de résultats 5.4, nous examinons la distance entre E_0 et chacun des lieux :

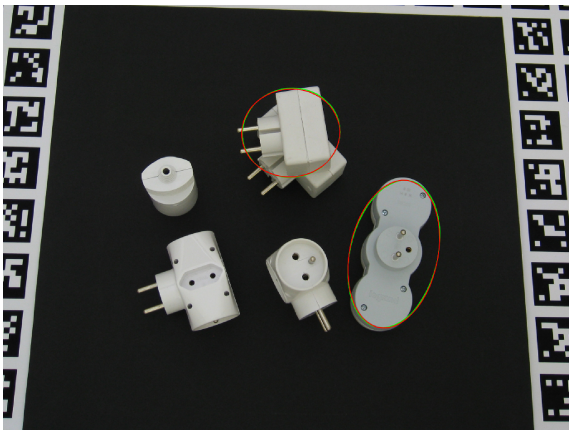
$$d(E_0, \mathscr{L}_1) = \min_{M \in \mathscr{L}_1} d(M, E_0)$$
$$d(E_0, \mathscr{L}_2) = \min_{N \in \mathscr{L}_2} d(N, E_0)$$



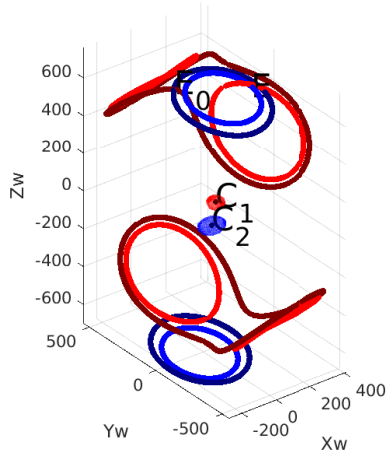
(a) Ellipses (bruit : 1 pixel).



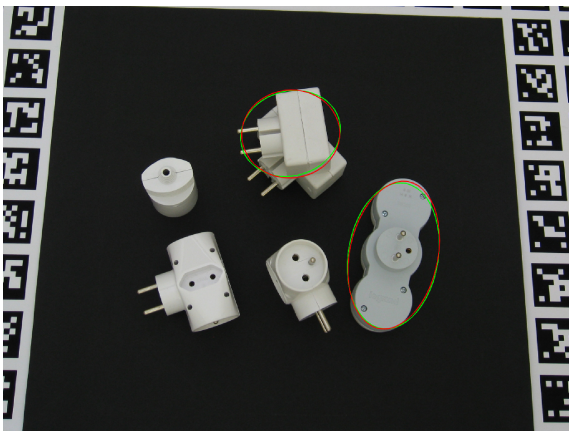
(b) Trajectoires (bruit : 1 pixel).



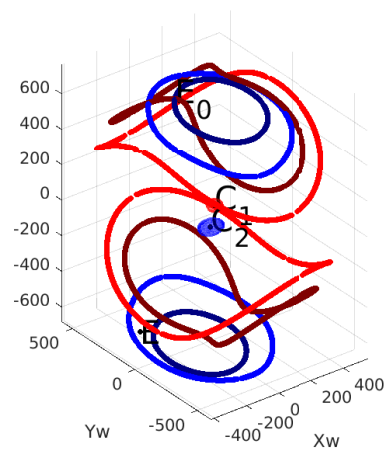
(c) Ellipses (bruit : 5 pixels).



(d) Trajectoires (bruit : 5 pixels).



(e) Ellipses (bruit : 10 pixels).



(f) Trajectoires (bruit : 10 pixels).

FIGURE 5.6 – Effet du bruit de détection des ellipses (colonne 1) sur les lieux des caméras solutions (colonne 2). Les ellipses correctes sont en vert, et les ellipses bruitées en rouge. Les trajectoires correctes de caméras sont en bleu et rouge foncé, et les trajectoires issues des ellises bruitées sont en bleu et rouge clair.

Puisqu'il existe deux orientations de caméras ${}^{w}R^{(1)}_{cam}(E_{\mathscr{L}_1})$ et ${}^{w}R^{(2)}_{cam}(E_{\mathscr{L}_1})$ (respectivement ${}^{w}R^{(1)}_{cam}(E_{\mathscr{L}_2})$ et ${}^{w}R^{(2)}_{cam}(E_{\mathscr{L}_2})$) associées à la position $E_{\mathscr{L}_1}$ (resp. $E_{\mathscr{L}_2}$) (cf résultat 5, page 88), nous définissons la distance angulaire entre les lieux $\mathscr{L}_1$ et $\mathscr{L}_2$ comme la plus petite des quatre distances entre ces orientations :

$$d^{\circ}(\mathscr{L}_1, \mathscr{L}_2) = \min_{i\in\{1,2\}} \min_{j\in\{1,2\}} d^{\circ}({}^{w}R^{(i)}_{cam}(E_{\mathscr{L}_1}), {}^{w}R^{(j)}_{cam}(E_{\mathscr{L}_2}))$$

Les deux orientations correspondant à cette distance minimale sont transformées en quaternions puis moyennées, comme proposé dans [MCCO07], pour obtenir l'orientation ${}^{w}R_{cam}$ de la caméra estimée, que nous comparons, dans le tableau 5.4, avec la vérité terrain ${}^{w}R_{cam0}$.

	Bruit de détection		
	1 pixel	5 pixels	10 pixels
$d(E_0, \mathscr{L}_1)$ (cm)	2.2	14.1	31.7
$d(E_0, \mathscr{L}_2)$ (cm)	2.4	5.7	14.1
$d(\mathscr{L}_1, \mathscr{L}_2)$ (cm)	0.8	0.9	10.8
$d^{\circ}(\mathscr{L}_1, \mathscr{L}_2)$ (°)	3.0	7.7	37.0
$\boldsymbol{d(E, E_0)}$ **(cm)**	**5.2**	**35.7**	**127.2**
$\boldsymbol{d^{\circ}({}^{w}R_{cam}, {}^{w}R_{cam0})}$ **(°)**	**4.2**	**29.8**	**227.3**

TABLE 5.4 – Effet du bruit de détection des ellipses sur l'estimation de la caméra. Les erreurs en position et en orientation de la caméra estimée sont présentées dans les deux dernières lignes du tableau. Les ellipsoïdes ont des longueurs d'axes comprises entre 3.0 et 6.7 cm, et la vérité terrain de la caméra est située à environ 75 cm des ellipsoïdes.

Comme illustré en figure 5.6 et dans le tableau 5.4, les lieux des caméras solutions sont sensibles au bruit de détection sur les ellipses, et tenter de les intersecter peut donner des résultats aberrants. De plus, dans certains cas, le domaine de définition de la variable σ, calculé à l'aide du théorème 2 (page 85), peut être vide, ce qui empêche alors toute détermination des caméras solutions.

5.6 Conclusion

Dans ce chapitre, nous avons étudié théoriquement le problème de détermination des caméras satisfaisant une correspondance ellipse - ellipsoïde, et en avons donné les solutions en fonction du type d'ellipsoïde considéré. Puis nous avons étudié la robustesse de ces résultats vis-à-vis du bruit de détection sur les ellipses, et avons mis en évidence une sensibilité assez importante de la méthode. Cette étude mériterait toutefois d'être approfondie, en étudiant par exemple l'utilisation du théorème 3 (page 90) pour le calcul de σ, ou en s'intéressant au cas des sphéroïdes, pour lesquelles nous disposons d'une expression analytique de σ. Dans le chapitre suivant (chapitre 6), nous proposons une méthode pour estimer l'orientation de la caméra à partir de deux correspondances ellipse - ellipsoïde, basée sur une hypothèse simplificatrice le plus souvent vérifiée en pratique. Cette estimation de l'orientation nous permet ensuite de déduire la position de la caméra, en tirant à nouveau profit du découplage présenté au chapitre 4.

Chapitre 6

Estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde

Sommaire

6.1 Estimation de pose à partir de deux paires ellipse - ellipsoïde . . . 107
6.1.1 Résumé de la méthode . . . 108
6.1.2 Orientation de la caméra . . . 108
6.1.3 Position de la caméra . . . 113
6.1.4 Algorithme d'estimation de la pose . . . 113
6.2 Estimation robuste de la pose et affinement . . . 114
6.2.1 Procédure de type RANSAC pour l'estimation de la pose . . . 114
6.2.2 Affinement de la pose . . . 114
6.3 Expériences et évaluation . . . 115
6.3.1 Expériences sur la base T-LESS . . . 115
6.3.2 Expériences sur la base Freiburg . . . 120
6.4 Conclusion . . . 121

Dans ce chapitre, nous proposons une méthode pour estimer la pose d'une caméra à partir des détections d'objets dans l'image. Nous allons montrer qu'il est possible d'approximer la pose à l'aide d'une expression analytique ne dépendant que d'un seul paramètre angulaire. Une recherche exhaustive permet d'estimer la valeur de ce paramètre. Cette approximation de la pose est basée sur deux hypothèses simplificatrices vérifiées dans la plupart des cas réels. En théorie, deux objets suffisent pour estimer la pose, mais, en pratique, un troisième objet permet d'obtenir une plus grande confiance dans le résultat proposé. Cette méthode, à l'instar de celle présentée au chapitre 4, permet également d'associer automatiquement les détections d'objets dans l'image aux instances du modèle 3D. Nous supposons toujours que les paramètres intrinsèques de la caméra sont connus.

6.1 Estimation de pose à partir de deux paires ellipse - ellipsoïde

Dans cette partie, nous présentons le processus d'estimation de la pose de la caméra dans le cas minimal de deux correspondances ellipse - ellipsoïde. La méthode exploite le découplage, introduit au chapitre 4, entre l'orientation et la position de la caméra, et repose sur une approximation analytique de la pose en fonction d'un seul paramètre angulaire.

Angle	Erreur d'approximation, en °
[T-LESS] θ_1	2.20 ($\pm$0.86)
[T-LESS] θ_2 (ellipses VT)	0.29 ($\pm$0.25)
[T-LESS] θ_2 (ellipses BE)	1.94 ($\pm$2.19)
[Freiburg] θ_1	1.33 ($\pm$1.07)
[Freiburg] θ_2 (ellipses BE)	1.12 ($\pm$0.94)

TABLE 6.1 – Erreurs angulaires moyennes ($\pm$ écart-type) dues aux approximations, sur les images utilisées pour les tests : une séquence typique de la base d'images T-LESS [HHO+17] (test_canon/08), et une sous-séquence de la base Freiburg [SEE+12] ($Fr2/desk$: 788 cameras). Les ellipses obtenues par projection des ellipsoïdes avec la vérité terrain des matrices de projection sont nommées *VT* (Vérité Terrain), et celles inscrites dans les Boîtes Englobantes sont nommées *BE*.

6.1.1 Résumé de la méthode

Pour estimer l'orientation de la caméra, notre méthode repose sur deux hypothèses faibles, qui permettent de réduire les trois degrés de liberté du problème de détermination de l'orientation à un seul. Plus précisément, nos hypothèses sont les suivantes :

1. l'angle de roulis de la caméra est nul,
2. la droite reliant les centres des ellipsoïdes se projette sur la droite reliant les centres des ellipses projetées.

Notre démarche est comparable avec [TSH+18], bien que Toft *et al.* fassent des hypothèses plus fortes que les nôtres pour atteindre le même nombre de degrés de liberté dans le processus d'estimation de la pose. En effet, ils supposent que la direction de la gravité est connue dans le système de coordonnées de la caméra (c'est-à-dire que l'axe y de la caméra est colinéaire à l'axe z du monde : plan de caméra vertical), alors que nous supposons simplement une coplanarité entre l'axe x de la caméra et le plan horizontal du monde (hypothèse 1), ce qui rajoute un degré de liberté et permet de couvrir plus de cas. Ils supposent, de plus, qu'une correspondance de points 2D - 3D est connue dans le repère de la caméra, alors que nous utilisons l'approximation selon laquelle la droite reliant les projections des centres des ellipsoïdes et celle reliant les centres des ellipses projetées coïncident (hypothèse 2).

Une fois l'orientation de la caméra calculée, sa position est déduite en utilisant les considérations théoriques sur le découplage (chapitre 4, partie 4.2.2).

6.1.2 Orientation de la caméra

La première hypothèse correspond au cas où l'axe des x de la caméra, porté par le vecteur $\boldsymbol{i_{cam}}$ (cf. figure 6.1), appartient au plan horizontal du monde (angle $\theta_1 = 0$). La deuxième hypothèse revient à dire que le vecteur unitaire $\boldsymbol{c} = (\boldsymbol{C_2} - \boldsymbol{C_1})/||\boldsymbol{C_2} - \boldsymbol{C_1}||$ reliant les centres des deux ellipsoïdes appartient au plan passant par le centre de la caméra et les centres des ellipses (angle $\theta_2 = 0$). Les angles θ_1 et θ_2 sont présentés sur la figure 6.1. Nous verrons dans les expériences que cela conduit à d'excellentes approximations. À titre d'illustration, les erreurs induites par ces approximations sur les bases d'images utilisées pour les tests sont présentées dans le tableau 6.1. Les erreurs sont de l'ordre de 2°, ou plus faibles encore.

Pour obtenir une expression analytique de l'orientation de la caméra, nous considérons tout d'abord trois bases orthonormées directes : $\mathscr{B}_w = (\boldsymbol{i_w}, \boldsymbol{j_w}, \boldsymbol{k_w})$, appelée la base *monde*, dans

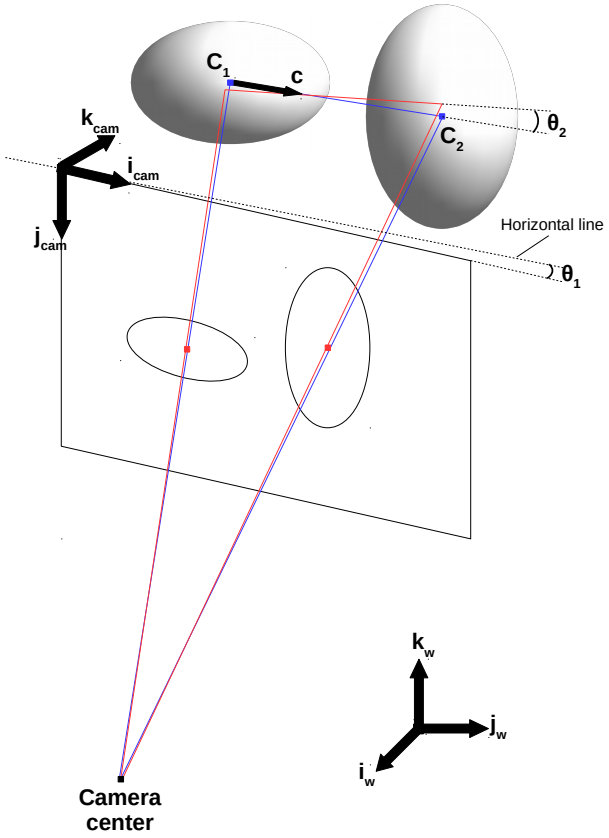


FIGURE 6.1 – Illustrations des ellipsoïdes, ellipses projetées, plan image, centre de la caméra, ainsi que des bases vectorielles associées au monde et à la caméra. Dans notre méthode, les angles θ_1 et θ_2 sont approximés à zéro.

laquelle les ellipsoïdes et le vecteur $\boldsymbol{c}$ sont connus ; $\mathscr{B}_{cam} = (\boldsymbol{i_{cam}}, \boldsymbol{j_{cam}}, \boldsymbol{k_{cam}})$, appelée la base *caméra*, dans laquelle les ellipses sont connues ; et $\mathscr{B}_p = (\boldsymbol{i_p}, \boldsymbol{j_p}, \boldsymbol{k_p})$, où $\boldsymbol{i_p}$ et $\boldsymbol{j_p}$ sont deux vecteurs orthogonaux quelconques qui appartiennent au plan passant par le centre de la caméra et les centres des ellipses (presenté en rouge dans la figure 6.1), et où $\boldsymbol{k_p} = \boldsymbol{i_p} \times \boldsymbol{j_p}$ est orthogonal à ce plan ($\times$ représente le produit vectoriel). Nous considérons enfin une quatrième base, appelée base *intermédiaire* : $\mathscr{B}_{int} = (\boldsymbol{i_{cam}}, \boldsymbol{c}, \boldsymbol{i_{cam}} \times \boldsymbol{c})$. Pour que $\mathscr{B}_{int}$ soit une base, nous supposons que $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ ne sont pas colinéaires (le cas où ils le sont est développé ci-après dans le paragraphe *Cas dégénéré*). Nous appelons enfin $\boldsymbol{v^{(b)}} = (v_x^{(b)} v_y^{(b)} v_z^{(b)})^\top$ l'expression de n'importe quel vecteur $\boldsymbol{v}$ dans n'importe quelle base $\mathscr{B}_b$. De cette manière, la matrice de changement de base entre $\mathscr{B}_{int}$ et $\mathscr{B}_w$ est donnée par

$$ {}^{w}P_{int} = \begin{pmatrix} cos(\theta_1)cos(\alpha) & c_x^{(w)} & cos(\theta_1)sin(\alpha)c_z^{(w)} - sin(\theta_1)c_y^{(w)} \\ cos(\theta_1)sin(\alpha) & c_y^{(w)} & sin(\theta_1)c_x^{(w)} - cos(\theta_1)cos(\alpha)c_z^{(w)} \\ sin(\theta_1) & c_z^{(w)} & cos(\theta_1)(cos(\alpha)c_y^{(w)} - sin(\alpha)c_x^{(w)}) \end{pmatrix} $$

où α est un angle inconnu qui donne la direction de la projection du vecteur $\boldsymbol{i_{cam}}$ sur le plan horizontal $(\boldsymbol{i_w}, \boldsymbol{j_w})$, et où la dernière colonne est obtenue comme produit vectoriel entre les deux premières. Les colonnes contiennent les expressions des vecteurs de la base $\mathscr{B}_{int}$ dans la base $\mathscr{B}_w$. En particulier, l'expression $\boldsymbol{c^{(w)}}$ de $\boldsymbol{c}$ dans la base monde (deuxième colonne) est connue. Sous l'hypothèse 1 ($\theta_1 = 0$), ${}^{w}P_{int}$ s'écrit

$$ {}^{w}\tilde{P}_{int} = \begin{pmatrix} cos(\alpha) & c_x^{(w)} & sin(\alpha)c_z^{(w)} \\ sin(\alpha) & c_y^{(w)} & -cos(\alpha)c_z^{(w)} \\ 0 & c_z^{(w)} & cos(\alpha)c_y^{(w)} - sin(\alpha)c_x^{(w)} \end{pmatrix} \tag{6.1} $$

De la même manière, la matrice de changement de base entre $\mathscr{B}_{int}$ et $\mathscr{B}_{cam}$ est donnée par

$$ {}^{cam}P_{int} = \begin{pmatrix} 1 & cos(\theta_2)(cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)}) + sin(\theta_2)k_{p,x}^{(cam)} \\ 0 & cos(\theta_2)(cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)}) + sin(\theta_2)k_{p,y}^{(cam)} \\ 0 & cos(\theta_2)(cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)}) + sin(\theta_2)k_{p,z}^{(cam)} \end{pmatrix} $$

$$ \begin{pmatrix} 0 \\ -cos(\theta_2)(cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)}) - sin(\theta_2)k_{p,z}^{(cam)} \\ cos(\theta_2)(cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)}) + sin(\theta_2)k_{p,y}^{(cam)} \end{pmatrix} $$

où β est un angle inconnu qui donne la direction de la projection du vecteur $\boldsymbol{c}$ sur le plan $(\boldsymbol{i_p}, \boldsymbol{j_p})$ passant par le centre de la caméra et les centres des ellipses. Là encore, les colonnes contiennent les expressions des vecteurs de la base $\mathscr{B}_{int}$ dans la base $\mathscr{B}_{cam}$. Il est important de noter que les expressions $(i_p^{(cam)}, j_p^{(cam)}, k_p^{(cam)})$ des vecteurs de la base $\mathscr{B}_p$ dans la base $\mathscr{B}_{cam}$ sont connues, puisqu'elles ne dépendent que des positions des centres des ellipses dans l'image et des paramètres intrinsèques de la caméra. Sous l'hypothèse 2 ($\theta_2 = 0$), ${}^{cam}P_{int}$ s'écrit

$$ {}^{cam}\tilde{P}_{int} = \begin{pmatrix} 1 & cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)} & 0 \\ 0 & cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)} & -(cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)}) \\ 0 & cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)} & cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)} \end{pmatrix} \tag{6.2} $$

L'orientation de la caméra est donnée par la matrice

$$^{w}R_{cam} = {}^{w}P_{int}\,{}^{cam}P_{int}^{-1}$$

et notre but est de calculer l'orientation approximative

$$^{w}\tilde{R}_{cam} = {}^{w}\tilde{P}_{int}\,{}^{cam}\tilde{P}_{int}^{-1}. \tag{6.3}$$

Degré de liberté Nous allons démontrer que $^{w}\tilde{R}_{cam}$ dépend seulement de α (un seul degré de liberté).

En effet, puisque les vecteurs $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ sont unitaires, l'angle γ entre eux vérifie

$$cos(\gamma) = \boldsymbol{i_{cam}} \cdot \boldsymbol{c}$$

où $\cdot$ désigne le produit scalaire entre deux vecteurs.

Puisque $\mathscr{B}_w$ est une base orthonormée, et que l'on suppose que $\boldsymbol{i_{cam}}$ est horizontal, l'utilisation des expressions de $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ dans cette base permet d'écrire

$$cos(\gamma) = \begin{pmatrix} cos(\alpha) \\ sin(\alpha) \\ 0 \end{pmatrix} \cdot \begin{pmatrix} c_x^{(w)} \\ c_y^{(w)} \\ c_z^{(w)} \end{pmatrix} = cos(\alpha)c_x^{(w)} + sin(\alpha)c_y^{(w)} \tag{6.4}$$

Puisque $\mathscr{B}_{cam}$ aussi est une base orthonormée, γ vérifie également

$$cos(\gamma) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)} \\ cos(\beta)i_{p,y}^{(cam)} + sin(\beta)j_{p,y}^{(cam)} \\ cos(\beta)i_{p,z}^{(cam)} + sin(\beta)j_{p,z}^{(cam)} \end{pmatrix} = cos(\beta)i_{p,x}^{(cam)} + sin(\beta)j_{p,x}^{(cam)} \tag{6.5}$$

L'égalité entre les expressions (6.4) et (6.5) permet d'exprimer β en fonction de α. Plus précisément, l'équation (6.5) peut se réécrire

$$cos(\gamma) = \sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}(cos(\beta)cos(\delta) + sin(\beta)sin(\delta))$$

où δ est défini tel que

$$\begin{cases} cos(\delta) = \frac{i_{p,x}^{(cam)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} \\ sin(\delta) = \frac{j_{p,x}^{(cam)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}} \end{cases}$$

Après avoir appliqué l'identité trigonométrique :

$$cos(\beta)cos(\delta) + sin(\beta)sin(\delta) = cos(\beta - \delta)$$

nous pouvons écrire

$$cos(\beta - \delta) = \frac{cos(\gamma)}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}}$$

d'où, en injectant (6.4)

$$cos(\beta - \delta) = \frac{cos(\alpha)c_x^{(w)} + sin(\alpha)c_y^{(w)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}}$$

Finalement, il ne reste que deux possibilités pour β en fonction de α :

$$\beta = \delta \pm \arccos\left(\frac{cos(\alpha)c_x^{(w)} + sin(\alpha)c_y^{(w)}}{\sqrt{{i_{p,x}^{(cam)}}^2 + {j_{p,x}^{(cam)}}^2}}\right) \tag{6.6}$$

Cas de colinéarité Si $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ sont colinéaires, la méthode d'estimation de l'orientation présentée ci-dessus ne peut pas être appliquée, puisque $\mathscr{B}_{int}$ n'est plus une base. Cependant, nous allons voir que, dans ce cas, il est possible d'exprimer directement les vecteurs de la base $\mathscr{B}_{cam}$ dans la base $\mathscr{B}_w$ en fonction d'une unique inconnue angulaire α'. L'orientation de la caméra est alors directement obtenue par

$$ {}^{w}\tilde{R}_{cam} = \begin{pmatrix} | & | & | \\ \boldsymbol{i}_{\boldsymbol{cam}}^{(\boldsymbol{w})} & \boldsymbol{j}_{\boldsymbol{cam}}^{(\boldsymbol{w})} & \boldsymbol{k}_{\boldsymbol{cam}}^{(\boldsymbol{w})} \\ | & | & | \end{pmatrix} $$

La colinéarité implique que $\boldsymbol{c}$ est horizontal ($c_z^{(w)} = 0$), et que $\boldsymbol{i_{cam}} = \pm\boldsymbol{c}$, ce qui donne, dans la base $\mathscr{B}_w$:

$$ \boldsymbol{i}_{\boldsymbol{cam}}^{(\boldsymbol{w})} = \begin{pmatrix} c_x^{(w)} \\ c_y^{(w)} \\ 0 \end{pmatrix}, \quad \text{ou} \quad \boldsymbol{i}_{\boldsymbol{cam}}^{(\boldsymbol{w})} = \begin{pmatrix} -c_x^{(w)} \\ -c_y^{(w)} \\ 0 \end{pmatrix} $$

Ensuite, puisque $\boldsymbol{i_{cam}}$ est horizontal :

$$ \boldsymbol{i_{cam}} \perp \boldsymbol{k_w} $$

et comme, par ailleurs,

$$ \boldsymbol{i_{cam}} \perp (\boldsymbol{k_w} \times \boldsymbol{i_{cam}}) $$

et

$$ \boldsymbol{i_{cam}} \perp \boldsymbol{j_{cam}} $$

on en déduit que les vecteurs $\boldsymbol{j_{cam}}$, $\boldsymbol{k_w}$ et $\boldsymbol{k_w} \times \boldsymbol{i_{cam}}$ sont coplanaires. Puisque $\boldsymbol{k_w}$ et $\boldsymbol{k_w} \times \boldsymbol{i_{cam}}$ forment une base orthonormée de ce plan, et que $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ sont colinéaires, il existe un angle α' tel que

$$ \boldsymbol{j_{cam}} = cos(\alpha')\boldsymbol{k_w} + sin(\alpha')(\boldsymbol{k_w} \times \boldsymbol{c}) $$

d'où l'expression de $\boldsymbol{j_{cam}}$ dans la base monde est :

$$ \begin{aligned} \boldsymbol{j}_{\boldsymbol{cam}}^{(\boldsymbol{w})} &= cos(\alpha') \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + sin(\alpha') \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} c_x^{(w)} \\ c_y^{(w)} \\ 0 \end{pmatrix} \\ &= cos(\alpha') \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + sin(\alpha') \begin{pmatrix} -c_y^{(w)} \\ c_x^{(w)} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} -sin(\alpha')c_y^{(w)} \\ sin(\alpha')c_x^{(w)} \\ cos(\alpha') \end{pmatrix} \end{aligned} $$

Le dernier vecteur de base est obtenu en faisant le produit vectoriel entre les deux premiers, en se rappelant que $||\boldsymbol{c}||^2 = {c_x^{(w)}}^2 + {c_y^{(w)}}^2 = 1$. Finalement, l'orientation de la caméra s'écrit

directement en fonction de α' :

$$ {}^{w}\tilde{R}_{cam} = \begin{pmatrix} c_x^{(w)} & -sin(\alpha')c_y^{(w)} & cos(\alpha')c_y^{(w)} \\ c_y^{(w)} & sin(\alpha')c_x^{(w)} & -cos(\alpha')c_x^{(w)} \\ 0 & cos(\alpha') & sin(\alpha') \end{pmatrix} $$

$$ \text{ou bien} \qquad {}^{w}\tilde{R}_{cam} = \begin{pmatrix} -c_x^{(w)} & -sin(\alpha')c_y^{(w)} & -cos(\alpha')c_y^{(w)} \\ -c_y^{(w)} & sin(\alpha')c_x^{(w)} & cos(\alpha')c_x^{(w)} \\ 0 & cos(\alpha') & -sin(\alpha') \end{pmatrix} \tag{6.7} $$

6.1.3 Position de la caméra

Nous avons montré que la position de la caméra peut être déduite de son orientation, dès lors qu'au moins une correspondance ellipse - ellipsoïde est connue. Le calcul de la position est détaillé au chapitre 4, partie 4.2.2. Chacune des deux paires ellipse - ellipsoïde définissant une position de caméra, le milieu du segment les reliant est finalement conservé.

6.1.4 Algorithme d'estimation de la pose

Lorsque le vecteur $\boldsymbol{c}$ est horizontal, les vecteurs $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ peuvent éventuellement être colinéaires. En pratique, nous commençons donc par appliquer un seuil sur l'angle entre le vecteur $\boldsymbol{c}$ et sa projection horizontale pour déterminer si l'éventualité de colinéarité doit être traitée ou non.

Si le test est négatif (les vecteurs $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ ne peuvent pas être colinéaires), ${}^{w}\tilde{R}_{cam}$ n'a qu'un seul degré de liberté (α). Nous décidons d'appliquer une recherche exhaustive sur les valeurs potentielles de α pour trouver la meilleure. Pour ce faire, nous discrétisons uniformément l'intervalle $[0°; 360°]$ en N valeurs. Pour chacune d'elles, nous calculons les deux valeurs de β possibles à l'aide de (6.6), et dérivons les deux orientations possibles de la caméra à l'aide de (6.3). Au total, nous calculons $2N$ orientations de caméras.

Lorsque le vecteur $\boldsymbol{c}$ est horizontal, nous effectuons une recherche exhaustive sur α et α'. Lors de la première recherche (α), nous supposons que $\boldsymbol{i_{cam}}$ et $\boldsymbol{c}$ ne sont pas colinéaires, et calculons les orientations de caméras à l'aide de (6.3). Lors de la seconde recherche (α'), nous supposons qu'ils sont colinéaires, et obtenons les orientations à partir de (6.7). Au total, nous calculons $4N$ orientations de caméras lorsque $\boldsymbol{c}$ est horizontal.

Ensuite, pour chaque orientation de caméra, nous déduisons la position correspondante en utilisant la méthode décrite dans la partie 6.1.3, et évaluons l'exactitude de la pose complète en mesurant les distances de Jaccard entre les ellipses détectées et reprojetées. Plus précisément, en considérant A et B deux régions d'une image délimitées par des ellipses, la distance de Jaccard $J(A, B)$ entre les deux ellipses est obtenue par soustraction à 1 du score IoU entre les deux régions :

$$ J(A, B) = 1 - IoU(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} $$

où $|A \cap B|$ est l'aire de l'intersection des deux ellipses, et $|A \cup B|$ l'aire de leur union. Finalement, la pose sélectionnée est celle qui minimise la distance de Jaccard moyennée sur les deux paires ellipse - ellipsoide.

6.2 Estimation robuste de la pose et affinement

En pratique, la scène est le plus souvent composée de plus de deux objets, donc notre méthode d'estimation de la pose doit être à même de gérer n'importe quel nombre de détections d'objets dans l'image. Pour cela, nous construisons un algorithme de type RANSAC pour obtenir la meilleure pose initiale possible, suivie d'une étape d'affinement pour améliorer sa précision.

6.2.1 Procédure de type RANSAC pour l'estimation de la pose

L'idée principale de la méthode est de considérer successivement chaque paire possible d'objets détectés dans l'image. Soit $N_{2Dpairs}$ le nombre de ces paires. Lorsque la correspondance entre les objets détectés dans l'image et les objets présents dans le modèle est connue, les $N_{2Dpairs}$ poses sont calculées en utilisant l'algorithme présenté dans 6.1, la seule différence étant que l'erreur de reprojection utilisée pour classer les poses potentielles est moyennée sur toutes les correspondances ellipse-ellipsoïde, et pas seulement sur les deux utilisées pour calculer la pose. Enfin, la meilleure pose globale est celle qui minimise la moyenne des distances de Jaccard entre les détections et les reprojections.

Lorsque la correspondance entre les données 2D et 3D est inconnue a priori, notre procédure de type RANSAC est capable d'intégrer facilement ce niveau d'incertitude supplémentaire. En effet, nous supposons que les détections 2D et les objets 3D sont caractérisés par des étiquettes (typiquement des classes d'objets), et les correspondances 2D - 3D potentielles sont déterminées sur la base d'une compatibilité entre étiquettes. Ensuite, pour chaque paire de détections d'objets, nous testons simplement toutes les combinaisons possibles entre elles et les objets 3D compatibles. Cette recherche exhaustive est possible car le nombre d'objets dans une scène, même de grande taille, reste relativement faible (des dizaines d'objets au maximum). Pour chaque pose de caméra, la correspondance 2D - 3D finale est déterminée comme l'ensemble des paires ellipse - ellipsoïde compatibles pour lesquelles la distance de Jaccard est inférieure à un certain seuil (0,5 dans les expériences). De telles paires sont appelées paires *inliers*. La meilleure pose parmi l'ensemble de poses calculées est celle qui maximise le nombre d'inliers, et qui minimise la distance de Jaccard moyenne lorsque les nombres d'inliers sont égaux.

Cette procédure n'est pas un RANSAC puisque nous traitons toutes les correspondances possibles. Toutefois, nous utilisons les notions d'inliers et d'outliers pour mesurer les performances de chaque modèle, et c'est pourquoi nous utilisons tout de même ce terme.

6.2.2 Affinement de la pose

Une fois qu'une première estimation de la pose de la caméra a été calculée, on peut appliquer une étape d'affinement qui consiste à optimiser une erreur de reprojection des ellipsoïdes en fonction des paramètres de pose de la caméra. Là encore, le paradigme de modélisation ellipse - ellipsoïde considéré dans cette méthode permet de réduire le nombre de paramètres de la fonction objectif de six à trois. Les avantages et les limites d'une telle méthode sont examinés dans la partie 6.3.1.

En outre, rien n'empêche de modifier cette étape d'affinement pour tirer parti des indices locaux (*e.g.* points, segments), par exemple en utilisant tout algorithme d'optimisation de pose qui nécessite une estimation initiale.

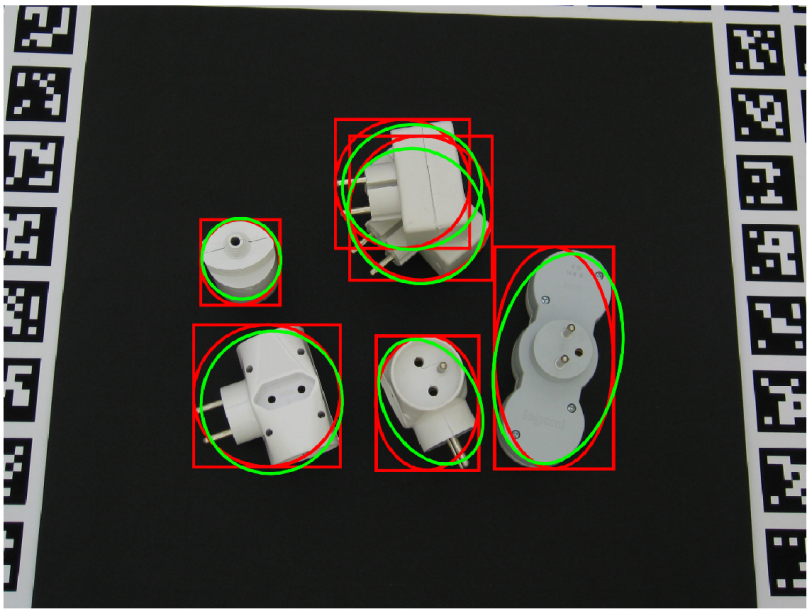


FIGURE 6.2 – Pour simuler des détections d'objets, des boîtes englobantes (en rouge) sont placées autour de chaque masque d'objet (non représentés). Dans certaines expériences, les ellipses inscrites (en rouge) sont considérées comme projections des ellipsoïdes du modèle. Cela induit une erreur sur les ellipses détectées par rapport à la vérité terrain (en vert), obtenue en projetant les ellipsoïdes avec la vérité terrain des matrices de projection.

6.3 Expériences et évaluation

Nous avons évalué la robustesse de notre méthode à partir d'expériences menées sur les bases d'images T-LESS [HHO+17] et Freiburg [SEE+12].

6.3.1 Expériences sur la base T-LESS

La base d'images T-LESS [HHO+17] est composée de douze scènes comprenant environ 500 caméras chacune. Chaque scène contient quelques objets symétriques très faiblement texturés, de dimensions comprises entre 10 et 30 cm, et qui sont placés les uns à côté (ou sur) les autres. Les caméras sont approximativement situées sur une demi-sphère de rayon 75 cm autour du barycentre des objets. Dans les expériences, nous traitons la séquence *test_ canon/08* qui contient 504 images et 6 objets. Il est important de noter que nous ne considérons que les images RVB et ignorons les informations de profondeur.

Dans ce qui suit, nous considérons soit les ellipses *vérité terrain* (VT), c'est-à-dire les ellipses obtenues par reprojection des ellipsoïdes avec les matrices de projection correctes, soit les ellipses *boîtes englobantes* (BE), *i.e.* celles inscrites dans les boîtes de détection des objets 2D. La différence entre ces deux types d'ellipses est illustrée en figure 6.2.

Quel que soit le type d'ellipses considéré, nous supposons que le problème d'association entre les données 2D et 3D est résolu, c'est-à-dire que la correspondance correcte entre les détections d'objets et les objets du modèle est connue au moment des tests.

Ellipses	Nb. d'objets	Erreur en orientation (°)	Erreur en position (cm)
VT	2	3.37 (± 31.62)	3.99 (± 23.65)
	3	2.71 (± 0.96)	3.03 (± 1.44)
	4	2.51 (± 0.91)	2.77 (± 1.38)
	5	2.50 (± 0.90)	2.83 (± 1.37)
	6	2.46 (± 0.89)	2.76 (± 1.36)
BE	2	9.99 (± 65.07)	12.23 (± 43.06)
	3	4.41 (± 7.77)	6.14 (± 9.33)
	4	3.78 (± 2.56)	5.03 (± 3.18)
	5	3.36 (± 2.18)	4.48 (± 2.67)
	6	3.15 (± 1.96)	4.09 (± 2.42)

TABLE 6.2 – T-LESS : Erreurs médianes (± écart-type), sur l'ensemble des caméras, de notre méthode d'estimation de pose de type RANSAC.

Estimation de pose de type RANSAC

Afin d'évaluer la précision de notre méthode en fonction du nombre d'objets présents dans la scène, nous avons fait varier le nombre N d'objets considérés ($2 \leq N \leq 6$). Pour chaque image, nous avons ensuite choisi au hasard N objets, et avons calculé la pose de la caméra. L'influence du biais de détection (type d'ellipses) est aussi examinée. Les résultats sont présentés dans le tableau 6.2.

On peut noter que le cas à seulement 2 objets est difficile. En effet, la pose est calculée à partir de ces deux paires ellipse - ellipsoïde, sans aucune correspondance supplémentaire pour valider son exactitude, conduisant à des poses aberrantes dans un nombre de cas significatif, mis en évidence par les valeurs importantes d'écart-type. Malgré cela, notre méthode atteint un niveau de précision de pose acceptable, compte tenu de l'erreur induite par nos hypothèses simplificatrices (voir le tableau 6.1 en comparaison), ainsi que du biais de détection éventuel sur les ellipses (BE). Une propriété importante est l'amélioration de la précision lorsque le nombre d'objets dans la scène augmente.

Affinement de la pose basé objet

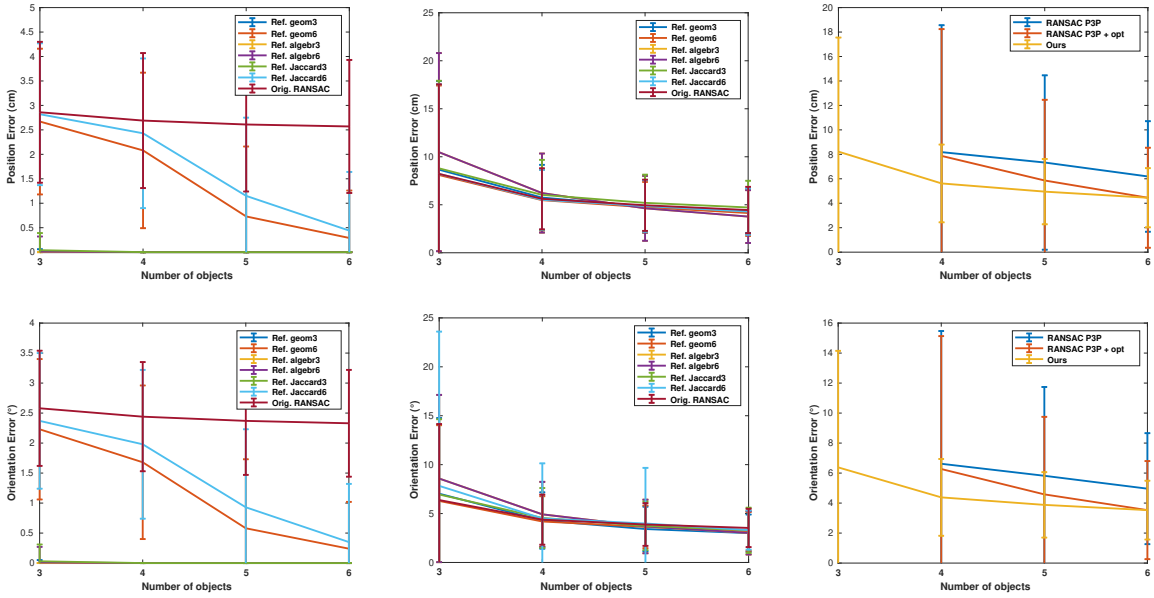
Nous avons ensuite affiné les poses obtenues (désignées par *Orig. RANSAC* dans la figure 6.3 (deux premières colonnes) en optimisant trois différents types d'erreurs de reprojection. Le premier est l'erreur géométrique introduite dans [NMS19] (distance entre les sommets des boîtes englobantes des ellipses détectées et reprojetées, représentées respectivement par les vecteurs $\mathbf{b}_i$ et $\beta_{(\mathbf{x},\mathbf{q}_i)}$ dans l'équation (6.8)), le second est l'erreur algébrique dérivée de [CRD16, RCD18] (distance algébrique entre les vecteurs formés par les 5 paramètres caractérisant les ellipses duales en coordonnées homogènes : voir l'équation (6.9), et le chapitre 2 pour les notations), et le troisième est la distance de Jaccard présentée dans la partie 6.1.4.

$$\text{Erreur géométrique : } \sum_i ||\mathbf{b}_i - \beta_{(\mathbf{x},\mathbf{q}_i)}||^2 \tag{6.8}$$

$$\text{Erreur algébrique : } \sum_i ||\beta_i C_i^* - P Q_i^* P^\top||^2 \tag{6.9}$$

Nous avons optimisé chaque type d'erreur en fonction des six paramètres de pose de la caméra (nommées *Ref. geom6*, *Ref. algebr6* et *Ref. Jaccard6*, pour *refinement*) ou seulement des

trois paramètres d'orientation, auquel cas la position de la caméra est déduite de son orientation comme expliqué dans la partie6.1.3 (*Ref. geom3*, *Ref. algebr3*, et *Ref. Jaccard3*). Les résultats sont présentés en figure 6.3.



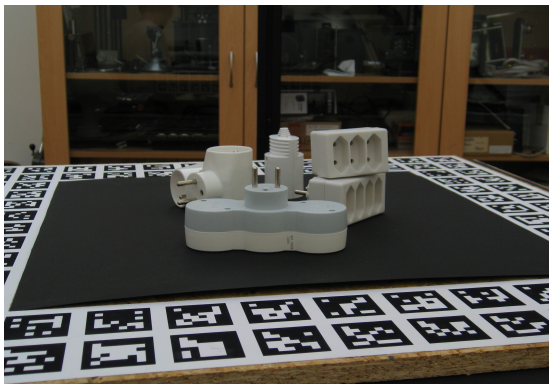
Ellipses VT | Ellipses BE | Comparaison avec PnP

FIGURE 6.3 – T-LESS : Erreurs moyennes (avec écart-type) en position et en orientation avant et après l'étape d'affinement : les ellipses VT dans la première colonne, les ellipses BE dans la deuxième, et les ellipses BE sans affinement (*ours*) en comparaison avec un PnP sur les centres des boîtes dans la dernière colonne.

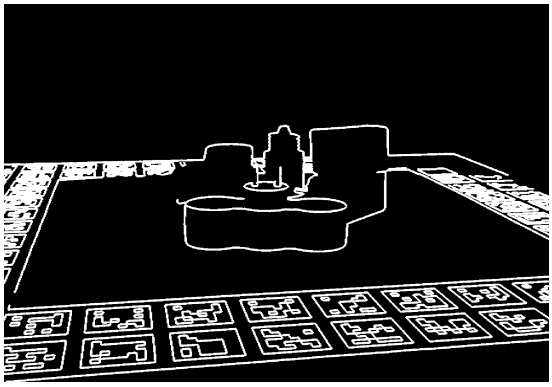
Lorsque le niveau de bruit sur les ellipses détectées est nul (ellipses VT), l'étape d'affinement permet de corriger les erreurs induites par les deux hypothèses simplificatrices initiales. Plus précisément, si les erreurs géométrique et de Jaccard à 6 degrés de liberté ne permettent pas une convergence globale lorsque le nombre d'objets est faible, les autres types d'erreurs peuvent être utilisés pour converger vers la vérité terrain de la pose (toutes les courbes correspondantes sont cachées par la courbe verte). Cependant, cela est faux lorsque l'on considère les ellipses biaisées (BE). En effet, les poses après affinement ne sont pas, en moyenne, plus précises que les poses avant affinement, et sont même souvent moins précises. Cela montre qu'une étape d'affinement basée sur des ellipses/ellipsoïdes n'est pas appropriée lorsque les ellipses considérées souffrent de biais de détection. Dans ce cas, une méthode d'optimisation de la pose basée sur des indices locaux doit être préférée. Par exemple, la figure 6.4 illustre, sur un cas typique, l'optimisation, en fonction des six paramètres de pose, de la distance entre les contours des objets projetés et les contours détectés dans l'image par un filtre de Canny.

Hypothèse sur l'angle de roulis

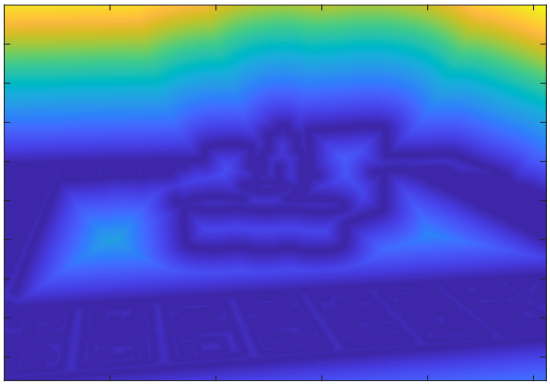
Pour évaluer la robustesse de l'optimisation sur les ellipses - ellipsoïdes par rapport à l'erreur induite par notre première hypothèse (angle de roulis θ_1 égal à 0), nous avons modifié la valeur intialement supposée pour θ_1 dans le but de créer des erreurs initiales plus importantes, puis mesuré les erreurs finales après avoir appliqué notre méthode d'estimation de la pose. Nous



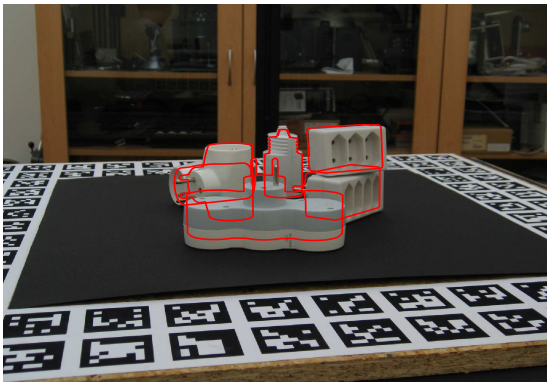
(a) Image de test



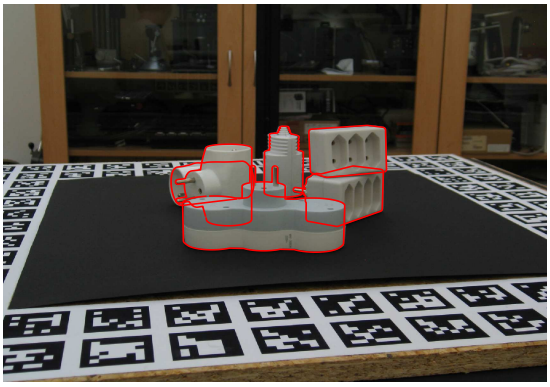
(b) Filtrage de Canny



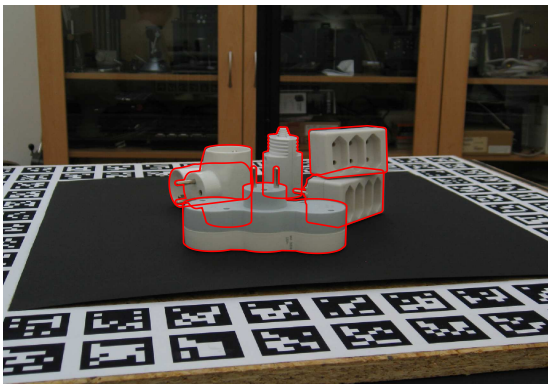
(c) Carte de distance



(d) Contours des objets projetés avec la pose avant optimisation



(e) Contours des objets projetés avec la pose après optimisation



(f) Contours des objets projetés avec la vérité terrain de la pose

FIGURE 6.4 – Illustration de l'effet d'une optimisation de la pose basée sur des indices locaux (contours). L'erreur de pose avant optimisation est (2.70° ; 4.87 cm). L'erreur après optimisation est (0.74° ; 1.06 cm).

Valeur initiale	0°	-5°	-10°
Erreur initiale	2.20 (± 0.86)	7.20 (± 0.86)	12.20 (± 0.86)
Erreur finale (3 obj.)	0.01 (± 0.10)	0.14 (± 1.08)	0.80 (± 3.10)
Erreur finale (4 obj.)	0.00 (± 0.00)	0.02 (± 0.36)	0.09 (± 0.95)
Erreur finale (5 obj.)	0.00 (± 0.00)	0.01 (± 0.21)	0.00 (± 0.00)
Erreur finale (6 obj.)	0.00 (± 0.00)	0.00 (± 0.00)	0.00 (± 0.00)

TABLE 6.3 – T-LESS : Erreurs, initiales et finales, moyennes (± écart-type) sur l'angle θ_1 (en °) en fonction de l'hypothèse initiale.

considérons les ellipses correctes VT, et optimisons l'erreur géométrique à trois degrés de liberté. Les résultats présentés dans le tableau 6.3 montrent que notre méthode converge le plus souvent vers la vérité terrain de la pose (cf. erreurs finales moyennes), et ne s'en éloigne que rarement (cf. valeurs finales d'écart-type en comparaison avec les erreurs initiales). De plus, un grand nombre d'objets présent dans la scène rend notre processus plus robuste à l'erreur initiale introduite par l'hypothèse sur θ_1.

Comparaison avec PnP

Pour évaluer l'intérêt de notre approche basée sur les ellipsoïdes pour résoudre le problème d'estimation de pose de la caméra, nous l'avons comparée à une approche basée points dans laquelle les objets sont assimilés à leurs centroïdes (centres des ellipsoïdes en 3D, centres des boîtes englobantes en 2D). Dans ce cas, un algorithme classique RANSAC P3P est utilisé pour calculer la pose de la caméra, suivi d'une optimisation à 6 degrés de liberté de l'erreur de reprojection sur les points. Il est important de noter que notre méthode ne nécessite que 2 objets pour calculer la pose, alors que l'approche par points nécessite au moins 4 objets. Cependant, le cas à 2 objets induit beaucoup d'échecs, et n'est donc pas présenté dans les résultats. Les résultats que nous reportons dans la figure 6.3 (dernière colonne) pour notre méthode sont ceux appelés *orig. RANSAC* dans la deuxième colonne (ellipses inscrites dans les boîtes englobantes, et aucune optimisation, puisque nous avons montré qu'elle n'améliorait pas la pose). La figure 6.3 (dernière colonne) montre que notre méthode est en moyenne plus précise lorsque le nombre d'objets est inférieur à 6, et que dans tous les cas, la modélisation par ellipse - ellipsoïde permet une plus grande confiance dans les résultats, puisque l'écart-type de l'erreur de pose est significativement plus faible dans notre cas.

Comparaison avec CorNet [PIL19]

À notre connaissance, presque toutes les méthodes qui ont rapporté des résultats sur le jeu de données T-LESS sont des méthodes d'estimation de pose d'objets qui nécessitent un entraînement spécifique. CorNet [PIL19] est légèrement différent, puisqu'il prédit les poses de *nouveaux* objets (objets non vus pendant l'entraînement), en détectant ses coins dans l'image et en inférant leurs poses, puis en les mettant en correspondance avec les coins du modèle 3D CAO. L'article cité présente des résultats sur l'objet 20 de la scène 08, en utilisant deux métriques différentes. La métrique 3D $ADD_{X\%}$ mesure le pourcentage d'images pour lesquelles la distance moyenne entre les points 3D transformés à l'aide des poses correcte et prédite est inférieure à $X\%$ du diamètre de l'objet. La métrique 2D $detection_X$ mesure le pourcentage d'images pour lesquelles le score IoU entre les masques corrects et reprojetés est supérieur à X. À titre de comparaison, le tableau 6.4 présente les résultats de CorNet et les nôtres.

Méthode	Métrique					
	$ADD_{10\%}$	$ADD_{20\%}$	$ADD_{30\%}$	$\text{detection}_{0.8}$	$\text{detection}_{0.5}$	$\text{detection}_{0.4}$
CorNet [PIL19]	10.0	40.4	56.1	-	-	34.1
Nous (2 obj.)	15.9	38.5	56.3	66.7	93.5	95.0
Nous (3 obj.)	23.8	50.4	70.8	81.9	99.6	99.8
Nous (4 obj.)	34.7	64.9	82.5	92.3	100.0	100.0
Nous (5 obj.)	34.7	69.2	88.7	96.4	100.0	100.0
Nous (6 obj.)	41.7	74.2	92.9	98.6	100.0	100.0

TABLE 6.4 – T-LESS : Comparaison avec CorNet [PIL19] sur la Scène 08 / Objet 20. Quelle que soit la métrique considérée, les résultats reportés sont les pourcentages d'images correctes.

Les poses obtenues avec notre méthode sont calculées à partir des ellipses BE. Comme cela a déjà été mis en avant dans le tableau 6.2, l'efficacité de notre méthode augmente avec le nombre d'objets. Bien que notre précision 3D soit comparable avec celle de CorNet lorsque seulement deux objets sont considérés, elle augmente avec plus d'objets. De plus, l'erreur faite sur les projections 2D est bien plus faible dans notre cas que dans celui de CorNet.

6.3.2 Expériences sur la base Freiburg

La base de données Freiburg [SEE+12], déjà utilisée au chapitre 4, fournit des environnements vastes et réalistes qui contiennent plusieurs objets d'intérêt, ce qui la rend adaptée à l'évaluation des méthodes d'estimation de pose de caméra à partir d'objets. Dans nos expériences, nous considérons un sous-ensemble de 788 caméras de la séquence *Freiburg2/desk*. Ces images ont été sélectionnées de telle sorte qu'au moins trois objets soient correctement détectés par YOLO [RF18] dans chacune d'entre elles.

Les modèles ellipsoïdaux des objets ont été construits en amont, à partir d'une douzaine d'images choisies parmi les 2965 images de la séquence, en utilisant la méthode décrite dans [RCD18]. Contrairement aux expériences menées sur T-LESS (partie 6.3.1), le problème d'association des données 2D et 3D n'est pas résolu ici, ce qui signifie que la méthode n'a pas connaissance de la correspondance correcte entre les détections d'objets 2D et les instances de modèles 3D. Cependant, les étiquettes YOLO ont été transférées sur les ellipsoïdes 3D pendant la phase de construction du modèle et, au moment des tests, nous utilisons notre procédure de type RANSAC étendue, présentée dans la partie 6.2.1, pour associer les données 2D et 3D tout en estimant la pose de la caméra. Les résultats sont présentés dans la figure 6.5, en comparaison avec l'approche PnP décrite précédemment.

Concernant RANSAC P3P, un petit seuil de distance conduit à éliminer la plupart des images (moins de 4 inliers), mais donne des résultats précis sur 75% des images restantes, alors qu'un seuil haut permet de calculer une pose dans un plus grand nombre de cas, mais au prix d'une dégradation de la précision. Au contraire, notre méthode sans paramètre a été capable de traiter toutes les images et fournit les résultats les plus précis : 4.76° (± 3.40°) en moyenne en orientation, et 12.26 cm (± 8.19 cm) en moyenne en position, sur les 788 images. La plus faible précision des poses ici, en comparaison avec les expériences menées sur T-LESS, vient du fait que les boîtes englobantes automatiquement détectées par YOLO souffrent très souvent d'occultations et/ou de bruit important.

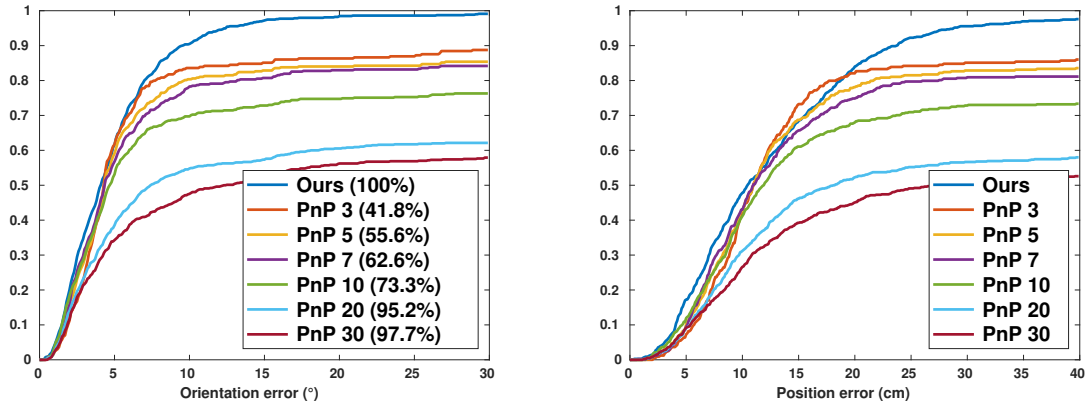


FIGURE 6.5 – Freiburg : Fonctions de distribution des erreurs en orientation (à gauche) et en position (à droite) en comparaison avec PnP. $PnP\ X$ désigne le seuil de distance X (en pixels) utilisé pour séparer les inliers des outliers. Les pourcentages entre parenthèses indiquent la proportion d'images pour lesquelles la méthode mentionnée a réussi à calculer une pose.

6.4 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode d'estimation de pose de caméra basée sur les objets, qui tire profit du paradigme ellipse - ellipsoïde introduit précédemment, et qui repose sur deux hypothèses simplificatrices qui n'introduisent que de faibles erreurs sur les données testées. L'estimation de la pose est ainsi transformée en un problème à un degré de liberté, résolu par une recherche exhaustive sur le paramètre inconnu. Les résultats ont montré une précision de la pose compatible avec une tâche de relocalisation, cette pose pouvant ensuite être affinée par une méthode itérative classique. La principale limite de notre méthode est pour l'instant le processus de détection des ellipses, puisque nous avons montré que le fait de considérer les ellipses insérées dans les boîtes détectées affecte la précision de la méthode et ne permet pas de concevoir une étape d'affinement fiable basée uniquement sur ces indices. En revanche, la précision semble suffisante, y compris lorsque les hypothèses simplificatrices sont relativement éloignées de la réalité, pour permettre à une méthode de type recalage basé contours de converger. D'autre part, la prise en compte de la vérité terrain des ellipses conduit à un niveau de précision de la pose très important.

Conclusion

Dans cette thèse, nous avons travaillé sur le positionnement visuel dans des environnements complexes, et nous nous sommes intéressés en particulier à l'information apportée par les objets présents dans la scène pour répondre à cette problématique.

Résumé des contributions

Après avoir proposé une définition fonctionnelle du concept de lieu en environnement industriel, comme zone d'interaction autour d'un objet d'intérêt, nous avons abordé la reconnaissance de lieux comme une tâche de récupération d'images dans laquelle la similarité entre l'image inconnue et les images de référence est mesurée en deux étapes, afin d'obtenir un bon compromis entre rapidité et performance. La validité des images présentant les plus grandes similarités avec l'image inconnue est ensuite évaluée par estimation de la géométrie épipolaire liant l'image inconnue et chacune des images récupérées. La mesure de similarité et l'estimation de la géométrie sont guidées par le calcul de correspondances de niveau objet entre régions d'intérêt des deux images. Cette méthode présente notamment l'avantage de ne pas nécessiter d'apprentissage spécifique à l'environnement d'application, mais est adaptée à tout type d'environnement industriel.

Une fois cette première étape d'identification de lieu traitée, nous nous sommes intéressés au positionnement. Nous avons notamment cherché à nous appuyer sur les objets d'intérêt présents dans la scène pour le calcul de pose de caméra. La modélisation des objets sous forme d'ellipsoïdes (modèle), associée à la modélisation de leurs projections dans l'image sous forme d'ellipses, nous a permis de développer un cadre théorique et des méthodes de calcul de pose fonctionnant à partir de seulement un ou deux objets. Ces méthodes présentent des performances compatibles avec une tâche de relocalisation.

Plus précisément, nous avons montré que la position de la caméra peut être déduite de son orientation et d'une correspondance ellipse - ellipsoïde. Si l'orientation est obtenue par ailleurs, nous avons ainsi montré qu'il est possible d'en déduire la position. Nous avons également résolu le problème théorique de détermination de l'ensemble des caméras satisfaisant une correspondance ellipse - ellipsoïde, en fonction du type d'ellipsoïde considéré (ellipsoïde non dégénéré, sphéroïde ou sphère), et avons conduit une première étude de la sensibilité au bruit de ces résultats.

Nous avons finalement proposé une approximation analytique de l'orientation de la caméra à partir de deux correspondances ellipse - ellipsoïde. Dans ce cas, il est possible d'obtenir l'orientation et la position de la caméra à partir de deux détections d'objets en théorie, même si une troisième est nécessaire en pratique, en raison du bruit sur les données, pour assurer une précision suffisante. Nous avons notamment montré que, en termes de calcul de pose, la modélisation des objets par des ellipses - ellipsoïdes permet d'atteindre des performances supérieures à celles obtenues par assimilation de ces mêmes objets à des points.

Perspectives

Ces travaux, qui ont fait l'objet d'un dépôt de brevet et, pour une partie d'entre eux [1], de publications dans des conférences nationales et internationales (cf. page 7), ouvrent des perspectives de recherches nombreuses et intéressantes.

A court terme, l'étude de la sensibilité au bruit des résultats théoriques du chapitre 5 mériterait d'être approfondie, notamment en ce qui concerne le cas des sphéroïdes, et une analyse des résultats en fonction du repère dans lequel le problème est résolu (cône ou ellipsoïde) devrait être menée. Ensuite, il serait intéressant de travailler sur la détection des ellipses. En effet, dans les expériences menées au cours de cette thèse, nous avons toujours considéré les ellipses inscrites dans les boîtes englobantes fournies par le détecteur d'objet, qui sont nécessairement alignées avec les axes de l'image, créant ainsi un biais de détection préjudiciable à une estimation précise de la pose. Être en mesure de détecter des ellipses plus proches de celles obtenues par reprojection des ellipsoïdes avec la vérité terrain de la matrice de projection permettrait ainsi d'améliorer les performances d'estimation de la pose. L'utilisation de détections d'objets sous forme de masques [HGDG17] pourrait éventuellement permettre cela. Les contributions théoriques apportées par ce travail de recherche pourraient également être exploitées en vue d'une reconstruction dynamique de la scène (SLAM), et non plus seulement dans le cadre d'une estimation de pose de caméra basée modèle. Le découplage orientation - position que nous avons mis en évidence et démontré permettrait notamment de réduire les degrés de liberté du problème et d'envisager une solution plus efficace que celle existante (QuadricSLAM).

Ces travaux ouvrent, par ailleurs, des perspectives très intéressantes dans les grands environnements, puisque la scène peut être résumée en quelques objets d'intérêt, sous la forme d'un modèle sémantique compact. Nous avons notamment proposé une manière de gérer les multiples correspondances possibles entre les détections d'objets dans l'image et les instances du modèle 3D, sans que la combinatoire n'explose puisque nous n'avons besoin que d'une ou deux correspondances pour estimer une position ou une orientation de caméra. Enfin, les considérations développées dans ce manuscrit interrogent la notion d'objet, en particulier vis-à-vis du positionnement, et posent la question de sa modélisation la plus efficace. Qu'est-ce qu'un bon objet pour le positionnement ? Une modélisation sous forme d'ellipsoïde non dégénéré est-elle nécessaire ? Une modélisation moins fidèle, sous forme de sphéroïde, voir de sphère permettrait-elle d'être plus robuste au bruit ?

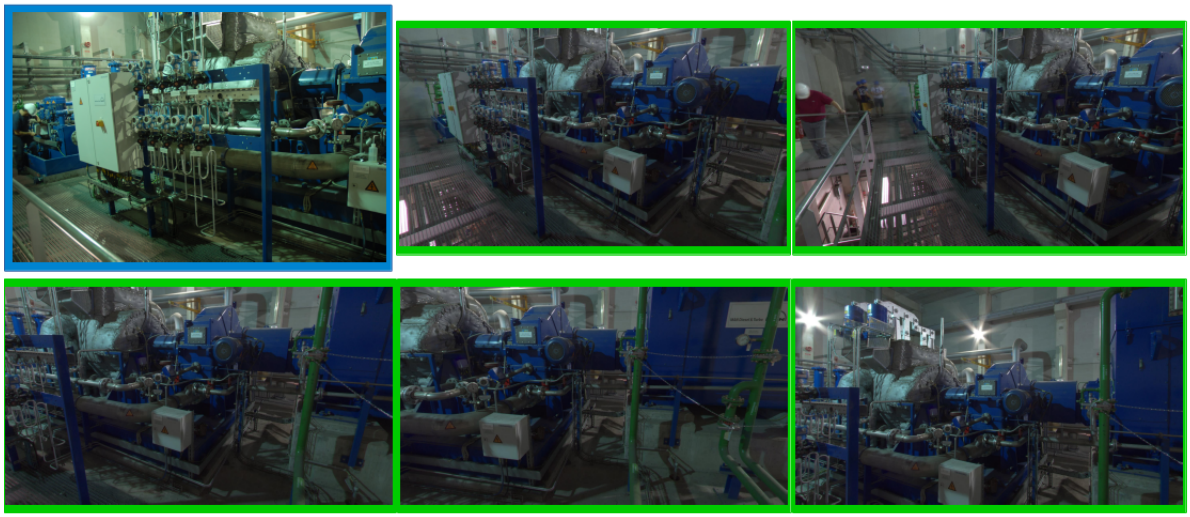
Personnellement, je me suis engagé dans la valorisation de ces travaux avec l'objectif de les confronter à des cas réels, et ainsi tenter de répondre à un certain nombre de questions encore ouvertes, concernant par exemple l'adaptation à de nouveaux environnements, le choix des objets à reconstruire, la capture d'images pour la reconstruction, la méthode de détection des objets, la problématique temps-réel pour le calcul de pose, etc.

1. Les travaux non publiés (correspondant principalement aux chapitres 5 et 6) vont faire l'objet de soumissions à des conférences et/ou journaux internationaux.

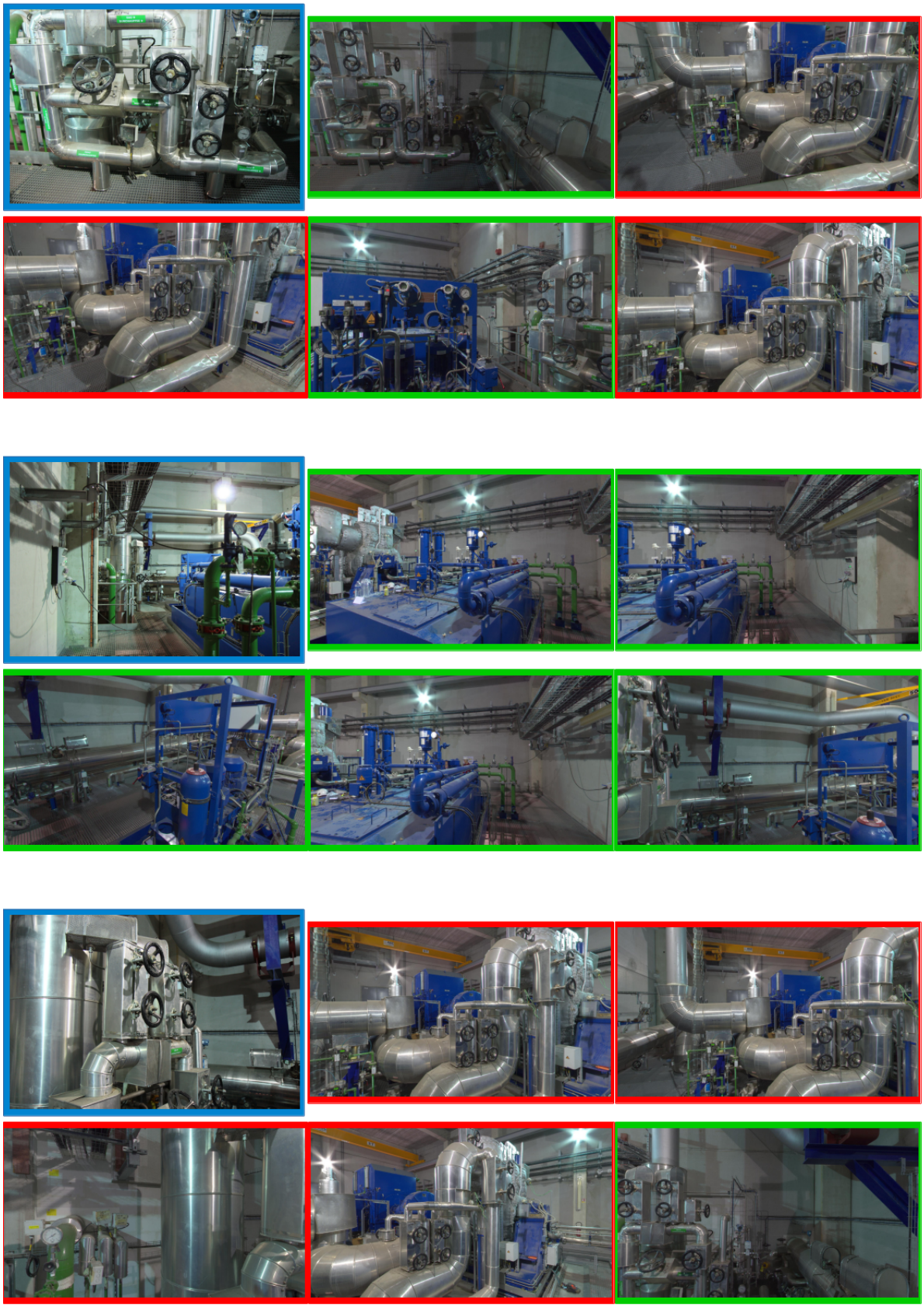
Annexe A

Exemples de résultats de notre méthode de sélection des lieux candidats par approche mixte.

Pour chaque exemple, l'image de test est présentée en haut à gauche sur fond bleu, puis les cinq meilleures images de référence sont présentées de gauche à droite et de haut en bas, de la première à la cinquième. Les images jugées correctes sont sur fond vert, celles incorrectes sur fond rouge. La décision sur le caractère correct ou incorrect des images récupérées se fait sur la base d'objets d'intérêt détourés à la main dans les images : l'image récupérée est considérée comme correcte si elle possède au moins un objet d'intérêt en commun avec l'image de test. Il est à noter que, parfois, la base d'images de référence contient moins de cinq images montrant un objet présent dans l'image inconnue. Les images récupérées ne peuvent donc pas toutes être correctes dans ce cas. Les images ont toutes été acquises dans l'Usine d'Electricité de Metz.




HYDAC
HYDAC
HYDAC



Annexe B

Théorème 2 : code Maple

```
> with(linalg);
> A:=matrix([[lA1,0,0],[0,lA2,0],[0,0,lA3]]);
> B:=matrix([[lB1,0,0],[0,lB2,0],[0,0,lB3]]);
> M_A:=transpose(vandermonde([lA1,lA2,lA3]));
> d:=(det(A)/det(B))^(1/3);
> V:=transpose(matrix([[trace(inverse(A))-trace(inverse(B))*m/d,1-m^3,
trace(B)*d*m^2-trace(A)*m^3]]));
> Delta2:=multiply(inverse(M_A),V);
> Delta:=transpose(matrix([[sqrt(Delta2[1,1]),sqrt(Delta2[2,1]),
sqrt(Delta2[3,1])]]));
> inv_B:=evalm(d/m*(inverse(A)-multiply(Delta,transpose(Delta))));
> eigenvalues(inv_B);
```

Annexe C

Théorème 3 : code Maple

```
> with(linalg);
> A:=matrix([[lA1,0,0],[0,lA2,0],[0,0,lA2]]);
> B:=matrix([[lB1,0,0],[0,lB2,0],[0,0,lB3]]);
> M_B:=transpose(vandermonde([lB1,lB2,lB3]));
> d:=(det(A)/det(B))^(1/3);
> V_:=transpose(matrix([[trace(inverse(A))-trace(inverse(B))*m/d,1-m^3,
trace(B)*d*m^2-trace(A)*m^3]]));
> V:=multiply(inverse(matrix([[1,0,0],[0,d*m^2,0],[0,0,d^2*m^4]])),V_);
> Delta2:=multiply(inverse(M_B),V);
> Delta:=transpose(matrix([[sqrt(Delta2[1,1]),sqrt(Delta2[2,1]),
sqrt(Delta2[3,1])]]));
> inv_A:=evalm(multiply(Delta,transpose(Delta))+evalm(m/d*inverse(B)));
> eigenvalues(inv_A);
```

Annexe D

Etude complète des configurations du Résultat 7.

D'après le théorème 3, un scalaire $\sigma = dm^2$ est solution de l'équation (2.6) si et seulement si les trois éléments du vecteur suivant sont simultanément positifs :

$$\begin{pmatrix} \mathbf{\Delta}^2_{cone,x}(m) \\ \mathbf{\Delta}^2_{cone,y}(m) \\ \mathbf{\Delta}^2_{cone,z}(m) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \lambda_{B',1} & \lambda_{B',2} & \lambda_{B',3} \\ \lambda^2_{B',1} & \lambda^2_{B',2} & \lambda^2_{B',3} \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/dm^2 & 0 \\ 0 & 0 & 1/d^2m^4 \end{pmatrix} \begin{pmatrix} tr(A^{-1}) - \frac{tr(B'^{-1})}{d}m \\ 1 - m^3 \\ tr(B')dm^2 - tr(A)m^3 \end{pmatrix}$$

Or, en développant le terme de droite, on obtient le système suivant

$$\begin{cases} \mathbf{\Delta}^2_{cone,x}(m) = \frac{1}{m^2} P_1(m) \\ \mathbf{\Delta}^2_{cone,y}(m) = \frac{1}{m^2} P_2(m) \\ \mathbf{\Delta}^2_{cone,z}(m) = \frac{1}{m^2} P_3(m) \end{cases}$$

où

$$\begin{cases} P_1(x) = k_1 \left(x - \frac{\lambda_{B',1}}{\lambda_{A,simple}} d\right) \left(x - \frac{\lambda_{B',1}}{\lambda_{A,double}} d\right)^2 \\ P_2(x) = k_2 \left(x - \frac{\lambda_{B',2}}{\lambda_{A,simple}} d\right) \left(x - \frac{\lambda_{B',2}}{\lambda_{A,double}} d\right)^2 \\ P_3(x) = k_3 \left(x - \frac{\lambda_{B',3}}{\lambda_{A,simple}} d\right) \left(x - \frac{\lambda_{B',3}}{\lambda_{A,double}} d\right)^2 \end{cases}$$

avec

$$\begin{cases} k_1 = \frac{-\lambda_{B',2}\lambda_{B',3}}{\lambda_{B',1}(\lambda_{B',1}-\lambda_{B',2})(\lambda_{B',1}-\lambda_{B',3})d} \\ k_2 = \frac{-\lambda_{B',1}\lambda_{B',3}}{\lambda_{B',2}(\lambda_{B',2}-\lambda_{B',1})(\lambda_{B',2}-\lambda_{B',3})d} \\ k_3 = \frac{-\lambda_{B',1}\lambda_{B',2}}{\lambda_{B',3}(\lambda_{B',3}-\lambda_{B',1})(\lambda_{B',3}-\lambda_{B',2})d} \end{cases}$$

Le lieu des solutions scalaires m est le sous-ensemble de $\mathbb{R}$ sur lequel $P_1(x)$, $P_2(x)$ et $P_3(x)$ sont simultanément positifs. L'étude des variations des $P_i(x)$ va démontrer que, dans tous les cas, ce lieu se réduit à un seul scalaire.

Pour réaliser cette étude de signes, nous choisissons $\lambda_{B',1}$ et $\lambda_{B',2}$ tels que $|\lambda_{B',1}| > |\lambda_{B',2}|$, puis identifions 4 configurations possibles en fonction des valeurs propres de A et B' (voir tableau D.1).

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	#1	#2
$\lambda_{A,simple} > \lambda_{A,double}$	#3	#4

TABLE D.1 – Configurations possibles du problème.

Le tableau D.2 montre le signe que prend d dans ces différentes configurations.

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	+	-
$\lambda_{A,simple} > \lambda_{A,double}$	+	-

TABLE D.2 – Signe de d.

Puis les tableaux D.3, D.4 et D.5 donnent le signe de k_1, k_2, k_3, en explicitant le signe du numérateur, de celui de la partie du dénominateur impliquant les valeurs propres de B', et de celui de d.

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	$\frac{+}{-+} = -$	$\frac{+}{+-} = -$
$\lambda_{A,simple} > \lambda_{A,double}$	$\frac{+}{-+} = -$	$\frac{+}{+-} = -$

TABLE D.3 – Signe de k_1.

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	$\frac{+}{++} = +$	$\frac{+}{--} = +$
$\lambda_{A,simple} > \lambda_{A,double}$	$\frac{+}{++} = +$	$\frac{+}{--} = +$

TABLE D.4 – Signe de k_2.

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	$\frac{-}{++} = -$	$\frac{-}{--} = -$
$\lambda_{A,simple} > \lambda_{A,double}$	$\frac{-}{++} = -$	$\frac{-}{--} = -$

TABLE D.5 – Signe de k_3.

Appelons S_i la racine simple de $P_i(x)$, et D_i sa racine double, de telle sorte que

$$S_i = \frac{\lambda_{B',i}}{\lambda_{A,simple}} d \quad \text{et} \quad D_i = \frac{\lambda_{B',i}}{\lambda_{A,double}} d.$$

Les tableaux D.6 et D.7 montrent l'ordre de ces racines.

On peut ainsi constater que les racines de $P_1(x)$ et $P_2(x)$ sont négatives, tandis que celles de $P_3(x)$ sont positives. Comme $k_3 < 0$ dans toutes les configurations possibles, on a

$$\forall x < 0,\ P_3(x) > 0.$$

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	$S_i < D_i < 0$	$S_i < D_i < 0$
$\lambda_{A,simple} > \lambda_{A,double}$	$D_i < S_i < 0$	$D_i < S_i < 0$

TABLE D.6 – Ordre des racines de $P_i(x)$ ($i = 1$ ou $i = 2$).

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	$0 < D_3 < S_3$	$0 < D_3 < S_3$
$\lambda_{A,simple} > \lambda_{A,double}$	$0 < S_3 < D_3$	$0 < S_3 < D_3$

TABLE D.7 – Ordre des racines de $P_3(x)$.

Donc nous avons seulement à nous intéresser aux signes de $P_1(x)$ et $P_2(x)$ pour déterminer le lieu des valeurs possibles de m. Le tableau D.8 compare les racines de ces deux polynômes.

	$\lambda_{B',1} < \lambda_{B',2} < 0$ et $\lambda_{B',3} > 0$	$\lambda_{B',1} > \lambda_{B',2} > 0$ et $\lambda_{B',3} < 0$
$\lambda_{A,simple} < \lambda_{A,double}$	$S_1 < S_2$ $D_1 < D_2$	$S_1 < S_2$ $D_1 < D_2$
$\lambda_{A,simple} > \lambda_{A,double}$	$S_1 < S_2$ $D_1 < D_2$	$S_1 < S_2$ $D_1 < D_2$

TABLE D.8 – Comparaison des racines de $P_1(x)$ et $P_2(x)$.

Dans la suite nous montrons que, pour chacune des 4 configurations possibles, deux sous-cas apparaissent : l'un conduit à une unique solution pour m, et l'autre conduit à une absence de solution, donc est impossible.

Configuration #1 : Il est possible de distinguer deux cas : $S_1 < S_2 < D_1 < D_2$ et $S_1 < D_1 < S_2 < D_2$. Le second cas n'aboutit à aucune solution pour m (les trois polynômes ne sont jamais simultanément positifs). Pour le premier cas, les signes des trois pôlynomes sont présentés dans le Tableau D.9.

x	$-\infty$		S_1	S_2	D_1	D_2		0
$P_1(x)$		+	0	-	0	-		
$P_2(x)$		-		0	+	0	+	
$P_3(x)$					+			

TABLE D.9 – Signe des $P_i(x)$ dans la Configuration #1.

Dans la configuration #1, la seule valeur possible pour m est donc D_1.

Configuration #2 : Cette configuration est identique à la configuration #1.

Configuration #3 : Il est possible de distinguer deux cas : $D_1 < D_2 < S_1 < S_2$ et $D_1 < S_1 < D_2 < S_2$. Le second cas n'aboutit à aucune solution pour m (les trois polynômes ne sont jamais simultanément positifs). Pour le premier cas, les signes des trois pôlynomes sont présentés dans le Tableau D.10.

Dans la configuration #3, la seule valeur possible pour m est donc D_2.

Configuration #4 : Cette configuration est identique à la configuration #3.

x	$-\infty$		D_1	D_2	S_1	S_2		0
$P_1(x)$		$+$	0	$+$	0	-		
$P_2(x)$		-		0	-	0	$+$	
$P_3(x)$				$+$				

TABLE D.10 – Signe des $P_i(x)$ dans la Configuration #5.

Finalement, il est possible de distinguer 2 configurations, selon la forme de l'ellipsoïde. Si $\lambda_{A,simple} < \lambda_{A,double}$

$$m = D_1 = \frac{\lambda_{B',1}}{\lambda_{A,double}} d$$

Si $\lambda_{A,simple} > \lambda_{A,double}$

$$m = D_2 = \frac{\lambda_{B',2}}{\lambda_{A,double}} d$$

Puis $\sigma = dm^2$ donne la valeur correspondante de σ.

Bibliographie

[AGT+16] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad : CNN architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5297–5307, 2016.

[AGT+18] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad : CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6) :1437–1451, 2018.

[ANT08] H. Avron, E. Ng, and S. Toledo. A generalized courant-fischer minimax theorem. `https://escholarship.org/uc/item/4gb4t762`, August 2008.

[BADP17] Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J. Pappas. Probabilistic data association for semantic SLAM. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 1722–1729, 2017.

[BBCS12] Sid Ying-Ze Bao, Mohit Bagra, Yu-Wei Chao, and Silvio Savarese. Semantic structure from motion with points, regions, and objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2703–2710, 2012.

[BFG05] Herbert Bay, Vittorio Ferrari, and Luc Van Gool. Wide-baseline stereo matching with line segments. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 329–336, 2005.

[BGV92] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*, pages 144–152, 1992.

[BS11] Sid Ying-Ze Bao and Silvio Savarese. Semantic structure from motion. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 2025–2032, 2011.

[BTG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF : speeded up robust features. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pages 404–417, 2006.

[CCB+17] Silvia Cascianelli, Gabriele Costante, Enrico Bellocchio, Paolo Valigi, Mario Luca Fravolini, and Thomas A. Ciarfuglia. Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features. *Robotics and Autonomous Systems*, 92 :53–65, 2017.

[CDF+04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF : binary robust independent elementary features. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 778–792, 2010.

[CRD16] Marco Crocco, Cosimo Rubino, and Alessio Del Bue. Structure from motion with objects. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4141–4149, 2016.

[CRV+15] Alberto Crivellaro, Mahdi Rad, Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. A novel representation of parts for accurate 3d object detection and tracking in monocular images. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4391–4399, 2015.

[CRV+18] Alberto Crivellaro, Mahdi Rad, Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. Robust 3d object tracking from monocular images using stable parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6) :1465–1479, 2018.

[CS15] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. *International Journal of Computer Vision*, 112(2) :239–254, 2015.

[Ebe99] David Eberly. Perspective projection of an ellipsoid. `https://www.geometrictools.com/`, March 1999. Updated version : November 12, 2013.

[Ebe07] David Eberly. Reconstructing an ellipsoid from its perspective projection onto a plane. `https://www.geometrictools.com/`, May 2007. Updated version : March 1, 2008.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395, 1981.

[FWH10] Bin Fan, Fuchao Wu, and Zhanyi Hu. Line matching leveraged by point correspondences. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 390–397, 2010.

[FWH12] Bin Fan, Fuchao Wu, and Zhanyi Hu. Robust line matching through line-point invariants. *Pattern Recognition*, 45(2) :794–805, 2012.

[GBRD17] Paul Gay, Vaibhav Bansal, Cosimo Rubino, and Alessio Del Bue. Probabilistic structure from motion with objects (psfmo). In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3094–3103, 2017.

[Gir15] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1440–1448, 2015.

[GOSP13] Petr Gronát, Guillaume Obozinski, Josef Sivic, and Tomás Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 907–914, 2013.

[GSOP16] Petr Gronát, Josef Sivic, Guillaume Obozinski, and Tomás Pajdla. Learning and calibrating per-location classifiers for visual place recognition. *International Journal of Computer Vision*, 118(3) :319–336, 2016.

[GVL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[HCH10] Edward Hsiao, Alvaro Collet, and Martial Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 2653–2660, 2010.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017.

[HHO+17] Tomas Hodan, Pavel Haluza, Stepán Obdrzálek, Jiri Matas, Manolis I. A. Lourakis, and Xenophon Zabulis. T-LESS : an RGB-D dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Santa Rosa, CA, USA, March 24-31, 2017*, pages 880–888, 2017.

[HLI+12] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary R. Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision - ACCV 2012 - 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I*, pages 548–562, 2012.

[HR11] Joel A. Hesch and Stergios I. Roumeliotis. A direct least-squares (DLS) method for pnp. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 383–390, 2011.

[HZ04] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision (Second Edition)*. Cambridge University Press, 2004.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[JDS09] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1169–1176, 2009.

[JDSP10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3304–3311, 2010.

[JGF+16] Qi Jia, Xinkai Gao, Xin Fan, Zhongxuan Luo, Haojie Li, and Ziyao Chen. Novel coplanar line-points invariants for robust line matching across views. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 599–611, 2016.

[JH98] Tommi S. Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11, NIPS*

Conference, Denver, Colorado, USA, November 30 - December 5, 1998, pages 487–493, 1998.

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, pages 675–678, 2014.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 : 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[KSP10] Jan Knopp, Josef Sivic, and Tomás Pajdla. Avoiding confusing features in place recognition. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, pages 748–761, 2010.

[KWK09] Taemin Kim, Jihwan Woo, and In-So Kweon. Probabilistic matching of lines for their homography. In *Proceedings of the International Conference on Image Processing, ICIP 2009, 7-10 November 2009, Cairo, Egypt*, pages 3489–3492, 2009.

[KZ02] Jana Kosecká and Wei Zhang. Video compass. In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part IV*, pages 476–490, 2002.

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD : single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37, 2016.

[LMD17] Jimmy Li, David Meger, and Gregory Dudek. Context-coherent scenes of objects for camera pose estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 655–660, 2017.

[LMD19] Jimmy Li, David Meger, and Gregory Dudek. Semantic mapping for view-invariant relocalization. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 7108–7115, 2019.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[LSFP15] Juan López, Roi Santos, Xosé R. Fernández-Vidal, and Xose Manuel Pardo. Two-view line matching algorithm based on context and appearance in low-textured images. *Pattern Recognition*, 48(7) :2164–2184, 2015.

[LSLL12] Haifeng Li, Dezhen Song, Yan Lu, and Jingtai Liu. A two-view based multilayer feature graph for robot navigation. In *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pages 3580–3587, 2012.

[LSN+16] Stephanie M. Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David D. Cox, Peter I. Corke, and Michael J. Milford. Visual place recognition : A survey. *IEEE Transactions on Robotics*, 32(1) :1–19, 2016.

[LXMD18] Jimmy Li, Zhaoqi Xu, David Meger, and Gregory Dudek. Semantic scene models for visual localization under large viewpoint changes. In *15th Conference on Computer and Robot Vision, CRV 2018, Toronto, ON, Canada, May 8-10, 2018*, pages 174–181, 2018.

[MCCO07] F.L. Markley, Y. Cheng, J.L. Crassidis, and Y. Oshman. Averaging quaternions. *AIAA Journal of Guidance, Control, and Dynamics*, 30(4) :1193–1196, July-Aug 2007.

[MMM12] Lionel Moisan, Pierre Moulon, and Pascal Monasse. Automatic homographic registration of a pair of images, with A contrario elimination of outliers. *IPOL Journal*, 2 :56–73, 2012.

[MMM16] Lionel Moisan, Pierre Moulon, and Pascal Monasse. Fundamental matrix of a stereo pair, with A contrario elimination of outliers. *IPOL Journal*, 6 :89–113, 2016.

[Moa02] M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1) :1–16, 2002.

[MUS16] Éric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality : A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12) :2633–2651, 2016.

[NMS19] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam : Dual quadrics from object detections as landmarks in object-oriented SLAM. *IEEE Robotics and Automation Letters*, 4(1) :1–8, 2019.

[NYD15] Joe Yue-Hei Ng, Fan Yang, and Larry S. Davis. Exploiting local features from deep networks for image retrieval. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2015, Boston, MA, USA, June 7-12, 2015*, pages 53–61, 2015.

[ORL18] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, pages 125–141, 2018.

[OT01] Aude Oliva and Antonio Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3) :145–175, 2001.

[PIL19] Giorgia Pitteri, Slobodan Ilic, and Vincent Lepetit. Cornet : Generic 3d corners for 6d pose estimation of new objects without retraining, 2019.

[PSM10] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 143–156, 2010.

[PZC+17] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G. Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 2011–2018, 2017.

[RCD18] Cosimo Rubino, Marco Crocco, and Alessio Del Bue. 3d object localisation from multi-view image detections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6) :1281–1294, 2018.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pages 430–443, 2006.

[RDGF16] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016.

[RF18] Joseph Redmon and Ali Farhadi. Yolov3 : An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[RHGS15] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN : towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28 : Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB : an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2564–2571, 2011.

[SEE+12] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 573–580, 2012.

[SFB16] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A simple and effective method to detect orthogonal vanishing points in uncalibrated images of man-made environments. In *Eurographics 2016 - Short Papers, Lisbon, Portugal, May 9-13, 2016*, pages 33–36, 2016.

[SFB18] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A-contrario horizon-first vanishing point detection using second-order grouping laws. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 323–338, 2018.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9, 2015.

[SML13] Elena Stumm, Christopher Mei, and Simon Lacroix. Probabilistic place recognition with covisibility maps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 4158–4163, 2013.

[SML+16a] Elena Stumm, Christopher Mei, Simon Lacroix, Juan I. Nieto, Marco Hutter, and Roland Siegwart. Robust visual place recognition with graph kernels. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4535–4544, 2016.

[SML16b] Elena S. Stumm, Christopher Mei, and Simon Lacroix. Building location models for visual place recognition. *The International Journal of Robotics Research*, 35(4) :334–356, 2016.

[SMLC15] Elena Stumm, Christopher Mei, Simon Lacroix, and Margarita Chli. Location graphs for visual place recognition. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 5475–5480, 2015.

[SMM16a] Yohann Salaün, Renaud Marlet, and Pascal Monasse. The multiscale line segment detector. In *Reproducible Research in Pattern Recognition - First International Workshop, RRPR@ICPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers*, pages 167–178, 2016.

[SMM16b] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Robust and accurate line- and/or point-based pose estimation without manhattan assumptions. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, pages 801–818, 2016.

[SMT+18] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomás Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8601–8610, 2018.

[SSJ+15] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks : Viewpoint-robust, condition-robust, training-free. In *Robotics : Science and Systems XI, Sapienza University of Rome, Rome, Italy, July 13-17, 2015*, 2015.

[SSS08] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2) :189–210, 2008.

[Sut74] I. E. Sutherland. Three-dimensional data input by tablet. *Proceedings of the IEEE*, 62(4) :453–461, April 1974.

[SZ03] Josef Sivic and Andrew Zisserman. Video google : A text retrieval approach to object matching in videos. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1470–1477, 2003.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[TSF18] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 292–301, 2018.

[TSH+18] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, pages 391–408, 2018.

[TSOP15] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomás Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11) :2346–2359, 2015.

[TSPO13] Akihiko Torii, Josef Sivic, Tomás Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 883–890, 2013.

[vGJMR12] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD : a line segment detector. *IPOL Journal*, 2 :35–55, 2012.

[WNY09] Lu Wang, Ulrich Neumann, and Suya You. Wide-baseline image matching using line signatures. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 1311–1318, 2009.

[WP10] David S. Wokes and Phil L. Palmer. Perspective reconstruction of a spheroid from an image plane ellipse. *International Journal of Computer Vision*, 90(3) :369–379, 2010.

[Wyl08] C R Wylie. *Introduction to projective geometry*. Dover, New York, NY, 2008.

[YTLF16] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT : learned invariant feature transform. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 467–483, 2016.

[ZD14] C. Lawrence Zitnick and Piotr Dollár. Edge boxes : Locating object proposals from edges. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 391–405, 2014.

[ZK13] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *J. Visual Communication and Image Representation*, 24(7) :794–805, 2013.

[ZWA+16] Amir Roshan Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3d representation via pose estimation and matching. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, pages 535–553, 2016.

[ZWJ16] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. *CoRR*, abs/1608.05684, 2016.

[ZYT18] Liang Zheng, Yi Yang, and Qi Tian. SIFT meets CNN : A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5) :1224–1244, 2018.

Résumé

La Réalité Augmentée peut être définie comme la superposition de la réalité et d'éléments (sons, images 2D, 3D, vidéos, etc.) calculés par un système informatique en temps réel. En pratique, ce terme désigne l'ajout d'éléments visuels, soit dans le champ de vision d'un observateur par l'intermédiaire de lunettes spécifiques (ex. : Microsoft Hololens, Magic Leap One), soit sur un écran à travers lequel l'observateur voit la réalité (généralement un smartphone ou une tablette). Au cours de ce travail de recherche, nous nous sommes intéressés au déploiement de la Réalité Augmentée dans un contexte industriel, et plus particulièrement aux défis que des environnements industriels de grande taille (usines, centrales, navires) représentent en termes d'analyse et de traitement des images. Nous avons notamment étudié le recours aux objets d'intérêt présents dans la scène pour reconnaître le lieu dans lequel se trouve l'observateur puis calculer sa position précise par rapport à l'environnement. Les applications visées sont, entre autres, l'aide à la fabrication, l'aide à la maintenance, la documentation et la formation.

Après avoir proposé une définition fonctionnelle du concept de lieu en environnement industriel, comme zone d'interaction autour d'un objet d'intérêt, nous avons abordé la reconnaissance de lieux comme une tâche de récupération d'images dans laquelle la similarité entre l'image inconnue et les images de référence est mesurée en deux étapes. La validité des images présentant les plus grandes similarités avec l'image inconnue est ensuite évaluée par estimation de la géométrie épipolaire liant l'image inconnue et chacune des images récupérées. La mesure de similarité et l'estimation de la géométrie sont guidées par le calcul de correspondances de niveau objet entre régions d'intérêt des deux images.

Pour calculer la pose de la caméra, nous avons ensuite tiré profit des objets d'intérêt présents dans la scène, en utilisant pour cela une modélisation de ces derniers sous forme d'ellipsoïdes, les projections des objets dans l'image étant modélisées sous forme d'ellipses. Nos contributions au problème d'estimation de pose de caméra à partir de correspondances ellipse - ellipsoïde sont d'ordre à la fois théorique et pratique. Nous avons notamment montré qu'il existe une paramétrisation des solutions du problème à un seul ellipsoïde, et, par ailleurs, que le problème d'estimation de pose de caméra peut être réduit à un problème d'estimation de son orientation seulement. Nous avons également proposé une manière robuste de traiter les multiples appariements possibles entre les objets détectés dans l'image et les objets présents dans le modèle 3D de la scène.

Mots-clés: Localisation, Reconnaissance d'objets, Vision par ordinateur.

Abstract

Augmented Reality can be defined as the superimposition of reality and elements (sounds, 2D and 3D images, videos, etc.) calculated in real time by a computer system. In practice, this term refers to the addition of visual elements, either in the field of view of an observer through specific glasses (e.g. Microsoft Hololens, Magic Leap One), or on a screen through which the observer sees reality (usually a smartphone or tablet). During this research work, we were interested in the deployment of Augmented Reality in an industrial context, and more particularly in the challenges that large industrial environments (factories, plants, ships) represent in terms

of image analysis and processing. In particular, we investigated the use of objects of interest present in the scene to recognize the place of the observer and then calculate his precise position with respect to the environment. Applications include manufacturing assistance, maintenance assistance, documentation and training.

After proposing a functional definition of the concept of place in an industrial environment, as a zone of interaction around an object of interest, we approached place recognition as an image retrieval task in which the similarity between the unknown image and the reference images is measured in two steps. The validity of the images with the greatest similarity to the unknown image is then assessed by epipolar geometry estimation between the unknown image and each of the retrieved images. The similarity measurement and geometry estimation are guided by the calculation of object-level correspondences between regions of interest of the two images.

To calculate the camera pose, we then took advantage of the objects of interest present in the scene, using a modeling of the latter in the form of ellipsoids, the projections of the objects in the image being modeled as ellipses. Our contributions to the problem of estimating camera pose from ellipse - ellipsoid correspondences are both theoretical and practical. In particular, we have shown that there is a parametrization of the solutions to the one-ellipsoid problem, and, moreover, that the camera pose estimation problem can be reduced to an orientation estimation problem only. We have also proposed a robust way to handle the multiple possible matches between the objects detected in the image and the objects present in the 3D scene model.

Keywords: Localization, Object recognition, Computer vision.