



HAL
open science

Etude de caméras sphériques : du traitement des images aux applications en robotique

Cédric Demonceaux

► **To cite this version:**

Cédric Demonceaux. Etude de caméras sphériques : du traitement des images aux applications en robotique. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université de Bourgogne, 2012. tel-00862979

HAL Id: tel-00862979

<https://theses.hal.science/tel-00862979>

Submitted on 18 Sep 2013

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

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

Synthèse des travaux scientifiques présentée par

Cédric Demonceaux

pour obtenir le diplôme d'

Habilitation à Diriger des Recherches

Discipline Sciences et Techniques

Titre :

Etude de caméras sphériques : du traitement des images aux applications en robotique

Soutenue le 27 novembre 2012 devant le jury composé de :

M. Ryad Benosman, Maître de Conférences HDR, Université Pierre et Marie Curie/Inst. de la Vision
M. David Fofi, Professeur, Université de Bourgogne/Le2i
M. Simon Lacroix, Directeur de Recherche, LAAS CNRS Toulouse (rapporteur)
M. Eric Marchand, Professeur, Université de Rennes 1/IRISA (rapporteur)
M. Youcef Mezouar, Professeur, Institut Français de Mécanique Avancée/Institut Pascal
M. Patrick Rives, Directeur de Recherche, INRIA Sophia Antipolis (rapporteur)
M. Pascal Vasseur, Professeur, Université de Rouen/LITIS

Avant-Propos

La vision omnidirectionnelle permet de percevoir l'environnement sur 360° . C'est un atout considérable pour un robot mobile puisque grâce à cette particularité, il peut tirer parti d'une information globale de la scène à tout moment. Les premières recherches allant dans ce sens remontent en 1990 avec les travaux de Yasushi Yagi [YK90] qui a proposé d'utiliser un dispositif nommé COPIS en associant une caméra perspective et un miroir conique pour obtenir une vue panoramique de la scène. En 1995, Mouaddib et Pégard [MP95] présentent un procédé similaire pour localiser un robot dans son environnement. Bien que ce système possède des avantages pour la localisation, il se révèle peu adéquat pour reconstruire l'environnement 3D. D'autres systèmes catadioptriques (caméra+miroir) ont par la suite été employés [Yag99]. En 1997, Nayar [Nay] en a fait une analyse théorique et a démontré que pour obtenir une vision omnidirectionnelle à l'aide d'une caméra et un miroir, il n'existait que deux configurations possibles. L'une consiste à associer une caméra orthographique avec un miroir parabolique (caméra paracatadioptrique), l'autre associe une caméra perspective avec un miroir hyperbolique dont l'un des foyers correspond au centre optique de la caméra. Ces deux types de capteurs ont connu de nombreux développements puisque la propriété d'un point de vue unique leur confère une géométrie projective semblable aux caméras perspectives classiques. Pour ces caméras, les méthodes d'estimation du déplacement développées pour les caméras perspectives deviennent dès lors applicables dès que nous travaillons sur les plans projectifs dont le centre de projection est confondu avec le foyer du miroir [SPH98]. C'est à cette période que l'on voit apparaître les premières méthodes de reconstruction 3D d'environnement à partir de caméra catadioptrique centrale [DS02, Stu00]. En 2001, Geyer et Daniilidis [GD01] ainsi que Barreto et Araujo [BA01] ont démontré que les caméras à projection centrale (perspective, catadioptrique à point de vue unique) étaient en fait équivalentes à une image sphérique dans laquelle la projection s'effectuait en son centre. Cette propriété importante a permis de développer les premières méthodes de traitement des images adaptées aux images catadioptriques [DMB02]. Plus récemment, Ying et Hu [YH04] ainsi que Courbon et al. [CMEM07] ont démontré que l'on pouvait également représenter certaines caméras fish-eye par des images sphériques. Ainsi, la sphère nous fournit un cadre générique pour l'étude de plusieurs types de caméras telles que les caméras perspectives, catadioptriques à point de vue unique, fish-eyes, LadyBug.

Ce manuscrit a pour objectif de montrer que la modélisation sphérique offre un outil performant pour ces caméras. Il synthétise une grande partie de mes travaux menés depuis mon recrutement en tant que Maître de Conférences en septembre 2005. Ces travaux ont débuté dans l'équipe Perception en Robotique du laboratoire Modélisation, Information et Systèmes (M.I.S

EA 4290) de l'Université de Picardie Jules Verne puis continuent, depuis septembre 2010, dans le Département Vision du Laboratoire Electronique, Informatique et Image (Le2i, UMR CNRS 6306). Durant ma thèse, je me suis intéressé à l'analyse du mouvement dans des séquences d'images perspectives. Suite à ma nomination en 2005, j'ai réorienté mes travaux sur l'étude des caméras catadioptriques qui constitue le thème phare de l'Equipe Perception en Robotique. Dans un premier temps, je me suis intéressé à l'adaptation de méthodes usuelles développées dans le cadre de l'imagerie perspective pour des images catadioptriques [DV06]. Dans ces travaux, j'ai travaillé sur l'image plane catadioptrique. Par la suite, mes recherches m'ont amené à travailler sur la sphère pour déterminer des algorithmes relativement génériques utilisables pour des caméras centrales ou quasi-centrales..

Ce mémoire est structuré en quatre chapitres qui seront suivis d'un bilan des contributions et des perspectives découlant de ce travail. Le premier chapitre détaillera mon *cursus* universitaire et professionnel puis décrira l'ensemble de mes activités depuis ma nomination en tant que Maître de Conférences en septembre 2005. Le deuxième chapitre a pour objectif de rappeler brièvement l'équivalence entre les différents types de caméras considérés dans ce manuscrit et les images sphériques. Les trois autres chapitres aborderont mes travaux de recherche allant du traitement des images sphériques bas niveaux aux problèmes de plus haut niveau comme suit :

- le chapitre 2 sera consacré aux traitements des images grand angle. Il sera organisé en deux parties. Dans un premier temps, nous verrons, qu'à cause de fortes distorsions dans ce type d'images, les méthodes usuelles de filtrage d'images sont inappropriées. Puis nous montrerons comment la sphère d'équivalence peut nous aider à traiter ces images sans modifier les filtres utilisés en imagerie perspective. Dans un second temps, nous proposerons une méthode d'estimation du flot optique pour les images sphériques. Cette deuxième partie a été réalisée dans le cadre de la thèse d'Amina Radgui (bourse d'excellence marocaine) co-dirigée par El Mustapha Mouaddib (Professeur à l'Université de Picardie Jules Verne), Mohammed Rziza (Professeur Habilité de l'Université de Mohammed V, Rabat) et Driss Aboutajdine (Professeur de l'Enseignement Supérieur de l'Université de Mohammed V, Rabat),
- le chapitre 3 a pour objectif de montrer que les particularités des images sphériques permettent de définir de nouvelles méthodes d'estimation du mouvement pour des caméras paramétriques. C'est ainsi que nous verrons différentes approches pour estimer la rotation relative entre deux vues sphériques. Nous montrerons que deux points de la scène en correspondance dans deux images consécutives suffisent pour estimer la translation dès lors que la rotation est connue entre ces deux vues. Par la suite, nous verrons comment profiter de l'extraction de droites et des points de la scène pour développer une méthode linéaire d'estimation de la translation. Enfin, un ajustement de faisceaux sur la sphère sera réalisé pour améliorer l'estimation de la pose de la caméra sphérique et la reconstruction 3D. Ces travaux ont été effectués durant la thèse de Jean-Charles Bazin (KAIST, Corée du Sud) dirigée par Pascal Vasseur (Professeur à l'Université de Rouen) et In So Kweon (Professeur au KAIST, Corée du Sud) pour laquelle j'ai collaboré ainsi que la thèse de Dieu Sang Ly (Bourse Ministère) co-encadrée avec Claude Pégard (Professeur à l'Université de Picardie Jules Verne),

- le chapitre 4 sera dédié aux applications en robotique aérienne. Nous montrerons une nouvelle fois comment la sphère d'équivalence permet de développer des algorithmes pour l'estimation d'attitude et d'altitude d'un UAV à partir de capteurs atypiques incluant une caméra fish-eye. Dans un premier temps, nous montrerons comment un système stéréoscopique composé d'une caméra fish-eye et d'une caméra perspective permet de bénéficier des particularités de chaque caméra lors des phases de décollage et d'atterrissage d'un drone (vue grand angle grâce à la fish-eye mais distordue et vue précise mais restrictive grâce à la perspective). Ces travaux ont été menés dans le cadre d'un projet de la région Picardie (ALTO) à partir de la thèse de Damien Eynard co-encadrée avec Pascal Vasseur et Vincent Frémont (Maître de Conférences à l'Université de Technologie de Compiègne). Dans une deuxième partie, nous proposerons un capteur original muni d'une caméra fish-eye et d'un laser circulaire. Nous verrons comment ce dispositif permet de calculer l'altitude et l'attitude d'un drone dans des conditions de faible luminosité. Ce travail a été réalisé dans la cadre de la thèse d'Ashutosh Natraj co-encadrée avec Pascal Vasseur et Peter Sturm (Directeur de Recherche, INRIA Grenoble).

Table des matières

1	Activités scientifiques, administratives et pédagogiques	9
1.1	Parcours universitaire et professionnel	9
1.1.1	Parcours universitaire	9
1.1.2	Parcours professionnel	10
1.2	Activités de recherche	10
1.3	Activités scientifiques	11
1.4	Activités d'enseignement et responsabilité pédagogique	12
1.5	Encadrements	13
1.5.1	Doctorants	13
1.5.2	Post-Doctorants	15
1.5.3	Master M2	16
1.6	Projets	17
1.7	Publications	18
2	Modèle sphérique unifié	25
2.1	Introduction	25
2.2	Caméra catadioptrique à point de vue unique	27
2.2.1	Caméra hypercatadioptrique	27
2.2.2	Caméra paracatadioptrique	28
2.3	Caméra fish-eye	29
2.4	Modèle sphérique unifié	30
3	Traitement des images grand angle	33
3.1	Filtrage d'images grand angle	34
3.1.1	Voisinage géodésique	34
3.1.2	Convolution	36
3.2	Détection de points d'intérêts et mise en correspondance	40
3.2.1	Détecteur de Harris	40
3.2.2	Mise en correspondance : ZNCC	42
3.3	Flot optique	44
3.3.1	Ondelettes sphériques	45
3.3.2	Estimation du flot optique sphérique	47
3.3.3	Résultats	47

3.4	Conclusion	54
4	Estimation du déplacement de caméras sphériques	57
4.1	Estimation de la rotation d'une caméra sphérique	58
4.1.1	Approche combinatoire	59
4.1.2	Cas des faisceaux orthogonaux	62
4.2	Estimation de la translation	67
4.2.1	A partir de points : Algorithme des 2 points	68
4.2.2	A partir de droites	70
4.2.3	Estimation de la translation à partir de points et droites	73
4.3	Reconstruction 3D et ajustement de faisceaux	74
4.3.1	Reconstruction de points	74
4.3.2	Reconstruction de droites	75
4.3.3	Ajustement de faisceaux	75
4.4	Résultats	78
4.4.1	Estimation de la translation par l'algorithme des 2 points	78
4.4.2	Estimation de la translation à partir de droites	78
4.5	Conclusion	87
5	Capteur sphérique pour la robotique aérienne	89
5.1	Introduction	90
5.2	Capteur stéréoscopique : Fish-eye/Perspectif	91
5.2.1	Estimation de l'altitude du drone	93
5.2.2	Estimation du déplacement du drone à l'aide d'un capteur stéréo hybride	93
5.2.3	Résultats	96
5.3	Capteur Fish-eye/Laser	99
5.3.1	Modélisation analytique	100
5.3.2	Resultats expérimentaux	103
5.4	Conclusion	106
	Bilan et perspectives	109

Chapitre 1

Activités scientifiques, administratives et pédagogiques

Ce premier chapitre présente mes activités scientifiques, administratives et pédagogiques. Dans un premier temps, j'énumérerai mon parcours universitaire et professionnel. Puis je présenterai mes axes de recherche ainsi que l'ensemble de mes activités depuis ma nomination en 2005 en tant que Maître de Conférences. Ce chapitre se terminera par la liste de mes publications.

1.1 Parcours universitaire et professionnel

1.1.1 Parcours universitaire

2004	Doctorat de l'Université de Picardie Jules Verne (UPJV) Titre : « Etude du mouvement dans des séquences d'images par analyse d'ondelettes et modélisation markovienne hiérarchique. Application à la détection d'obstacles en milieu routier. » Qualification : 26 ^{ème} et 61 ^{ème} section
2001	D.E.A. Analyse Appliquée, UPJV
2000	Maîtrise de Mathématiques Pures, UPJV
1999	Licence de Mathématiques, UPJV
1998	DEUG MIAS, MP* Lycée Louis Thuillier Amiens
1997	MPSI Lycée Louis Thuillier Amiens

1.1.2 Parcours professionnel

2010- ...	Chaire CNRS/Université de Bourgogne Le2i UMR 6306, Département Vision IUT du Creusot, Département GEII.
2005-2010	Maitre de Conférences à l'Université de Picardie Jules Verne, MIS EA 4290, Equipe Perception en Robotique UFR des Sciences.
2001-2004	Moniteur à l'UFR de Mathématiques et d'Informatique d'Amiens

1.2 Activités de recherche

Recruté en tant que Maître de Conférences en septembre 2005 dans l'équipe Perception en Robotique du M.I.S. E.A. 4290 de l'Université de Picardie Jules Verne, mes recherches avaient dans un premier temps pour thème la vision catadioptrique. A partir de 2005, je me suis notamment intéressé :

- aux traitements des images catadioptriques, je cherchais par exemple comment adapter certaines méthodes de traitement d'images « classiques » aux images catadioptriques,
- à l'utilisation de ce capteur pour des applications en robotique aérienne telles que l'estimation du roulis et du tangage à partir d'une caméra catadioptrique montée sur le robot volant,
- à l'odométrie visuelle basée vision catadioptrique.

Mes travaux sur le traitement d'images catadioptriques ont débuté avec l'adaptation des champs de Markov [R3]¹ et le calcul du flot optique [W5]. Les approches consistaient à reprendre pas à pas les hypothèses employées dans le cas des images perspectives pour les adapter aux images à fortes distorsions. Concernant l'estimation d'attitude d'un UAV, nous avons dans un premier temps appliqué les champs de Markov adaptés pour la segmentation (terre/ciel) [C4]. Puis, pour développer une méthode d'estimation d'attitude en milieu urbain [C7], nous avons proposé un nouvel algorithme d'estimation des droites de la scène sur des images catadioptriques [C8]. Cet algorithme de détection de droites, nous a par la suite permis de développer des méthodes d'estimation de la rotation d'une caméra catadioptrique [C13] puis d'estimer les six degrés de liberté liés à sa position [R4].

Tous ces travaux partaient du constat que les caméras catadioptriques avaient un fort intérêt pour la robotique mais que l'utilisation de ce type de caméras nécessitait de repenser les méthodes développées pour les caméras classiques. Par la suite, nous avons étendu ces travaux pour tout type de caméra à point de vue unique (perspective, fish-eye, catadioptrique, lady-bug...). Comme toutes ces caméras peuvent être considérées comme des caméras sphériques ([YH04]), nous avons concentré nos efforts sur l'étude des images sphériques [R6][R8].

En septembre 2010, j'ai rejoint le laboratoire Le2i dans le département Vision au sein du thème SINC (Systèmes d'Imagerie Non-Conventionnelle) sur une chaire mixte CNRS/ Université de Bourgogne. A partir de cette période, j'ai élargi mon domaine de recherche en vision sphérique sur un spectre plus large de système de caméras. Par exemple, j'étudie l'apport de

1. Les références ici se rapportent à la section 1.7 de ce chapitre.

systèmes stéréoscopiques atypiques pour la robotique aérienne comme par exemple une caméra fish-eye couplée avec une caméra perspective [R11] ou avec un laser [R12]. Plus récemment, nous étudions l'utilisation d'un capteur muni d'une caméra fish-eye et d'une PTZ pour de la surveillance aérienne dans le cadre d'une thèse co-financé par la DGA et le Conseil Régional de Bourgogne. Enfin, depuis janvier 2012, je participe à un projet ANR international avec la Corée du Sud pour lequel je suis le coordinateur local au sein du Le2i et dans lequel nous nous intéressons à l'étude d'un banc multi-caméras fish-eye asynchrones monté sur un véhicule.

1.3 Activités scientifiques

Jun 2013	Co-organisation avec David Fofi d'ORASIS 2013.
Nov. 2011	Invitation d'une semaine à l'université Mohammed V de Rabat par Pr. Driss Aboutajdine pour travailler sur la thèse de Fatima Zhara Benamar.
Nov. 2009	Séjour de trois semaines au KAIST pour travailler sur la thèse de Jean Charles Bazin.
Oct. 2009	Invitation d'une semaine à l'université Mohammed V de Rabat par Pr. Driss Aboutajdine pour travailler sur la thèse d'Amina Radgui.
Oct. 2008	Séjour de deux semaines au KAIST en Corée du Sud, invité par Pr. Kweon. A cette occasion, l'université coréenne a organisé un workshop sur la vision omnidirectionnelle pour les étudiants d'un consortium d'universités coréennes (ECNIR, Education Center for Network-based Intelligent Robotics) au cours duquel j'ai présenté un séminaire sur le traitement d'images omnidirectionnelles.
Avr. 2008	Invitation d'une semaine à l'université Mohammed V de Rabat par Pr. Driss Aboutajdine pour présenter mes travaux devant les étudiants du L.R.I.T.
Janv. 2008	Collaboration avec le Professeur W. Shen de l'Université de Sun Yat-Sen en Chine. Nous avons co-dirigé deux étudiants de M2 recherche sur des problématiques de sécurisation des aéroports chinois.

Depuis 2005, j'ai eu l'occasion d'être relecteur pour les journaux :

- IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),
- Machine Vision and Applications journal (MVA, Springer-Verlag),
- IEEE Transactions on Robotics (TRO),
- International Journal of Image and Graphics (IJIG),
- International Journal of Fuzzy Systems (IJFS),
- Journal of Electronic Imaging (FEI),
- Journal of Zhejiang University Science

ainsi que pour les conférences :

- European Conference on Computer Vision 2012 (ECCV 2012),
- Reconnaissance de Forme et Intelligence Artificielle (RFIA),
- IEEE Conference on Intelligent Transportation Systems 2006 (ITSC 2006, ITSC 2008)
- IEEE Conference on Robotics and Automation (ICRA'06, ICRA'08, ICRA'09, ICRA'10)
- IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09, IROS'10)
- IEEE International Conference on Quality Control by Artificial Vision (QCAV'07).

– IEEE International Conference on Image Processing (ICIP'09, ICIP'10, ICIP'11)

J'ai participé à trois jurys de thèses :

- 2010 **J-F Layerle**, "Conception d'un capteur d'observation simultanée de l'activité oculaire et de la scène routière en situation de conduite automobile".
Président : E. Marchand,
Rapporteurs : P. Martinet, F. Jurie,
Examineurs : P. Vasseur, C. Demonceaux,
Directeurs : E.M. Mouaddib, X. Savatier.
- 2010 **P.E. Leni**, "Nouvelles méthodes de traitement de signaux multidimensionnels par décomposition suivant le théorème de Superposition de Kolmogorov".
Président : A. Baskurt,
Rapporteurs : J. Chanussot, A. Baskurt,
Examineurs : B. Igel'nik, D.A. Sprecher, C. Demonceaux,
Directeurs : F. Truchetet, Y. Fougerolle.
- 2010 **G. Gallegos Garrido**, "Développement d'un capteur composite Vision/Laser à couplage serré pour le SLAM d'intérieur".
Président : J.P. Marmorat,
Rapporteurs : P. Martinet, E.M. Mouaddib,
Examineurs : A. Victorino, C. Demonceaux,
Directeur : P. Rives.

De plus, comme l'habilitation à diriger des recherches n'est pas nécessaire pour rapporter une thèse en Espagne, j'ai eu l'occasion de rapporter la thèse de M. Eval Bladimir Bacca Cortes de l'Université de Gérone : "Appearance-based mapping and localization using feature stability histograms for mobile robot navigation " (juin 2012).

1.4 Activités d'enseignement et responsabilité pédagogique

De 2005 à 2010, j'ai dispensé des cours au sein du Département EEA de l'UFR des Sciences de l'Université de Picardie Jules Verne. Pendant ces cinq années, j'étais responsable de l'apprentissage du Master EEAI. A ce titre, j'étais chargé de faire le lien entre les entreprises et la formation EEA, de démarcher des entreprises susceptibles d'être intéressées par nos formations, d'organiser les soutenances des apprentis.

Depuis 2010, j'interviens au département Génie Electrique et Informatique Industrielle de l'IUT du Creusot, dans le Master Computer Vision et le Master "VIBOT / Erasmus Mundus Masters in Vision and RoBOTics".

Mes principales activités en tant qu'enseignant se résument donc comme suit :

2005-2010 : IUP GEII Amiens (environ 200 UC), responsable apprentissage

– Techniques Mathématiques pour l'ingénieur, L3 (CM, TD, TP)

Complexes, calcul intégral, décomposition des fractions rationnelles en éléments simples, algèbre matricielle, équation différentielle.

- Vision et traitement des images, M1 (CM, TD)
Formation d'une image, extraction de caractéristiques dans l'image (contours, droite, texture...), segmentation d'images, morphologie mathématiques, détection de mouvement.
- Vision Avancée, M2 (CM, TD, TP)
Vision en lumière structurée, vision omnidirectionnelle.
- Statistiques descriptives, L3 Pro (TD)
Statistique simple, statistique double, série temporelle, concentration, indice.

2010 : Séminaire pédagogique dans le cadre du Module Image du LIRMM à Montpellier. Ce séminaire avait pour objectif de montrer les intérêts et les avancées de la recherche en vision omnidirectionnelle.

2010 - ... : IUT GEII Le Creusot

- Mathématiques, L1 (32 CM)
Complexes, équations différentielles, géométrie vectorielle, intégrales, transformée de Laplace, algèbre matricielle.

2010 - ... : Master Vibot / Master Computer Vision, Le Creusot (12 CM)

- Optical flow and Time to contact.

1.5 Encadrements

Depuis ma prise de fonction en 2005, j'ai eu l'occasion d'encadrer 6 thèses de doctorat (dont 3 soutenues), 2 post-doctorants et 7 stages de master recherche. Ces encadrements sont détaillés ci-dessous.

1.5.1 Doctorants

Trois thèses que je co-encadrais ont été soutenues :

- 2007-2010 **A. Radgui** (bourse d'excellence marocaine) "Estimation du flot optique dans les images omnidirectionnelles"
Co-direction : E.M. Mouaddib (MIS-UPJV), M. Rziza (LRIT-Mohammed V Agdal, Maroc), D. Aboutajdine (LRIT-Mohammed V Agdal, Maroc), soutenue le 13 décembre 2010, Université de Mohammed V Agdal, Maroc.
Jury : D. Aboutajdine (Faculté des Sciences de Rabat, président); R. Benslimane (Ecole Supérieur de Technologie de Fès, rapporteur); J.-M. Chassery (GIPSA-Lab, rapporteur); E. M. Mouaddib, M. Rziza, C. Demonceaux (co-encadrants).

- 2008-2011 **D. Eynard** (bourse Région Picardie) "Capteur de stéréovision hybride pour la navigation des drones"
Co-direction P. Vasseur (LITIS-Univ Rouen), V. Frémont (Heudiasyc-UTC), thèse soutenue le 7 novembre 2011, Université de Picardie Jules Verne.
Jury : C. Pégard (UPJV, Président) ; S. Lacroix (LAAS, rapporteur) ; T. Hamel (I2S, rapporteur) ; P. Fabiani (ONERA, examinateur) ; P. Vasseur, V. Frémont, C. Demonceaux (co-encadrants).
- 2008-2011 **S. Ly** (bourse ministère) "Structure from motion with hybrid cameras using point and line features"
Co-direction C. Pégard (MIS-UPJV), soutenue 24 octobre 2011, Université de Picardie Jules Verne
Jury : M. Mallem (IBISC, Président) ; P. Sturm (INRIA, rapporteur) ; Y. Mezouar (LASMEA, rapporteur) ; P. Vasseur (Litis, examinateur) ; C. Pégard, C. Demonceaux (co-encadrants).

La thèse d'Amina Radgui avait pour objectif de calculer le flot optique dans les images catadioptriques et sphériques. Pour ce faire, elle a développé deux méthodes originales. La première consistait en une adaptation de l'algorithme de Lucas-Kanade sur des images catadioptriques [W5]. La seconde, qui sera détaillée dans le chapitre 2, étudiait l'équation de contrainte de flot optique sur des images sphériques [R5]. A. Radgui est actuellement enseignant-chercheur à Institut National des Postes et Télécommunications au Maroc.

Damien Eynard s'intéressait à l'étude d'un capteur stéréoscopique hybride (Fish-eye/perspective) pour la navigation d'un drone d'un le cadre d'un projet de la région Picardie en collaboration avec l'Université de Technologie de Compiègne. Durant son doctorat, Damien Eynard a développé une méthode d'estimation de l'altitude du drone à l'aide de son système stéréoscopique sans mise en correspondance entre les deux images fortement hétérogènes. Puis il a proposé une méthode d'estimation du déplacement du drone à partir de la fusion d'informations entre la caméra fish-eye et la caméra perspective [R11]. Ses travaux seront détaillés au chapitre 4. Damien Eynard est actuellement en contrat post-doctoral au LITIS.

Dieu Sang Ly a repris le travail d'un ancien doctorant de l'équipe avec qui je travaillais : Jean-Charles Bazin. Jean-Charles Bazin avait développé une méthode pour estimer le déplacement d'une caméra catadioptrique. Alors que cette approche nécessitait la détection de droites de la scène pour estimer la rotation puis la détection de points pour la translation [R4]. L'objectif de Sang Ly était de s'affranchir de la détection des points pour estimer la translation en développant une méthode fondée sur les droites [C28]. Son travail sera détaillé au chapitre 3. Sang Ly est actuellement en post-doc au Le2i.

Je co-encadre actuellement trois thèses :

- 2009-2012 **A. Natraj** (MIS, Université de Picardie), bourse région Picardie :
« Estimation de pose d'un UAV par lumière structurée ».
Soutenance prévue fin 2012.
Co-direction avec P. Vasseur (LITIS, Université de Rouen), P. Sturm (INRIA Grenoble).

- 2010-2013 **F.Z. Benamar** (LRIT, Université de Mohammed V Agdal, Maroc), bourse co-tutelle programme STIC France-Maroc :
 « Estimation du temps à collision d'un robot mobile à l'aide d'une caméra omnidirectionnelle ».
 Soutenance prévue fin 2013.
 Co-direction avec D. Aboutajdine (LRIT, Rabat), S. Elfkihi (LRIT, Rabat), E.M. Mouaddib (MIS, Université de Picardie).
- 2011-2014 **F. Rameau** (Le2i), bourse DGA/région Bourgogne :
 « Surveillance aérienne à l'aide d'un système de vision hybride (Fish-eye+caméra PTZ) ».
 Soutenance prévue en octobre 2014.
 Co-direction avec D. Fofi (Le2i), D. Sidibé (Le2i)

Ashustosh Natraj s'intéresse à l'estimation de l'altitude et l'attitude d'un drone à partir d'un capteur composé d'un laser circulaire et d'une caméra sphérique [C33]. Ce capteur nous permet de calculer les paramètres de navigation du drone en cas de faible luminosité. Nous présenterons une partie de ses travaux dans le chapitre 4.

Fatima Zahra Benamar calcule le temps à collision d'un robot, c'est-à-dire le temps qu'il reste au robot avant d'entrer en contact avec l'obstacle à l'aide d'une caméra catadioptrique [C32]. Elle développe actuellement une méthode estimer le temps à collision et l'orientation de la surface de l'obstacle. Ses recherches font suite aux travaux d'Amina Radgui et permettent de continuer la collaboration avec nos collègues de l'Université de Mohammed V de Rabat.

François Rameau étudie quant à lui l'apport du capteur composé d'une caméra fish-eye et d'une caméra PTZ pour la surveillance aérienne dans le cadre d'une bourse DGA/région Bourgogne. Il s'agit d'utiliser une caméra grand angle pour surveiller l'espace survolé par le drone et d'analyser les objets suspects à l'aide de la caméra PTZ.

1.5.2 Post-Doctorants

- 11/06 - 07/07 **A. Mazin**, Détection de plans par vision omnidirectionnelle.
 10/06 - 08/07 **C. Simler**, Odométrie visuelle par vision omnidirectionnelle.

Alexander Mazin et Christophe Simler ont été recrutés dans le cadre du projet ANR CaViAR. A. Mazin devait détecter les zones planes dans une image catadioptrique à partir de l'extraction de droites parallèles. Christophe Simler avait pour objectif de comparer les méthodes d'odométrie visuelle classiques développées pour les caméras perspectives sur des images sphériques [C10].

1.5.3 Master M2

- 2012 **Giulia Meneghetti** "SFM using unsynchronized stereoscopic camera".
Co-direction D.Fofi (Le2i).
G. Meneghetti cherchait comment estimer le déplacement d'un banc stéréo non synchronisé. Ce travail rentrait dans le cadre du projet DrAACaR.
- 2012 **Alper Akcoltekin** "SFM using Infra-Red Camera".
Co-direction D.Fofi (Le2i).
A. Akcoltekin estimait le déplacement d'une caméra infra-rouge.
- 2011 **François Rameau** "Suivi d'objets dans des images omnidirectionnelles".
Co-direction D. Sidibé (Le2i), D. Fofi (Le2i)
F. Rameau a adapté les filtrages particulaire et le *mean-shift* pour des caméras sphériques
- 2008 **Huan Yue** "Détection de visages dans une séquence d'images".
Co-direction W. Shen (SIST-Univ de Sun Yat-Sen, Chine).
H. Yue comparait les différentes méthodes de détection de visages pour une application dans les aéroports chinois.
- 2008 **Qu Ning** "Traitement d'images pour la reconnaissance d'empreintes digitales".
Co-direction W. Shen (SIST-Univ de Sun Yat-Sen, Chine).
Q. Ning a étudié différentes méthodes de classification d'empreintes digitales pour vérifier qu'elles étaient les méthodes les plus efficaces pour les aéroports chinois.
- 2005 **David Aout** "Calcul d'attitude d'un UAV par vision omnidirectionnelle".
Co-direction P. Vasseur (LITIS-Univ Rouen).
D. Aout avait pour objectif d'améliorer et d'implémenter des algorithmes d'estimation d'attitude développés dans l'équipe.
- 2004 **Bading Bessi Kama** "Estimation du flot optique sur les images omnidirectionnelles".
Co-direction O. Goubet (LAMFA-UPJV).
Dans le cadre du projet CNRS MathStic, B. Bessi Kama a étudié les méthodes d'estimation du flot optique sur des images paracatadioptriques.

1.6 Projets

Durant ces années, j'ai participé et participe à 4 projets internationaux, 3 projets nationaux et 2 projets régionaux que je détaille ci-dessous² :

Projets Internationaux

- 2012 - 2014 Projet NRF-ANR Programme Blanc International, DrAACaR - Driver Assistance by Asynchronous Camera Ring.
Partenaires :
LITIS, Université de Rouen (P. Vasseur)
RCVLab, KAIST, Corée du Sud (I.S. Kweon)
USRG, KAIST, Corée du Sud (D. H. Shim)
Le2i, Université de Bourgogne (C. Demonceaux)
- 2012 - 2015 Projet FNR CORE (Luxembourg), FAVE - Fusion Approaches for Visual systems Enhancement.
Partenaires :
SnT, Université du Luxembourg (D. Aouada)
IEE S.A. (B. Mirbach)
Le2i, Université de Bourgogne (D. Fofi)
- 2008-2009 Projet STAR du programme Hubert Curien (Egide) NOVA : Aide au pilotage de robot volant par vision omnidirectionnelle.
Partenaires :
MIS, Université de Picardie Jules Verne (P. Vasseur)
RCVLab KAIST, Corée du Sud (I.S. Kweon)
Le2i, Université de Bourgogne (D. Fofi)
- 2007-2011 Projet VOLUBILIS du programme Hubert Curien (Egide) : Traitement et indexation de l'image à la vidéo : Application à la restauration et à la préservation du patrimoine culturel marocain.
Partenaires :
MIS, Université de Picardie Jules Verne (E.M. Mouaddib)
LRIT-Université Mohammed V Agdal, Rabat (D. Aboutajdine)
EST, Fes (R. Benslimane)
LMPA, Université du Littoral Côte d'Opale (A. Salam)
IMS-ENSEIRB, Bordeaux (M. Najim)

Projets Nationaux

- 2011 - 2012 Projet PEPS INSIS, MOSCA - MOuvement et Structure en Configurations Atypiques.
Partenaires :
ISIT, Université d'Auvergne Clermont 1 (A. Bartoli)
Le2i, Université de Bourgogne (D. Fofi)

2. Les noms entre parenthèses correspondent aux coordinateurs locaux.

- 2005 - 2008 Projet ANR Blanc CAVIAR : CAtdioptric VIsion for Aerial Robots.
Partenaires :
MIS, Université de Picardie Jules Verne (P. Vasseur)
INRIA Grenoble (P. Sturm)
INRIA Sophia-Antipolis (P. Rives)
LAAS CNRS, Toulouse (S. Lacroix)
Le2i, Université de Bourgogne (D. Fofi)
- 2003 Projet CNRS-MathStic : FLOT Vision omnidirectionnelle.
Partenaires :
LAMFA, Université de Picardie Jules Verne (O.Goubet)
MIS, Université de Picardie Jules Verne (D. Kachi)
- Projets Régionaux
- 2008 - 2011 Projet Régional Intermodalité, Energie, Transport : Décollage et atterrissage
automatiques de robots volants autonomes.
Partenaires :
Heudiasyc, Université de Technologie de Compiègne (R. Lozano)
MIS, Université de Picardie Jules Verne (C. Pégard)
- 2001- 2004 Projet Régional DiVA : Perception coopérative inter-véhicules pour la détection
d'imprévus dans un milieu routier.
Partenaires :
Heudiasyc, UTC (M. Shawky)
MIS, Université de Picardie Jules Verne (D. Kachi)

1.7 Publications

Articles de revues à comité de lecture

[R12] A. Natraj, S. Ly, D. Eynard, C. Demonceaux, P. Vasseur. Omnidirectional vision for UAV : applications to attitude, motion and altitude estimation for day and night conditions. In *Journal of Intelligent and Robotics Systems*, A paraître.

[R11] D. Eynard, P. Vasseur, C. Demonceaux, V. Frémont. Real Time UAV Altitude, Attitude and Motion Estimation from Hybrid Stereovision. In *Autonomous Robots*, Vol 33, Issue 1-2, Août 2012, pp.157-172.

[R10] J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. In *International Journal of Robotics Research*, Vol 31, Issue 1, Janvier 2012, pp. 63-81.

[R9] A.E.R. Shabayek, C. Demonceaux, O. Morel, D. Fofi. Vision Based UAV attitude Estimation : Progress and Insights. In *Journal of Intelligent and Robotics Systems*, Vol 65, Issue 1, Janvier 2012, pp. 295-308.

[R8] C. Demonceaux, P. Vasseur, Y. Fougerolle. Central Catadioptric Image Processing with Geodesic Metric. In *Image Vision Computing*, Vol.29, Issue 12, Novembre 2011, pp. 840-849.

[R7] F. Rameau, D. Sidibé, C. Demonceaux, D. Fofi. Visual Tracking with Omnidirectional Cameras : An Efficient Approach. *IET Electronic Letters*, Vol.47, Issue 21, Octobre 2011, pp. 1183- 1184.

[R6] A. Radgui, C. Demonceaux, E. Mouaddib, M. Rziza, D. Aboutajdine. Optical flow estimation from multichannel spherical image decomposition. *Computer Vision and Image Understanding - Elsevier*, Vol. 115, Issue 9, Septembre 2011, pp. 1263-1272.

[R5] A. Radgui, C. Demonceaux, E. Mouaddib, M. Rziza, D. Aboutajdine. Adapted approach for Omnidirectional Egomotion Estimation. *International Journal of Computer Vision and Image Processing*, Vol. 1, N°1, Janvier-Mars 2011, pp. 1-13.

[R4] J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon. Motion Estimation by Decoupling Rotation and Translation in Catadioptric Vision. *Computer Vision and Image Understanding - Elsevier*, Vol. 114, Issue 2, Février 2010, pp. 254-273.

[R3] C. Demonceaux, P. Vasseur. Markov Random Fields for Catadioptric Image Processing. *Pattern Recognition Letters - Elsevier*, VOL. 27, Issue 16, Décembre 2006, pp. 1957-1967.

[R2] C. Demonceaux, P. Vasseur. Champs de Markov pour le Traitement d'Images Catadioptriques. *Traitement du Signal - Numéro Spécial sur la Vision Omnidirectionnelle*, Vol. 22, N°5, 2005, pp.443-452.

[R1] C. Demonceaux, A. Potelle, D. Kachi-Akkouche. Obstacle detection in road scene based on motion analysis. *IEEE Transactions on Vehicular Technology - Special section on in-vehicle computer vision systems - Novembre 2004*, Volume 53, numéro 6, page 1649-1656.

Articles dans des congrès internationaux avec actes et comité de lecture

[C34] F. Rameau, A. Habed, C. Demonceaux, D. Sidibé, D. Fofi. Self-calibration of PTZ camera using new LMI constraints. In *11th Asian Conference on Computer Vision (ACCV 2012)*, Novembre 2012, Daejeon, Corée du Sud.

[C33] A. Natraj, P. Sturm, C. Demonceaux, P. Vasseur. A Geometrical Approach For Vision Based Attitude And Altitude Estimation For UAVs In Dark Environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'12)*, Octobre 2012, Vilamoura, Portugal.

[C32] F.Z. Benamar, C. Demonceaux, S. El Fkihi, E.M. Mouaddib, D. Aboutajdine. Time to Contact estimation on paracatadioptric cameras. In *International Conference on Pattern Recog-*

inition (ICPR 2012), Novembre 2012, Tsukuba, Japon.

[C31] A. Natraj, S. Ly, D. Eynard, C. Demonceaux, P. Vasseur. Omnidirectional vision for UAV : applications to attitude, motion and altitude estimation for day and night conditions. In International Conference on Unmanned Aircraft Systems (ICUAS 2012), Juin 2012, Philadelphie, PA, USA.

[C30] J.C. Bazin, Y.D. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I.S. Kweon, M. Pollefeys. Globally Optimal Line Clustering and Vanishing Point Estimation in Manhattan World. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'12), Juin 2012, Providence, Rhode Island, USA.

[C29] A. Natraj, C. Demonceaux, P. Vasseur, P. Sturm. Vision based attitude and altitude estimation for UAVs in dark environments. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'11), Septembre 2011, San Francisco, USA.

[C28] S. Ly, C. Demonceaux, P. Vasseur, C. Pégard. Motion estimation for hybrid cameras using point and line feature fusion. The Twelfth IAPR Conference on Machine Vision Applications (MVA 2011), Juin 2011, Nara, Japon.

[C27] D. Eynard, P. Vasseur, C. Demonceaux, V. Frémont. UAV Motion Estimation using Hybrid Stereoscopic Vision. The Twelfth IAPR Conference on Machine Vision Applications (MVA 2011), Juin 2011, Nara, Japon.

[C26] A.E.R. Shabayek, C. Demonceaux, O. Morel, D. Fofi, Vision Based UAV attitude Estimation : Progress and Insights. In International Conference on Unmanned Aircraft Systems (ICUAS 2011), Mai 2011, Denver, CO, USA.

[C25] T. Kim, Z. Moratto, A.V. Nefian, S. Ly, C. Demonceaux, D. Fofi. Robust Orbital Refinement of the Appolo Trajectory Data for the Ames Stereo Pipeline. In 42nd Lunar and Planetary Science Conference, Mars 2011, The Woodlands, Texas, USA.

[C24] J.C. Bazin, Y. Jeong, P.Y. Laffont, I.S. Kweon, C. Demonceaux, P. Vasseur. An Original Approach For Automatic Plane Extraction By Omnidirectional Vision. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'10), Octobre 2010, Taipei, Taiwan.

[C23] D. Eynard, P. Vasseur, C. Demonceaux, V. Frémont. UAV Altitude Estimation by Mixed Stereoscopic Vision. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'10), Octobre 2010, Taipei, Taiwan.

[C22] B. Magnier, F. Comby, O. Strauss, J. Triboulet, C. Demonceaux. Highly Specific Pose Estimation with a Catadioptric Omnidirectional Camera. In IEEE Int. Conf. on Imaging Systems and Techniques (IST'10), Juillet 2010, Thessaloniki, Grece.

[C21] S. Ly, C. Demonceaux, P. Vasseur. Multi-camera topology recovery using lines. International Conference on Computer Vision Theory and Applications (VISAPP'10), Mai 17-21, 2010, Angers, France.

[C20] S. Ly, C. Demonceaux, P. Vasseur. Translation estimation for single viewpoint cameras using lines. In IEEE Int. Conf. on Robotics and Automation (ICRA'10), Mai 2010, Anchorage, Alaska.

[C19] P. Vasseur, C. Demonceaux. Central Catadioptric Line Matching for Robotic Applications. In IEEE Int. Conf. on Robotics and Automation (ICRA'10), Mai 2010, Anchorage, Alaska.

[C18] A. Radgui, C. Demonceaux, E.M. Mouaddib, M. Rziza, D. Aboutajdine. Omnidirectional Egomotion Estimation From Adapted Motion Field. In IEEE 5th Int. Conf. on Signal-Image Technology and Internet-Based Systems (SITIS'09), Décembre 2009, Marrakech, Maroc.

[C17] C. Demonceaux, P. Vasseur. Omnidirectional Image Processing Using Geodesic Metric. In IEEE Int. Conf. on Image Processing (ICIP'09), Novembre 2009, Le Caire, Egypte.

[C16] J.C. Bazin, K.J. Yoon, I.S. Kweon, C. Demonceaux, P. Vasseur. Particle Filter Approach Adapted to Catadioptric Images for Target Tracking Application. In 20th British Machine Vision Conference, (BMVC'09), Septembre 2009, Londres, UK.

[C15] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. Dynamic Programming and Sky-line Extraction in Catadioptric Infrared Images. In IEEE Int. Conf. on Robotics and Automation (ICRA'09), Mai 2009, Kobe, Japon.

[C14] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. Improvement of Feature Matching in Catadioptric Images using Gyroscope Data. In 19th IAPR Int. Conf. on Pattern Recognition (ICPR'08), Décembre 2008, Tampa, FL, USA.

[C13] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. A Robust Top-Down Approach for Rotation Estimation and Vanishing Points Extraction by Catadioptric Vision in Urban Environment In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'08), Septembre 2008, Nice, France.

[C12] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. Automatic Calibration of Catadioptric Cameras In Urban Environment. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'08), Septembre 2008, Nice, France.

[C11] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. UAV Attitude Estimation by Vanishing Points In Catadioptric Image. In IEEE Int. Conf. on Robotics and Automation (ICRA'08), Mai 2008, Pasadena, Californie, USA.

[C10] C. Simler, C. Demonceaux, P. Vasseur. Proposition And Comparison Of Catadioptric Homography Estimation Methods In IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT'07), Décembre 2007, Santiago, Chili.

[C9] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. UAV attitude estimation by combining horizon-based and homography-based approaches for catadioptric image. In Proc. on 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'07), September 2007, Toulouse, France.

[C8] J.C. Bazin, C. Demonceaux, P. Vasseur. Fast Central Catadioptric Line Extraction. In 3rd Iberian Conference On Pattern Recognition and Image Analysis (IbPRIA'07), Juin 2007, Girone, Espagne.

[C7] C. Demonceaux, P. Vasseur, C. Pégard. UAV Attitude Computation by Omnidirectional Vision in Urban Environment. In IEEE Int. Conf. on Robotics and Automation (ICRA'07), Avril 2007, Rome, Italie.

[C6] C. Demonceaux, P. Vasseur, C. Pégard. Robust Attitude Estimation with Catadioptric Vision. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'06), Octobre 2006, Pékin, Chine.

[C5] C. Demonceaux, P. Vasseur. Adaptive Markov Random Fields for Omnidirectional Vision. In 18th IAPR Int. Conf. on Pattern Recognition (ICPR'06), Août 2006, Hong Kong.

[C4] C. Demonceaux, P. Vasseur, C. Pégard. Omnidirectional vision on UAV for attitude computation. In IEEE Int. Conf. on Robotics and Automation (ICRA'06), Mai 2006, Orlando, Floride.

[C3] C. Demonceaux, D. Kachi-Akkouche. Fast motion estimation and motion segmentation using multi-scale approach. In IEEE Int. Conf. on Image Processing (ICIP'04), Octobre 2004, Singapour.

[C2] C. Demonceaux, A. Potelle, D. Kachi-Akkouche. Obstacle detection in road scene based on road detection and motion analysis. In 5th IFAC/EURON Intelligent Autonomous Vehicles 2004 (IAV 04), Juillet 2004, Lisbonne, Portugal.

[C1] C. Demonceaux, D. Kachi-Akkouche. Robust obstacle detection with monocular vision based on motion analysis. In IEEE Intelligent Vehicles Symposium (IV'04), Juin 2004, Parme, Italie.

Articles dans des Workshops Internationaux

[W6] F. Rameau, D. Sidibé, C. Demonceaux, D. Fofi. Tracking Moving Objects With a Catadioptric Sensor Using Particle Filter. In 11th Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS'11), Novembre 2011, Barcelone, Espagne.

[W5] A. Radgui, C. Demonceaux, M. Rziza, E.M. Mouaddib, D. Aboutajdine. An adapted Lucas-Kanade's method for optical flow estimation in catadioptric images. In 8th Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS'08), Octobre 2008, Marseille, France.

[W4] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. Vanishing Points Matching for Rotation Estimation. In 8th Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS'08), Octobre 2008, Marseille, France.

[W3] J.C. Bazin, I.S. Kweon, C. Demonceaux, P. Vasseur. Rectangle Extraction in Catadioptric Images. In 7th Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS'07), Octobre 2007, Rio de Janeiro, Brésil.

[W2] C. Demonceaux, D. Kachi-Akkouche. Motion detection using wavelet analysis and hierarchical markov model. Workshop on Spatial Coherence for Visual Motion Analysis (SCV-MA'04), Mai 2004, Prague.

[W1] C. Demonceaux, D. Kachi-Akkouche. Optical flow estimation in omnidirectional images using wavelet analysis. Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS'03), Juin 2003, Madison, Wisconsin, USA.

Articles dans des congrès nationaux

[N7] F. Rameau, D. Sidibé, C. Demonceaux, D. Fofi. Une approche performante de suivi visuel pour les caméras catadioptriques. RFIA 2012, 18ème Congrès de Reconnaissance des Formes et Intelligence Artificielle, Janvier 2012, Lyon.

[N6] D. Eynard, P. Vasseur, C. Demonceaux, V. Frémont. Estimation de mouvement d'un drone à partir d'un capteur stéréo hybride. Congrès de jeunes chercheurs en vision par ordinateur, ORASIS 2011, Praz-sur-Arly, France, Juin 2011.

[N5] S. Ly, C. Demonceaux, P. Vasseur, C. Pégard. Estimation de pose d'un réseau de caméras hybrides à partir de droites. Congrès de jeunes chercheurs en vision par ordinateur, ORASIS 2011, Praz-sur-Arly, France, Juin 2011.

[N4] S. Ly, C. Demonceaux, P. Vasseur. Estimation de mouvement d'un système stéréoscopique hybride utilisant les droites. RFIA 2010, 17ème Congrès de Reconnaissance des Formes et Intelligence Artificielle, Janvier 2010, Caen.

[N3] D. Eynard, P. Vasseur, C. Demonceaux, V. Frémont. Estimation temps réel de l'altitude

d'un drone à partir d'un capteur de stéréovision mixte. RFIA 2010, 17ème Congrès de Reconnaissance des Formes et Intelligence Artificielle, Janvier 2010, Caen.

[N2] A. Radgui, C. Demonceaux, E. Mouaddib, M. Rziza, D. Aboutajdine. Caméras Omnidirectionnelles : Principes et Modélisations. In JOTI2008, Journées d'Optique et de Traitement de l'Information, Mohammedia, Maroc, Avril 2008.

[N1] C. Demonceaux, P. Vasseur. Mesure d'Attitude pour les Drones par Vision Catadioptrique Centrale. RFIA 2006, 15ème Congrès de Reconnaissance des Formes et Intelligence Artificielle, Janvier 2006, Tours.

Chapitre 2

Modèle sphérique unifié

2.1 Introduction

Dans ce travail, nous nous sommes intéressés à différents types de caméras : les caméras perspectives (Fig. 2.1(a)), les caméras catadioptriques à point de vue unique (Fig. 2.1(c)), les caméras fish-eye (Fig. 2.1(e)) et les Lady-Bug¹ (Fig. 2.1(g)). Les images obtenues à partir de ces caméras sont de prime abord très différentes les unes des autres. Les caméras perspectives permettent d'obtenir une photographie de la scène proche de la perception humaine (Fig. 2.1(b)). Les caméras catadioptriques aboutissent à une image à 360° de la scène au prix de fortes distorsions (Fig. 2.1(d)). Les optiques fish-eye permettent d'obtenir un champ de vue proche de 180° mais là aussi avec d'importantes distorsions (Fig. 2.1(f)). Enfin, les caméras du type Lady-Bug consistent en l'association de plusieurs caméras perspectives pour offrir une vision panoramique de l'environnement (Fig. 2.1(h)). Pour travailler avec l'ensemble de ces caméras, il est important de trouver un cadre générique dans lequel toutes ces caméras peuvent être étudiées. En 2001, Geyer et Daniilidis [GD01] ainsi que Barreto et Araujo ont montré que les images catadioptriques à point de vue unique pouvaient être considérées comme des images sphériques. Quelques années plus tard, Hu et al. [YH04] et Courbon et al. [CMEM07] ont démontré que nous pouvions aussi considérer les images fish-eye comme des images sphériques. Plus récemment, Kim et al. [KHK10] ont montré que pour un banc de caméras perspectives dont les centres optiques sont proches relativement à la profondeur de la scène, il était préférable d'utiliser un modèle sphérique plutôt que d'utiliser un modèle totalement générique.

L'objectif de ce chapitre est de rappeler comment ces caméras dites centrales ou quasi-centrales peuvent être considérées comme des caméras sphériques et de fournir les équations de passage entre les images acquises par ces caméras et leur représentation sphérique. Pour ce faire, nous allons reprendre pas à pas la formation des images catadioptriques à point de vue unique et fish-eye pour montrer que nous pouvons retrouver ces images à l'aide d'une double projection sur une sphère.

1. développée par la société PointGrey, www.ptgrey.com

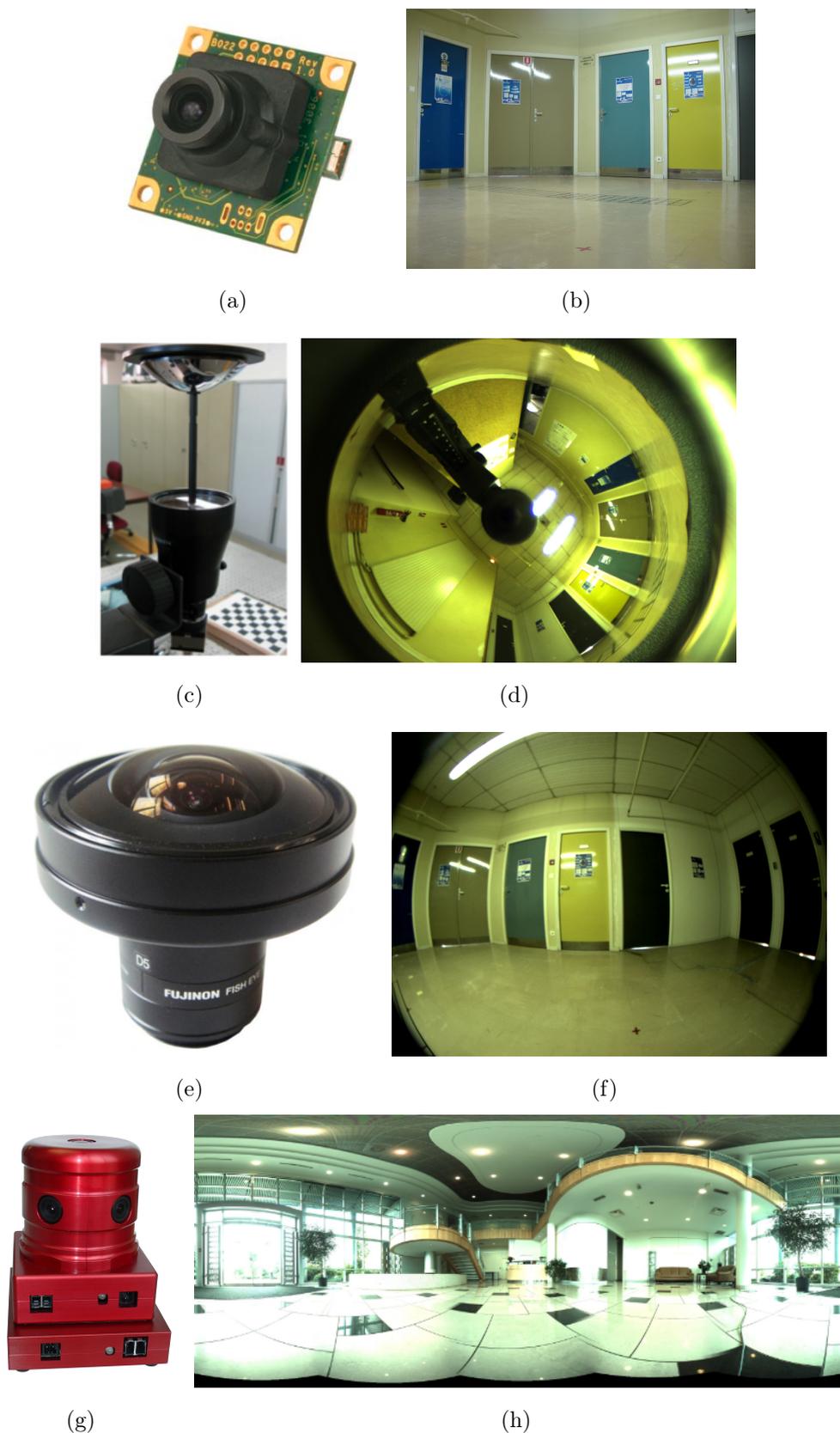


FIGURE 2.1 – Caméras étudiées dans ce manuscrit (a) caméra perspective (b) image perspective, (c) caméra paracatadioptrique (d) image paracatadioptrique, (e) optique fish-eye (f) Image fish-eye, (g) LadyBug (h) Image panoramique obtenue avec une LadyBug.

2.2 Caméra catadioptrique à point de vue unique

Nayar [Nay] a démontré qu'il existait quatre configurations physiquement réalisables pour obtenir un système de vision à point de vue unique à l'aide d'une caméra et d'un miroir. La première consiste à placer un miroir plan perpendiculairement au centre optique d'une caméra perspective de focale f à une distance $f/2$. La deuxième possibilité associe un miroir concave ellipsoïdal dont le second foyer correspond au centre optique de la caméra perspective. La troisième solution utilise un miroir hyperboloïde convexe dont le second foyer est confondu avec le centre optique de la caméra perspective. Enfin, une quatrième possibilité est de considérer un miroir paraboïde et une caméra orthographique. Seules les deux dernières solutions offrent un intérêt en pratique c'est la raison pour laquelle nous ne détaillerons que ces deux configurations.

2.2.1 Caméra hypercatadioptrique

Considérons un système catadioptrique obtenu en combinant un miroir hyperboloïde et une caméra perspective de focale f_e dont le centre optique est confondu avec le second foyer du miroir (Fig. 2.2).

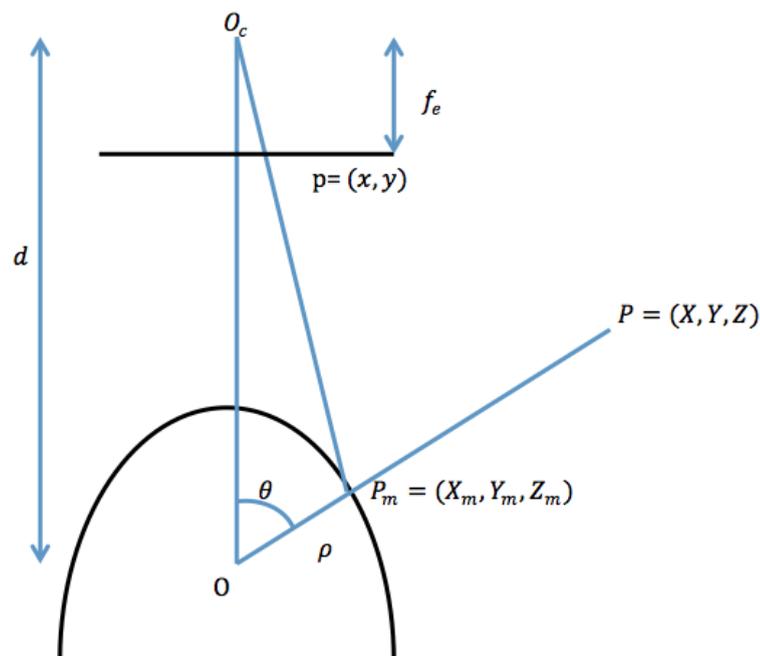


FIGURE 2.2 – Formation d'une image hypercatadioptrique.

Ce miroir a pour équation polaire :

$$\rho = \frac{p}{1 + e \cos(\theta)} \quad (2.1)$$

Un point 3D $P = (X, Y, Z)^T$ est projeté sur le miroir au point $P_m = (X_m, Y_m, Z_m)^T$ où :

$$(X_m, Y_m, Z_m)^T = \frac{\rho}{\sqrt{X^2 + Y^2 + Z^2}}(X, Y, Z)^T \quad (2.2)$$

Ce point est ensuite projeté dans le plan image au point $p = (x, y)^T$ à l'aide des matrices de projection suivantes :

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_e & 0 & 0 \\ 0 & f_e & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix} \quad (2.3)$$

où d est la distance entre la caméra et le miroir.

Or pour respecter le point de vue unique, $d = \frac{2ep}{1-e^2}$ (la caméra se situe au second foyer de l'hyperboloïde) et comme $\cos(\theta) = Z/\|P\|$, nous aboutissons à la relation suivante :

$$p = (x, y) = \left(\frac{\frac{1-e^2}{1+e^2} f_e X}{\frac{2e}{1+e^2} \sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{\frac{1-e^2}{1+e^2} f_e Y}{\frac{2e}{1+e^2} \sqrt{X^2 + Y^2 + Z^2} + Z} \right) \quad (2.4)$$

2.2.2 Caméra paracatadioptrique

Dans le cas d'une caméra paracatadioptrique, nous devons utiliser une caméra orthographique pour respecter le point de vue unique (Fig. 2.3). Nous pouvons retrouver les équations de projection en reprenant les calculs précédents et en considérant que $f_e \mapsto +\infty$, $d \mapsto +\infty$, $e = 1$, on a alors :

$$\frac{1-e^2}{1+e^2} f_e = \frac{2e^2}{1+e^2} p \frac{f_e}{d} \mapsto p,$$

d'où les relations suivantes :

$$(x, y) = \left(\frac{pX}{\sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{pY}{\sqrt{X^2 + Y^2 + Z^2} + Z} \right) \quad (2.5)$$

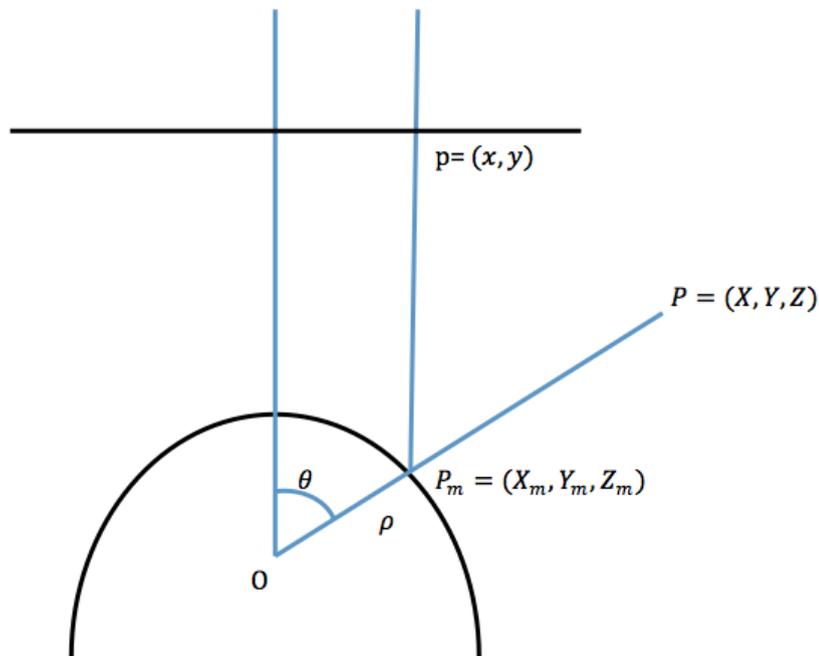


FIGURE 2.3 – Formation d’une image paracatadioptrique.

2.3 Caméra fish-eye

Pour obtenir un champ de vue plus important qu’une caméra perspective, une optique fish-eye introduit des distorsions radiales. Il existe différents modèles pour définir ces distorsions que nous pouvons classer en deux catégories [YH04, CMEM07].

La première considère la relation entre l’image fish-eye et l’image perspective associée. Soit $p = (x, y)$ l’image fish-eye d’un point $P = (X, Y, Z)^T$ de l’espace et notons $p' = (x', y')$ le point associé sur le modèle sténopé (Fig 2.4). Il s’agit alors de définir la relation entre $r = \sqrt{x^2 + y^2}$ et $r' = \sqrt{x'^2 + y'^2}$. Cette relation peut être modélisée par différentes transformations (polynomiales, logarithmiques, rationnelles) [CMEM07].

Par exemple, Fitzgibbon [Fit01] propose le modèle suivant :

$$r' = k_1 \frac{r}{1 - k_2 r^2} \quad (2.6)$$

La seconde approche consiste à définir une relation entre l’angle d’incidence θ et r . Là encore, il existe différents modèles (polynomiaux, rationnels, sinusoïdales...) que nous pouvons retrouver dans [CMEM07].

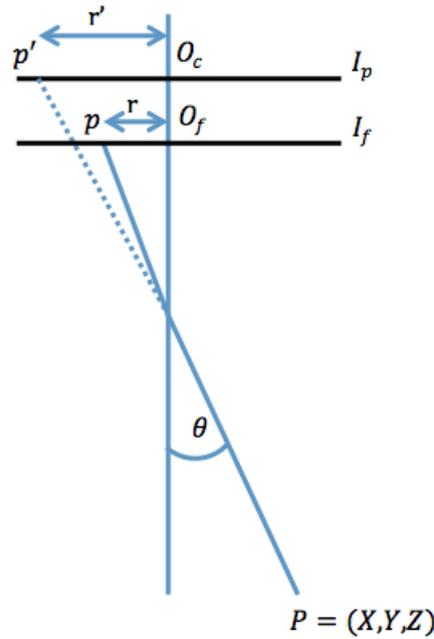


FIGURE 2.4 – Formation d’une image Fish-eye.

2.4 Modèle sphérique unifié

Nous allons voir que les images acquises à l’aide des caméras ci-dessus peuvent être obtenues à l’aide d’une double projection via une sphère unité (Fig 2.5). Dans un premier temps, le point 3D $P_s = (X_s, Y_s, Z_s)^T$ est projeté sur une sphère de rayon 1 en P_s .

$$P_s = \frac{1}{\|P\|} P \quad (2.7)$$

Le point P_s est ensuite projeté sur le plan image en $p = (x, y)$ à partir d’un point de O_c distant de l du centre de la sphère en direction du pôle nord. P_s et p vérifient les relations suivantes :

$$\frac{x}{X_s} = \frac{f_s}{l + Z_s} \quad \text{et} \quad \frac{y}{Y_s} = \frac{f_s}{l + Z_s}. \quad (2.8)$$

En combinant (2.7) et (2.8) :

$$p = (x, y) = \left(\frac{X f_s}{l \sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{Y f_s}{l \sqrt{X^2 + Y^2 + Z^2} + Z} \right). \quad (2.9)$$

Remarquons que pour $l = 0$, ce modèle est équivalent au modèle sténopé de focal f_s .

De plus, en comparant les équations (2.4) et (2.9), nous observons que les deux modèles sont équivalents si on pose : $l = \frac{2e}{1+e^2} < 1$ et $f_s = \frac{1-e^2}{1+e^2} f_e$.

De même à partir de (2.5) et (2.9), une image paracatadioptrique est équivalente à cette double

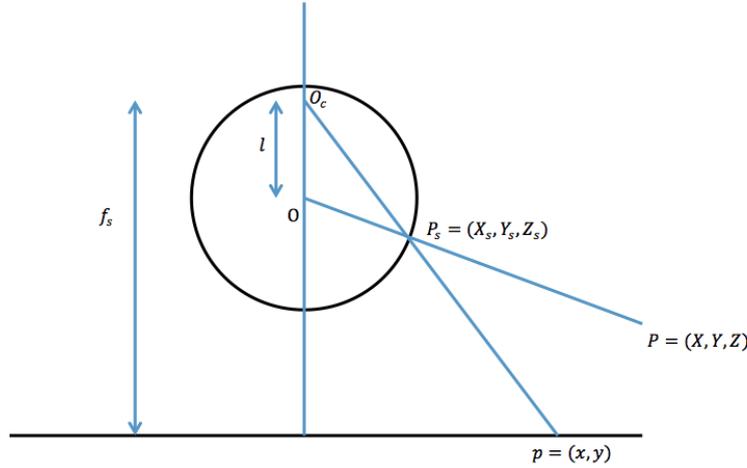


FIGURE 2.5 – Double projection sur une sphère unitaire

projection sur la sphère unité en posant : $l = 1$ et $f_s = p$ (projection stéréographique). Enfin, en considérant le cas où $l = 0$ et $f_s = f_p$ qui est équivalent à une projection perspective de focale f_p et le cas stéréographique $l = 1$ et $f_s = f_e$, nous obtenons deux images [YH04] :

$$(x', y') = \left(f_p \frac{X}{Z}, f_p \frac{Y}{Z} \right) \quad \text{et} \quad (x, y) = \left(\frac{f_e X}{\sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{f_e Y}{\sqrt{X^2 + Y^2 + Z^2} + Z} \right) \quad (2.10)$$

vérifiant :

$$r' = \sqrt{x'^2 + y'^2} = 2 \frac{f_p}{f_e} \frac{r}{1 - \frac{1}{f_e^2} r^2} \quad \text{avec} \quad r = \sqrt{x^2 + y^2}. \quad (2.11)$$

Ce qui permet de retrouver la transformation entre une fish-eye et l'image perspective associée proposée par Fitzgibbon et montre que le modèle sphérique intègre les distorsions d'une optique fish-eye. Nous pouvons d'ailleurs améliorer le modèle en considérant une projection sphérique quelconque ($l \neq 1$) comme Courbon et al. l'ont montré dans [CMEM07].

Enfin, dans le cas d'un banc de caméras perspectives dont les centres optiques sont confondus, le modèle sphérique est bien sûr lui aussi adapté à ce système de vision. Par contre, un tel dispositif ne peut pas physiquement être réalisé mais Kim et al. [KHK10] ont montré que le modèle sphérique restait une très bonne approximation dans le cas où les centres optiques des caméras sont proches relativement à la profondeur de la scène.

Par conséquent, nous venons de revoir que toutes les caméras utilisées dans ces travaux peuvent être considérées comme sphérique. La sphère offre donc un cadre générique pour étudier ces caméras centrales ou quasi-centrales.

La suite de ce manuscrit est consacrée à démontrer que cette modélisation permet de développer des algorithmes génériques pour ces caméras. Dans le prochain chapitre, nous verrons comment cette modélisation permet d'adapter les méthodes de traitement d'images classiques et de calculer le flot optique sur des images grand angle. Puis, dans un quatrième chapitre, nous développerons une nouvelle méthode d'estimation du déplacement de caméras sphériques (centrales

ou quasi-centrales) en tenant compte des particularités de ces images. Enfin, nous utiliserons ce formalisme pour des applications en robotique aérienne.

Chapitre 3

Traitement des images grand angle

Traiter une image consiste à la transformer afin d'en modifier les caractéristiques en vue d'une amélioration (restauration d'images) ou d'une détection de primitives (contours, points, droites...). Ces opérations dites de bas niveau sont les premières étapes nécessaires pour des applications de plus haut niveau : calcul de pose d'une caméra, estimation du déplacement d'un robot, reconstruction 3D... Bien évidemment, de nombreux outils existent désormais pour traiter une image. Néanmoins, les méthodes usuelles sont pour la plupart inadaptées aux images grand angle à cause des fortes distorsions inhérentes à ce type d'image. Pour s'en convaincre, la figure 3.1 qui correspond à une image prise par une caméra catadioptrique en est un exemple. Sur cette image, le point P1 est séparé du même nombre de pixels de P2 que P3 avec P4, alors que les points correspondants de la scène ne sont pas à la même distance. Ainsi, les méthodes de filtrage classique, qui tiennent compte de la position des points les uns par rapports aux autres, se révèlent inappropriées pour ce type d'image. C'est pourquoi, il faut adapter les méthodes usuelles en tenant compte des particularités de ces images. Ce premier chapitre s'intéresse aux travaux réalisés dans ce cadre pour des images grand angle à travers différents exemples et applications.

Ces travaux ont débuté en 2005 dès mon recrutement en tant que maître de conférences. Dans un premier temps, nous nous sommes penchés sur l'utilisation des champs de Markov pour les images catadioptriques. Les champs de Markov sont des outils forts utiles en traitement d'images. Ils sont fondés sur la notion même d'interdépendance entre les pixels voisins d'une image. Or comme nous venons de le remarquer, ces hypothèses classiques se révèlent inadaptées et se doivent d'être modifiées pour des images grand angle. C'est ainsi que nous avons revu la notion de voisinage sur ce type d'image pour adapter les méthodes markoviennes en conséquence ([DV06]). En parallèle, nous nous sommes intéressés au problème du calcul du flot optique sur ces images. Nous avons ainsi modifié les hypothèses de la méthode de Lucas-Kanade ([RDM⁺08]). Ces travaux ne concernaient qu'un seul type d'image grand angle à savoir les images catadioptriques sur lesquelles nous travaillions directement sur l'image plane. Par la suite, nous avons choisi d'étendre ces approches pour une plus grande catégorie de caméras. C'est ainsi que nous avons développé des méthodes sur des images sphériques qui recouvrent tout type d'images à point de vue unique (perspectives, catadioptriques, fish-eye, lady-bug...). Nous consacrerons ce chapitre à détailler ces travaux. Dans une première partie, nous verrons comment on peut traiter une image sphérique. Puis nous nous intéresserons à l'estimation du flot optique sphérique.

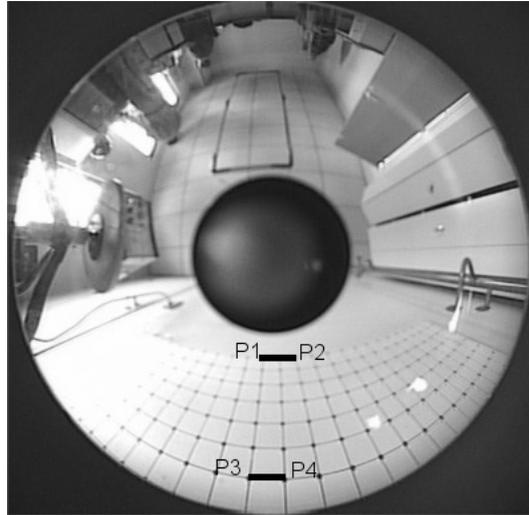


FIGURE 3.1 – $d(P1, P2) = d(P3, P4)$ alors que le point $P1$ n'a pas physiquement la même influence sur $P2$ que $P3$ sur $P4$.

3.1 Filtrage d'images grand angle

Comme nous l'avons vu, il n'est pas pertinent d'appliquer directement sur ces images des opérateurs de convolution classique. Tous les filtrages définis à partir d'un noyau se doivent d'être adaptés afin de tenir compte de la particularité de ces images. Nous pouvons par exemple redresser les images avant de leur appliquer les traitements classiques mais cette étape nécessite d'interpoler les images et modifie donc la nature de celles-ci. Nous pouvons aussi modifier les noyaux des filtres afin de les rendre isotropes mais cela nécessite de redéfinir chaque opérateur. Dans notre travail, nous avons proposé une méthode de filtrage des images grand angle en modifiant simplement la métrique. Nous verrons que cette modification permet d'utiliser les filtres usuels. Nous résumons par la suite notre approche, une étude plus détaillée a été publiée dans [DVF11].

3.1.1 Voisinage géodésique

Notons P la projection qui transforme une image à point de vue unique I de \mathbb{R}^2 en son image sphérique équivalente sur \mathcal{S}^2 :

$$P : \begin{array}{ccc} \mathbb{R}^2 & \rightarrow & \mathcal{S}^2 \\ x & \mapsto & x_s = (\theta, \phi) \end{array} \quad (3.1)$$

Pour tenir compte de la géométrie particulière de nos images, nous avons proposé d'utiliser la distance géodésique :

$$\forall x_s = (\theta, \phi), y_s = (\theta', \phi') \in \mathcal{S}^2 \\ d_{\mathcal{S}^2}(x, y) = \arccos \left[\begin{pmatrix} \cos(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) \\ \cos(\theta) \end{pmatrix}^T \cdot \begin{pmatrix} \cos(\phi') \sin(\theta') \\ \sin(\phi') \sin(\theta') \\ \cos(\theta') \end{pmatrix} \right].$$

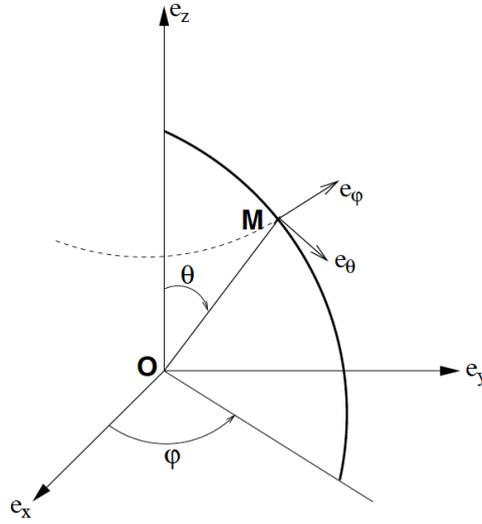


FIGURE 3.2 – Coordonnées sphériques et base locale.

Notons $x_s \in \mathcal{S}^2$, la projection d'un pixel x de l'image sur la sphère d'équivalence ($x_s = P(x)$). Nous définissons le voisinage continu $\mathcal{V}_r(x)$ du pixel x de l'image comme suit :

$$\mathcal{V}_r(x) = \{y_s \in \mathcal{S}^2, d_{\mathcal{S}^2}(x_s, y_s) \leq r \mid P(y_s) = x\} \quad (3.2)$$

Ce voisinage résulte de l'intersection entre le cône d'apex O (centre de la sphère) de direction Ox_s , et d'angle r , avec la sphère \mathcal{S}^2 .

Pour convoluer avec un filtre discret, il est nécessaire d'introduire une version discrète de (3.2). Considérons alors le plan tangent π à \mathcal{S}^2 en x_s . Ce plan admet pour équation $x_s + a\vec{e}_\theta + b\vec{e}_\phi$, $(a, b) \in \mathbb{R}^2$, où $(\vec{e}_\theta, \vec{e}_\phi)$ sont les vecteurs de base en $x_s = (\theta, \phi)$ (fig 3.2) :

$$\vec{e}_\theta = \frac{\partial O\vec{M}}{\partial \theta} = \begin{pmatrix} \cos \phi \cos \theta \\ \sin \phi \cos \theta \\ -\sin \theta \end{pmatrix}, \quad \vec{e}_\phi = \frac{1}{\sin \theta} \frac{\partial O\vec{M}}{\partial \phi} = \begin{pmatrix} -\sin \phi \\ \cos \phi \\ 0 \end{pmatrix}.$$

La grille régulière de $(2N+1)^2$ de π que nous avons proposée est définie de la façon suivante :

$$x_s + \tan(nr)\vec{e}_\theta + \tan(pr)\vec{e}_\phi, \quad -N \leq n, p \leq N \quad (3.3)$$

c'est-à-dire que le voisinage du point x_s correspond aux $(2N+1)^2$ points de \mathcal{S}^2 qui vérifient¹ :

$$\mathcal{V}_r^N(x) = \{x_s(n, p) = \frac{x_s + \tan(nr)\vec{e}_\theta + \tan(pr)\vec{e}_\phi}{\|x_s + \tan(nr)\vec{e}_\theta + \tan(pr)\vec{e}_\phi\|}, \quad -N \leq n, p \leq N, x_s = P(x)\}. \quad (3.4)$$

Par analogie au cas plan, on a ainsi construit une grille régulière de $(2N+1)^2$ points centrée en x_s telle que la distance géodésique entre un point et ses plus proches voisins est exactement égale à r :

1. La notion de voisinage géodésique définie ici requiert une phase d'interpolation. Dans les expérimentations montrées par la suite, nous avons simplement obtenu $I_{\mathcal{V}_r^N(x)}$ par une interpolation au "plus proche voisin".

$$\forall y \in \mathcal{V}_r^N(x) \quad \min_{z \in \mathcal{V}_r^N(x) \setminus y} d_{S^2}(y, z) = r. \quad (3.5)$$

Remarquons, que nous aurions très bien pu définir le voisinage sur le plan tangent π pour lequel la distance euclidienne $d_{\mathbb{R}^2}(x, y)$ est appropriée. Nous aurions alors :

$$\mathcal{V}_{r, \mathbb{R}^2}^N(x) = \{x_s + \tan(r)(n\vec{e}_\theta + p\vec{e}_\phi), \quad -N \leq n, p \leq N\}. \quad (3.6)$$

Sur ce voisinage $\mathcal{V}_{r, \mathbb{R}^2}^N(x)$, les points définissent un quadrillage régulier et l'on a une relation semblable à (3.5) :

$$\forall y_\pi \in \mathcal{V}_{r, \mathbb{R}^2}^N(x), \quad \min_{z_\pi \in \mathcal{V}_{r, \mathbb{R}^2}^N(x) \setminus y_\pi} d_{\mathbb{R}^2}(y_\pi, z_\pi) = \tan r.$$

$\mathcal{V}_{r, \mathbb{R}^2}^N(x)$ et $\mathcal{V}_r^N(x)$ sont bien évidemment différents car $\tan nr \neq n \tan r$. Néanmoins, si l'on considère un angle de vue de $r = 5^\circ$, qui correspondrait à un découpage de la sphère en $180/5 \times 360/5 = 36 \times 72$ pixels et si l'on considère un voisinage de taille 9×9 , les points extrêmes des voisinages $\mathcal{V}_{r, \mathbb{R}^2}^N(x)$ et $\mathcal{V}_r^N(x)$ seraient distants de moins de 0.02 pixels sur π . C'est pourquoi, on peut considérer le voisinage géodésique comme une approximation du voisinage euclidien sur π projeté sur la sphère. Par conséquent, les filtres classiques peuvent être utilisés directement sur (3.5).

3.1.2 Convolution

Filtrer une image consiste à appliquer un masque pondéré en chaque pixel de l'image. Cette opération est réalisée à partir d'une convolution. Nous montrons par la suite comment redéfinir une convolution à partir de la métrique géodésique et nous verrons que grâce à cette formulation, les filtres classiques restent valides.

Pour un pixel donné x d'une image I , nous considérons son voisinage sphérique $\mathcal{V}_r^N(x)$ que nous reprojets dans le plan image :

$$I_{\mathcal{V}_r^N(x)}(n, p) = I(P^{-1}(x_s(n, p))) \quad -N \leq n, p \leq N.$$

où les $x_s(n, p)$ sont les $(2N + 1)^2$ voisins de la grille régulière centrée en $P(x)$.

Soit H un filtre de taille $(2N + 1) \times (2N + 1)$, nous définissons le produit de convolution de l'image I par le filtre H au point $x \in \mathbb{R}^2$ comme suit :

$$\begin{aligned} \forall x \in \mathbb{R}, IH(x) &= I_{\mathcal{V}_r^N(x)} * H(x) \\ &= \sum_{i=-N}^N \sum_{j=-N}^N I_{\mathcal{V}_r^N(x)}(i, j) H(i, j). \end{aligned} \quad (3.7)$$

A partir de la définition ci-dessus, nous pouvons calculer la dérivée de l'image sphérique au point x_s par un filtre de Sobel classique :

$$\|\nabla I(x_s)\|^2 \simeq |I_{\mathcal{V}_r^1(x)} * S|^2 + |I_{\mathcal{V}_r^1(x)} * S^T|^2.$$

où

$$S = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

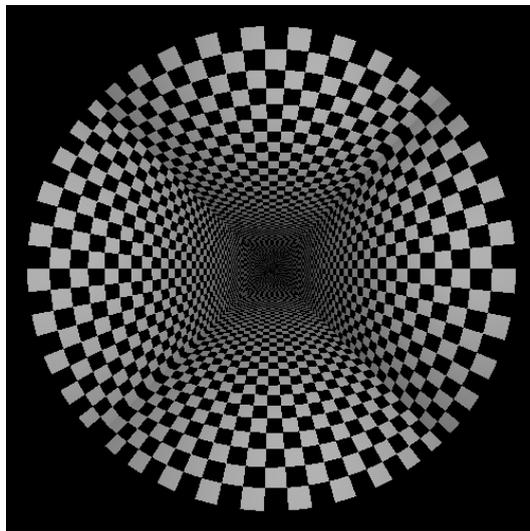


FIGURE 3.3 – Image catadioptrique test.

Afin de vérifier le comportement du filtrage géodésique, nous appliquons notre filtrage sur une image de synthèse (fig 3.3) et comparons avec un filtrage classique et également avec la méthode proposée par Bogdanova et al. ([BBTV07]) qui calcule le gradient sphérique. Cette image virtuelle est issue d'une caméra catadioptrique placée à l'intérieur d'un parallélépipède rectangle quadrillé. Alors qu'un filtrage classique semble "écraser" les contours au centre de l'image, la méthode de Bogdanova et al. atténue au contraire la réponse au filtre sur cette zone. Notre approche semble être un bon compromis entre ces résultats (3.4(f)). Cela se confirme sur une image réelle (fig. 3.5). Ainsi, même s'il est difficile sur des données qualitatives de tirer des conclusions, on peut affirmer que l'approche géodésique pour le calcul du gradient est comparable à la méthode de Bogdanova et al. Par contre, cette méthode qui consiste à calculer le gradient sphérique à l'aide d'une Jacobienne n'est pas appropriée pour des filtrages d'images quelconques alors que la méthode que nous avons proposée reste facilement utilisable pour tout type de filtrage.

Par exemple, on peut considérer le filtrage avec une Gaussienne comme une pondération des points les uns par rapport aux autres en fonction de leur distance au point en question. On peut donc remplacer la Gaussienne :

$$G_x(y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\|x-y\|^2}{2\sigma^2}} \quad (x, y) \in \mathbb{R}^2,$$

par

$$G_x(y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{d_{\mathcal{S}^2}(x,y)}{2\sigma^2}} \quad (x, y) \in \mathcal{S}^2. \quad (3.8)$$

On peut de la même façon définir le Laplacien de Gaussienne :

$$LoG_x(y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{d_{\mathcal{S}^2}(x,y)}{2\sigma^2} \right] \exp -\frac{d_{\mathcal{S}^2}(x,y)}{2\sigma^2}. \quad (3.9)$$

Ainsi, alors qu'à partir de l'analyse harmonique, la gaussienne ou le laplacien nécessitent d'être recalculés pour le traitement d'images sphériques [Bül02][DMB02], notre approche permet

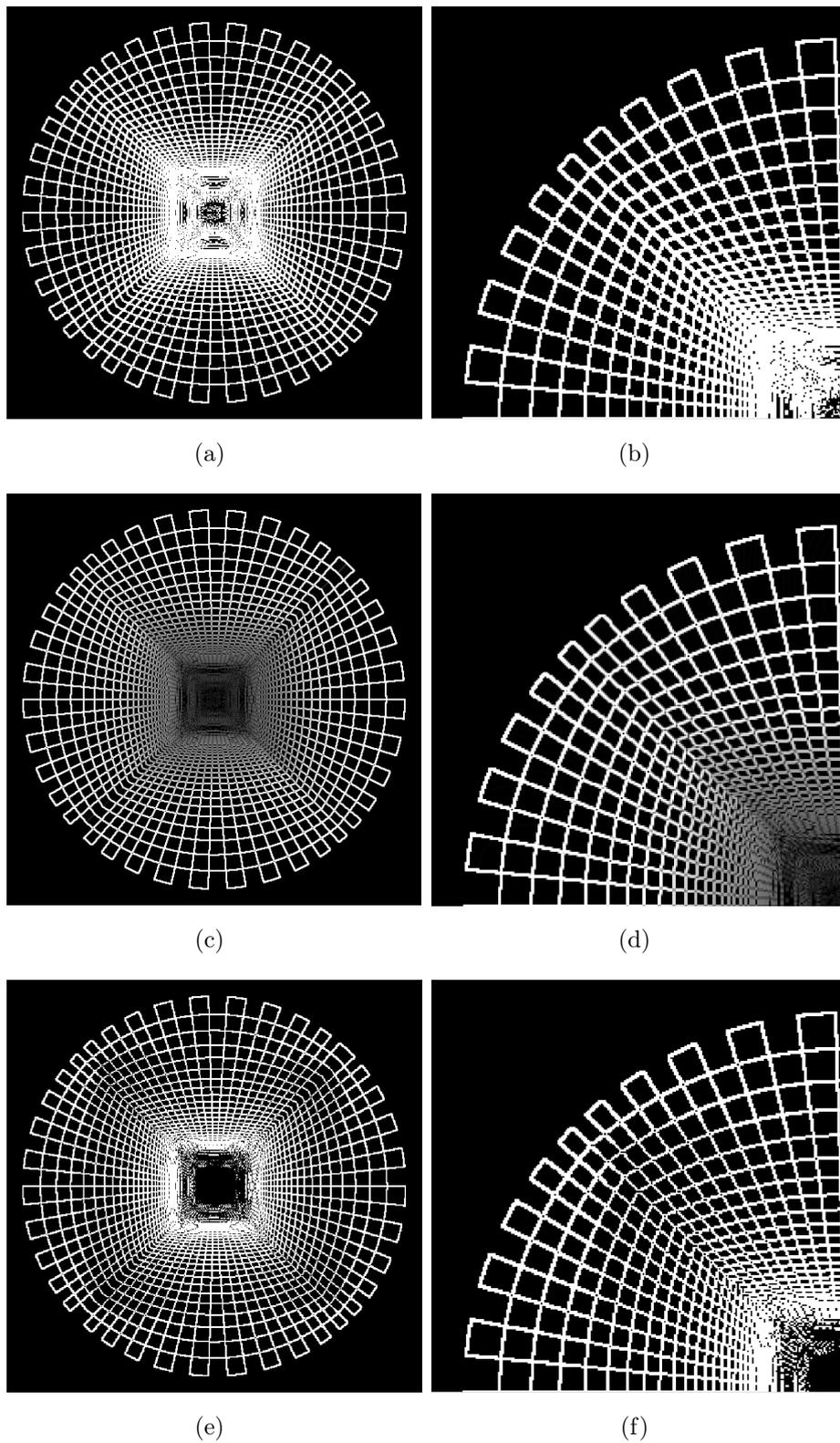


FIGURE 3.4 – Résultat de l'opérateur gradient sur l'image test. (a),(b) approche classique, (c) (d) méthode de Bogdanova [BBTV07], (e)(f) approche géodésique.

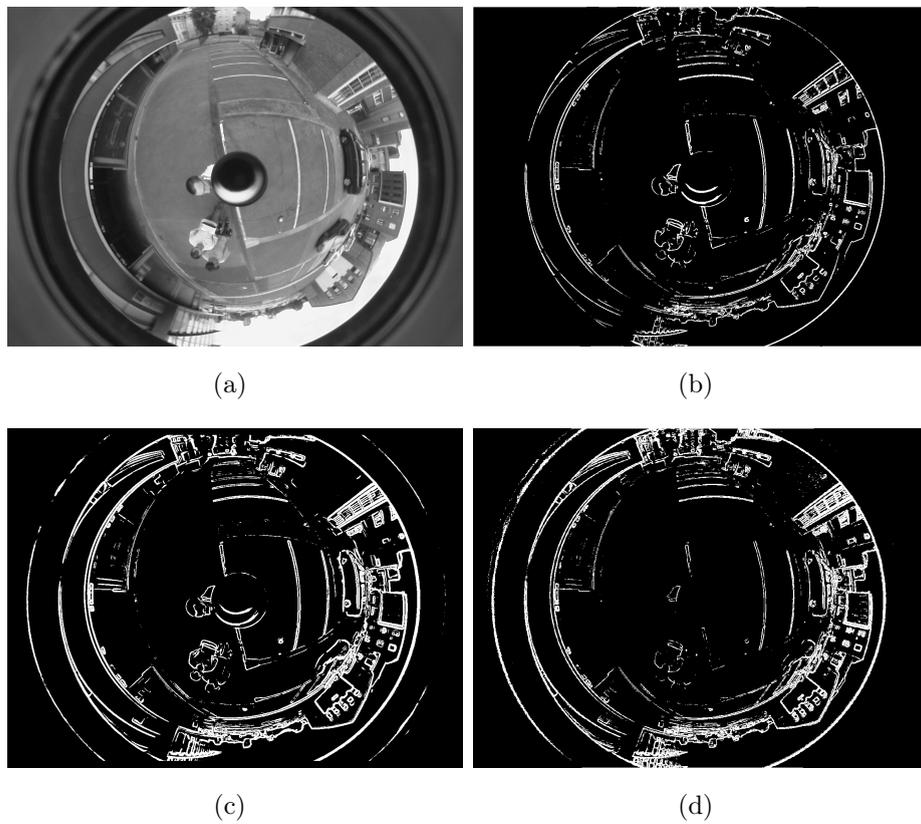


FIGURE 3.5 – Norme du gradient sur une image réelle. (a) image réelle, (b) approche classique, (c) approche géodésique, (d) méthode de Bogdanova [BBTV07]

Image	Harris Adapté		Harris Classique	
	I_1	I_2	I_1	I_2
Points d'intérêts	231	227	212	239
Points sur les casiers	50	51	55	36
Répétabilité	43 →86%		29 →80%	

TABLE 3.1 – Détecteur de Harris, approche adapté/approche classique.

une adaptation directe des filtres conventionnels. En effet, grâce à (3.8, 3.9) et au produit de convolution défini en (3.7) les masques utilisés en traitement d'images perspectives restent valides.

3.2 Détection de points d'intérêts et mise en correspondance

Nous allons montrer comment l'approche développée dans le paragraphe précédent permet d'adapter le détecteur de coins de Harris et d'améliorer la mise en correspondance pour ce type d'images.

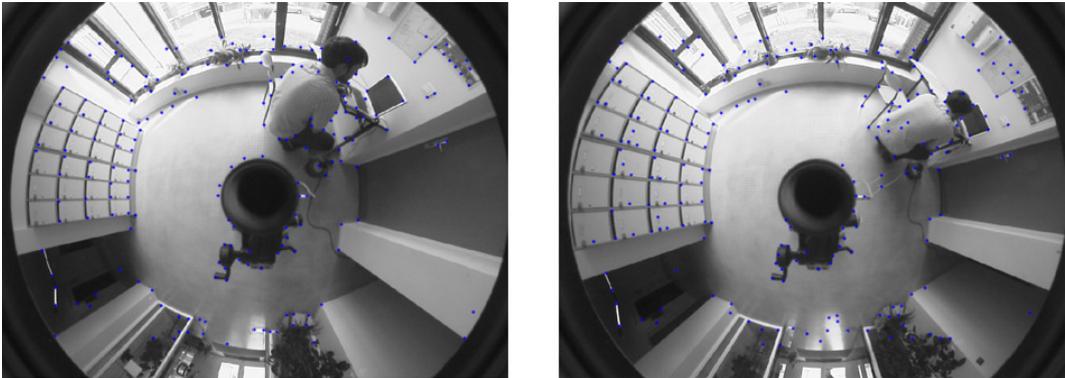
3.2.1 Détecteur de Harris

Le détecteur de Harris est un détecteur de coins dans les images très utile en vision par ordinateur. En traitement d'images perspectives, détecter les coins par la méthode de Harris consiste à étudier les valeurs propres de la matrice M en chaque point X :

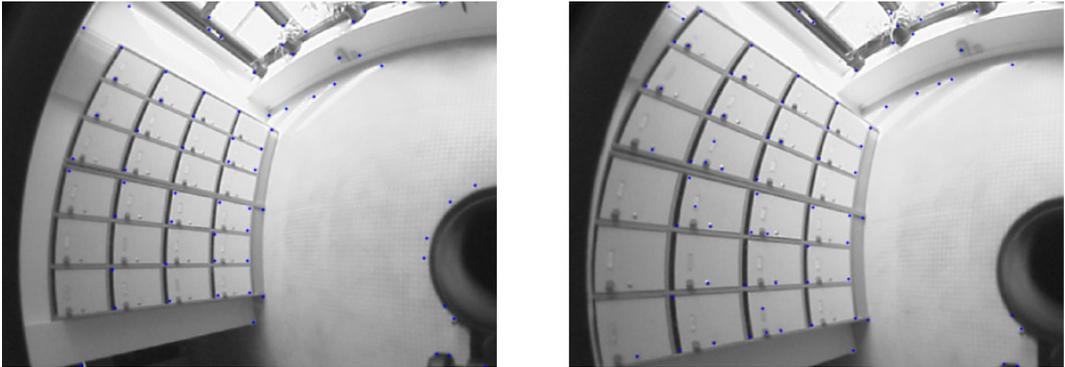
$$M(X) = \begin{bmatrix} L_x^2(X, \sigma) & L_x L_y(X, \sigma) \\ L_x L_y(X, \sigma) & L_y^2(X, \sigma) \end{bmatrix} \quad (3.10)$$

où $L_i(X, \sigma) = \frac{\partial}{\partial i} G_X(X, \sigma) * I(X)$

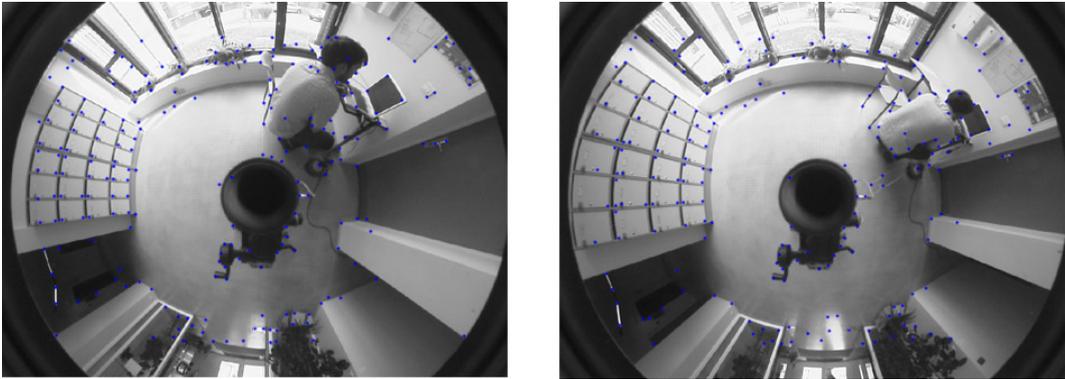
Dans le cas des images omnidirectionnelles, cette définition reste valable en utilisant la convolution définie en (3.7) et (3.8). Les résultats du détecteur de Harris sont montrés figure 3.6 sur deux images consécutives d'une séquence. Dans Tab 3.1, nous avons comparé le facteur de répétabilité [SMB00] du détecteur avec un détecteur de Harris classique. Sur le casier (fig 3.6(b) et 3.6(d)), 50 coins ont été détectés sur I_1 dans le cas du Harris proposé et 55 avec un Harris classique. Sur la deuxième image, le Harris proposé détecte 51 coins alors que la Harris classique n'en détecte que 36. Dans le cas classique, seul 29 coins identiques ont été détectés dans les deux images soit une répétabilité de 80%, avec le Harris calculé avec la métrique géodésique, nous avons 43 coins en commun soit une répétabilité de 86%.



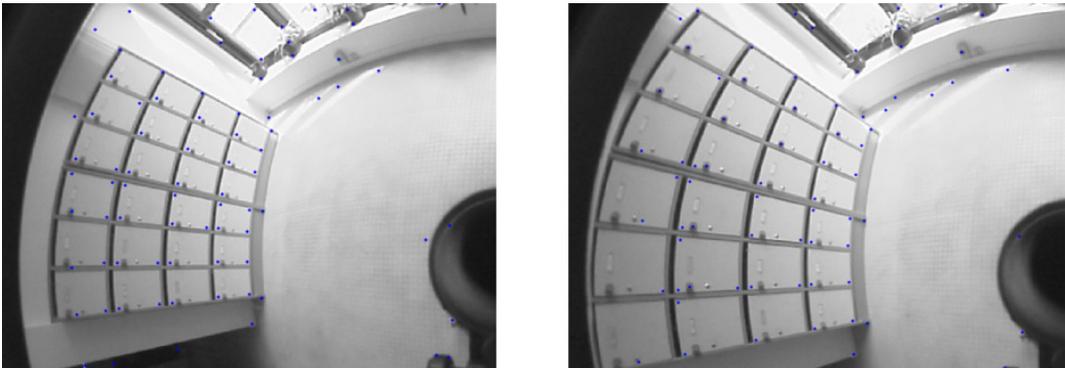
(a)



(b)



(c)



(d)

FIGURE 3.6 – Détecteur de Harris. (a)-(b) méthode classique,(c)-(d) méthode adaptée

3.2.2 Mise en correspondance : ZNCC

La mise en correspondance de points caractéristiques dans les images est aussi un problème important en vision par ordinateur. Les mesures de similarité par corrélation sont elles aussi à revoir car elles sont définies à partir du voisinage entre pixels. Montrons par exemple, qu'une modification du voisinage pour le calcul de la ZNCC "*Zero mean Normalized Cross-Correlation*" permet d'améliorer les résultats de mise en correspondance. La ZNCC est une mesure définie comme suit :

$$ZNCC(x, y) = \frac{\sum_{i \in \mathcal{V}(x)} \sum_{j \in \mathcal{V}(y)} (I_1(i) - \overline{I_1(x)})(I_2(j) - \overline{I_2(y)})}{\sqrt{\sum_{i \in \mathcal{V}(x)} (I_1(i) - \overline{I_1(x)})^2 \sum_{j \in \mathcal{V}(y)} (I_2(j) - \overline{I_2(y)})^2}} \quad (3.11)$$

où $\overline{I_1(x)}$ (respectivement $\overline{I_2(y)}$) est le niveau de gris moyen de l'image I_1 (respectivement I_2) dans le voisinage $\mathcal{V}(x)$ (respectivement $\mathcal{V}(y)$).

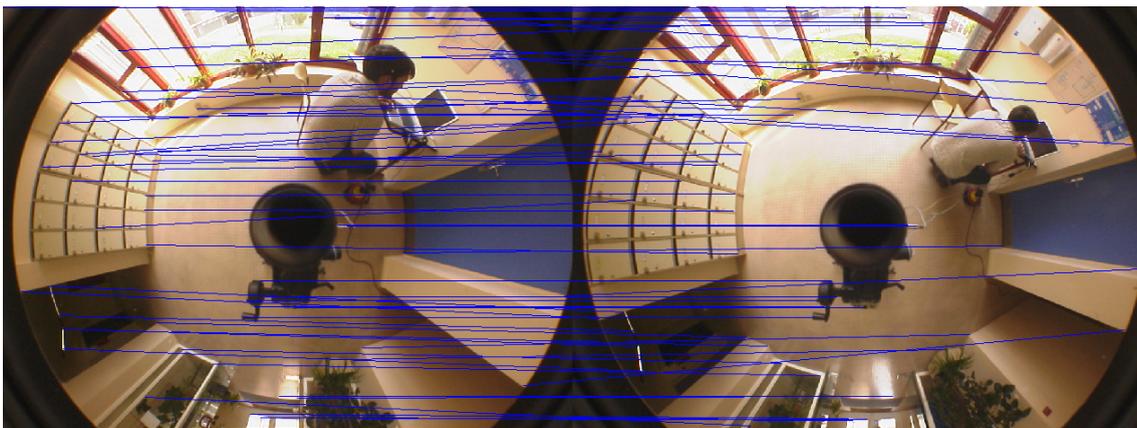
Nous avons comparé un voisinage classique de taille 7×7 avec un voisinage adapté \mathcal{V}_r^3 (fig 3.7). Sur les mêmes points d'intérêt et avec les mêmes seuils, une ZNCC classique permet de mettre en correspondance 65 points dont 53 corrects soit 18.4% d'outliers alors qu'une adaptation du voisinage à partir de (3.4) permet de mettre en correspondance 71 points dont 63 corrects soit 11.2% d'outliers (figure 3.7).

La figure 3.9 compare les résultats de la détection de coins et de mise en correspondance sur une séquence extérieure de 21 images consécutives (figure 3.8). Contrairement aux résultats précédents où la mise en correspondance avait été effectuée avec un détecteur de Harris classique, nous avons ici comparé avec un voisinage adapté dans les deux phases : Harris et correspondance. Les barres vertes (respectivement rouges) de l'histogramme de la figure 3.9 correspondent aux nombres de points mis en correspondance avec un voisinage adapté (resp. classique), les barres bleues (resp. noires) dénombrent les correspondances incorrectes avec un voisinage adapté (resp. classique). Ce résultat montre une réelle amélioration de la détection et de la mise en correspondance des points avec un voisinage tenant compte des propriétés de ces images. En effet, en moyenne, sur les 21 images de la séquence la méthode adaptée permet de mettre en correspondance 152 points alors qu'avec un voisinage classique, on trouve seulement 116 correspondances avec un nombre d'échec moyen respectif de 3.5 et 14.5.

Par conséquent, nous venons de voir que modifier le voisinage des images à fortes distorsions par passage sur la sphère d'équivalence permet d'améliorer les techniques de filtrage, de détection et de mise en correspondance. Nous allons voir par la suite, que la sphère peut aussi permettre d'estimer le flot optique sur ces images avec une plus grande précision.



(a)



(b)

FIGURE 3.7 – Mise en correspondance par ZNCC. (a) méthode classique : 65 points en correspondances dont 53 corrects, (b) méthode adaptée, 71 points en correspondances dont 63 corrects.

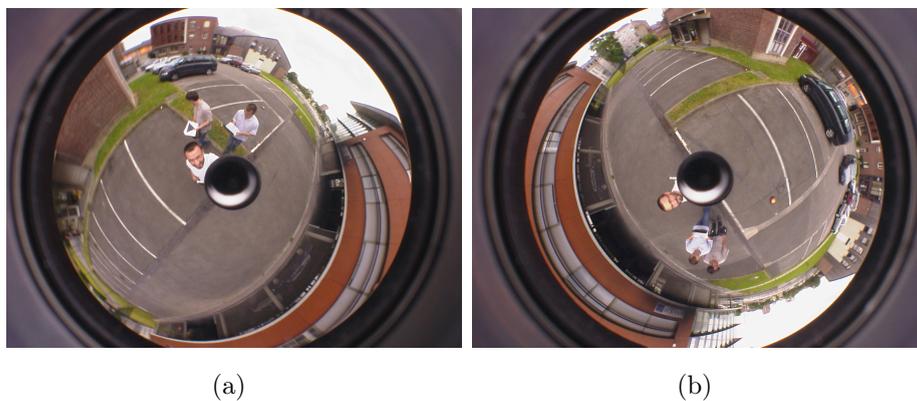


FIGURE 3.8 – (a) (b) 2 images de la séquence réelle.

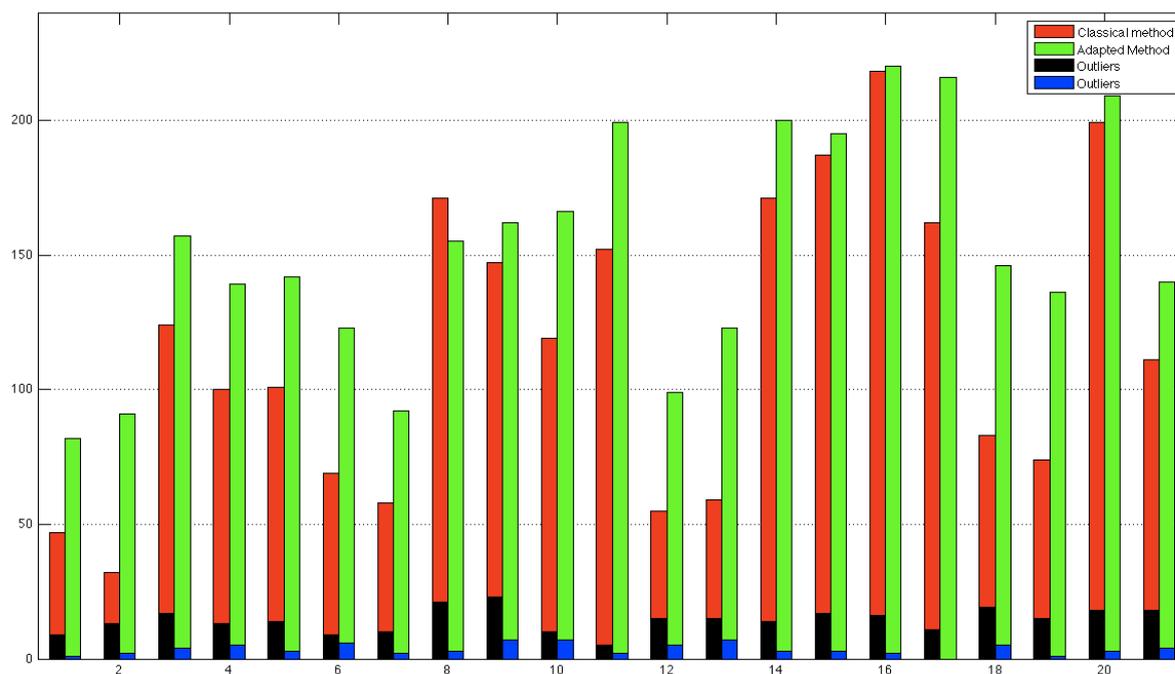


FIGURE 3.9 – Détection de points et mise en correspondance.

3.3 Flot optique

Calculer le flot optique dans une séquence d'images consiste à déterminer le déplacement dans les images des points physiques de la scène. Ce déplacement dit "mouvement apparent" est la projection 2D du déplacement 3D de ces points. Ce problème a été très largement étudié dans la littérature puisqu'on y trouve de nombreuses applications : détection et suivi d'objet en mouvement, navigation de robot, odomètre visuelle. Mais une fois encore, ces méthodes ne sont pas directement applicables aux images sphériques.

Considérons une séquence d'images perspectives $I(x, y, t)$, calculer le flot optique consiste à déterminer la vitesse (v_x, v_y) des pixels (x, y) entre les instants t et $t + 1$. En supposant que la luminosité des points de la scène n'est pas modifiée entre ces deux instants, estimer le flot optique revient à résoudre la désormais bien connue équation de contrainte du flot optique (ECFO) eq.(3.12) :

$$\frac{\partial I}{\partial x} \cdot v_x + \frac{\partial I}{\partial y} \cdot v_y + \frac{\partial I}{\partial t} = 0, \quad (3.12)$$

Cette équation linéaire à deux inconnues (v_x, v_y) ne suffit pas à déterminer le flot optique et aboutit donc au problème dit "de l'ouverture". C'est pourquoi, une autre contrainte est nécessaire pour calculer le flot dans les images. Dès 1981, deux types d'hypothèses ont été considérées. Lucas et Kanade [LT81] supposent le flot constant localement alors que Horn et Schunck [HS81] supposent le flot lisse sur l'ensemble de l'image. Depuis, diverses méthodes inspirées de ces travaux ont été développées afin d'améliorer les résultats. Les méthodes dites locales utilisent désormais des bases de fonctions plus optimales que des gaussiennes proposées par la méthode originale de Lucas et Kanade [WM95] [Ber98] [BP02]. Les méthodes dites globales tentent d'affaiblir la contrainte de lissage du flot introduit par Horn et Schunck en remplaçant la norme de Tikhonov par des régularisations en norme L^1 [Coh93] ou par des estimateurs robustes [BA93].

Dans le cas d'images sphériques $I_s(\theta, \varphi, t)$, l'équation (3.12) n'est pas valide et les méthodes classiques doivent être repensées. Si nous supposons la luminosité constante au cours du temps, l'équation de contrainte EFCO devient :

$$\frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} v_\varphi + \frac{\partial I_s}{\partial \theta} v_\theta + \frac{\partial I_s}{\partial t} = 0 \quad (3.13)$$

C'est cette équation qu'il faut désormais résoudre sur l'image sphérique. Pour ce faire, Daniilidis et al. [DMB02] adaptent la méthode de Lucas et Kanade sur la sphère en supposant le flot constant au voisinage du point courant et en pondérant par une gaussienne sphérique. Par la suite, Tomic et al. [TBFV05] proposent un schéma multirésolution pour surmonter les problèmes inhérents au calcul du flot optique à savoir, l'aliasing temporel. Les méthodes globales ont elles aussi fait l'objet d'adaptation sur les images sphériques. Par exemple, Mochizuki et al. [MI08] proposent une formulation variationnelle à l'image de ce que Horn et Schunck proposaient avec un terme de lissage de Tikhonov sur la sphère. Comme dans le cas des images perspectives, pour éviter un trop fort lissage de l'estimation du flot, la norme L^2 de Tikhonov peut être remplacée par une variation totale L^1 [BFV09].

Dans le cas d'images perspectives, les méthodes d'estimation du flot optique par projection de l'ECFO sur une base de fonction ont montré une très bonne efficacité et un bon compromis entre rapidité d'exécution, précision et densité de l'estimation [Ber98] [BP02] [WM95]. C'est pourquoi, nous avons proposé une adaptation de ces méthodes aux images sphériques que nous allons préciser dans la suite.

3.3.1 Ondelettes sphériques

Pour une étude approfondie de l'analyse en ondelettes sphériques, on se reportera aux travaux de Vandergheynst et Antoine [AV99]. Nous illustrons ici leur intérêt pour l'estimation du flot

sphérique par analogie à ce qui a été développé dans le plan euclidien [Ber98].

Dans ce travail, nous avons choisi les ondelettes sphériques de Morlet. Ces ondelettes sont en fait une adaptation des ondelettes de Gabor aux signaux sphériques qui ont elles mêmes démontré leur efficacité pour le calcul du flot optique en imagerie perspective [BP02]. Ces ondelettes sont définies à partir de la projection stéréographique des ondelettes de Gabor ψ_M définies sur le plan par :

$$\psi_M(x) = e^{ik_0x} e^{-|x|^2}$$

et deviennent sur la sphère :

$$\Psi_M(\theta, \varphi) = \frac{e^{i|k_0| \tan(\theta/2) \cos(\varphi - \varphi_0)} e^{-1/2 \tan^2(\theta/2)}}{1 + \cos(\theta)} \quad (3.14)$$

où φ_0 est la direction de l'ondelette et $|k_0| \geq 6$ pour qu'elle soit admissible ([ADJV02]). La figure (3.10) présente une ondelette de Morlet sphérique avec deux directions différentes.

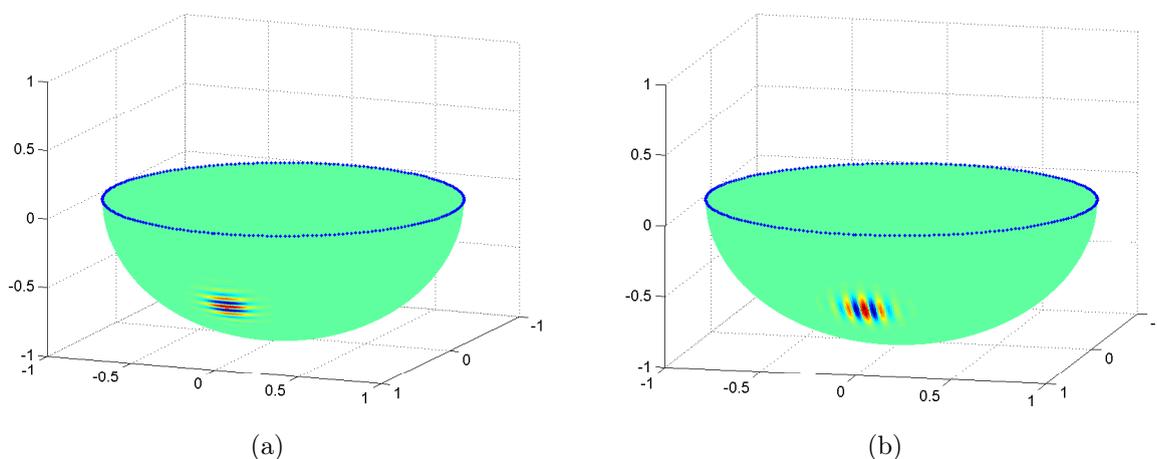


FIGURE 3.10 – Ondelette de Morlet sphérique, (a) $\varphi_0 = 0$ (b) $\varphi_0 = \pi/2$

Sur le plan, pour appliquer la transformée en ondelette à une fonction de L^2 , il faut projeter cette fonction sur une famille d'ondelettes dilatées et translatées. Par analogie, sur la sphère, les translations sur le plan sont remplacées par des rotations de $SO(3)$. Ainsi, les coefficients d'ondelettes d'une image I_s de S^2 sont définis par :

$$W_{\Psi} I_s(\eta) = \langle I_s, \Psi(g^{-1}\eta) \rangle = \int_{S^2} I_s(\eta) \overline{\Psi(g^{-1}\eta)} d\eta, \quad g \in SO(3) \quad (3.15)$$

où : $d\eta = \sin \theta d\theta d\varphi$.

En pratique, ces transformées sont calculées grâce à la Toolbox YAWTB².

2. <http://sites.uclouvain.be/ispgroup/yawtb/>

3.3.2 Estimation du flot optique sphérique

Pour surmonter le problème de l'ouverture, il suffit donc de projeter l'équation (3.13) sur une base d'ondelettes sphériques. En effet, chaque ondelette de la base permet d'ajouter une équation et transforme ainsi le problème sous-déterminé en un problème sur-déterminé. Considérons donc une famille d'ondelettes $\Psi_i(\theta, \varphi)$ (où $i=1..N$), l'équation (3.13) est vérifiée si :

$$\langle \Psi_i(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} v_\varphi + \frac{\partial I_s}{\partial \theta} v_\theta + \frac{\partial I_s}{\partial t} \rangle = 0, \quad \forall i = 1 \dots M \quad (3.16)$$

A l'image de la méthode de Lucas et Kanade, pour résoudre ce système, nous devons supposer le flot constant sur le support des ondelettes sphériques. Cette hypothèse permet d'aboutir au système d'équations linéaires suivant :

$$\langle \Psi_i(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} \rangle v_\varphi + \langle \Psi_i(\theta, \varphi), \frac{\partial I_s}{\partial \theta} \rangle v_\theta + \langle \Psi_i(\theta, \varphi), \frac{\partial I_s}{\partial t} \rangle = 0 \quad (3.17)$$

Remarquons de plus que les ondelettes utilisées (ondelettes de Morlet) sont complexes. Par conséquent, nous avons à disposition $2 * N$ équations et deux inconnues :

$$\begin{pmatrix} \text{Re}(\langle \Psi_1(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} \rangle) & \text{Re}(\langle \Psi_1(\theta, \varphi), \frac{\partial I_s}{\partial \theta} \rangle) \\ \text{Im}(\langle \Psi_1(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} \rangle) & \text{Im}(\langle \Psi_1(\theta, \varphi), \frac{\partial I_s}{\partial \theta} \rangle) \\ \dots & \dots \\ \text{Re}(\langle \Psi_N(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} \rangle) & \text{Re}(\langle \Psi_N(\theta, \varphi), \frac{\partial I_s}{\partial \theta} \rangle) \\ \text{Im}(\langle \Psi_N(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} \rangle) & \text{Im}(\langle \Psi_N(\theta, \varphi), \frac{\partial I_s}{\partial \theta} \rangle) \end{pmatrix} \begin{pmatrix} v_\varphi \\ v_\theta \end{pmatrix} = \begin{pmatrix} -\text{Re}(\langle \Psi_1(\theta, \varphi), \frac{\partial I_s}{\partial t} \rangle) \\ -\text{Im}(\langle \Psi_1(\theta, \varphi), \frac{\partial I_s}{\partial t} \rangle) \\ \dots \\ -\text{Re}(\langle \Psi_N(\theta, \varphi), \frac{\partial I_s}{\partial t} \rangle) \\ -\text{Im}(\langle \Psi_N(\theta, \varphi), \frac{\partial I_s}{\partial t} \rangle) \end{pmatrix} \quad (3.18)$$

Ainsi, déterminer le flot optique en (θ, φ) revient à minimiser la fonctionnelle linéaire suivante :

$$\vec{V} = \arg \min_{v_\theta, v_\varphi} \sum_{i=1}^N [\langle \Psi_i(\theta, \varphi), \frac{1}{\sin \theta} \frac{\partial I_s}{\partial \varphi} \rangle v_\varphi + \langle \Psi_i(\theta, \varphi), \frac{\partial I_s}{\partial \theta} \rangle v_\theta + \langle \Psi_i(\theta, \varphi), \frac{\partial I_s}{\partial t} \rangle]^2 \quad (3.19)$$

En pratique, cette minimisation est effectuée au sens des moindres carrés à l'aide de deux seuils. Un premier vérifie si le système sur-déterminé est bien conditionné et le second vérifie si la solution déterminée est conforme au système c'est-à-dire que l'équation (3.19) aboutit à une valeur proche de zéro. Ces seuils sont en fait des critères permettant de valider la contrainte de constance du flot sur le support des fonctions.

3.3.3 Résultats

Nous présentons ici les résultats obtenus à partir de notre méthode. La méthode que nous avons développée se rapproche fortement de celle proposée par Bruno [BP02] dans le cas des

images perspectives. Nous proposons donc dans un premier temps de comparer ces deux techniques afin de montrer l'intérêt de passer sur la sphère pour des images catadioptriques. Puis nous verrons que notre méthode reste performante même par rapport à des méthodes adaptées aux images sphériques.

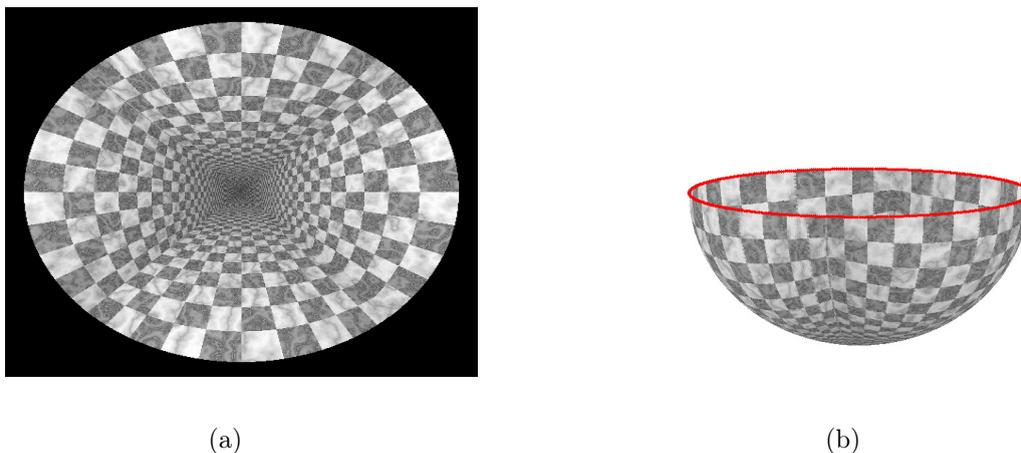


FIGURE 3.11 – (a) Image paracatadioptrique de synthèse, (b) image sphérique équivalente.

Pour ce faire, nous avons estimé le flot sur des séquences d'images de synthèse générées par POV-Ray³ pour lesquelles les champs de déplacement sont connus. Ces séquences correspondent à une caméra paracatadioptrique positionnée dans une pièce composée de quatre murs plans texturés (Fig.(3.11)). Pour comparer cette méthode aux méthodes usuelles, nous allons projeter nos champs de vecteurs sphériques (v_θ, v_φ) directement sur les images paracatadioptriques à l'aide de la relation suivante :

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{-\cos \varphi}{2 \sin^2 \frac{\theta}{2}} & -\cot \frac{\theta}{2} \sin \varphi \\ \frac{-\sin \varphi}{2 \sin^2 \frac{\theta}{2}} & \cot \frac{\theta}{2} \cos \varphi \end{bmatrix} \begin{bmatrix} v_\theta \\ v_\varphi \end{bmatrix} \quad (3.20)$$

Les résultats sont présentés respectivement figure (3.12), figure (3.13) et figure (3.14) pour des mouvements de translation pure, de rotation pure et des mouvements combinés.

Ces figures montrent que la méthode de Bruno éprouvent des difficultés à calculer le flot à l'extrémité de l'image surtout dans le cas où le déplacement est important. Cette remarque est vérifiée lorsque nous nous intéressons à l'erreur angulaire :

$$e = \arccos\left(\frac{u_e u_r + v_e v_r + 1}{\sqrt{u_r^2 + v_r^2 + 1} \sqrt{u_e^2 + v_e^2 + 1}}\right) \quad (3.21)$$

où $V_e = (u_e, v_e)$ est le champ de vecteur estimé et $V_r = (u_r, v_r)$ le champ réel. La figure (3.15) présente l'erreur angulaire pour différents mouvements de caméra. Une couleur bleue correspond à une erreur nulle et une couleur rouge reflète une erreur importante dans l'estimation du flot. On

3. <http://www.povray.org/>

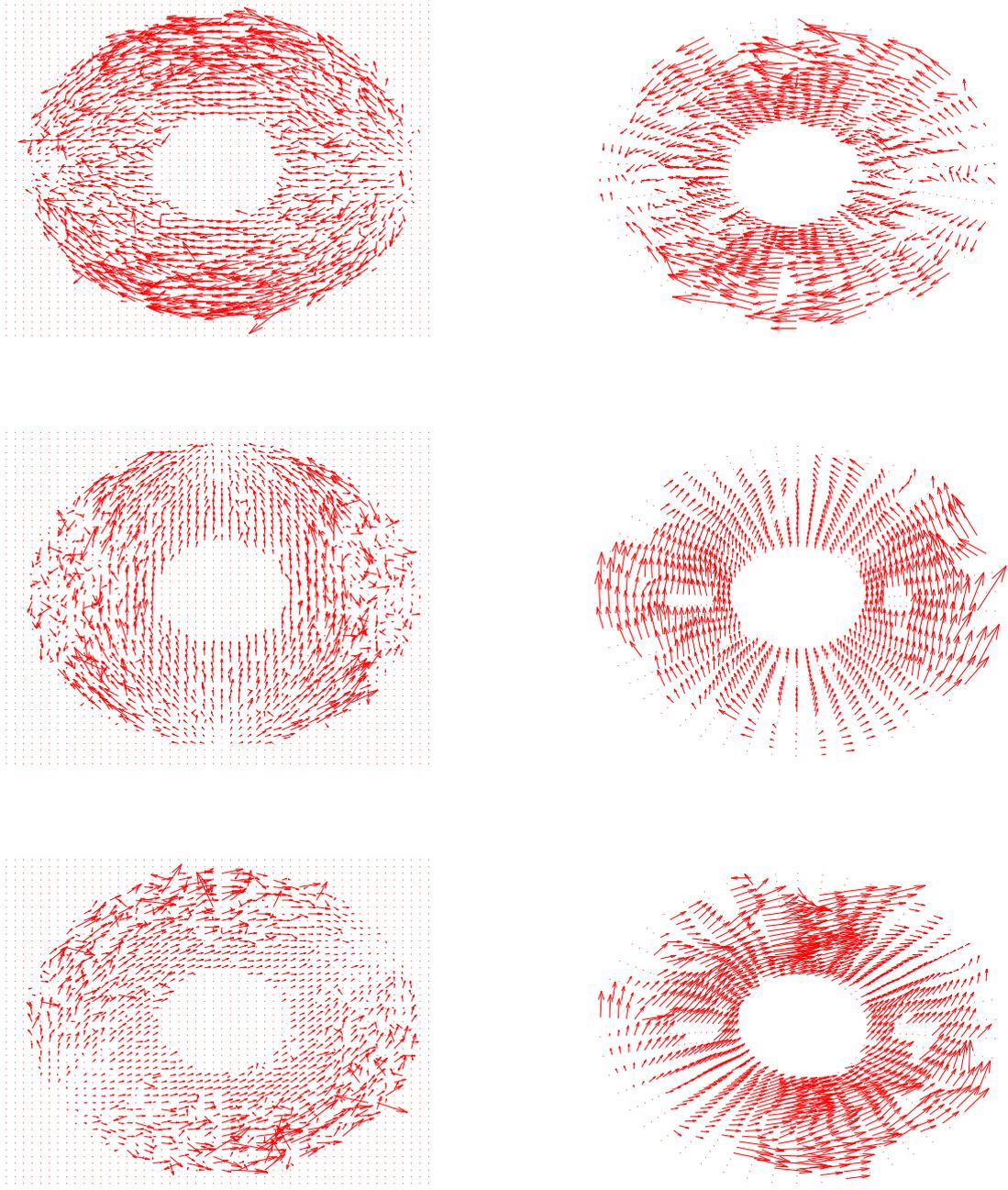


FIGURE 3.12 – Flot optique dans le cas d’une translation pure. La colonne de gauche correspond à la méthode de Bruno et celle de droite à la nôtre. 1^{ère} ligne : $\vec{T} = [-1\text{cm}, 0, 0]$, 2^{ème} ligne $\vec{T} = [0, 3\text{cm}, 0]$ et 3^{ème} ligne $\vec{T} = [5\text{cm}, 3\text{cm}, 0]$.

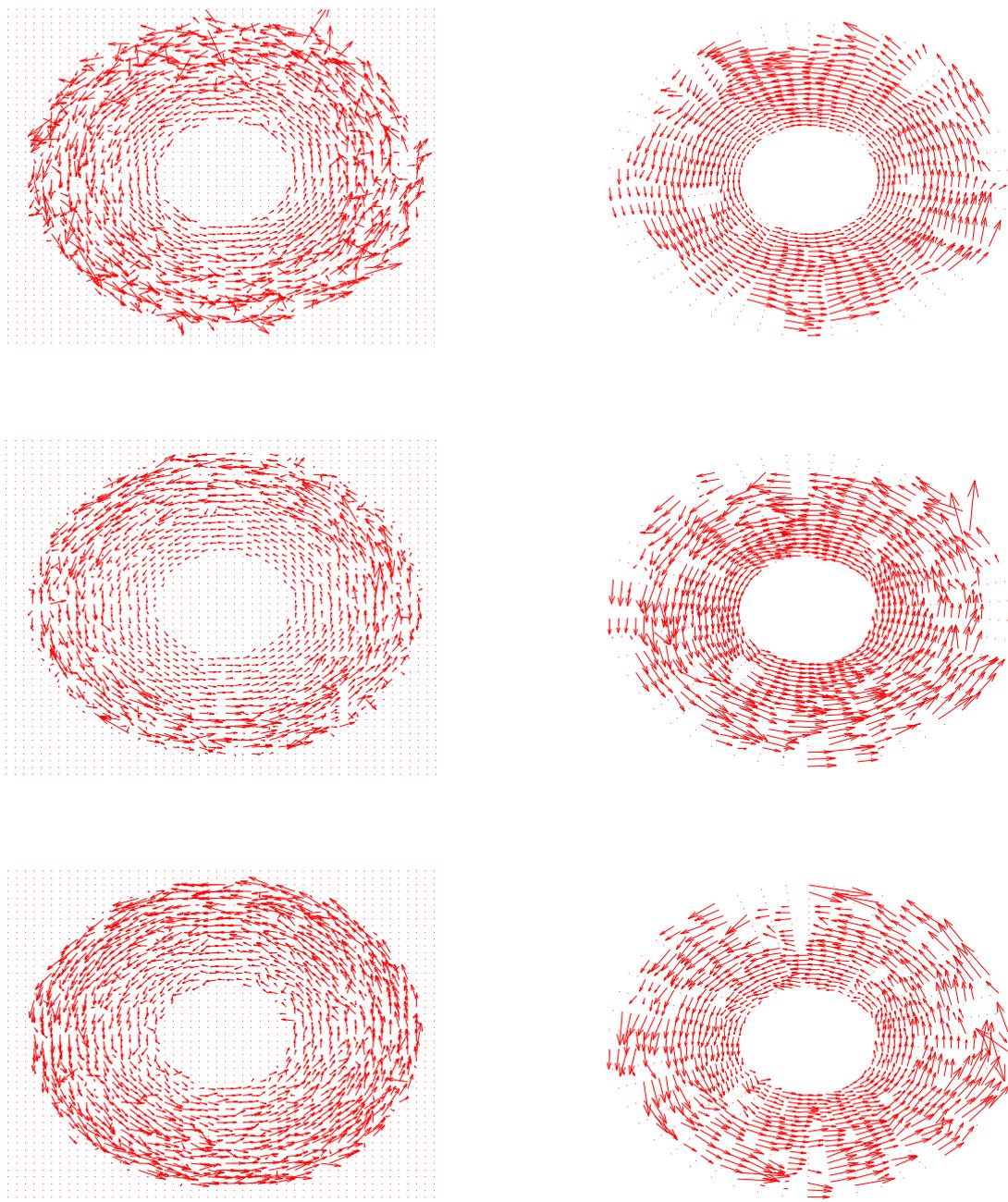


FIGURE 3.13 – Flot optique dans le cas d’une rotation pure. La colonne de gauche correspond à la méthode de Bruno et celle de droite à la nôtre. 1^{ère} ligne $R = [0, 0, 2^\circ]$, 2^{ème} ligne $R = [0, 0, 1^\circ]$ et 3^{ème} ligne $R = [0, 0, 0.5^\circ]$.

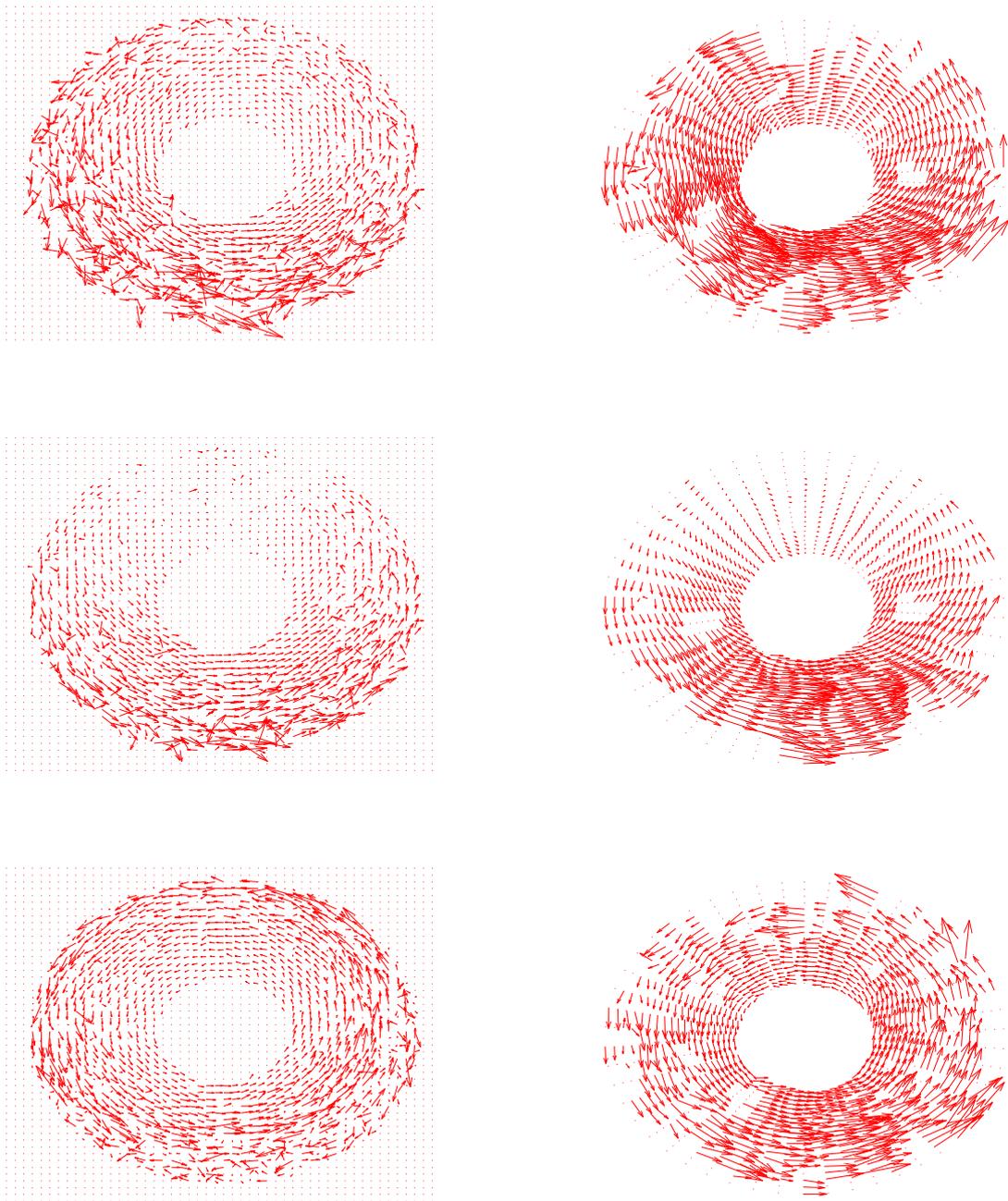


FIGURE 3.14 – Flot optique dans le cas d'un mouvement combiné. La colonne de gauche correspond à la méthode de Bruno et celle de droite à la nôtre. 1^{ère} ligne : $R = [0, 0, 2^\circ]$ et $\vec{T} = [2\text{cm}, -1\text{cm}, 0]$, 2^{ème} ligne $R = [0, 0, 1^\circ]$ et $\vec{T} = [2\text{cm}, 0, 0]$ et 3^{ème} ligne $R = [0, 0, 1^\circ]$ et $\vec{T} = [0.5\text{cm}, 0.2\text{cm}, 0]$.

remarque ici que les erreurs sont plus faibles avec notre méthode adaptée pour ce type d'image et montre donc l'intérêt du passage sur la sphère.

Le tableau 3.2 compare notre approche avec quatre méthodes d'estimation du flot optique pour différents déplacements. La première méthode correspond à l'approche de Lucas et Kanade [LT81] qui reste encore très populaire en robotique mobile. La deuxième approche est celle de Bruno citée plus haut. La troisième proposée par Radgui et al. [RDM⁺08] consiste en une adaptation de la méthode de Lucas et Kanade pour des images catadioptriques sans passer par la sphère d'équivalence. La quatrième proposée par Danilidis et al. [DMB02] est aussi une adaptation de Lucas et Kanade mais où les calculs s'effectuent sur la sphère. Ce tableau démontre une nouvelle fois l'intérêt du passage sur la sphère pour estimer le flot car les méthodes sphériques obtiennent de meilleurs résultats que les méthodes calculant le flot sur le plan image. De plus, il démontre que notre méthode permet d'améliorer l'estimation du flot optique sur ce type d'image surtout dans le cas de grands déplacements.

Séquences		Méthodes classiques		Méthode adaptée sur le plan image	Méthodes adaptées sur la sphère	
		Lucas et al. [LT81]	Bruno et al. [BP02]	Radgui et al. [RDM ⁺ 08]	Danilidis et al. et [DMB02]	Notre approche
Translat.	$\vec{T} = [-1\text{cm}, 0, 0]$	7.99°	8.28°	5.81°	5.33°	4.94°
	$\vec{T} = [0, 3\text{cm}, 0]$	14.11°	16.00°	8.63°	11.90°	4.56°
	$\vec{T} = [5\text{cm}, 3\text{cm}, 0]$	18.62°	14.92°	16.46°	19.49°	4.71°
Rotation	$R = [0, 0, 1^\circ]$	8.60°	10.19°	7.97°	7.59°	5.07°
	$R = [0, 0, 2^\circ]$	16.73°	13.98°	16.00°	12.91°	4.01°
	$R = [0, 0, 0.5^\circ]$	5.61°	7.46°	5.20°	7.50°	7.78°
Mouvement combiné	$R = [0, 0, 0.5^\circ]$ $\vec{T} = [0.3\text{cm}, 0, 0]$	8.97°	7.46°	5.55°	5.73°	5.36°
	$R = [0, 0, 1^\circ]$ $\vec{T} = [2\text{cm}, 0, 0]$	15.39°	11.53°	9.84°	7.71°	4.57°
	$R = [0, 0, 1^\circ]$ $\vec{T} = [0.5\text{cm}, 0, 0]$	11.33°	12.64°	8.00°	7.40°	5.15°
	$R = [0, 0, 1^\circ]$ $\vec{T} = [0.5\text{cm}, 0.2\text{cm}, 0]$	11.35°	12.94°	8.05°	7.28°	5.04°
	$R = [0, 0, 2^\circ]$ $\vec{T} = [2\text{cm}, -1\text{cm}, 0]$	16.07°	14.14°	15.70°	12.25°	5.70°

TABLE 3.2 – Erreur angulaire moyenne de cinq méthodes d'estimation du flot optique sur différentes séquences d'images catadioptriques.

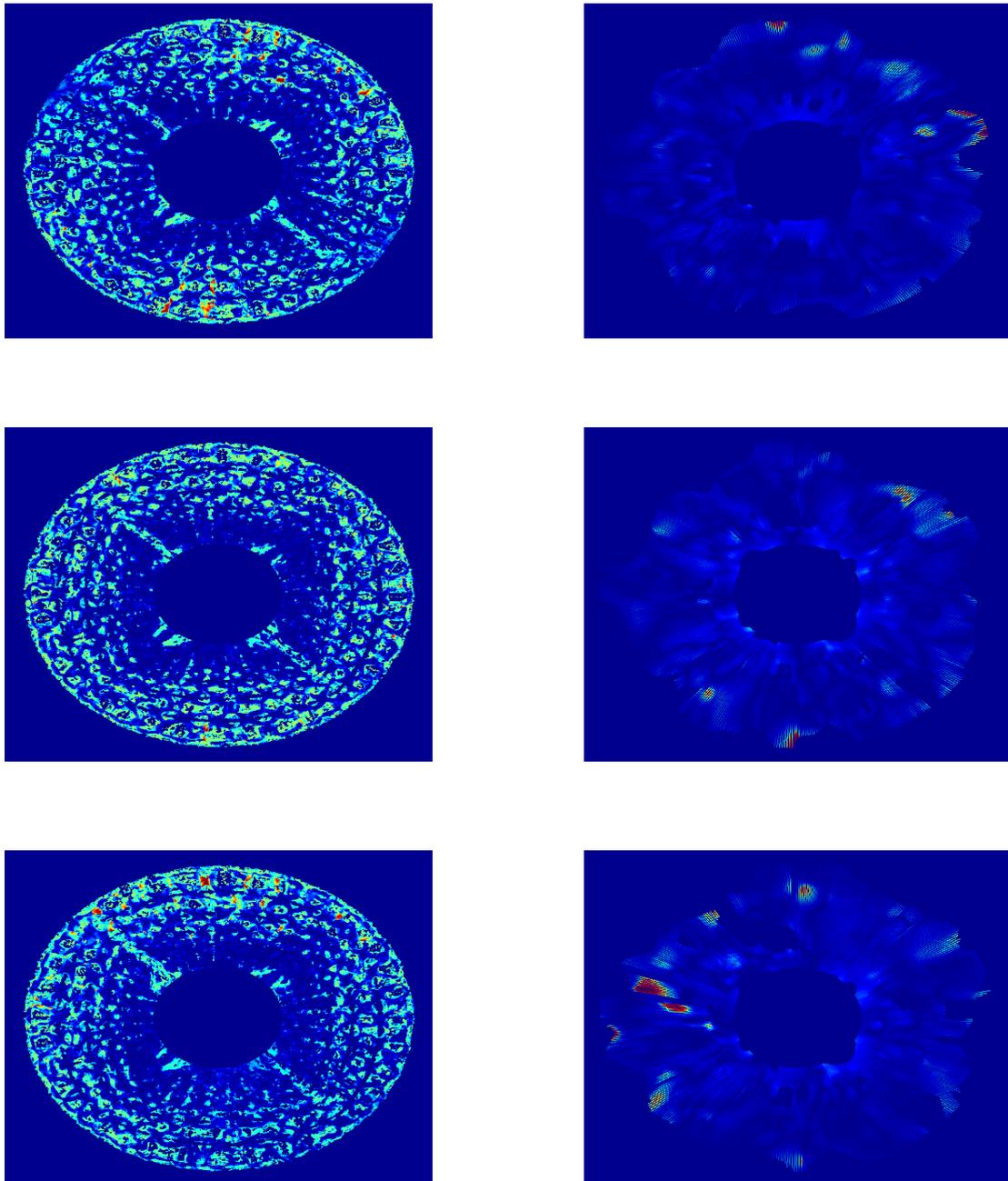


FIGURE 3.15 – Cartes des erreurs angulaires pour différents types de déplacements. Haut : Translation pure en X et Y , milieu : Rotation pure de 1° autour de Z , bas : mouvement combinée. Colonne de gauche : méthode de Bruno. Colonne de droite : notre méthode.

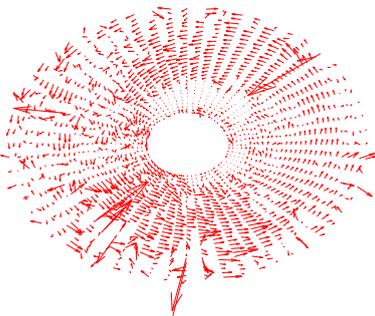
Enfin, nous montrons deux résultats sur séquences réelles acquises à partir d'une caméra catadioptrique calibrée grâce à la toolbox de calibrage de Mei [MR07] et montée sur un robot mobile. Le résultat de la figure (3.16) correspond à un mouvement du robot composé d'une translation suivant l'axe Y et d'une rotation autour de Z. Dans la seconde séquence (Fig. 3.17), le robot est immobile alors qu'un objet de la scène (en haut à droite de l'image) se déplace. Dans les deux cas, on remarque que la méthode de décomposition du flot sur une base d'ondelettes sphériques est plus appropriée qu'une technique classique.



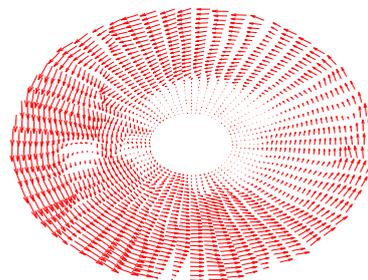
(a)



(b)



(c)



(d)

FIGURE 3.16 – (a), (b) images originales, la caméra se déplace suivant l'axe Y avec une rotation autour de son axe optique Z, (c) méthode de Bruno, (d) méthode adaptée.

3.4 Conclusion

La reconnaissance d'objets, l'estimation du déplacement d'une caméra, la surveillance de scène, la reconstruction 3D etc... nécessitent souvent une phase de calcul de contours, de détection de points d'intérêts, de mise en correspondance. Or, comme nous venons de le voir, ces phases doivent être repensées pour s'adapter au mieux aux caméras grand angle. Nous nous

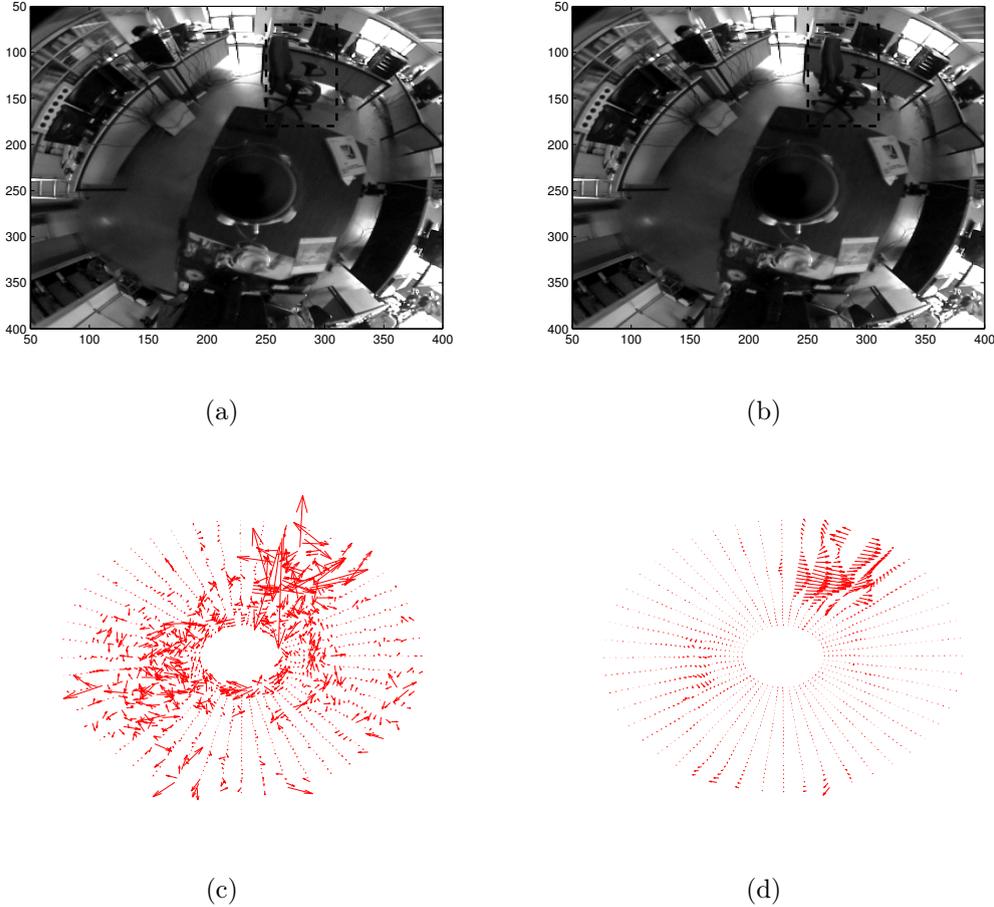


FIGURE 3.17 – (a), (b) images originales, le robot est immobile et un objet se déplace dans la scène, (c) méthode de Bruno, (d) méthode adaptée.

sommes donc efforcés dans ce chapitre à montrer comment nous pouvions adapter les méthodes classiques grâce à l'équivalence sphérique des images grand angle. Ainsi, nous avons vu que modifier la mesure entre les points a permis d'améliorer le détecteur de points d'intérêts de Harris et la mise en correspondance ([DVF11]). Nous avons aussi montré que la sphère pouvait améliorer le calcul du flot optique ([RDM⁺11]).

Ces travaux ont donné lieu à plusieurs publications ainsi qu'à la soutenance d'une thèse en collaboration avec l'Université de Mohammed V de Rabat dans le cadre d'un projet Egide Volubis. Aujourd'hui, ce partenariat se poursuit sous la forme d'une thèse en co-tutelle financée par une bourse d'excellence marocaine et par le programme STIC France-Maroc.

Chapitre 4

Estimation du déplacement de caméras sphériques

Il est possible d'acquérir l'information sur la pose et le mouvement d'un robot à l'aide de différents types de capteurs. Alors que les pertes de signaux sont fréquentes pour les GPS (Global Positioning System) et que les centrales inertielles peuvent accumuler les erreurs relatives au cours du temps, les approches basées vision semblent être un choix judicieux. La plupart des travaux existant utilisent des caméras perspectives. Or ces caméras ont un champ de vue relativement limité qui conduit parfois à des situations ambiguës (figure 4.1). De plus, un champ de vue

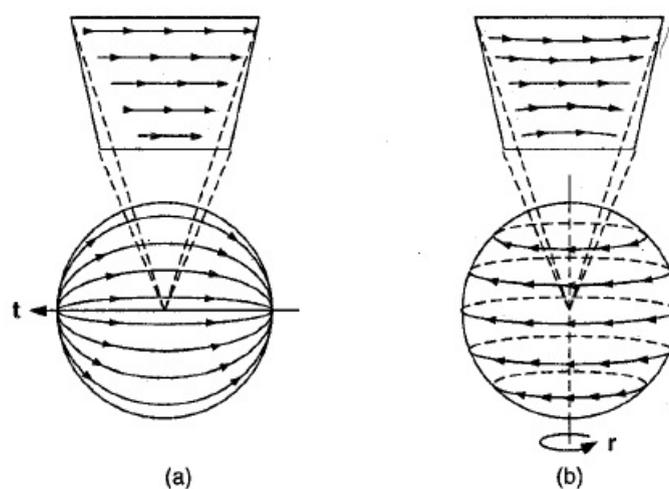


FIGURE 4.1 – Champ de vecteurs lors d'un petit déplacement. (a) petite translation, (b) petite rotation. On remarque que sur le plan perspectif, les champs sont quasi-identiques alors qu'ils sont très différents sur l'image grand angle sphérique associée (image tirée de [GN98]).

important permet de recueillir de nombreuses informations sur l'environnement (figure 4.2). C'est pourquoi un système de vision omnidirectionnelle offre de nombreux avantages.

Pour estimer le déplacement de caméras paramétriques, nous pouvons utiliser des algorithmes totalement génériques définis à partir des lignes de vues [RLS06]. Dans notre cas, comme nos caméras respectent le point de vue unique, cette approche totalement générique ne se révèle pas judicieuse. Une autre possibilité serait de considérer un modèle paramétrique plus générique que le modèle sténopé [SB] mais cette solution nécessite de très nombreux points pour estimer le déplacement. C'est pourquoi nous avons opté une nouvelle fois par une modélisation sphérique de ces images. Cela permet d'avoir un cadre suffisamment générique pour les caméras utilisées sans augmenter le nombre de paramètres à estimer. Bien évidemment, comme l'image sphérique respecte la géométrie projective, les méthodes classiques de SFM (*Structure from motion*) développées pour les caméras *pinholes* sont tout à fait admissibles. Par exemple, l'algorithme des 8 points [Har97], des 5 points [Nis04] sont parfaitement applicables sur nos images (nous verrons d'ailleurs des comparaisons avec ces méthodes par la suite).

Dans nos approches, nous avons choisi de travailler directement sur la sphère et de tenir compte des propriétés et avantages de ce type d'image. En effet, nous verrons que celle-ci facilite la recherche et l'estimation de point de fuite dans la scène [Bar83]. Car quelle que soit la pose de la caméra, les points de fuite existent toujours dans l'image sphérique. De plus, ces points de fuite, étant invariant à la translation entre deux prises de vues, ils permettent de calculer très facilement la rotation. C'est pourquoi nous avons opté pour un découplage entre la rotation et la translation afin d'estimer le déplacement de la caméra. Ainsi, nous estimons la rotation à l'aide des points de fuite pour en déduire la translation entre les prises de vues. Ce travail a été développé à partir des travaux de thèse de Jean-Charles Bazin qui a effectué son doctorat au KAIST (Corée du Sud) sous la direction de InSo Kweon et de Pascal Vasseur ainsi que de Dieu Sang Ly qui a effectué son doctorat au MIS dirigé par Claude Pégard et moi-même. Avec Jean-Charles Bazin, nous proposons d'estimer la rotation à l'aide des droites présentes dans la scène puis la translation à partir de points. Sang Ly a depuis utilisé les droites pour estimer la translation. Notons que ces méthodes nécessitent préalablement la détection des droites dans la scène. Pour ce faire, nous avons proposé une méthode fondée sur une approche d'approximation polygonale mais qui n'est pas évoquée ici. Nous invitons le lecteur intéressé à lire [BDV07].

Dans un premier temps, nous verrons différentes approches que nous avons proposées pour estimer la rotation d'une caméra sphérique. Puis dans une deuxième partie, nous calculerons la translation à partir des points et/ou des droites de la scène. La troisième partie sera consacrée aux algorithmes de reconstruction 3D et d'ajustement de faisceaux. Enfin, une quatrième et dernière partie validera les méthodes sur des expérimentations synthétiques et réelles.

4.1 Estimation de la rotation d'une caméra sphérique

Nous détaillons ici comment nous avons estimé la rotation d'une caméra sphérique à l'aide des points de fuite de la scène. L'ensemble de ce travail est fondée sur la relation entre la direction des points de fuite de la scène et la direction de droites parallèles dans la scène [DVP07] que l'on

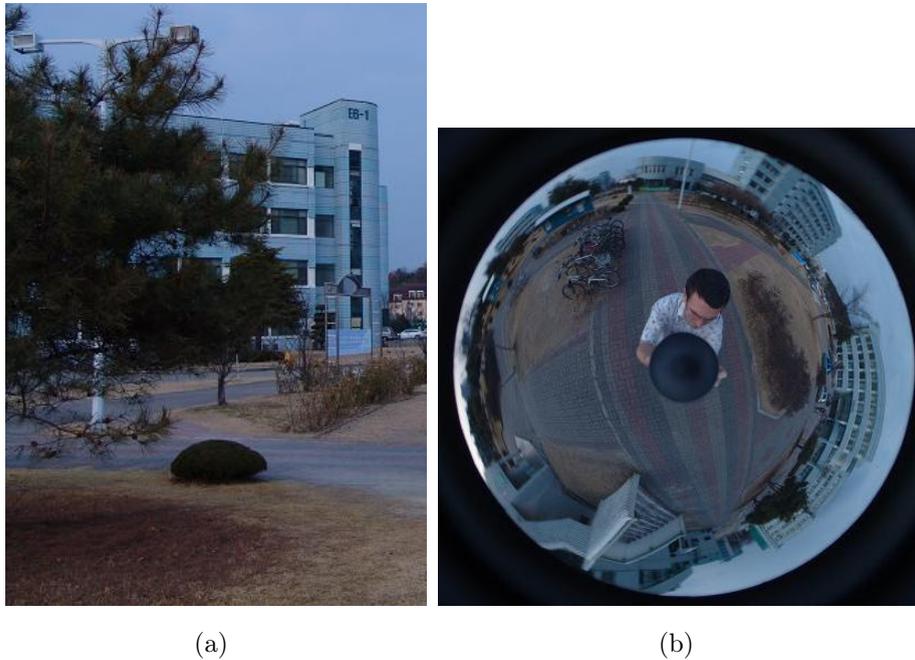


FIGURE 4.2 – Comparaison entre image perspective (a) et image omnidirectionnelle (b) d'une même scène.

peut résumer comme suit :

Proposition :

Considérons deux droites L_1 et L_2 de l'espace. Notons C_1 et C_2 leur projection sur la sphère unitaire, ces deux cercles se coupent en deux points I_1 et I_2 . Si les deux droites ont la même direction 3D notée \vec{u} alors I_1 et I_2 sont deux points antipodaux et $\overrightarrow{I_1 I_2} = \vec{u}$ (Figure 4.3).

Ainsi, à l'aide des points de fuite dans l'image (I_1 et I_2), nous pouvons déduire la direction d'un faisceau de droites parallèles. Si la scène est composée d'au moins deux faisceaux de droites parallèles, nous pouvons obtenir deux directions principales de la caméra relativement à la scène et il est alors facile d'estimer la rotation entre deux prises de vue. Nous avons proposé trois approches pour estimer la rotation à l'aide de cette propriété. La première, très combinatoire, consiste à rechercher les points de fuite comme points qui passent par le plus grand nombre de droites à la manière d'une transformée de Hough ([DVP06]). Une autre approche utilise l'hypothèse selon laquelle les faisceaux de droites parallèles recherchés sont orthogonaux deux à deux. Enfin, une troisième approche consiste à déterminer de façon conjointe les points de fuite dans l'image et les droites associées.

4.1.1 Approche combinatoire

Considérons N droites de vecteur normal respectif n_i , $i = 1 \dots N$ (Figure 4.3). Pour chaque paire de droites ($N(N-1)/2$ combinaisons), nous calculons la direction commune d par $d = n_i \times n_j$, $i > j$. d est en fait une direction potentielle pour un point de fuite. Une droite p peut

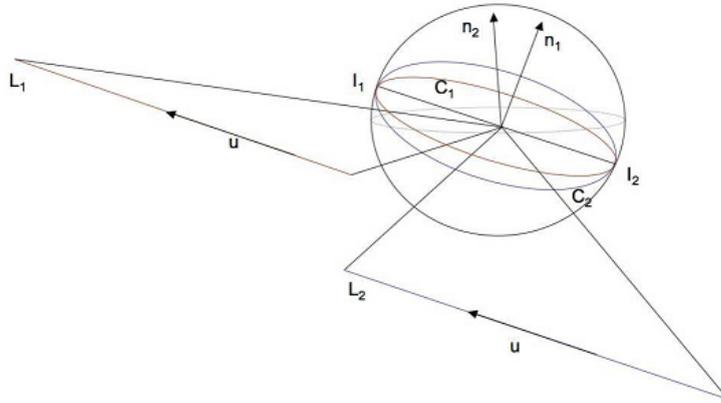


FIGURE 4.3 – Projection de droites parallèles sur la sphère unitaire.

être considérée comme appartenant à un faisceau de droites de direction d si $d^T \cdot n_p = 0$. En pratique, si le produit scalaire $d^T \cdot n_p$ est petit, on considère que la droite p peut appartenir au faisceau et nous incrémentons de 1 la valeur de l'accumulateur $A(d)$. Les directions dominantes d correspondent par la suite au plus grand nombre de votes reçus par $A(d)$. Ces directions sont ensuite mises à jour comme solution de :

$$\hat{d} = \arg \min \sum_{i=1}^P d^T \cdot n_i \quad (4.1)$$

où $n_i, i = 1 \dots P$ sont les vecteurs normaux des droites associées au faisceau de direction \hat{d} .

Par la suite, si nous sommes capables sur deux images consécutives de détecter deux directions dominantes de la scène, nous sommes en mesure de calculer la rotation entre ces images. Notons d_1 respectivement d_2 les deux directions relativement à la position de la première image de la caméra et d'_1 et d'_2 les deux directions dans la seconde image. Comme ces directions sont invariantes à la translation subie par la caméra, nous avons :

$$d'_1 = R d_1 \text{ et } d'_2 = R d_2 \quad (4.2)$$

La rotation R est estimée à l'aide de la décomposition de Rodrigues comme suit :

Proposition :

Considérons une rotation R telle que $d'_1 = R d_1$ et $d'_2 = R d_2$, où $\|d_1\| = \|d_2\| = 1$.

Si nous définissons R par son axe de rotation N et un angle ϕ

$$R u = \cos \phi u + (1 - \cos \phi)(u \cdot N)N + \sin \phi(N \wedge u),$$

alors N et ϕ vérifient :

- si $d'_1 = d_1$ et $d'_2 = d_2$ (cas 1), alors $R = I$, la caméra ne subit pas de rotation.
- si $d'_1 = d_1$ et $d'_2 \neq d_2$ (cas 2), alors

$$N = d_1, \cos \phi = \frac{d'_2 \cdot d_2 - (d_2 \cdot N)^2}{1 - (d_2 \cdot N)^2} \text{ et } \sin \phi = \frac{d'_2 \cdot (N \wedge d_2)}{\|N \wedge d_2\|^2}$$

Par symétrie, nous obtenons une relation semblable en échangeant le rôle de d_1 et d_2 .

– sinon (cas 3), i.e. $d'_1 \neq d_1$ et $d'_2 \neq d_2$, alors

$$N = \frac{(d_1 - d'_1) \wedge (d_2 - d'_2)}{\|(d_1 - d'_1) \wedge (d_2 - d'_2)\|}, \quad \cos \phi = \frac{d'_2 \cdot d_2 - (d_2 \cdot N)^2}{1 - (d_2 \cdot N)^2} \quad \text{et} \quad \sin \phi = \frac{d'_2 \cdot (N \wedge d_2)}{\|N \wedge d_2\|^2}$$

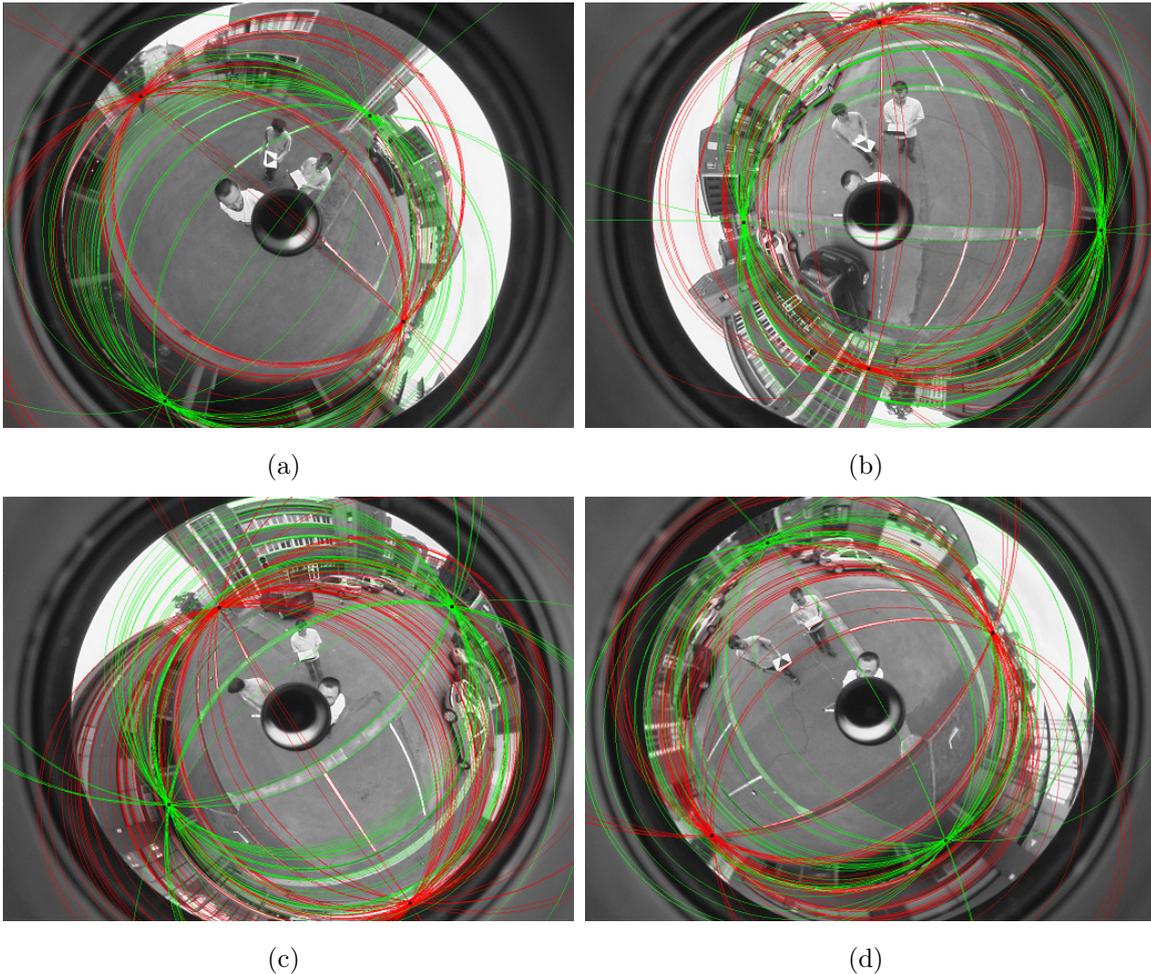


FIGURE 4.4 – Détection des deux points de fuite et des droites parallèles associées.

Les résultats présentés ici correspondent à une séquence de 520 images prises sur le parking de l'université. Pour évaluer nos résultats, nous avons embarqué une centrale inertielle sur la caméra. La séquence correspondait à un parcours d'environ 40 mètres avec des angles de roulis tangage et lacet oscillant respectivement entre $\pm 40^\circ$, $\pm 60^\circ$, $\pm 360^\circ$. La figure 4.4 montre des exemples d'extraction de points de fuite dans la séquence et la figure 4.5 compare l'évolution de l'estimation des angles de rotation avec les données de la centrale. L'erreur moyenne des trois angles est respectivement de 1.2° , 1.3° and 3.9° .

Notons que dans de nombreuses séquences urbaines, les faisceaux de droites sont souvent orthogonaux deux à deux et c'est d'ailleurs le cas sur les résultats présentés. Avec cette méthode, nous obtenons un angle moyen entre la direction des deux faisceaux orthogonaux de 89.7° avec

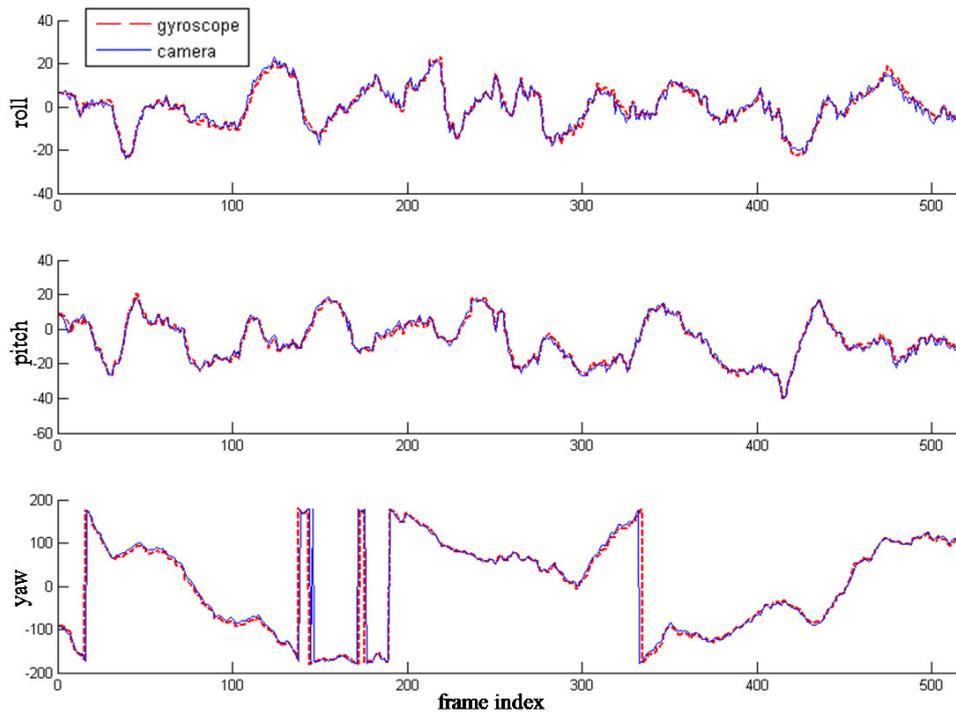


FIGURE 4.5 – Comparaison du roulis (haut), tangage (milieu) et lacet (bas) de notre méthode (courbe bleue) par rapport aux données IMU (courbe rouge).

un écart type de 4.7° . Cette petite erreur peut être évitée si nous tenons compte de la contrainte d'orthogonalité entre les directions des faisceaux directement dans le problème d'optimisation.

4.1.2 Cas des faisceaux orthogonaux

Dans de nombreuses situations, en l'occurrence en milieu urbain, les faisceaux de droites parallèles de la scène sont souvent orthogonaux deux à deux. Ainsi, pour améliorer l'estimation des points de fuite, nous avons proposé d'utiliser cette contrainte en recherchant la rotation entre la base canonique (e_1, e_2, e_3) (position verticale de la caméra) et la base de \mathbb{R}^3 définie par les points de fuite (n_1, n_2, n_3) (figure 4.6) en minimisant :

$$\arg \max_{\alpha, \beta, \gamma} \sum_{i=1}^3 \sum_{j=1}^N \delta(R_{\alpha, \beta, \gamma} \vec{e}_i, \vec{n}_j) \quad (4.3)$$

$$\delta(\vec{u}, \vec{v}) = \begin{cases} 1 & \text{si } |\arccos(\vec{u}, \vec{v}) - \frac{\pi}{2}| \leq \tau \\ 0 & \text{sinon} \end{cases} \quad (4.4)$$

Il s'agit en fait de maximiser le nombre de droites de la scène appartenant à l'un des faisceaux orthogonaux deux à deux. Cette modélisation a l'avantage de détecter directement les droites appartenant aux faisceaux et de calculer dans le même temps la rotation optimale. Cependant, la résolution de (4.3) est un problème délicat à cause de sa forte non linéarité.

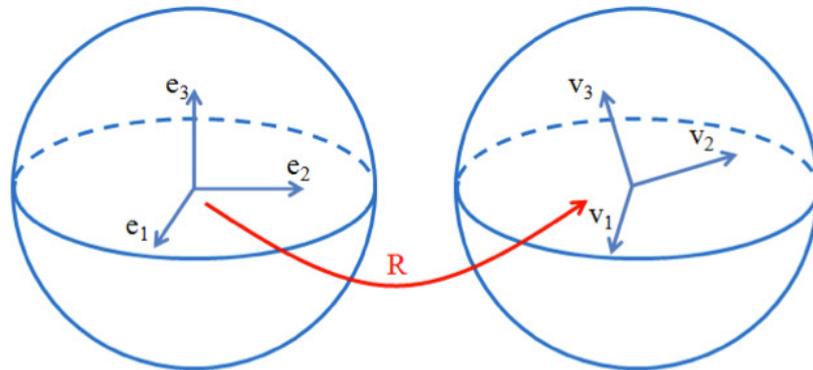


FIGURE 4.6 – Rotation entre la base canonique et les points de fuite.

Approche Multirésolution

Une solution consiste à résoudre le problème à l'aide d'un algorithme multirésolution en échantillonnant grossièrement dans un premier temps l'espace des solutions possibles puis en affinant autour des meilleures solutions à chaque échelle comme présenté dans l'algorithme 1.

Nous avons utilisé cette approche dans [BDVK12]. Les résultats présentés ici (Fig. 4.7,4.8,4.9) sur des séquences prises par Google Street View où nous avons supposé un déplacement non holonome à vitesse constante montrent que nous sommes capables de déterminer avec une bonne précision le déplacement du véhicule. A titre de comparaison, la figure 4.10 présente le résultat obtenu avec la méthode proposée par Mouragnon et al. [MLD⁺06] basée sur l'algorithme du 3 points [NS06]).

Approche Séparation et Evaluation : Branch and Bound

Les méthodes de séparation et évaluation dite "*Branch and Bound*" (BnB) développée en recherche opérationnelle ont été récemment popularisées dans la communauté de vision par ordinateur par Hartley et Kahl [HK09]. Ces approches permettent d'obtenir une solution globalement optimale à (4.3). L'idée de base des BnB est de diviser l'espace de recherche en petits intervalles et de supprimer ceux qui ne peuvent contenir une solution meilleure que celle déjà trouvée. Les méthodes BnB ont récemment été appliquées à des tâches diverses comme l'estimation de mouvement pour des caméras à champ de vue non recouvrant [KLH10] ou pour la calibration d'une caméra rotative [CAKB10]. Dès que l'espace de recherche des solutions est un espace ordonné, la partie séparation n'est pas très compliquée. Il suffit en fait d'échantillonner l'espace et d'affiner cet échantillonnage au fur et à mesure des itérations. Par contre, la phase d'évaluation, c'est-à-dire le principe permettant de conclure s'il faut ou pas scruter l'espace courant, est souvent très délicate à mettre en oeuvre. Dans notre cas, afin de déterminer une borne supérieure de la fonction de coût (4.3), nous avons utilisé l'analyse par intervalle [AM00]. Cette approche permet de vérifier si un intervalle $[\underline{\alpha}, \bar{\alpha}]$, $[\underline{\beta}, \bar{\beta}]$, $[\underline{\gamma}, \bar{\gamma}]$ est susceptible de proposer une solution meilleure que la solution courante. Ce principe est présenté dans l'algorithme 2.

Algorithm 1 Estimation de la rotation avec contrainte d'orthogonalité

{**Entrée** : ListN : Liste des normales dans l'image courante}

{**Entrée** : $(\alpha_{est}, \beta_{est}, \gamma_{est})$: estimation a priori}

$D_{original}$ = offset

K = pas d'échantillonnage

T_{goal} = précision

$J = 0$ niveau de précision

while $J == 0$ ou $T > T_{goal}$ **do**

 MaxValue == 0

$J = J + 1$

$D = D_{original} * (2/K)^{J-1}$

$T = D_{original} * (2/K)^J$

for $\alpha = \alpha_{est} - D : T : \alpha_{est} + D$ **do**

for $\beta = \beta_{est} - D : T : \beta_{est} + D$ **do**

for $\gamma = \gamma_{est} - D : T : \gamma_{est} + D$ **do**

$Value = \sum_{i=1}^M \sum_{j=1}^N \delta(R_{\alpha, \beta, \gamma} \vec{e}_i, ListN_j)$

if $Value > MaxValue$ **then**

 MaxValue = Value

$(\alpha_{max}, \beta_{max}, \gamma_{max}) = (\alpha, \beta, \gamma)$

end if

end for

end for

end for

$(\alpha_{est}, \beta_{est}, \gamma_{est}) = (\alpha_{max}, \beta_{max}, \gamma_{max})$

end while

Return $(\alpha_{est}, \beta_{est}, \gamma_{est})$

Algorithm 2 Recherche des bornes par analyse d'intervalle

{**Entrée** : intervalle de rotation $I = \{[\underline{\alpha}, \overline{\beta}], [\underline{\alpha}, \overline{\beta}], [\underline{\gamma}, \overline{\gamma}]\}$ }

{Calculer la matrice de rotation de l'intervalle $I : R^I$ }

{Calculer les 3 VP v_j^I de R^I , $j = 1, 2, 3$ }

for chaque v_j **do**

for chaque n_i **do**

$$c_{ij} = |n_i^T v_j| = [\underline{c}_{ij}, \overline{c}_{ij}]$$

if $\overline{c}_{ij} < T$ **then**

$z_{i,j} = [1, 1]$ (c'est forcément un inlier)

else

if $\underline{c}_{ij} \geq T$ **then**

$z_{i,j} = [0, 0]$ (c'est forcément un outlier)

else

$z_{i,j} = [0, 1]$ (on ne peut rien conclure)

end if

end if

end for

end for

Return $Z^I = \sum_i \sum_j z_{ij} = [\underline{Z}^I, \overline{Z}^I]$

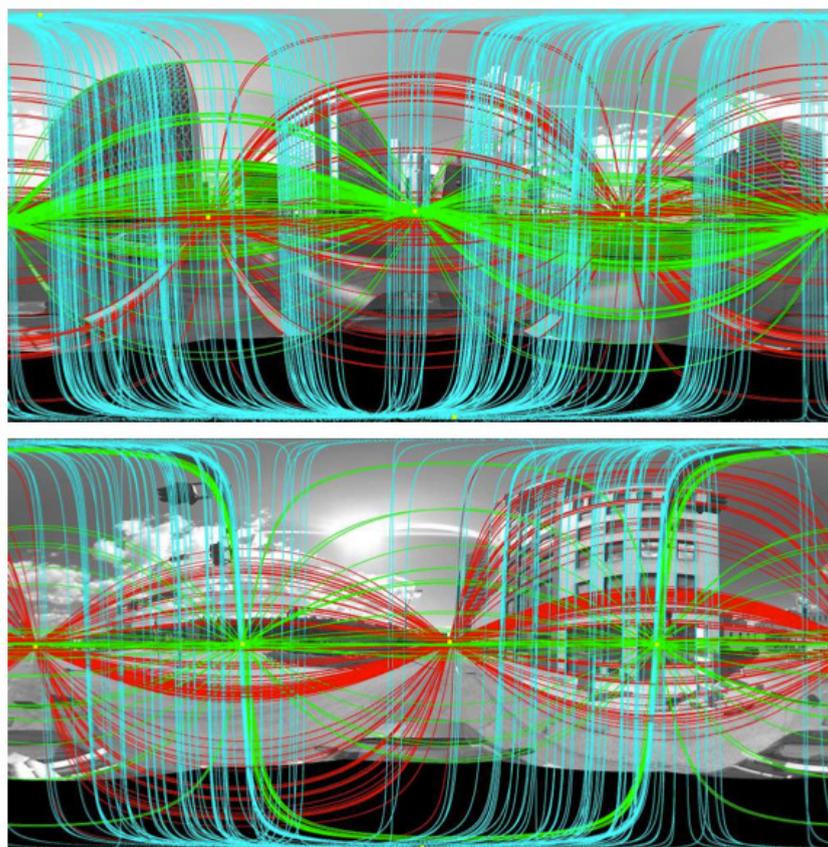


FIGURE 4.7 – Détection des faisceaux de droites parallèles et des points de fuites sur la séquence Google Street.

Ainsi, chaque intervalle est caractérisé par son nombre maximal et minimal de droites susceptibles d'appartenir à un faisceau. Les ensembles dont le nombre maximal de droites est inférieur à la plus grande borne inférieure ne sont pas pris en compte. Les ensembles restant sont alors sous-échantillonnés jusqu'à convergence (fig. 4.11).

Pour vérifier le comportement de la méthode, nous avons synthétisé une scène contenant deux ou trois groupes orthogonaux de 5 à 100 lignes parallèles dont les normales ont été bruitées par un bruit gaussien de 3° et où quelques lignes aberrantes ont été ajoutées.

La figure 4.12 compare notre méthode avec cinq méthodes proposées dans la littérature. Deux méthodes RANSAC notées respectivement Seq-Ransac [SSS⁺08] pour laquelle la recherche s'effectue de manière séquentielle et Sim-Ransac où l'on impose l'orthogonalité des faisceaux, les méthodes fondées sur la transformée de Hough [QM89] (avec un échantillonnage plus ou moins précis (Hough-100, Hough-200, Hough-300)), l'algorithme EM [AT00] et enfin la méthode présentée en section 4.1.1. On remarque que Sim-Ransac est plus performant que le Seq-Ransac en moyenne car il impose intrinsèquement l'orthogonalité. La recherche quasi-exhaustive fonctionne très bien et EM parvient à améliorer les résultats de Hough-100. Notre méthode obtient 32 *inliers*, ce qui correspond exactement au nombre d'*inliers* de la scène. Nous avons aussi testé notre



FIGURE 4.8 – Résultat du déplacement avec contraintes de non holonomie.

approche sur les images York urbain [DEE08] dont les faisceaux de droites ont été obtenus manuellement. Cette base de données se compose de 102 images perspectives. Chaque image dans la base de données contient deux ou trois points de fuite. A partir de l'ensemble des lignes extraites, nous appliquons notre algorithme pour trouver les faisceaux puis nous les avons comparés aux résultats réels. Pour chacune des images de la base, nous avons réussi à obtenir les bons faisceaux, ce qui démontre le bon comportement de notre méthode. La figure 4.13 montre deux résultats obtenus par notre méthode.

4.2 Estimation de la translation

Dans cette section, nous supposons désormais que nous sommes capables de déterminer la rotation relative entre deux positions de la caméra. L'estimation du déplacement se résume donc au calcul d'une translation entre ces deux prises de vue. Durant nos travaux, nous avons développé des méthodes permettant d'estimer cette rotation à partir des points et/ou des droites de la scène. Elles sont rappelées par la suite.

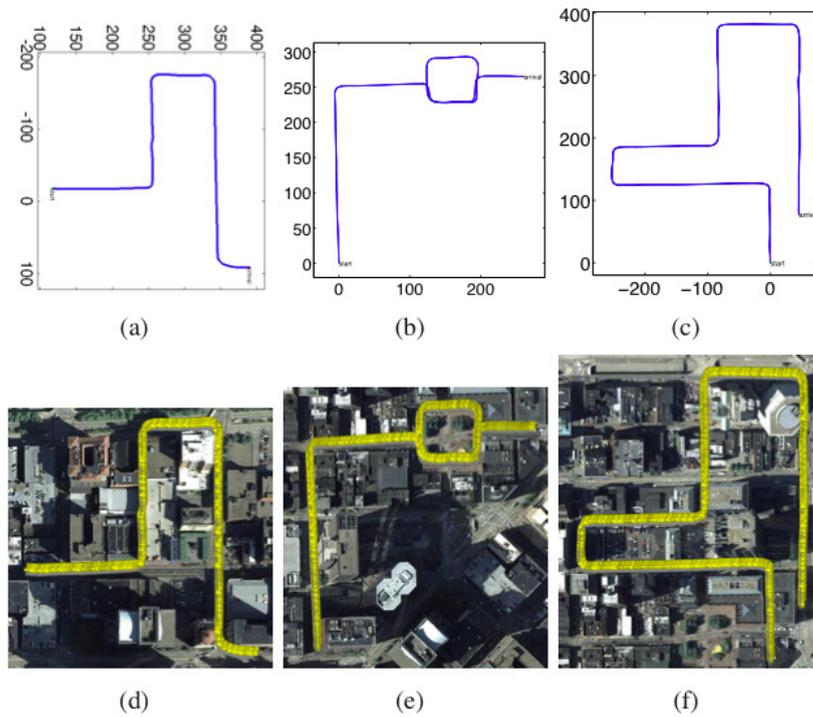


FIGURE 4.9 – Résultat du déplacement avec contraintes de non holonomie.

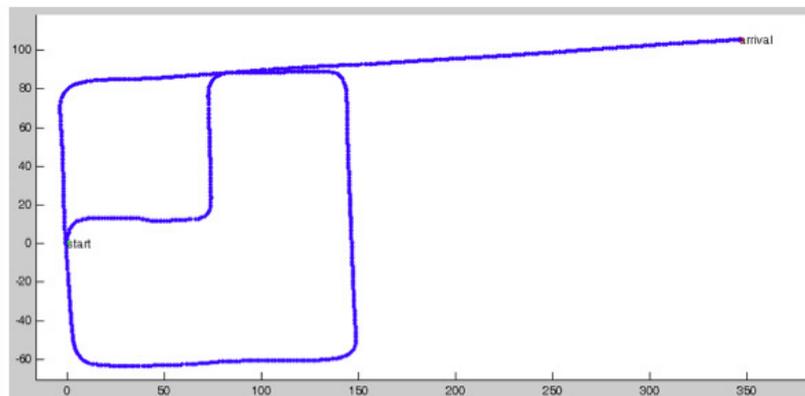


FIGURE 4.10 – Estimation du déplacement de la séquence GoogleStreet View obtenue par [MLD⁺06] basée sur l’algorithme du 3 points [NS06].

4.2.1 A partir de points : Algorithme des 2 points

Supposons donc la rotation R_2 connue entre les deux vues, nous souhaitons calculer la translation \mathbf{t} . Pour ce faire, considérons un point 3D P^i et notons \mathbf{p}_1^i (respectivement \mathbf{p}_2^i) la projection de ce point dans la première prise de vue (respectivement dans la seconde). A partir de la géométrie

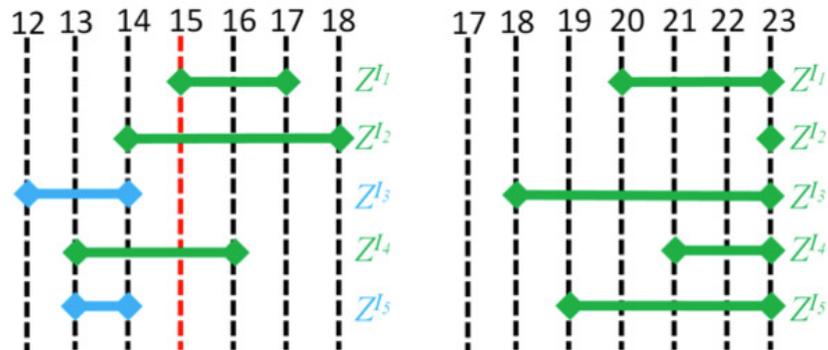


FIGURE 4.11 – Gestion de la liste intervalle. Gauche : On ne considère pas I_3 et I_5 intervalles car leur borne supérieure est inférieure à la borne inférieure de I_1 . A droite : l’algorithme se termine parce que la borne supérieure de l’intervalle de I_2 est égale à sa borne inférieure et que cette valeur est la valeur maximale possible.

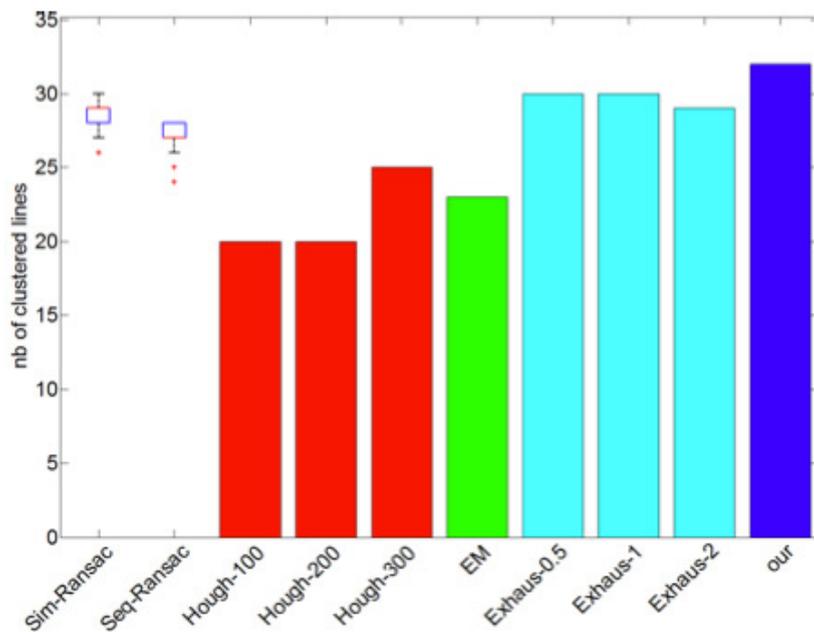


FIGURE 4.12 – Comparaison du nombre d’inliers détectés avec l’approche BnB par rapport à la littérature

épipolaire, nous avons :

$$(\mathbf{R}_2 \mathbf{p}_1^i \times \mathbf{p}_2^i)^\top \mathbf{t} = 0 \quad (4.5)$$

Ainsi chaque correspondance fournit une équation et par conséquent deux correspondances suffisent pour estimer \mathbf{t} à l’échelle :

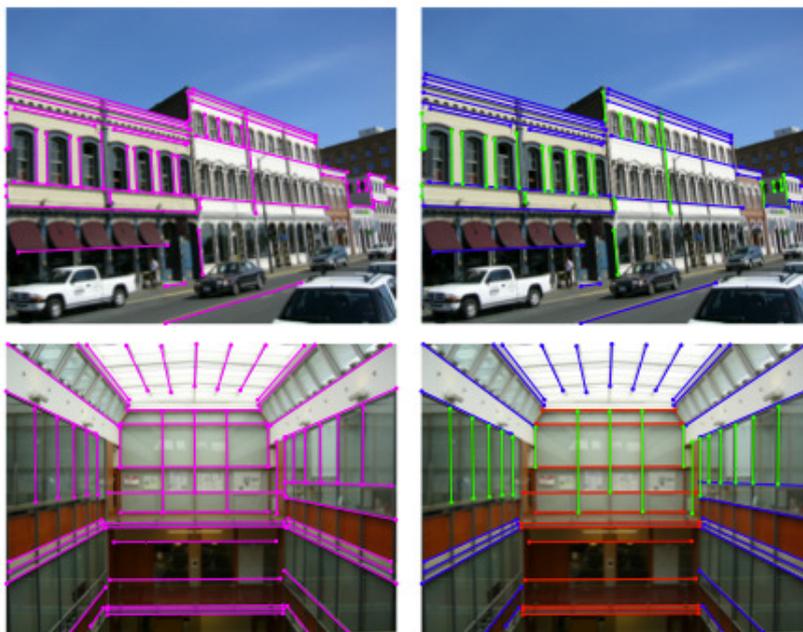


FIGURE 4.13 – Résultat de l’approche BnB sur des données réelles. gauche : droites détectées, droite : faisceaux de droites

$$\mathbf{t} \sim (\mathbb{R}_2 \mathbf{p}_1^1 \times \mathbf{p}_2^1) \times (\mathbb{R}_2 \mathbf{p}_1^2 \times \mathbf{p}_2^2). \quad (4.6)$$

En pratique, comme nous disposons souvent de plus de deux points en correspondance, il est préférable d’estimer la translation avec l’ensemble de points à notre disposition pour obtenir une meilleure précision et réduire ainsi la sensibilité aux bruits. C’est pourquoi nous calculons la translation dans une procédure type RANSAC [FB81]. Notons par contre que comme l’estimation de la translation ne nécessite que deux points, l’algorithme du RANSAC ne complexifie que très peu la procédure comme nous pouvons l’observer dans le tableau 4.1. Ce tableau 4.1 montre clairement que l’algorithme des 2 points nécessite beaucoup moins d’itérations que les algorithmes de 4, 5, 7 ou 8 points, en particulier lorsque le pourcentage d’*outliers* est élevé. Concrètement, pour 50 % de valeurs aberrantes, le nombre d’itérations pour l’algorithme des 2 points avec RANSAC est respectivement de 4, 9, 35 et 69 fois inférieur à celui des algorithmes du 4, 5, 7, 8 points.

4.2.2 A partir de droites

L’estimation de la rotation requiert la détection des droites de la scène, il est donc intéressant de voir comment ces primitives peuvent aussi être utilisées pour estimer la translation entre les deux vues. Les problématiques de la géométrie multi-vues de droites ont été largement étudiées [FM95, HZ04, MSKS05]. Nous rappelons dans cette section la géométrie multi-vues de droites

Taille de l'échantillon	Nombre d' <i>outliers</i> ϵ						
	5%	10%	20%	25%	30%	40%	50%
s							
2	2	3	5	6	7	11	17
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

TABLE 4.1 – Nombre d'échantillons nécessaires pour assurer (avec une probabilité $p = 0,99$) qu'au moins un échantillon n'a aucune valeur aberrante pour un nombre donné s de points minimums et une proportion des valeurs aberrantes ϵ [HZ04].

développée par Torii et al. [TIO05] pour des caméras sphériques et développons une méthode linéaire pour estimer les translations à partir des rotations connues et des correspondances de droites à travers au moins trois vues.

Supposons que la scène est composée de m caméras à point de vue unique \mathbf{C}_i ($i = 1 \dots m$ avec $m \geq 3$) comme illustré dans la figure 4.14. Considérons le centre de la première caméra comme origine du repère monde. Notons $[\mathbf{R}_i | \mathbf{t}_i]$ la [Rotation|translation] entre \mathbf{C}_i et l'origine, ainsi $[\mathbf{R}_1 | \mathbf{t}_1] = [\mathbf{I} | \mathbf{0}]$. La projection sur les images sphériques d'une droite \mathbf{L} dans l'espace 3D est un grand cercle \mathbf{l}_i . Nous pouvons exprimer \mathbf{L} vectoriellement par $\mathbf{L} = \mathbf{X}_0 + \mu \mathbf{d}$ où $\mathbf{L}, \mathbf{X}_0, \mathbf{d} \in \mathbb{R}^3$ et $\mu \in \mathbb{R}$. Enfin, nous notons $\mathbf{n}_i \in \mathbb{R}^3$ les normales aux plans passant la droite \mathbf{L} et le centre \mathbf{C}_i .

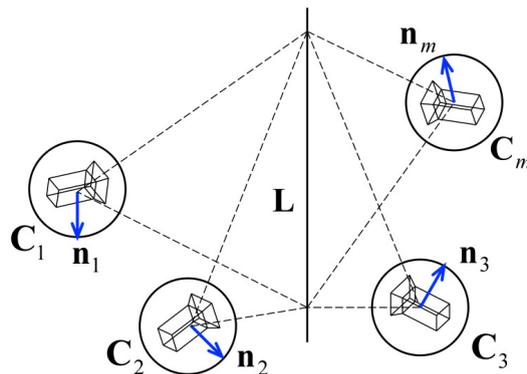


FIGURE 4.14 – Projection sphérique d'une droite dans l'espace 3D

Comme la ligne \mathbf{L} est projetée sur des grands cercles \mathbf{l}_i de normales respectives \mathbf{n}_i dans \mathbf{C}_i , nous avons la relation suivante

$$\mathbf{n}_i^T (\mathbf{R}_i \mathbf{L} + \mathbf{t}_i) = 0 \quad (4.7)$$

Considérons un triplet de caméras composé de la première et deux autres a et b . Nous notons ce triplet par $(1, a, b)$ où $2 \leq a, b \leq m$ et $a \neq b$. La relation entre ces trois vues peut être décrite

par l'équation (4.7) avec $i = 1, a, b$ et réécrite comme suit

$$\mathbf{A}\hat{\mathbf{L}} = 0 \quad (4.8)$$

où

$$\mathbf{A} = \begin{bmatrix} \mathbf{n}_1^\top & 0 \\ \mathbf{n}_a^\top \mathbf{R}_a & \mathbf{n}_a^\top \mathbf{t}_a \\ \mathbf{n}_b^\top \mathbf{R}_b & \mathbf{n}_b^\top \mathbf{t}_b \end{bmatrix} \quad \text{et} \quad \hat{\mathbf{L}} = (\mathbf{L}^\top, 1)^\top$$

Comme ce système admet une solution non-nulle, la matrice \mathbf{A} de taille 3×4 associée doit être de rang 2. Il existe donc une dépendance entre les trois lignes $\mathbf{r}_1, \mathbf{r}_2$ et \mathbf{r}_3 de \mathbf{A} qui peut s'écrire $\mathbf{r}_1 = \alpha \mathbf{r}_2 + \beta \mathbf{r}_3$. En remarquant que $r_{14} = 0$, nous pouvons poser $\alpha = \gamma \mathbf{t}_b^\top \mathbf{n}_b$ et $\beta = -\gamma \mathbf{t}_a^\top \mathbf{n}_a$ pour une valeur quelconque de γ . Ainsi, nous obtenons les relations suivantes

$$\mathbf{n}_1^\top = \alpha \mathbf{n}_a^\top \mathbf{R}_a + \beta \mathbf{n}_b^\top \mathbf{R}_b \quad (4.9)$$

$$\mathbf{n}_1 = \alpha \mathbf{R}_a^\top \mathbf{n}_a + \beta \mathbf{R}_b^\top \mathbf{n}_b \quad (4.10)$$

$$\mathbf{n}_1 = \gamma \mathbf{t}_b^\top \mathbf{n}_b \mathbf{R}_a^\top \mathbf{n}_a - \gamma \mathbf{t}_a^\top \mathbf{n}_a \mathbf{R}_b^\top \mathbf{n}_b \quad (4.11)$$

$$[\mathbf{n}_1]_\times \mathbf{R}_a^\top \mathbf{n}_a \mathbf{n}_b^\top \mathbf{t}_b - [\mathbf{n}_1]_\times \mathbf{R}_b^\top \mathbf{n}_b \mathbf{n}_a^\top \mathbf{t}_a = 0 \quad (4.12)$$

L'équation (4.12) relie les correspondances de droites dans un triplet de vues par les transformations $[\mathbf{R}_a | \mathbf{t}_a]$ et $[\mathbf{R}_b | \mathbf{t}_b]$. Elle permet une estimation linéaire des translations $\mathbf{t}_{a,b}$ à partir de rotations $\mathbf{R}_{a,b}$ et de normales $\mathbf{n}_{1,a,b}$.

Avec une correspondance de droites entre m caméras, nous avons $(m-1)(m-2)/2$ triplets de vue $(1, a, b)$ ou relations trilineaires dans (4.12). Ces relations peuvent être combinées dans un seul système qui permet une estimation linéaire de translations $\mathbf{t}_{2\dots m}$ à partir de rotations $\mathbf{R}_{2\dots m}$ et correspondances de normales $\mathbf{n}_{1\dots m}$

$$\mathbf{B}\mathbf{X} = 0 \quad (4.13)$$

où \mathbf{B} est une matrice de taille $3(m-1)(m-2)/2$ -par- $3(m-1)$

$$\mathbf{B} = \begin{bmatrix} -[\mathbf{n}_1]_\times \mathbf{R}_3^\top \mathbf{n}_3 \mathbf{n}_2^\top & [\mathbf{n}_1]_\times \mathbf{R}_2^\top \mathbf{n}_2 \mathbf{n}_3^\top & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -[\mathbf{n}_1]_\times \mathbf{R}_m^\top \mathbf{n}_m \mathbf{n}_{m-1}^\top & [\mathbf{n}_1]_\times \mathbf{R}_{m-1}^\top \mathbf{n}_{m-1} \mathbf{n}_m^\top \end{bmatrix}$$

et

$$\mathbf{X} = (\mathbf{t}_2^\top, \mathbf{t}_3^\top, \dots, \mathbf{t}_m^\top)^\top$$

L'ajout d'une autre correspondance de droite augmente la taille de la matrice \mathbf{B} de $3(m-1)(m-2)/2$ lignes. Ainsi, avec n_L correspondances de droites entre m cameras, \mathbf{B} devient une matrice de taille $3n_L(m-1)(m-2)/2$ -par- $3(m-1)$. Par conséquent, nous pouvons obtenir la solution de \mathbf{X} à l'aide d'une décomposition en valeur singulière (*Single Value Decomposition* - SVD) de la matrice \mathbf{B} suivante :

$$\mathbf{B} = \begin{bmatrix} -[\mathbf{n}_1^1] \times \mathbf{R}_3^T \mathbf{n}_3^1 \mathbf{n}_2^{1T} & [\mathbf{n}_1^1] \times \mathbf{R}_2^T \mathbf{n}_2^1 \mathbf{n}_3^{1T} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -[\mathbf{n}_1^{n_L}] \times \mathbf{R}_3^T \mathbf{n}_3^{n_L} \mathbf{n}_2^{n_L T} & [\mathbf{n}_1^{n_L}] \times \mathbf{R}_2^T \mathbf{n}_2^{n_L} \mathbf{n}_3^{n_L T} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -[\mathbf{n}_1^1] \times \mathbf{R}_m^T \mathbf{n}_m^1 \mathbf{n}_{m-1}^{1T} & [\mathbf{n}_1^1] \times \mathbf{R}_{m-1}^T \mathbf{n}_{m-1}^1 \mathbf{n}_m^{1T} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -[\mathbf{n}_1^{n_L}] \times \mathbf{R}_m^T \mathbf{n}_m^{n_L} \mathbf{n}_{m-1}^{n_L T} & [\mathbf{n}_1^{n_L}] \times \mathbf{R}_{m-1}^T \mathbf{n}_{m-1}^{n_L} \mathbf{n}_m^{n_L T} \end{bmatrix}$$

4.2.3 Estimation de la translation à partir de points et droites

Nous avons vu, en section 3.2.1, que deux points en correspondance \mathbf{p}_a et \mathbf{p}_b entre deux caméras sphériques a et b sont liées par la transformation $[\mathbf{R}|\mathbf{t}]$ par la relation suivante :

$$(\mathbf{R}\mathbf{p}_a \times \mathbf{p}_b)^T \mathbf{t} = 0 \quad (4.14)$$

Il est évident que la contrainte bilinéaire dans (4.14) et la contrainte trilinéaire dans (4.12) peuvent être enveloppées dans un système linéaire qui permet le calcul des translations à partir de points et droites. Par exemple, considérons trois caméras avec n_L correspondances de droites $(\mathbf{n}_1^{1\dots n_L}, \mathbf{n}_2^{1\dots n_L}, \mathbf{n}_3^{1\dots n_L})$ entre trois vues (1, 2 et 3), n_{12} correspondances de points $(\mathbf{p}_1^{1\dots n_{12}}, \mathbf{p}_2^{1\dots n_{12}})$ entre deux vues (1 et 2), et n_{13} correspondances de points $(\mathbf{p}_1^{1\dots n_{13}}, \mathbf{p}_3^{1\dots n_{13}})$ entre deux vues (1 et 3), le système suivant peut résoudre les translations \mathbf{t}_2 et \mathbf{t}_3 en utilisant la décomposition en valeur singulière de la matrice \mathbf{G} :

$$\mathbf{G} \begin{pmatrix} \mathbf{t}_2 \\ \mathbf{t}_3 \end{pmatrix} = 0 \quad (4.15)$$

où

$$\mathbf{G} = \begin{bmatrix} (\mathbf{R}_2 \mathbf{p}_1^1 \times \mathbf{p}_2^1)^T & 0 \\ \dots & \dots \\ (\mathbf{R}_2 \mathbf{p}_1^{n_{12}} \times \mathbf{p}_2^{n_{12}})^T & 0 \\ \hline 0 & (\mathbf{R}_3 \mathbf{p}_1^1 \times \mathbf{p}_3^1)^T \\ \dots & \dots \\ 0 & (\mathbf{R}_3 \mathbf{p}_1^{n_{13}} \times \mathbf{p}_3^{n_{13}})^T \\ \hline -[\mathbf{n}_1^1] \times \mathbf{R}_3^T \mathbf{n}_3^1 \mathbf{n}_2^{1T} & [\mathbf{n}_1^1] \times \mathbf{R}_2^T \mathbf{n}_2^1 \mathbf{n}_3^{1T} \\ \dots & \dots \\ -[\mathbf{n}_1^{n_L}] \times \mathbf{R}_3^T \mathbf{n}_3^{n_L} \mathbf{n}_2^{n_L T} & [\mathbf{n}_1^{n_L}] \times \mathbf{R}_2^T \mathbf{n}_2^{n_L} \mathbf{n}_3^{n_L T} \end{bmatrix}$$

Cependant, la combinaison des correspondances entre les vues (1 et 2) et (1 et 3) réduit quelques relations dépendantes. Par conséquent, au lieu d'utiliser les correspondances de points entre les vues (1 et 2) et (1 et 3), nous relierons n_{23} correspondances de points ($\mathbf{p}_2^{1\dots n_{23}}, \mathbf{p}_3^{1\dots n_{23}}$) entre deux vues (2 et 3) et intégrons cette contrainte dans la matrice \mathbf{G} :

$$\mathbf{G} = \begin{bmatrix} -(\mathbf{R}_3\mathbf{R}_2^\top \mathbf{p}_2^1 \times \mathbf{p}_3^1)^\top \mathbf{R}_3\mathbf{R}_2^\top & (\mathbf{R}_3\mathbf{R}_2^\top \mathbf{p}_2^1 \times \mathbf{p}_3^1)^\top \\ \dots & \dots \\ -(\mathbf{R}_3\mathbf{R}_2^\top \mathbf{p}_2^{n_{23}} \times \mathbf{p}_3^{n_{23}})^\top \mathbf{R}_3\mathbf{R}_2^\top & (\mathbf{R}_3\mathbf{R}_2^\top \mathbf{p}_2^{n_{23}} \times \mathbf{p}_3^{n_{23}})^\top \\ \hline -[\mathbf{n}_1^1] \times \mathbf{R}_3^\top \mathbf{n}_3^1 \mathbf{n}_2^{1\top} & [\mathbf{n}_1^1] \times \mathbf{R}_2^\top \mathbf{n}_2^1 \mathbf{n}_3^{1\top} \\ \dots & \dots \\ -[\mathbf{n}_1^{n_L}] \times \mathbf{R}_3^\top \mathbf{n}_3^{n_L} \mathbf{n}_2^{n_L\top} & [\mathbf{n}_1^{n_L}] \times \mathbf{R}_2^\top \mathbf{n}_2^{n_L} \mathbf{n}_3^{n_L\top} \end{bmatrix}$$

4.3 Reconstruction 3D et ajustement de faisceaux

Nous venons de voir comment nous pouvons estimer les six degrés de liberté de mouvement d'une caméra sphérique. Par la suite, nous allons voir comment reconstruire l'environnement et affiner nos résultats par ajustement de faisceaux tout en tenant compte de la géométrie sphérique.

4.3.1 Reconstruction de points

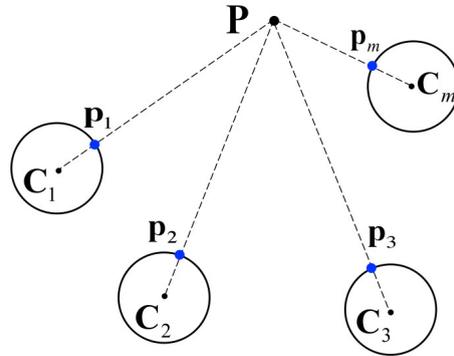


FIGURE 4.15 – Reconstruction de points

Chaque point \mathbf{P} dans l'espace 3D est reconstruit par la triangulation des lignes de vue, chacune passant par \mathbf{P} , le centre de caméra \mathbf{C}_i et le point image projeté sur la sphère \mathbf{p}_i (4.16) comme illustré figure 4.15. La solution linéaire de \mathbf{P} peut être obtenue à partir de l'équation (4.17) en utilisant le pseudo-inverse de la matrice \mathbf{M} .

$$\mathbf{P} = \mathbf{C}_i + \lambda_i(\mathbf{p}_i - \mathbf{C}_i) \quad (4.16)$$

$$\mathbf{M}\hat{\mathbf{P}} = \mathbf{C} \quad (4.17)$$

où

$$M = \begin{bmatrix} I & \mathbf{C}_1 - \mathbf{p}_1 & 0 & \dots & 0 \\ I & 0 & \mathbf{C}_2 - \mathbf{p}_2 & \dots & 0 \\ & & \dots & & \\ I & 0 & 0 & \dots & \mathbf{C}_m - \mathbf{p}_m \end{bmatrix}$$

$$\hat{\mathbf{P}} = (\mathbf{P}^\top, \lambda_1, \lambda_2, \dots, \lambda_m)^\top$$

$$\mathbf{C} = (\mathbf{C}_1^\top, \mathbf{C}_2^\top, \dots, \mathbf{C}_m^\top)^\top$$

4.3.2 Reconstruction de droites

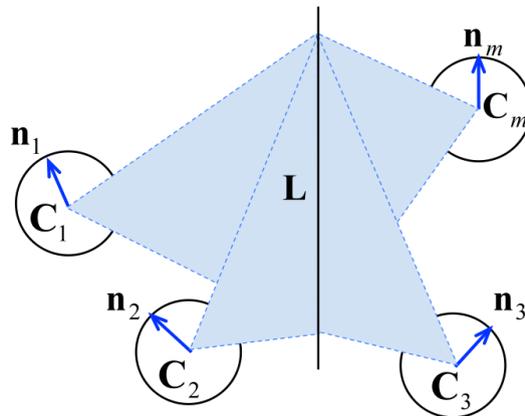


FIGURE 4.16 – Reconstruction de droites

Chaque droite 3D est reconstruite par l'intersection des plans 3D passant par les correspondances de droites entre les vues. Elles sont calculées à partir de la transformation entre des caméras et les correspondances de normales

$$\mathbf{N}\hat{\mathbf{L}} = 0 \quad (4.18)$$

où

$$\mathbf{N} = \begin{bmatrix} \mathbf{n}_1^\top & 0 \\ \mathbf{n}_2^\top \mathbf{R}_2 & \mathbf{n}_2^\top \mathbf{t}_2 \\ \vdots & \vdots \\ \mathbf{n}_m^\top \mathbf{R}_m & \mathbf{n}_m^\top \mathbf{t}_m \end{bmatrix} \quad \text{et} \quad \hat{\mathbf{L}} = (\mathbf{L}^\top, 1)^\top$$

La décomposition en valeur singulière de $\mathbf{N} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ peut être utilisée pour définir la droite d'intersection des plans [HZ04].

4.3.3 Ajustement de faisceaux

L'ajustement de faisceaux est une étape d'optimisation pour affiner les paramètres de caméras et de structure 3D en minimisant l'erreur de re-projection. Nous présentons premièrement la paramétrisation de caméras, points et droites 3D et puis la procédure de minimisation.

Paramétrisation

Le faisceau de lignes de vue passant par les centres de caméras et les primitives dans l'espace 3D est défini par les paramètres de caméras et primitives 3D.

- Chacune des caméras est paramétrée par un vecteur $\mathbf{c}_i = (r_0, r_1, r_2, r_3, t_x, t_y, t_z)_i$ ($i = 1 \dots m$) où (r_0, r_1, r_2, r_3) est la représentation par les quaternions de la rotation et (t_x, t_y, t_z) la translation.
- Chacun des n_P points 3D est décrit par un vecteur $\mathbf{P}^j = (p_x, p_y, p_z)^j$ ($j = 1 \dots n_P$).
- Chacune des n_L droites 3D est représentée par un vecteur $\mathbf{L}^k = (e_x^1, e_y^1, e_z^1, e_x^2, e_y^2, e_z^2)^k$ ($k = 1 \dots n_L$) établi par deux points \mathbf{e}^1 et \mathbf{e}^2 sur cette droite.

Le vecteur de paramètres dans l'optimisation est défini par l'ensemble des paramètres décrivant m caméras, n_P points et n_L droites :

$$\mathbf{W} = (\mathbf{c}_1 \dots \mathbf{c}_m, \mathbf{P}^1 \dots \mathbf{P}^{n_P}, \mathbf{L}^1 \dots \mathbf{L}^{n_L})$$

La fonction de coût et la minimisation

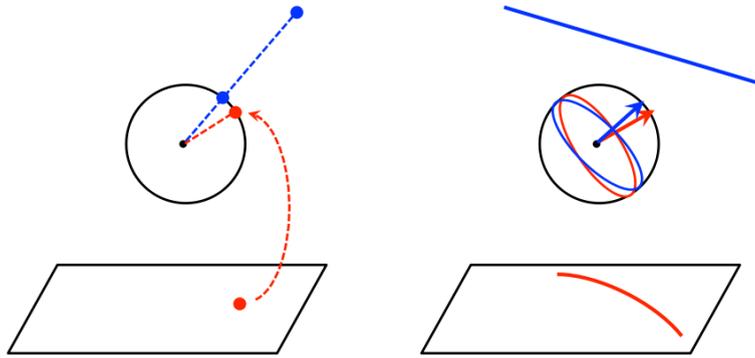


FIGURE 4.17 – L'erreur de re-projection de points et droites définie sur la sphère

L'ajustement de faisceaux minimise l'erreur de re-projection suivante

$$\min_{\mathbf{c}_i, \mathbf{P}^j, \mathbf{L}^k} \left(\sum_{i=1}^m \sum_{j=1}^{n_P} d_P(\hat{\mathbf{p}}_i^j, \mathbf{s}(\mathbf{c}_i, \mathbf{P}^j))^2 + \sum_{i=1}^m \sum_{k=1}^{n_L} d_L(\hat{\mathbf{n}}_i^k, \mathbf{s}(\mathbf{c}_i, \mathbf{L}^k))^2 \right) \quad (4.19)$$

où

$$d_P = \left\| \hat{\mathbf{p}}_i^j \times \frac{\mathbf{R}_i \mathbf{P}^j + \mathbf{t}_i}{\|\mathbf{R}_i \mathbf{P}^j + \mathbf{t}_i\|} \right\| \quad \text{et} \quad d_L = \left\| \hat{\mathbf{n}}_i^k \times \frac{(\mathbf{R}_i \mathbf{e}^{1k} + \mathbf{t}_i) \times (\mathbf{R}_i \mathbf{e}^{2k} + \mathbf{t}_i)}{\|(\mathbf{R}_i \mathbf{e}^{1k} + \mathbf{t}_i) \times (\mathbf{R}_i \mathbf{e}^{2k} + \mathbf{t}_i)\|} \right\|$$

où

- \mathbf{s} est la projection sphérique du point 3D \mathbf{P}^j ou de la droite 3D \mathbf{L}^k dans la caméra \mathbf{c}_i .
- $\hat{\mathbf{p}}_i^j$ et $\hat{\mathbf{n}}_i^k$ sont les re-projections sur la sphère du point d'image j et de la droite d'image k dans la caméra i .
- d_P représente la distance géodésique entre deux points : le point observé et le point prédit sur la sphère. Sur la gauche de la figure 4.17, le point observé dans l'image et sa re-projection sur la sphère sont en rouge. Le point en bleu est prédit à partir de paramètres de caméras et de points 3D.

- d_L représente la différence angulaire entre deux normales : la normale de la droite observée et la normale de la droite prédite sur la sphère. Sur la droite de la figure 4.17, la droite image et sa re-projection sphérique sont en rouge. Le grand cercle et la normale en bleu sont prédits à partir des paramètres des caméras et des droites 3D.

La minimisation peut être résolue par l'algorithme non-linéaire de Levenberg-Marquardt. La valeur initiale du vecteur de paramètres \mathbf{W}_0 est fournie par l'estimation de mouvement et la reconstruction 3D.

Chaque ligne de la matrice Jacobienne est calculée pour chaque point ou chaque droite dans chaque caméra.

Points :

$$\frac{\partial d_P}{\partial \mathbf{W}} = \left[\frac{\partial d_P}{\partial \mathbf{c}_1} \cdots \frac{\partial d_P}{\partial \mathbf{c}_m}, \frac{\partial d_P}{\partial \mathbf{P}^1} \cdots \frac{\partial d_P}{\partial \mathbf{P}^{n_P}}, 0 \dots 0 \right] \quad (4.20)$$

Droites :

$$\frac{\partial d_L}{\partial \mathbf{W}} = \left[\frac{\partial d_L}{\partial \mathbf{c}_1} \cdots \frac{\partial d_L}{\partial \mathbf{c}_m}, 0 \dots 0, \frac{\partial d_L}{\partial \mathbf{L}^1} \cdots \frac{\partial d_L}{\partial \mathbf{L}^{n_L}} \right] \quad (4.21)$$

En pratique, la dérivation de d_P et d_L définies dans l'équation (4.19) est assez complexe. Ainsi, nous représentons la différence entre l'observation et la prédiction par les vecteurs $d_P = \hat{\mathbf{p}}_i^j \times (\mathbf{R}_i \mathbf{P}^j + \mathbf{t}_i)$ et $d_L = \hat{\mathbf{n}}_i^k \times [(\mathbf{R}_i \mathbf{e}^{1k} + \mathbf{t}_i) \times (\mathbf{R}_i \mathbf{e}^{2k} + \mathbf{t}_i)]$.

En notant que $\Delta = \frac{\partial d_P}{\partial \mathbf{c}}$, $\Gamma = \frac{\partial d_P}{\partial \mathbf{P}}$, $\Upsilon = \frac{\partial d_L}{\partial \mathbf{c}}$ et $\Omega = \frac{\partial d_L}{\partial \mathbf{L}}$, la Jacobienne a pour expression :

$$\mathbf{J} = \left[\begin{array}{cccc|cccc|cccc} \Delta_1^1 & 0 & \dots & 0 & \Gamma_1^1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \Delta_2^1 & \dots & 0 & \Gamma_2^1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Delta_m^1 & \Gamma_m^1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots \\ \Delta_1^{n_P} & 0 & \dots & 0 & 0 & 0 & \dots & \Gamma_1^{n_P} & 0 & 0 & \dots & 0 \\ 0 & \Delta_2^{n_P} & \dots & 0 & 0 & 0 & \dots & \Gamma_2^{n_P} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Delta_m^{n_P} & 0 & 0 & \dots & \Gamma_m^{n_P} & 0 & 0 & \dots & 0 \\ \hline \Upsilon_1^1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \Omega_1^1 & 0 & \dots & 0 \\ 0 & \Upsilon_2^1 & \dots & 0 & 0 & 0 & \dots & 0 & \Omega_2^1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Upsilon_m^1 & 0 & 0 & \dots & 0 & \Omega_m^1 & 0 & \dots & 0 \\ \vdots & \vdots \\ \Upsilon_1^{n_L} & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \Omega_1^{n_L} \\ 0 & \Upsilon_2^{n_L} & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \Omega_2^{n_L} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Upsilon_m^{n_L} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \Omega_m^{n_L} \end{array} \right] \quad (4.22)$$

4.4 Résultats

4.4.1 Estimation de la translation par l'algorithme des 2 points

Cette partie analyse la précision et la robustesse de l'algorithme des 2 points sur des données de synthèses et des séquences réelles.

Dans un premier temps, nous avons analysé la sensibilité de l'algorithme des 2 points aux bruits. Pour ce faire, nous avons synthétisé 1000 images sphériques composées de 100 points auxquels nous avons ajouté un bruit blanc gaussien d'écart type 1 et 3 pixels. Dans une première série d'expériences, la véritable rotation est directement utilisée pour le calcul de T par l'algorithme des 2 points. Pour simuler le fait que la rotation ne peut pas être parfaitement estimée à partir des images, nous avons effectué une deuxième série d'expériences où les trois angles de rotation ont été bruités par un bruit blanc gaussien d'écart type croissant : $[0^\circ, \dots, 2^\circ]$. A titre de comparaison, nous avons également appliqué l'algorithme des 5 points pour calculer la matrice essentielle [Nis04]. La figure 4.18 représente l'erreur moyenne en degrés pour l'estimation de la direction de la translation (le bruit sur les points étant de 1° pour les expériences en haut de la figure et 3° pour le bas). Comme l'algorithme des 5 points et l'algorithme des 2 points avec rotation connue ne dépendent pas du bruit de rotation, leur erreur est constante. Nous pouvons remarquer que l'erreur de l'algorithme des 2 points avec une rotation bruitée, augmente linéairement en fonction du niveau de bruit sur la rotation, mais qu'il reste plus performant que les 5 points jusqu'à un bruit de plus de 1.5° dans la rotation.

Les résultats de l'algorithme sur une séquence réelle (Fig. 4.19) sont proposés figure 4.20. Cette séquence est composée de 33 images prises par une caméra catadioptrique en translation pure avec un intervalle de 50 cm en chaque prise de vue. L'algorithme des 2 points se traduit graphiquement par l'estimation de l'intersection entre des grands cercles sur la sphère. En effet, soit $(P_i, P'_i), i = 1 \dots N$, N points et $n_i = R \times P_i P'_i$, chaque vecteur n_i définit un grand cercle C_i (fig 4.3) et l'équation (4.6) signifie que tous ces grands cercles se coupent en deux points diamétralement opposés $I_1, I_2 \in \mathbb{S}^2$ et comme pour l'estimation de la rotation, nous avons $I_1 I_2 \sim T$.

Ainsi, la figure 4.19 présente l'estimation de translation entre deux images consécutives de la séquence. Sur la figure 4.19(a), nous avons projeté les grands cercles C_i des N points appariés dans les images catadioptriques. Si les points sont parfaitement appariés, les grands cercles devraient se couper en un seul point. Ce qui n'est pas le cas ici mais le RANSAC permet de réduire cette erreur (Fig. 4.19(b)).

La trajectoire est tracée dans le repère associé de la première caméra. La figure 4.20 montre le mouvement de la caméra estimé par la méthode proposée, ce qui est cohérent avec le mouvement rectiligne réel.

4.4.2 Estimation de la translation à partir de droites

Nous comparons la précision de l'estimation de translations basée droites à celle basée points en créant 100 points et 100 droites en 3D répartis au hasard dans une sphère d'un rayon de 5 mètres. Quatre caméras avec une *baseline* moyenne de 0.5 mètres observent ces primitives

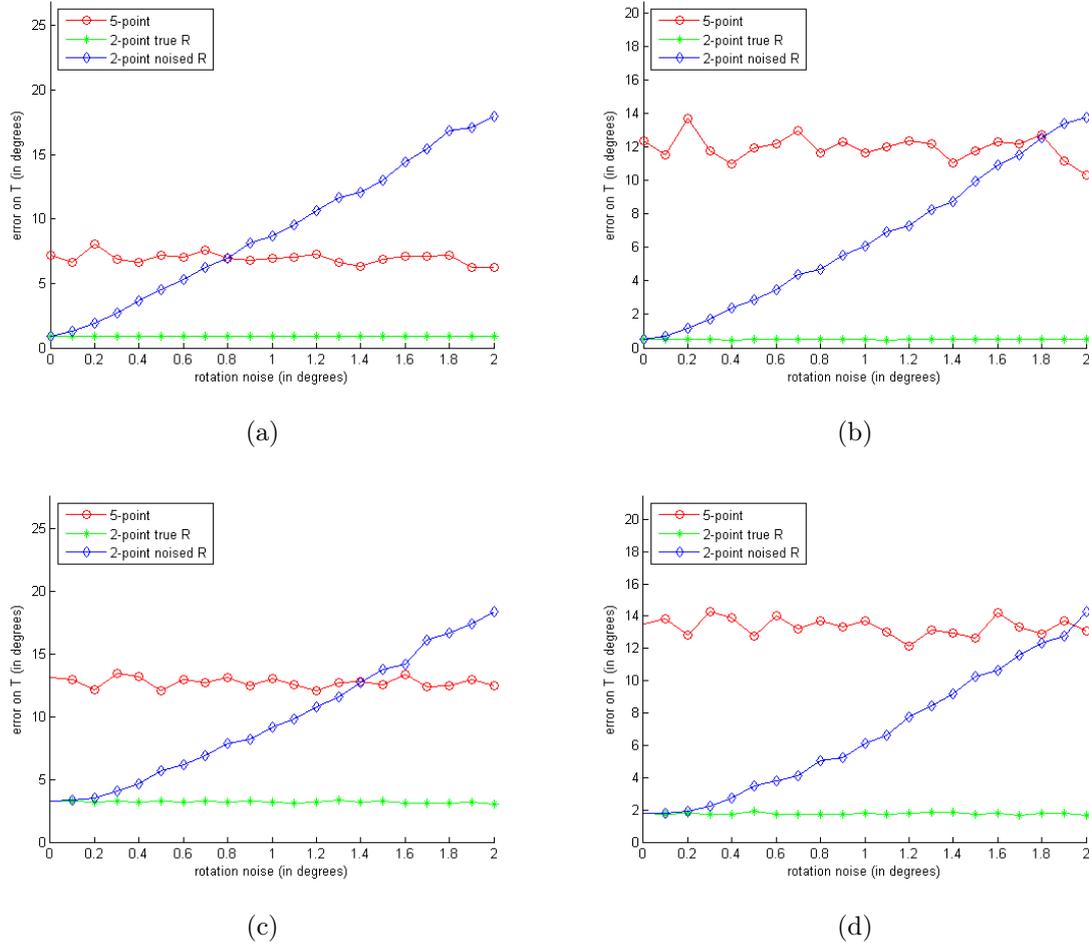


FIGURE 4.18 – Comparaison de la moyenne (colonne de gauche) et de l'écart type (colonne de droite) de l'erreur en degrés (axe Y) avec l'algorithme des 5 points et l'algorithme des 2 points. La courbe rouge (resp. verte) montre l'évolution de l'algorithme du 5 points (resp. du 2 points). La courbe bleue repérantes l'estimation de la translation à l'aide des 2 points en bruitant la rotation avec un bruit blanc donc l'écart type augmente en fonction de l'abscisse. Sur la rangée du haut, les points de données ont été bruités par une distribution gaussienne de moyenne nulle et d'écart type 1 pixel. Sur la ligne du bas, l'écart type est de 3 pixels. Elle montre clairement que l'algorithme des 2 points est plus fiable que la méthode de 5 points jusqu'à un certain niveau de bruit de rotation, en particulier lorsque le bruit augmente.

à une distance de 10 mètres. Les translations entre ces caméras ont été estimées en utilisant l'algorithme des 2 points, l'estimation basée droites (section 4.2.2) et la combinaison de points et droites (section 4.2.3). Comme les points de l'algorithme des 2 points et les normales de notre proposition sont définis sur la sphère unitaire, nous pouvons les paramétrer par les angles

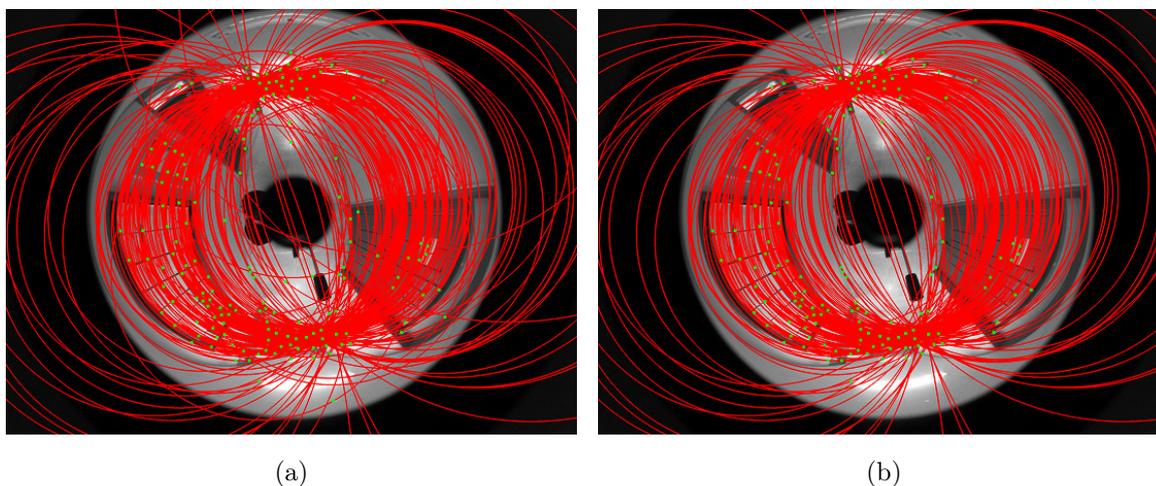


FIGURE 4.19 – (a) Grands cercles C_i projetés dans l'image catadioptrique (b) grands cercles retenus pour l'estimation de la translation après le RANSAC.

d'azimut et d'élévation. Pour vérifier la sensibilité des méthodes aux bruits de mesure, nous avons ajouté un bruit blanc gaussien (moyenne de zéro et écart type de 0 à 0.1 degré) à chaque angle. De plus, pour simuler l'imprécision dans l'estimation des rotations, nous avons perturbé chaque angle de rotation par un bruit blanc Gaussien. La figure 4.21 montre l'erreur angulaire moyenne du calcul des translations après 1000 essais. Nous pouvons remarquer que la méthode basée droites est plus robuste en présence de bruit. Ceci montre un apport important de notre méthode par rapport aux méthodes basées points.

Afin d'évaluer l'ajustement de faisceaux, nous avons réalisé une autre simulation dans laquelle nous avons créé 10 points et 10 droites aléatoirement dans une sphère d'un rayon de 5 mètres. Trois caméras avec une *baseline* moyenne de 0.5 mètres observaient ces primitives à une distance de 10 mètres. Les points et les normales sont perturbés de la même manière que lors de la première simulation. Tout d'abord, la pose des caméras a été calculée en utilisant l'algorithme des 5 points [Nis04] implémenté par Stewenius et al. [SEN06]. Ensuite, en utilisant les rotations estimées, nous avons re-estimé les translations à partir de points et droites. Enfin, nous avons affiné la pose des caméras et de la structure 3D par l'ajustement de faisceaux proposé dans la section 4.3.3. L'objectif de cette simulation est de comparer l'algorithme des 5 points à la combinaison de points et droites et de vérifier l'optimisation de l'ajustement de faisceaux. La figure 4.21 montre l'erreur angulaire moyenne de l'estimation de translations après 1000 essais. Nous pouvons observer que la combinaison de points et droites a amélioré l'estimation par rapport à celle n'utilisant que les points et que l'ajustement de faisceaux a réussi à réduire l'erreur de l'estimation.

Dans le cas réel, une séquence d'images d'un damier a été capturée par une caméra fisheye. Cette caméra a été calibrée par la méthode de calibrage développée dans [MR07]. Puis nous avons détecté les droites dans ces images. A partir de six images de cette séquence (figure 4.22), nous avons effectué la mise en correspondance de droites, retrouvé la pose des caméras et reconstruit le damier en 3D. Notre estimation a été comparée aux paramètres extrinsèques fournis par le

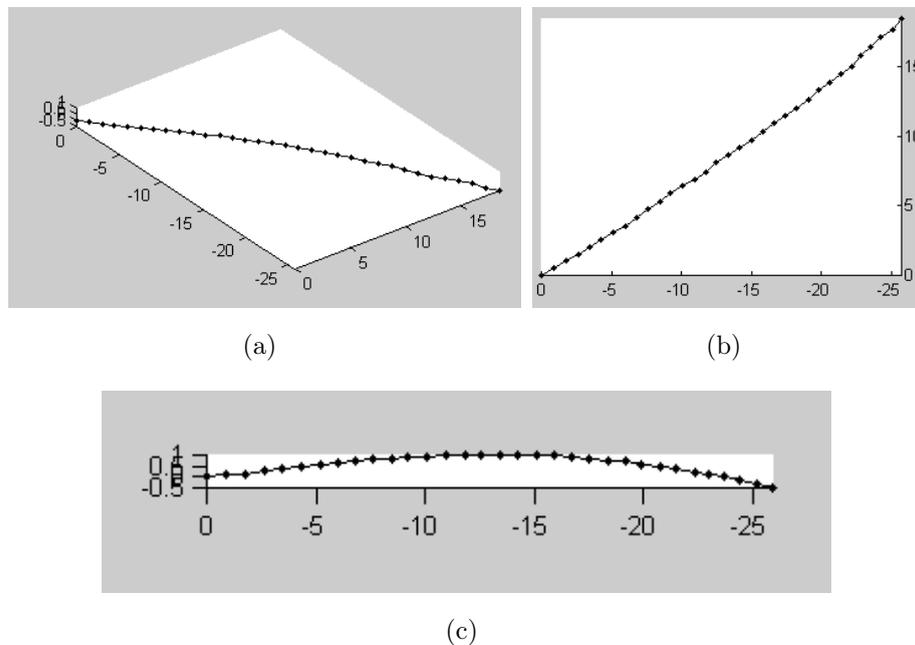


FIGURE 4.20 – Estimation du déplacement : (a), vue 3D, (b), (c) vues de côté.

calibrage, dont la différence est donnée dans le tableau 4.2. La différence entre deux rotations R_i et R_j est définie à partir de la distance entre leur représentation sous forme quaternion \mathbf{r}_i et \mathbf{r}_j (eq. 4.23). La différence entre deux translations est définie par l'angle entre leur direction. Nous pouvons remarquer que la pose des caméras calculée par notre méthode est proche de celle fournie par le calibrage. La reconstruction 3D est illustrée sur la figure 4.23.

$$d(R_i, R_j) = \min(\|\mathbf{r}_i - \mathbf{r}_j\|, \|\mathbf{r}_i + \mathbf{r}_j\|) \quad (4.23)$$

Nous avons ensuite évalué les différentes approches d'estimation de pose basées : (i) points (l'algorithme des 5 points), (ii) droites (section 4.2.2) et (iii) points et droites (section 4.2.3). Trois images capturées par deux caméras fish-eye (\mathbf{C}_1 et \mathbf{C}_2) et une caméra perspective (\mathbf{C}_3) sont montrées sur la figure 4.24. A partir de correspondances de points et droites, nous avons estimé la pose des caméras et la structure 3D et affiné le résultat par l'ajustement de faisceaux. Une vue de la reconstruction est illustrée sur la figure 4.25.

Pour évaluer la reconstruction à l'échelle, nous avons vérifié la dimension des portes reconstruites (tableau 4.3). Quatre portes avec les bordures extraites comme illustrées figure 4.24) sont notées Porte 1 à 4 de gauche à droite. En utilisant le rapport hauteur/largeur de chaque porte reconstruite, nous avons déduit et comparé sa hauteur estimée à sa hauteur réelle. L'approche basée droites a fourni de meilleurs résultats que l'algorithme des 5 points et il n'y avait pas de différence importante entre l'estimation basée droites et l'estimation basée points et droites. Cette différence s'explique du fait que l'estimation basée droites est moins sensible aux bruits que l'estimation basée points. Toutefois le résultat n'est pas satisfaisant pour la Porte 1 car elle se situe près du bord de l'image où se trouve une distorsion importante, particulièrement dans

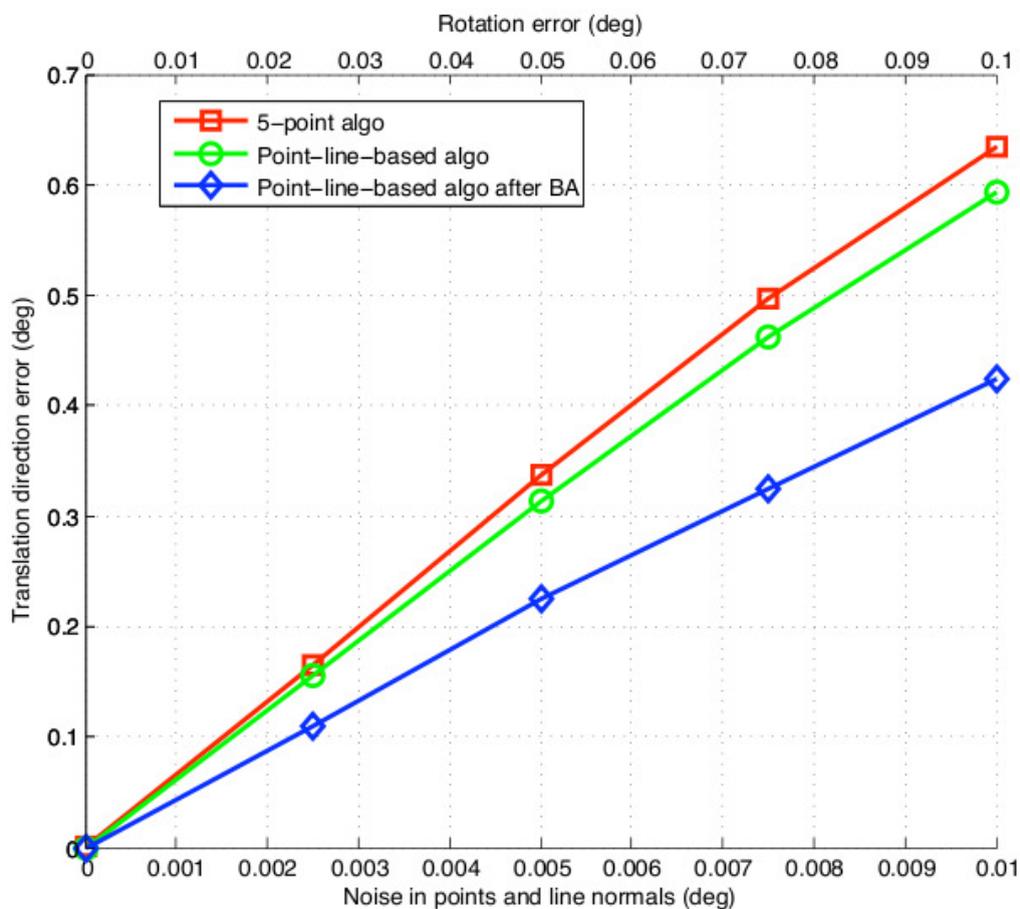


FIGURE 4.21 – L’erreur de l’estimation des translations par l’algorithme des 5 points (rouge), la méthode basée points et droites (vert) et l’ajustement de faisceaux (bleu)

les images fish-eye. La re-projection de droites 3D dans une des images fish-eye est montrée sur la figure 4.26. Nous pouvons percevoir que l’approche basée points a produit une erreur de re-projection importante tandis que les approches basées droites fonctionnent convenablement.

R_i	Différence des rotations	t_i	Différence des translations (degrés)
R_2	0.003	t_2	1.473
R_3	0.003	t_3	1.819
R_4	0.006	t_4	0.981
R_5	0.008	t_5	2.658
R_6	0.002	t_6	1.338

TABLE 4.2 – Différence entre le calcul de pose basé droite et le calibrage extrinsèque

Une séquence plus longue en extérieur prise à l’aide d’un robot Pioneer équipé d’une caméra catadioptrique a permis d’évaluer l’estimation de mouvement basée droites 4.27. Le résultat

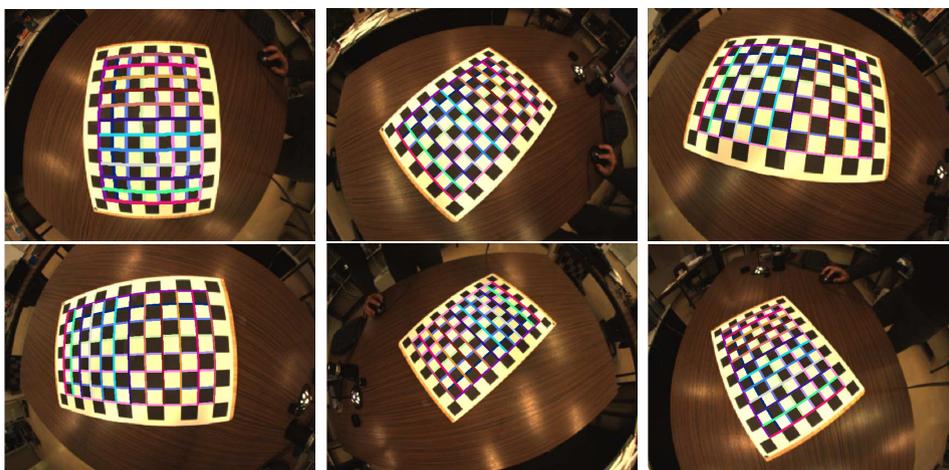


FIGURE 4.22 – Images fish-eye d'un damier avec la détection et l'appariement de droites

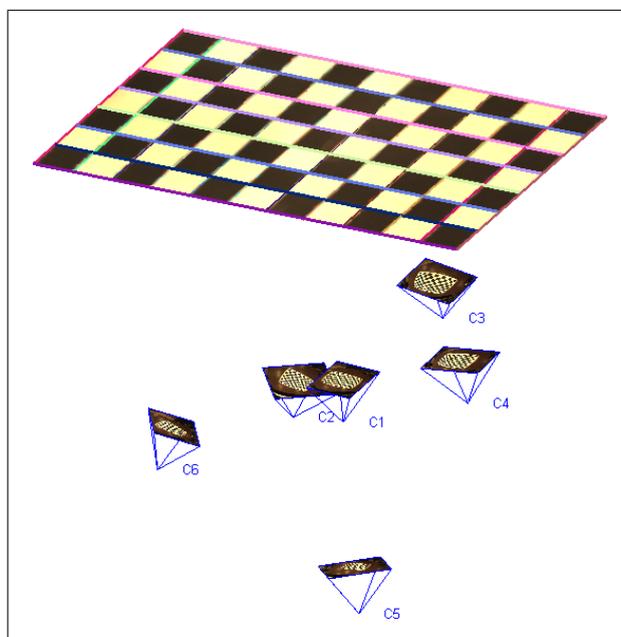


FIGURE 4.23 – La pose des caméras et le damier en 3D

d'estimation de mouvement est montré sur la figure 4.28 et comparé avec la vérité terrain.

Le même robot équipé d'une caméra catadioptrique sur la figure 4.27 s'est déplacé le long d'une trajectoire rectangulaire. A partir de la séquence capturée par la caméra, nous avons réalisé le suivi de points et droites et retrouvé la trajectoire du robot en utilisant différentes approches : l'algorithme des 5 points, des 2 points, l'approche basée droites et la combinaison de points et droites. La figure 4.29 illustre quelques images de la séquence avec le suivi de primitives.

La figure 4.30 compare l'estimation basée droites avec l'algorithme des 5 points. Nous pouvons observer que la trajectoire retrouvée par l'algorithme des 5 points n'est pas fermée. Cet algorithme



FIGURE 4.24 – Images fisheye et perspectives avec les correspondances de points et droites

Méthodes	Porte 1	Porte 2	Porte 3	Porte 4
5-point	242	199	215	212
Basée droites	228	206	206	200
Basée points et droites	229	206	206	204
Vérité terrain	203	203	203	203

TABLE 4.3 – Reconstruction à partir de la séquence de portes

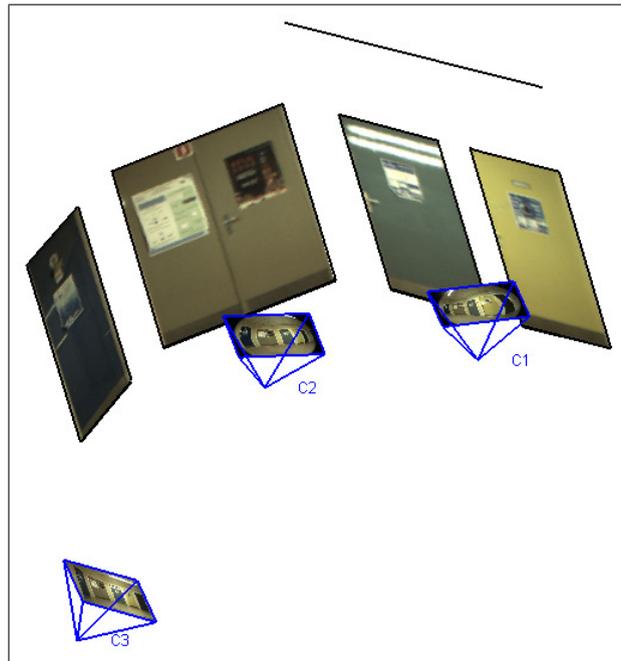


FIGURE 4.25 – La pose des caméras et la structure 3D

est ici mis en défaut à cause de faux appariements. Au contraire, la méthode basée droites retrouve le véritable déplacement du robot malgré là aussi des correspondances erronées.

La figure 4.31 compare l'estimation de mouvement basée droites (rouge), l'algorithme des 2 points (noir) et la combinaison de points et droites (bleu). Cette figure confirme les résultats des simulations car l'estimation basée droites est plus précise que l'algorithme des 2 points.

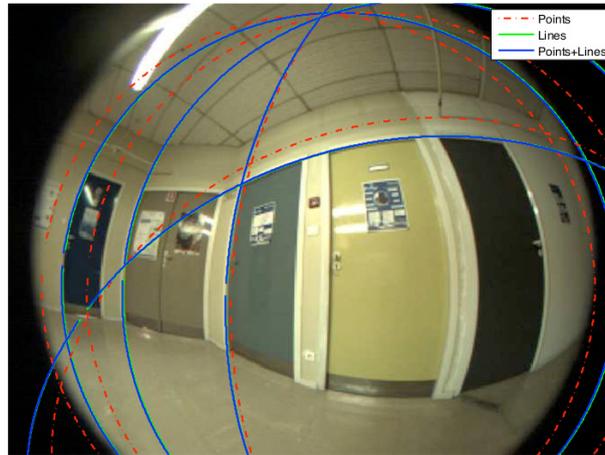


FIGURE 4.26 – La re-projection dans l'image fish-eye de la droite reconstruite en 3D

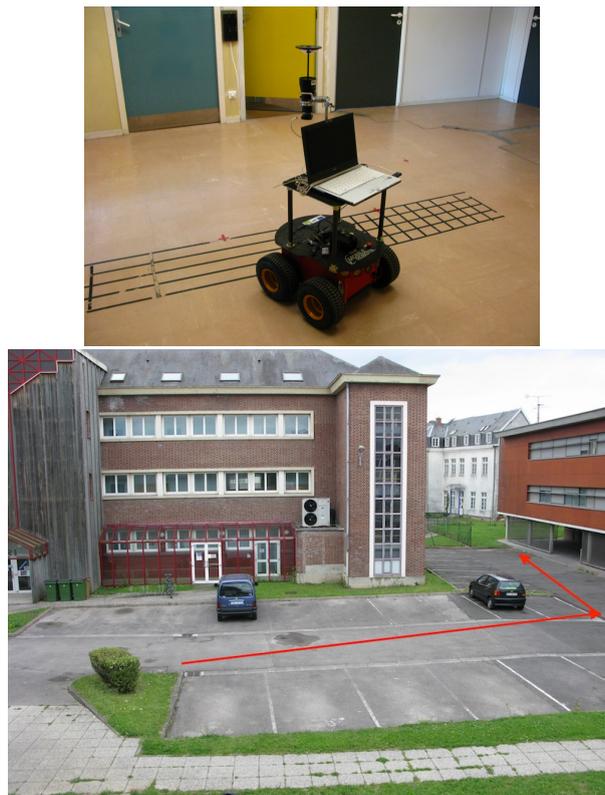


FIGURE 4.27 – Le robot mobile Pioneer équipé d'une caméra catadioptrique et son parcours dans le parking

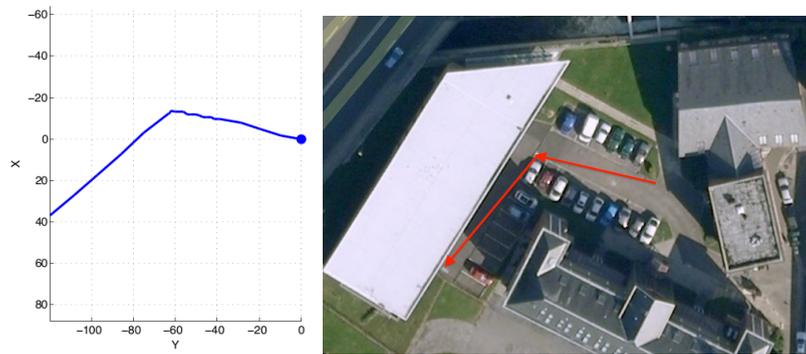


FIGURE 4.28 – L'estimation de déplacement du robot (bleu) comparée avec la vérité terrain (rouge)

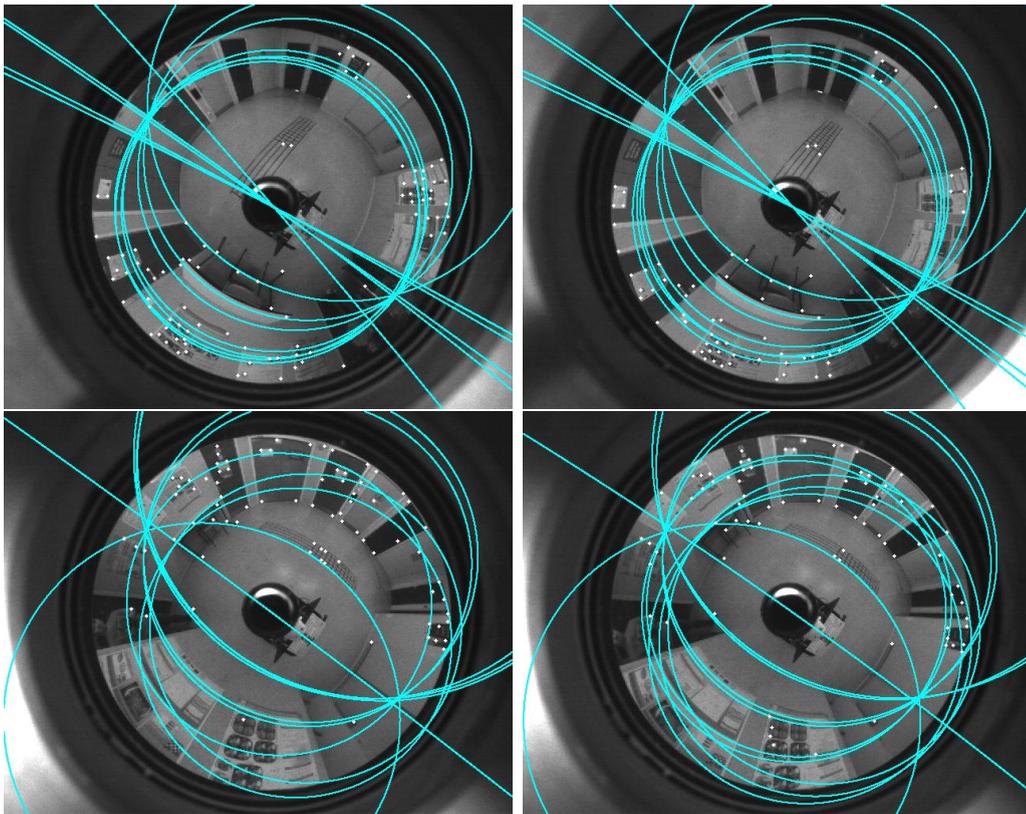


FIGURE 4.29 – Le suivi de points et droites dans quelques images de la séquence

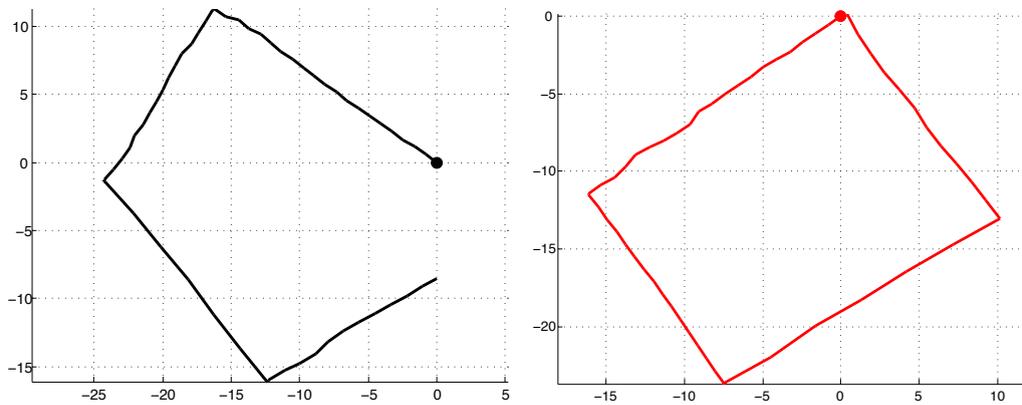


FIGURE 4.30 – Estimation de mouvement par l’algorithme des 5 points (gauche) et la méthode basée droites (droite)

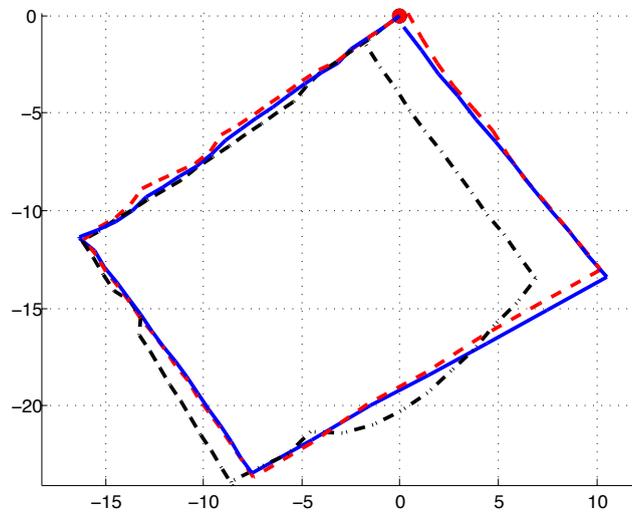


FIGURE 4.31 – Estimation de mouvement par la méthode basée droite (rouge), l’algorithme des 2 points (noir) et la combinaison de points et droites (bleu)

4.5 Conclusion

Dans ce chapitre, nous avons détaillé une méthode originale de SFM pour une caméra sphérique. Cette méthode utilise les avantages des images sphériques pour découpler le mouvement en estimant dans un premier temps la rotation puis la translation dans un second temps. La rotation est calculée à l’aide des points de fuite de la scène, toujours présents dans ce type d’image, et permet donc une estimation plus précise et robuste. Pour calculer cette rotation, trois approches ont été proposées en fonction des hypothèses sur les points de fuite : une transformée de Hough, une approche multirésolution et une approche "Bound and Branch". Une fois cette

rotation connue, le problème revient à estimer le déplacement dans le cas d'une translation pure. Cette translation pure peut alors être calculée à l'aide de seulement deux points de la scène et aboutit à une méthode rapide et robuste. Nous avons aussi montré que nous pouvions calculer la translation par une méthode linéaire en utilisant à la fois les points et les droites (détectées pour estimer la rotation). Ceci permet d'utiliser un nombre important de primitives disponibles dans l'environnement pour le calcul de SfM. En utilisant le modèle de projection unifié, la méthode proposée est applicable à tout type de caméras à point de vue unique (comme perspectives, catadioptriques centrales) ou encore caméras fish-eye. De plus, étant donné que la combinaison de la vision perspective avec la vision omnidirectionnelle devient de plus en plus intéressante (comme nous le verrons dans le chapitre suivant), l'approche proposée promet plusieurs applications. Elle peut être utilisée pour estimer la topologie d'un réseau de caméras hybrides ou le mouvement d'un robot mobile équipé d'un système de vision hybride ([LDV10b, LDV10a]).

Ces travaux ont été menés dans le cadre de trois projets, deux nationaux (ANR Blanc CaViAR, PEPS MOSCA) et un projet international (Hubert Curien, STAR). Ils ont donné lieu à la soutenance de la thèse de Dieu Sang Ly et de celle de Jean-Charles Bazin avec lequel j'ai collaboré durant sa thèse. Ces recherches se poursuivent dans le cadre d'un projet NRF-ANR Blanc International avec la Corée du Sud (DrAACaR) qui permet de continuer la collaboration avec nos collègues du KAIST qui avait débuté avec la thèse de Jean-Charles. Dans ce nouveau projet pour lequel je suis le coordinateur local pour le Le2i, nous nous intéressons à l'étude d'un banc de caméras sphériques asynchrones embarqué sur un véhicule.

Chapitre 5

Capteur sphérique pour la robotique aérienne

Ce chapitre vise à démontrer l'intérêt d'un capteur grand angle pour la robotique aérienne. Depuis 2005, j'ai participé à différents projets de recherche ayant trait à la robotique aérienne. De 2005 à 2008, j'ai participé au projet ANR Blanc CAVIAR (Catadioptric Vision for Aerial Robots) qui regroupait cinq laboratoires : le MIS de l'université de Picardie Jules Verne, Le LAAS CNRS de Toulouse, l'équipe AROBAS de l'INRIA Sophia-Antipolis, l'équipe PERCEPTION de l'INRIA Rhône-Alpes et le Le2i. Ce projet avait pour objectif d'étudier l'apport de la vision omnidirectionnelle pour les drones. C'est dans ce cadre qu'ont débuté les travaux sur l'estimation d'attitude d'un UAV à partir d'une caméra catadioptrique ([DVP06, DVP07]). De 2008 à 2009, j'ai participé à un projet STAR du programme Hubert Curien dénommé NOVA (Aide au pilotage de robot volant par vision omnidirectionnelle) auquel participaient le MIS, le Le2i et le laboratoire RCVLab du KAIST (Corée du Sud). Ce projet a consisté à équiper le drone d'une caméra catadioptrique et de transmettre au pilote au sol toutes les informations que celle-ci pouvait collecter sur le vol de l'UAV. Ce projet a permis de débiter la collaboration avec la Corée du Sud et de participer à l'encadrement de la thèse de Jean-Charles Bazin. De 2008 à 2011, le projet ALTO (Décollage et atterrissage automatiques de robots volants autonomes) financé par la région Picardie s'intéressait à la phase d'atterrissage et de décollage de l'avion sans pilote. Enfin, de 2009 à 2012, j'ai co-encadré une thèse "appui à l'émergence" (financement de jeune HDR de la région Picardie) obtenue par Pascal Vasseur sur l'estimation de l'altitude et de l'attitude du drone en vision nocturne. Ce sont ces deux thèmes que nous allons aborder par la suite. Dans les deux cas, nous avons choisi d'utiliser une caméra fish-eye couplée avec un autre capteur. Dans le projet ALTO, la thèse de Damien Eynard consistait à coupler la caméra fish-eye avec une caméra perspective pour en déduire les paramètres du vol alors qu'Ashutosh Natraj dont la thèse est financée par l'appui à l'émergence utilise une caméra fish-eye avec un laser. Ainsi après une brève introduction sur la vision pour la robotique aérienne, nous allons revenir en détail sur les travaux de Damien Eynard et d'Ashutosh Natraj.

5.1 Introduction

Les drones ou UAV (*Unmanned Aerial Vehicles*) sont de plus en plus présents de nos jours. Alors que durant le XXème siècle, leurs applications se limitaient à des tâches militaires, ils sont aujourd'hui utilisés dans des applications civiles. Par exemple, en 2008, les drones ont joué un rôle important pour la recherche et le sauvetage de victimes lors du passage de l'ouragan Katrina en Louisiane. En France, les forces de police surveillent certaines manifestations avec des drones (Fig 5.1(a)). En juillet 2009, les pompiers de Paris ont testé le drone Minirec (Fig 5.1(b)) de la société Bertin pour comptabiliser le nombre de personnes prises au piège lors d'un incendie d'immeuble. Les pompiers des Landes ont aussi fait très récemment l'acquisition de drones pour analyser en temps réel la progression des flammes lors des feux de forêt. Alors qu'à



FIGURE 5.1 – (a) Drone ELSA utilisé par le GIPN et les CRS, (b) Drone Minirec utilisé par les pompiers de Paris.

l'heure actuelle ces drones sont encore pilotés au sol, la communauté scientifique cherche à les rendre de plus en plus autonomes à travers différents projets de recherche. Au niveau national, on peut citer : ROVOLINT (Programme Robea) dont l'objectif était de réaliser une plateforme robotique volante destinée à des expériences de navigation autonome tridimensionnelle, RESSAC (ONERA) qui visait à rendre la navigation d'un hélicoptère Yamaha RMax la plus autonome possible, AEROB (Programme Robea) et R-Discover (Programme CONTINT de l'ANR) qui s'intéressent à la coopération entre engins terrestres et aériens, SCUAV (Programme Psirob de l'ANR) sur la stabilisation et l'asservissement visuel d'un drone VTOL (*Vertical Take-Off and Landing*), ou encore notre projet CaViAR (Programme Blanc –ANR) pour l'utilisation de la vision omnidirectionnelle pour la navigation en extérieur et en milieu urbain. Sur le plan international également, de très nombreux projets ont été ou sont développés dans le but d'accroître l'autonomie des robots volants : AURORA (Autonomous Unmanned Remote Monitoring Robotic Airship) pour l'observation, COMETS pour la surveillance et la détection coopérative entre robots volants hétérogènes, MuFly pour le développement d'un micro-hélicoptère autonome, PEGASE pour l'approche et le guidage autonomes sur piste, constituent une liste non-exhaustive des travaux menés sur ce sujet et démontrent l'intérêt et les enjeux de ce domaine. Ainsi, les drones ont reçu beaucoup d'attention au cours de la dernière décennie, par rapport à la commande et la vision par ordinateur de bord. L'intérêt pour ces sujets a été largement axé sur l'accroissement de l'autonomie UAV, qui comprend des manœuvres telles que l'atterrissage et le décollage.

Dans ces projets, un certain nombre de capteurs ont été utilisés pour estimer les paramètres de navigation du drone. L'altitude peut être déterminée par un GPS, un altimètre, un radar,

ou un laser. L'attitude peut être fournie par une centrale inertielle (IMU), et le mouvement par GPS ou par un radar. Néanmoins, les systèmes de positionnement global (GPS) offre une mauvaise précision à la fois verticalement (précision entre 25 et 50 mètres) et horizontalement (moins de 15 mètres). Ils sont de plus sensibles à la transmission des signaux dans les milieux urbains. Le radar peut estimer à la fois l'altitude et la vitesse avec plus de précision, mais il exige des capteurs actifs qui consomment de l'énergie. Les altimètres sont très largement utilisés mais dépendent fortement des variations de pression. Ce qui implique une erreur de précision comprise entre 6% et 7 % les rendant très imprécis en milieu fermé. Enfin, les IMU peuvent fournir des indications de vitesse, d'accélération, d'attitude et d'orientation, mais sont soumis à la dérive et à l'accumulation des erreurs. Les capteurs de vision permettent eux aussi de calculer les paramètres nécessaires au vol. D'une précision souvent plus faible que les capteurs cités précédemment ils peuvent malgré tout donner d'autres informations importantes sur la scène, sur les objets en présence. . . C'est pourquoi ces capteurs sont très utiles dans ce type d'applications. Différents types de capteurs ont d'ailleurs été proposés pour la navigation de drone : une seule caméra perspective [BNM01, BMZF06, CSZ04, GOSB03], un capteur stéréoscopique perspectif [WHH⁺05, MTB⁺10], catadioptrique [MCMO10]. Nous invitons à ce sujet le lecteur intéressé à se reporter à l'article suivant [SDMF12] pour un état de l'art des méthodes basées vision pour la navigation de drones. Dans ce qui suit et bien que nous nous sommes aussi intéressés à des capteurs catadioptriques pour les UAV ([DVP06, DVP07]), nous avons choisi d'évoquer nos solutions développées à partir de capteurs stéréoscopiques hybrides. Dans un premier temps, nous détaillerons le travail de Damien Eynard qui a effectué sa thèse dans le cadre du projet ALTO avec un capteur muni d'une caméra fish-eye et d'une caméra perspective puis nous évoquerons le travail d'Ashutosh Natraj qui a utilisé une caméra fish-eye et un laser circulaire.

5.2 Capteur stéréoscopique : Fish-eye/Perspectif

La caméra fish-eye offre un angle de vue proche de 180° mais au prix de fortes distorsions et d'une résolution non homogène sur l'ensemble de l'image. A l'inverse, une caméra perspective permet d'avoir une image de très grande résolution mais avec un champ de vue limité. Coupler les deux caméras permet de bénéficier des avantages de chacune. Par exemple, la caméra fish-eye permet d'estimer l'attitude du drone ([DVP07]) alors que la caméra perspective peut de son côté suivre des objets sur le sol. De plus, le fait d'utiliser un capteur binoculaire permet de calculer l'altitude métrique du drone contrairement aux systèmes monoculaires qui ne nous donne qu'une information à l'échelle.

Dans cette partie, nous allons montrer comment utiliser ce capteur stéréoscopique hybride (Fig. 5.2) pour estimer l'altitude et la translation subie par le drone pendant son déplacement. Nous supposons connue la rotation du robot volant (cette rotation peut être obtenue par la caméra fish-eye ou par IMU) et comme dans les chapitres précédents, la caméra fish-eye sera modélisée par son image sphérique équivalente. Notons que nous aurions pu modéliser les deux caméras par leur modèle sphérique équivalent et ainsi travailler dans des espaces de même nature. Mais la transformation de l'image perspective en image sphérique exige une phase d'interpolation qui engendre un temps de calcul plus important. C'est pourquoi nous avons préféré modéliser la

fish-eye par son image sphérique équivalente et la caméra perspective par son plan image (Figure 5.2).

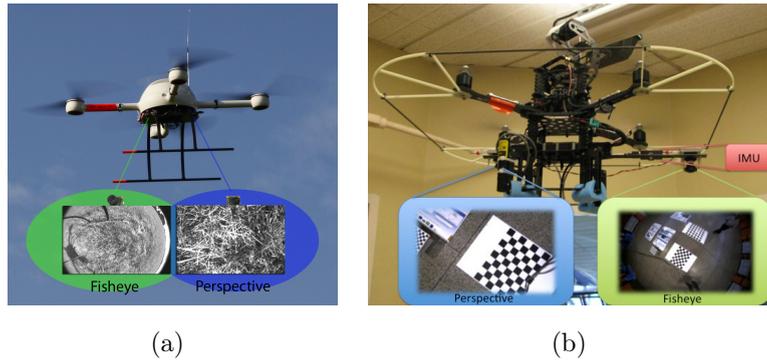


FIGURE 5.2 – (a) MD4-1000 (Microdrones GmbH) muni du capteur Fish-eye/perspective. (b) Pelican (Ascending Technologies) muni du capteur Fish-eye/perspective.

Pour bénéficier des avantages d'un capteur stéréoscopique, il faut étudier le lien entre la caméra sphérique et la caméra perspective. Dans cette application, étant donné que le capteur filme majoritairement le sol, on peut supposer que les points 3D se situent principalement sur le même plan. Lorsqu'un plan est filmé par deux caméras perspectives, il est bien connu qu'il existe une homographie entre les deux images [HZ04]. Dans le cas de deux images sphériques, Mei a démontré que cette propriété était toujours valide [MBMR06]. Dans notre cas (fig. 5.2), comme la fish-eye est considérée comme une image sphérique, la géométrie projective est respectée et il existe aussi une homographie entre la fish-eye et la caméra perspective :

$$\mathbf{H} = \mathbf{R}_c - \mathbf{t}_c \frac{\mathbf{n}^T}{d} \quad (5.1)$$

où \mathbf{R}_c et \mathbf{t}_c sont respectivement la rotation et la translation entre les deux caméras, \mathbf{n} et d représentent la normale et la distance du plan du sol par rapport à la caméra perspective. Ici, \mathbf{R}_c et \mathbf{t}_c sont connues grâce à un calibrage préalablement réalisé par la Toolbox Hyscas¹. \mathbf{n} est en fait l'attitude du drone que nous pouvons obtenir soit à l'aide d'une centrale inertielle soit par les algorithmes décrits dans le chapitre précédent. Ainsi, seule d qui traduit l'altitude du capteur est à déterminer. Bien évidemment, il suffirait d'un point mis en correspondance entre les deux images pour estimer d . Néanmoins, à cause de la forte hétérogénéité entre les deux images, il est difficile de mettre en correspondance les points entre ces images. C'est pourquoi nous avons opté pour une approche dite de "plane-sweeping" [Col96]. Cette approche consiste déterminer de manière exhaustive la meilleure valeur d en balayant l'ensemble des solutions possibles.

1. <http://www.hyscas.com/>

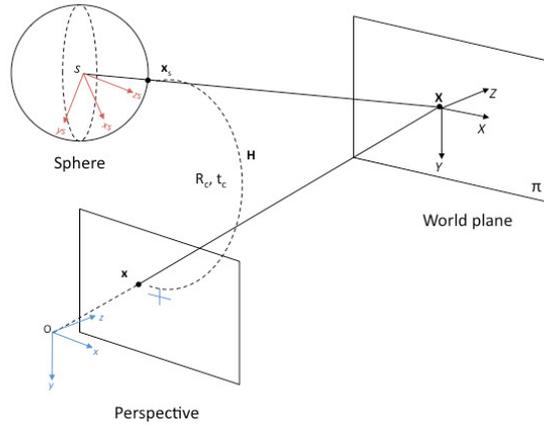


FIGURE 5.3 – Modélisation du capteur Fish-eye/perspective

5.2.1 Estimation de l'altitude du drone

Chaque valeur de d de (5.1) définit une homographie et chaque homographie permet de construire une image transformée \mathbf{I}^* de l'image fish eye \mathbf{I}_s originale :

$$\mathbf{I}^*(\mathbf{x}, d) = \mathbf{I}_s\left(\left(\mathbf{R}_c - \mathbf{t}_c \frac{\mathbf{n}^T}{d}\right)\mathbf{x}\right) \quad (5.2)$$

Le "plane-sweeping" consiste à comparer l'ensemble des images $\mathbf{I}^*(\mathbf{x}, d)$ à l'image issue de la caméra perspective \mathbf{I}_{ref} . La valeur de d permettant de réaliser le minimum de $|\mathbf{I}_{ref}(x) - \mathbf{I}^*(x, d)|$ pour les pixels x constituant le sol, correspond à l'altitude du drone. Il faut donc résoudre le problème d'optimisation suivant :

$$\begin{aligned} \hat{d} &= \operatorname{argmin}_d \left(\sum_{\mathbf{x} \in \mathbf{G}} |\mathbf{I}_{ref}(\mathbf{x}) - \mathbf{I}^*(\mathbf{x}, d)| \right) \\ \mathbf{G} &= \left\{ \mathbf{x}, \frac{\sum_{\mathbf{x} \in W_{\mathbf{x}}} |\mathbf{I}_{ref}(\mathbf{x}) - \mathbf{I}^*(\mathbf{x}, \hat{d})|}{\sum_{\mathbf{x} \in W_{\mathbf{x}}} (\mathbf{I}_{ref}(\mathbf{x}))} < \delta \right\} \end{aligned} \quad (5.3)$$

où $W_{\mathbf{x}}$ est le voisinage du pixel x et δ un seuil fixe. Notons que nous avons choisi l'image perspective comme image de référence pour travailler dans le plan euclidien plutôt que sur la sphère où le voisinage est plus difficile à calculer comme nous l'avons vu au chapitre 2.

Pour déterminer la solution de 5.3, le "plane-sweeping" balaye l'ensemble des solutions possibles de manière exhaustive comme détaillé dans l'algorithme 3 dans lequel d_{min} et d_{max} définissent l'intervalle de recherche, s le pas de discrétisation et Δd le critère de convergence. En pratique, pour l'estimation de l'altitude à l'instant t , l'intervalle de recherche est mise à jour autour de l'estimation d_{t-1} à l'instant $t - 1$ et des limites physiques du drone.

5.2.2 Estimation du déplacement du drone à l'aide d'un capteur stéréo hybride

A cette étape, nous connaissons l'attitude et l'altitude du drone. Il nous reste à déterminer le vecteur de translation entre deux prises d'images du capteur. Comme pour l'estimation de

Algorithm 3 Estimation de l'altitude et segmentation du sol**Estimation**($d_{min}, d_{max}, s, \Delta d$)*{Initialisation}*

$$a_0 = \widehat{d}_{-1} = d_{min}$$

$$b_0 = \widehat{d}_0 = d_{max}$$

 $\mathbf{G} = \{\text{pixels de l'image de référence}\}$ **while** $|\widehat{d}_k - \widehat{d}_{k-1}| > \Delta d$ **do***{Estimation de la meilleure altitude d }*

$$\widehat{d}_{k+1} = \operatorname{argmin}_{d \in \{\frac{b_k - a_k}{s-1} t + a_k; t \in [0, s-1]\}} \sum_{\mathbf{x} \in \mathbf{G}} |\mathbf{I}_{ref}(\mathbf{x}) - \mathbf{I}^*(\mathbf{x}, d_k)|$$

{Estimation du sol}

$$\mathbf{G} = \{\mathbf{x}, \frac{\sum_{\mathbf{x}_1 \in W_{\mathbf{x}}} |I_{ref}(\mathbf{x}_1) - \mathbf{I}^*(\mathbf{x}_1, \widehat{d})|}{\sum_{\mathbf{x}_1 \in W_{\mathbf{x}}} I_{ref}(\mathbf{x}_1)} < \delta\}$$

$$a_{k+1} = \widehat{d}_{k+1} - \frac{b_k - a_k}{s-1}$$

$$b_{k+1} = \widehat{d}_{k+1} + \frac{b_k - a_k}{s-1}$$

$$k = k + 1$$

end while**Return** \widehat{d}

l'altitude, pour éviter une étape de mise en correspondance entre des images très différentes (perspective et fish-eye), nous avons choisi d'estimer cette translation à partir des deux caméras de façon indépendante grâce à l'algorithme présenté au chapitre précédent.

Notons \mathbf{R}_{t+1} la rotation du drone à l'instant $t + 1$, $\mathbf{x}_{s_i, t}$ et $\mathbf{x}_{s_i, t+1}$ les points sur la sphère aux instants t et $t + 1$ (respectivement $\mathbf{x}_{p_i, t}$ et $\mathbf{x}_{p_i, t+1}$ pour les points de la caméra perspective) et \mathbf{t}_{t+1} la translation à estimer (fig. 5.2.2). Dans le cas de points quelconques de l'environnement, nous pouvons reprendre l'algorithme du chapitre précédent en tenant compte du changement de repère entre les deux caméras :

$$\mathbf{C}_h \mathbf{t}_{t+1} = \mathbf{D}_h \quad (5.4)$$

où

$$\mathbf{C}_h = \begin{bmatrix} \mathbf{R}_{t+1} \mathbf{x}_{s_0, t} \times \mathbf{x}_{s_0, t+1} \\ \vdots \\ \mathbf{R}_{t+1} \mathbf{x}_{s_m, t} \times \mathbf{x}_{s_m, t+1} \\ \mathbf{R}_{t+1} \mathbf{x}_{p_0, t} \times \mathbf{x}_{p_0, t+1} \\ \vdots \\ \mathbf{R}_{t+1} \mathbf{x}_{p_n, t} \times \mathbf{x}_{p_n, t+1} \end{bmatrix} \quad (5.5)$$

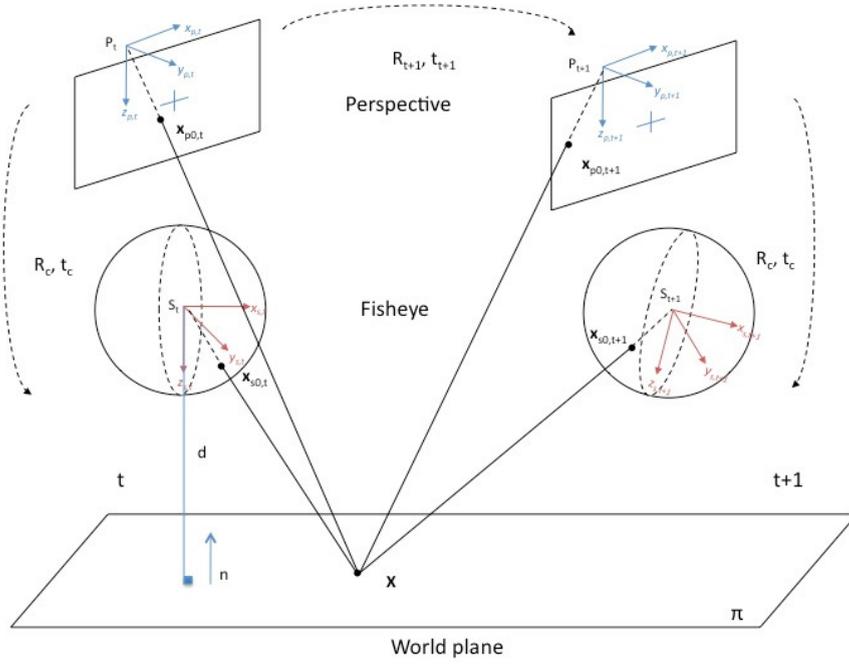


FIGURE 5.4 – Position du capteur entre deux prises de vues

$$\mathbf{D}_h = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (\mathbf{R}_{t+1}\mathbf{x}_{p_0,t} \times \mathbf{x}_{p_0,t+1}) \cdot (\mathbf{R}_{t+1}\mathbf{t}_c - \mathbf{t}_c) \\ \vdots \\ (\mathbf{R}_{t+1}\mathbf{x}_{p_n,t} \times \mathbf{x}_{p_n,t+1}) \cdot (\mathbf{R}_{t+1}\mathbf{t}_c - \mathbf{t}_c) \end{bmatrix} \quad (5.6)$$

$$\mathbf{A}_h \mathbf{t}_{t+1} = \mathbf{B}_h \quad (5.7)$$

Par contre, pour les points du sol, la position 3D est connue et vérifie donc l'équation suivante :

$$(\mathbf{x}_{a,t+1} \times \mathbf{t}_{t+1}) = -d_a (\mathbf{x}_{a,t+1} \times \mathbf{R}_{t+1}\mathbf{x}_{a,t}) \quad a = s, p \quad \mathbf{x}_{a,\cdot} \in \mathbf{G} \quad (5.8)$$

qui se traduit sous forme matricielle par :

$$\mathbf{A}_a \mathbf{t}_{t+1} = \mathbf{B}_a \quad a = s, p \quad (5.9)$$

où

$$\mathbf{A}_a = \begin{bmatrix} 0 & -x_{a,t+1} & x_{a,t+1} \\ z_{a,t+1} & 0 & -x_{a,t+1} \\ -y_{a,t+1} & x_{a,t+1} & 0 \end{bmatrix} \quad (5.10)$$

$$\mathbf{B}_s = -d_s (\mathbf{x}_{s,t+1} \times \mathbf{R}_{t+1}\mathbf{x}_{s,t})^T \quad (5.11)$$

$$\mathbf{B}_p = -d_p ((\mathbf{x}_{p,t+1} \times \mathbf{R}_{t+1}\mathbf{x}_{p,t}) + \mathbf{x}_{p,t+1} \times ((\mathbf{R}_{t+1} - \mathbf{I})\mathbf{t}_c))^T \quad (5.12)$$

L'ensemble de ces systèmes est ensuite concaténé pour obtenir la valeur de \mathbf{t}_{t+1} par moindres carrés couplés à un RANSAC.

Dans le paragraphe suivant, nous présentons les résultats obtenus à l'aide de notre approche.

5.2.3 Résultats

Altitude

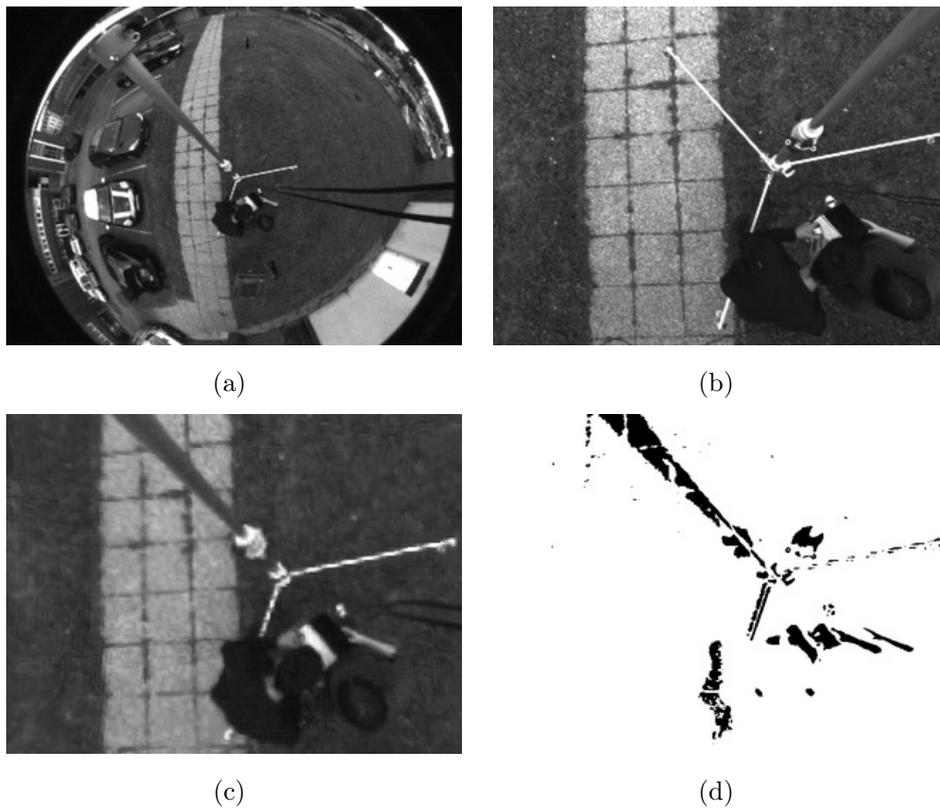


FIGURE 5.5 – Images test (a) vue fish-eye, (b) vue perspective, (c) image fish-eye projetée dans l'image perspective après le "plane-sweeping", (d) segmentation du sol (en blanc)

Pour estimer l'altitude à partir du "plane-sweeping", il est nécessaire de connaître l'attitude du capteur. Ainsi, la précision de l'altitude est fortement dépendante de celle de l'attitude. Les approches développées dans le chapitre précédent peuvent nous aider à calculer l'attitude du capteur puisque celui-ci est muni d'une caméra sphérique. Nous avons montré qu'avec une caméra fish-eye, l'estimation de l'assiette à l'aide des points de fuite [DVP07] ou par détection de l'horizon [DVP06] engendre des erreurs inférieures à 3° ce qui n'influence que très légèrement l'estimation de l'attitude ([EVDF12]). Ces méthodes sont donc parfaitement appropriées et permettent donc de s'affranchir d'une centrale inertielle.

Dans un premier temps, nous avons monté un capteur hybride muni d'une caméra perspective et d'une fish-eye distante de 447 mm sur un bras télescopique (fig. 5.5). Les attitudes estimées sont présentées dans le tableau 5.1 et comparées à la vérité terrain obtenue grâce à un télémètre laser. Un résultat est présenté dans la figure 5.5.

Altitude réelle	Altitude estimée
2187mm	2200mm
3244mm	3250mm
4072mm	4200mm
5076mm	5202mm

TABLE 5.1 – Comparaison de notre méthode avec un capteur hybride monté sur un bras télescopique (fig 5.5) avec $s = 6$ et $\delta = 25$

Nous avons par la suite embarqué un capteur hybride dont la *baseline* était de 314mm sur un drone Pelican (fig 5.2(b)) où les calculs étaient effectués en ligne (fig. 5.6(d)) à la vitesse de 80 hertz². Même si dans ce cas, nous ne disposons pas de la vérité terrain, l'estimation semble parfaitement cohérente avec le déplacement du drone durant le vol.

Estimation du mouvement

Pour valider la méthode d'estimation de la translation, nous avons utilisé un robot Kuka (fig. 5.7(b)) grâce auquel nous connaissons parfaitement le déplacement de notre capteur. La trajectoire accomplie par le robot est un carré de 33 cm de côté à une hauteur de 1 m. La figure 5.7(c) compare la méthode proposée à la vérité terrain en rouge. La trajectoire bleue correspond à l'estimation de la translation où la rotation est connue par IMU et la courbe verte représente la trajectoire estimée dans le cas où la rotation est déduite de la vérité terrain. Nous noterons ici une légère dérive de la trajectoire déduite de l'IMU due à la mauvaise précision de l'IMU employée.

Enfin, la figure 5.8 présente la trajectoire estimée du Pelican à l'aide de notre méthode. Malheureusement, nous ne disposons pas dans ce cas de vérité terrain mais la trajectoire tracée reflète bien le comportement du drone lors du vol.

Nous venons de présenter une méthode utilisant une caméra sphérique pour la navigation d'un drone. Cette caméra couplée avec une caméra perspective classique permet d'estimer l'attitude, l'altitude et le déplacement du drone. Nous allons voir par la suite une autre solution que nous avons proposée utilisant une caméra sphérique avec une laser circulaire. Ce travail a été effectué par Ashutosh Natraj durant sa thèse financée par la région Picardie.

2. <http://www.youtube.com/watch?v=ubXzf0eLud4>

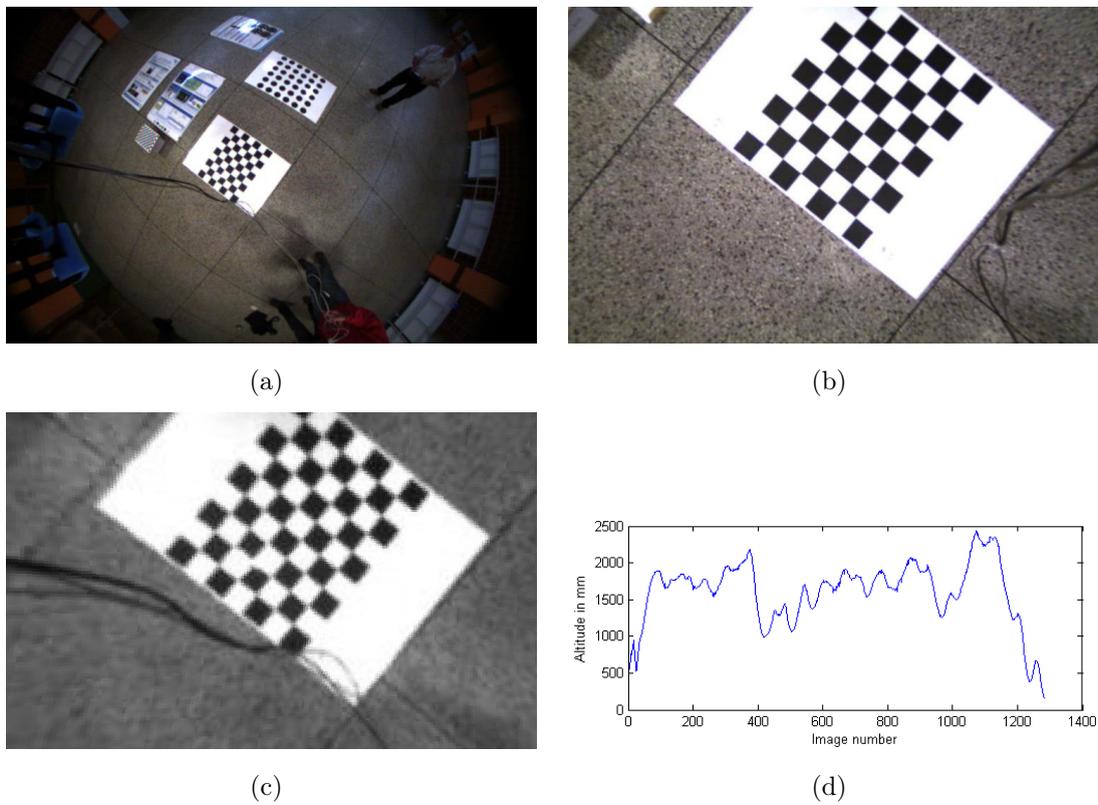


FIGURE 5.6 – Capteur hybride embarqué sur un UAV (a) vue fish-eye (b) vue perspective (altitude 1912mm), (c) image fish-eye projetée dans l'image perspective après la phase de "plane-sweeping", (d) calcul de l'altitude en ligne.

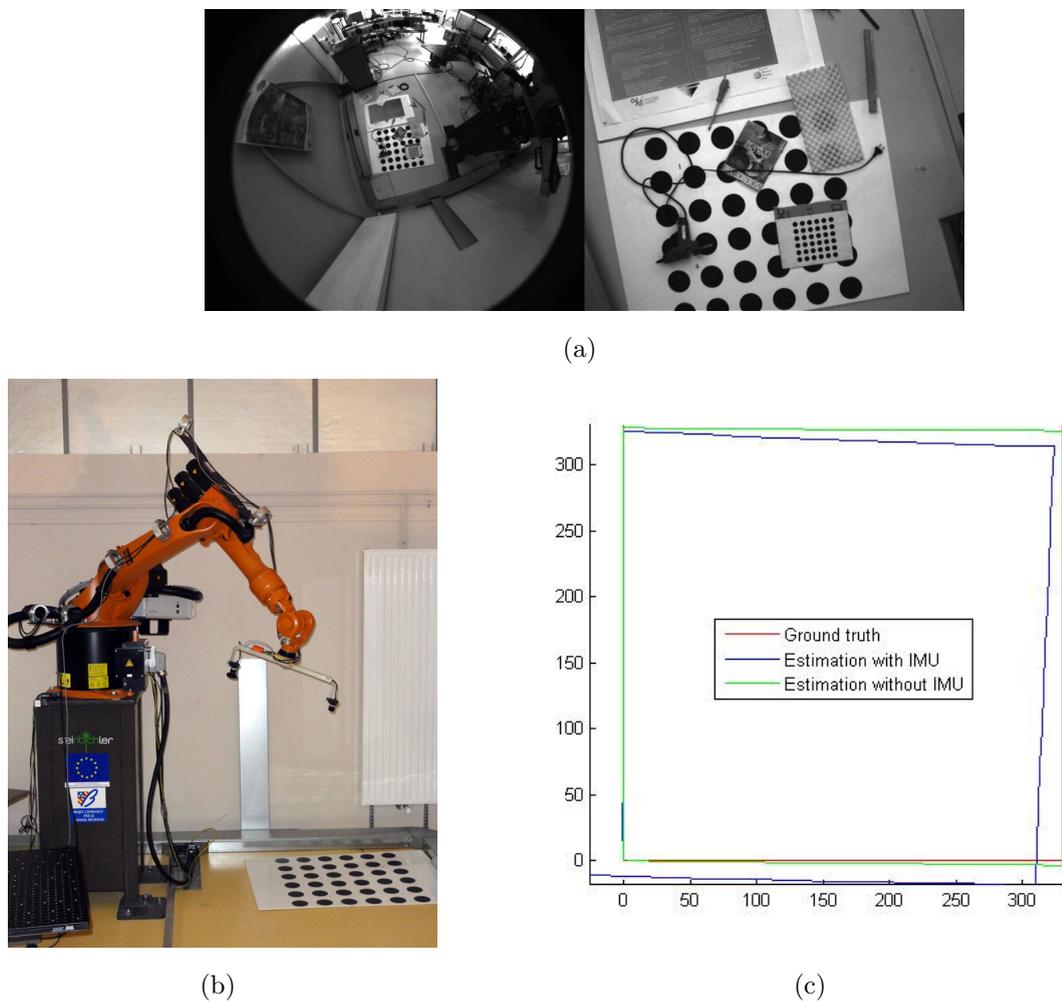


FIGURE 5.7 – Validation des algorithmes sur un robot Kuka. (a) images du capteur, (b) robot Kuka muni du capteur, (c) estimation de la trajectoire.

5.3 Capteur Fish-eye/Laser

Dans la section précédente, nous avons montré comment calculer l'altitude et l'attitude d'un drone grâce à un capteur stéréoscopique composé d'une caméra perspective et d'une caméra fish-eye. Ici, nous utiliserons un autre système constitué d'une caméra fish-eye et d'un laser circulaire. Ce système ne pourra pas estimer le déplacement du robot volant puisque l'image perçue par la caméra est invariante pour tout déplacement parallèle au sol plan. Par contre, ce capteur a l'avantage de fonctionner en milieu peu éclairé voire dans le noir absolu. Il s'agit ici de déduire l'altitude et l'attitude du drone à partir des déformations du projecteur laser dans l'image fish-eye. Pour ce faire, nous allons, comme dans l'ensemble de ces travaux, considérer l'image fish-eye par sa modélisation sphérique équivalente. Puis nous montrerons comment l'inclinaison et la hauteur du capteur par rapport au sol induisent une modification de l'image du laser sur la sphère.



FIGURE 5.8 – Trajectoire estimée du Pelican.

5.3.1 Modélisation analytique

Notons X_L , X_C et X_s respectivement les coordonnées d'un point 3D dans les repères du Laser, de la Caméra et de la Sphère. Soit π le plan du sol d'équation

$$X_L \cdot \vec{n} + d = 0. \quad (5.13)$$

dans le repère du Laser. \vec{n} et d correspondent respectivement à l'attitude du capteur et à sa hauteur relativement au sol. Ce sont ces valeurs que nous recherchons par la suite. En pratique, nous disposons de l'image du laser sur la caméra sphérique. Il s'agit de déterminer comment $\vec{n} = [n_1, n_2, n_3]^T$ et d modifient cette image.

La projection du laser sur le sol définit un cône de révolution d'équation :

$$x_L^2 + y_L^2 = z_L^2 \tan^2 \theta, \quad (5.14)$$

où θ est l'angle d'ouverture du laser supposé connu par calibrage. Les points 3D définis dans le repère du Laser ont pour équation $X_C = RX_L + b$ dans le repère caméra. (R et b sont eux aussi connus par calibrage intrinsèque). Pour simplifier les calculs, nous supposons par la suite que $R = I$ et $b_1 \neq 0, b_2 = b_3 = 0$. Nous avons donc :

$$X_L = X_C - b. \quad (5.15)$$

Ainsi, le plan π a pour équation dans le repère caméra :

$$(X_C - b) \cdot \vec{n} + d = 0, \quad (5.16)$$

Soit :

$$x_C n_1 - b_1 n_1 + y_C n_2 + z_C n_3 + d = 0 \quad (5.17)$$

Or, la relation entre le repère de la sphère et de la caméra est donnée par un scalaire λ :

$$X_s = \lambda X_c \quad (5.18)$$

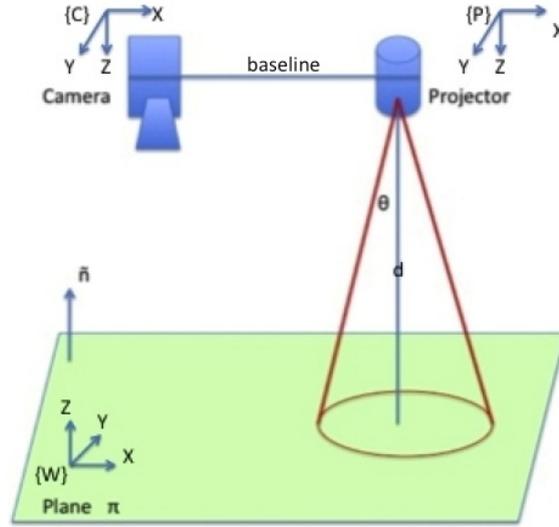


FIGURE 5.9 – Système fishe eye/ Laser circulaire

où

$$\begin{cases} x_s = \lambda x_c \\ y_s = \lambda y_c \\ z_s = \lambda z_c \end{cases}$$

Et par substitution dans l'équation (5.17) on déduit :

$$x_s n_1 - \lambda b_1 n_1 + y_s n_2 + z_s n_3 + \lambda d = 0 \quad (5.19)$$

Par conséquent,

$$\lambda = \frac{-(x_s n_1 + y_s n_2 + z_s n_3)}{d - b_1 n_1} \quad (5.20)$$

De plus, d'après les équations (5.15) et (5.14), la conique a pour équation dans le repère caméra :

$$(x_c - b_1)^2 + (y_c - b_2)^2 - (z_c - b_3)^2 \tan^2 \theta = 0 \quad (5.21)$$

Soit

$$\begin{aligned} (x_s - \lambda b_1)^2 + y_s^2 - z_s^2 \tan^2 \theta &= 0 \\ x_s^2 + \lambda^2 b_1^2 - 2\lambda b_1 x_s + y_s^2 - z_s^2 \tan^2 \theta &= 0 \end{aligned} \quad (5.22)$$

dans le repère de la sphère. Enfin en substituant la valeur de λ dans (5.22), on obtient :

$$\begin{aligned} d^2 x_s^2 + [n_2^2 b_1^2 + (d - b_1 n_1)^2] y_s^2 \\ + [n_3^2 b_1^2 - (d - b_1 n_1)^2 \tan^2 \theta] z_s^2 + [2b_1 n_2 d] x_s y_s \\ + [2b_1 n_3 d] x_s z_s + [2b_1^2 n_2 n_3] y_s z_s = 0 \end{aligned} \quad (5.23)$$

Cette équation caractérise l'image du laser sur la sphère en fonction de l'altitude d et de l'attitude \vec{n} . Il suffit de comparer cette équation avec la véritable image fournie par l'image sphérique pour en déduire ces paramètres.

Considérons l'image sphérique du laser, tous ses points $(x_s, y_s, z_s) \in \mathcal{S}^2$ vérifient l'équation d'une quadrique du type ([NDVS11]) :

$$Ax_s^2 + By_s^2 + Cz_s^2 + 2Dx_sy_s + 2Ex_sz_s + 2Fy_sz_s = 0, \quad (5.24)$$

où les six coefficients peuvent être déterminés par RANSAC.

Nous avons donc en comparant (5.23) et (5.24)

$$\frac{B}{A} = \frac{[n_2^2 b_1^2 + (d - b_1 n_1)^2]}{d^2} \quad (5.25)$$

$$\frac{C}{A} = \frac{[n_3^2 b_1^2 - (d - b_1 n_1)^2 \tan^2 \theta]}{d^2} \quad (5.26)$$

$$\frac{D}{A} = \frac{b_1 n_2}{d} \quad (5.27)$$

$$\frac{E}{A} = \frac{b_1 n_3}{d} \quad (5.28)$$

$$\frac{F}{A} = \frac{b_1^2 n_2 n_3}{d^2} \quad (5.29)$$

$$\frac{B}{A} = \frac{n_2^2 b_1^2}{d^2} + \left(1 - \frac{b_1 n_1}{d}\right)^2 \quad (5.30)$$

$$\frac{B}{A} = \left(\frac{D}{A}\right)^2 + \left(1 - \frac{b_1 n_1}{d}\right)^2 \quad (5.31)$$

Posons $\frac{b_1 n_1}{d} = G$, G est la solution de :

$$\frac{B}{A} = \left(\frac{D}{A}\right)^2 + (1 - G)^2, \quad (5.32)$$

soit :

$$G = \left(1 \pm \sqrt{\frac{B}{A} - \left(\frac{D}{A}\right)^2}\right) \quad (5.33)$$

A partir de (5.27), (5.28), nous avons :

$$G^2 + \left(\frac{D}{A}\right)^2 + \left(\frac{E}{A}\right)^2 = \left(\frac{b_1}{d}\right)^2. \quad (5.34)$$

Ainsi, l'altitude a pour formule :

$$d^2 = \frac{(b_1 * A)^2}{(G * A)^2 + D^2 + E^2} \quad (5.35)$$

et l'attitude $\vec{n} = (n_1, n_2, n_3)^T$ vérifie :

$$\begin{aligned} n_1 &= \frac{Gd}{b_1}, \\ n_2 &= \frac{D}{A} \frac{d}{b_1}, \\ n_3 &= \frac{E}{A} \frac{d}{b_1}. \end{aligned} \quad (5.36)$$

Notons qu'il existe deux solutions pour G et donc pour d et \vec{n} . Ces deux solutions correspondent aux deux plans d'intersection entre les deux cônes définies à partir de la caméra et du laser. En pratique, nous retenons la solution pour laquelle \vec{n} définit le plan le plus horizontal par rapport à la *baseline* entre la caméra et le laser (figure 5.10).

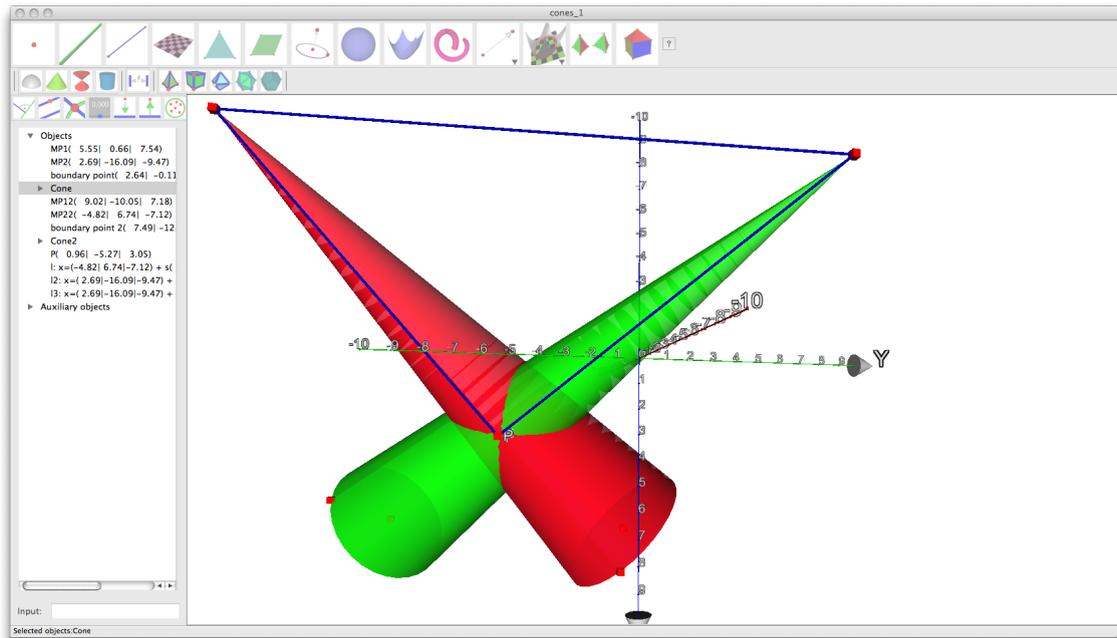


FIGURE 5.10 – Intersection des deux cônes de vues.

5.3.2 Resultats expérimentaux

Cette approche a été validée sur des données de synthèse pour en vérifier la robustesse et sur des séquences réelles. Dans le cas synthétique, nous avons donc simulé notre capteur sous Matlab (figure 5.11). Le tableau 5.2 présente des résultats dans le cas idéal où aucune donnée n'est bruitée. Le tableau 5.3 présente des exemples d'estimation dans le cas où les points de l'image fish-eye ont été bruités avec un bruit blanc d'écart type 0.5. Ces données montrent une erreur moyenne dans le cas idéal de 4,7 cm pour l'altitude, de $1,02^\circ$ pour le roulis et de $0,53^\circ$ pour le tangage. Dans le cas bruité, les erreurs augmentent légèrement, 7,8 cm pour l'altitude, $1,23^\circ$ pour le roulis et $1,26^\circ$ pour le tangage. Cette robustesse de la méthode s'explique par le RANSAC dans la phase d'estimation de coefficients de la quadrique (5.24). La figure 5.12 montre le comportement de la méthode proposée pour une pose simulée de l'UAV à 150 cm d'altitude, un roulis 6° et un tangage 10° en fonction du bruit introduit sur les points de la caméra.

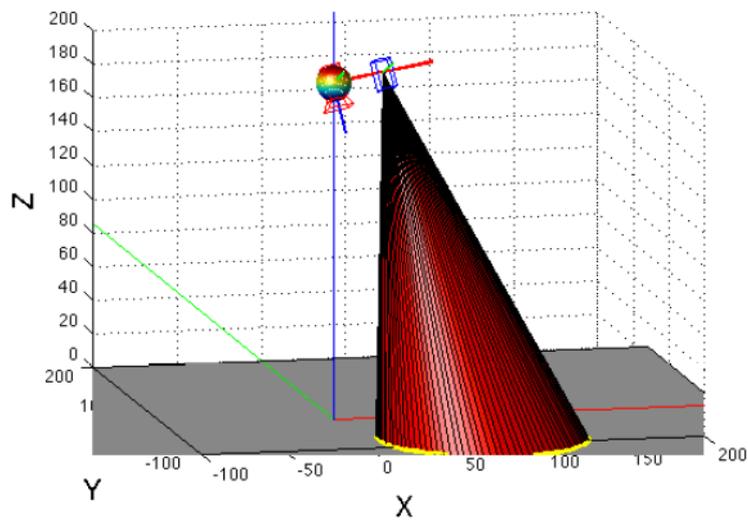


FIGURE 5.11 – Représentation de la simulation

TABLE 5.2 – Cas synthétique

	Altitude (m)	Roulis (deg)	Tangage (deg)
Vérité terrain	1	0	0
Valeur estimée	1	0	0
Vérité terrain	2	15	1
Valeur estimée	2	13.02	0.03
Vérité terrain	2	0	10
Valeur estimée	1.98	0.02	9.19
Vérité terrain	1	16	24
Valeur estimée	1.07	14.32	22.16
Vérité terrain	1	30	0
Valeur estimée	0.9	30.16	0.07
Vérité terrain	1	5	9
Valeur estimée	1.04	30.16	0.07
Vérité terrain	1	30	12
Valeur estimée	0.901	26.640	11.980

Les premiers tests réels ont été réalisés à partir de séquences d'images acquises par notre capteur tenu à la main (Figure 5.13) dont la caméra fish-eye est séparée d'environ 30 cm du laser. Sur le dispositif, nous avons ajouté une centrale inertielle et un télémètre pour pouvoir comparer avec une vérité terrain. Les résultats sont présentés figure 5.14. Les erreurs moyennes sont respectivement de 7, 14 cm pour une altitude variant entre 50 cm et 2 m, de 2, 03° en roulis et de 1, 65° en tangage. Enfin, nous avons embarqué le système sur un quadrirotor (Figure 5.2(a)).

TABLE 5.3 – Cas synthétique bruité avec un bruit blanc d'écart type 0.5

	Altitude(m)	Roulis (deg)	Tangage (deg)
Vérité terrain	1.2	0	0
Valeur estimée	1.14	0	0
Vérité terrain	7	0	0
Valeur estimée	6.34	0	0
Vérité terrain	2	0	1
Valeur estimée	1.99	0.02	0.05
Vérité terrain	8	12	10
Valeur estimée	7.68	13.56	7.19
Vérité terrain	1	30	10
Valeur estimée	0.93	27.38	10.71
Vérité terrain	2	20	15
Valeur estimée	1.89	17.38	13.71
Vérité terrain	2	12	30
Valeur estimée	1.91	10.15	26.93

Les valeurs obtenues sont comparées avec un télémètre et une centrale inertielle et présentées figure 5.15.

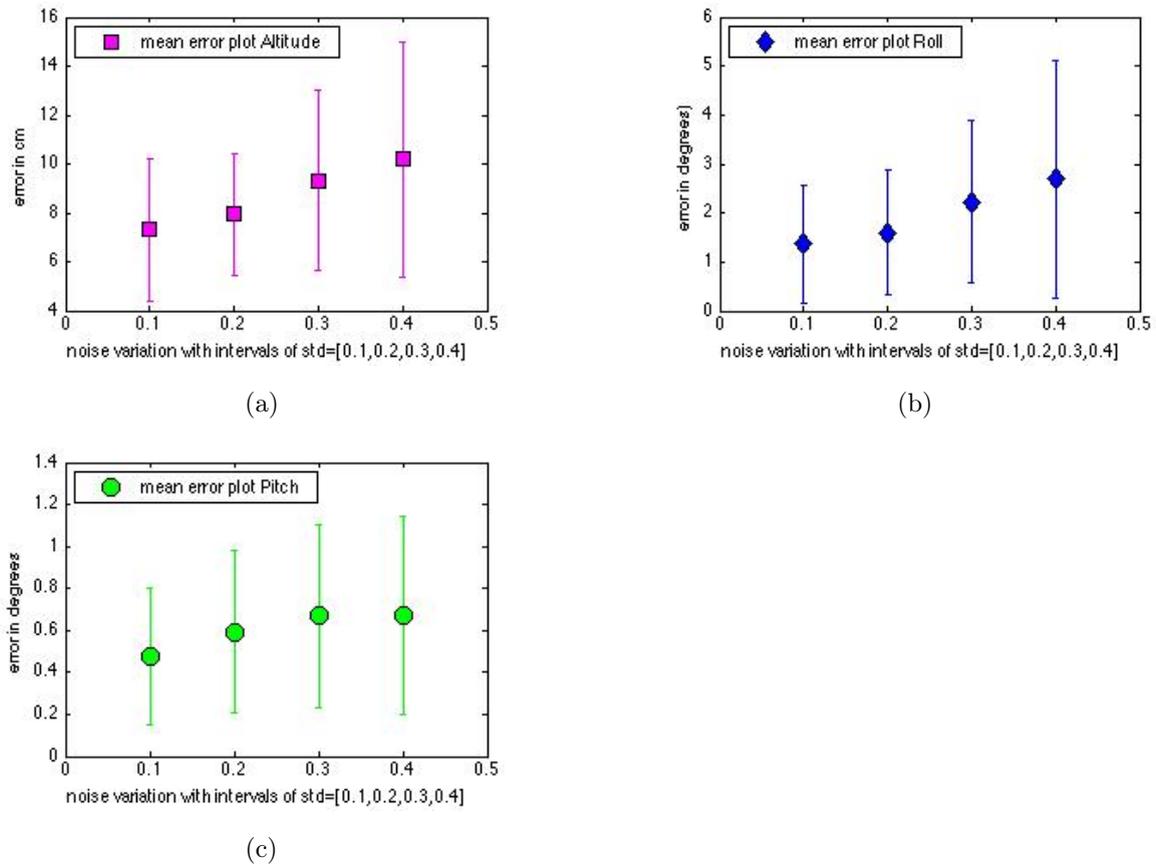


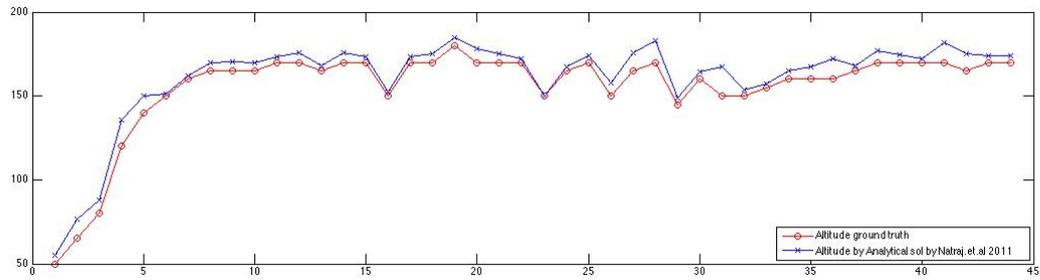
FIGURE 5.12 – Erreur moyenne sur le calcul de l’altitude (a), du roulis (b) et du tangage (c) en fonction du bruit dans l’image.



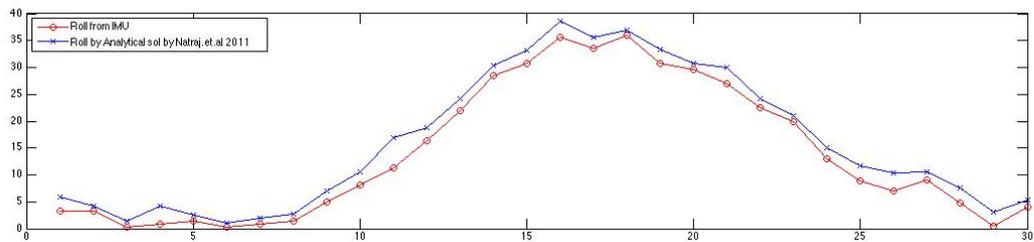
FIGURE 5.13 – Images issues d’une séquence réelle.

5.4 Conclusion

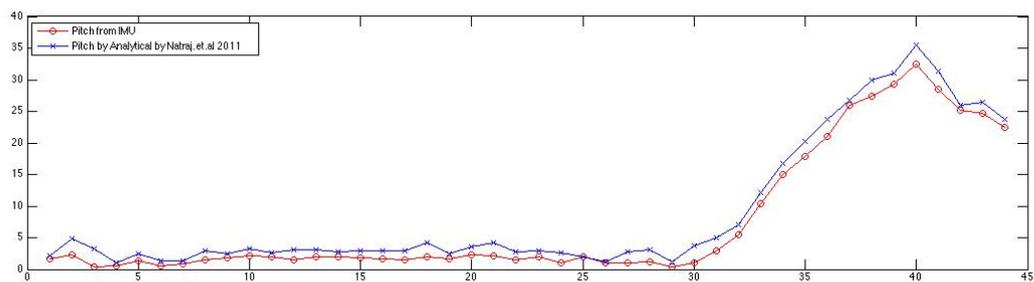
Nous avons présenté ici deux capteurs stéréoscopiques hybrides pour la robotique aérienne. Nous avons montré une fois de plus comment l’équivalence entre les images grand angle et les



(a)



(b)



(c)

FIGURE 5.14 – Estimation de l'altitude, du roulis et du tangage à l'aide de notre capteur tenu à la main.

images sphériques pouvait nous aider à estimer certains paramètres de navigation d'un drone.

Dans le premier cas, l'association entre une caméra fish-eye et une caméra perspective a montré que nous pouvions obtenir les six degrés de liberté du drone. La caméra fish-eye est capable d'estimer la rotation du robot volant, l'altitude est calculée grâce au "plane-sweeping" et la translation en fusionnant les informations acquises par les deux caméras. La méthode globale proposée utilise donc toutes les informations à sa disposition, tout en évitant les mises en correspondance de points entre images hétérogènes. Le deuxième capteur présenté dans ce chapitre, consiste en l'association d'une caméra fish-eye et d'un laser circulaire pour estimer l'altitude et l'assiette du drone en milieu nocturne. La méthode proposée déduit les paramètres de vol à l'aide de la déformation sur l'image fish-eye du cercle projeté au sol.

Ces travaux ont donné lieu à plusieurs publications ainsi qu'à la soutenance de la thèse de

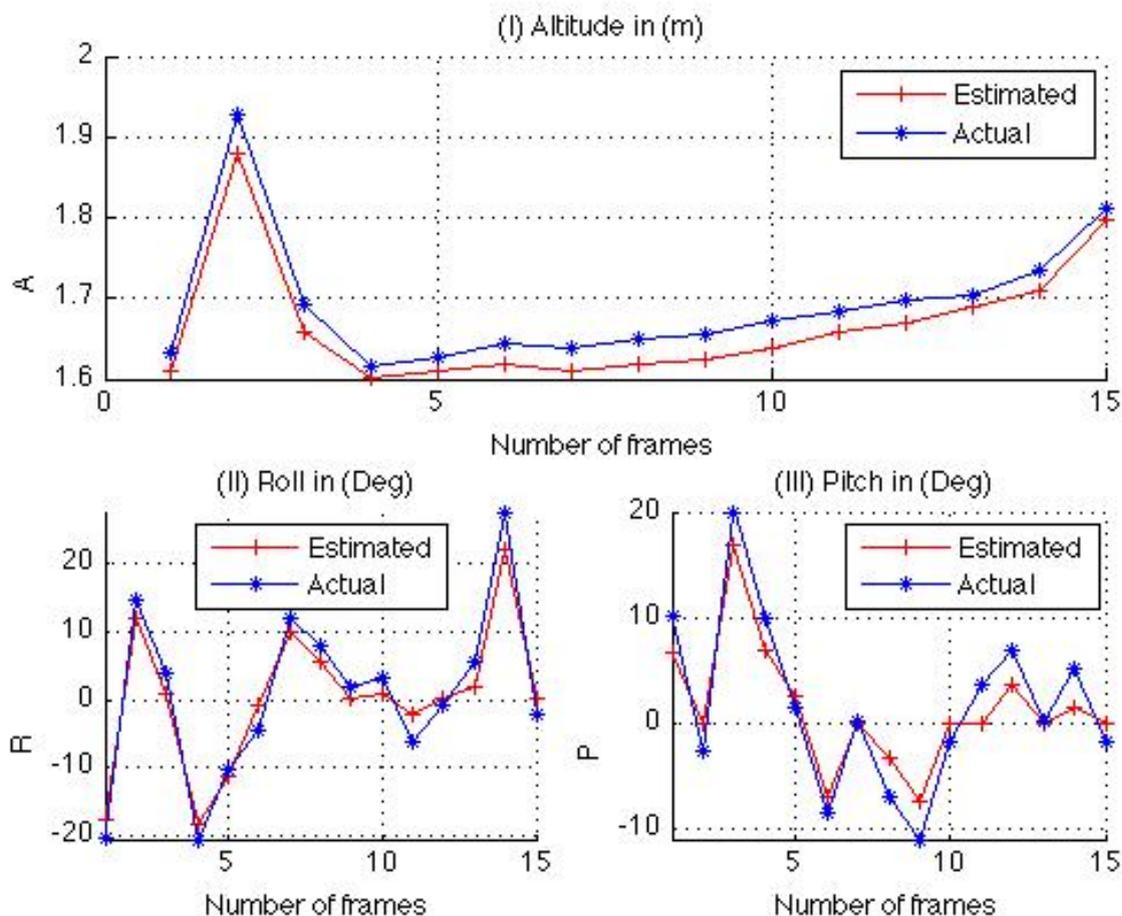


FIGURE 5.15 – Estimation de l’altitude, du roulis et du tangage à l’aide de notre capteur embarqué sur un quadrirotor.

Damien Eynard qui sera prochainement suivie par celle d’Ashutosh Natraj. L’étude de capteurs atypiques pour les drones continue puisqu’une thèse DGA/région de Bourgogne a débuté en septembre 2011 sur un capteur muni d’une caméra fish-eye et d’une caméra PTZ. Il s’agit dans ce projet d’utiliser la caméra fish-eye pour observer les lieux survolés par le drone et d’utiliser la caméra PTZ pour analyser les objets suspects.

Bilan et perspectives

Ce document synthétise mes sept premières années d'activités de recherche. Alors que durant ma thèse (2001-2004) je me suis intéressé à l'analyse du mouvement dans des séquences d'images perspectives, j'ai concentré mes travaux sur les caméras omnidirectionnelles à partir de ma nomination en septembre 2005 en tant que Maître de Conférences. D'un point de vue quantitatif, ces recherches ont conduit à la publication de onze revues à comité de lecture, d'une trentaine de conférences internationales et trois soutenances de thèses. Elles ont été menées dans le cadre de quatre projets internationaux, trois projets nationaux et deux projets régionaux. Tous ces travaux ont pour fil conducteur l'imagerie sphérique. Ils ont pour objectif de mettre en évidence l'intérêt de cette représentation pour les caméras à point de vue unique et fish-eye. Comme nous l'avons choisi dans ce manuscrit, ils peuvent se classer en trois catégories : traitement des images, *structure from motion* d'une caméra sphérique et applications en robotique. Nous allons rappeler brièvement les contributions et proposer des perspectives pour chacune d'entre elle.

Traitement des images grand angle

A cause des fortes distorsions engendrées par l'ajout d'un miroir ou par l'utilisation d'une optique fish-eye, les méthodes usuelles de traitement des images ne peuvent être directement appliquées sur ces images. Grâce à l'équivalence sphérique, on peut bien évidemment considérer ces images comme des images sphériques et utiliser les outils d'analyse harmonique sphérique [DH94] comme ont proposé Daniilidis et al. [DMB02], Bülow [Bül02], Bigot et al. [BKD08] ou encore Hadj-Abdelkader et al. [HAMR08]. Ces approches offrent effectivement un cadre théorique pour l'analyse des images sphériques mais présentent néanmoins certains inconvénients. L'image sphérique traitée n'est pas véritablement l'image acquise par la caméra et cela nécessite donc une phase d'interpolation des pixels pour obtenir un échantillonnage régulier sur la sphère. En pratique, cette étape est effectuée par une interpolation bi-linéaire dans le plan image de la caméra qui n'est théoriquement pas correcte et engendre donc des artéfacts avant même de traiter les images. De plus, avec ce formalisme, chaque outil de traitement d'images : convolution, filtrage, détection de points d'intérêt... doit être repensé. Dans nos travaux, nous avons montré que la plupart des outils classiques pouvaient être utilisés en modifiant préalablement la métrique de ces images. Par exemple de cette manière, nous avons pu réutiliser les approches classiques de détection de points de Harris ou de mise en correspondance de points d'intérêt par corrélation sans en modifier les approches. Par contre, certains problèmes ne peuvent être résolus de cette manière. Par exemple, pour calculer le flot optique dans les images catadioptriques, il faut revoir l'équation de contrainte du flot (BCCE) pour que celle-ci tienne compte des distorsions des

caméras grand angle. Ceci nous a conduit à faire l'étude du flot directement sur la sphère et à utiliser les ondelettes sphériques pendant la thèse d'Amina Radgui.

Pour faire suite à ce travail, nous envisageons d'examiner trois points précis : la mise en correspondance de droites, le traitement d'images non centrales et le calcul du temps à collision.

La mise en correspondance de droites dans les images sphériques est une étape nécessaire pour automatiser la procédure de calibrage extrinsèque de réseaux de caméras paramétriques que nous avons développée dans [LDV10a]. Alors que des méthodes de mise en correspondance de points commencent à voir le jour dans le cas d'images hétérogènes paramétriques ([PGS08, LW08, CMBP⁺12]), il existe peu de méthode à l'heure actuelle capables de mettre en correspondance les droites dans ces images. Les seules méthodes existantes supposent un très faible déplacement entre les deux prises de vue ([MFV09, VD10]) et ne peuvent fonctionner dans le cas d'un réseau de caméras hybrides. Pour ce faire, nous envisageons de définir un descripteur local des droites sphériques à l'aide de la mesure géodésique proposée précédemment.

Dans ce manuscrit, nous avons montré que modifier la métrique pour filtrer les images paramétriques a permis d'améliorer les résultats de filtrage de ce type d'image. Cette méthode consistait en une approximation entre le filtrage directement sur la sphère et le filtrage sur le plan tangent au point courant. Nous pouvons étendre ce principe dans le cas d'images non centrales. En effet, si celles-ci sont définies à partir de leur ligne de vue [SRT⁺11], nous pouvons construire en chaque pixel de la caméra un filtrage à l'aide du plan orthogonal à sa ligne de vue.

Enfin, nous nous intéressons depuis peu aux applications du calcul du flot optique sphérique pour l'estimation du temps à collision d'un robot mobile muni d'une caméra catadioptrique. Cette étude conduite par Fatima Zahra Benamar (doctorante en co-tutelle entre l'Université de Picardie Jules Verne et l'Université de Mohammed V de Rabat) commence à fournir des résultats encourageants [BDEF⁺12].

Structure from motion sphérique

Grâce à la sphère d'équivalence, nous disposons d'un espace générique pour développer des méthodes de SFM pour toutes les caméras paramétriques (perspectives, catadioptrique à point de vue unique, fish-eye...). Comme cette modélisation sphérique préserve la géométrie projective, les méthodes classiques de *structure from motion*, fondées sur la géométrie projective, restent valides. Dans notre travail, nous avons proposé des solutions tenant compte des particularités des images sphériques. En effet, avec ce type d'images, les points de fuite sont toujours visibles et peuvent donc être estimés de façon plus robuste que sur des images perspectives. Nos contributions portent ici sur trois points majeurs. Dans un premier temps, nous avons développé des méthodes d'extraction des points de fuite et déduit la rotation relative entre deux positions de la caméra. Puis, nous avons défini un algorithme original ne nécessitant que deux points pour calculer la translation entre ces deux images sphériques dès lors que la rotation relative est connue. Cet algorithme des 2 points se révèle très efficace en robotique aérienne puisque l'on dispose de la rotation relative entre les deux prises de vue grâce à la centrale inertielle à chaque instant et demeure très rapide et robuste en l'associant à une procédure RANSAC. Enfin, nous avons aussi profité de l'extraction des droites de la scène pour l'estimation de la rotation pour définir une méthode de calcul linéaire de la translation entre deux vues à partir des droites et points. Ces

différents points ont été étudiés au sein des thèses de Jean-Charles Bazin et Dieu Sang Ly. Cette thématique de recherche est encore un sujet d'étude que nous allons développer car nous débutons un nouveau projet de recherche international NRF-ANR nommé DrAACaR. Ce projet a pour objectif de localiser et de reconstruire l'environnement autour d'un véhicule muni d'un banc de caméras fish-eye asynchrones. Durant ce projet, trois approches seront à étudier. La première dite une approche 2D/2D consistera à synchroniser les caméras à partir d'une contrainte épipolaire sphérique. La deuxième dite approche 3D/3D aura pour objectif de recalibrer les nuages de points 3D obtenus par les différentes caméras sphériques. Enfin, une approche intermédiaire 3D/2D dans laquelle nous supposons qu'une caméra nous fournit un nuage de points 3D que nous devons étoffer à l'aide d'une autre caméra. La thèse de Dandi Pani Paudel co-dirigée avec Adlane Habed (LSIIT, Strasbourg) et Pascal Vasseur (LITIS, Rouen) qui débute en septembre 2012 a pour objet cette dernière problématique. Nous nous intéresserons à la localisation et la reconstruction à l'aide de connaissance *a priori* sur la structure de l'environnement.

Applications en robotique aérienne

L'ensemble des travaux exposés dans ce manuscrit ont pour objectif d'améliorer la perception d'un robot. Dans les deux premiers chapitres, nous avons vu comment nous pouvions considérer les caméras grand angle pour améliorer les traitements bas et haut niveaux et ainsi estimer le déplacement d'un robot. Dans ce dernier chapitre, nous avons choisi de mettre l'accent sur la robotique aérienne pour laquelle nous avons développé des systèmes de vision originaux tirant profit d'une caméra sphérique couplée avec une caméra perspective ou d'un laser. La thèse de Damien Eynard a permis de développer des méthodes originales pour déduire d'un capteur hybride fish-eye/perspective, l'altitude et l'attitude d'un drone lors des phases d'atterrissage et décollage. Le couplage d'une caméra fish-eye et d'un laser circulaire a été étudié et est encore étudié par Ashutosh Natraj. Pour ce faire, Ashutosh Natraj a proposé deux méthodes l'une analytique détaillée dans ce manuscrit et récemment une géométrique fondée sur l'intersection de coniques [NSDV12].

L'utilisation de systèmes atypiques reste une problématique en développement dans notre équipe puisque nous avons débuté en septembre 2011 une thèse financée par la DGA et la région Bourgogne. La thèse de François Rameau co-dirigé avec Désiré Sidibé (Le2i, Le Creusot) et David Fofi (Le2i, Le Creusot) s'intéresse à un couplage caméra fish-eye et PTZ.

Bibliographie

- [ADJV02] J. Antoine, L. Demanet, L. Jacques, and P. Vandergheynst. Wavelets on the sphere : Implementation and approximations. *Applied and Computational Harmonic Analysis*, 13 :177–200, 2002.
- [AM00] G. Alefeld and G. Mayer. Interval analysis : theory and applications. *Journal of Computational and Applied Mathematics*, 121(1-2) :421 – 464, 2000.
- [AT00] M.E. Antone and S.J. Teller. Automatic recovery of relative camera rotations for urban scenes. In *CVPR'00*, pages 2282–2289, 2000.
- [AV99] J.-P Antoine and P Vandergheynst. Wavelets on the 2-sphere : A group-theoretical approach. *Applied and Computational Harmonic Analysis*, 7(3) :262 – 291, 1999.
- [BA93] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. Int. Conf. on Computer Vision*, pages 231–236, Berlin, Allemagne, 1993.
- [BA01] J.P. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *CVPR*, pages II :422–427, 2001.
- [Bar83] S.T. Barnard. Interpreting perspective image. *Artif. Intell.*, 21(4) :435–462, 1983.
- [BBTV07] I. Bogdanova, X. Bresson, J. Thiran, and P. Vandergheynst. Scale-space analysis and active contours for omnidirectional images. *IEEE Transactions on Image Processing*, 16(7) :1888–1901, 2007.
- [BDEF⁺12] F. Z. Benamar, C. Demonceaux, S. El Fkihi, E. M. Mouaddib, and D. Aboutajdine. Time to contact estimation on paracatadioptric cameras. In *International Conference on Pattern Recognition*, November 2012.
- [BDV07] J.-C Bazin, C. Demonceaux, and P. Vasseur. Fast central catadioptric line extraction. In *IbPRIA (2)*, pages 25–32, 2007.
- [BDVK12] J.-C. Bazin, C. Demonceaux, P Vasseur, and I.S. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. *International Journal of Robotics Research*, 31(1) :63–81, 2012.
- [Ber98] C. Bernard. Discrete wavelet analysis : A new framework for fast optic flow computation. In *ECCV98*, page II : 354, 1998.
- [BFV09] L. Bagnato, P. Frossard, and P. Vandergheynst. Optical flow and depth from motion for omnidirectional images using a tv-l1 variational framework on graphs. In *IEEE International Conference on Image Processing - ICIP*, 2009.

- [BKD08] S. Bigot, D. Kachi, and S. Durand. Spherical edge detector : Application to omnidirectional imaging. In *ACTVS*, pages 554–565, 2008.
- [BMZF06] A. Beyeler, C. Mattiussi, J.-C Zufferey, and D. Floreano. Visionbased altitude and pitch estimation for ultra-light indoor aircraft. In *in IEEE International Conference on Robotics and Automation ICRA '06*, pages 2836–2841, 2006.
- [BNM01] G. Barrows, C. Neely, and K. Miller. Optic flow sensors for mav navigation. 195 :557–574, 2001.
- [BP02] E. Bruno and D. Pellerin. Robust motion estimation using spatial gabor-like filters. *Signal Processing*, 82(2) :297–309, 2002.
- [Bül02] T. Bülow. Multiscale image processing on the sphere. In *DAGM Symposium Symposium for Pattern Recognition*, pages 609–617, 2002.
- [CAKB10] M. Chandraker, S. Agarwal, D. Kriegman, and S. Belongie. Globally optimal algorithms for stratified autocalibration. *Int. J. Comput. Vision*, 90(2) :236–254, 2010.
- [CMBP⁺12] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J.-P. Thiran. Scale invariant feature transform on the sphere : Theory and applications. *Int. J. Comput. Vision*, 98(2) :217–241, June 2012.
- [CMEM07] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In *IROS*, pages 1683–1688, 2007.
- [Coh93] I. Cohen. Nonlinear variational method for optical flow computation. In *proc. Scandinavian Conference on Image Analysis*, pages 523–530, Mai 1993.
- [Col96] R. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Vision and Pattern Recognition*, pages 358–363, June 1996.
- [CSZ04] J.S. Chahl, M. V. Srinivasan, and S.-W. Zhang. Landing strategies in honeybees and applications to uninhabited airborne vehicles. *I. J. Robotic Res.*, 23(2) :101–110, 2004.
- [DBL10] *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010*. IEEE, 2010.
- [DEE08] P. Denis, J.H. Elder, and F. J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *ECCV (2)*, pages 197–210, 2008.
- [DH94] J. R. Driscoll and D. M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Adv. in Appl. Math.*, pages 202–250, 1994.
- [DMB02] K. Daniilidis, A. Makadia, and T. Bulow. Image processing in catadioptric planes : Spatiotemporal derivatives and optical flow computation. In *Workshop on Omnidirectional Vision (in conjunction with ECCV 2002)*, pages 3–12, 2002.
- [DS02] P. Doubek and T. Svoboda. Reliable 3d reconstruction from a few catadioptric images. In *Proceedings of the Third Workshop on Omnidirectional Vision, OMNIVIS '02*, pages 71–78, Washington, DC, USA, 2002.
- [DV06] C. Demonceaux and P. Vasseur. Markov random fields for catadioptric image processing. *Pattern Recognition Letters*, 27(16) :1957–1967, 2006.

-
- [DVF11] C. Demonceaux, P. Vasseur, and Y. Fougerolle. Central catadioptric image processing with geodesic metric. *Image Vision Computing*, 29(12) :840–849, 2011.
- [DVP06] C. Demonceaux, P. Vasseur, and C. Pégard. Omnidirectional vision on uav for attitude computation. In *IEEE International Conference on Robotics and Automation 2006 (ICRA '06)*, pages 2842–2847, Orlando, FL, US, May 2006. IEEE.
- [DVP07] C. Demonceaux, P. Vasseur, and C. Pégard. Uav attitude computation by omnidirectional vision in urban environment. In *IEEE International Conference on Robotics and Automation 2007 (ICRA '07)*, pages 2017–2022, Roma, Italy, April 2007. IEEE.
- [EVDF12] D. Eynard, V. Vasseur, C. Demonceaux, and V. Frémont. Real time uav altitude, attitude and motion estimation from hybrid stereovision. *Auton. Robots*, 33(1-2) :157–172, 2012.
- [FB81] M. A. Fischler and R. C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 1769–1786, 1981.
- [Fit01] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR (1)*, pages 125–132, 2001.
- [FM95] O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *ICCV*, pages 951–956, 1995.
- [GD01] C. Geyer and K. Daniilidis. Catadioptric projective geometry. *IJCV*, 45(3) :223–243, December 2001.
- [GN98] J. Gluckman and S. K. Nayar. Ego-motion and omnidirectional cameras. In *ICCV*, pages 999–1005, 1998.
- [GOSB03] W. E. Green, P. Y. Oh, K. Sevcik, and G. Barrows. Autonomous landing for indoor flying robots using optic flow. In *ASME International Mechanical Engineering Congress and Exposition*, pages 1347–1352, 2003.
- [HAMR08] H. Hadj-Abdelkader, E. Malis, and P. Rives. Spherical image processing for accurate visual odometry with omnidirectional cameras. In *IEEE Workshop on Omnidirectional Vision and Camera Networks, OMNIVIS'08*, Marseille, France, October 2008.
- [Har97] R.I. Hartley. In defense of the eight-point algorithm. *PAMI*, 19(6) :580–593, June 1997.
- [HK09] R. Hartley and F. Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1) :64–79, 2009.
- [HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3) :185–203, August 1981.
- [HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [KHK10] J.-S. Kim, M. Hwangbo, and T. Kanade. Spherical approximation for multiple cameras in motion estimation : Its applicability and advantages. *Comput. Vis. Image Underst.*, 114(10) :1068–1083, 2010.

- [KLH10] J.-H. Kim, H. Li, and R. Hartley. Motion estimation for nonoverlapping multicamera rigs : Linear algebraic and l_∞ geometric solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 :1044–1059, 2010.
- [LDV10a] S. Ly, C. Demonceaux, and P. Vasseur. Multi-camera topology recovery using lines. In Paul Richard and José Braz, editors, *VISAPP (1)*, pages 245–250, 2010.
- [LDV10b] S. Ly, C. Demonceaux, and P. Vasseur. Translation estimation for single viewpoint cameras using lines. In *ICRA*, pages 1928–1933, 2010.
- [LT81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proc. of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.
- [LW08] L. Lu and Y. Wu. Quasi-Dense Matching between Perspective and Omnidirectional Images. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008*, Marseille, France, 2008.
- [MBMR06] C. Mei, S. Benhimane, E. Malis, and P. Rives. Homography-based tracking for central catadioptric cameras. In *IROS*, 2006.
- [MCMO10] I. F. Mondragón, P. Campoy, C. Martinez, and M. Olivares. Omnidirectional vision applied to unmanned aerial vehicles (uavs) attitude and heading estimation. *Robot. Auton. Syst.*, 58(6) :809–819, June 2010.
- [MFV09] S. Mosaddegh, D. Fofi, and P. Vasseur. A generic method of line matching for central imaging systems under short-baseline motion. In Pasquale Foggia, Carlo Sansone, and Mario Vento, editors, *ICIAP*, pages 939–948, 2009.
- [MI08] Y. Mochizuki and A. Imiya. Featureless visual navigation using optical flow of omnidirectional image sequence. In *Workshop of SIMPAR'08*, pages 307–318, 2008.
- [MLD⁺06] E. Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Real time localization and 3d reconstruction. In *CVPR (1)*, pages 363–370. IEEE Computer Society, 2006.
- [MP95] E.M. Mouaddib and C. Pégard. Localization using omnidirectional vision. In *Proceedings Int. Conf. an Advanced Robotics.*, pages 133–138, 1995.
- [MR07] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, April 2007.
- [MSKS05] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision : From Images to Geometric Models*. Springer-Verlag, 2005.
- [MTB⁺10] R. J. D. Moore, S. Thurrowgood, D. P. Bland, D. Soccol, and M. V. Srinivasan. Uav altitude and attitude stabilisation using a coaxial stereo vision system. In *ICRA [DBL10]*, pages 29–34.
- [Nay] S. K. Nayar. Catadioptric omnidirectional camera. pages 482–488, June.
- [NDVS11] A. Natraj, C. Demonceaux, P. Vasseur, and P. Sturm. Vision based attitude and altitude estimation for uavs in dark environments. In *IROS*, pages 4006–4011, 2011.

-
- [Nis04] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26 :756–777, June 2004.
- [NS06] D. Nistér and H. Stewénus. A minimal solution to the generalized 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 2006.
- [NSDV12] A. Natraj, P. Sturm, C. Demonceaux, and P. Vasseur. A geometrical approach for vision based attitude and altitude estimation for uavs in dark environments. In *IROS*, 2012.
- [PGS08] L. Puig, J. Guerrero, and P. Sturm. Matching of omnidirectional and perspective images using the hybrid fundamental matrix. In *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS*, Marseille, France, 2008.
- [QM89] Long Quan and Roger Mohr. Determining perspective structures using hierarchical hough transform. *Pattern Recognition Letters*, 9(4) :279 – 286, 1989.
- [RDM⁺08] A. Radgui, C. Demonceaux, E. Mouaddib, M. Rziza, and D. Aboutajdine. An adapted lucas-kanade’s method for optical flow estimation in catadioptric images. In *In OMNIVIS’2008, the Eighth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, in conjunction with ECCV 2008*, Marseille, France, October 2008.
- [RDM⁺11] A. Radgui, C. Demonceaux, E. Mouaddib, M. Rziza, and D. Aboutajdine. Optical flow estimation from multichannel spherical image decomposition. *Computer Vision and Image Understanding*, 115(9) :1263–1272, 2011.
- [RLS06] S. Ramalingam, S. Lodha, and P. Sturm. A generic structure-from-motion framework. *Computer Vision and Image Understanding*, 103(3) :218–228, sep 2006.
- [SB] P. Sturm and J. Barreto. General imaging geometry for central catadioptric cameras. In *10th European Conference on Computer Vision, ECCV 2008, October, 2008*, pages 609–622, Marseille, France.
- [SDMF12] A. E. R. Shabayek, C. Demonceaux, O. Morel, and D. Fofi. Vision based uav attitude estimation : Progress and insights. *Journal of Intelligent and Robotic Systems*, 65(1-4) :295–308, 2012.
- [SEN06] H. Stewénus, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60 :284–294, June 2006.
- [SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000.
- [SPH98] T. Svoboda, T. Pajdla, and V. Hlavác. Epipolar geometry of panoramic cameras. In *Computer Vision - ECCV 98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume I*, volume 1406, pages 218–231, 1998.
- [SRT⁺11] P. Sturm, S. Ramalingam, J-P. Tardif, S. Gasparini, and J. Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1-2) :1–183, 2011.

- [SSS⁺08] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, pages 159 :1–159 :10, 2008.
- [Stu00] P. Sturm. A method for 3d reconstruction of piecewise planar objects from single panoramic images. In *Proceedings of the IEEE Workshop on Omnidirectional Vision, Hilton Head Island, USA*, pages 119–126, June 2000.
- [TBFV05] I. Tomic, I. Bogdanova, P. Frossard, and P. Vandergheynst. Multiresolution Motion Estimation for Omnidirectional Images. In *Proceedings of European Signal Processing Conference*, 2005.
- [TIO05] A. Torii, A. Imiya, and N. Ohnishi. Two- and three- view geometry for spherical cameras. In *OMNIVIS*, 2005.
- [VD10] P. Vasseur and C. Démonceaux. Central catadioptric line matching for robotic applications. In *ICRA [DBL10]*, pages 2562–2567.
- [WHH⁺05] L. K. Wang, S.-C. Hsieh, E. C.-W. Hsueh, F.-B. Hsaio, and K.-Y. Huang. Complete pose determination for low altitude unmanned aerial vehicle using stereo vision. In *IROS*, pages 108–113. IEEE, 2005.
- [WM95] J. Weber and J. Malik. Robust computation of optical-flow in a multiscale differential framework. *International Journal of Computer Vision*, 14(1) :67–81, January 1995.
- [Yag99] Y. Yagi. Omnidirectional sensing and its applications. *IEICE Trans, on Information and Systems*, E82-D(3) :568–579, 1999.
- [YH04] X. Ying and Z. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *Computer Vision - ECCV 2004*, volume 3021, pages 442–455, 2004.
- [YK90] Y. Yagi and S. Kawato. Panorama scene analysis with conic projection. In *Proceedings IEEE/RSJ International Workshop on Intelligent Robots and Systems '90*, pages 181–187, July 1990.