



HAL
open science

Contributions à la vision omnidirectionnelle

Bertrand Vandepoortaele

► **To cite this version:**

Bertrand Vandepoortaele. Contributions à la vision omnidirectionnelle. Electronique. Institut National Polytechnique (Toulouse), 2006. Français. NNT : 2006INPT037H . tel-04594430

HAL Id: tel-04594430

<https://hal.science/tel-04594430v1>

Submitted on 30 May 2024

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

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

THÈSE

présentée

devant l'Institut National Polytechnique de Toulouse

pour obtenir le titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

ÉCOLE DOCTORALE INFORMATIQUE ET TÉLÉCOMMUNICATIONS
SPÉCIALITÉ SIGNAL, IMAGE, ACOUSTIQUE & OPTIMISATION

par

Bertrand VANDEPORTAELE

Équipes d'accueil : Vision Par Calculateur André Bruel (IRIT)
& Groupe Systèmes Optoélectroniques (LEN7)

***Contributions à la vision omnidirectionnelle:
Étude, Conception et Étalonnage de capteurs pour l'acquisition
d'images et la modélisation 3D***

Soutenue le 8 Décembre 2006 devant le jury composé de:

Directeur de Recherche	Patrick	RIVES	INRIA , Sophia Antipolis	
	HDR Simon	LACROIX	LAAS, Toulouse	Rapporteurs
Professeur	El Mustapha	MOUADDIB	CREA, Amiens	
	HDR Peter	STURM	INRIA Rhône-Alpes	
Professeur	Philippe	MARTHON	IRIT, Toulouse	Directeurs de thèse
Professeur	Michel	CATTOEN	LEN7, Toulouse	

Remerciements

Je souhaite tout d'abord remercier mes directeurs de thèse Michel Cattoen et Philippe Marthon, pour leurs conseils et l'attention qu'ils m'ont témoignée pendant ces trois années.

Un grand merci également à Pierre Gurdjos, pour son aide et ses conseils lors de ma dernière année de thèse. Il a largement participé aux travaux présentés au chapitre 4 et plus ponctuellement à certains des travaux présentés au chapitre 3.

Je souhaite exprimer ma profonde gratitude à Simon Lacroix, El Mustapha Mouaddib et Peter Sturm, pour avoir accepté d'être rapporteurs de cette thèse ainsi que pour les travaux de recherche qu'ils ont effectués et qui ont aidé à son accomplissement. Merci aussi à Patrick Rives pour avoir accepté de présider le jury.

Je suis reconnaissant envers Jean Conter et Philippe Puech, avec qui j'ai pu beaucoup apprendre et partager une passion commune pour le développement de capteurs, alors que nous étions entourés de spécialistes du logiciel. Merci aussi aux autres permanents de l'équipe Vision Par Calculateur André Bruel, ainsi qu'à ceux du Laboratoire d'Électronique de l'ENSEEIH.

Je tiens également à exprimer ma gratitude envers Yves Megret, professeur à l'INSA de Toulouse, pour son aide dans la réalisation du capteur 3D omnidirectionnel et tous ses conseils très instructifs dans le domaine de la mécanique.

Je souhaite aussi remercier João Barreto, Christopher Geyer et Xianghua Ying pour avoir répondu de manière sympathique à toutes les questions que je leur ai posées. Merci aussi à Yonatan Wexler pour la mosaïque présentée au chapitre 1, qui immortalisa un très agréable repas en compagnie de certains des chercheurs dont les travaux ont le plus inspiré cette thèse.

Merci aussi à Michel Dayde et Thierry Bosch, pour m'avoir accueilli dans leurs laboratoires respectifs.

Je remercie également tous les stagiaires et thésards passés et présents des laboratoires IRIT et LEN7 qui ont contribué à faire de ces trois années de thèse un moment agréable, notamment Clovis Tauber, Matthijs Douze, Pascaline Parisot, Christophe Dehais, Ariel Choukroun, Marco Mora et Cezar Plesca ainsi que Caroline Bes, Olivier Bernal et Laurent Gatet.

Je souhaite aussi adresser un grand merci à Frédéric Lerasle, du LAAS, pour m'avoir fait découvrir la vision par ordinateur et m'avoir par la même occasion *contaminé* d'une passion qui n'a jamais faibli depuis cinq ans.

Merci aussi à mes amis, scientifiques ou non, pour m'avoir écouté pendant des heures leur expliquer mes recherches en me laissant croire qu'elles les intéressaient.

Enfin, un grand merci à ma famille pour m'avoir soutenu, à ma mère Martine, pour m'avoir poussé à continuer les études et à ma compagne Claire, dont la patience, la gentillesse et les encouragements durant ces trois années ont été la plus belle preuve d'amour qu'elle ait pu me faire.

Publications connexes à ce document

Conférences internationales avec actes et comité de lecture

- **ORIENT-CAM, A CAMERA THAT KNOWS ITS ORIENTATION AND SOME APPLICATIONS,**
CIARP 2006 : 11th Iberoamerican Congress on Pattern Recognition,
VANDEPORTAELE Bertrand, DEHAIS Christophe, CATTOEN Michel, MARTHON Philippe.
- **A FAST DETECTOR OF LINE IMAGES ACQUIRED BY AN UNCALIBRATED PARACATADIOPTRIC CAMERA,**
ICPR 2006 : 18th International Conference on Pattern Recognition,
VANDEPORTAELE Bertrand, CATTOEN Michel, MARTHON Philippe.
- **A NEW LINEAR CALIBRATION METHOD FOR PARACATADIOPTRIC CAMERAS,**
ICPR 2006 : 18th International Conference on Pattern Recognition,
VANDEPORTAELE Bertrand, CATTOEN Michel, MARTHON Philippe, GURDJOS Pierre.
- **REALTIME LOCALIZATION OF A CENTRAL CATADIOPTRIC CAMERA USING VERTICAL LINES,**
VISAPP 2006 : International Conference on Computer Vision Theory and Applications,
VANDEPORTAELE Bertrand, CATTOEN Michel, MARTHON Philippe.
- **IMAGE RECTIFICATION AND SYNTHESIS FOR AN IMAGE BASED AUTOMATIC GUIDANCE SYSTEM IN A 3D MODELLED SITE,**
PSIP 2005 : Physics in Signal and Image Processing,
VANDEPORTAELE Bertrand, CATTOEN Michel, MARTHON Philippe.

Conférences et colloques nationaux

- **UN SYSTÈME DE GUIDAGE DE PERSONNES BASÉ IMAGE,**
ORASIS 2005 : Congrès jeunes chercheurs en Vision par ordinateur,
VANDEPORTAELE Bertrand.
- **SYSTÈME DE GUIDAGE PAR L'IMAGE DANS UN SITE MODELISÉ EN 3 DIMENSIONS,**
Colloque de l'École Doctorale Informatique et Télécommunication, 2004,
VANDEPORTAELE Bertrand.

Revue Locale

- **DÉVELOPPEMENT D'UN SYSTÈME D'ACQUISITION 3D OMNIDIRECTIONNEL,**
Revue française CANAL N7, 2006, *VANDEPORTAELE Bertrand.*

Table des matières

Introduction, contexte et problématique	7
La vision par ordinateur	7
Différents champs de vision	8
La vision large champ et ses applications	9
Applications robotiques	9
Applications pour l’homme	9
Contexte de la thèse : Un système de guidage basé image pour les personnes	11
La localisation	11
Un système simple basé apparence utilisé en robotique	12
Contraintes différentes dans le cas de la localisation de personnes	12
Un système adapté au guidage des personnes	13
Contributions et organisation du mémoire	15
Autres contributions	16
1 L’imagerie à large champ de vision	17
1.1 Différents moyens d’accroître le champ visuel d’une caméra	17
1.1.1 Une caméra standard en rotation	17
1.1.2 Des caméras avec champ de vision étendu	23
1.2 Modèles de projection pour la vision panoramique et omnidirectionnelle	32
1.2.1 Exemple de modèle pour caméra standard : le trou d’épingle	33
1.2.2 Des modèles adaptés aux caméras à large champ de vision	34
1.2.3 Des modèles de caméras centrales	35
1.2.4 Des modèles de caméras non centrales	36
1.2.5 Un modèle de caméra radiale	39
1.2.6 Un modèle général pour une ou plusieurs caméras quelconques	39
1.3 Le problème de l’étalonnage géométrique	40
1.3.1 Étalonnage avec un modèle paramétrique	40

1.3.2	Étalonnage avec un modèle non paramétrique	40
1.3.3	Différents types d'observations	41
1.4	Reprojection d'images	44
1.4.1	Reprojection à partir de modèles centraux	44
1.4.2	Reprojection à partir de modèles non centraux	45
1.4.3	Reprojection de textures	45
2	Conception et réalisation d'une caméra utilisant un capteur d'orientation pour le calcul de mosaïques d'images orientées	47
2.1	Motivations	47
2.1.1	Point de départ	48
2.1.2	Des capteurs déjà utilisés pour la vision	48
2.1.3	Intérêt d'un nouveau capteur	49
2.2	Le capteur d'orientation	49
2.2.1	Principe	49
2.2.2	Description du capteur	50
2.2.3	Étalonnage du capteur	52
2.2.4	Détermination de l'orientation	54
2.3	La caméra équipée du capteur d'orientation	57
2.3.1	Description	57
2.3.2	Étalonnage de l'orientation relative	57
2.4	Mosaïquage d'images	59
2.4.1	Avantages liés au capteur d'orientation	59
2.4.2	La création de mosaïque	61
2.4.3	Résultats	62
2.5	Affinement de mosaïque	65
2.5.1	Détection de correspondances	66
2.5.2	Ajustement de faisceaux	68
2.5.3	Post Traitement	72
2.5.4	Résultats	73
2.6	Conclusions et perspectives	74
2.6.1	Perspectives	74
3	Caméra paracatadioptrique centrale : Détection de l'orientation et étalonnage géométrique grâce aux images de droites	79
3.1	La caméra paracatadioptrique	79
3.1.1	Un modèle géométrique	80

3.1.2	Équivalence avec la projection sphérique puis stéréographique	82
3.2	Projection des droites	82
3.2.1	Propriété des images de droites	83
3.2.2	Les droites parallèles	84
3.3	Détection des images de droites avec des paramètres de la caméra connus	85
3.3.1	Ajustement d'une image de droite sur un ensemble de points	85
3.3.2	Détection robuste des images de droites	88
3.4	Détermination de l'orientation	93
3.4.1	Regroupement des images de droites potentiellement parallèles	93
3.4.2	Un critère pour définir l'intersection proche de plusieurs plans	95
3.4.3	Affinement de l'estimation de l'intersection	95
3.4.4	Estimation de l'orientation d'un plan	96
3.4.5	Estimation de l'orientation de la caméra	97
3.5	Méthodes existantes pour l'étalonnage	101
3.5.1	La méthode de Geyer (G1) et Daniilidis utilisant les points de fuite	101
3.5.2	La méthode de Geyer (G2) et Daniilidis utilisant la contrainte sur les images de droites	101
3.5.3	La méthode de Geyer (G3) et Daniilidis utilisant l'espace des cercles sur le paraboloïde	102
3.5.4	La méthode de Geyer (G4) et Daniilidis utilisant l'espace des cercles sur la sphère	103
3.5.5	La méthode de Barreto (B) et Araujo	103
3.5.6	La méthode de Ying (Y) et Zha	103
3.5.7	D'autres méthodes	104
3.6	Une nouvelle méthode pour l'étalonnage à partir de droites	105
3.6.1	Motivations	105
3.6.2	Paramétrisation des images de droites	106
3.6.3	Traduction de la contrainte sur les équations de plan et étalonnage	107
3.6.4	Résultats sur des données de synthèse	111
3.6.5	Résultats sur des images réelles	115
3.7	Détection robuste des droites et étalonnage	115
3.7.1	Description de la méthode	116
3.7.2	Résultats	118
3.8	Conclusions	120

4 Caméra catadioptrique non centrale avec miroir quadrique : Un modèle direct pour la projection des points et ses applications

4.1	Motivations et définition du problème	122
4.1.1	Modèles directs et inverses de caméra	122
4.1.2	Intérêts et applications du modèle direct de caméra	124
4.1.3	Méthodes existantes pour la détermination du modèle direct de caméra catadioptrique non centrale	125
4.1.4	Nouvelle approche et ses apports	126
4.2	Réflexion de la lumière sur une surface	127
4.2.1	Formulation utilisant les angles	127
4.2.2	Formulation utilisant la distance du trajet lumineux	128
4.3	Réflexion dans le cas 2D pour les miroirs sphériques et cylindriques	129
4.3.1	Un problème bidimensionnel	129
4.3.2	Intersection et tangence de deux coniques	131
4.3.3	Tangence d'un cercle et d'une conique dont les foyers sont fixés	134
4.4	Réflexion dans le cas 3D pour tous les miroirs quadriques	138
4.4.1	Intersection et tangence de deux quadriques	138
4.4.2	Tangence entre une quadrique modélisant un miroir et une quadrique dont les foyers sont fixés	140
4.4.3	Considérations annexes	142
4.5	Résultats	144
4.5.1	Calcul des points de tangence	144
4.6	Applications	152
4.6.1	Projection d'un point	152
4.6.2	Projection d'une droite ou d'un segment	155
4.6.3	Reprojection de texture	157
4.6.4	Étalonnage géométrique	160
4.7	Conclusions	165
4.7.1	Conclusions sur l'étalonnage	166
5	Conception et réalisation d'un capteur 3D omnidirectionnel catadioptrique	167
5.1	Description du capteur	167
5.1.1	Principe	168
5.1.2	Avantages de ce type de capteur	169
5.1.3	La mécanique	170
5.1.4	L'électronique	170
5.1.5	La caméra et l'optique	172
5.1.6	Le logiciel	173
5.2	Étalonnage du capteur	173

5.2.1	Un modèle géométrique du capteur	174
5.2.2	La mire 3D	175
5.2.3	Estimation des paramètres du modèle	176
5.2.4	Résultats	181
5.3	Reconstruction 3D	182
5.3.1	Géométrie épipolaire	182
5.3.2	Rectification épipolaire	185
5.3.3	Triangulation des points	194
5.3.4	Appariement des homologues	194
5.3.5	Reconstruction de texture	197
5.3.6	Résultats	199
5.4	Critiques et perspectives	204
5.4.1	Vers un meilleur modèle pour la caméra paracatadioptrique	204
5.4.2	Vers un meilleur modèle pour représenter la scène	205
5.4.3	Vers un plus grand nombre d'images pour l'acquisition	206
5.4.4	Vers l'acquisition active	207
5.5	Conclusions	209
Conclusions et perspectives		211
	Contributions	211
	Perspectives	212
	Vers la localisation basée image	213
Annexes		214
A Quelques définitions		215
B Synthèse d'images pour des caméras centrales à large champ de vision		217
	Gestion de l'antialiasage	219
C Calcul du quaternion correspondant à une matrice de rotation		223
D Simplification de l'algorithme d'étalonnage proposé par Barreto		225
E Détails sur les calculs du chapitre 3 et exemples		227
	Code MAPLE pour obtenir le polynôme pour la réflexion sur la sphère	227
	Polynôme pour un point à coordonnées finies	228
	Polynôme pour un point à l'infini	228

F Anaglyphe sphérique	231
G Articles sur la localisation basée image	233
Bibliographie	250

Introduction, contexte et problématique

La vision par ordinateur

Les personnes extérieures au domaine de la vision me demandant le sujet de mes recherches sont souvent surprises lorsque je leur répond que j'essaie de permettre à une machine de *voir*. Certaines d'entre elles pensent qu'il suffit pour cela de brancher une webcam à un ordinateur afin qu'il puisse acquérir des images et ne voient pas en quoi il est nécessaire d'effectuer des recherches. D'autres, au contraire, sont conscientes de la difficulté d'analyser ces images et de percevoir l'environnement et pensent même parfois la tâche impossible. Dans tous les cas, très peu sont réellement conscientes des méthodes pouvant être employées pour extraire l'information utile émanant d'une caméra...

Aujourd'hui, les énormes progrès dans le domaine, aidés par l'augmentation de la puissance de calcul des ordinateurs, la miniaturisation des caméras et la diminution des prix du matériel, permettent une utilisation de plus en plus fréquente des systèmes de vision dans des domaines très variés, tels que la vidéo surveillance (détection d'intrusion, d'incendie), les applications militaires (guidage de missiles, repérage d'installations), médicales (détection de tumeurs par IRM), robotiques (perception de l'environnement du robot), spatiales (recherche d'astres), industrielles (métrologie sans contact, inspection), et même ludiques (réalité augmentée, robotique ludique).

Pourtant force est de constater que malgré ces progrès, le domaine de la vision est encore en perpétuelle évolution. Chaque année, de nouvelles méthodes sont proposées, de nouvelles applications imaginées et de nouveaux capteurs développés, faisant de ce domaine une source intarissable pour les chercheurs. L'une des particularités de la vision est sa pluridisciplinarité. En fait, elle implique un travail de programmation, mais aussi de bonnes connaissances pratiques des mathématiques, tant en algèbre qu'en géométrie et en statistiques. Le traitement du signal, l'Intelligence Artificielle et même l'électronique y sont largement utilisés. C'est ce qui, à mon goût, fait tout le charme de la vision par ordinateur, demandant un perpétuel apprentissage et permettant sans cesse d'étendre le domaine de ses connaissances.

Différents champs de vision

la figure 1 représente différents types de champs de vision et montre des images correspondantes. Une caméra sera dite *directionnelle* si son champ de vision couvre moins d'un hémisphère. Le terme *standard* sera aussi employé pour ces caméras car elles sont les plus courantes. Une caméra sera dite *panoramique* si son champ de vision couvre 360° autour d'un axe. Enfin, nous retiendrons le terme *omnidirectionnelle* pour les caméras dont le champ de vision couvre toute la sphère. Le terme *à large champ* sera employé pour une caméra disposant d'un champ de vision important, supérieur à celui d'une caméra standard.

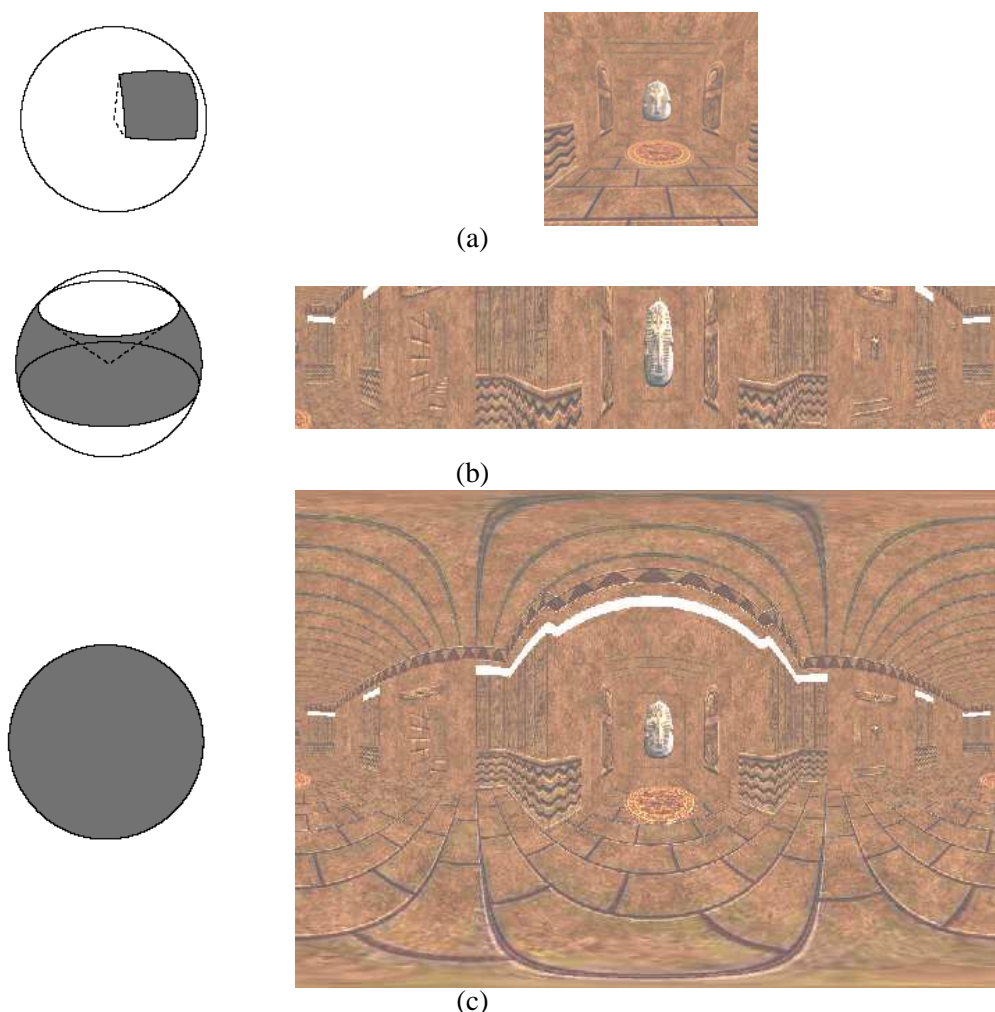


FIG. 1 – Champs de vision et images d'une caméra standard ou directionnelle avec 90° d'angle de vision horizontalement et verticalement (a), panoramique avec 33° d'angle de vision verticalement (b) et omnidirectionnelle (c).

Dans ce document, les termes panoramique et omnidirectionnel sont différenciés de la

même manière que dans [1]. Il y a souvent confusion entre ces termes à cause de l'anglais, langue dans laquelle "omnidirectional camera" est traduit en français par caméra panoramique.

La vision large champ et ses applications

Les images et caméras permettant un large champ de vision sont de plus en plus utilisées et trouvent des applications dans différents domaines, dont certaines sont mentionnés ici.

Applications robotiques

La robotique mobile est le domaine qui utilise le plus les caméras à large champ visuel. Ainsi, un robot tel le Scout visible sur la figure 2 (a) est équipé d'une optique catadioptrique en hauteur lui permettant de percevoir l'environnement tout autour de lui en complément d'une ceinture de sonars. La vision est surtout utilisée pour la localisation basée image, la détection d'obstacles et la coopération entre plusieurs robots. Un grand nombre de documents ([2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] par exemple) traitent de l'utilisation d'images grand angle, panoramiques ou omnidirectionnelles. On pourra noter l'existence d'un article de Yu et al. [14] datant de 1992 et proposant l'utilisation d'un caméra en rotation pour la localisation d'un robot.

Les robots terrestres ne sont pas les seuls à avoir besoin d'un champ de vision étendu, les robots aériens (Unmanned Aircraft Vehicule (UAV)) utilise aussi des caméras particulières. Ainsi l'hélicoptère visible sur la figure 2 (b), proposé par Hrabar et al. dans [15], est capable d'effectuer une tâche d'asservissement visuel en utilisant une caméra catadioptrique. Demonceaux et Vasseur proposent dans [16] une méthode permettant de déterminer l'attitude d'un drone, basée sur l'analyse de la projection de l'horizon sur une image d'une caméra catadioptrique centrale.

Les caméras à grand champ de vision sont aussi utilisées dans les missions d'exploration extra-terrestre par les Rovers. La figure 2 (c) montre une image acquise par le Mars Exploration Rover (MER) en utilisant une optique grand angle.

Applications pour l'homme

Les images panoramiques et omnidirectionnelles sont aussi de plus en plus souvent utilisées dans des applications destinées directement aux humains.

L'application la plus couramment utilisée est sans doute le mosaïquage d'images, qui permet, comme il sera montré au chapitre 1, de générer des images à large champ de vision en combinant plusieurs images correspondant à des champs visuels plus restreints. En astronomie, par exemple, les mosaïques permettent d'avoir simultanément la résolution suffisante pour l'observation des corps lointains et un champ visuel assez grand pour permettre leur localisation relative.

La vision panoramique trouve aussi des applications commerciales, par exemple pour permettre à des clients d'effectuer des visites virtuelles de bâtiments, avant d'acheter ou louer.

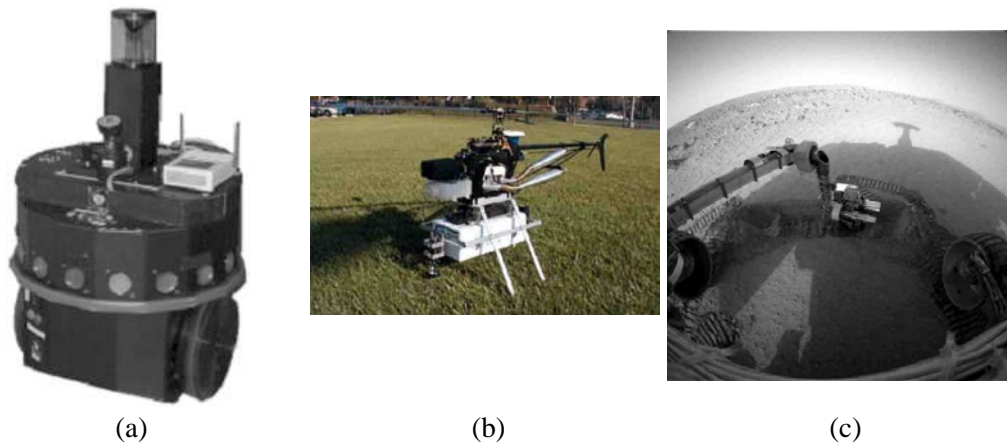


FIG. 2 – (a) Le robot Scout et sa caméra catadioptrique. (b) Un hélicoptère autonome avec caméra catadioptrique. (c) Une photo prise par le Rover MER sur Mars avec une optique grand angle (<http://photojournal.jpl.nasa.gov/catalog/PIA05339>).

En utilisant une technologie telle que Quicktime VR, il est possible de faire pivoter une caméra virtuelle pour percevoir l'environnement alentour, et ceci depuis un navigateur Internet quelconque.

Des applications de vidéo surveillance ont aussi été proposées, par exemple par Ikeda et al. dans [17], Sogo et al. dans [18], p359 ou Cielniak et al. dans [19]. La localisation et le suivi de personnes avec un nombre réduit de caméras couvrant un large champ visuel sont possibles. Des études du comportement humain dans les lieux publics sont menées en utilisant ces capteurs par exemple par Charif et al. dans [20].

Une application de visio-conférence, telle que celle proposée dans [18] p 39 par Baker et Nayar peut aussi tirer parti du large champ de vision des caméras. Dans leur exemple, plusieurs interlocuteurs sont situés autour d'une table sur laquelle se trouve une caméra catadioptrique. Autant d'images en projection perspective plane que de différentes personnes sont générées, chacune d'elle correspondant à une caméra orientée vers l'un des interlocuteurs. Ceci permet d'éviter l'emploi de nombreuses caméras standard et surtout assure le cadrage correct des personnes dans les différentes vidéos, même si les personnes se déplacent.

Les caméras à large champ peuvent aussi être utilisées dans le secteur automobile, par exemple pour forcer le respect des distances de sécurité ou aider au freinage automatique, comme le propose Fabrizio dans [21].

Le codage des vidéos peut aussi utiliser les images panoramiques, grâce à une séparation du fond et des objets, comme le proposent Anandan et Irani dans [18] p393. L'utilisation de panoramas est d'ailleurs prévue dans le format MPEG4 pour le codage du fond. Ainsi, au lieu d'encoder localement les transformations à appliquer à l'image, un panoramique du fond est généré à partir d'une séquence. Une transformation globale (homographie) codée par huit paramètres est ensuite estimée sur chaque image de la séquence. Les parties de l'image ne correspondant pas à cette transformation sont codées séparément sous forme de sprites, qui

sont des portions d'images avec des zones transparentes. Le rendu de la séquence s'effectue simplement par affichage du panorama transformé sur lequel les sprites sont surimposés.

La réalité augmentée peut aussi utiliser les images panoramiques, que ce soit pour réaliser l'incrustation vidéo des objets virtuels dans un fond réel ou même pour changer le fond comme le propose Douze dans [22].

Contexte de la thèse : Un système de guidage basé image pour les personnes

Une autre application pour la vision large champ est la localisation par l'image. Les différents travaux effectués durant cette thèse tentent de résoudre certains des défis soulevés par cette problématique.

La localisation

Le GPS (Global Positioning System) est aujourd'hui largement utilisé pour localiser et guider des personnes en milieu extérieur, dans leurs véhicules ou dans la rue grâce à des appareils portatifs. Un récepteur GPS, pour trianguler sa position en 3D, doit recevoir les signaux provenant d'au moins quatre satellites sur la constellation qui en compte vingt quatre.

Pour les applications de localisation à l'intérieur des bâtiments, le GPS n'est pas toujours utilisable, car les signaux émanant des satellites ne sont pas forcément captés. Dans ce cas, le LPS (Local Positioning System) peut être utilisé. Celui-ci utilise des bornes émettrices fixes installées à l'intérieur du site au lieu des satellites. Ce système nécessite donc l'installation d'infrastructures spécifiques et n'est malgré tout pas à l'abri de problèmes d'interférences ou de zones d'ombre radio.

Une alternative à l'emploi du LPS permettant de s'affranchir de la nécessité d'infrastructures spécifiques consiste à utiliser la vision par ordinateur. Elle est largement utilisée en robotique mobile pour la localisation. Deux approches sont possibles :

-La première consiste à privilégier la résolution au détriment du champ visuel et à détecter des détails dans la scène afin de localiser la caméra. Cette approche sous entend une interprétation assez poussée des images avec détection et segmentation de zones discriminantes.

-La seconde approche permet d'appréhender plus globalement l'environnement au détriment de la résolution. Ce type d'approche est qualifié de basé apparence car la localisation est alors effectuée en comparant l'apparence de l'image à localiser avec celles d'images d'une base de données. Avec une telle approche, le large champ visuel de la caméra est mis à profit pour rendre la localisation plus robuste à d'éventuels changements du site ou à des occultations. C'est cette approche qui est privilégiée dans la suite de la thèse.

Un système simple basé apparence utilisé en robotique

La figure 3 montre une procédure simple de localisation par l'image utilisant l'apparence adapté à un robot mobile. Une caméra panoramique, montée sur le robot, est utilisée pour acquérir une base de données constituée d'images en différentes positions échantillonnées sur le site. Ensuite, le robot, pour être localisé, acquiert une nouvelle image avec la même caméra. L'appariement de cette image avec une image de la base permet la localisation approximative du robot.

Seule l'orientation autour de l'axe vertical est considérée non constante et l'appariement doit donc être réalisé indépendamment de la rotation du robot. Comme montré par Pajdla dans [11], cette rotation correspond à une translation dans une image cylindrique ou sphérique, et l'appariement peut être effectué en autorisant un décalage entre les images.

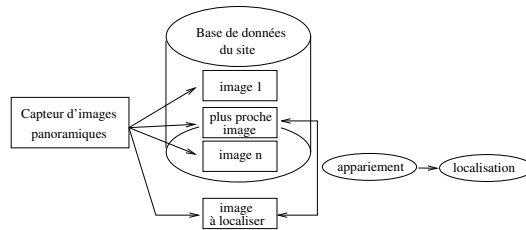


FIG. 3 – Un système simple de localisation par l'image.

De nombreuses recherches ont été effectuées sur la localisation utilisant l'apparence. Par exemple, Gonzalez et Lacroix proposent dans [23, 24], une méthode de localisation de robot mobile basée sur l'analyse d'histogrammes. Gonzalez-Buesa et al. utilisent dans [8] des chaînes représentant l'environnement, qu'ils appariant avec des algorithmes inspirés de la biologie. Des méthodes utilisant l'analyse en composantes principales sont présentées par Jogan et al. dans [25] et par Paletta et al. dans [5].

Menegatti et al. décrivent dans [26] un algorithme permettant l'analyse dynamique de la position du robot afin d'améliorer la localisation grâce à une étude probabiliste basée sur ses positions antérieures.

Remazeilles, Chaumette et Gros proposent dans [27] une méthode pour la navigation de robots mobiles utilisant la vision. Une loi de contrôle est définie en recherchant un parcours valide au travers d'un graphe dont les nœuds sont les images d'une base d'apprentissage du site dans lequel le robot se déplace.

On pourra aussi noter les travaux de Briggs et al. [28], plutôt orientés sur le suivi de position que sur la localisation absolue, mais présentant une approche intéressante à base d'images panoramiques 1D.

Contraintes différentes dans le cas de la localisation de personnes

Un tel système n'est pas directement utilisable pour guider des personnes car il existe des contraintes différentes.

Des caméras différentes : Dans l'application visée, les utilisateurs à localiser doivent pouvoir utiliser une caméra différente de celle avec laquelle le site a été acquis, par exemple celle d'un appareil téléphonique portable.

Des conditions de prises de vue différentes : Alors que la caméra d'un robot mobile est toujours située à la même hauteur par rapport au sol et dans une orientation fixe par rapport au robot, ceci n'est pas le cas pour une personne tenant une caméra. Pour résoudre ce problème, il faudrait acquérir des images de la base à différentes altitudes et être capable d'apparier les images alors qu'elles sont acquises avec des orientations différentes.

Un système adapté au guidage des personnes

Nous avons proposé un synoptique de système plus adapté au guidage des personnes, schématisé sur la figure 4. Ce système satisfait des contraintes spécifiques, qui sont différentes de celles de la robotique mobile.

À terme, ce système pourrait être utilisé pour le guidage dans les lieux publics (musées, administrations...), où il fournirait une aide notamment aux déficients visuels. Les images à localiser pourraient provenir d'un téléphone portable équipé d'une caméra et les traitements de vision seraient alors déportés vers un serveur recevant les images par liaison téléphonique, le guidage pouvant quant à lui être réalisé en temps réel par la voix.

Un tel système de localisation est proposé par Ravi et al. dans [29], qui discutent surtout les protocoles de communications, de la consommation énergétique et d'autres détails techniques au détriment des algorithmes de vision.

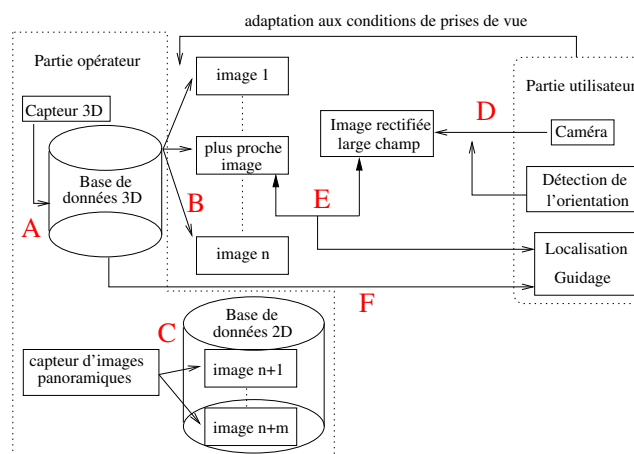


FIG. 4 – Un système plus complexe et mieux adapté à la localisation de personnes.

A : Acquisition d'un modèle 3D orienté pour représenter le site : Un capteur 3D omnidirectionnel permet l'acquisition d'un modèle 3D orienté et texturé du site qui est utilisé comme

majeure partie de la base de données. L'orientation du modèle est fournie grâce à un capteur électronique. L'utilisation d'un tel modèle pour représenter le site présente de nombreux avantages par rapport à une base de données d'images 2D statique parmi lesquels :

- Possibilité de générer les images en accord avec les conditions réelles de l'utilisateur qui souhaite se localiser, notamment à l'altitude de prise de vue de l'utilisateur.
- Compacité au niveau du stockage.
- Facilité de mise à jour par l'opérateur. En effet, il n'est pas nécessaire d'acquérir des centaines d'images du site chaque fois qu'il change. Une simple modification au niveau du modèle suffit.
- Possibilité d'ajouter de l'information une fois pour toutes au niveau du modèle 3D au lieu d'avoir à le faire sur chaque image.

B : Synthèse adaptative d'images orientées à partir du modèle 3D du site : Une base d'images omnidirectionnelles orientées de synthèse est générée dynamiquement à partir du modèle 3D pour s'adapter aux conditions de prise de vue de l'utilisateur. La connaissance de la hauteur de prise de vue, fixe pour un utilisateur donné, permet de générer des images de la base de données correspondant à la même altitude, ce qui doit faciliter l'appariement des images. Une méthode de localisation utilisant des images synthétisées à partir d'un modèle ont été présentés par Cauchois et al. dans [30].

C : Acquisition et reprojection d'images panoramiques 2D du site : Si une partie du site ne peut pas être modélisée en 3D, il est toujours possible d'acquérir des images panoramiques à l'aide d'une caméra panoramique. Si cette caméra dispose d'un centre de projection unique, ses images peuvent être reprojeter sur des représentations 2D omnidirectionnelles orientées, correspondant à la même projection que celle des images synthétisées à partir du modèle 3D.

D : Reprojection de l'image de la caméra de l'utilisateur en tenant compte de son orientation : La caméra de l'utilisateur est utilisée pour acquérir une image de l'environnement. Une nouvelle image est générée par reprojection pour permettre l'appariement avec les images de la base. L'orientation de la caméra doit être déterminée afin d'effectuer la reprojection. Ceci peut être fait de plusieurs façons, dont deux sont étudiées ici.

Dans le cas où la caméra est panoramique, son orientation peut être détectée à partir de l'image en faisant une hypothèse sur la nature de la scène observée. S'il y a présence d'un grand nombre de droites verticales, il est possible d'estimer l'orientation de la caméra par rapport au plan horizontal. Seule l'orientation autour de l'axe vertical ne peut être détectée par cette approche et l'appariement doit donc autoriser une rotation autour de cet axe, comme dans le cas simple appliqué à la robotique.

Pour éviter l'utilisation d'une caméra catadioptrique, qui n'est pas très adaptée à la manipulation par des personnes de par son encombrement et sa fragilité, une caméra standard, plus compacte, peut être employée. Dans ce cas, il est possible de l'équiper d'un capteur électronique d'orientation, ce qui permet de reprojeter directement son image. Plusieurs images acquises depuis la même position peuvent être intégrées afin de générer une mosaïque d'images

orientée, offrant un champ visuel accru tout en conservant l'avantage d'une résolution élevée, permettant la détection de détails de la scène.

E et F : Appariement des images et guidage à l'intérieur du site grâce au modèle 3D : L'appariement des images de l'utilisateur et la base permet la localisation. En plus d'être localisé, l'utilisateur peut être guidé à l'intérieur du site grâce à la connaissance de son modèle 3D.

Les travaux proposant cette architecture pour la localisation des personnes ont fait l'objet de publications, dans [31] et [32].

Contributions et organisation du mémoire

Ce document est composé de cinq chapitres relatant les principaux travaux effectués durant la thèse.

- Dans le premier [chapitre](#), le domaine de la vision panoramique et plus généralement large champ est présenté afin de positionner les travaux effectués pendant la thèse par rapport à l'état de l'art du domaine.
Différents types de caméras sont illustrés et leurs caractéristiques propres comparées. Des modèles de projections utilisables pour ces caméras sont détaillés, pour le cas des caméras centrales et non centrales. Une présentation rapide des méthodes d'étalonnage est effectuée, suivie d'une discussion sur les amers pouvant être employés pour l'étalonnage géométrique. Les techniques de reprojection d'images avec différents modèles sont expliquées.
- Le second [chapitre](#) présente une caméra équipée d'un capteur électronique d'orientation développé pendant la thèse.
Des méthodes d'étalonnage de l'ensemble sont proposées. Cette caméra est ensuite utilisée pour générer automatiquement des mosaïques d'images. L'emploi du capteur permet alors d'augmenter virtuellement le champ de vision de la caméra et d'obtenir des images orientées en temps réel. Il permet aussi l'acquisition aisée de scènes en mouvement ainsi que l'acquisition haute dynamique. Un descripteur de points exploitant l'information d'orientation fournie par le capteur est proposé. Il est finalement utilisé pour détecter des correspondances entre les différentes images et permettre un ajustement de faisceaux destiné à affiner la mosaïque.
- Le troisième [chapitre](#) traite de la caméra paracatadioptrique, capteur comprenant une optique télécentrique et un miroir paraboloidal.
Une méthode permettant la détection des images de droites et de l'orientation de la caméra est présentée. L'étalonnage géométrique est étudié et une nouvelle méthode est proposée. Elle est comparée avec les méthodes existantes et ensuite utilisée pour déterminer les paramètres de la caméra de manière robuste grâce à l'observation d'une scène contenant des droites.
- Dans le quatrième [chapitre](#), les caméras catadioptriques non centrales avec miroir en forme de quadrique sont étudiées.

Un modèle direct de caméra pour la projection de points de la scène vers l'image est proposé. La détermination de l'image d'un point est ramenée à un problème de tangence entre deux quadriques, lui-même menant à la recherche des racines d'un polynôme. Les différentes quadriques sont étudiées et le modèle de projection est détaillé. Diverses applications sont ensuite présentées, telles que la reprojection de textures ou la projection de segments de droites pour le calcul des courbes de correspondances stéréoscopiques. L'étalonnage d'une caméra non centrale est finalement effectué grâce à cette méthode.

- Dans le cinquième et dernier [chapitre](#), un second capteur réalisé pendant la thèse et qui permet l'acquisition de modèles 3D d'intérieurs de bâtiments en disposant d'un champ de vision omnidirectionnel est décrit.

Ce capteur est principalement constitué d'une caméra paracatadioptrique en rotation. Un protocole est proposé pour l'étalonnage géométrique du capteur, grâce à l'emploi d'une mire 3D spécifique. La reconstruction 3D est ensuite effectuée de différentes manières, en utilisant les images directement fournies par la caméra ou en les rectifiant épipolairement afin de simplifier l'appariement automatique des points homologues. Une application interactive de construction de modèles 3D texturés est présentée et des perspectives sont proposées pour améliorer le capteur.

Autres contributions

La localisation à proprement parler n'est pas étudiée dans ce document. Néanmoins, des travaux ont été effectués sur ce sujet durant la thèse et certains ont été publiés.

Dans [33], présenté en Annexe [G](#), une méthode de localisation basée sur la détection de lignes verticales de la scène avec une caméra paracatadioptrique est présentée. La localisation est alors effectuée à l'intérieur d'une carte 2D contenant les positions des droites verticales et des segments occultants représentant les murs. La méthode proposée est assez proche de celle, assez ancienne, de Marhic et al. dans [34] sauf qu'elle autorise une orientation libre de la caméra, indispensable pour la localisation des personnes. Aussi, au lieu d'utiliser des invariants projectifs, une méthode d'estimation robuste a été préférée.

Dans [35], présenté en Annexe [G](#), des résultats préliminaires sur une méthode de localisation utilisant l'apparence sont présentés. La caméra à localiser est celle présentée au [chapitre 2](#). Elle est utilisée pour obtenir des images invariantes aux rotations, qui, après appariement avec une base d'images panoramiques permettent la localisation. La méthode utilisée pour l'appariement des images est basée sur une mesure de corrélation.

Cette méthode a été validée sur un site assez petit, échantillonné par 200 images dans la base de données. L'extension à un site de plus grande taille nécessiterait l'acquisition d'un nombre trop élevé d'images pour la réaliser à la main, et n'a donc pas été réalisée. L'alternative consiste en l'acquisition du modèle 3D de la scène et en la synthèse d'image, mais ceci n'a pas encore été testé.

Chapitre 1

L'imagerie à large champ de vision

Ce chapitre introductif sur le domaine de la vision à large champ présente ses particularités et particulièrement ses avantages par rapport à la vision par ordinateur plus conventionnelle. Différents types de caméras ou méthodes permettant d'augmenter le champ de vision sont présentés et comparés en 1.1. Les caméras catadioptriques sont particulièrement étudiées. Elles sont classifiées en deux groupes : les caméras centrales et non centrales.

Des modèles géométriques de projection sont introduits en 1.2 afin de pouvoir associer soit une demi droite dans l'espace à un point de l'image soit un point dans l'image à un point dans l'espace après l'étalonnage géométrique présenté en 1.3. Ceci est ensuite utilisé en 1.4 par exemple pour projeter les images sur différentes surfaces ou pour synthétiser des images.

Les apports de la thèse sont repérés par un signe ✓ afin de permettre au lecteur de les situer rapidement dans le domaine.

1.1 Différents moyens d'accroître le champ visuel d'une caméra

Différents moyens d'obtenir des images panoramiques ou omnidirectionnelles sont discutés ici. Le choix à effectuer dépend de l'application. Fermüller et Aloimonos proposent dans [36] une méthodologie de choix de caméra, inspirée de la biologie, en fonction de l'application visée.

1.1.1 Une caméra standard en rotation

Le champ visuel humain, tel que représenté sur la figure 1.1 (a), couvre approximativement 180° horizontalement et 130° verticalement. Pour avoir une perception omnidirectionnelle de son environnement, l'humain, contrairement à la mouche, doit tourner sa tête et balayer la scène de son regard.

Une machine peut utiliser le même principe pour augmenter son champ de vision et reconstruire une image à partir de plusieurs images directionnelles acquises depuis une position fixe. Cette technique est connue sous le nom de mosaïquage d'images et trouve de très nombreuses



FIG. 1.1 – Champs visuels pour différents yeux. (a) L’humain, (b) La mouche, naturellement dotée d’un champ de vision quasi omnidirectionnel

applications commerciales. Elle a été discutée dans de très nombreux articles, notamment par Shum et Szeliski dans [18] p 226 et en une version totalement omnidirectionnelle dans [18] p291 par Nayar et Karmarkar.

1.1.1.1 Une caméra tournée manuellement et librement

Dans le cas où l’on dispose de plusieurs images acquises dans des orientations différentes et inconnues a priori, il est nécessaire de détecter des similarités dans les différentes images.

Des méthodes géométriques : La détection des similarités est généralement effectuée grâce à des points d’intérêt communs aux différentes images. Ces points peuvent par exemple être des images des coins, repérables efficacement et avec une répétabilité satisfaisante par des détecteurs tels que celui de Harris [37] ou le KLT (Kanade-Lucas-Tomasi Feature Tracker)[38]. Dans le cas de fortes variations d’échelle dans les différentes images, une version invariante à l’échelle de détecteur peut être utilisée. Par exemple, le détecteur SIFT [39], permet de détecter des extrema locaux dans l’espace d’échelle qui est une représentation de l’image à différentes résolutions. Grâce à cette analyse multirésolution et à la détection d’une échelle particulière, il est alors possible d’obtenir un taux de répétabilité du détecteur assez bon malgré des variations d’échelle importantes.

Le MSER, présenté dans [40] par Matas et al. utilise un algorithme de type bassin versant pour détecter des régions stables dans les images, dont les centres de gravité définissent des points utilisables pour les recherches de correspondances.

La similarité entre les points détectés dans les différentes images est ensuite mesurée à l’aide d’une mesure plus ou moins complexe. Des déformations géométriques du voisinage des points détectés ayant différents degrés de liberté peuvent être prises en compte afin de mesurer au mieux la similarité. Une méthode largement utilisée est celle de David Lowe [39], SIFT (Scale Invariant Features Transform), consistant à générer un descripteur local du voisinage du point considéré dans l’image qui soit invariant à des transformations affines d’amplitude

modérée. La ressemblance entre tous les points d'intérêt des différentes images est mesurée. Des méthodes d'optimisation robuste sont alors utilisées afin de calculer les transformations (appelées *homographies* ou *projectivités*) à appliquer à chaque image pour que le plus grand nombre de points ayant une forte similarité se retrouvent proches dans l'image mosaïquée.

Les homographies entre les différentes images peuvent aussi être déterminées grâce à l'observation de projections de droites ou de plans communs aux différentes images.

Des méthodes géométriques et photométriques : Certaines méthodes permettent de détecter les différentes transformations à appliquer aux images non pas en utilisant des points d'intérêt mais en mesurant directement la différence de valeur au niveau des pixels des images après transformation, par exemple celle d'Irani et al. dans [41] ou de Bartoli dans [42]. Ces méthodes permettent d'affiner les transformations à appliquer lorsque l'on dispose d'une solution initiale approchée. Grâce à cette approche, il est possible de gérer des transformations photométriques entre les différentes images. Ces transformations, à appliquer à chaque valeur de pixel indépendamment de son voisinage, sont estimées simultanément aux transformations géométriques. Ceci permet de s'adapter à la dynamique de l'appareil qui intègre la lumière. En effet, une caméra équipée d'un système automatique de contrôle d'exposition est capable de régler de manière autonome la durée pendant laquelle la lumière doit être intégrée, via sa commande de *shutter*. Le *gain*, facteur multiplicatif à appliquer aux valeurs de luminosité mesurées par la caméra, peut aussi être réglé automatiquement. Ces réglages sont effectués individuellement sur chaque image et il en résulte des différences d'exposition sur l'image mosaïquée si celles-ci ne sont pas prises en compte.

Itérativement, l'ensemble des transformations est affiné afin que les différences de valeur des pixels reprojétés dans l'image mosaïquée soient les plus petites possibles.

Exemple de mosaïquage : La figure 1.2 montre trois photographies prises sur la grande muraille de Chine. Ces trois photographies font partie d'une séquence de vingt images acquises lors d'une rotation de la caméra légèrement supérieure à 360° autour de l'axe vertical. Le logiciel Autostitch de Brown et Lowe dont le fonctionnement est décrit dans [43] est utilisé pour générer une image panoramique en projection sphérique.

L'un des problèmes inhérents à l'augmentation du champ de vision par mosaïquage d'images est son incapacité à acquérir des scènes en mouvement. Ainsi, sur la figure 1.2, le panoramique présente des défauts à cause du mouvement de la personne entre les images (b) et (c). Il y a en effet apparition d'un 'fantôme', dû au mixage des valeurs provenant des différentes images. Ici, ce mouvement parasite ne couvre qu'une petite portion de l'image et n'a donc pas fait échouer l'estimation d'homographie car celle-ci est robuste, ce qui signifie qu'elle est capable de fournir un résultat correct même en présence d'un certain nombre de données erronées. Néanmoins, il n'a pas été possible de déterminer automatiquement qu'il y avait eu mouvement et d'utiliser la texture du fond pour la zone considérée de la mosaïque reconstruite.

Enfin, le centre optique de la caméra doit rester fixe pour éviter l'apparition d'une disparité entre les différentes images. Celles-ci doivent donc idéalement être acquises alors que la caméra décrit une pure rotation autour de son centre optique. En pratique, une légère translation de la

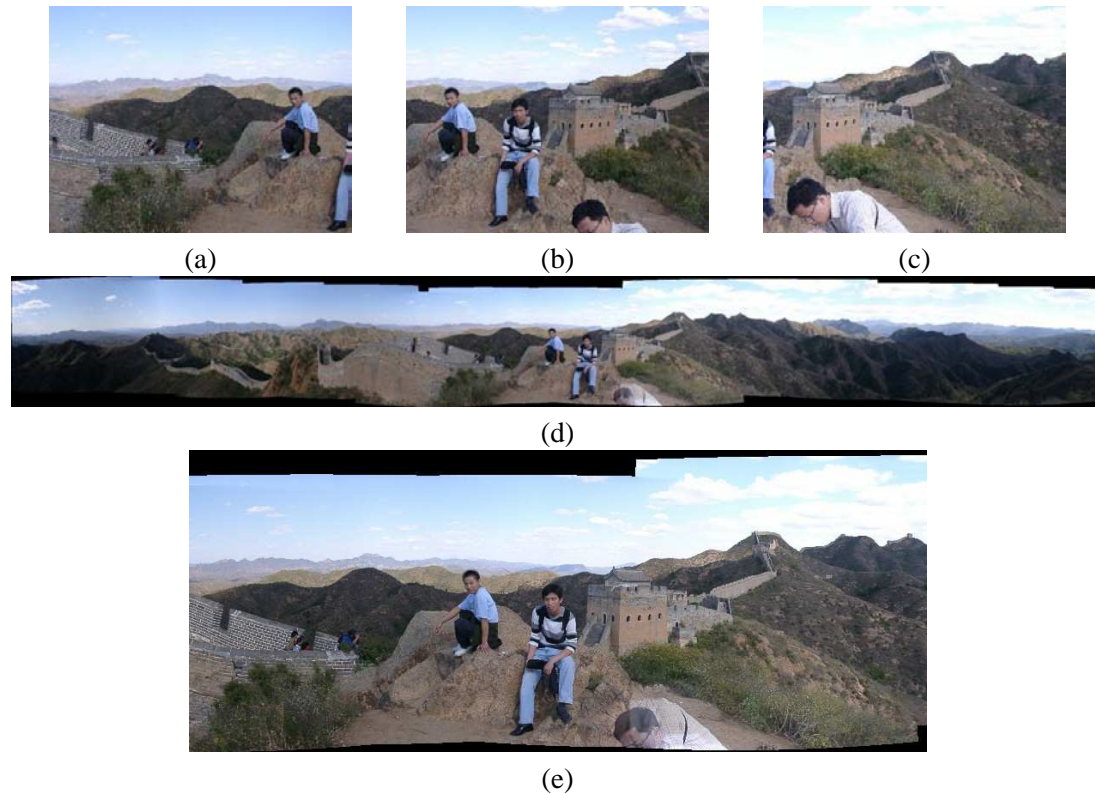


FIG. 1.2 – (a), (b) et (c) Trois photographies acquises avec une caméra standard depuis la même position. (d) L'image panoramique en projection sphérique. (e) Gros plan sur la partie centrale.

caméra peut être tolérée si le mouvement est faible par rapport à la distance des objets observés.

✓ *Apport de la thèse présenté au chapitre 2* \implies La caméra ORIENT-CAM équipée d'un capteur permettant de connaître son orientation. Elle est utilisée pour générer des images omnidirectionnelles orientées par mosaïquage.

1.1.1.2 Une caméra tournée manuellement avec un axe de rotation contraint

Lorsque le mouvement de la caméra est contraint en rotation autour d'un unique axe (l'axe vertical, par exemple, aligné avec l'axe des colonnes de l'image est considéré dans le reste du paragraphe), il est possible de générer des images panoramiques en utilisant d'autres approches. Ici, c'est l'apparence de l'image mosaïquée plutôt que sa géométrie (basée sur un critère de distance entre des points d'intérêt) qui permet de mesurer la qualité de l'image reconstruite.

Une approche simpliste consiste à générer l'image panoramique en recollant des ensembles de colonnes (tranches) situées au centre des différentes images acquises. La largeur de ces

tranches et la position à laquelle elles doivent être recopiées sont fonction du mouvement de la caméra. Celui-ci est estimé en utilisant une mesure de corrélation prenant en compte les parties gauches et droites des images. La figure 1.3 montre le résultat d'une telle méthode. Cette approche ne fonctionne correctement que si la scène est statique. Un mouvement de la scène projeté au centre de l'image entraîne une mauvaise reconstruction car les bandes recopiées depuis des images voisines dans le temps ne se correspondent alors plus dans l'image. Un mouvement de la scène projeté sur les côtés de l'image cause une mauvaise estimation du mouvement de la caméra, ce qui génère aussi des erreurs au niveau du panoramique reconstruit.



FIG. 1.3 – Une image panoramique générée simplement par collage de bandes verticales après estimation du mouvement par recherche de corrélation maximale.

Parmi les travaux récents dans ce domaine, ceux de Wexler et Simakov [44] sont particulièrement intéressants, puisqu'ils permettent de générer des panoramiques même lorsque la scène est mobile. Le fond (fixe) de la scène doit néanmoins couvrir une grande partie des images et les objets en mouvement doivent être vus entièrement au moins sur une image pour que la méthode fournisse de bons résultats. Une analyse d'un volume 3D défini par l'image et le temps permet de détecter la cohérence spatio-temporelle de la scène, ce volume étant simplement la juxtaposition des différentes images. Des tranches verticales d'images sont utilisées comme montré précédemment mais elles peuvent être situées à des positions quelconques dans les images. Le choix des tranches à utiliser est effectué en cherchant un chemin dans le volume image/temps de manière à minimiser les artefacts sur le panoramique reconstruit. Enfin, il faut noter que cette méthode fonctionne aussi pour générer des images en cas de mouvement de translation bien qu'il y ait alors disparité entre les images.

La figure 1.4 montre une image générée avec cette méthode. On peut y noter un mosaïquage relativement correct des objets mobiles malgré des mouvements importants. Anecdotiquement, la rotation effectuée par la caméra étant supérieure à 360 degrés, il est possible de voir les personnes présentes dans différentes positions.

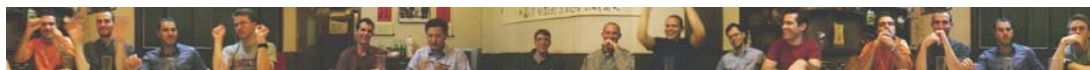


FIG. 1.4 – Une image panoramique générée en employant la méthode spatio-temporelle de Wexler et Simakov.

Rav-Acha et al. ont proposé dans [45] une approche semblable (analyse dans un volume) afin de reconstruire des vidéos panoramiques, qu'ils appellent *Dynamosaics*. Au lieu de juxtaposer des bandelettes verticales d'images, ce sont des bandelettes de vidéo qui sont mises bout à bout à la fois dans l'espace et dans le temps. Cet artifice fonctionne étonnamment bien pour les scènes comportant des phénomènes répétitifs dans le temps tels que des chutes d'eau ou des feuilles dans le vent.

On pourra néanmoins noter que toutes ces méthodes, si elles fournissent des images agréables à regarder, ne respectent pas un modèle de projection géométrique strict, tel que cylindrique ou sphérique. Elles correspondent en fait à des images en projection perspective par morceaux. C'est pour cette raison que les droites des scènes sont souvent découpées en plusieurs segments sur les images, comme par exemple au centre de la figure 1.4.

✓ *Apport de la thèse présenté au chapitre 2* \implies La caméra ORIENT-CAM facilite le mosaïquage d'images de scènes en mouvement.

1.1.1.3 Une caméra motorisée

Un moyen de simultanément contraindre et connaître le mouvement de la caméra consiste à la motoriser en rotation. Il n'est alors plus nécessaire de détecter la pose relative des différentes images, ce qui évite des erreurs d'estimation.

De multiples caméras utilisant ce principe ont été développées, par exemple par Benosman et al. dans [46]. Au sein du laboratoire IRIT, Philippe Puech et Jean Conter ont quant à eux développé deux caméras basées sur ce principe (voir figure 1.5). Romain et al. ont proposé dans [47] une caméra panoramique 3D active à base de deux capteurs d'image et d'un projecteur laser.

Contrairement à ce qui vient d'être montré, la caméra employée ne fournit pas nécessairement une image plane, une image linéaire suffisant si la caméra tourne lentement par rapport à la fréquence d'acquisition de la ligne image. L'utilisation de caméra linéaire rend par contre impossible l'application de méthodes permettant l'acquisition de scènes mobiles telles que celle montrée au paragraphe précédent.

D'un autre côté, les capteurs CCD linéaires disposent habituellement d'une plus haute résolution que les capteurs matriciels. La résolution autour de l'axe de rotation est quant à elle directement liée à l'échantillonnage angulaire, qui dépend de la vitesse de rotation de la caméra et de la fréquence d'acquisition des images.

Avec cette approche, il est aisé de contrôler finement l'exposition et la dynamique, par exemple en acquérant les images avec différents réglages, ce qui permet d'obtenir des images à large champ de vision dites HDR (pour High Dynamic Range, ou grande dynamique) telle que celles proposées par Waggershauer dans [48].

La première version de caméra panoramique, développée à l'IRIT, la PANOCAM, possède un axe de rotation et un capteur CCD 2700 pixels. La seconde version, la PCAM-ST, possède 2 axes de rotation, ce qui lui permet d'avoir un champ de vision quasi omnidirectionnel. De plus,

grâce à un ensemble complexe de prisme et de miroirs, cette dernière est capable d'obtenir une projection de la scène depuis deux positions légèrement espacées vers un unique capteur CCD linéaire de 2700 pixels. Ceci permet la stéréoscopie et donc la reconstruction en 3D de l'environnement.

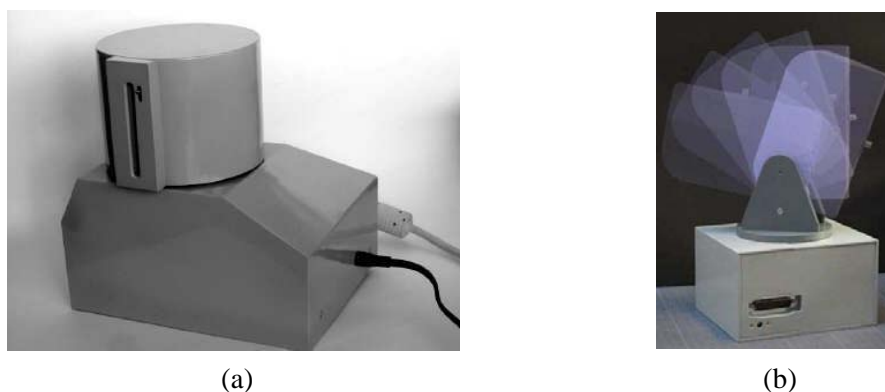


FIG. 1.5 – Caméras utilisant des capteurs CCD linéaires en rotation. (a) PANOCAM, une caméra panoramique à un axe de rotation. (b) PCAM-ST, la version stéréoscopique et quasi omnidirectionnelle (2 axes de rotation).

1.1.2 Des caméras avec champ de vision étendu

L'utilisation d'une caméra en rotation, si elle permet d'obtenir des images avec des résolutions assez importantes présente le gros inconvénient de ne pas pouvoir percevoir l'environnement de manière omnidirectionnelle instantanément. La solution à ce problème trouvée par la nature consiste en des yeux possédant un champ visuel plus important. Ainsi, chez les mammifères, le cheval possède un champ visuel couvrant presque 340° . Chez les insectes, des structures différentes d'yeux, constituées de multiples cellules, les ommatidies, permettent d'obtenir des champs visuels encore plus étendus. La figure 1.1 (d) montre une tête de mouche sur laquelle on peut discerner les différentes facettes. Chez l'araignée sautante, ce sont jusqu'à quatre paires d'yeux qui lui permettent d'avoir un champ visuel important.

Pour des contraintes technologiques, les capteurs électroniques de lumière, qu'ils soient en CMOS ou CCD, sont plans. Il n'est pas facile à l'heure actuelle de courber cette surface sensible, ni de développer une optique associée. Pour ces raisons, deux autres approches sont principalement utilisées pour augmenter le champ de vision. La première consiste en l'utilisation de plusieurs capteurs plans, orientés dans différentes directions. La seconde consiste à dévier la lumière en employant des lentilles ou des miroirs.

1.1.2.1 Plusieurs caméras

La figure 1.6 montre trois exemples de caméras comportant plusieurs capteurs. La caméra RingCam de Microsoft en comporte cinq et permet d'acquérir des vidéos panoramiques [49].

La Caméra Ladybug de Point Grey Research utilisée notamment dans [50] par Sato et al. ou dans [51] par Kangni et al. en comporte le même nombre mais disposées différemment, de manière à ce que leurs centres optiques soient confondus et qu'une des caméras pointe vers le haut. Enfin, les caméras First Person d'Incited Media sont constituées de plusieurs caméras observant une surface réfléchissante en forme de pyramide, ce qui permet d'aligner les reflets des centres optiques des différentes caméras alors que ceux-ci n'auraient pas pu être alignés directement pour cause d'encombrement. Il faut noter qu'ici, les miroirs sont localement plans et n'augmentent nullement le champ visuel individuel des caméras.

Avec ce type de caméras, si les centres optiques sont alignés, les différentes homographies entre les images peuvent être estimées une fois pour toutes. Ensuite la mosaïque peut être reconstruite à la fréquence d'acquisition des caméras standard qui composent l'ensemble. Ces caméras permettent d'atteindre des résolutions raisonnables, par contre elles ne s'affranchissent pas des problèmes liés aux différentes expositions, à moins que le contrôle d'exposition des différentes caméras soit centralisé.

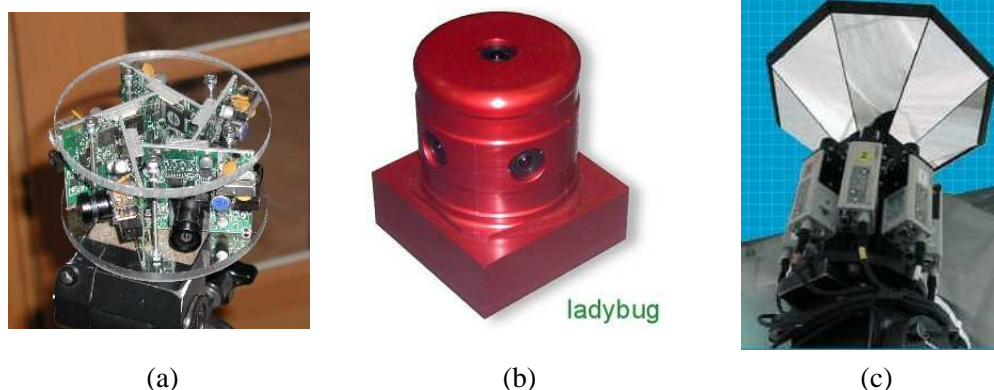


FIG. 1.6 – Caméras à large champ de vision par emploi de plusieurs caméras directionnelles. (a) Caméra Ring Cam de Microsoft, (b) Caméra LadyBug, (c) Caméra First Person.

1.1.2.2 Une caméra unique à large champ de vision

Il existe différents moyens d'augmenter le champ de vision d'une caméra en collectant la lumière puis en la projetant sur une surface sensible plane. Lorsque le champ de vision croît, ceci est de plus en plus difficile à réaliser avec une optique de caméra standard. Il est alors possible d'utiliser un ensemble complexe de plusieurs lentilles pour augmenter le champ de vision. On parle alors de systèmes dioptriques. Il est aussi possible d'utiliser des miroirs pour dévier la lumière, ce que font les caméras catadioptriques. La figure 1.7 montre des images obtenues par chacun de ces types de caméras.

Les caméras grand angle se révèlent être très intéressantes pour les applications de Structure From Motion (SFM), qui consistent à estimer simultanément la géométrie de la scène et le déplacement de la caméra à partir des images de celle-ci [52, 53, 54, 55, 56]. En effet, ces ca-



FIG. 1.7 – Images de caméra unique à large champ de vision (a) provenant d'une optique fisheye (Image © Derek Langley 2006 / www.darknessandlight.co.uk). (b) provenant d'une caméra catadioptrique.

méras, contrairement aux caméras standard, ne sont pas dans une configuration instable lorsque le déplacement de la caméra est effectué dans l'axe de visée. Il est donc possible d'obtenir une reconstruction de la scène quel que soit le mouvement appliqué à la caméra.

L'utilisation de ces caméras génère au niveau de l'image ce que beaucoup appellent des *distorsions*. Ce mot n'est pas réellement adéquat, et il est préférable de parler d'images obtenues par des *projections différentes* de la projection perspective plane, et avec lesquelles l'image d'une droite n'est pas une droite. Il est possible dans certains cas, de générer une image proche (nous reviendrons sur cette notion dans le paragraphe sur la modélisation des capteurs à centre de projection unique) d'une projection perspective à partir de l'image d'un fisheye, mais cette image ne peut couvrir tout le champ de vision car la projection plane est limitée à 180° . En pratique, pour éviter des dimensions d'image trop importantes, l'angle de vision pour une projection perspective est plutôt limité aux alentours de 160° .

Les outils de la vision perspective doivent être utilisés avec beaucoup de précautions et généralement il est nécessaire de les réétudier spécifiquement pour ces caméras. Ainsi, des algorithmes tels que la détection d'images de droites, la recherche de points homologues, le filtrage spatial ne sont pas réutilisables directement.

✓ *Apport de la thèse présenté au chapitre 3* \implies Une méthode spécifique de détection d'images de droites dans les images d'une caméra à large champ avec miroir paraboloidal.

A titre d'exemple, l'adaptation de la transformée de fourrier 2D est réalisable pour les caméras omnidirectionnelles centrales en utilisant les harmoniques sphériques comme l'ont proposé Geyer et al. dans [57].

1.1.2.3 Caméras dioptriques grand angle

Le Fisheye, bien connu des photographes, est un complément d'optique grand angle permettant d'atteindre des champs de vision hémisphériques ou supérieurs. Il a initialement été développé pour l'observation du ciel en astronomie. Son utilisation grand public s'est ensuite développée grâce à la production de masse de la lentille Nikon de 16 mm à la fin des années 60. Le fisheye possédant le champ de vision le plus large (220°) a été créé par la même compagnie et pèse plus de 5 kilogrammes. Le judas situé sur une porte est une version bien plus légère d'optique fisheye. Néanmoins, en photographie, ce type d'optique est généralement assez encombrant et coûteux, car constitué d'un grand nombre de lentilles, ceci notamment afin d'éviter les aberrations chromatiques, dues à la mauvaise convergence des rayons de longueurs d'ondes différentes.

Leur gros avantage sur les caméras catadioptriques est que le champ visuel n'est pas occulté par la caméra à cause de la réflexion dans le miroir (voir au centre des images de la figure 1.7 (a)). Par contre, hormis un coût élevé, ils présentent l'inconvénient d'être relativement grossièrement modélisés par une caméra centrale. En effet, les différents rayons de projection d'une telle caméra ne s'intersectent pas en un point, mais génèrent une caustique. En fonction de l'application, cette caustique pourra ou non être assimilée à un unique point. Nous reviendrons dans la suite du document sur ce point.

La figure 1.8 montre deux caméras équipées d'optiques fisheye. Sur la droite, le prototype SphereCam [58], possède un champ de vision complètement omnidirectionnel grâce à l'emploi de deux optiques fisheye montées dans des directions opposées.



(a)



(b)

FIG. 1.8 – Caméras utilisant des compléments optiques de type fisheye. (a) une version de base, (b) SphereCam, une version omnidirectionnelle.

De nouveaux types de caméras utilisant des lentilles commencent à être développées, inspirées du modèle de l'œil de la mouche. L'équipe de nanotechnologie biomoléculaire de Berkeley [59] constitue ainsi des ommatidies artificielles en résine. Ces microlentilles, situées sur une sphère, captent la lumière qui est ensuite conduite vers les photos-récepteurs via des fibres optiques. Actuellement, les prototypes utilisent des photos-récepteurs spécialisés mais l'emploi

de capteurs matriciels standard est prévu, ce qui devrait permettre l'emploi massif de cette technologie.

1.1.2.4 Caméras catadioptriques

Les caméras catadioptriques comportent des miroirs dans lesquels une caméra observe le reflet de la scène. Leur nom provient des dioptries, les éléments réfractifs, et de catoptrique qui signifie *la science de la vision réfléchie* d'après [60]. Le miroir est utilisé pour obtenir le même effet qu'une courbure de la rétine grâce à la réflexion. La plus grande partie de cette thèse traite de ce type de caméras.

Le premier brevet industriel d'un système catadioptrique a été déposé en 1970, par D. W. Rees [61]. Ce système était destiné à augmenter le champ de vision d'un opérateur assis dans une tourelle de tir, plusieurs images en projection perspective étant générées à partir de l'image fournie par une caméra fixant un miroir hyperboloïdal.

C'est seulement en 1990 que Yagi propose l'utilisation d'une caméra observant un miroir conique dans une application de robotique mobile [62]. Plus tard, en 1997, Baker & Nayar ont introduit la géométrie des caméras catadioptriques panoramiques dans la communauté de la vision par ordinateur dans [63].

Depuis, l'engouement des chercheurs pour les caméras catadioptriques n'a cessé de s'accroître et de très nombreux travaux ont été réalisés, surtout dans le domaine de la robotique mobile. Le gros avantage de ces caméras, leur champ de vision large, fournit alors aux robots la possibilité d'appréhender leur environnement de manière plus globale qu'avec les œillères que représentent les caméras standard. Elles permettent, à moindre frais car une seule caméra est nécessaire, de couvrir 360° autour d'un axe. L'augmentation du champ de vision d'une caméra par emploi d'un miroir se fait évidemment au détriment de la résolution à capteur d'image identique. Néanmoins, ce manque peut être comblé par l'utilisation conjointe avec une ou plusieurs caméras standard. La caméra à large champ de vision se comporte alors en détecteur de zones d'intérêt, vers lesquelles le robot peut pointer ses caméras directionnelles pour percevoir les détails.

Nous nous intéressons ici uniquement aux caméras catadioptriques destinées à élargir le champ de vision. On pourra cependant noter une autre utilisation intéressante des miroirs qui consiste à obtenir deux images différentes en projection perspective avec seulement une caméra et deux miroirs plans tel que l'ont proposé Gluckman et Nayar dans [64, 65].

La figure 1.9 présente trois caméras catadioptriques différentes. La première est la caméra paracatadioptrique qui a été utilisée dans la plupart des expériences de cette thèse. Elle est constituée d'un appareil photo 4 mégapixels et d'un complément paracatadioptrique OneShot 360 commercialisé par la société PANOSMART. La deuxième est une caméra hypercatadioptrique utilisée par Nishimura dans [66]. Cette caméra est équipée d'un cache transparent assurant le rôle de protection contre les rayures et les poussières. Pour éviter les reflets parasites engendré par ce cache en plastique, un cône central opaque est employé. Enfin la troisième est une caméra panoramique stéréoscopique réalisée par Fiala [6] utilisant un double miroir sphérique. L'intérêt de cette caméra réside dans le fait qu'un seul capteur d'image est nécessaire, les deux images de la scène provenant des deux reflets dans les miroirs. Néanmoins, ce

genre d'assemblage est peu utilisé à cause de la faible résolution de l'image correspondante au miroir central et à la faible disparité liée au rapprochement des miroirs, qui implique une faible précision sur l'estimation de la profondeur.

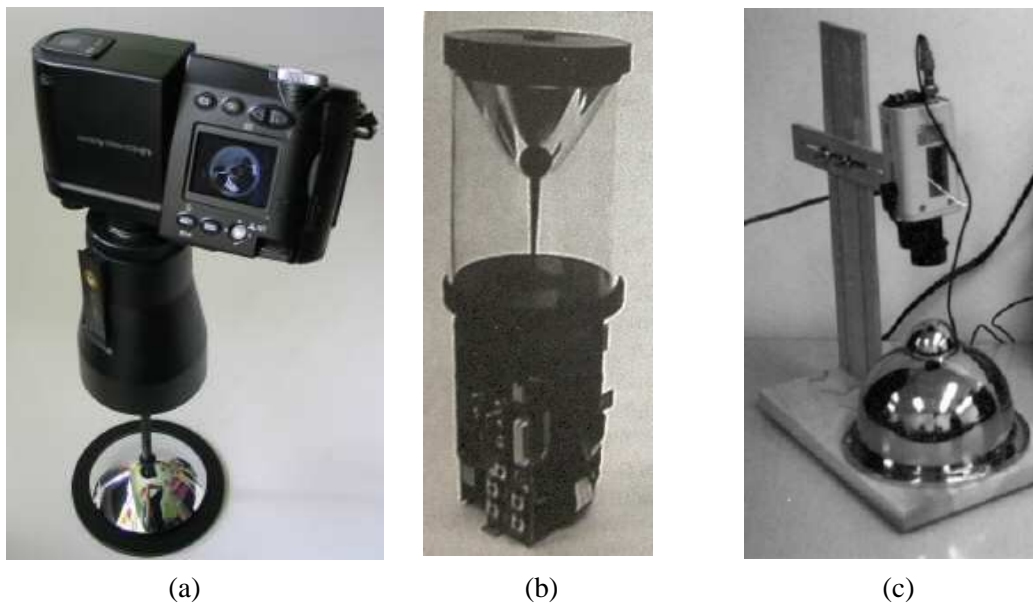


FIG. 1.9 – Exemples de caméras catadioptriques. (a) Caméra paracatadioptrique avec complément OneShot 360, (b) Caméra avec miroir hyperboloïdal. (c) Caméra stéréoscopique avec double miroir sphérique.

En combinant différentes caméras et miroirs, il est possible d'obtenir des caméras catadioptriques très variées. Un exemple marquant est fourni par Wachs et al. dans [67]. Ces derniers utilisent une caméra perspective haute résolution pointée vers 31 miroirs sphériques.

Pour plus d'informations: Les problèmes liés à la mise au point, l'astigmatisme, etc... pour les différents capteurs ne sont pas abordés ici. Chacun possède des propriétés particulières et le lecteur est invité à consulter [18] pour de plus amples informations sur les caméras catadioptriques.

1.1.2.5 Les caméras catadioptriques centrales

Certaines caméras catadioptriques possèdent un centre de projection unique, qui est un point par lequel les rayons lumineux de la scène doivent passer pour être perçus par la caméra. Cette propriété est très intéressante puisqu'elle permet la reprojexion de l'image de la caméra sur une surface quelconque et l'emploi de la géométrie épipolaire, permettant une recherche efficace des points homologues dans les applications de stéréoscopie. Le calcul de la projection d'un point sur l'image est très simple dans ce cas puisque l'on sait que tous les rayons de projection convergent en un point. Nous verrons au [chapitre 4](#) que ceci est bien plus complexe

dans le cas des caméras non centrales.

Les assemblages de miroir et de caméra permettant d'obtenir un centre de projection unique sont décrits ici. Il est important de noter que ce n'est pas uniquement le miroir qui permet le caractère central mais l'ensemble formé par le miroir et la caméra. La figure 1.10 présente certains des cas les plus intéressants car technologiquement réalisables. Dans les cas où la caméra qui observe le miroir est centrale, elle sera dénommée perspective pour éviter les confusions avec le caractère central de l'ensemble catadioptrique.

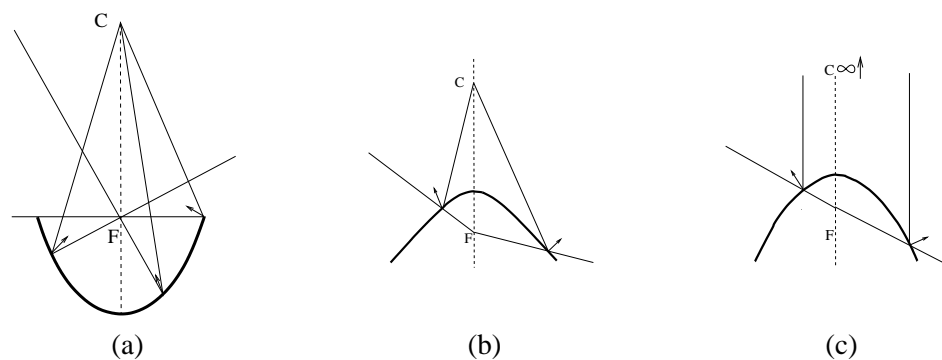


FIG. 1.10 – Trois caméras catadioptriques centrales : Les rayons de projection des différents points de la scène s'intersectent en un foyer situé sur l'axe de révolution du miroir. (a) miroir ellipsoïdal et caméra perspective. (b) miroir hyperboloïdal et caméra perspective. (c) miroir paraboloidal et caméra télécentrique.

Miroir ellipsoïdal et caméra perspective : Le miroir employé est concave. La caméra est placée à l'intérieur, avec son centre optique exactement situé sur le second foyer de l'ellipsoïde. L'ensemble possède un centre de projection unique situé au niveau du premier foyer. Ce montage est assez peu utilisé en pratique car son champ vision est limité à l'hémisphère moins la partie occultée par la caméra.

Miroir hyperboloïdal et caméra perspective : Cette fois, le miroir est convexe et permet un champ visuel supérieur à l'hémisphère. Le miroir physique n'occupe qu'une seule des deux parties de l'hyperboloïde. Comme pour le miroir ellipsoïdal, le centre optique de la caméra perspective doit coïncider parfaitement avec le second foyer de l'hyperboloïde, ce qui est difficile à assurer en pratique. Cet assemblage porte le nom de caméra hypercatadioptrique.

Miroir paraboloidal et caméra télécentrique : Ce montage ressemble au précédent, sauf que le second foyer du miroir est situé à l'infini. Une caméra réalisant une projection orthographique grâce à une optique télécentrique est nécessaire pour obtenir un centre de projection unique dans ce cas. Le gros intérêt par rapport aux autres montages est la plus grande facilité d'alignement de la caméra et du miroir. En effet, seul l'axe de visée doit être aligné avec l'axe

de révolution du parabolioïde. La translation de la caméra n'a pas d'influence sur le fait que la caméra soit centrale. Par contre, elle influe sur le champ de vision à cause de l'occultation due à la caméra. Grâce à cette invariance à la translation, il est possible de translater librement la caméra pour obtenir le meilleur compromis encombrement/champ visuel en fonction de l'application visée. Cet assemblage porte le nom de caméra paracatadioptrique.

L'inconvénient de la caméra paracatadioptrique est lié à l'emploi d'une optique télécentrique, qui est relativement chère et encombrante. Néanmoins, celle-ci peut être remplacée par un miroir sphérique et une caméra perspective. Un nouveau problème d'alignement se pose alors pour faire coïncider la caméra perspective avec le centre de la sphère.

✓ *Apport de la thèse présenté au chapitre 3* \implies Une méthode d'étalonnage géométrique de ce type de caméra et de détection des images de droites.

✓ *Apport de la thèse présenté au chapitre 5* \implies Un capteur 3D omnidirectionnel comportant une caméra paracatadioptrique.

Autres cas : D'autres miroirs permettent d'obtenir une caméra centrale. Par exemple, un miroir plan observé par une caméra perspective possède un centre de projection unique situé derrière le miroir, en un point symétrique au centre optique de la caméra perspective par rapport au plan du miroir. Cet assemblage n'augmente pas directement le champ de vision.

Enfin, d'autres assemblages avec une caméra non réalisables en pratique permettent théoriquement un centre de projection unique.

Le miroir conique permettant un centre de projection unique : En s'affranchissant du modèle de projection en trou d'épingle pour la caméra et en utilisant les lois optiques des lentilles minces, Lin et Bajcsy ont montré dans [68] qu'il est possible d'obtenir une caméra à centre de projection unique en utilisant un miroir conique et une caméra standard. Le capteur obtenu, s'il n'a pas un champ visuel très important, possède de très bonnes propriétés. Ainsi le flou (defocus blur) généré par la courbure des autres miroirs utilisés dans les caméras catadioptriques n'est pas présent avec les miroirs coniques, qui sont plans au niveau des coupes coplanaires à l'axe du cône.

1.1.2.6 Les caméras catadioptriques non centrales

En général, hormis les cas présentés précédemment, les caméras catadioptriques n'admettent pas un centre de projection unique et sont donc non centrales. On appelle *caustique* la surface générée par les différents rayons de projection. Dans le cas d'une réflexion, le terme *catacaustique* est préféré. Cette surface est tangente à l'ensemble des rayons de projection de la caméra après réflexion sur le miroir. Swaminathan et al. ont proposé dans [69] une étude très intéressante de ces surfaces. Ieng et Benosman fournissent une étude détaillée des caustiques

et de leur application aux caméras catadioptriques dans [70].

L'utilisation des caméras non centrales est relativement complexe par rapport à celles disposant d'un unique centre de projection. Malgré la complexité accrue des modèles de projection, ces caméras peuvent présenter d'autres avantages. Par exemple, les miroirs sphériques et coniques sont faciles à produire et donc peu coûteux. D'autres miroirs permettent d'obtenir des résolutions plus homogènes dans l'image. Il est aussi possible d'optimiser l'utilisation de la surface sensible du capteur d'image en diminuant la surface occupée par le reflet de la caméra.

La figure 1.11 montre trois exemples de caméras catadioptriques non centrales.

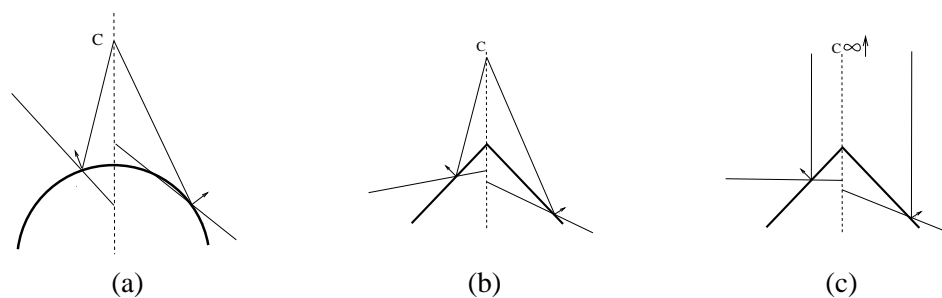


FIG. 1.11 – Trois caméras catadioptriques non centrales : Les rayons de projection des différents points ne s'intersectent pas en un point unique. (a) miroir sphérique et caméra perspective. (b) miroir conique et caméra perspective. (c) miroir conique et caméra télécentrique.

1.1.2.7 Variation angulaire de la résolution

Une particularité des caméras catadioptriques est la forte variation de résolution angulaire radiale dans l'image. Une caméra standard possède en effet une résolution angulaire qui varie peu lorsque l'on s'éloigne du centre de l'image. A l'opposé, celle-ci peut être très différente dans le cas des caméras catadioptriques. Sur la figure 1.12, l'angle entre deux rayons de projections associés à deux pixels voisins est tracé en fonction de la distance au centre de l'image pour deux caméras différentes. La première (a) est une caméra trou d'épingle, disposant d'un champ de vision de 21° . La seconde (b) est une caméra paracatadioptrique dont le champ visuel couvre 102° . Les courbes montrent une variation d'angle de moins de 13% entre les pixels du centre de l'image et ceux de la périphérie pour la caméra standard alors qu'il y a plus de 63% d'écart avec la caméra catadioptrique.

Des capteurs d'images ou des miroirs spécifiques : Dans certaines applications, il est préférable d'avoir une résolution à peu près constante. Gachter et al. [71] ont tenté de l'obtenir en combinant un miroir d'une forme particulière avec un capteur CCD log-polaire en remplacement du capteur matriciel. Berry et al. proposent une approche similaire dans [72] ainsi qu'Ollis et al. dans [73]. Chahl et Srinivasan dans [74] étudient les formes de miroir permettant d'optimiser l'homogénéité de résolution angulaire. Avec un tel capteur, il est possible d'échantillonner angulairement l'espace avec un pas constant. L'image fournie peut alors directement

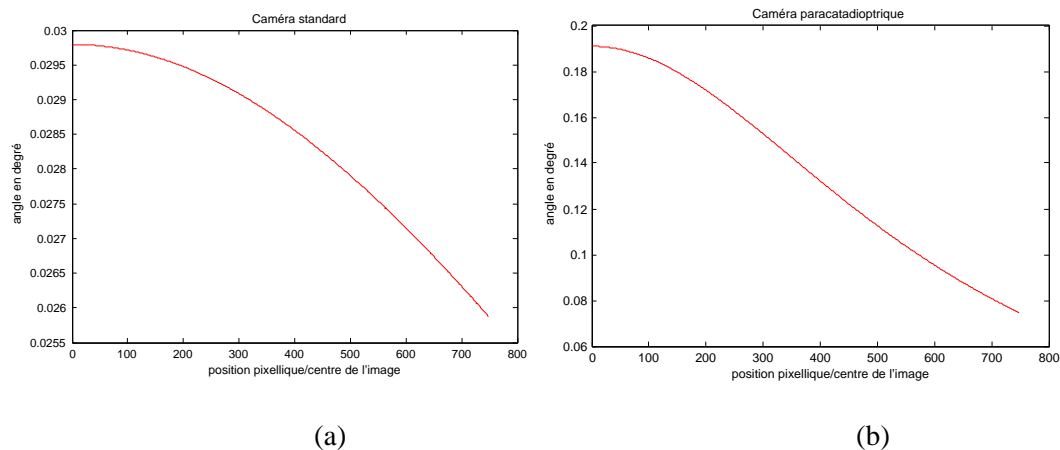


FIG. 1.12 – Angle entre deux rayons de projection associés à des pixels voisins en fonction de la distance par rapport au centre de l'image (a) dans le cas d'une caméra standard, (b) dans le cas d'une caméra paracatadioptrique.

être reprojctée sur un cylindre. Malgré cet avantage, ce type de capteur est peu utilisé car il est difficile à produire et surtout à aligner avec le miroir. Ce dernier point est pourtant crucial, car si le centre du capteur d'image n'est pas exactement aligné avec l'axe de révolution du miroir, le capteur perd ses bonnes propriétés.

Une caméra avec miroir permettant d'approximer une caméra grand angle perspective : Hicks et al. proposent dans [75, 76] deux familles de surfaces réfléchives permettant d'obtenir des images assez proches de celles qui seraient obtenues par projection perspectives pour des objets situés dans un plan perpendiculaire à l'axe de révolution du miroir. Grâce à leur approche, il est possible d'utiliser des outils de la vision standard pour certaines tâches, les images de ces caméras ayant de bonnes propriétés d'homogénéité de résolution.

1.2 Modèles de projection pour la vision panoramique et omnidirectionnelle

Un modèle géométrique de caméra modélise le processus de formation de l'image au niveau géométrique et permet d'exprimer les relations existantes entre les points 3D d'une scène et leurs projections sur l'image de la caméra. Ce modèle est nécessaire lorsque l'on souhaite connaître soit la position pixellique de l'image d'un point de l'espace (*modèle direct* de la caméra) soit le rayon de projection correspondant à une position dans l'image (*modèle inverse* de la caméra).

1.2.1 Exemple de modèle pour caméra standard : le trou d'épingle

Le modèle le plus simple pour une caméra 2D est le modèle trou d'épingle, aussi appelé *pin-hole*. Il s'agit d'un modèle de caméra centrale, dont le centre est précisément le trou d'épingle. La lumière qui atteint la surface sensible de la caméra doit passer par ce trou et emprunte le chemin le plus court, la ligne droite. L'intérêt de ce modèle est qu'il approxime correctement la projection effectuée par une caméra standard et qu'il correspond, au niveau des calculs, à un simple produit matriciel en coordonnées homogènes.

Un modèle est dit *paramétré* s'il dépend de paramètres. Ainsi, un modèle de ce type peut être commun à un ensemble de caméras, l'adaptation à chacune d'elles se faisant en jouant sur ses paramètres. Ceux-ci sont communément séparés en deux groupes.

Les paramètres *intrinsèques* influent sur les propriétés propres au capteur, qui ne dépendent pas du point de vue. Il faut noter que c'est toute la chaîne d'acquisition des images qui est modélisée ici, incluant la caméra, l'échantillonnage effectué par la carte d'acquisition. Dans le cas du trou d'épingle, une matrice de projection K contenant ces paramètres exprime la relation entre l'image et les points exprimés dans un repère ω_C lié à la caméra. Ce repère est positionné sur le centre de projection et ses axes x et y sont alignés respectivement avec les axes des lignes et des colonnes u et v du plan image. L'expression de K peut être plus ou moins complexe, en fonction du nombre de paramètres pris en compte (pixels carrés ou non, orthogonalité des lignes et colonnes, etc...).

Les paramètres *extrinsèques*, à l'opposé, régissent le placement relatif de la caméra par rapport à la scène et ne font pas à proprement parler partie du modèle de projection. Ils permettent néanmoins de passer des coordonnées de points exprimées dans un repère de travail ω_T lié à la scène à celles exprimées dans le repère de la caméra ω_C , qui sont projetées par le modèle. Cette transformation est modélisée par une matrice de transformation rigide M .

Pour le trou d'épingle, dans le cas où les lignes et colonnes du capteur d'image sont orthogonales, la matrice K est exprimée en fonction des paramètres de focales α_u et α_v selon les axes de l'image u et v . La projection orthogonale du centre optique sur le plan image est paramétrée par p_u et p_v , aussi appelés coordonnées du point principal.

La matrice M est définie à partir de la matrice de rotation R qui amène les axes de ω_T dans la même orientation que ceux de ω_C et du vecteur de translation t qui représente la position du centre du repère ω_C exprimé dans ω_T . Les coordonnées homogènes de l'image d'un point P dans ω_T sont alors obtenues par un simple produit. Les coordonnées cartésiennes du point dans l'image correspondent au représentant de p pour lequel w est égal à un. La figure 1.13 montre la projection vers le plan image.

Pour plus d'informations: Les caméras standard ont été largement étudiées. L'ouvrage [77] propose une bonne synthèse des modèles géométriques correspondant, notamment les modèles de caméras prenant en compte les distorsions radiales et tangentiels. Des bases de géométrie projectives sont aussi proposées.

$$M = \begin{pmatrix} R & -Rt \end{pmatrix}; P_{(\omega_T)} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}; P_{(\omega_C)} = M P_{(\omega_T)} \quad (1.1)$$

$$K = \begin{pmatrix} \alpha_u & 0 & p_u \\ 0 & \alpha_v & p_v \\ 0 & 0 & 1 \end{pmatrix}; p = \begin{pmatrix} u \\ v \\ w \end{pmatrix}; p = K P_{(\omega_C)} \quad (1.2)$$

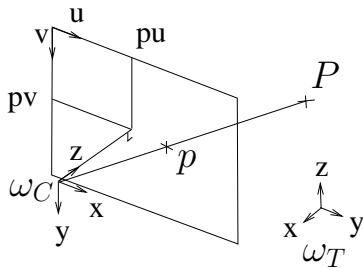


FIG. 1.13 – La projection d'un point P en p avec la caméra trou d'épingle.

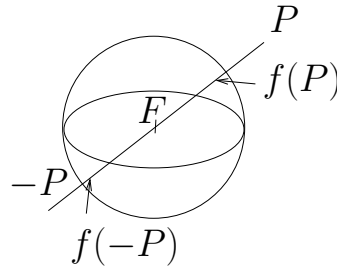


FIG. 1.14 – Les différentes projections de deux points P et $-P$ dans le cas d'une caméra dont le champ de vision dépasse l'hémisphère.

1.2.2 Des modèles adaptés aux caméras à large champ de vision

Les modèles des caméras à large champ de vision sont généralement plus complexes et ne peuvent être exprimés directement sous forme matricielle. On peut les classer en deux familles, selon que le modèle effectue une projection centrale ou non.

Comme il a été montré précédemment, les dispositifs permettant d'obtenir des images avec des champs de vision importants sont variés. Il en résulte une grande diversité des modèles correspondant. Ceux-ci pourront correspondre à des configurations statiques si le capteur est supposé immobile ou bien il sera possible d'y intégrer la connaissance a priori du mouvement du capteur (par exemple la rotation, dans le cas d'un capteur CCD linéaire en rotation). De même, il existe des modèles qui modélisent plusieurs caméras simultanément.

Lors du choix d'un modèle pour une caméra, il est nécessaire de trouver un compromis entre faible complexité du modèle et adéquation avec la réalité. En effet, il est toujours possible de simplifier celui-ci en faisant des hypothèses (par exemple, on peut supposer un alignement parfait entre l'optique et le miroir d'une caméra catadioptrique). On diminue alors le nombre de paramètres et donc la complexité des méthodes pour les estimer et pour utiliser le modèle. Mais un modèle trop simple ne permet pas de bien représenter le système réel. Le choix doit donc être effectué en fonction de l'application au cas par cas.

Par exemple, comme nous l'avons vu précédemment, les conditions requises pour qu'une caméra catadioptrique soit centrale sont strictes. On est néanmoins souvent amené à utiliser des modèles centraux pour des caméras non parfaites afin de pouvoir appliquer des méthodes destinées uniquement aux caméras centrales (par exemple, la rectification épipolaire, présentée au [chapitre 5](#)).

On pourra enfin noter qu'il est possible d'utiliser plusieurs modèles pour une même caméra. Ainsi Micusik et al. proposent dans [52], d'utiliser un modèle central pour initier une reconstruction 3D à partir de plusieurs images. Un modèle de caméra non centrale est utilisé dans un second temps afin de raffiner la reconstruction.

✓ *Apport de la thèse présenté au chapitre 5* \implies Lors de l'utilisation de la caméra paracatadioptrique dans le capteur 3D omnidirectionnel, un choix de modèle a été effectué de manière empirique pour répondre à l'application.

1.2.3 Des modèles de caméras centrales

Un des principaux avantages des modèles de caméras centrales est de pouvoir utiliser un certain nombre de méthodes héritées des caméras standard. Ainsi, il est possible, après adaptation, d'utiliser une géométrie épipolaire permettant une recherche efficace des points homologues pour la stéréoscopie. L'estimation d'homographies est aussi possible. Contrairement aux caméras standard, les points utilisés ne sont pas situés dans le plan de l'image mais doivent être élevés sur une autre surface, telle que celle du miroir ou d'une sphère virtuelle. Il est ainsi possible par exemple de mosaïquer des images issues de caméras catadioptriques pour générer des images omnidirectionnelles à partir d'un nombre réduit d'images. Kang propose une méthode de mosaïquage d'images grand angle dans [78].

Les outils de géométrie projective doivent cependant être utilisés avec précaution pour les caméras à large champ de vision car avec plus de 180° de champ de vision, il est possible de voir dans deux directions opposées. Ainsi, considérons le centre optique au centre d'un repère dans lequel deux points P et $-P$ sont observables par la caméra. Ces deux points ne se projettent pas à la même position et pourtant ils définissent deux rayons de projection colinéaires (voir la figure 1.14). Pour résoudre ce problème, il est nécessaire d'utiliser la géométrie projective orientée, dont Laveau et Faugeras font une description dans [79].

Il existe différents types de modèles, plus ou moins généraux, permettant de modéliser la projection effectuée par différentes caméras.

1.2.3.1 Des modèles unifiés pour les caméras centrales

La diversité des caméras grand angle pose le problème de l'unification, qui permet d'utiliser un modèle commun pour toute une famille de capteurs. Geyer et al. [80] ont proposé une unification de modèle pour diverses caméras centrales, catadioptriques ou de type trou d'épingle sans distorsion optique. Ce modèle est équivalent à une projection en deux étapes. La première est effectuée sur une sphère unitaire par projection centrale depuis le centre de la sphère. Elle

est suivie d'une projection centrale depuis un point sur l'axe reliant les pôles de la sphère. Ce point peut se situer au pôle nord dans le cas de la caméra paracatadioptrique, au centre pour une caméra standard et entre les deux pour une caméra avec miroir hyperboloïdal. Cette équivalence est détaillée dans le [chapitre 3](#) avec la figure 3.3. Ying et Hu ont proposé dans [81], un autre modèle géométrique permettant l'unification des caméras catadioptriques centrales et dioptriques grand angle (fisheye) considérées centrales.

1.2.3.2 Des modèles spécifiques pour les caméras centrales

On pourra néanmoins noter que ces approches unificatrices, si elles fournissent un cadre général à l'étude d'une famille de capteurs, peuvent aussi ajouter des degrés de liberté inutiles. Par exemple, employer le modèle de Geyer et al. [80] pour une caméra paracatadioptrique au lieu d'un modèle spécifique peut poser des problèmes lors de l'étalonnage. Geyer et al. ont d'ailleurs proposé dans [82] un modèle spécifique pour la caméra paracatadioptrique. De nombreux autres modèles spécifiques sont disponibles dans la littérature, par exemple dans [83] pour les caméras hypercatadioptriques.

Un modèle central spécifique aux caméras paracatadioptriques reprenant celui de Geyer et al. [82] est présenté au [chapitre 3](#).

1.2.4 Des modèles de caméras non centrales

Les contraintes d'alignement permettant d'obtenir des caméras centrales sont rarement respectées entièrement. Il en ressort que les réalisations de caméras centrales ne le sont généralement pas exactement. Elle sont par contre plus ou moins bien modélisées par un modèle central selon que l'erreur d'alignement est petite ou grande et il est possible d'utiliser la notion de déviation d'une caméra centrale.

La caméra télécentrique utilisée dans les capteurs paracatadioptriques dévie aussi souvent d'une optique télécentrique parfaite, et la caméra possède en réalité un centre optique situé relativement loin du miroir au lieu d'être à l'infini. Les rayons qu'elle reçoit ne sont donc pas exactement parallèles et cela contribue au fait que la caméra catadioptrique n'est pas exactement centrale.

Le caractère central d'une caméra est donc à définir en fonction de l'application visée, et de la précision nécessaire. Ainsi par exemple, la caméra paracatadioptrique est considérée centrale au [chapitre 3](#) pour la détection de droite et la rectification d'image. Par contre, au [chapitre 5](#), le modèle central s'avère trop approximatif pour permettre une reconstruction 3D précise.

Un modèle de caméra non central peut être défini de plusieurs façons selon le type de caméra. Le plus simple est basé sur l'étude de la trajectoire des rayons lumineux. Les différents éléments optiques et/ou réfléchissants sont modélisés ainsi que leurs positionnements respectifs.

✓ *Apport de la thèse présenté au [chapitre 4](#)* \implies un modèle pour une caméra catadioptrique non centrale avec un miroir de forme quadrique est présenté.

Afin de modéliser une caméra catadioptrique, il est aussi possible d'utiliser une surface pa-

ramétrique représentant sa catacaustique, comme l'ont proposé Swaminathan et al. dans [69]

Bakstein et Pajdla fournissent dans [84] un survol rapide de différents modèles de caméras non centrales. Micusik et al. proposent dans [52] un modèle adaptable à différentes caméras qui possèdent une symétrie le long d'un axe perpendiculaire au plan image.

Pajdla propose dans [85] une modélisation d'un ensemble de caméras non centrales dénommées *obliques*. Des propriétés intéressantes sont décrites pour la stéréoscopie.

Dans [86], Pless fournit un modèle de caméra non centrale composée d'un ensemble de caméras centrales. Ce modèle permet de gérer les images de plusieurs caméras centrales comme si elles émanaient d'une caméra unique disposant de plusieurs centres de projection.

Un modèle de type *Light Field* est employé par Wachs et al. pour la caméra comportant 31 miroirs sphériques proposée dans [67]. Ce type de modèle est généralement employé lors de la mesure d'un grand ensemble de rayons lumineux. Ils sont très utiles pour l'IBR (Image Based Rendering) qui permet de synthétiser de nouvelles images à partir d'images de la scène plutôt qu'à partir d'un modèle 3D. Les auteurs utilisent un modèle permettant de prendre en compte les phénomènes de réfraction à l'intérieur du plastique recouvrant la surface réfléchissante du miroir.

Dans [87] Ho et al. proposent un modèle spécifique pour les caméras équipées d'optique fisheye.

1.2.4.1 Calcul du rayon de projection associé à un point dans l'image

Dans le cas des caméras catadioptriques, l'obtention du rayon de projection associé à une position dans l'image est généralement assez aisée, en utilisant les méthodes de *lancer de rayons* (Raytracing) [88] empruntées au domaine de la synthèse d'images.

Un premier rayon est émis depuis la caméra (perspective ou télécentrique), et son intersection avec le miroir est calculée. La normale à la surface au point d'intersection est ensuite utilisée pour calculer le rayon réfléchi, qui est le rayon de projection pour la caméra catadioptrique.

1.2.4.2 Calcul de l'image d'un point de la scène

Avec un modèle non central, s'il est possible de modéliser plus précisément une caméra, il est par contre assez difficile de calculer l'image d'un point, son rayon de projection n'étant pas contraint de passer par un point particulier en l'absence d'hypothèse simplificatrice.

La méthode généralement utilisée, notamment par Mikusic et al. [52, 89], Lhuillier [90] et Goncalves et Araujo [91], pour obtenir la projection d'un point P par une caméra non centrale opère en deux temps. Une solution initiale approchée p_0 est tout d'abord estimée avec un modèle central approximant la caméra. Un rayon de projection r_0 est alors calculé à partir de cette solution en utilisant un modèle non central. La position de la projection p_i est alors affinée au cours d'itérations i de manière à minimiser la distance entre le rayon de projection r_i correspondant et le point P . Cette méthode n'est pas optimale, de par le fait qu'elle nécessite

un modèle central approchant assez bien le modèle non central. De plus, il est nécessaire d'utiliser des méthodes d'optimisation non linéaires itératives pour affiner la projection, ce qui peut nécessiter un temps important et risque de poser des problèmes de divergence ou d'existence de minima locaux.

✓ *Apport de la thèse présenté au chapitre 4* \implies Une méthode permettant la détermination directe de l'image d'un point avec une caméra catadioptrique non centrale lorsque le miroir utilisé est une quadrique.

1.2.4.3 Des simplifications de modèle

Dans certain cas, il est possible de faire des hypothèses simplificatrices sur les modèles de caméras non centrales, par exemple un alignement parfait de l'axe optique avec l'axe de révolution du miroir pour une caméra paracatadioptrique. Dans ce cas, le centre optique n'est pas défini comme étant un seul point, mais un ensemble situé sur cet axe. Il en résulte que ces modèles sont plus facilement exploitables que les modèles non centraux généraux. La caméra catadioptrique avec miroir conique en est un bon exemple [92, 93].

Ramalingam et al. donnent le nom d'*axiales* aux caméras dont les rayons de projection s'intersectent le long d'un axe. Ils proposent dans [94] des modèles adaptés. De même, Aliaga décrit dans [95] un modèle pour caméra paracatadioptrique en relaxant l'hypothèse de centre de projection unique. Un modèle axial utilisant un polynôme de degré cinq est proposé.

La caméra catadioptrique stéréoscopique proposée par Fiala [6] (visible sur la figure 1.9), comportant un double miroir sphérique utilise un autre exemple de simplification de modèle. En supposant les centres des sphères alignés sur l'axe optique de la caméra, il est possible d'établir un modèle de projection simplifié dans lequel les deux images d'un même point de la scène (un reflet pour chaque miroir) se projettent sur un même rayon dans l'image. Le même principe est utilisé par Jang et al. dans [96] avec l'emploi de caméras centrales. Dans [97, 98], Shih-Schon et Bajcsy décrivent une caméra employant le principe inverse. Deux caméras perspectives observent un unique miroir conique. L'alignement supposé parfait des deux caméras et du miroir permet de simplifier le modèle. Malheureusement, la recherche des homologues avec les caméras non centrales est en général (lorsque l'on ne fait aucune hypothèse) plus complexe.

1.2.4.4 Méthode pour la stéréoscopie dans le cas général

Avec des modèles de caméras non centrales, il n'est généralement pas adéquat de parler de géométrie épipolaire, les épipoles étant les images des centres optiques des caméras projetés sur les images d'autres caméras. A cause de l'absence de centres optiques, le terme choisi ici est *courbe de correspondance stéréoscopique*. Contrairement aux courbes épipolaires qui possèdent des propriétés de réciprocity (un point situé sur une courbe épipolaire dans une image se retrouve sur la courbe correspondante dans l'autre image et vice et versa), ce n'est pas le cas pour les caméras non centrales en général.

Fabrizio propose dans [21, 99] une méthode permettant d'obtenir numériquement ce genre

de courbes sous la forme d'une liste de pixels candidats. Cette méthode est basée sur une mesure de distance spatiale entre les rayons de projection de différentes caméras. La recherche des pixels correspondant à une distance minimale permet l'établissement de la courbe dans l'image. L'intérêt de cette méthode est qu'elle est totalement générale, mais son inconvénient est sa lenteur liée au calcul des distances entre les rayons de projection de chaque pixel.

✓ *Apport de la thèse présenté au chapitre 4* \implies Une méthode permettant une détermination efficace de l'image d'une (ou portion de) droite avec une caméra catadioptrique non centrale lorsque le miroir utilisé est une quadrique. Grâce à cette méthode, il est possible de projeter un rayon de projection et donc d'obtenir des courbes de correspondances stéréoscopiques.

1.2.4.5 Mise en garde concernant les hypothèses

Lors de la modélisation d'une caméra, il est nécessaire de bien prendre en compte tous les phénomènes optiques ou même mécaniques qui peuvent avoir une influence sur la formation des images. Ainsi par exemple, si l'on souhaite modéliser l'alignement entre la caméra et le miroir d'une caméra catadioptrique, il est nécessaire de faire intervenir une transformation rigide entre les repères liés à chacun de ces deux objets. Ces paramètres seront considérés comme intrinsèques si la caméra et le miroir sont fixés l'un à l'autre sans possibilité de mouvement. Il faudra néanmoins faire attention lors de l'utilisation d'une caméra équipée par exemple d'un zoom ou d'un système de mise au point. Un changement des réglages peut en effet entraîner un déplacement effectif du centre optique de la caméra, ce qui modifie l'alignement avec le miroir.

1.2.5 Un modèle de caméra radiale

Un modèle de caméra radiale 1D est intéressant pour les caméras possédant un axe de révolution (catadioptrique avec miroir et optique bien alignés, fisheye, perspective).

Thirthala et Pollefeys ont proposé son utilisation dans [100, 101]. De même, Tardif et al, dans [102, 103], étudient un tel modèle et proposent des méthodes pour l'étalonnage.

Avec un tel modèle, les mesures effectuées dans l'image apportent une information d'appartenance à un plan plutôt qu'à un rayon de projection, la distance du point au centre de l'image n'étant pas prise en compte. Tous les plans passent par le centre de l'image. L'intérêt de ce modèle est qu'il permet de s'affranchir des éventuelles distorsions ou du fait que la caméra soit axiale (et donc non centrale). Par contre, comme chaque image apporte moins d'information, un minimum de trois vues dans des configurations non dégénérées est nécessaire pour effectuer une reconstruction 3D au lieu de deux avec un modèle non radial.

1.2.6 Un modèle général pour une ou plusieurs caméras quelconques

Sturm et al proposent dans [104, 105] un modèle non paramétré de caméra, adapté à tous les types de caméras, y compris les non centrales et même les ensembles de plusieurs caméras. Ce modèle permet de définir un rayon de projection associé à chaque pixel de l'image. En

jouant sur les degrés de liberté de ces rayons, il est possible de modéliser les différents types de caméras.

1.3 Le problème de l'étalonnage géométrique

L'étalonnage permet d'ajuster le modèle à la caméra qui est étalonnée en estimant ses paramètres. Ceci peut être effectué grâce à l'observation de points ou d'autres primitives de la scène dans une ou plusieurs images.

Lorsque les motifs observés pour l'étalonnage sont connus et créés pour cette tâche, ils constituent une *mire* et portent le nom d'*amers*. Ceux-ci peuvent être de différents types.

Les motifs utilisés pour l'étalonnage peuvent aussi être issus d'une scène quelconque, par exemple sous la forme de points d'intérêt ou de droites, qui, une fois appariées dans différentes images, fournissent l'information nécessaire à l'étalonnage.

Lors de l'étalonnage d'un capteur, si l'on souhaite obtenir une précision homogène sur toute sa surface, il est nécessaire d'avoir des observations bien réparties sur celle-ci. Ceci peut poser des problèmes avec les caméras à large champ de vision, car les motifs observés doivent entourer la caméra pour se projeter en différents endroits. Ceci est bien évidemment impossible à effectuer avec une mire plane, pourtant couramment utilisée pour étalonner les caméras standard, comme proposé par Zhang dans [106]. Le problème peut être résolu ici en acquérant plusieurs images de la mire dans des configurations différentes de manière à ce qu'elle se projette sur toute la surface du capteur.

Il existe un très grand nombre de méthodes pour les différents modèles de caméras permettant l'étalonnage, et utilisant différents types d'observations.

1.3.1 Étalonnage avec un modèle paramétrique

Dans le cas des modèles paramétriques, l'étape d'étalonnage consiste à estimer les paramètres du modèle à partir d'observations. Le jeu de paramètres obtenu doit permettre de minimiser un certain critère lié à une fonction de coût qui mesure si les observations correspondent au modèle estimé.

1.3.2 Étalonnage avec un modèle non paramétrique

La méthode de Sturm [104] déjà évoquée est basée sur l'observation d'une mire d'étalonnage dans différentes configurations inconnues a priori. La mise en correspondance des différentes observations (au moins 2 pour les caméras centrales et 3 pour les caméras non centrales) permet l'étalonnage grâce à une recherche des mouvements qui amène les observations de mêmes points sur des droites communes. Ces mouvements sont estimés grâce à un tenseur. En partant d'une solution initiale proche obtenue par l'observation d'une seule image de la mire et en appliquant un ajustement de faisceaux, il est possible de raffiner les rayons de projection associés à chaque position pixellique où il y a eu observation ainsi que les différentes poses de la mire (exceptée la première qui est fixée).

On pourra noter que cette méthode est tellement générale qu'elle permet l'étalonnage d'un ensemble de caméras comme s'il s'agissait d'une caméra unique. Par contre, elle ne permet pas a priori la projection directe sur l'image d'un point 3D ainsi que l'obtention d'un rayon de projection associé à un pixel en précision subpixelique, à moins de procéder à une interpolation des rayons de projection entre les pixels.

1.3.3 Différents types d'observations

Différentes primitives peuvent être utilisées pour l'étalonnage. Les plus couramment utilisées sont énoncées ici. Chacune possède des avantages et des inconvénients et il est donc nécessaire de faire un choix.

Par exemple, l'article [107] de Huang et al présente une étude comparative pour les caméras panoramiques à base de CCD linéaires en rotation. Cette étude compare trois approches, une basée point 3D, une autre basée sur la mise en correspondance d'images de points et la dernière sur l'analyse d'images de droites parallèles.

1.3.3.1 Des observations fournissant une position ponctuelle

Les points sont la primitive la plus naturelle à utiliser pour l'étalonnage. L'un des avantages des primitives qui fournissent la position d'un point est qu'elles permettent une mesure objective de l'erreur après étalonnage, sous forme d'une erreur de projection entre les positions observées et les positions prédites par le modèle. Ceci est moins clair par exemple lorsque les primitives sont des droites.

Par contre, ces méthodes requièrent la mise en correspondance des observations, soit avec d'autres observations, soit avec un modèle 2D ou 3D connu, ce qui n'est pas nécessaire pour certains autres types de primitives.

Des points à des positions connues : La détection de la projection des points dans l'image peut être très précise, grâce à l'emploi de méthodes subpixeliques. Par exemple, lorsque l'amer est un point défini par l'intersection d'un damier, il est possible de détecter sa projection en ajustant un parabolioïde hyperbolique dans l'image.

Le point détecté est ensuite mis en correspondance avec une position 2D ou 3D de la mire, et les paramètres du modèle sont estimés pour minimiser la somme des carrés des erreurs de reprojexion.

✓ *Apport de la thèse présenté au chapitre 5* \implies Utilisation d'une mire 3D spécifique constituée de points pour la caméra paracatadioptrique.

✓ *Apport de la thèse présenté au chapitre 4* \implies Une méthode d'étalonnage pour les caméras catadioptriques non centrales à base de miroir quadrique.

Des motifs auto-identifiants : Fiala et al. ont proposé dans [108] des motifs d'étalonnage auto-identifiant, nommés *Artag*. Ces motifs ressemblent aux Datamatrix, des codes barres bi-dimensionnels. Ils peuvent être détectés et identifiés automatiquement, ce qui évite d'avoir à chercher à quels points 3D ils correspondent. Chacun permet d'obtenir soit un point central soit quatre points situés au niveau des coins du motif. Ils sont employés pour les caméras perspectives standard car il y a homographie entre le motif plan observé et son image. Ils ne sont malheureusement pas directement utilisables avec les caméras grand champ, pour lesquelles cette dernière propriété n'est pas vérifiée.

Des disques : Les disques sont utilisés pour les caméras perspectives, avec lesquelles ils se projettent en ellipses (dans le cas où la caméra réalise une pure projection perspective). Dans le cas des caméras grand angle, cette propriété pourra être utilisée uniquement si la projection est localement proche d'une projection perspective. Ceci n'est pas étudié dans la suite.

Des points à des positions inconnues : Il est possible d'étalonner une caméra sans utiliser de mire. On parle alors d'autocalibrage. Ces méthodes utilisent généralement des contraintes géométriques (épipolaires) sur l'observation depuis différentes positions de points fixes de la scène. Ces contraintes se répercutent sur le rang d'une *matrice fondamentale* [77] qui contraint les positions respectives des projections de ces points. Les paramètres de la caméra peuvent ensuite être extraits de cette matrice fondamentale.

Kang décrit dans [78] une méthode adaptée à une caméra paracatadioptrique considérée centrale et décrivant une rotation. Le suivi de points caractéristiques, dans une séquence d'images pendant la rotation permet, grâce aux contraintes de la géométrie épipolaire, d'estimer les paramètres.

Geyer et al. décrivent dans [109] les propriétés de la matrice fondamentale permettant la reconstruction 3D à partir de deux images obtenues avec une caméra paracatadioptrique centrale telle que décrite dans [110]. Sturm généralise cette étude dans [111] pour des ensembles de trois ou quatre caméras centrales de différents types (paracatadioptriques et/ou perspectives).

Dans [52], Micusik et al. proposent une extension de l'algorithme des huit points [77] utilisé pour les caméras standard. A l'aide de neuf correspondances de points, ils montrent qu'il est possible d'obtenir une reconstruction euclidienne de la scène dans le cas où les paramètres de la caméra sont fixes, le modèle de la caméra étant adaptable à des caméras non centrales lors d'une étape de raffinement de la reconstruction.

1.3.3.2 Des observations de droites

Les droites sont particulièrement adaptées pour l'étalonnage des caméras grand angle. Elles peuvent être utilisées soit par groupes, formant alors des points de fuite, soit individuellement. Dans le cas de certaines caméras catadioptriques, elles contiennent l'information qui permet d'estimer tous les paramètres du modèle, contrairement aux caméras standard comme le notent Geyer et al. dans [80]. En effet, dans le cas des caméras standard, chaque droite fournit deux informations mais ajoute deux inconnues, alors que dans le cas des caméras catadioptriques

considérées, ce sont trois informations qui sont fournies par chaque observation de droite. Ainsi, Geyer et al. proposent dans [82, 112] des méthodes pour l'étalonnage de la caméra paracatadioptrique. Barreto et al. font de même dans [113]. Ying et al. l'étendent dans [114] pour d'autres caméras catadioptriques.

✓ *Apport de la thèse présenté au chapitre 3* \implies Une méthode basée sur celle de Geyer et al. [82] est proposée pour l'étalonnage de la caméra paracatadioptrique à partir de droites.

On pourra finalement noter la possibilité d'estimer les distorsions (radiales par exemple) grâce à l'observation de droites avec une caméra standard, comme le proposent par exemple Devernay et Faugeras dans [115]. La fonction de correction est telle que les images de droites corrigées doivent être des segments de droites (pour respecter la projection perspective).

1.3.3.3 Des observations de sphères

L'intérêt de la sphère est que sa projection dans l'image définit un contour qui peut être précisément détecté. Pour une caméra centrale, ce contour est l'image d'un petit cercle à la surface de la sphère observée, définissant un cône avec le centre optique de la caméra. Dans le cas des caméras perspectives et de certaines caméras catadioptriques, ce contour a une forme de conique fermée.

Zhang et al. utilisent cette propriété dans [116] pour étalonner des caméras standard. Ying et al. l'utilisent dans [117, 114] pour des caméras catadioptriques centrales autres que paracatadioptriques. Hormis pour cette dernière, l'observation de la conique fermée dans l'image permet un étalonnage plus précis qu'avec des droites. Ceci est dû au fait que les droites se projettent en portions de coniques, qu'il est plus difficile de détecter précisément qu'une conique fermée correspondant à la projection d'une sphère.

1.3.3.4 L'observation du contour du miroir

Dans le cas des capteurs catadioptriques, il est parfois possible d'utiliser le miroir lui-même pour étalonner la caméra. Fabrizio propose dans [21] une méthode pour les caméras catadioptriques avec caméra perspective uniquement. Cette méthode est basée sur l'estimation des homographies entre le plan image et deux plans de référence sur lesquels figurent deux cercles (le contour du miroir et le contour d'une aiguille conique). Il propose également d'intégrer plusieurs images pour aider à la détection des cercles. Finalement, il propose des caméras catadioptriques dotées de motifs sur ces plans pour aider à l'étalonnage.

Kang compare dans [78] une approche basée sur la détection du contour du miroir avec une autre utilisant un suivi de points dans une séquence pour une caméra paracatadioptrique. Il en conclut que la méthode utilisant l'image du contour du miroir est moins précise. Geyer compare avec une approche utilisant des images de droites dans [82], pour la même caméra, et parvient aux mêmes conclusions.

Mashita et al. proposent dans [118] une méthode basée sur l'analyse des contours de miroirs

pour estimer les erreurs d’alignement entre le miroir et la caméra perspective.

1.4 Reprojection d’images

L’étude des modèles de projection de caméra permet de générer des images correspondant à un modèle à partir d’une ou plusieurs images correspondant à un autre modèle. Ceci a par exemple déjà été montré lors du mosaïquage d’images sur la figure 1.2 où une image en projection sphérique est générée à partir de plusieurs images en projection perspective plane.

1.4.1 Reprojection à partir de modèles centraux

Lorsque les deux modèles de caméra sont centraux, il est possible de générer des *reprojections* parfaites des images. Ceci permet par exemple d’obtenir une image en projection cylindrique à partir d’une image panoramique d’une caméra paracatadioptrique centrale comme le montre l’exemple de la figure 1.15. Un cylindre est échantillonné en hauteur et en azimut pour des positions pixelliques respectivement en lignes et colonnes. Les coordonnées 3D des points du cylindre sont alors projetées dans l’image d’origine en utilisant le modèle du capteur et la couleur des pixels correspondant est déterminée par simple recopie des valeurs depuis l’image d’origine.



FIG. 1.15 – (a) Une image d’une caméra paracatadioptrique. (b) La reprojection de cette image sur un cylindre, représenté ici sur un plan en coordonnées cylindriques.

✓ *Apport de la thèse présenté au chapitre 3* \implies Une méthode utilisant la projection centrale pour obtenir une image invariante aux rotations de la caméra paracatadioptrique avec laquelle elle a été acquise.

✓ *Apport de la thèse présenté en Annexe B* \implies Une méthode générique exploitant la re-

projection pour la synthèse d'images. Cette méthode permet de générer des images par une projection centrale quelconque en utilisant les fonctionnalités des cartes accélératrices 3D.

1.4.2 Reprojection à partir de modèles non centraux

Lorsque l'un ou les deux modèles de caméra sont non centraux, il n'est pas possible a priori d'effectuer de reprojection parfaite. Il est alors nécessaire d'opérer une reprojection approchée.

Par exemple, en approximant une caméra fisheye par un modèle central, Ying et Hu proposent dans [81], de reprojeter son image vers l'image d'une caméra catadioptrique et vice et versa. De même, dans [119], Shigang et al. reprojettent sur une sphère deux de ces images. Ho et al. présentent dans [87] une méthode pour générer des images en projection presque perspective à partir d'images obtenues avec une caméra fisheye.

1.4.3 Reprojection de textures

Lorsque la géométrie de la scène observée est connue, il est possible de reconstruire sa texture. Ceci est généralement effectué en calculant des images planes à projeter sur des facettes du modèle 3D de la scène, comme montré dans le [chapitre 5](#) lors de la reconstruction 3D d'une scène. Ces images sont calculées par un échantillonnage spatial de la surface dont la texture est recherchée et par l'attribution d'une couleur à chacune des positions correspondantes.

✓ *Apport de la thèse présenté au [chapitre 4](#)* \implies La méthode proposée pour obtenir l'image d'un point avec une caméra catadioptrique non centrale avec miroir en forme de quadrique est utilisée pour calculer les textures de faces de la scène.

Chapitre 2

Conception et réalisation d'une caméra utilisant un capteur d'orientation pour le calcul de mosaïques d'images orientées

Dans ce chapitre, un prototype de capteur électronique d'orientation est présenté, en 2.2. Ce capteur permet de mesurer une orientation absolue sans contact mécanique grâce à des capteurs magnétiques et des accéléromètres. Des méthodes d'étalonnage des différents sous capteurs qui le constituent sont proposées. Ensuite, une caméra hybride, la ORIENT-CAM, équipée de ce capteur, est détaillée, en 2.3. Une méthode permettant l'étalonnage de l'orientation relative de la caméra par rapport au capteur d'orientation est proposée. Finalement, la caméra est utilisée pour générer des images omnidirectionnelles orientées par mosaïquage d'images, en 2.4. L'orientation estimée de la caméra est alors affinée grâce à l'analyse d'image et à l'application de l'algorithme d'ajustement de faisceau, présenté en 2.5. Finalement des perspectives sont proposées pour améliorer le capteur.

2.1 Motivations

Dans ce chapitre, nous nous intéressons au développement d'un capteur électronique capable de mesurer sa propre orientation dans l'espace. Cette étude, si elle sort du domaine de la vision, a permis la réalisation d'une caméra hybride, la ORIENT-CAM, dont les applications en vision sont évidentes. Le capteur électronique proposé, une fois attaché à une caméra, permet la détermination de son orientation de manière absolue. Celle-ci est définie localement dans un repère orthonormal de référence ω_r dans lequel deux axes correspondent à la verticale et à la direction du nord projeté dans le plan horizontal.

La connaissance de l'orientation de la caméra permet le mosaïquage d'images tel que présenté au chapitre 1. Dans ce chapitre, une méthode de mosaïquage tenant compte de l'informa-

tion fournie par le capteur d'orientation est proposée.

2.1.1 Point de départ

L'emploi d'un capteur d'orientation permet d'aider l'estimation de pose d'une caméra et donc facilite la localisation par l'image. Des premiers tests ont été effectués en attachant à une caméra un accéléromètre deux axes réalisé à l'IRIT par Jean Conter et appelé musaraigne. Avec ce capteur, il est possible de détecter l'inclinaison de la caméra et de projeter son image sur une caméra virtuelle qui serait tenue avec une certaine orientation. Ceci permet de générer des images invariantes aux rotations de la caméra sur les axes de roulis et de tangage. Ainsi, l'image de la figure 2.1 (a) est rectifiée grâce à l'orientation fournie par les accéléromètres ce qui permet de générer l'image de la figure 2.1 (b) grâce à une simple homographie.



FIG. 2.1 – (a) Une image acquise par la caméra. (b) La même image après rectification en utilisant l'orientation fournie par les accéléromètres.

Il a alors été décidé de pousser plus loin l'étude, en développant un nouveau capteur permettant d'obtenir l'orientation complète de la caméra, ramenant le problème d'estimation de la pose de la caméra aux seuls trois paramètres de position.

Une fois ce capteur réalisé, plusieurs autres applications ont été trouvées, notamment le mosaïquage d'images, qui permet une augmentation du champ de vision de la caméra, très utile pour la localisation par l'image.

2.1.2 Des capteurs déjà utilisés pour la vision

Certains appareils photos du commerce contiennent des capteurs d'orientation d'un type différent de celui proposé ici. En effet, ceux-ci permettent généralement de déterminer uniquement si l'image acquise est dans l'orientation portrait ou paysage, afin d'effectuer une rotation à 90° de l'image si nécessaire.

L'utilisation de capteurs plus complets à déjà été proposée. Par exemple, dans [120], Lobo et al. décrivent un système combinant une caméra et des capteurs inertiels pour détecter une référence verticale et en déduire une orientation partielle, la rotation autour de l'axe vertical

ne pouvant être détectée. Dans [121], Schon et Gustafsson proposent un système plus complet, équipé de gyroscopes et fournissant une orientation relative autour des trois axes.

2.1.3 Intérêt d'un nouveau capteur

Dans le système proposé par Schon et Gustafsson, les gyroscopes permettent la mesure des variations d'orientation autour des trois axes. Afin d'obtenir une valeur angulaire de l'orientation, il est nécessaire d'intégrer ces variations autour d'une orientation de référence. Il faut donc étalonner le capteur avant chaque utilisation, et ses mesures dérivent dans le temps, au fur et à mesure que les erreurs s'accumulent.

Un capteur permettant d'obtenir une orientation absolue est souhaitable afin de permettre une utilisation sans étalonnage fréquent. La disponibilité de circuits iMEMS (integrated Micro Electro Mechanical Systems) tels que des accéléromètres et des magnétomètres permet le développement d'un tel capteur.

Malheureusement ces derniers sont sensibles à des perturbations pouvant nuire à la détection de l'orientation. Ainsi, la présence d'un champ magnétique parasite ou d'une accélération liée à un déplacement du capteur empêchent l'obtention de l'orientation exacte. Dans ces conditions, la redondance, liée à l'emploi de plusieurs capteurs, peut être utilisée pour détecter les cas où l'orientation fournie est erronée. C'est pour cette raison que sept mesures sont effectuées par le capteur pour déterminer la rotation qui possède seulement trois degrés de liberté.

2.2 Le capteur d'orientation

2.2.1 Principe

Le capteur développé est constitué d'accéléromètres et de magnétomètres. Il est basé sur une boussole numérique proposée dans une note d'application de la société Honeywell [122] mais présente de nombreuses améliorations.

Les lignes de champ magnétique terrestre suivent la trajectoire représentée sur la figure 2.2. Ces lignes de champ, projetées dans un plan tangent à la sphère terrestre (horizontal), pointent dans la direction du nord magnétique. Celui-ci est en mouvement, à une vitesse approximative de 40km/an. Il est actuellement situé à 1900km du pôle nord géographique. La différence angulaire entre les pôles magnétiques et géographiques est appelée déclinaison magnétique, et sa valeur dépend du lieu considéré.

Les boussoles numériques peuvent être classées en deux catégories, selon qu'elles disposent ou non d'une correction sur les axes de tangage et roulis. Les plus simples disposent de deux capteurs magnétiques orthogonaux situés dans le plan du capteur qui est supposé horizontal. La projection des lignes de champ dans le plan horizontal est donc obtenue de facto. La direction du nord magnétique est alors calculée grâce à la tangente entre les mesures des deux capteurs. Un capteur de ce type, le CMP03, est utilisé dans le capteur 3D présenté au chapitre 5. Ce genre de boussole ne permet pas la mesure lorsque le capteur n'est pas tenu horizontalement.

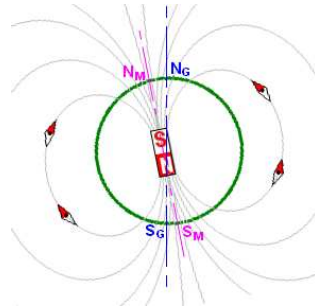


FIG. 2.2 – Les lignes de champ magnétique et l'écart entre le nord magnétique et le nord géographique.

Des boussoles équipées de corrections sont utilisées dans ces cas là. La solution proposée par Honeywell [122] consiste à utiliser deux accéléromètres pour mesurer l'inclinaison du plan du capteur. En effet, en l'absence d'accélération liée à un mouvement, la seule accélération mesurée est celle due à la gravité. Cette mesure est proportionnelle au sinus de l'angle entre la verticale (direction de la gravité) et l'axe de l'accéléromètre considéré. Grâce à la mesure sur deux axes, il est possible de détecter l'orientation du plan du capteur par rapport à l'horizontale. Le champ magnétique est alors mesuré en trois dimensions, grâce à trois magnétomètres montés orthogonalement. Le vecteur 3D correspondant à la mesure dans le repère du capteur ω_c (visible sur la figure 2.4 (b)) est ensuite ramené dans le plan horizontal estimé grâce aux mesures des accéléromètres puis ses deux composantes dans ce plan sont utilisées pour déterminer la direction du nord magnétique comme dans le cas simple.

Le capteur développé reprend ce principe mais en ajoutant deux accéléromètres, de manière à obtenir une mesure de l'accélération sur trois axes orthogonaux, afin de permettre une mesure précise de l'inclinaison dans toutes les configurations. De plus, l'orientation est calculée en tenant compte de toutes les informations fournies par les différents capteurs au lieu de n'exploiter que la projection du champ magnétique dans le plan horizontal. Ceci permet d'obtenir une mesure plus précise et de pouvoir détecter les parasites qui faussent la mesure.

2.2.2 Description du capteur

La figure 2.3 montre le typon double couche et les principaux composants du capteur. Le prototype de capteur d'orientation est visible sur la figure 2.4. Ses dimensions sont 45*60*10mm. Elles pourraient être facilement réduites par l'emploi de méthode de fabrication industrielle, telle qu'un circuit imprimé multi-couches.

Les capteurs magnétiques HONEYWELL HMC1051Z (1 axe) et HMC1052 (2 axes) sont utilisés pour obtenir une mesure 3D du champ magnétique (sur les axes a_1, a_2 et a_3 montrés sur la figure 2.4). Ces circuits comportent des ponts de Wheatstone dont la résistance varie proportionnellement au champ magnétique présent sur l'axe correspondant, entraînant une variation de la tension à ses bornes, dans une plage très réduite. Il est donc nécessaire d'amplifier plusieurs centaines de fois cette tension afin de pouvoir la mesurer précisément.

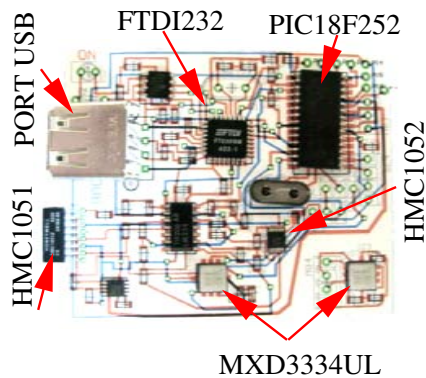


FIG. 2.3 – Le circuit électronique du capteur en version papier et les principaux composants.

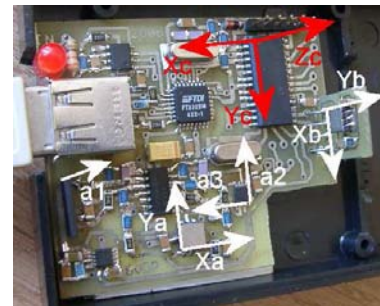


FIG. 2.4 – Le capteur d'orientation et ses différents axes de mesures. a_1 et Yb sont tous deux orthogonaux à la carte électronique principale. X_c, Y_c et Z_c sont les axes du repère du capteur ω_c .

Deux accéléromètres deux axes MEMSIC MXD3334UL montés orthogonalement sont utilisés pour mesurer le champ gravitationnel. Chaque MXD3334UL inclut deux accéléromètres eux mêmes montés orthogonalement dans le circuit. L'usage d'un unique circuit MXD3334UL mènerait à des faibles précisions lorsque l'un de ses deux axes se rapproche de la verticale. Ceci est dû au fait que les mesures effectuées par les accéléromètres correspondent à la projection du champ gravitationnel. Quand l'angle entre l'axe de mesure et la verticale est proche de 90° , un changement d'orientation de quelques degrés ne peut pas être détecté précisément car son sinus varie très peu.

Utiliser au moins trois capteurs montés orthogonalement permet de s'assurer que deux axes sont toujours suffisamment éloignés de la verticale et assure donc plus de précision dans les mesures. Finalement, l'accélération est mesurée sur les axes X_a, Y_a, X_b et Y_b montrés sur la figure 2.4.

La sortie de ces accéléromètres est de type digitale PWM (modulée en largeur d'impulsion). Un rapport cyclique de 50% est obtenu sur une voie lorsque la mesure effectuée correspond à une accélération nulle. Celui-ci varie proportionnellement de $\pm 20\%$ pour une mesure de $\pm 1G$.

Un microcontrôleur Microchip PIC 18F252 reçoit les mesures des différents capteurs et les envoie à un ordinateur via liaison USB grâce à un circuit FTDI FT232BM. L'ensemble des informations des différents capteurs est envoyé à 100Hz, ce qui permet une estimation de l'orientation en temps réel dans une application de vision par ordinateur.

Il est intéressant de noter que les capteurs fournissent des informations redondantes. Seulement trois paramètres doivent être déterminés (la rotation) et les mesures provenant des sept capteurs sont disponibles. Ceci permet de calculer une orientation plus précise et de détecter des incohérences telles que des accélérations parasites dues à des mouvements du capteur ou des perturbations du champ magnétique. Une mesure de confiance dans l'orientation retrouvée est obtenue en vérifiant la cohérence des mesures des différents capteurs avec l'orientation calculée.

2.2.3 Étalonnage du capteur

Les données fournies par les différents capteurs doivent être converties pour être utilisées pour déterminer l'orientation. La transformation adéquate est obtenue grâce à une étape d'étalonnage réalisée en trois temps. Tout d'abord, les accéléromètres sont étalonnés, le capteur étant tenu dans des positions statiques. Ensuite, les magnétomètres sont étalonnés, alors que le capteur est mis en rotation de manière à obtenir le plus de mesures différentes possibles. Finalement, la direction du champ magnétique terrestre à la position d'utilisation du capteur est acquise.

Dans la suite, les différents composants sont considérés parfaitement positionnés sur le circuit imprimé. En pratique ce n'est évidemment pas le cas et il faudrait idéalement estimer les erreurs d'alignement entre les différents capteurs. Néanmoins, ceci n'est pas facile à réaliser et l'hypothèse simplificatrice permet d'obtenir des résultats satisfaisants pour notre application.

La figure 2.5 montre 3661 échantillons sur les sept différents canaux. Les 280 premiers correspondent à 14 positions statiques contenant chacune 20 échantillons et servent à l'étalonnage des accéléromètres. Tous les échantillons sont ensuite utilisés pour l'étalonnage des magnétomètres.

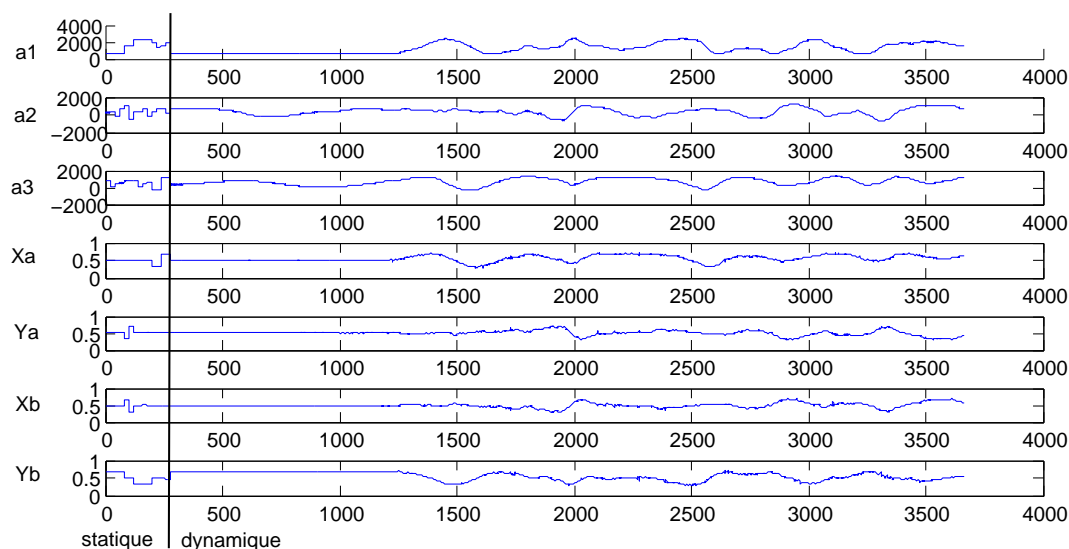


FIG. 2.5 – Données brutes des différents capteurs pendant les deux phases de l'étalonnage, statique et dynamique.

Étalonnage des accéléromètres : Les accéléromètres doivent être étalonnés en position statique pour éviter les accélérations parasites. Le capteur d'orientation est maintenu dans différentes orientations connues et plusieurs mesures sont effectuées sur chaque voie pour calculer des valeurs moyennes correspondant à des orientations extrêmes, verticales et horizontales. Une valeur d'accélération exprimée en unité g est ensuite calculée.

La figure 2.6 (a) montre les données traitées des différentes voies des accéléromètres après correction. Les bornes correspondant à $\pm 1g$ sont représentées en pointillés. L'étalonnage permet d'obtenir des valeurs mesurées x_a , y_a , x_b et y_b comprises entre $\pm 1g$ lorsque le capteur est tenu verticalement et de $0g$ lorsqu'il est tenu horizontalement. Dans le cas où les capteurs fournissent des mesures sortant de ces bornes, les valeurs sont écrêtées.

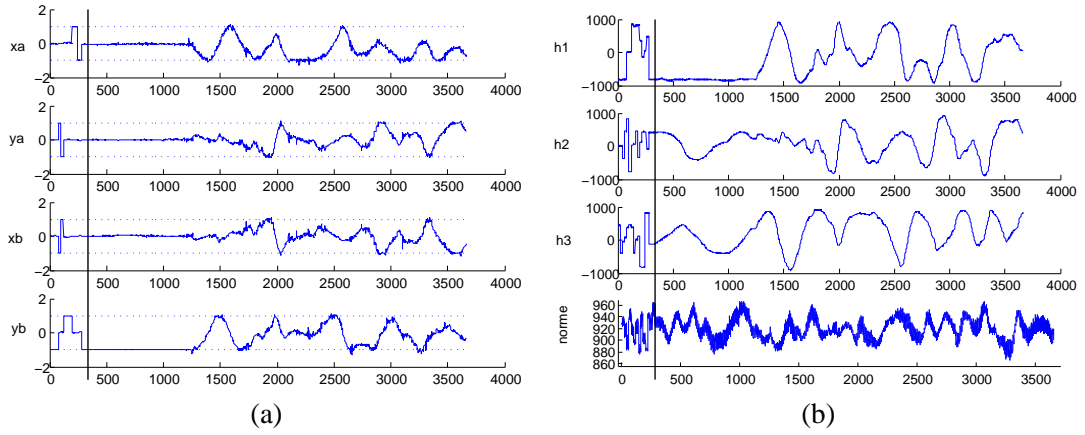


FIG. 2.6 – Données traitées des différents voies des (a) accéléromètres et (b) magnétomètres. Pour les magnétomètres, la dernière courbe correspond à la variation de la norme du champ magnétique mesuré après correction.

Étalonnage des magnétomètres : Chaque magnétomètre fournit une sortie analogique qui est amplifiée puis numérisée par les convertisseurs analogiques numériques 10 bits du microcontrôleur. Un offset et un gain doivent être appliqués à chaque voie pour assurer que le champ magnétique tridimensionnel mesuré ait une norme la plus constante possible quand le capteur est tourné dans différentes directions. Ces paramètres sont estimés après acquisition d'une séquence de mesures alors que le capteur est mis en rotation dans un environnement magnétique *propre* dans lequel les perturbations doivent être négligeables par rapport au champ magnétique terrestre.

Soit f_1 , f_2 et f_3 (resp. o_1 , o_2 et o_3), les facteurs (resp. offsets) à appliquer à chacune des voies. La correction C d'un vecteur $(a_1 a_2 a_3)^T$ correspondant à une mesure du champs magnétique est donnée par :

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = C \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} k f_1 a_1 + o_1 \\ k f_2 a_2 + o_2 \\ k f_3 a_3 + o_3 \end{pmatrix}; k = \frac{1}{\sqrt{f_1^2 + f_2^2 + f_3^2}} \quad (2.1)$$

Une fonction de coût, dont la valeur est minimale quand la variance de la norme corrigée pour les différents échantillons est la plus faible, est alors minimisée pour obtenir le meilleur

jeu de paramètres $[f_1 f_2 f_3 o_1 o_2 o_3]^T$:

$$(f_1 f_2 \widehat{f_3} o_1 o_2 o_3) = \arg \min_{(f_1 f_2 f_3 o_1 o_2 o_3)} \text{variance} \left(C \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right) \quad (2.2)$$

Le paramètre de normalisation k permet d'éviter que les paramètres de gain ne convergent vers des valeurs nulles, ce qui minimiserait certes la variance mais ne permettrait pas l'obtention d'une mesure de champ magnétique utilisable. Dans un premier temps, la minimisation était effectuée avec le paramètre f_1 fixé à un, mais la méthode proposée a permis une meilleure convergence dans les expériences réalisées.

La figure 2.6 (b) montre les données traitées des différentes voies des magnétomètres après correction. La dernière ligne représente la norme du champ magnétique après correction, dont la variance a été minimisée. Entre les échantillons 280 et 1260, le capteur a été tourné dans le plan horizontal et la valeur mesurée sur la voie a_1 est donc à peu près constante.

Les valeurs mesurées sur les différentes voies des magnétomètres $(h_1 h_2 h_3)^T$ sont normalisées :

$$L = (l_1 l_2 l_3)^T = N((h_1 h_2 h_3)^T) = \frac{(h_1 h_2 h_3)^T}{\|(h_1 h_2 h_3)^T\|} \quad (2.3)$$

Étalonnage du champ magnétique : L'étape finale d'étalonnage permet d'estimer la direction 3D du champ magnétique terrestre à la position d'utilisation du capteur. Le champ a une composante verticale qui dépend de la latitude comme visible sur la figure 2.2. L'angle entre le champ magnétique et le plan horizontal est estimé en tenant le capteur horizontalement et en mesurant l'angle directement. La déclinaison magnétique peut aussi être étalonnée grâce à la même opération. Il suffit pour cela de pointer le capteur dans la direction du nord géographique (en utilisant par exemple comme référence deux points facilement identifiables sur une carte.) Il est néanmoins plus simple d'utiliser une valeur de déclinaison fournie pour la position où le capteur est utilisé. Cette valeur de déclinaison pourra être considérée constante dans le voisinage du lieu d'utilisation car elle varie peu à l'intérieur des pays éloignés des pôles.

La mesure L_0 correspondant à un capteur tenu horizontalement et pointant vers le nord géographique (ω_c aligné avec ω_r) est acquise :

$$L_0 = (l_1 l_2 l_3)_0^T = N((h_1 h_2 h_3)_0^T) \quad (2.4)$$

2.2.4 Détermination de l'orientation

Une fois l'étape d'étalonnage achevée, une première estimation de l'orientation du capteur peut être effectuée en utilisant uniquement deux accéléromètres et la projection 2D du champ magnétique dans le plan horizontal retrouvé d'après les mesures des deux accéléromètres comme proposé dans [122]. L'orientation est ensuite affinée en prenant en compte les données des sept capteurs. L'orientation est paramétrée par un quaternion, ce qui permet d'éviter des problèmes de discontinuité et le gimbal-lock présents avec une paramétrisation par angles d'Euler.

Soit $Q = [abcd]^T$ le quaternion correspondant à l'orientation du capteur. La matrice de rotation R_Q (de ω_r vers ω_c) correspondante est définie par l'équation (2.5).

$$R_Q = \begin{bmatrix} 1 - 2b^2 - 2c^2 & 2ab - 2cd & 2ac + 2bd \\ 2ab + 2cd & 1 - 2a^2 - 2c^2 & 2bc - 2ad \\ 2ac - 2bd & 2bc + 2ad & 1 - 2a^2 - 2b^2 \end{bmatrix} \quad (2.5)$$

Soit L_R (resp. G_R), le vecteur correspondant au champ magnétique (resp. gravitationnel) après rotation par R_Q , avec $V_0 = (0 \ 0 \ 1)^T$ dans ω_r :

$$L_R = \begin{pmatrix} l_{1r} \\ l_{2r} \\ l_{3r} \end{pmatrix} = R_Q L_0; \quad G_R = \begin{pmatrix} g_{1r} \\ g_{2r} \\ g_{3r} \end{pmatrix} = R_Q V_0 \quad (2.6)$$

Le vecteur de résidus Δ_t correspondant à une mesure sur les sept capteurs à l'instant t est défini par l'équation (2.7). Les différents résidus sont considérés comme ayant la même variance, ce qui n'est pas le cas dans la réalité mais suffit pour l'application considérée. Cette hypothèse nécessite tout de même une mise à l'échelle des différents résidus pour qu'ils soient de même ordre de grandeur. Ceci est obtenu grâce au fait que les vecteurs L_0 sont normalisés, ce qui entraîne des résidus du même ordre que ceux liés au vecteur unitaire V_0 . Les voies Xb et Ya sont censées mesurer l'accélération sur le même axe, leur influence est donc pondérée par un facteur 0.5. La pose \widehat{Q}_{st} (statique à l'instant t) est estimée en minimisant la norme de Δ_t , pour un instant t considéré :

$$\Delta_t = \begin{pmatrix} a_1(t) - l_{1r} \\ a_2(t) - l_{2r} \\ a_3(t) - l_{3r} \\ x_a(t) - g_{1r} \\ 0.5 (y_a(t) + g_{2r}) \\ 0.5 (x_b(t) - g_{2r}) \\ y_b(t) - g_{3r} \end{pmatrix}; \quad \widehat{Q}_{st} = \arg \min_Q \|\Delta_t\|^2 \quad (2.7)$$

A cause du bruit de mesure, un filtrage temporel est appliqué pour améliorer la stabilité et la précision. Dans un premier temps, il a été envisagé d'utiliser un filtrage de Kalman comme proposé par Shin dans [123] et dont Welch et al. fournissent les détails dans [124]. Comme il n'y a pas de connaissance a priori du mouvement du capteur à un instant donné, il aurait fallu considérer le capteur comme étant immobile. Sous cette condition, le filtre de Kalman est équivalent à un estimateur de type moindres carrés standard qui lui a été préféré.

Soit w la taille, en nombre d'échantillons, d'une fenêtre de filtrage temporel permettant de choisir les mesures utilisées entre les instants $T - w$ et T pour calculer l'orientation à l'instant T . Les échantillons dans cette fenêtre sont pondérés différemment grâce au facteur ρ_t pour faire en sorte que les échantillons proches dans le temps soient plus influents. Une fonction de

pondération en forme de rampe remplit bien ce rôle et fournit des résultats satisfaisants. Augmenter (resp. diminuer) w rend l'orientation détectée plus stable (resp. réactive). Des valeurs de w entre 10 et 20 fournissent de bons résultats dans notre application. La pose \widehat{Q}_{dT} (dynamique à l'instant T) est estimée en minimisant le critère suivant, pour un instant T considéré :

$$\rho_t = \frac{t + w - T}{w}; \widehat{Q}_{dT} = \arg \min_Q \sum_{t=T-w}^T (\rho_t^2 \|\Delta_t\|^2) \quad (2.8)$$

Un démonstrateur a été développé afin de montrer le bon fonctionnement du capteur et plus particulièrement pouvoir régler divers paramètres et visualiser les résidus sur les différentes voies en fonction des différentes conditions d'utilisation. Une vue de synthèse de la carte électronique orientée est générée à une fréquence de 100 images par secondes en utilisant l'orientation détectée. La figure 2.7 montre des photographies de l'écran sur lequel l'image de synthèse est affichée en temps réel et du capteur placé devant l'écran dans différentes orientations. Dans les cas (g) et (h), l'orientation affichée est erronée, à cause d'une accélération ou d'un champ magnétique parasite. L'incohérence des mesures est détectée grâce à une somme des résidus trop élevée après minimisation de l'équation (2.8).

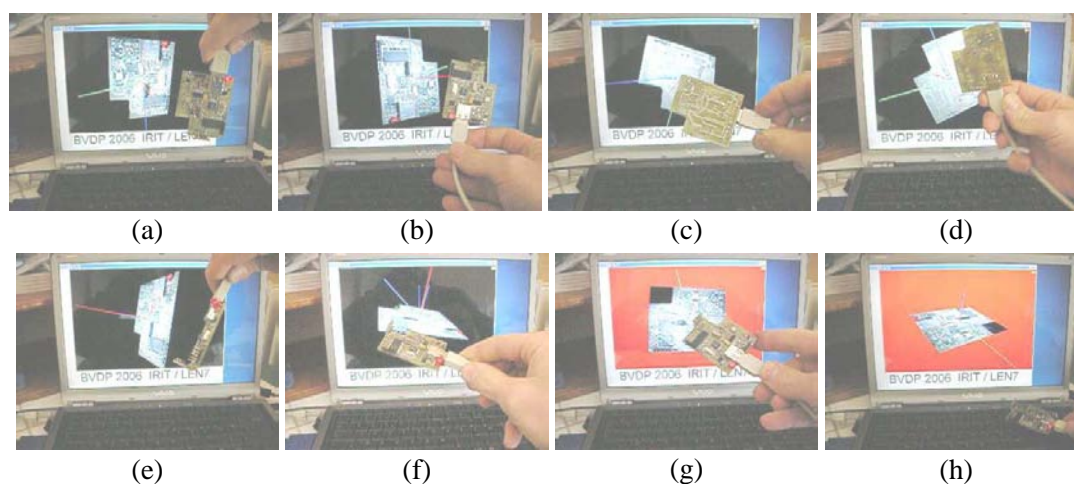


FIG. 2.7 – Photographie du démonstrateur : (a-f) Orientations retrouvées correctement. (g) Orientation erronée (et détectée comme telle) due à une accélération parasite provoquée par un choc avec le doigt. (h) Orientation erronée (et détectée comme telle) due à un champ magnétique parasite émis par l'ordinateur.

2.3 La caméra équipée du capteur d'orientation

2.3.1 Description

Le capteur d'image utilisé dans le prototype provient d'une Webcam Logitech QuickCam Pro 4000 et est utilisé à une résolution de 352×288 pixels à un débit de 15 images par seconde. Cette caméra permet le contrôle automatique ou manuel de l'exposition via les paramètres de durée d'exposition et de gain. De plus, notre caméra génère très peu de distorsions radiales et peut donc être assez bien approximée par un simple modèle de caméra trou d'épingle.

La figure 2.8 (a) montre le premier prototype dans lequel le capteur d'orientation est simplement fixé au boîtier de la webcam. Les figures 2.8 (b) et (c) montrent le second prototype, dans lequel la caméra et le capteur d'orientation sont intégrés à l'intérieur d'un unique boîtier dont les dimensions sont $100 \times 60 \times 22$ mm.

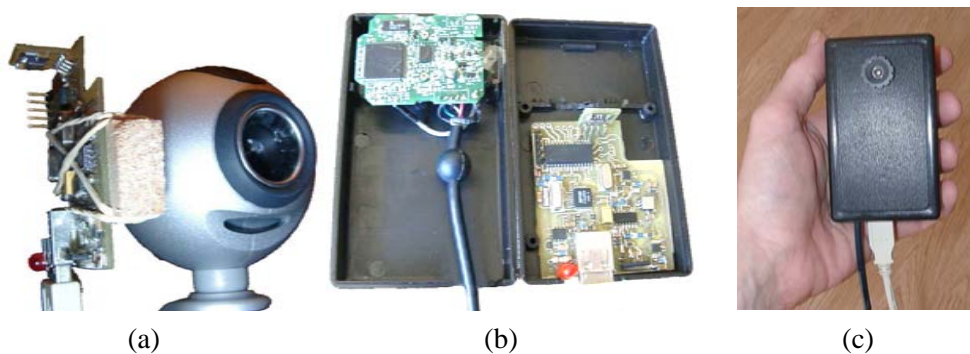


FIG. 2.8 – (a) Le premier prototype de caméra. (b) Intégration de l'électronique dans le boîtier de la seconde caméra. (c) La seconde caméra vue de l'extérieur.

2.3.2 Étalonnage de l'orientation relative

Une fois le capteur d'orientation étalonné, il est nécessaire d'étalonner la caméra complète, c'est à dire estimer ses paramètres intrinsèques et son orientation relativement à celle du capteur d'orientation.

Ceci est réalisé en acquérant des images j d'un motif en forme d'échiquier dont un des axes est orienté vers le nord. Nous utilisons alors la boîte à outils pour Matlab de Bouguet [125] pour estimer les poses de la caméra correspondant à chaque image $(R_{i(j)}, T_{i(j)})$ ainsi que les paramètres intrinsèques de la caméra, considérés constants pour toutes les images.

En utilisant l'orientation $R_{c(j)}$ (de ω_r vers ω_c) telle que représentée sur les figures 2.4 et 2.9 (a) fournie par le capteur d'orientation pour chaque image, la rotation R_δ (du repère caméra ω_i au repère ω_c) est alors calculée afin de minimiser l'écart entre $R_{i(j)}R_\delta$ et $R_{c(j)}$ dans les n images, $(R_{i(j)})$ du repère ω_r vers ω_i .

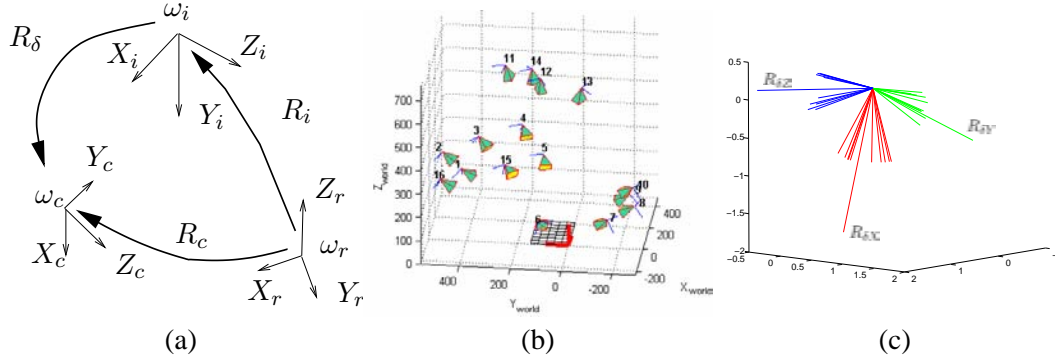


FIG. 2.9 – (a) Les différents repères et les rotations correspondantes. (b) Différentes poses de caméra estimées. (c). Le R_δ obtenu ainsi que l'erreur angulaire correspondant aux différentes poses (amplifiée 5 fois).

Une matrice de rotation peut s'écrire sous la forme d'un vecteur et d'un angle de rotation autour de ce vecteur. Cet angle de rotation est utilisé comme norme pour mesurer l'écart entre deux matrices de rotation dans l'équation (2.9).

Nous souhaitons aligner les repères obtenus par rotation de $R_{i(j)}R_\delta$ et de $R_{c(j)}$, ce qui est équivalent à aligner $R_{i(j)}R_\delta R_{c(j)}^T$ avec le repère correspondant à la matrice identité et donc à minimiser l'angle de rotation induit par cette matrice. Dans le critère suivant, $\Phi(R)$ désigne l'angle associé à la matrice de rotation R . R_δ est paramétré par un vecteur de norme unitaire en coordonnées sphériques (2 paramètres) et un angle de rotation autour de ce vecteur (1 paramètre).

$$\widehat{R}_\delta = \arg \min_{R_\delta} \sum_{j=1}^n \|\Phi(R_{i(j)}R_\delta R_{c(j)}^T)\|^2 \quad (2.9)$$

Cette estimation permet d'obtenir une orientation relative de la caméra plus précise qu'à partir d'une seule image. De plus, afin d'éviter un biais dans l'estimation de R_δ , il est préférable d'échantillonner au mieux l'espace des orientations possibles.

Ce critère est minimisé en utilisant l'algorithme de Levenberg Marquardt en partant d'une solution initiale $R_{\delta(0)} = R_{c(0)}^T R_{i(0)}$, $R_{c(0)}$ et $R_{i(0)}$ étant les rotations correspondant à une pose quelconque.

Les différentes poses de la caméra estimées à partir des images sont visibles sur la figure 2.9 (b). La figure 2.9 (c) montre les axes du repère obtenu après rotation par R_δ . Les grands axes représentent le repère obtenu à partir de toutes les poses alors que les petits montrent ceux obtenus à partir d'une seule pose. Les écarts angulaires sont agrandis cinq fois pour améliorer la visibilité.

L'angle maximum formé entre $R_{i(j)}R_\delta$ et les différents $R_{c(j)}$ est inférieur à cinq degrés dans cette expérience. Cette valeur est évidemment bien plus importante lorsque le capteur subit des accélérations ou un champ magnétique parasite.

2.4 Mosaïquage d'images

Une application évidente pour la caméra est le mosaïquage direct d'images à partir de l'orientation fournie par le capteur.

Une mosaïque d'images peut être générée à partir d'images acquises depuis des points de vues identiques. Au plus, le déplacement de la caméra doit être très petit par rapport à la distance de la scène observée pour éviter le phénomène de parallaxe. Les images doivent donc idéalement être obtenues par pure rotation de la caméra. Nous considérons dans la suite que les paramètres intrinsèques de la caméra ne changent pas entre les images.

Les images peuvent être générées en projection plane, cylindrique, sphérique ou autre.

2.4.1 Avantages liés au capteur d'orientation

Le logiciel AutoStitch, développé par Brown, en utilisant le descripteur SIFT, tel que décrit dans [43] permet de mosaïquer des images sans connaissance a priori de l'orientation de la caméra et d'obtenir des résultats très bons, quand certaines conditions sont vérifiées. L'approche proposée ici possède principalement six avantages par rapport à celle n'utilisant pas de capteur d'orientation :

Traitements simplifiés : Comme l'orientation de la caméra est connue a priori, il n'est pas nécessaire de détecter des points homologues dans les images afin de les mosaïquer. Certes, la mosaïque obtenue est grossière à cause de l'imprécision du capteur d'orientation mais elle pourra être affinée par la suite. Le mosaïquage correspond alors uniquement à la projection des différentes images sur une surface donnée en utilisant les informations du capteur d'orientation. Ce traitement requiert très peu de ressources matérielles et il est envisageable de l'effectuer directement dans un appareil photo numérique. Ainsi, il est possible, directement sur le site où les photos sont acquises, d'obtenir l'image mosaïquée et de refaire l'acquisition si nécessaire.

Appariements simplifiés et plus robustes : Lorsque l'on souhaite affiner la mosaïque, il est nécessaire de détecter des correspondances entre les images. Nous montrerons par la suite comment le capteur d'orientation permet de simplifier cette détection et de la rendre plus discriminante qu'en son absence, en utilisant des approches purement géométriques ou des approches utilisant des descripteurs des points qui exploitent l'orientation fournie par le capteur.

Choix automatique des images : Lorsque l'on souhaite acquérir des images pour les mosaïquer, il est nécessaire de les choisir afin d'obtenir un recouvrement suffisant entre les différentes images et de couvrir tout le champ souhaité.

Si c'est une vidéo qui est acquise, le choix des images est fait à posteriori lors du mosaïquage mais la résolution de chaque image n'est généralement pas très élevée.

Si ce sont différentes images fixes qui sont acquises, la résolution peut alors être plus élevée, mais le choix des images doit être fait sur le site. L'orientation fournie en temps réel par le

capteur permet de choisir les instants auxquels acquérir les différentes images pour assurer un compromis entre qualité du recouvrement et nombre d'images à stocker.

De plus, comme la mosaïque est générée en temps réel, il est possible à tout instant de connaître les zones manquantes afin de pointer la caméra dans leur direction pour compléter automatiquement la mosaïque.

Traitements des zones non texturées ou répétitives : Certaines scènes contiennent de grosses portions non texturées ou répétitives et leurs images sont impossibles ou très difficiles à mosaïquer automatiquement, celles-ci ne contenant pas suffisamment d'information pour déterminer leur orientation respective. Grâce à l'information du capteur, il est possible de mosaïquer automatiquement de telles images.

Genèse d'images orientées : Les images reconstruites sont orientées, et donc l'horizon de la scène est projeté sur une ligne droite horizontale dans l'image et la première colonne de l'image peut par exemple correspondre à l'orientation du nord. Ceci permet d'éviter d'obtenir des mosaïques non réalistes avec des lignes d'horizon courbes. Dans l'application de localisation par l'image, cette connaissance de l'orientation simplifie l'estimation de pose. Ceci permet aussi d'aider le géoréférencement des mosaïques dans le cadre d'applications de navigation virtuelle, car il suffit alors d'indiquer leurs positions.

Cette propriété peut aussi être mise à profit pour générer des images haute dynamique (High Dynamic Range, utilisées par exemple par Waggershauer dans [48]) en acquérant plusieurs mosaïques depuis la même position. Ces mosaïques, acquises avec des paramètres d'exposition différents, n'ont qu'à être légèrement alignées pour générer une mosaïque dont la dynamique est plus grande et il est donc possible d'obtenir un rendu moins affecté par les sur et sous-expositions.

Genèse d'images omnidirectionnelles : La connaissance de l'orientation simplifie grandement la création d'images complètement omnidirectionnelles. Sans l'aide du capteur d'orientation, le problème de *fermeture* apparaît lorsque l'on souhaite mosaïquer des images acquises lors d'une rotation complète de la caméra sur 360 degrés. À cause des erreurs accumulées dans l'estimation des poses successives, il est assez difficile de mosaïquer la dernière image et la première. Ce problème peut être évité en utilisant l'ajustement de faisceaux (Bundle Adjustment) comme proposé par Lauchlanand et al. dans [126] et par Brown et Lowe dans [43] pour estimer la solution d'alignement complète au lieu de mosaïquer itérativement chaque nouvelle image. Néanmoins ce type de méthode nécessite une bonne détection des différentes correspondances entre les images et une bonne estimée initiale des différentes poses.

Grâce à l'orientation fournie par le capteur, le problème de fermeture n'est pas présent, chaque image étant individuellement associée à une orientation absolue.

Gestion des scènes mobiles : Finalement, notre caméra permet de traiter très efficacement les scènes comportant des parties mobiles qui sont considérées comme des outliers lors de la reconstruction de la mosaïque.

Dans le cas où la zone en mouvement recouvre la majeure partie de l'image, il n'est généralement pas possible de calculer automatiquement les poses des différentes images, le pourcentage d'outliers étant trop important. L'alignement des images risque alors d'être effectué sur la forme en mouvement plutôt que sur le fond. Grâce à l'orientation fournie par le capteur, le mouvement de la forme peut être détecté car il est incohérent avec la rotation mesurée de la caméra.

Dans le cas où la zone en mouvement recouvre une petite partie de l'image, la pose correspondant à l'image peut généralement être estimée correctement grâce à des méthodes robustes, sans nécessiter de capteur d'orientation. Par exemple, sur la figure 1.2 du chapitre 1, le mouvement de la personne au premier plan entre les images (b) et (c) n'a pas faussé l'alignement des images mais a fait apparaître un fantôme sur l'image (e).

Pour éviter ce problème, il faut disposer d'un nombre suffisant (au moins deux) d'images dans lequel le fond est visible. Le fond et la (ou les) forme(s) mobile(s) peuvent alors être segmentées en tirant partie du fait que seul le fond est fixe et donc stable une fois reprojété sur l'image mosaïquée.

Le capteur d'orientation peut être mis à profit pour aider à cette tâche. En effet, il est possible de détecter en temps réel les zones en mouvement dans les images, en les reprojétant individuellement sur la mosaïque et en comparant les valeurs issues de différentes images.

2.4.2 La création de mosaïque

En utilisant l'orientation fournie par le capteur d'orientation, chaque image de la caméra est projetée sur la mosaïque. Néanmoins, à cause des imprécisions dans l'orientation détectée pour les différentes images, les images reprojétées ne se connectent pas exactement. Nous montrons d'abord comment obtenir une mosaïque approximative en temps réel puis utilisons l'ajustement de faisceaux pour améliorer sa qualité.

Dans la suite, nous considérons une mosaïque en projection sphérique, c'est à dire que chaque pixel de l'image résultante correspond à un point sur la sphère dont les coordonnées sont liées à la position pixellique dans la mosaïque. N'importe quelle surface peut être utilisée pour échantillonner l'image mais la sphère présente l'intérêt d'autoriser la représentation du champ de vision omnidirectionnel sur une image rectangulaire avec une paramétrisation adaptée.

Une image en projection sphérique est une représentation sur un plan de points à la surface de la sphère. L'élévation ϕ et l'azimut θ , respectivement compris entre $[0; 2\pi[$ et $[0; \pi[$ sont une représentation possible pour définir les coordonnées d'un point sur une sphère de rayon unitaire. Elles portent le nom de coordonnées sphériques ou polaires. L'image en projection sphérique correspond à un échantillonnage le long des colonnes (resp. lignes) pour différents θ (resp. ϕ).

Soit P_i un point 3D correspondant à un pixel i et dont les coordonnées sphériques sont (ϕ_i, θ_i) , ses coordonnées cartésiennes sur la sphère sont :

$$P_i = [x_i \ y_i \ z_i]^T; x_i = \cos(\theta_i) \sin(\phi_i); y_i = \sin(\theta_i) \sin(\phi_i); z_i = \cos(\phi_i) \quad (2.10)$$

Soit $Q_j = [abcd]^T$ le quaternion correspondant à l'orientation \widehat{R}_j de la caméra pour

l'image j , déduite de l'orientation fournie par le capteur $R_{c(j)}$ et de la rotation R_δ obtenue par étalonnage. Les indices j sont omis dans les paramètres des quaternions pour simplifier les expressions. Une méthode d'obtention des coefficients du quaternion correspondant à une matrice de rotation est donnée en Annexe C.

$$Q_j : \widehat{R}_j = R_{c(j)} R_\delta^T \quad (2.11)$$

Soit K la matrice modélisant la caméra trou d'épingle définie au chapitre 1 par l'équation (1.2). Un point P_i sur la sphère est projeté sur l'image j en un point dont les coordonnées homogènes sont $p_{Hij} = [u_{ij} \ v_{ij} \ w_{ij}]^T$ et dont les coordonnées cartésiennes sont $p_{ij} = [u_{ij}/w_{ij} \ v_{ij}/w_{ij}]^T$ obtenues par la fonction $p_{ij} = D(p_{Hij})$. L'image p_{ij} du point P_i dans l'image j est obtenue par $p_{ij} = D(K\widehat{R}_j P_i)$.

Si p_{ij} est situé à l'intérieur des frontières de l'image et si la composante z de $\widehat{R}_j P_i$ est positive (c'est à dire que la caméra fait face au point), alors le point P_i est visible sur l'image j et les valeurs RGB de p_{ij} sont utilisées dans la mosaïque pour le pixel i .

Ce calcul est effectué pour tous les points de la sphère et pour chaque image de la séquence acquise pour mettre à jour la mosaïque.

2.4.3 Résultats

Dans cette partie, nous nous intéressons aux résultats de mosaïques grossières pouvant être obtenues avec des traitements très simples, implémentables à l'intérieur d'un appareil photo numérique par exemple.

2.4.3.1 Création d'une mosaïque sphérique

La figure 2.10 montre une image de la caméra, sa projection sur la mosaïque orientée reconstruite en basse résolution (1024*512 pixels) et la mosaïque complète. Celle-ci est obtenue en temps réel à partir d'une séquence d'approximativement dix secondes. Aucun traitement n'est effectué à posteriori ; La mosaïque résulte seulement de la projection des différentes images avec écrasement des anciennes valeurs de la mosaïque par les nouvelles images acquises.

La colonne de gauche de l'image pointe vers le nord et l'horizon de la scène correspond à la ligne horizontale centrale dans la mosaïque. Le champ de vision de la caméra est très réduit et la scène contient de nombreuses et larges zones non texturées sur les murs et le plafond ainsi que des zones sur-exposées au niveau de la fenêtre. Pour ces raisons, les images correspondantes n'auraient pas pu être mosaïquées automatiquement, car elles ne contiennent pas de point caractéristique.

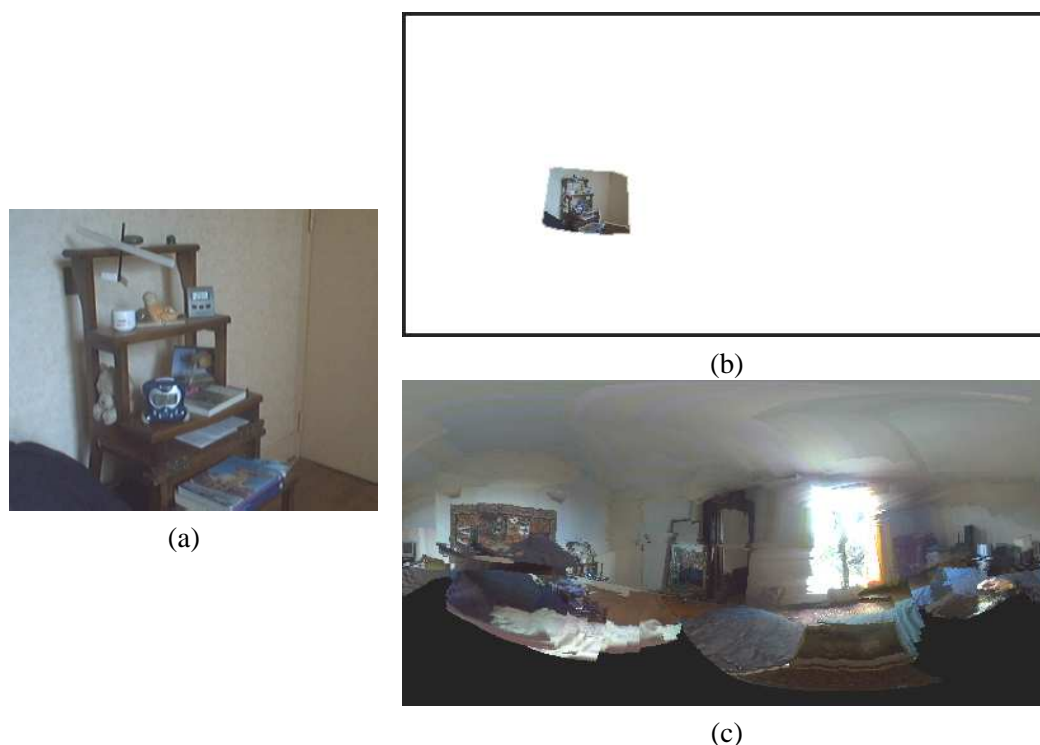


FIG. 2.10 – (a) Une image acquise par la caméra. (b) L'image projetée sur la mosaïque. (c) La mosaïque résultante de l'intégration de toutes les images acquises.

2.4.3.2 Création d'une mosaïque plane

La figure 2.11 montre une mosaïque en projection plane. Celle-ci est obtenue en temps réel à partir d'une séquence d'approximativement sept secondes. Comme précédemment, aucun traitement n'est effectué à posteriori.

Le centre de la mosaïque pointe vers la direction visée par la première image acquise avec la contrainte que l'horizon de la scène soit une ligne horizontale située à mi hauteur de la mosaïque.

Ce traitement permet d'augmenter virtuellement le champ de vision de la caméra, qui atteint 160° sur la mosaïque. Comme précédemment, l'emploi du capteur d'orientation a permis de mosaïquer les zones sur-exposées, situées sur la droite de l'image.

2.4.3.3 Mosaiquage de scènes mobiles

Le capteur d'orientation est maintenant utilisé pour mosaïquer des images contenant des zones mobiles. La figure 2.12 (g) montre une mosaïque résultante d'une séquence de 130 images. Les images (a-e) sont des images tirées de cette séquence dans lesquelles il y a occultation du fond par une personne mobile.



FIG. 2.11 – Mosaïque en projection plane. Le champ de vision est de 160 degrés.

Chaque image de la séquence est projetée individuellement sur la mosaïque (par exemple l'image (a) vers la partie de mosaïque (f)). La surface de la sphère est alors découpée en portions (des carrés dans la mosaïque). Un histogramme est alors calculé pour chaque portion de chaque image. Les portions incohérentes entre les différentes images sont considérées comme mobiles et ne sont donc pas prises en compte pour le calcul de la mosaïque finale.

Dans cette expérience, on pourra noter que le mur est d'un ton chair assez proche de celui de la peau de la personne en mouvement mais ceci n'a pas empêché la mosaïque d'être correctement reconstruite.

Une autre remarque importante est que l'occultation recouvre la majeure partie de l'image, par exemple en (a). Malgré ceci, les rares zones non occultées sont détectées et utilisées pour générer la mosaïque, ce qui aurait été très difficile à faire (voir impossible) en l'absence de l'information d'orientation fournie par le capteur.

2.4.3.4 Mosaïquage haute dynamique

La caméra est utilisée pour acquérir deux mosaïques depuis la même position mais avec des paramètres d'exposition différents. Ces deux mosaïques sont ensuite fusionnées pour obtenir une mosaïque dont la dynamique est plus grande que celle de la caméra.

Comme cette thèse ne traite pas particulièrement des images HDR, l'estimation de la réponse du capteur d'image en fonction de la luminosité reçue n'a pas été étudiée. Ainsi, la mosaïque HDR générée n'est pas réellement une représentation de la luminance de la scène mais simplement un mixage entre les deux mosaïques permettant de faire apparaître les différents détails.

Dans l'exemple visible sur la figure 2.13, les mosaïques (a) et (b) sont mixées en la mosaïque (c). Le mixage est effectué directement sur les images grâce au fait que les mosaïques sont orientées. On pourra noter, une nouvelle fois, que les zones surexposées de l'image (b) ont été correctement mosaïquées grâce au capteur d'orientation. Sur la mosaïque HDR obtenue, les détails tels que l'écran d'ordinateur portable, montré par une flèche, sont visibles alors qu'ils ne l'étaient pas sur (a).

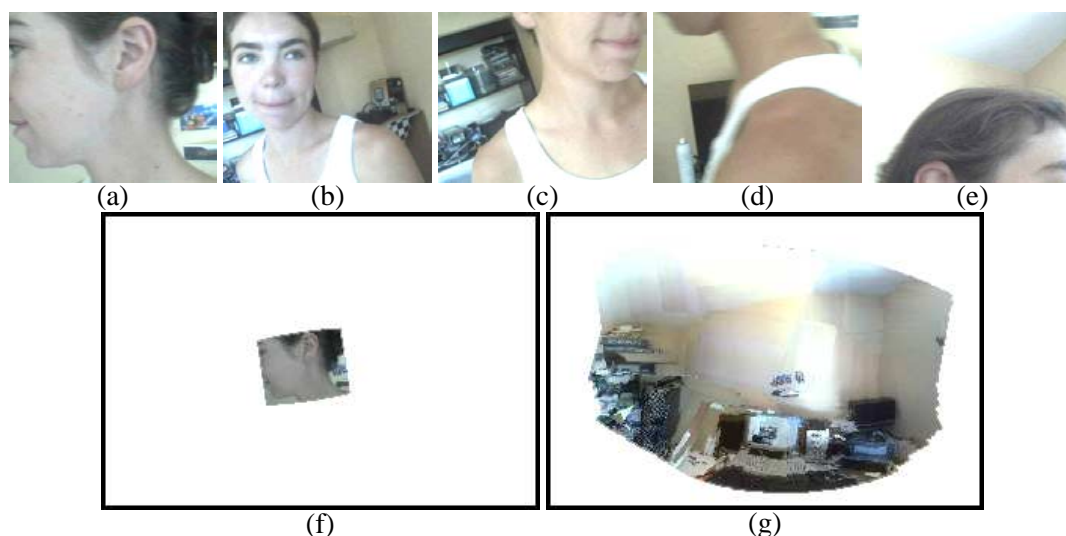


FIG. 2.12 – Cinq images (a-e) utilisées pour la mosaïque et comportant l'image d'une personne mobile. (f) L'image (a) reprojétée sur une portion de la mosaïque. (g) La mosaïque finale.

Dans cette expérience, les deux mosaïques (a) et (b) sont acquises individuellement. Dans le cas d'un traitement embarqué, il est envisageable de modifier dynamiquement les réglages d'exposition pour multiplexer temporellement l'acquisition des deux mosaïques et empêcher ainsi un déplacement de la caméra entre les deux acquisitions. Dans ce cas, la mosaïque HDR peut être générée en temps réel.

2.5 Affinement de mosaïque

La qualité de la mosaïque peut être améliorée grâce à une étape d'alignement des images comme le propose McLauchlan et al. dans [127]. Partant de la solution initiale fournie par le capteur d'orientation, cette étape permet de raffiner les paramètres. La métrique choisie pour mesurer l'alignement est basée sur un critère géométrique de distance entre des points correspondants dans les différentes images. Ceci a été préféré à un critère photométrique, tel que proposé par exemple par Bartoli dans [42], car il est moins complexe à minimiser et permet la prise en compte des correspondances de points entre les multiples images.

L'étape d'affinage nécessite la détection de correspondances de points entre les images puis une étape d'estimation de paramètres permettant d'aligner ces correspondances, appelée ajustement de faisceaux.

Finalement, la mosaïque est générée grâce à un post-traitement permettant de cacher les jointures restantes entre les différentes images.

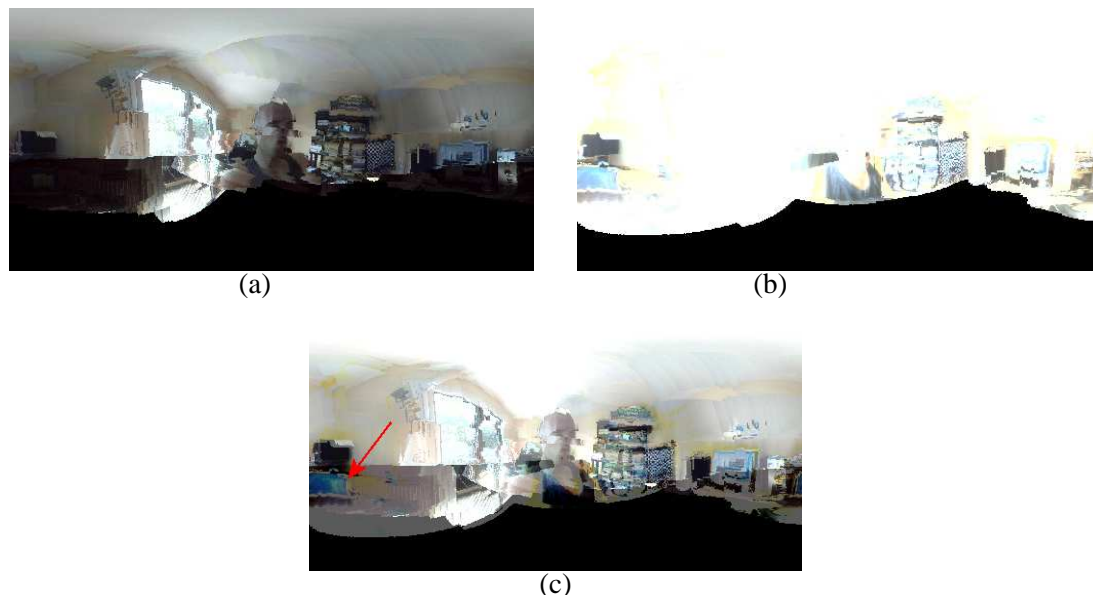


FIG. 2.13 – Mosaïque haute dynamique (c) obtenue à partir de deux mosaïques : (a) plutôt sous-exposée et (b) plutôt sur-exposée.

2.5.1 Détection de correspondances

Toutes les images ne sont pas nécessaires pour raffiner la mosaïque car la caméra acquiert les images à une fréquence de 15 images par seconde. Un sous ensemble doit donc être sélectionné afin d'éviter des calculs inutiles. La décimation est très simple et consiste à sélectionner les images de manière à ce qu'elles aient au moins la moitié de leur surface en commun avec une autre image. L'orientation approchée fournie par le capteur est suffisamment précise pour calculer le jeu d'images à conserver.

La recherche de correspondances ne peut pas être effectuée sur chaque pixel des images retenues et certains doivent donc être sélectionnés. Nous utilisons le détecteur de Harris [37] pour détecter des coins dans les images. Des correspondances sont ensuite établies entre des coins détectés dans différentes images.

Deux méthodes pour l'appariement sont proposées. La première n'utilise pas l'information photométrique et permet un traitement très rapide, avec quelques échecs lorsque le nombre de points détectés est important. La seconde nécessite plus de calcul mais est plus discriminante, menant à un taux moindre de mauvaises correspondances.

Sans descripteur de points : Il est possible de détecter des correspondances entre les points des différentes images en utilisant des critères purement géométriques sur les positions détectées de coins.

Dans le mosaïquage d'images standard, où il n'y a pas de connaissance de l'orientation des vues a priori, ceci est effectué grâce à des estimateurs robustes, tel le RANSAC [128] par

exemple. Des homographies sont alors estimées à partir de quadruplets de correspondances tirées au hasard, et le quadruplet pour lequel le plus grand nombre de points correspondent dans les deux images (sont à une petite distance) est considéré comme le bon et les couples correspondants comme inliers.

Ici, grâce à la connaissance de l'orientation approchée fournie par le capteur, il n'est pas nécessaire de tester aléatoirement les couples pour trouver une homographie. Au lieu de cela, nous projetons simplement les coins détectés dans les différentes images sur la sphère (support de la mosaïque) en utilisant l'orientation fournie par le capteur. Une correspondance est ensuite établie si des points projetés depuis différentes images sont à une distance inférieure à un seuil sur la sphère et qu'aucun autre point des mêmes images n'est trop proche dans le voisinage.

Avec descripteur de points : Dans le mosaïquage d'images standard, où il n'y a pas de connaissance de l'orientation des vues a priori, il est possible d'utiliser des descripteurs de points permettant l'appariement de points homologues plutôt que d'uniquement essayer d'apparier les coins avec un critère géométrique. Afin de permettre l'appariement dans des conditions de prises de vue différentes, ces descripteurs doivent être invariants aux transformations correspondantes. Le Scale Invariant Feature Transform (SIFT) de Lowe [39], par exemple, est fréquemment utilisé.

Dans le cas étudié, la connaissance de l'orientation est mise à profit pour générer des descripteurs de points qui ne soient pas invariants à ces transformations. Contrairement au descripteur SIFT par exemple, le descripteur proposé est très simple. La focale de notre caméra étant fixée, le descripteur est généré en projetant l'image sur la mosaïque (en utilisant l'orientation fournie par le capteur) et en considérant une fenêtre carrée autour du point considéré. Même si le ré-échantillonnage dû à la projection sur la mosaïque entraîne une légère perte de précision pour la localisation des homologues, les résultats obtenus sont satisfaisants.

L'intérêt d'un tel descripteur est qu'il est plus discriminant que le SIFT, car il n'autorise pas de transformations affines entre les homologues. De plus, son calcul est très rapide, puisqu'il correspond simplement à la reprojexion de l'image sur la sphère. Finalement, un surcroît de robustesse est obtenu en imposant que deux descripteurs correspondent à des positions proches une fois projetés sur la sphère en utilisant l'orientation fournie par le capteur d'orientation.

La figure 2.14 montre un exemple d'utilisation du descripteur. En (a) et (c), deux images issues de la caméra sont visibles. Les coins sont détectés et nous nous intéressons au coin supérieur de l'écran, détecté dans les deux images. Les zones de 50*50 pixels centrées sur ces points sont visibles en (b) et (d). De tels descripteurs ne peuvent pas être comparés directement, car leurs orientations sont différentes. En revanche, si l'on projette les images de la caméra sur la sphère, avec un échantillonnage en coordonnées sphériques de 4000*2000 pixels, les images (e) et (g) sont obtenues. En observant des fenêtres de 65*65 pixels autour de la projection des coins sur la sphère, on obtient les images (f) et (h). Ces descripteurs peuvent être directement comparés, avec une mesure de corrélation type ZNCC ou SSD par exemple.

Les faibles erreurs d'estimation d'orientation par le capteur ne semblent pas avoir d'influence importante localement et ne gênent donc pas la création de ces descripteurs.

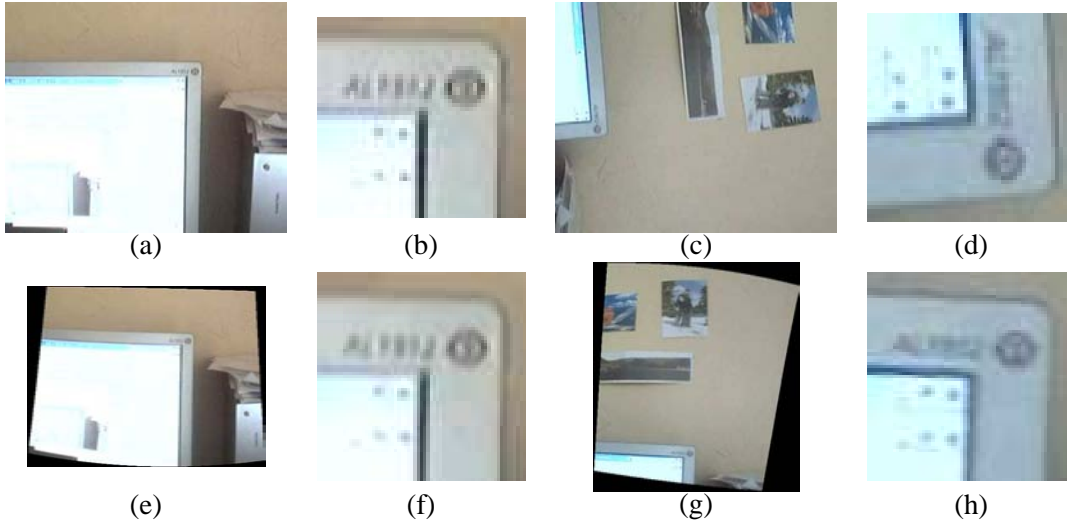


FIG. 2.14 – (a) et (c) Deux images obtenues par la caméra et (b) et (d) les descripteurs correspondants au coin supérieur de l'écran à partir de ces images. (e) et (g) Les images reprojetées sur la sphère et (f) et (h) les descripteurs correspondants.

2.5.2 Ajustement de faisceaux

L'ajustement de faisceaux est utilisé pour améliorer l'estimation de l'orientation des différentes images et les paramètres intrinsèques de la caméra. Son utilisation pour créer des mosaïques a déjà été proposée par exemple par Lauchlanand et al. dans [126] mais sans utiliser la connaissance de l'orientation fournie par le capteur. Le lecteur pourra trouver plus d'informations sur l'ajustement de faisceaux dans l'article de Triggs et al. [129] ou le livre de Slama [130]. Cette méthode, issue du domaine de la photogrammétrie, est généralement utilisée pour raffiner une reconstruction 3D en multiscopie mais est facilement adaptable à d'autres problèmes. Elle consiste à minimiser une fonction de coût f définie par l'équation (2.12), permettant d'aligner les données prédites par un modèle aux différentes observations dans les images. K est la matrice caméra, définie par l'équation 1.2. R_j est la matrice rotation associée à la caméra pour l'image j et P_i est la position du point i sur la sphère.

La minimisation est effectuée itérativement en utilisant l'algorithme de Levenberg-Marquart. Une projection du point i détectée dans l'image j est notée $\tilde{p}_{ij} = [\tilde{u}_{ij} \ \tilde{v}_{ij}]^T$ et est appelée une observation. Chaque observation fournit deux résidus, $res_{u(ij)}$ et $res_{v(ij)}$, dont les équations sont données par (2.13) et (2.14).

L'ajustement de faisceaux permet, sur les m points et n images, d'affiner les paramètres de pose (ici les quaternions Q_j), les paramètres intrinsèques (supposés constants) ainsi que les positions des points P_i sur la mosaïque, tous regroupés en un vecteur de paramètres x .

Les distorsions radiales de la caméra n'ont pas été prises en compte car la caméra utilisée ne le nécessite pas. Néanmoins, elles peuvent être intégrées très facilement dans le modèle pour l'ajustement de faisceaux. De même, il est très facile d'adapter la méthode pour qu'elle fonctionne avec des paramètres de caméra variables.

$$f = \sum_{j=1}^n \left(\sum_{i=1}^m (\|D(KR_j P_i) - [\widetilde{u}_{ij} \quad \widetilde{v}_{ij}]^T\|^2) \right) = \sum_{j=1}^n \left(\sum_{i=1}^m (res_{u(ij)}^2 + res_{v(ij)}^2) \right) \quad (2.12)$$

$$res_{u(ij)} = \frac{\alpha_u \left((1 - 2b^2 - 2c^2)x_i + (2ab - 2cd)y_i + (2ac + 2bd)z_i \right)}{(2ac - 2bd)x_i + (2bc + 2ad)y_i + (1 - 2a^2 - 2b^2)z_i} + p_u - \widetilde{u}_{ij} \quad (2.13)$$

$$res_{v(ij)} = \frac{\alpha_v \left((2ab + 2cd)x_i + (1 - 2a^2 - 2c^2)y_i + (2bc - 2ad)z_i \right)}{(2ac - 2bd)x_i + (2bc + 2ad)y_i + (1 - 2a^2 - 2b^2)z_i} + p_v - \widetilde{v}_{ij} \quad (2.14)$$

La matrice jacobienne J du problème d'ajustement de faisceaux est définie par les sous matrices $J_{intrinseque}$, $J_{extrinseque}$ et J_{point} suivantes :

$$J_{intrinseque}(\mathbf{x}) = \begin{pmatrix} \frac{\delta res_{u(ij)}}{\delta \alpha_u}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta \alpha_v}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta p_u}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta p_v}(\mathbf{x}) \\ \frac{\delta res_{v(ij)}}{\delta \alpha_u}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta \alpha_v}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta p_u}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta p_v}(\mathbf{x}) \end{pmatrix} \quad (2.15)$$

$$J_{extrinseque}(\mathbf{x}) = \begin{pmatrix} \frac{\delta res_{u(ij)}}{\delta a}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta b}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta c}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta d}(\mathbf{x}) \\ \frac{\delta res_{v(ij)}}{\delta a}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta b}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta c}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta d}(\mathbf{x}) \end{pmatrix} \quad (2.16)$$

$$J_{point}(\mathbf{x}) = \begin{pmatrix} \frac{\delta res_{u(ij)}}{\delta x_i}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta y_i}(\mathbf{x}) & \frac{\delta res_{u(ij)}}{\delta z_i}(\mathbf{x}) \\ \frac{\delta res_{v(ij)}}{\delta x_i}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta y_i}(\mathbf{x}) & \frac{\delta res_{v(ij)}}{\delta z_i}(\mathbf{x}) \end{pmatrix} \quad (2.17)$$

Ces termes sont organisés dans la matrice jacobienne complète. La figure 2.15 montre un exemple de structure pour quatre images et l'observation de quinze points. Des pixels blancs indiquent des valeurs à zéro. Chacun des couples de lignes successives correspond aux deux résidus associés à une observation. La matrice est organisée en quatre parties (notées de 1 à 4), qui correspondent à chaque image j . Les lignes j contenant uniquement des zéros correspondent à des points qui ne sont pas observés dans l'image j . Les quatre valeurs de la colonne A correspondent à $J_{intrinseque}$ et chaque observation fournit de l'information sur les paramètres intrinsèques. La colonne B (resp. C,D,E) contient $J_{extrinseque}$ pour l'observation de la première (resp 2,3,4) image. Finalement, la colonne F contient les J_{point} à une position horizontale correspondant au numéro du point observé.

Partant d'un vecteur de paramètres \mathbf{x}_0 , la valeur est affinée à chaque itération k , grâce au calcul d'un incrément $\mathbf{d}_k = [\Delta(P) \quad \Delta(M)]^T$ permettant la mise à jour de \mathbf{x}_k , avec $\Delta(P)$ (resp. $\Delta(M)$) les incréments sur les paramètres intrinsèques et de poses (resp. les positions des points).

Déroulement d'une itération : La valeur numérique des termes de la jacobienne est calculée pour le vecteur de paramètres \mathbf{x}_k en cours. Ensuite, la matrice hessienne est approximée par $J(\mathbf{x}_k)^T J(\mathbf{x}_k)$.

Le gradient $-\epsilon = J(\mathbf{x}_k)^T r(\mathbf{x}_k) = [\epsilon(P) \quad \epsilon(M)]^T$ est composé de deux parties, $\epsilon(P)$ (resp. $\epsilon(M)$) étant lié aux paramètres intrinsèques et de poses (resp. les positions des points). $r(\mathbf{x}_k)$ est le vecteur obtenu par concaténation des résidus $res_{u(ij)}$ et $res_{v(ij)}$ pour les paramètres \mathbf{x}_k .

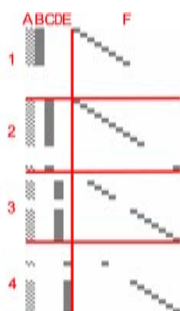


FIG. 2.15 – La structure de la jacobienne.

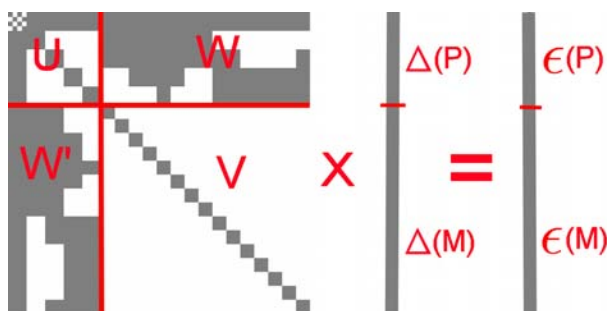


FIG. 2.16 – Le système des équations normales augmentées à résoudre à chaque itération.

λ_k est le paramètre de la méthode de Levenberg-Marquardt et est mis à jour à chaque itération pour choisir le comportement de la minimisation : Gauss Newton ou gradient en fonction de l'éloignement de \mathbf{x}_k à la solution. Il est réglé à 0.001 pour la première itération et divisé par 10 à chaque itération si l'évaluation de la fonction de coût est moindre qu'à l'itération précédente. Sinon, il est multiplié par 10.

Le système des équations normales augmentées à résoudre à chaque itération est le suivant :

$$(J(\mathbf{x}_k)^T J(\mathbf{x}_k) + \lambda_k I) \mathbf{d}_k = -J(\mathbf{x}_k)^T r(\mathbf{x}_k) \quad (2.18)$$

Le terme *augmentées* est dû au fait que le système fait intervenir $J(\mathbf{x}_k)^T J(\mathbf{x}_k) + \lambda_k I$ au lieu de l'approximation de la hessienne seule $J(\mathbf{x}_k)^T J(\mathbf{x}_k)$. On pourra noter que ces deux matrices ont la même structure. La figure 2.16 montre la forme particulière de ce système linéaire à résoudre à chaque itération k . Si cette structure épars du système n'est pas exploitée, l'incrément \mathbf{d}_k peut être obtenu de la manière suivante :

$$\mathbf{d}_k = -(J(\mathbf{x}_k)^T J(\mathbf{x}_k) + \lambda_k I)^{-1} J(\mathbf{x}_k)^T r(\mathbf{x}_k) \quad (2.19)$$

Le vecteur de paramètres est mis à jour en se déplaçant de l'incrément \mathbf{d}_k puis en normalisant les quaternions pour qu'ils correspondent à des rotations et les coordonnées des points pour les ramener sur la sphère unitaire (la mosaïque). Tout ceci est réalisée par la fonction de mise à jour M :

$$\mathbf{x}_{k+1} = M(\mathbf{x}_k, \mathbf{d}_k) \quad (2.20)$$

Condition d'arrêt des itérations : Le processus est itéré jusqu'à ce que la valeur de \mathbf{x}_k soit stabilisée. La convergence est en générale assez rapide, d'autant plus que la valeur de \mathbf{x}_0 est proche de la valeur optimale, ce qui est le cas ici, puisque la caméra est étalonnée et que les positions des points sur la sphère sont à peu près connues.

Un système épars : La résolution des équations normales augmentées fait intervenir des matrices de dimensions élevées. Il est possible de diminuer la complexité de la résolution en

exploitant la structure particulière du système d'équations, comme expliqué par Pollefeys dans [131]. L'équation (2.18) a la structure suivante :

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta(P) \\ \Delta(M) \end{pmatrix} = \begin{pmatrix} \epsilon(P) \\ \epsilon(M) \end{pmatrix} \quad (2.21)$$

Si V est inversible, les deux membres de l'équation peuvent être multipliés à gauche par $\begin{pmatrix} I & -WV^{-1} \\ 0 & I \end{pmatrix}$ pour obtenir le système suivant :

$$\begin{pmatrix} U - WV^{-1}W^T & 0 \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta(P) \\ \Delta(M) \end{pmatrix} = \begin{pmatrix} \epsilon(P) - WV^{-1}\epsilon(M) \\ \epsilon(M) \end{pmatrix} \quad (2.22)$$

Le système, sous cette forme, peut être résolu en deux étapes. Premièrement, $\Delta(P)$ est calculé en résolvant le système linéaire correspondant à la première ligne :

$$(U - WV^{-1}W^T)\Delta(P) = \epsilon(P) - WV^{-1}\epsilon(M) \quad (2.23)$$

Alors, le $\Delta(P)$ obtenu est utilisé pour calculer $\Delta(M)$:

$$\Delta(M) = V^{-1}(\epsilon(M) - W^T\Delta(P)) \quad (2.24)$$

Grâce à cette résolution en deux étapes, les calculs sont bien plus simples. En effet, le calcul de l'inverse de V est fait efficacement grâce à sa structure particulière. Le système de l'équation (2.23) est de même dimension que U , ce qui est beaucoup plus petit que le système original de l'équation (2.21).

En pratique, la résolution du système de l'équation (2.23) ne fait pas intervenir directement un calcul d'inverse mais une décomposition en valeurs singulières qui est plus robuste. Ainsi, la matrice $A = (U - WV^{-1}W^T)$ est décomposée en $A = U\Sigma V^T$, U et V étant orthogonales et Σ diagonale. La matrice inverse de A est obtenue par $A^{-1} = V\Sigma^{-1}U^T$.

Des orientations possibles bornées : Comme la précision de l'orientation fournie par le capteur est approximativement connue, nous définissons des limites pour les valeurs possibles des paramètres des quaternions durant les itérations. Si un quaternion sort de l'intervalle, il est ramené à la frontière afin d'assurer que les orientations des différentes images ne puissent pas trop dévier pendant l'ajustement.

Images sans observation : Certaines images ne contiennent aucun point détectés par le détecteur de Harris. Aucune observation n'est alors disponible et donc l'orientation de ces images n'a aucune influence sur la fonction de coût. L'orientation de ces images n'est donc pas modifiée pendant l'ajustement. Heureusement, les images qui ne contiennent pas de points de Harris sont souvent peu ou pas texturées, et donc de faibles erreurs d'alignement ne sont habituellement pas perceptibles.

Échelle des paramètres : Les paramètres à affiner n'ont pas la même échelle. Ainsi, les paramètres de focale ont une valeur de l'ordre de la centaine alors que les positions des points sont sur la sphère unitaire. Ceci fait que les paramètres intrinsèques ont moins tendance à changer durant l'optimisation que les autres paramètres.

Brown propose dans [43] d'utiliser une matrice de covariance C indiquant une confiance a priori et une échelle des valeurs des différents paramètres. Cette matrice est ensuite utilisée lors du calcul du pas :

$$\mathbf{d}_k = -(J(\mathbf{x}_k)^T J(\mathbf{x}_k) + \lambda_k C^{-1})^{-1} J(\mathbf{x}_k)^T r(\mathbf{x}_k) \quad (2.25)$$

Cette méthode a été testée mais n'a pas fourni de résultats probants sur nos données. Ceci peut s'expliquer par le fait que l'optimisation démarre avec un paramètre $\lambda_k = 0.001$ et que cette valeur a tendance à diminuer. Le terme C^{-1} est donc peu influent, l'optimisation suivant plutôt le schéma de Gauss Newton que celui de la descente du gradient.

Finalement, le fait que les paramètres intrinsèques évoluent peu n'est pas forcément un inconvénient car ils ont été estimés précisément lors de l'étalonnage.

Estimateur robuste : Comme il a déjà été expliqué, l'établissement des correspondances entre les images est robuste grâce à l'orientation fournie par le capteur. Il est néanmoins possible d'ajouter de la robustesse en limitant l'influence des résidus trop grands à l'aide d'une fonction de saturation $f(x)$ telle que :

$$f(x) = \begin{cases} |x|, & si |x| < x_{max} \\ x_{max}, & si |x| \geq x_{max} \end{cases} \quad (2.26)$$

2.5.3 Post Traitement

Une fois les paramètres affinés, les différentes images sont projetées sur la mosaïque avec les nouveaux paramètres. Malgré l'ajustement de faisceaux, il est très probable que les frontières des images ne se joignent pas parfaitement, par exemple à cause du fait que la caméra n'est pas exactement modélisée. Pour palier ce problème, un post-traitement est appliqué pour lisser les discontinuités aux frontières des images. Ceci permet dans un même temps de s'affranchir des problèmes photométriques, tel que le vignetage.

Pour réaliser ce traitement, le logiciel enblend [132] de Andrew Mihal a été utilisé. Ce dernier permet de générer une image unique à partir de plusieurs images alignées contenant un canal alpha indiquant pour chacun des pixels de chaque image s'il doit influencer sur l'image finale. Les mêmes pixels de plusieurs images peuvent bien évidemment devoir contribuer à la mosaïque et il est donc nécessaire de choisir un moyen de les pondérer pour obtenir un rendu satisfaisant.

Le logiciel enblend réalise cette tâche pour deux images en établissant dans un premier temps une courbe de transition entre les images. Cette courbe est utilisée pour générer un

masque binaire, renseignant pour chaque pixel laquelle des deux images doit être prépondérante.

Ensuite des pyramides multirésolutions sont générées pour les images à mixer et le masque. Ces pyramides sont obtenues en réalisant un filtrage par un Laplacien permettant de sélectionner différentes fréquences spatiales aux différents niveaux de la pyramide. Ainsi le niveau le plus haut (resp. bas) de la pyramide contient une image de dimension élevée (resp. faible) renseignant sur les hautes (resp. basses) fréquences présentes dans l'image.

Les différents niveaux des pyramides des deux images à mixer sont mélangés individuellement en utilisant un masque différent, adapté localement au contenu fréquentiel des images. Ceci explique le nom porté par ce genre de méthode, Multiband Blending, indiquant que les images sont mélangées dans des bandes de fréquences différentes.

Ensuite les différents niveaux obtenus sont recombinaés pour former l'image finale.

Le traitement de toutes les images de la mosaïque est effectué par paires d'images, en ajoutant une nouvelle image à la mosaïque existante à chaque itération.

2.5.4 Résultats

2.5.4.1 Ajustement de faisceaux

L'ajustement de faisceaux est appliqué à un sous ensemble de correspondances détectées dans les quatre images visibles sur la figure 2.18. Cette figure montre en (a) les positions des coins détectés dans les images (croix rouges) ainsi que les positions calculées pour les points correspondants sur la mosaïque (cercles bleus) obtenus à partir des différentes observations. En (a), les observations ne correspondent pas à la reprojection des points estimés sur la sphère, preuve que les images sont mal alignées.

L'ajustement de faisceaux est appliqué. Les figures 2.15 et 2.16 déjà étudiées correspondent à cet exemple. La figure 2.17 (a) montre la convergence de l'algorithme sous la forme de la valeur de la fonction de coût en fonction des itérations. La figure 2.17 (b) montre les valeurs des résidus triés avant et après l'ajustement. Les 44 premiers résidus sont nuls et correspondent à des lignes vides dans la jacobienne, soit des points pour lesquels il n'y a pas d'observation dans certaines images. Finalement la figure 2.17 (c) présente l'histogramme des résidus après l'ajustement, les résidus nuls ayant été supprimés. Cet histogramme est relativement centré autour de la classe 6. La valeur médiane correspondant à cette classe est 0, et l'estimation n'est donc pas biaisée.

La figure 2.18 (b) montre la reprojection des points après l'ajustement. Dans les images, les reprojections des points sont très proches des observations, montrant que l'ajustement de faisceaux a bien fonctionné.

2.5.4.2 Résultats sur la mosaïque

La figure 2.19 présente la mosaïque correspondante au cas étudié précédemment, et générée aux différentes étapes. En (a), l'ajustement de faisceaux n'est pas encore appliqué et la

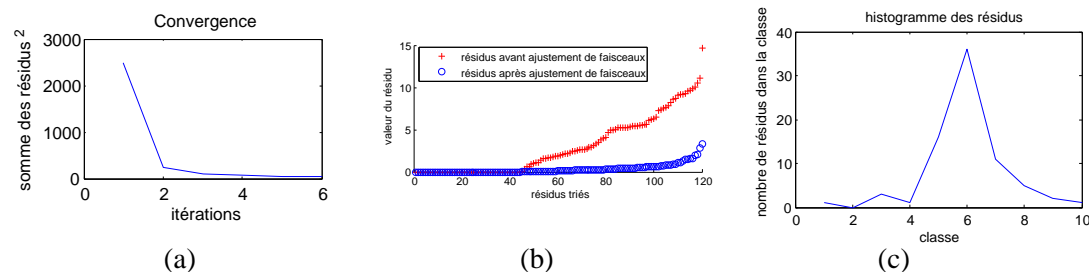


FIG. 2.17 – (a) Évolution de la valeur de la fonction de coût au cours des itérations. (b) Résidus avant et après l’ajustement de faisceaux. (c) Histogramme de la répartition des résidus après l’ajustement de faisceaux.

mosaïque est calculée uniquement à partir des données d’orientation fournies par le capteur. Plusieurs défauts apparaissent aux frontières des différentes images, par exemple dans la zone encadrée, agrandie en (b). Dans cette zone, on peut voir que l’image de gauche devrait être décalée vers la droite et le haut pour correspondre à celle de droite.

En (c), la mosaïque obtenue après l’ajustement de faisceaux est visible. Les erreurs présentes aux frontières sont largement atténuées, comme le montre le gros plan visible en (d). L’image de gauche y est cette fois correctement positionnée.

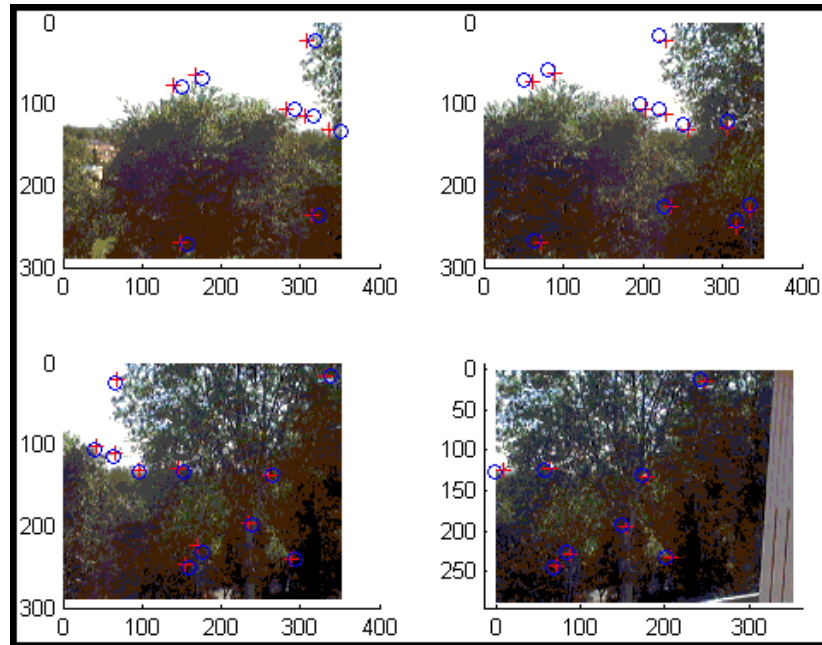
Finalement, après application du post-traitement, l’image visible en (e) est obtenue. Dans cette image, aucune frontière n’est visible, ce qui est confirmé par le gros plan (f).

2.6 Conclusions et perspectives

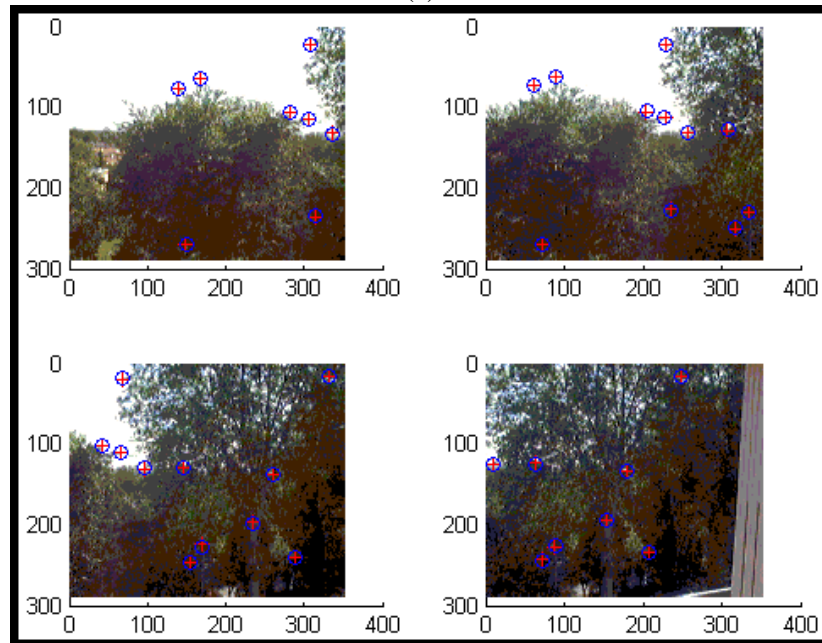
Ce chapitre a présenté une approche novatrice de la vision omnidirectionnelle grâce à l’emploi d’un capteur électronique pour l’obtention de mosaïques d’images orientées. Il a été montré que ce capteur permet de s’affranchir des problèmes faisant habituellement échouer le mosaïquage et qu’il est donc possible de considérer l’ensemble constitué de la caméra standard et du capteur d’orientation comme une caméra omnidirectionnelle à part entière. Cette caméra possède une résolution plus importante qu’une caméra catadioptrique, un encombrement et un coût moins élevé qu’un ensemble de plusieurs caméras. En contrepartie, cette caméra n’est pas capable de percevoir l’environnement de manière omnidirectionnelle instantanément. Nous avons néanmoins montré comment le capteur d’orientation aide au mosaïquage d’images des scènes mobiles, ce qui permet de résoudre partiellement ce dernier point, dans le cas où la réactivité n’est pas un critère essentiel.

2.6.1 Perspectives

La caméra présentée fonctionne correctement mais nécessite l’emploi d’un ordinateur pour permettre le mosaïquage des images. Une évolution naturelle consiste en l’intégration des traitements (au moins avant l’affinage, qui est coûteux en calcul) dans le boîtier de la caméra.



(a)



(b)

FIG. 2.18 – Quatre images (a) avant et (b) après ajustement de faisceaux. Les croix rouges (resp. cercles bleus) indiquent les positions des observations (resp. reprojection des points depuis la sphère). Sur les images de (b), ils coïncident.



FIG. 2.19 – La mosaïque générée à partir des quatre images : (a) avant l’ajustement de faisceaux, (b) gros plan sur la zone encadrée de (a), (c) après l’ajustement de faisceaux, (d) gros plan sur la zone encadrée de (c). (e) après post-traitement, (f) gros plan sur la zone encadrée de (e).

Une telle intégration, en plus de faciliter l’acquisition des mosaïques grâce à un encombrement moindre, permet dans un même temps d’étendre les performances de la caméra. Il serait ainsi possible de contrôler finement et dynamiquement la caméra, ce qui est actuellement impossible avec la Webcam USB.

Le contrôle automatique d’exposition de la caméra pourrait par exemple être utilisé, afin

d'intégrer les images à la mosaïque en tenant compte de ces paramètres, ce qui éviterait d'avoir à estimer une transformation à appliquer aux images pour qu'elles se correspondent au niveau photométrique. Il serait aussi possible, comme déjà évoqué, d'acquérir des mosaïques haute dynamique en multiplexant l'acquisition des images avec des expositions différentes.

Cette intégration permettrait aussi l'utilisation d'un zoom sur la caméra, dont la focale variable serait connue à chaque instant. Les images acquises avec des focales différentes pourraient alors être reprojctées sur la mosaïque directement. Ceci permettrait à l'utilisateur de faire des gros plans sur les zones de la mosaïque dans lesquelles il souhaite avoir une meilleure résolution tout en lui permettant de couvrir rapidement un large champ visuel dans les zones où une haute résolution n'est pas nécessaire.

Finalement, l'intégration permettrait de s'affranchir des écarts temporels entre la réception par l'ordinateur des images de la Webcam et de celle des informations d'orientation du capteur. Ce problème est actuellement réglé en imposant une vitesse de rotation faible, mais l'intégration autoriserait une vitesse de rotation plus élevée grâce au fait que les données seraient directement traitées dans la caméra.

Les travaux sur la caméra présentée dans ce chapitre ont été publiés dans [35], dont une copie est fournie en Annexe [G](#).

Chapitre 3

Caméra paracatadioptrique centrale : Détection de l'orientation et étalonnage géométrique grâce aux images de droites

Ce chapitre traite de la caméra paracatadioptrique. Dans un premier temps, en 3.1, la caméra et son modèle géométrique sont présentés. Ensuite, en 3.2, la projection des droites et des ensembles de droites parallèles est étudiée. Une nouvelle méthode robuste de détection des images de droites lorsque les paramètres de la caméra sont connus est alors proposée en 3.3. Cette méthode est ensuite utilisée en 3.4 pour détecter des ensembles de droites parallèles, à partir desquels une information d'orientation de la caméra est déduite, ce qui permet de générer des images invariantes aux rotations de la caméra par reprojection sur une surface dont l'orientation est fixe. Après avoir énuméré les méthodes d'étalonnage existantes en 3.5, une nouvelle méthode d'étalonnage de la caméra à partir d'images de droites est proposée en 3.6. Cette méthode étant particulièrement adaptée à une approche robuste, elle est finalement utilisée, en 3.7, pour simultanément détecter les images de droites et estimer les paramètres de la caméra dans le cas où les paramètres du modèle sont inconnus.

3.1 La caméra paracatadioptrique

Une caméra paracatadioptrique est composée d'un miroir paraboloidal et d'une optique télécentrique réalisant une projection orthographique vers un capteur d'image comme le montre la figure 3.1. Cette caméra possède un centre de projection, noté F , coïncidant avec le foyer du paraboloïde. En effet, soit P un point de la scène émettant de la lumière dans toutes les directions, le rayon passant par P et F , noté r_1 sur la figure, est le seul à être réfléchi parallèlement à l'axe de révolution du miroir grâce à la propriété du paraboloïde. L'optique télécentrique, dont on considère l'axe de visée parfaitement parallèle à l'axe de révolution du miroir dans tout ce

chapitre, projette sur le plan image uniquement les rayons qui arrivent parallèlement à cet axe. Ainsi les rayons réfléchis correspondant à des rayons ne passant pas par le centre optique F , tel r_2 , ne sont pas vus par la caméra, et c'est pour cette raison que la caméra est dite centrale.

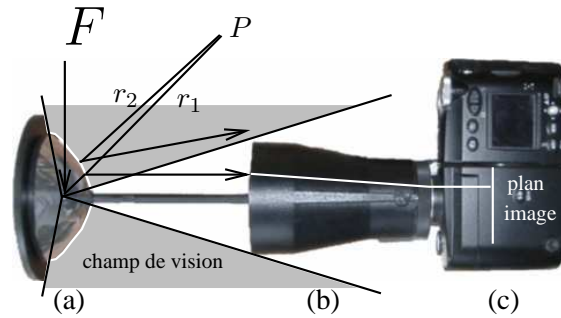


FIG. 3.1 – Une caméra paracatadioptrique. Le miroir (a) réfléchit la scène vers l'optique télécentrique (b) qui projette les rayons parallèles à son axe de visée vers le plan d'un capteur d'image (c).

La caméra paracatadioptrique est réputée plus facile à réaliser que la caméra hypercatadioptrique car seul le parallélisme entre l'axe de visée de l'optique et l'axe de révolution du miroir doit être respecté pour assurer un centre de projection unique. Ainsi, une translation de l'optique n'a aucune influence sur cette propriété. La translation de l'optique le long de son axe de visée influe par contre sur le champ de vision car celui-ci est limité d'un côté par la caméra comme montré sur la figure 3.1. La translation selon les deux autres axes doit, dans la mesure du possible, être telle que l'intégralité du miroir soit projeté dans l'image.

3.1.1 Un modèle géométrique

La figure 3.2 détaille la projection paracatadioptrique d'un point P . Soit ω_Q un repère orthonormal centré en F et dont les axes admettent pour vecteurs de base x , y et z exprimés en unités pixelliques. Le miroir est modélisé par Q_r , un paraboloïde, paramétré par h , exprimé en unités pixelliques, et dont l'axe de révolution est aligné avec Fz . Son équation est donnée par :

$$Q_r : z = -h + \frac{x^2 + y^2}{4h} \quad (3.1)$$

Attachons au plan image un repère dont l'origine est quelconque et tel que les vecteurs de base u et v soient colinéaires respectivement avec x et y . La projection orthographique de F dans la direction de z se situe sur le plan image en (u_0, v_0) . Ce plan image est appelé plan image parfait en opposition au plan image réel de la caméra dont les pixels ne sont pas forcément carrés et qui n'est pas forcément perpendiculaire à l'axe de révolution du paraboloïde.

Dans la littérature, (u_0, v_0) est souvent appelé point principal ou centre de l'image et h focale combinée. En effet, il n'est pas possible de dissocier le paramètre du paraboloïde représentant le miroir et la focale de la caméra orthographique uniquement à partir de l'image.

C'est pourquoi ces deux paramètres sont combinés pour former un paramètre du paraboloïde exprimé en unités pixelliques.

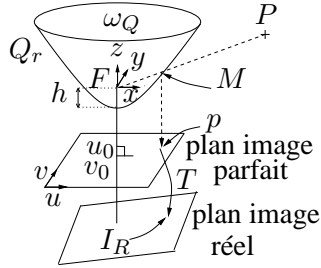


FIG. 3.2 – Projection d'un point avec la caméra paracatadioptrique.

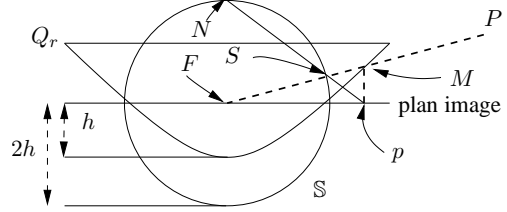


FIG. 3.3 – Équivalence entre la projection paracatadioptrique et la projection centrale sur la sphère suivie d'une projection stéréographique.

L'intersection de la demi-droite passant par F et P avec Q_r est située en M , dont les coordonnées sont données par :

$$M = k P = k \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} \in Q_r; k > 0 \quad (3.2)$$

En résolvant $M^T Q M = 0$ par rapport à k , on obtient un polynôme du second degré en k donc l'unique racine positive est :

$$k = \frac{2 h P_z + \sqrt{P_x^2 + P_y^2 + P_z^2}}{P_x^2 + P_y^2} \quad (3.3)$$

M est alors projeté orthographiquement sur le plan image parfait en un point p :

$$p = \begin{pmatrix} p_u \\ p_v \end{pmatrix} = \begin{pmatrix} u_0 + k P_x \\ v_0 + k P_y \end{pmatrix} \quad (3.4)$$

Pour prendre en compte la différence entre le plan image parfait et le plan image réel, la projection I_R de P dans le plan image réel est donnée par une transformation homographique T :

$$I_R = T p \quad (3.5)$$

Dans la suite, nous considérerons uniquement le plan image parfait, perpendiculaire à l'axe de visée et dans lequel les pixels sont carrés. Comme proposé par Geyer et al. dans [82], une rectification préliminaire peut être effectuée pour obtenir l'image dans ce plan. Cette transformation dépend uniquement de la configuration du capteur d'image et n'est pas affectée par des

translations du miroir ou des changements de focale. Geyer considère une transformation affine entre les plans images parfait et réel, correspondant à des pixels non carrés dans le plan image réel. Il propose d'estimer la rectification à effectuer en observant la projection du contour du miroir dans l'image qui est une ellipse. Soit T_2 la transformation à appliquer pour transformer cette ellipse en un cercle dans le plan image parfait, T est simplement la transformation inverse, obtenue par $T = T_2^{-1}$.

Le modèle de la caméra paracatadioptrique est donc paramétré par h, u_0, v_0 et éventuellement T .

3.1.2 Équivalence avec la projection sphérique puis stéréographique

Geyer et Daniliidis ont montré dans [80] que la projection qu'effectue une caméra paracatadioptrique est équivalente à la double projection suivante. Tout d'abord une projection à la surface d'une sphère de rayon $2h$ depuis son centre, confondu avec celui de la caméra paracatadioptrique. Elle est suivie d'une projection stéréographique (projection perspective depuis le pôle nord) vers le plan contenant l'équateur de la sphère. Ils ont montré la même équivalence pour d'autres capteurs catadioptriques à centre de projection unique avec comme seule différence que la seconde projection s'effectue depuis un autre point que le pôle nord, situé sur le segment défini par le pôle nord et le centre de la sphère.

L'équivalence entre les deux projections est visible sur l'exemple de la figure 3.3. Soit \mathbb{S} la sphère de rayon $2h$ centrée en F . Le point P est tout d'abord projeté à la surface de \mathbb{S} via F en un point S , lui-même projeté vers le plan image en p par une projection perspective de centre N situé au pôle supérieur de \mathbb{S} . p est bien la projection paracatadioptrique de P , car il est aussi la projection orthographique de M .

3.2 Projection des droites

Contrairement à la projection perspective plane, la projection paracatadioptrique ne transforme pas les droites en droites, sauf cas particuliers.

Soit D_i une droite de l'espace et π_i le plan défini par D_i et F . L'image de D_i est la projection sur le plan image de l'intersection du plan π_i avec la quadrique Q_r . La figure 3.4 montre la projection de trois types différents de segments de droites.

Le plan π_i issu d'une droite D_i est paramétré uniquement par sa normale $n_i = (a_i, b_i, c_i)^T$ car il contient F , situé à l'origine de ω_Q . L'image de D_i est généralement un arc de cercle défini par son centre (u_{ci}, v_{ci}) et son rayon r_{ci} comme par exemple D_1 qui est projetée en d_1 sur la figure 3.4. L'équation (3.7) fournit les paramètres d'un cercle dans le plan image en fonction des paramètres du plan donné par l'équation (3.6).

$$\pi_i : a_i x + b_i y + c_i z = 0 \quad (3.6)$$

$$u_{ci} = u_0 + 2h \frac{a_i}{c_i} ; v_{ci} = v_0 + 2h \frac{b_i}{c_i} ; r_{ci}^2 = \frac{4h^2 (a_i^2 + b_i^2 + c_i^2)}{c_i^2} \quad (3.7)$$

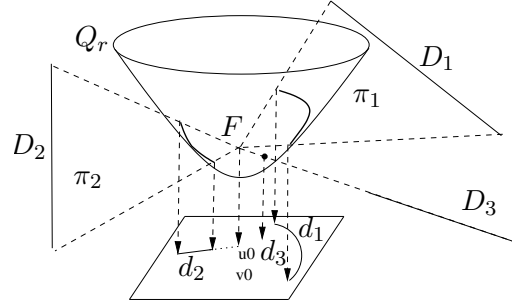


FIG. 3.4 – La projection paracatadioptrique de trois types de droites.

Le cas $c_i = 0$ pour lequel $r_{ci} \rightarrow \infty$ correspond à une D_i sécant avec Fz . Dans ce cas, la projection de D_i est un segment passant par la projection de F dans l'image. Une droite parallèle à z est projetée sur un segment dans l'image, telle D_2 qui est projetée en d_2 sur la figure 3.4.

Finalement, les D_i contenant F intersectent Q_r en un ou deux points. Une portion d'une telle droite située d'un seul côté du miroir est projetée en un point unique telle D_3 qui est projetée en d_3 sur la figure 3.4. Ce cas très particulier n'est pas considéré dans la suite pour deux raisons. Tout d'abord, il correspond à un cas très peu probable et ensuite, les droites de l'espace ne pourront de toute manière pas être identifiées à partir d'un seul point.

Nous pouvons conclure que la projection dans l'image d'un segment de droite peut être soit un arc de cercle, soit un segment, soit un point. Dans la suite, les images de droites qui sont des droites (resp. des cercles) seront dites linéaires (resp. circulaires).

3.2.1 Propriété des images de droites

Geyer a montré dans [82] que les d_i vérifient une contrainte liée au fait que le plan π_i correspondant passent par F . Le plan fronto-parallèle π_H est défini comme étant perpendiculaire à z et passant par F . Ce plan intersecte Q_r en un cercle c_H , de rayon $2h$, visible sur la figure 3.5. Ce cercle est appelé image du plan fronto-parallèle. Un plan π_i non confondu avec π_H intersecte ce cercle en deux points a et b situés aux antipodes. L'image d'une droite contenue dans π_i intersecte donc aussi c_H à ses antipodes.

Dans le cas des d_i circulaires, il est possible de formuler cette contrainte de la façon suivante. Soit r_{ci} le rayon du cercle d_i et d_{ci} la distance du centre de ce cercle au centre du cercle image du plan fronto-parallèle. On a alors :

$$4h^2 = r_{ci}^2 - d_{ci}^2 \quad (3.8)$$

La figure 3.6 fournit une interprétation géométrique de cette contrainte, dans laquelle le triangle ayant pour côtés $2h$, r_i et d_i doit être rectangle et donc doit respecter $(2h)^2 + d_{ci}^2 = r_{ci}^2$.

Dans le cas des d_i linéaires, il est facile de déduire que la droite image doit passer par le centre de c_H pour l'intersecter aux antipodes.

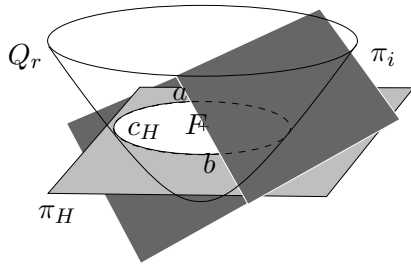


FIG. 3.5 – Propriété des images de droites : elles intersectent le cercle c_H , image du plan fronto-parallèle, en deux points (a et b) situés aux antipodes.

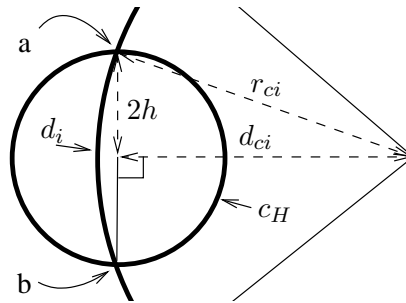


FIG. 3.6 – Interprétation géométrique de la contrainte sur les images de droites circulaires.

3.2.2 Les droites parallèles

Considérons maintenant un ensemble de droites parallèles D_i dans l'espace. Chacune définit un plan passant par F et leur ensemble définit un faisceau de plans dont l'axe, c'est à dire l'intersection commune, est située en une droite D_v . La droite D_v est parallèle aux D_i et passe par F . Il est possible de séparer les ensembles de droites parallèles en deux groupes en fonction de l'orientation de D_v .

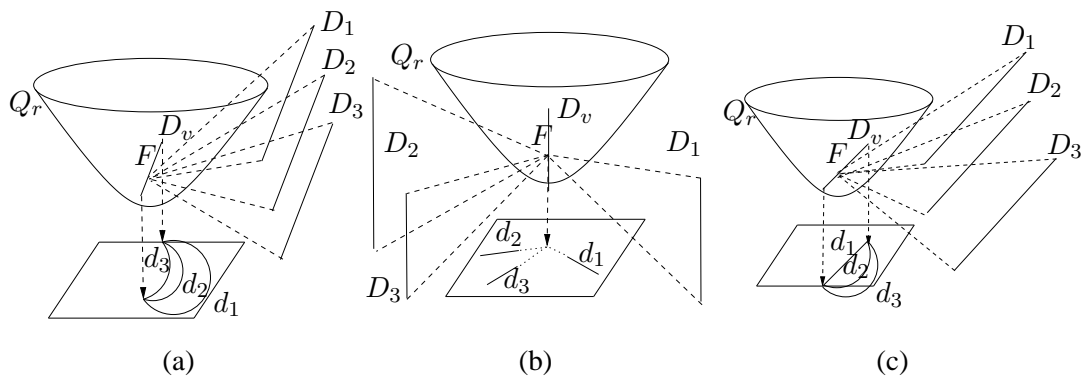


FIG. 3.7 – (a) Projection des droites parallèles D_i non parallèles à z . (b) Projection de droites D_i parallèles à z . (c) Cas particulier de la projection de droites D_i parallèles dont une est concourante avec Fz .

D_v n'est pas parallèle à z : Dans ce cas, D_v intersecte la surface du miroir Q_r en deux points qui après projection dans l'image sont les points de fuite associés à la direction D_v , en (a) sur la figure 3.7. Par conséquent, les images des droites D_i sont des arcs de cercle ayant en commun ces deux points. L'existence de deux points de fuite est due au champ de vision couvrant 360° de la caméra qui permet de voir dans les deux directions opposées.

D_v est parallèle à z : Dans ce cas, D_v intersecte physiquement Q_r en un point unique et les images des D_i sont des demi-droites passant par la projection orthographique de F sur le plan image, en (b) sur la figure 3.7.

Un cas particulier de droites parallèles : En (c) sur la figure 3.7 est montré un cas particulier de projection d'un ensemble de droites parallèles dans laquelle D_1 est concourante avec l'axe Fz . Il en résulte que d_1 est un segment joignant les deux points de fuite. Les projections de D_2 et D_3 sont des arcs de cercle passant par ces mêmes points.

3.3 Détection des images de droites avec des paramètres de la caméra connus

Nous nous intéressons ici à la détection automatique et robuste des images de droites (les d_i) dans l'image fournie par une caméra dont on connaît les paramètres du modèle.

3.3.1 Ajustement d'une image de droite sur un ensemble de points

Considérons dans un premier temps un ensemble de points $\{p_j\}_i$ observés dans l'image avec une certaine imprécision modélisée par un bruit de mesure. Ces points font partie de la même projection d_i de la droite D_i . Il est ici question de retrouver d_i à partir de $\{p_j\}_i$.

3.3.1.1 Ajustement direct dans l'image

L'ajustement peut se faire au niveau de l'image en faisant correspondre à $\{p_j\}_i$ un arc de cercle ou un segment de droite. Avec une telle approche, il est a priori nécessaire d'utiliser deux détecteurs distincts (un pour les droites et un pour les cercles) et la contrainte sur les d_i fournie par l'équation (3.8) n'étant pas imposée, n'importe quel cercle ou droite peut être détecté. C'est pourquoi il est préférable d'utiliser une méthode spécifique.

Baretto et Araujo proposent dans [133] une méthode sélective pour effectuer l'ajustement dans l'image. La projection paracatadioptrique d'un segment est détectée dans l'image comme une conique appartenant à une variété de coniques qui ne contient que les images de droites pouvant être obtenues avec une caméra donnée, dont les paramètres sont connus.

3.3.1.2 Ajustement à la surface du paraboloïde ou de la sphère équivalente

Comme il a déjà été vu, une droite D_i définit un plan π_i possédant deux degrés de liberté car il passe par F . Une méthode simple pour détecter les images de droites consiste à projeter les points sur Q_r en $\{M_j\}_i$ et à trouver un plan π_i passant par F et au mieux par l'ensemble $\{M_j\}_i$.

Fruhwith a déjà proposé dans [134] une méthode similaire pour détecter des cercles basée sur la sphère de Riemann en utilisant la projection stéréographique. Grâce à l'équivalence des projections montrée par Geyer, les calculs peuvent être effectués indifféremment sur la sphère ou le paraboloïde. Nous préférons les projeter sur le paraboloïde pour les deux raisons suivantes. Premièrement, la projection stéréographique sur la sphère concentre près du pôle nord une surface infinie du plan image. Ensuite, la formule de projection orthographique sur le paraboloïde est plus simple que la projection stéréographique.

La figure 3.8 montre la projection des points de l'image sur le paraboloïde Q_r en $\{M_j\}_i$. Un plan π_i est ensuite ajusté sur l'ensemble des $\{M_j\}_i$, avec la contrainte de passer par F .

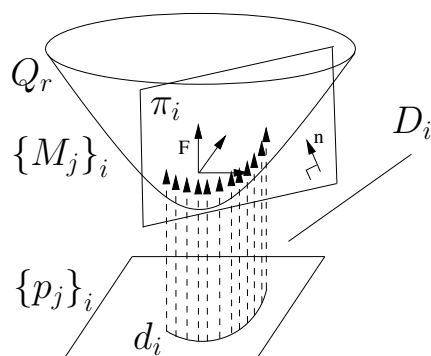


FIG. 3.8 – Projection des points $\{p_j\}_i$ appartenant à d_i à la surface du paraboloïde Q_r en $\{M_j\}_i$ puis ajustement d'un plan π_i correspondant à D_i .

En utilisant cette méthode de détection, les d_i circulaires et linéaires sont unifiées et peuvent donc être détectées de manière identique.

3.3.1.3 Précautions à prendre avant l'ajustement des plans

On souhaite obtenir l'image de droite qui minimise les résidus au niveau de l'image. Or la minimisation est effectuée dans l'espace au niveau des points sur le paraboloïde en utilisant un critère du type moindres carrés. Alors que l'on peut a priori considérer, à défaut de mieux, un bruit gaussien dont la variance est constante dans l'image (*homoscedasticité*), la projection des points sur le paraboloïde fait apparaître une amplification du bruit dépendant de la position de l'observation (*hétéroscedasticité*). Afin d'éviter un biais dans l'estimation, il est nécessaire de pondérer les points de la manière suivante pour tenir compte de la transformation avant

d'ajuster le plan.

Soit r_j la distance au centre de l'image du point p_j dont les coordonnées sont (x_j, y_j) :

$$r_j^2 = (x_j - u_0)^2 + (y_j - v_0)^2 \quad (3.9)$$

Considérons dans l'image un bruit de variance d_1 orienté dans l'axe du rayon (ce cas est le plus défavorable car alors l'amplification est maximale). Comme montré sur la figure 3.9, après projection sur le parabolöide, la distance entre $P(r_j)$ et $P(r_j + d_1)$ est supérieure à d_1 , la variance du bruit est donc amplifiée. La variance du bruit est d'autant plus amplifiée que le r_j correspondant au point est grand.

Comme d_1 est supposé petit, il est possible d'approximer localement la parabole par une droite tangente de pente $\frac{r_j}{2h}$ et d'utiliser la distance d_2 donnée par l'équation (3.10). Ceci permet d'obtenir la fonction de pondération donnée par l'équation (3.11) dont le tracé est visible sur la figure 3.10 pour $h = 1/4$.

$$d_2^2 = d_1^2 \left(\frac{r_j^2}{4h^2} + 1 \right) \quad (3.10)$$

$$\omega_j = d_1/d_2 = 1/\sqrt{\frac{r_j^2}{4h^2} + 1} \quad (3.11)$$

Les résidus e_j entre le plan ajusté défini par sa normale n_i et les points correspondants sur le parabolöide M_j sont définis par l'équation (3.12). Ils doivent être pondérés par ω_j afin d'estimer \hat{n}_i :

$$e_j = \frac{|n_i^T M_j|}{\|n_i\|} \quad (3.12)$$

$$\hat{n}_i = \widehat{(a_i b_i c_i)}^T = \arg \min_{(a_i b_i c_i)} \sum (\omega_j e_j)^2 \quad (3.13)$$

La solution du problème de minimisation correspondant à l'équation (3.13) est obtenue par décomposition en valeurs singulières de la matrice A_i construite à partir des coordonnées des points M_j pondérés d'après l'équation (3.14). Les matrices obtenues sont telles que U et V sont orthogonales et Σ est diagonale.

$$A_i = \begin{pmatrix} \omega_1 M_1^T \\ \vdots \\ \omega_n M_n^T \end{pmatrix}, A_i^T A_i = U \Sigma V^T \quad (3.14)$$

La solution au problème de minimisation équivalent donné par l'équation (3.15) est alors obtenue de la façon suivante. Les deux colonnes de V correspondant aux deux plus grandes

valeurs singulières indiquent le sous espace (le plan $\hat{\pi}_i$) dans lequel la répartition des points est la plus dispersée. La colonne de V correspondant à la plus petite valeur singulière est par conséquent orthogonale à ce sous espace et correspond donc au \hat{n}_i recherché.

$$\hat{n}_i = \arg \min_{\|n_i\|=1} \|A_i n_i\|^2 \quad (3.15)$$

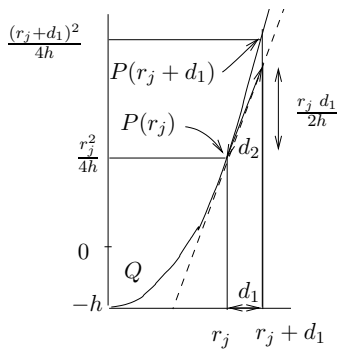


FIG. 3.9 – Amplification du bruit lors de l'élévation sur le paraboloïde Q .

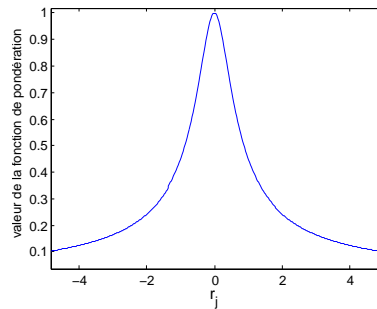


FIG. 3.10 – Fonction de pondération à appliquer aux résidus en fonction du rayon lorsque les points sont projetés sur le paraboloïde pour $h = 1/4$.

3.3.2 Détection robuste des images de droites

Une image de contour est calculée en conservant un certain nombre de points ayant le plus grand module de gradient après filtrage de l'image. Cette image est calculée à partir du canal de luminance de l'image et seulement dans la zone utile délimitée par deux cercles, correspondant au contour du miroir et au reflet de l'objectif. Cet ensemble de points de contour contient des points provenant de différents d_i ainsi que des points n'appartenant à aucun d_i . La détection robuste consiste en la détermination des différents d_i à partir de ces points de contour. Elle peut être effectuée de plusieurs manières.

3.3.2.1 Recherche dans un espace paramétrique échantillonné

Vasseur et Mouaddib proposent d'utiliser un espace paramétrique adapté à la recherche des images de droites dans [135]. Ce type d'espace paramétrique échantillonné est aussi connu sous le nom d'espace de la transformée de Hough [136].

Dans ce type d'approche, chaque point de l'image vote pour un ensemble de solutions. L'espace paramétrique est utilisé pour accumuler les votes et la (ou les) solutions retenues à la fin du traitement des points sont celles pour lesquelles la plus grande proportion de points auront voté.

Les auteurs projettent les points de l'image sur une sphère unitaire centrée sur le foyer du paraboloïde. Les plans π_i sont caractérisés par leurs normales paramétrées par une élévation et un azimut. Chaque projection d'un point sur la sphère permet de voter pour un ensemble de π_i possibles. Ces π_i forment un faisceau de plans passant par le point sur la sphère et par son centre. Leurs normales décrivent donc un grand cercle à la surface de la sphère. Les auteurs fournissent une expression très simple de la courbe à tracer dans l'espace paramétrique en fonction de la position du point sur la sphère.

Ils font cependant remarquer la difficulté de détecter correctement les pics dans un tel espace et mettent en garde contre des problèmes de sur-détection. Afin de résoudre ces problèmes, ils proposent de traiter individuellement les différents ensembles des pixels connexes plutôt que l'image entière, ce qui fait perdre de l'intérêt à cette méthode.

La figure 3.11 montre une image de contour obtenue avec la caméra paracatadioptrique et la représentation dans l'espace paramétrique de ses points de contour. On pourra noter qu'un grand nombre de points votent pour des orientations correspondantes aux pôles (ils sont sombres sur l'images (b)), sans que des pics y soient clairement détectables.

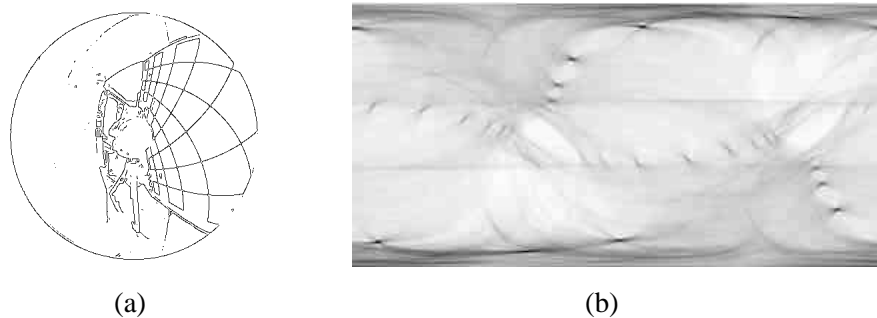


FIG. 3.11 – (a) Image de contour issue de la caméra paracatadioptrique. (b) Représentation dans l'espace paramétrique des π_i potentiels. Les lignes et colonnes correspondent respectivement à l'élévation et à l'azimut des normales aux π_i . Le score est d'autant plus élevé que le pixel est foncé.

Ying et Hu proposent dans [137] une méthode équivalente utilisant une paramétrisation légèrement différente de l'espace paramétrique.

3.3.2.2 Utilisation d'un estimateur robuste

Il existe des outils d'estimation robustes qui présentent une alternative à l'emploi d'un espace paramétrique. Ils permettent notamment d'obtenir une solution dans un espace continu, la précision du résultat n'étant pas limitée par l'échantillonnage de l'espace paramétrique. Ces outils permettent de séparer les *inliers* qui correspondent à une ou plusieurs solutions et les *outliers* qui ne correspondent à aucune solution. L'un des plus faciles à appréhender est le RANSAC, proposé par Fischler et Bolles dans [128] et dont Hartley et al. proposent des applications dans [77]. Cette méthode réalise un certain nombre de tirages aléatoires d'un nombre

minimum de candidats nécessaires au calcul d'une solution. Les tirages pour lesquels un grand nombre de points vérifient la solution sont considérés comme bons, et les points correspondant comme des inliers. Le nombre de tirages à effectuer pour garantir un certain taux de détection est fonction du pourcentage d'outliers dans les données.

Cet estimateur peut être utilisé pour la détection des images de droites. Pour cela, les différents points de contour sont tout d'abord projetés sur le paraboloïde en un ensemble de points $\{M_j\}$. Ensuite, la recherche des différents π_i est effectuée séquentiellement, l'un après l'autre. A chaque fois, un certain nombre de tirages aléatoires de couples de points M_e et M_f est effectué à partir de $\{M_j\}$. Le plan correspondant $\pi_i(e, f)$, défini par sa normale $n_i(e, f)$, est obtenu par l'équation (3.16). Un score sc_i est donné au tirage en comptant le nombre de points de $\{M_j\}$ à une certaine distance de $\pi_i(e, f)$. Le tirage avec le meilleur score est conservé et les points de $\{M_j\}$ vérifiant ce tirage sont retirés du jeu de données restant à traiter. La méthode est itérée en faisant à nouveau un tirage aléatoire jusqu'à ce qu'un certain nombre de solutions π_i soient trouvées ou que la dernière solution trouvée ne corresponde plus à suffisamment de points.

$$\pi_i(e, f) : n_i(e, f) = M_e \wedge M_f \quad (3.16)$$

3.3.2.3 Une stratégie de recherche plus efficace

La méthode précédente fournit de bons résultats mais nécessite un très grand nombre de tirages aléatoires et de tests. Il est possible, tout en conservant l'idée d'origine, de diminuer le nombre de tests à effectuer en ciblant les paires de points à tester. Les tirages étant alors connus dès la première itération, il est facile d'éliminer rapidement les tirages qui correspondent mal à un d_i et donc de continuer la recherche uniquement sur les couples potentiellement intéressants.

Un choix pour les couples à tester : Les couples de points M_e et M_f à partir desquels un plan $\pi_i(e, f)$ est testé sont appelés POIs (point d'intérêt) dans la suite.

Les différents ensembles $\{p_j\}_k$ de pixels connexes p_j sont tout d'abord récoltés depuis l'image de contour. Le squelette de chaque $\{p_j\}_k$ est calculé à l'aide d'un opérateur morphologique d'érosion. Ensuite, le plus long chemin dans le squelette de $\{p_j\}_k$ est recherché. Comme les deux extrémités du squelette sont souvent bruitées, ce sont deux points situés sur le squelette sur le chemin le plus long et à une certaine distance des extrémités qui sont utilisés comme POIs. Les expériences ont montré qu'une distance de trois pixels fournit de bon résultats.

Comme un $\{p_j\}_k$ peut correspondre à plusieurs images de droites différentes, il est nécessaire de pouvoir les séparer pour obtenir au moins autant de couples de POIs qu'il y a d'images de droites différentes dans $\{p_j\}_k$. Une décomposition hiérarchique des $\{p_j\}_k$ est donc effectuée afin d'obtenir les couples de POIs correspondants.

Une intersection de deux images de droites fait généralement apparaître un coin dans l'image en leur lieu de rencontre. Ces coins sont détectés pour séparer les différents d_i . Ils peuvent l'être par exemple en utilisant le détecteur de Harris [37]. Dans ce cas, les calculs de score du critère de Harris sont effectués uniquement dans le voisinage des pixels de contour.

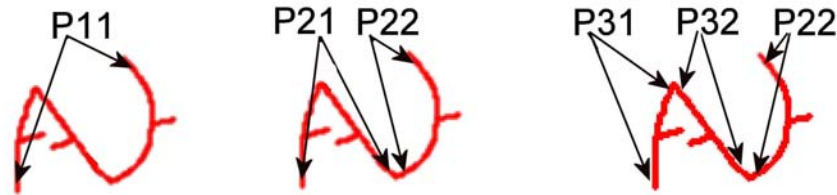


FIG. 3.12 – Décomposition d'un jeu de pixels $\{p_j\}_k$ et détection des couples de POIs aux différents niveaux.

Une autre approche donnant de bons résultats consiste à effectuer une approximation par segments du chemin le plus long dans le squelette. Le pixel correspondant à l'angle maximal entre deux segments voisins est alors retenu comme coin.

La décomposition consiste, une fois le coin localisé dans le squelette, à séparer le jeu $\{p_j\}_k$ en deux en ce point et à itérer sur les deux ensembles obtenus jusqu'à ce qu'ils soient trop petits ou qu'il n'y ait plus de point formant un coin.

Un jeu $\{p_j\}_k$ ayant un squelette plus long qu'un seuil doit être décomposé même si il ne comporte pas de coin, ceci afin de pouvoir séparer d'éventuelles d_i dont le point de rencontre n'aurait pas été détecté.

Dans le cas d'un contour fermé, les extrémités ne sont pas définies. Ce cas est assez fréquent, par exemple lorsque l'on observe l'horizon ou un objet très contrasté par rapport au fond. Dans ce cas, deux points sont aléatoirement sélectionnés lors de la première étape de décomposition, en s'assurant qu'ils sont quand même suffisamment espacés. Ensuite, la décomposition hiérarchique est effectuée normalement pour les niveaux inférieurs afin de séparer les différents d_i que peut contenir le contour.

La figure 3.12 montre trois étapes de la décomposition d'un ensemble de pixels de contour. A l'étape trois, l'ensemble $P22$ n'est pas décomposé car ceci mènerait à un ensemble trop petit. Finalement, cinq couples de POIs sont retenus pour traiter cet ensemble de pixels adjacents.

Les couples de POIs correspondant aux niveaux supérieurs de la décomposition sont conservés car rien ne garantit que la décomposition ne découpe pas un seul d_i en plusieurs parties. Or, l'estimation d'un plan π_i est d'autant plus précise que les deux POIs sont éloignés. Garder tous les niveaux de décomposition permet de détecter un d_i qui aurait été partitionné à tort aussi bien que deux d_i sécants.

La détection efficace des différents plans : Une fois les POIs extraits, tous les pixels de contour sont élevés sur Q_r . Il est ensuite nécessaire de détecter les plans π_i correspondants aux différents d_i .

L'algorithme proposé permet une détection rapide et séquentielle des différents plans π_i qui correspondent à un certain nombre de points de $\{M_j\}$. Un tableau bi-dimensionnel de N_R lignes et N_C colonnes contenant des valeurs booléennes est précalculé. N_R (resp. N_C) est égal au nombre de points M_j (resp. couples de POIs correspondants à un plan π_i). Chaque cellule

de ce tableau contient une valeur vraie (=1) seulement si la distance entre le point M_j et le plan π_i est inférieure à un seuil de distance d_{min} (en unité pixellique). Pour l'exemple, la taille du tableau est de cinq mégaoctets pour 20000 points de contour et 2000 couples de POIs.

Une fois ce tableau calculé, la détection robuste des plans est effectuée de manière très efficace simplement en comptant le nombre de valeur à 1 pour chaque couple de POIs en tant qu'un score sc_i . Le π_i ayant le score sc_i le plus élevé est retenu et les lignes correspondants à ses points sont mises à 0 dans le tableau afin que les points ne soient pris en compte qu'une seule fois. Le processus est itéré jusqu'à ce qu'il n'y ait plus de plan correspondant à un nombre minimum de points sc_{min} ou une fois qu'un nombre donné de plans a été trouvé. Afin d'éviter des tests redondants, un plan qui ne correspond pas à plus de sc_{min} points à une itération donnée ne sera pas considéré dans les suivantes car comme les points sont seulement retirés, les scores ne peuvent devenir que plus mauvais au fur et à mesure des itérations.

Les seuils sc_{min} et d_{min} sont réglés manuellement en fonction de l'application visée. Une petite valeur pour sc_{min} fournit un grand nombre de solutions mais requiert plus d'itérations, chacune d'elle étant plus longue car moins de couples de POIs auront été éliminés. d_{min} est réglé à une valeur correspondant à l'épaisseur des images de droites dans l'image de contour. Ceci permet de séparer les points provenant de différentes images de droites et de gérer les caméras qui dévient légèrement de la caméra paracatadioptrique idéale. Les M_j sont considérés comme appartenant à π_i si ils sont localisés dans un volume compris entre deux plans parallèles à une distance d_{min} de π_i . Une petite valeur de d_{min} conduit à des sur-détections car les points correspondant à des mêmes images de droites se retrouvent séparés. Une valeur élevée permet à plus de points d'être considérés comme appartenant à un π_i . Ces points sont donc retirés quand le π_i correspondant est détecté. Si ces points n'appartenaient pas effectivement au d_i considéré, ils vont biaiser un peu son estimation. Le deuxième problème qui se pose est que, comme ces points ont été retirés, l'estimation des plans suivants, qui contenaient aussi ces points est rendue plus difficile. Néanmoins, les expériences montrent qu'il est préférable de légèrement sur-estimer d_{min} pour les deux raisons suivantes. Premièrement, les points regroupés par erreur avec un π_i donné sont de toute manière très proches et ne peuvent pas biaiser beaucoup l'estimation. Deuxièmement, il reste généralement assez de points pour détecter les d_i aux itérations suivantes.

Même si un jeu de pixels adjacents est utilisé pour détecter une image de droite, ceci ne conduit pas à une sur-détection par exemple lorsqu'un d_i est coupé en deux dans l'image car les différents jeux sont utilisés uniquement pour guider la recherche des plans π_i à tester via la recherche des POIs. Les scores sc_i sont quant à eux calculés sur l'ensemble des pixels de contour, ce qui permet de regrouper des pixels appartenant à un même d_i dans le cas où ils seraient séparés dans l'image.

A la fin du processus de regroupement, un plan est estimé pour passer au mieux par les points correspondants. Ceci est effectué via un calcul du plan principal en utilisant le vecteur propre associé à la plus petite valeur propre de la matrice de covariance des points $\{M_j\}$ retenus, vecteur qui est normal au plan, comme ceci a déjà été montré avec l'équation (3.15).

3.3.2.4 Une estimation plus précise

Une fois qu'un plan π_i a été obtenu, il est possible de raffiner son estimation afin de minimiser une distance géométrique dans l'image. Ceci est fait lors d'une régression en distance orthogonale, en minimisant la distance dans l'image entre les points observés p_j et le d_i correspondant à π_i . Soient (u_j, v_j) les coordonnées du point p_j et u_{ci} , v_{ci} et r_{ci} les paramètres du cercle correspondant à d_i , donné par l'équation (3.7). Le critère à minimiser est donné par l'équation (3.17). Cette régression nécessite plusieurs itérations pour converger vers la solution.

$$\tilde{n}_i = \widetilde{(a_i b_i c_i)} = \arg \min_{(a_i b_i c_i)} \sum_j \left(\sqrt{(u_j - u_{ci})^2 + (v_j - v_{ci})^2} - r_{ci} \right)^2 \quad (3.17)$$

Néanmoins, le surcoût calculatoire est difficile à justifier, car la solution obtenue est très proche de celle fournie par la projection sur le paraboloïde, aussi bien sur des données de synthèse que sur les images réelles.

3.3.2.5 Résultats

La figure 3.13 montre la détection des d_i dans une image de contour aux itérations 1,2,3 et 29. L'étape d'affinement par régression en distance orthogonale n'est pas utilisée ici. 7241 points de contour et 424 couples de POIs ont été détectés. 29 d_i ont recouvert un nombre suffisant de p_j . Le temps de traitement total est inférieur à la seconde avec une implémentation en langage C. Des résultats plus poussés sont présentés dans la partie suivante.

3.4 Détermination de l'orientation

Une méthode permettant de détecter de manière robuste des ensembles de droites potentiellement parallèles dans la scène est présentée. A partir de cette détection, il est possible d'extraire une information d'orientation. Dans le cas d'un plan contenant au moins deux ensembles de droites parallèles, l'orientation complète du plan peut être retrouvée. Dans le cas d'une scène contenant un nombre suffisant de droites verticales, une rotation autour de deux axes peut être retrouvée, ce qui permet de générer des images qui soient invariantes aux rotations de la caméra autour de ces deux axes.

3.4.1 Regroupement des images de droites potentiellement parallèles

La recherche des images de droites parallèles peut se faire au niveau de l'image en recherchant des points de fuite à partir des d_i .

Dans le cas des caméras standard, Hartley et Zisserman, par exemple, proposent dans [77] de détecter les points de fuites et montrent diverses applications.

Dans le cas de la caméra paracatadioptrique, à cause de l'imprécision lors de l'estimation des d_i dans l'image à partir d'un petit arc de cercle et du fait que la caméra réelle dévie du

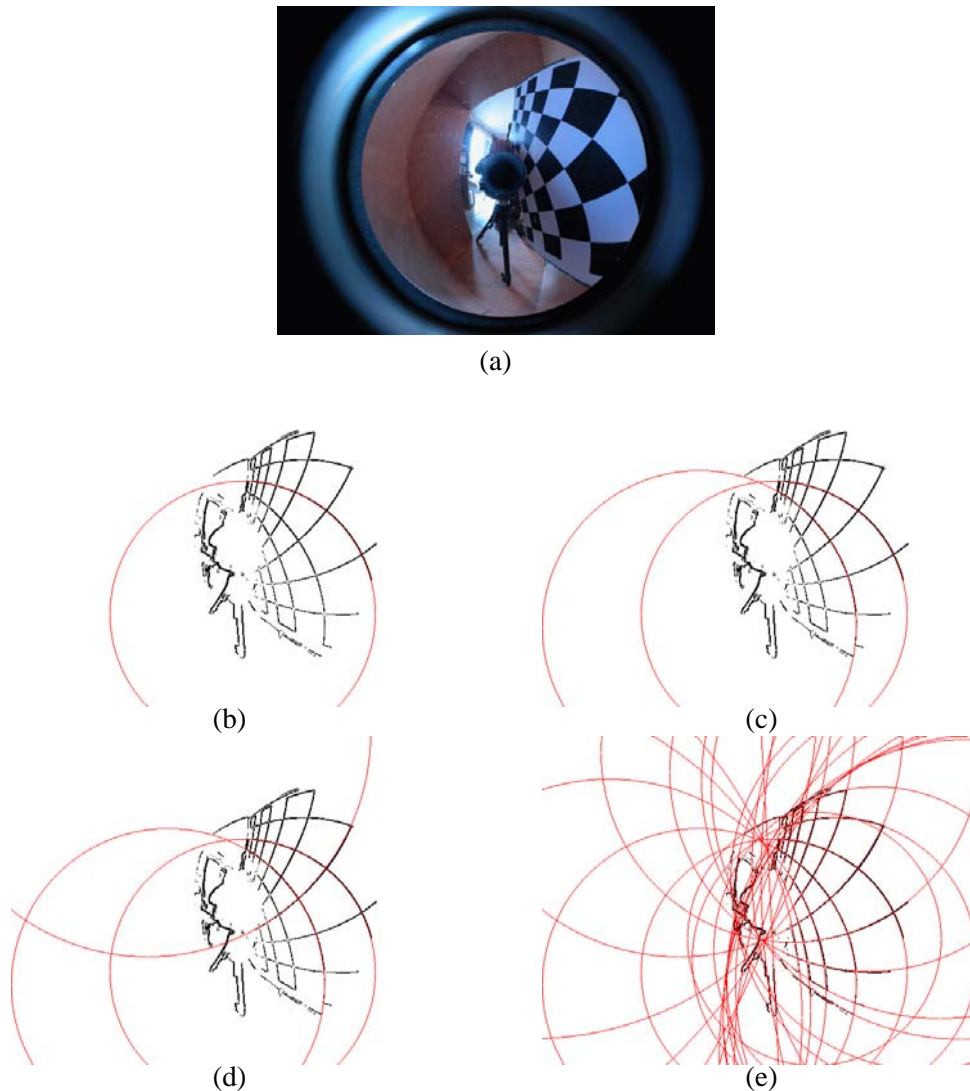


FIG. 3.13 – (a) L'image traitée et les d_i détectés aux itérations 1 (b), 2 (c), 3 (d) et 29 (e).

modèle parfait, les d_i ne s'intersectent pas parfaitement en des points de fuite bien définis. La définition d'une métrique permettant de mesurer comment plusieurs d_i s'intersectent en deux points de fuite n'est pas évidente.

Au lieu de cela, comme nous avons détecté les d_i via des plans π_i , nous proposons d'utiliser ces derniers pour détecter les ensembles de droites parallèles. Il a déjà été vu que les π_i définis par des droites D_i parallèles s'intersectent en une droite D_v qui définit un faisceau de plans et intersecte le paraboloidé en deux points de fuite. Il suffit donc de détecter les droites D_v ainsi que les plans π_i auxquels elles correspondent. En pratique, à cause du bruit dans les mesures, les plans π_i mesurés ne s'intersectent pas exactement en D_v . Il est néanmoins très

facile d'établir un critère défini sur une mesure angulaire qui permet de regrouper ces plans et d'obtenir un algorithme très simple et relativement robuste.

3.4.2 Un critère pour définir l'intersection proche de plusieurs plans

L'intersection de deux plans π_i et π_j respectivement de normales n_i et n_j est une droite de support $l(\pi_i, \pi_j)$ passant par F . Cette droite est paramétrée par le vecteur $g(\pi_i, \pi_j)$ normalisé et ramené dans l'hémisphère $z > 0$ défini par :

$$l(\pi_i, \pi_j) = n_i \wedge n_j \quad (3.18)$$

$$g(\pi_i, \pi_j) = \frac{l(\pi_i, \pi_j)}{\|l(\pi_i, \pi_j)\|} \text{sign}(l(\pi_i, \pi_j)_z) \quad (3.19)$$

Trois plans π_i, π_j et π_k ont une intersection proche d'une droite si un critère donné par une mesure de l'angle $\alpha(\pi_i, \pi_j, \pi_k)$ entre $g(\pi_i, \pi_j)$ et $g(\pi_i, \pi_k)$ est vérifié.

$$\alpha(\pi_i, \pi_j, \pi_k) = \text{acos}(g(\pi_i, \pi_j) \cdot g(\pi_i, \pi_k)) \quad (3.20)$$

Un seuil α_{seuil} est utilisé pour définir s'il y a intersection I lorsque l'angle formé entre les droites $g(\pi_i, \pi_j)$ et $g(\pi_i, \pi_k)$ est inférieur à ce seuil. Afin de tenir compte du fait que les droites sont représentées dans un unique hémisphère, il faut aussi que des droites entre lesquelles l'angle est proche de 180° soient considérées comme alignées.

$$I(i, j, k) = \begin{cases} 1 & \text{si } \alpha(\pi_i, \pi_j, \pi_k) < \alpha_{seuil} \\ 1 & \text{si } \alpha(\pi_i, \pi_j, \pi_k) > \pi - \alpha_{seuil} \\ 0 & \text{sinon} \end{cases} \quad (3.21)$$

Sur la figure 3.14, un faible écart angulaire entre $g(\pi_1, \pi_2)$, $g(\pi_1, \pi_3)$ et $g(\pi_1, \pi_4)$ permet de détecter que $\pi_1, \pi_2, \pi_3, \pi_4$ forment un ensemble ressemblant à un faisceau de plans et donc que D_1, D_2, D_3 et D_4 sont potentiellement parallèles dans l'espace. L'algorithme permettant de détecter tous les ensembles ressemblant à des faisceaux de plans est donné ci dessous :

Un score est attribué à chaque solution en sommant les scores sc_i correspondants aux plans π_i du faisceau. La meilleure solution de regroupement est donc celle contenant le plus de points p_j dans l'image.

3.4.3 Affinement de l'estimation de l'intersection

Une fois que les plans appartenant à un faisceau ont été détectés, une intersection plus précise est calculée en minimisant un critère angulaire. Soit θ et ϕ respectivement l'azimut et l'élévation de la droite $\Delta_{(\theta, \phi)}$ correspondant à la meilleure intersection des plans du faisceau considéré. Les coordonnées 3D correspondantes sont $\Delta_{(\theta, \phi)} = [\Delta_x, \Delta_y, \Delta_z]^T$ telles que $\Delta_x = (\cos(\theta) \cos(\phi))$, $\Delta_y = (\sin(\theta) \cos(\phi))$ et $\Delta_z = \sin(\phi)$. L'intersection de deux plans quelconques à partir du faisceau est utilisée comme valeur d'initialisation. Les différents plans

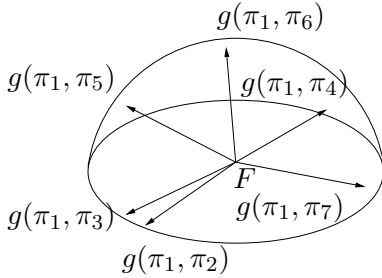


FIG. 3.14 – Angles entre les intersections des plans.

Algorithme de détection des faisceaux de plans :

```

∀i
  ∀j ≠ i
    L ← {i, j}
    compteur=2
    ∀k ≠ (i, j)
      if I(i, j, k)
        compteur=compteur+1
        L ← {L, k}
    si compteur ≥ 3
      alors il y a un faisceau de plans défini par πi
      et πj et ce faisceau contient les plans
      correspondants aux indices de la liste L

```

i sont pondérés par leur scores sc_i et $\Delta_{(\theta, \phi)}$ est calculé comme la droite la plus perpendiculaire à toutes les normales n_i aux plans pondérés en minimisant le critère suivant :

$$\widehat{(\theta, \phi)} = \arg \min_{(\theta, \phi)} \sum_i sc_i \left| \text{acos} \left(\Delta_{(\theta, \phi)}^T n_i \right) - \frac{\pi}{2} \right|^2 \quad (3.22)$$

3.4.4 Estimation de l'orientation d'un plan

Geyer et al. proposent dans [82] la projection d'un plan de la scène contenant deux ensembles de droites parallèles et observés par une caméra paracatadioptrique sur une image en projection perspective. Ils utilisent les points de fuite dans l'image après avoir effectué un étiquetage manuel des d_i .

La même opération peut être effectuée grâce à l'algorithme présenté ici mais de manière robuste. Nous utilisons une valeur de $\alpha_{seuil} = 4^\circ$ et des image de 568*426 pixels. Sur l'image visible sur la figure 3.15, l'algorithme trouve 131 combinaisons différentes à partir des 32 plans correspondants aux π_i détectés. La figure 3.15 montre, en (a) et (b), les deux orientations principales reconnues valides dans cette image. En (c), elle montre un cas d'orientation qui n'est pas valide et qui a obtenu un mauvais score. On peut voir sur cette figure un phénomène intéressant illustrant bien la difficulté de la recherche automatique des segments parallèles. Quatre segments ont été détectés comme potentiellement parallèles. Or deux d'entre eux (situés sur le damier) sont en fait perpendiculaires dans l'espace. La projection du point de rencontre des deux droites dans l'espace est interprétée comme un point de fuite. Il serait donc possible d'obtenir une image semblable à partir d'une scène dans laquelle ces deux segments seraient bien parallèles. Mais cette ambiguïté n'est présente que localement, et les assemblages de droites potentiellement parallèles qui sont invalides contiennent par conséquent peu de droites, à l'opposé des solutions valides qui en contiennent un grand nombre et ont donc généralement un bon score.

Nous montrons ici qu'il est possible de déterminer l'orientation d'une surface plane d'après

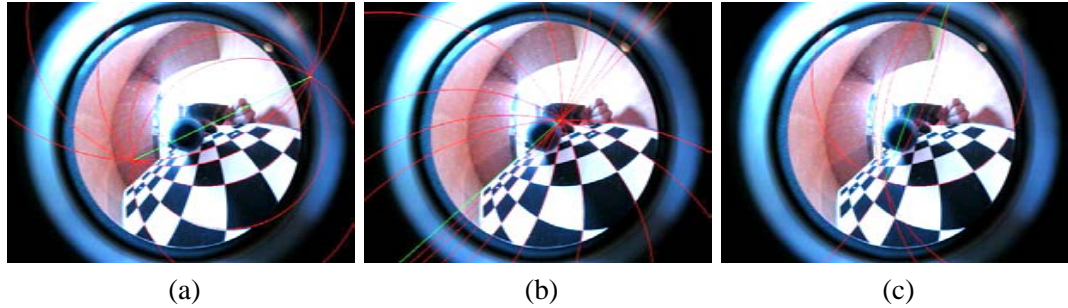


FIG. 3.15 – (a) et (b) Deux solutions de regroupement valides. (c) Solution de regroupement invalide.

l'observation de deux ensembles d'au moins deux droites parallèles. Soit D_{v1} et D_{v2} respectivement les vecteurs directeurs associés à chacun des ensembles de droites. La normale à la surface observée est donnée par $D_{v1} \wedge D_{v2}$. En échantillonnant un plan défini par cette normale selon les directions D_{v1} et D_{v2} , il est possible de générer une image semblable à celle obtenue par une caméra perspective visant perpendiculairement la surface avec ses axes de lignes et colonnes alignés avec D_{v1} et D_{v2} . La 3.16 montre deux exemples de détection de l'orientation avec reprojexion sur un plan. Les ensembles de droites parallèles détectés sont signalés en rouge et en bleu. Les segments reliant les points de fuite sur l'image sont visibles en vert pour chacune des orientations D_{v1} et D_{v2} . Dans le premier cas, D_{v1} et D_{v2} forment un angle assez grand avec l'axe de révolution du miroir. Dans le second, les segments associés à D_{v2} ont une orientation très proche de l'axe de révolution. Dans les deux cas, il est possible de déterminer l'orientation de la surface. Les deux images en projection perspective montrent l'orthogonalité des segments observés avec des erreurs de l'ordre de 2° sur les lignes en bordure de damier.

3.4.5 Estimation de l'orientation de la caméra

Comme les droites verticales sont souvent les plus nombreuses à l'intérieur des bâtiments, cette méthode peut être utilisée pour détecter leur orientation et donc en déduire l'orientation de la caméra autour de deux axes. Afin d'augmenter la robustesse de la détection des droites verticales, il faut réduire le domaine des orientations acceptables. L'orientation de la caméra pourra ainsi être restreinte dans un intervalle de valeur angulaire ($\pm 45^\circ$) autour de la verticale. Alors, il est facile d'éliminer certaines orientations candidates qui ne peuvent pas être verticales. L'orientation deux axes de la caméra est calculée comme la rotation $\widehat{R(\theta, \phi)}$ qui amène $\Delta_{(\theta, \phi)}$ en une droite verticale.

3.4.5.1 Validation de l'orientation retrouvée

Afin de valider la détection d'orientation deux axes, la caméra est montée sur une table rotative présentée au chapitre 5 entraînée par un moteur permettant un contrôle angulaire de grande précision (0.03° par pas). La méthode est appliquée aux images d'une scène contenant

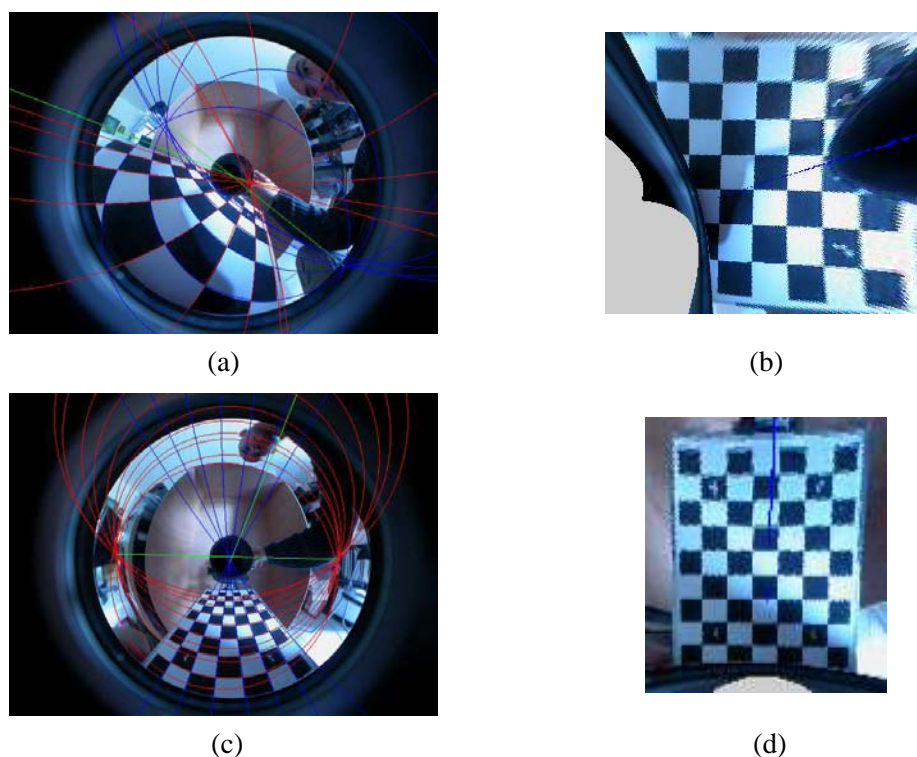


FIG. 3.16 – (a) Détection des deux orientations principales de droites dans une image quand les D_i ne sont pas parallèles à z . (b) Reprojection de l'image (a) dans un plan qui a été estimé à partir des droites. (c) Détection des deux orientations principales pour une des orientations des D_i proche de z . (d) Reprojection de l'image (c).

deux ensembles de droites parallèles disposées sur un motif 3D visible sur la figure 3.17 en (a), le paramètre α_{seuil} étant réglé à 4° . Dans cet exemple, 45 images de droites ont été détectées. Le meilleur jeu de droites en contient 23 et est montré sur la figure en (b). Les normales aux plans π_i et l'orientation $\Delta_{(\theta,\phi)}$ correspondante sont visibles en (e). Les deux points de fuite estimés dans le plan image sont visibles en gros plan en (f) et (g). Il est notable, en (g), que les d_i estimés ne s'intersectent pas bien au niveau du second point de fuite, ceci notamment car il est loin du centre de l'image. Une méthode s'appuyant sur une mesure de la distance entre les d_i et les points de fuite nécessiterait la prise en compte de ce phénomène, alors que ceci est transparent avec la formulation angulaire proposée.

La méthode est appliquée pour différents angles de rotations de la caméra et l'erreur d'estimation est tracée en fonction de l'orientation réelle en (c). Dix tests sont effectués sur différentes images pour neuf orientations différentes correspondant à des rotations de -60° à 60° par incrément de 15° . La courbe montre l'erreur maximum, minimum et moyenne. En (d), figure le même tracé pour l'autre axe en tournant le motif de $\pi/2$. Ces courbes montrent que l'estimation n'est pas biaisée et que l'erreur absolue maximale est inférieure à 2° . Il faut tout de même noter que ces résultats sont obtenus dans des conditions contrôlées où un grand nombre d'images de

droites très contrastées sont observables.

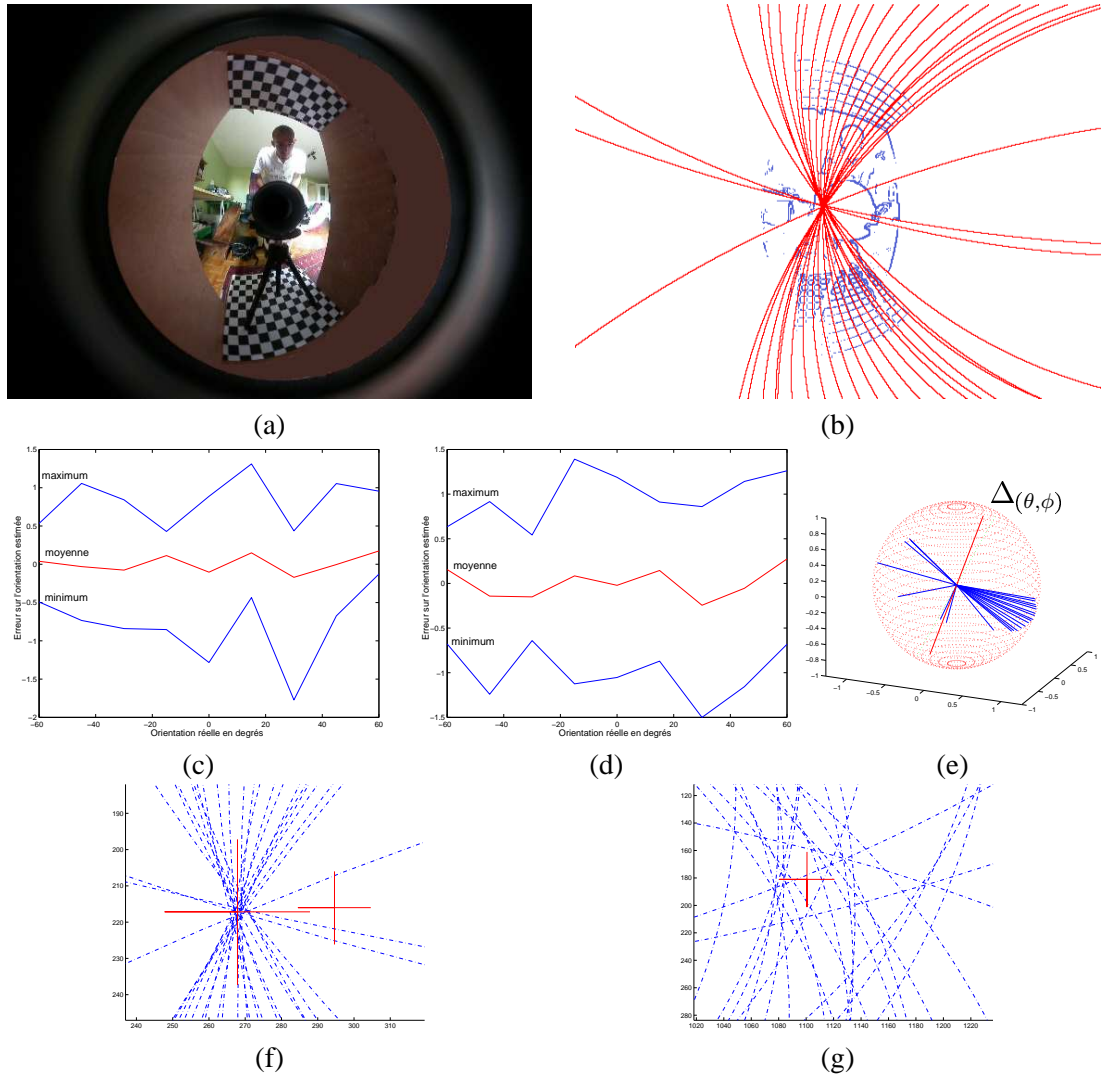


FIG. 3.17 – (a) Une image du motif 3D observé. (b) Les images de droites parallèles détectées. (c) Erreur maximum, moyenne et minimum en degrés en fonction de l'orientation réelle en degrés sur l'axe de rotation correspondant à une droite verticale dans l'image. (d) Le même tracé que (c) autour de l'autre axe de rotation. (e) Les normales aux différents plans du faisceau et l'orientation retrouvée montrée par la droite $\Delta(\theta, \phi)$. (f) Les d_i et un point de fuite (montré par la grosse croix) et le centre de l'image (petite croix). (g) Les d_i et le second point de fuite.

3.4.5.2 Rectification des images

Une fois l'orientation déterminée, il est possible de rectifier l'image pour obtenir une image correspondant à la même position et caméra, mais avec une orientation fixée autour de deux

axes. Pour cette raison, une telle image est dite invariante aux rotations de la caméra autour des deux axes considérés. La figure 3.18 (b) montre une image acquise par la caméra. En (c), la même image après rectification. La rectification consiste à reprojeter, sur le même capteur, l'image d'origine en tenant compte de la rotation qui amène le jeu de droites le plus important en position verticale. Comme visible en (a), chaque pixel (u_r, v_r) de l'image rectifiée est projeté sur le paraboloidé Q_r en P_1 avant d'effectuer une rotation pour passer du repère ω_{Q_r} à ω_{Q_i} et de le reprojeter sur le paraboloidé Q_i en P_2 puis dans l'image d'origine en (u_i, v_i) pour déterminer sa couleur.

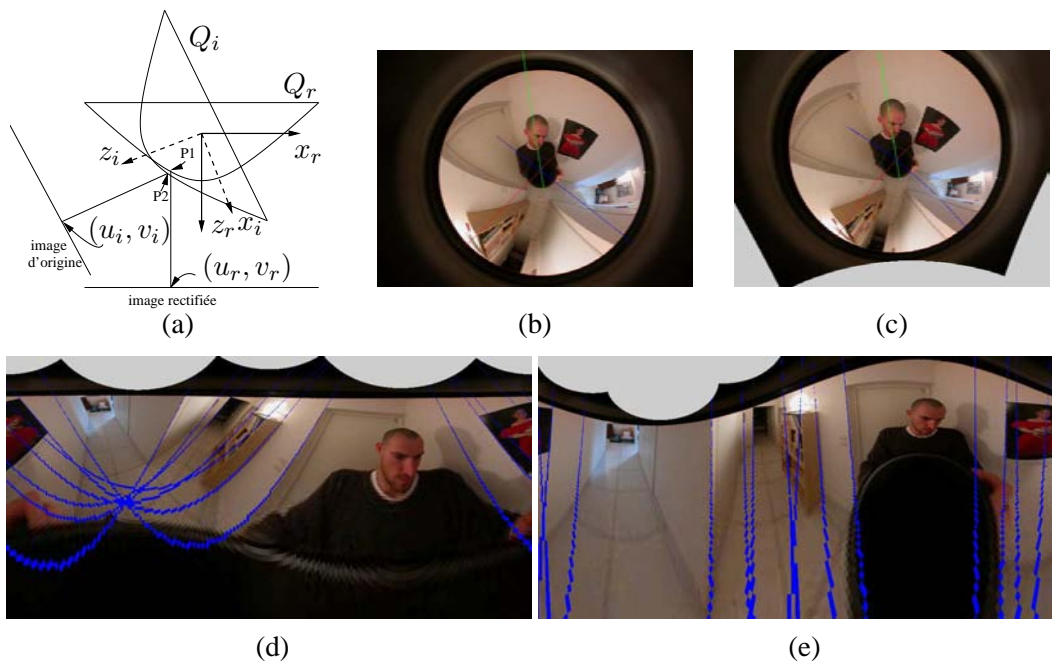


FIG. 3.18 – (a) Schéma de la reprojektion entre l'image d'origine et l'image rectifiée (voir le texte). (b) L'image d'origine. (c) L'image rectifiée. (d) Projection cylindrique de l'image d'origine. (e) Projection cylindrique de l'image rectifiée.

Ces images sont reprojettées sur des cylindres dont l'axe de révolution est aligné avec l'axe de révolution du paraboloidé. En (d), à partir de l'image d'origine, les images de droites sont des sinusoides et s'intersectent à peu près en un point visible sur l'image. Sur l'image (e), obtenue à partir de l'image rectifiée, l'axe du cylindre est aligné avec D_v et les points de fuite sont donc théoriquement situés à l'infini. Les images de droites reprojettées devraient être verticales et le sont presque (avec une erreur de l'ordre de 2°), preuve que l'orientation est correctement estimée. Une image complètement invariante aux rotations de la caméra pourrait alors être obtenue à partir de cette image en effectuant un décalage horizontal en détectant une colonne d'origine particulière, comme le proposent Pajdla et Hlavac dans [3] à partir d'une analyse de phase.

Ce traitement a été appliqué à 60 images d'intérieurs de bâtiments présentant suffisamment

de contours verticaux. Seules cinq images ont posé problème, à cause d'un carrelage ou de plaques de plafond présentant des droites parallèles. Si l'on impose que le capteur soit tenu verticalement avec une marge d'erreur de l'ordre de 20° , les erreurs présentes sur ces images n'apparaissent plus, les orientations détectées au niveau du sol ou du plafond sortant du domaine acceptable. Le même traitement a été appliqué à des scènes naturelles. Les résultats sont beaucoup moins bons mais la méthode fonctionne tout de même sur certaines images de scènes contenant des arbres qui génèrent des contours verticaux.

3.5 Méthodes existantes pour l'étalonnage

L'étalonnage de la caméra consiste ici à déterminer ses paramètres intrinsèques u_0 , v_0 et h , ce qui est équivalent à estimer le cercle image du plan fronto-parallèle. Il est possible d'étalonner la caméra paracatadioptrique de plusieurs manières différentes, comme le montre la liste de méthodes suivantes. Le code entre parenthèses, situé après le nom de l'auteur dans les titres suivants, permet un référencement plus aisé de chaque méthode par la suite.

3.5.1 La méthode de Geyer (G1) et Daniilidis utilisant les points de fuite

Geyer et Daniilidis ont proposé dans [112] une première méthode permettant l'étalonnage. Cette méthode s'appuie sur l'observation de deux jeux de droites parallèles situés dans un plan unique, chacun des jeux définissant deux points de fuite. Le centre (u_0, v_0) du cercle image du plan fronto-parallèle est obtenu en calculant l'intersection des droites reliant les points de fuite car les projections dans l'image des droites D_v sont des segments sécants au niveau de la projection de F .

Geyer utilise ensuite le fait que les points de fuite associés à des directions contenues dans un plan se trouvent sur un cercle. Ceci est dû au fait que l'intersection du plan en question avec le paraboloïde définit un cercle dans l'image après projection, exactement comme pour l'image d'une droite. Le rayon de ce cercle dépend du paramètre h . Ce paramètre est donc estimé, après avoir calculé le cercle ayant pour centre (u_0, v_0) passant au mieux par les différents points de fuite.

3.5.2 La méthode de Geyer (G2) et Daniilidis utilisant la contrainte sur les images de droites

Geyer et Daniilidis ont proposé une seconde méthode dans [82]. Celle-ci nécessite l'observation d'au moins trois droites quelconques mais qui ne soient pas toutes coplanaires. Elle utilise la contrainte sur les images de droites donnée par l'équation (3.8). La contrainte se traduit géométriquement par le fait que des sphères dont les équateurs sont les images de droites c_i s'intersectent en deux points $[u_0, v_0, 2h]$ et $[u_0, v_0, -2h]$ qui encodent les paramètres de la caméra comme visible sur la figure 3.19. Geyer fournit la méthode pour obtenir ce point en résolvant un système d'équations linéaires en minimisant une distance algébrique, dans le cas où l'on dispose de $n \geq 3$ images de droites.

Dans un premier temps, des cercles sont ajustés sur chacune des images de droites en utilisant un estimateur minimisant une erreur algébrique de manière linéaire.

Ensuite, la meilleure intersection des différentes sphères est calculée, toujours en minimisant une erreur algébrique et de manière linéaire.

Dans la suite, lorsqu'il est fait référence à la méthode de Geyer, sauf mention contraire, c'est cette méthode qui est concernée.

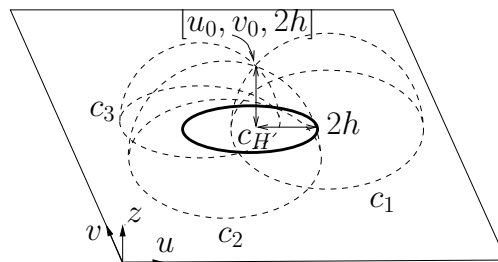


FIG. 3.19 – La méthode de Geyer et Daniilidis utilisant la contrainte sur les images de droites : Des sphères dont les équateurs sont les images de droites s'intersectent en un point qui fournit les paramètres intrinsèques de la caméra.

3.5.3 La méthode de Geyer (G3) et Daniilidis utilisant l'espace des cercles sur le paraboloid

Geyer et Daniilidis ont proposé dans [110], une méthode permettant la reconstruction 3D à partir de deux images d'une caméra paracatadioptrique non étalonnée. Cette méthode permet, grâce à la connaissance dans deux images de points homologues, d'estimer une matrice fondamentale exprimant les relations entre ces derniers. Elle n'est donc pas comparable aux méthodes déjà évoquées, qui utilisent une seule image de plusieurs droites.

Un cercle γ dans l'image est paramétré dans un espace des cercles sur le paraboloid par le point $\tilde{\gamma}$ qui est le polaire par rapport au paraboloid du plan π défini par l'élévation de ce cercle sur le paraboloid. Les auteurs montrent que les $\tilde{\gamma}$ correspondant à des images de droites se trouvent dans un plan dont le pôle par rapport au paraboloid est la conique absolue $\tilde{\omega}$. Ils définissent une matrice fondamentale de dimensions 4×4 et de rang 2. Les espaces nuls de cette matrice définissent deux droites dans l'espace, qui ne s'intersectent pas forcément, à cause du bruit dans les mesures des γ . $\tilde{\omega}$ est estimé comme étant le point équidistant aux deux droites définies par les deux espaces nuls et les paramètres de la caméra peuvent en être extraits très facilement.

3.5.4 La méthode de Geyer (G4) et Daniilidis utilisant l'espace des cercles sur la sphère

Geyer et Daniilidis ont ensuite proposé dans [109] un espace des cercles adapté à la représentation des points et des cercles en utilisant l'équivalence des projections déjà citée. Dans cet espace des cercles, les points à l'extérieur (respectivement à l'intérieur) de la sphère représentent les cercles de rayons réels (respectivement imaginaires). Les points à la surface de la sphère correspondent à des cercles de rayon nul et représentent donc les images de points. Ils ont montré que les images de droites par la caméra paracatadioptrique sont situées dans un plan comme dans le cas de la projection sur le paraboloïde. Les auteurs évoquent la possibilité d'étalonner la caméra en ajustant un plan sur trois points représentant des images de droites dans cet espace sans dire si cela a été testé et quels résultats auraient été obtenus. Le pôle de ce plan par rapport à la sphère est alors une estimée de l'image de la conique absolue. Néanmoins, rien ne garantit que ce plan, à cause du bruit dans l'estimation des images de droites γ , ne va pas intersecter la sphère, donnant lieu à un pôle hors de la sphère. Dans ce cas, l'estimée de l'image de la conique absolue serait un cercle de rayon réel, ne pouvant correspondre à des paramètres corrects.

3.5.5 La méthode de Barreto (B) et Araujo

L'algorithme de Barreto et Araujo présenté dans [138, 139] fournit une solution directe en deux étapes. En considérant, des paires d'images de droites, le point principal est tout d'abord estimé en minimisant une distance à un jeu de droites. Chacune de ces droites relie les points d'intersection de la paire de cercles. L'image de la conique absolue est ensuite ajustée à l'image des points cycliques (deux points imaginaires communs à tous les cercles) de tous les plans, dont les lignes de fuite sont les polaires du point principal par rapport aux différents cercles. Les paramètres de la caméra sont directement encodés par les éléments de l'image de la conique absolue comme pour les méthodes précédentes.

Une simplification de la méthode de Barreto et Araujo est proposée en Annexe D.

3.5.6 La méthode de Ying (Y) et Zha

Ying et Zha ont proposé dans [140] une méthode adaptée aux différentes caméras catadioptrique centrales. Cette méthode repose sur l'emploi de deux espaces paramétriques échantillonnés.

Elle permet l'étalonnage de la caméra par l'observation de trois droites ou plus. La principale différence avec les méthodes déjà citées est que celle-ci est fondamentalement robuste. Elle ne nécessite pas l'ajout d'une couche de type RANSAC [128], qui permet de tirer au sort des solutions à tester.

La méthode procède en deux étapes. Tout d'abord, un espace à trois dimensions est échantillonné pour différentes valeurs des paramètres intrinsèques u_0, v_0 et h . Ensuite pour chacune des valeurs de triplets de paramètres, un autre espace paramétrique, à deux dimensions cette fois-ci, est échantillonné. Ce second espace correspond à l'orientation des plans π qui cor-

respondraient à des images de droites sous l'hypothèse d'une caméra ayant les paramètres en cours de test. Comme nous l'avons vu précédemment, ces plans n'ont que deux degrés de liberté, car ils sont contraints de passer par F . Les auteurs utilisent le fait que pour un bon jeu de paramètres, beaucoup de points se situent au niveau des images de droites. Ils proposent une méthode pour mesurer si un jeu de paramètres est bon ou non. Pour cela, ils somment le nombre de points correspondants aux trois images de droites qui recouvrent le plus de points. Le jeu de paramètres pour lesquels cette somme est maximale est considéré comme le meilleur.

La figure 3.20 illustre ceci. On y voit une image de contour seuillée contenant des projections de droites. On s'intéresse à l'image d'une droite en particulier. Les paramètres u_0 et v_0 sont ici fixés à des valeurs correctes, par contre h varie de 400 à 590. On peut voir, sur l'image la plus à droite, que le cercle correspondant à l'image de la droite projetée correspond à un grand nombre de points de contour, contrairement aux autres cas. Il est possible d'en conclure que le paramètre h correspondant est le meilleur parmi les quatre.

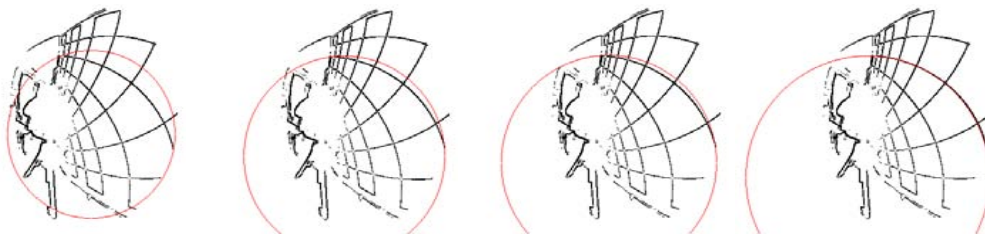


FIG. 3.20 – Différentes reprojections d'une image de droite pour différentes valeurs du paramètre h . Pour le bon jeu de paramètres, sur l'image la plus à droite, le cercle correspondant à la projection d'une droite passe par un grand nombre de points de contour.

Les intérêts majeurs de cette méthode sont qu'elle peut être adaptée à toutes les caméras catadioptriques centrales et avec une robustesse inhérente à l'emploi de l'espace paramétrique. Néanmoins, elle présente tout de même le gros inconvénient de nécessiter l'échantillonnage d'un espace paramétrique de dimension cinq, ce qui conduit à une combinatoire importante qu'il paraît difficile d'implémenter de manière efficace. De plus les questions habituellement liées aux analyses en espace paramétriques se posent, à savoir quel est le pas d'échantillonnage idéal, et dans quel intervalle est-il nécessaire d'échantillonner... On pourra quand même remarquer l'existence d'approches hiérarchiques [141] permettant d'optimiser ces paramètres.

3.5.7 D'autres méthodes

Vasseur et Mouaddib proposent dans [135] une méthode basée sur l'observation d'une mire constituée de points qui entourent la caméra en utilisant un modèle de caméra assez simple. Gonzalez-Barbosa et Lacroix dans [142], proposent un modèle de la caméra intégrant les erreurs d'alignement entre les différents éléments ainsi que les distorsions optiques de la lentille. Alors que la caméra résultante est non centrale, la caustique formée est assimilée à un point unique par approximation. Mei et Rives dans [143], propose un autre modèle considérant les

erreurs d'alignement nulles mais tenant mieux compte de la non uniformité de la résolution angulaire dans l'image lors de l'estimation des paramètres. Les deux dernières méthodes nécessitent l'emploi d'une mire plane observée dans plusieurs poses. La minimisation non linéaire d'un critère quadratique sur l'erreur de reprojection de ces points dans l'image est effectuée pour estimer les paramètres dans les différentes méthodes. Ces méthodes sont relativement coûteuses en ressources et ont tendance à fournir des minima locaux, notamment à cause du grand nombre de paramètres à estimer.

Kang explique, dans [78], comment étalonner une caméra paracatadioptrique en rotation en effectuant un suivi de points de la scène et en estimant la géométrie épipolaire.

Enfin, on peut noter la possibilité d'estimer les paramètres de la caméra par l'observation de la projection du contour du miroir dans l'image si l'on connaît parfaitement les dimensions de celui-ci. Néanmoins, cette méthode s'avère peu précise dans le cas des caméras paracatadioptriques, comme l'atteste Geyer dans [82] ainsi que nos propres expériences.

3.6 Une nouvelle méthode pour l'étalonnage à partir de droites

Une nouvelle méthode basée sur la méthode de Geyer (G2) et sur la projection à la surface d'un parabolôïde virtuel (G3) est présentée ici.

3.6.1 Motivations

Nous souhaitons une méthode permettant l'étalonnage automatique de la caméra grâce à l'observation d'une scène contenant des droites. Comme il n'est pas question ici d'étiqueter manuellement les points appartenant aux différentes images de droites, il est souhaitable que la méthode soit adaptée à une approche robuste.

La méthode de Geyer (G1) nécessite non seulement un étiquetage des points appartenant aux images des différentes droites mais en plus un second étiquetage pour les regrouper par faisceaux. De plus, elle nécessite qu'il y ait présence dans la scène de deux jeux de droites parallèles.

La méthode (G2), est plus adaptée aux contraintes de robustesse. Elle nécessite tout d'abord une détection des cercles (RANSAC sur les points de l'image par exemple), suivie ensuite d'un étalonnage à partir de trois images de droites (issues d'un second RANSAC, par exemple).

La méthode de Barreto (B), nécessite aussi la détection des images de droites dans un premier temps. Par contre, le tirage aléatoire est ensuite un peu plus critique, puisque la méthode nécessite un traitement par paire d'images de droites. Il faut donc faire attention de ne pas choisir des triplets présentant des singularités individuellement aux niveaux des paires.

Les méthodes (G3) et (G4), ne font pas appel directement à des images de droites.

La méthode de Ying (Y) semble par contre parfaitement adaptée à la contrainte de robustesse. Néanmoins sa combinatoire est réellement très grande, et la méthode proposée ici, permet d'accomplir la même tâche, dans le cas de la caméra paracatadioptrique, bien plus efficacement.

Le second avantage de la méthode proposée est qu'elle permet de gérer les images de droites linéaires, ce qui est un cas dégénéré pour les méthodes (G1), (G2) et (B), dans lesquelles elles correspondent à des cercles de rayon infini. Les droites se projetant presque en droites, donc en des cercles de rayon très grand, présentent des problèmes liés au codage de leurs paramètres en précision finie. Or ces droites sont courantes, par exemple lorsque l'axe de symétrie du miroir est vertical dans une scène d'intérieur de bâtiment remplie de segments de droites verticales. Pour tenir compte de ceci, les simulations sont effectuées non pas uniquement avec des distributions uniformes d'orientation de droites dans la scène, mais aussi avec des distributions bimodales plus proche des cas correspondant aux scènes citées.

Enfin, dernier avantage et non des moindres, il s'avère que cette nouvelle méthode fournit des résultats très satisfaisants pour l'estimation des paramètres, comme le prouvent les tests comparatifs effectués sur des données de synthèse et réelles.

3.6.2 Paramétrisation des images de droites

Soit Q_v le parabolôïde virtuel, différent de celui représentant le miroir et défini par l'équation (3.23) :

$$Q_v : z = x^2 + y^2 \quad (3.23)$$

Il est possible de reprojeter les images de droites sur Q_v par projection orthographique d'axe z .

L'équation (3.24) correspond à un cercle quelconque dans l'image :

$$(x - u_{ci})^2 + (y - v_{ci}) = r_{ci}^2 \quad (3.24)$$

Pour les images de droites circulaires, développer l'équation (3.24) et remplacer $x^2 + y^2$ par z depuis l'équation (3.23) amène à l'équation (3.25), représentant un plan dont les paramètres sont donnés par 3.26 :

$$(a_i \ b_i \ c_i \ \delta_i) (x \ y \ z \ 1)^T = 0 \quad (3.25)$$

$$(a_i \ b_i \ c_i \ \delta_i) = (-2u_{ci} \ -2v_{ci} \ 1 \ u_{ci}^2 + v_{ci}^2 - r_{ci}^2) \quad (3.26)$$

L'équation (3.27) correspond à une droite quelconque dans l'image :

$$k \ x + l \ y + m = 0 \quad (3.27)$$

Pour les images de droites linéaires, projeter la droite définie par 3.27 sur Q_v mène aussi à une équation de plan, dont les paramètres sont donnés par (3.28).

$$(a_i \ b_i \ c_i \ \delta_i) = (k \ l \ 0 \ m) \quad (3.28)$$

Chaque image de droite est donc projetée sur une courbe située dans un plan Π_i à la surface de Q_v , comme le montre la figure 3.21. Contrairement aux plans π_i du cas étalonné, les Π_i ne s'intersectent pas au niveau du foyer de Q_v .

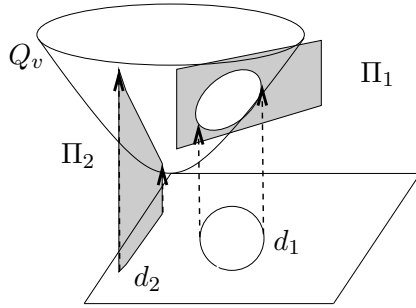


FIG. 3.21 – La projection sur le parabolôide Q_v d'un cercle (tel d_1) ou d'une droite (telle d_2) est contenue dans un plan (Π_1 ou Π_2).

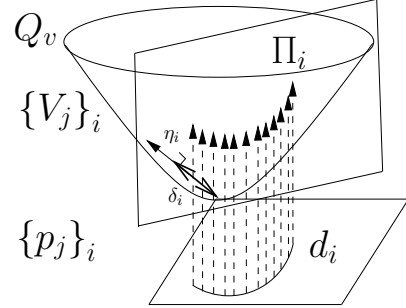


FIG. 3.22 – Élévation des points de l'image p_{j_i} sur Q_v en V_{j_i} puis ajustement d'un plan Π_i défini par sa normale η_i et sa distance à l'origine δ_i .

3.6.3 Traduction de la contrainte sur les équations de plan et étalonnage

Voyons maintenant comment la contrainte sur les images de droites est traduite au niveau de ces plans. Une image de droite circulaire satisfait l'équation (3.8) qui peut être réécrite sous la forme de l'équation (3.29) :

$$4h^2 = r_{ci}^2 - (u_{ci} - u_0)^2 - (v_{ci} - v_0)^2 \quad (3.29)$$

Une image de droite linéaire passe par (u_0, v_0) pour intersecter c_H aux antipodes et vérifie donc l'équation (3.30) :

$$k u_0 + l v_0 + m = 0 \quad (3.30)$$

Les paramètres (u_{ci}, v_{ci}, r_{ci}) du cercle correspondant à un plan $(a_i, b_i, c_i, \delta_i)$ sont donnés par l'équation (3.31) :

$$r_{ci}^2 = \frac{a_i^2 + b_i^2 - 4c_i\delta_i}{4c_i^2}, \quad u_{ci} = \frac{-a_i}{2c_i}, \quad v_{ci} = \frac{-b_i}{2c_i} \quad (3.31)$$

Remplacer u_{ci}, v_{ci} et r_{ci} dans l'équation (3.29) amène à l'équation (3.32) qui traduit le fait que tous les plans s'intersectent en un point unique, localisé en $\mathbf{x} = (u_0, v_0, u_0^2 + v_0^2 + 4h^2)^T$:

$$(a_i \ b_i \ c_i \ \delta_i) \begin{pmatrix} u_0 \\ v_0 \\ u_0^2 + v_0^2 + 4h^2 \\ 1 \end{pmatrix} = 0 \quad (3.32)$$

Une propriété intéressante est que cette équation est aussi valable pour les d_i linéaires (en insérant (3.28) dans (3.30)). c_i est dans ce cas égal à 0 car une d_i linéaire contraint uniquement la position du point principal et pas la focale combinée.

3.6.3.1 L'étalonnage

L'étalonnage consiste donc en l'estimation des plans Π_i correspondant aux d_i suivie d'un calcul de l'intersection de ces plans, qui fournit les paramètres u_0 , v_0 et h .

Normalisation et projection des points : Une étape préliminaire de normalisation est effectuée sur l'ensemble des points p_{j_i} correspondant aux d_i dans l'image. Cette étape sera détaillée dans la suite. Les p_{j_i} sont ensuite projetés sur Q_v en V_{j_i} comme le montre la figure 3.22.

Ajustement des plans : Les Π_i sont ajustés presque de la même façon que les π_i , la seule différence étant qu'il ne passent pas par l'origine. La pondération des points est effectuée comme lors de la projection sur Q_r en utilisant l'équation (3.11). Le centre Φ_i du nuage de points V_{j_i} est calculé selon l'équation (3.33) :

$$\Phi_i = \left(\sum_j \omega_j p_j \right) / \left(\sum_j \omega_j \right) \quad (3.33)$$

Ensuite, la matrice A_i est calculée (équation (3.34)) puis décomposée en valeurs singulières afin d'obtenir $\hat{\eta}_i$ comme pour la recherche des π_i :

$$A_i = \begin{pmatrix} \omega_1 (p_1 - \Phi_i) \\ \vdots \\ \omega_n (p_n - \Phi_i) \end{pmatrix} \quad (3.34)$$

δ_i est ensuite calculé simplement en imposant que le plan passe par Φ_i (équation (3.35)) :

$$\hat{\delta}_i = -\hat{\eta}_i \Phi_i \quad (3.35)$$

Calcul de l'intersection des plans dans le cas où il y a trois images de droites : Dans le cas où trois droites sont disponibles ($m = 3$), l'intersection \mathbf{x} des trois plans Π_i est obtenue en calculant $\mathbf{x} = \mathbf{N}^{-1} \mathbf{\Delta}$ à partir du système d'équations linéaires donné en (3.36) dans lequel \mathbf{N} et $\mathbf{\Delta}$ sont des concaténations de paramètres de plans.

$$\mathbf{N} = \begin{pmatrix} a_1 & b_1 & c_1 \\ \dots & \dots & \dots \\ a_m & b_m & c_m \end{pmatrix}, \mathbf{\Delta} = \begin{pmatrix} -\delta_1 \\ \dots \\ -\delta_m \end{pmatrix}, \mathbf{N}\mathbf{x} = \mathbf{\Delta} \quad (3.36)$$

Calcul de l'intersection des plans dans le cas où il y a plus de trois images de droites :

Dans le cas d'un système surcontraint, quand $m \geq 3$ droites sont disponibles, l'intersection des différents Π_i n'est pas définie car à cause du bruit dans les mesures, les Π_i ne s'intersectent pas en un point unique. Il est donc nécessaire de calculer une intersection approchée, minimisant un critère quadratique de distance aux différents Π_i . Une normalisation des différentes équations de plan telle que $\|a \ b \ c\| = 1$ permet d'exprimer la distance géométrique d'un point à ce plan par un simple produit scalaire.

Le x qui minimise la somme des carrés des distances géométriques aux différents plans normalisés est alors calculé par décomposition en valeurs singulières de $(N - \Delta)$, qui est une matrice de dimensions $(m, 4)$. Le vecteur propre associé à la plus petite valeur propre doit être deshomogénéisé grâce à une division par la valeur de sa quatrième ligne pour obtenir x . Les lignes de $(N - \Delta)$ peuvent être pondérées si les images de droites ne sont pas observées avec la même précision. Un critère, certainement non optimal, mais qui fournit de bons résultats, consiste à pondérer les différents Π_i par le nombre de points dans l'image à partir desquels ils ont été obtenus.

Dénormalisation du résultat : Une dénormalisation doit ensuite être effectuée pour obtenir les paramètres réels si les données ont été normalisées.

3.6.3.2 Existence d'une solution physiquement possible

Contrairement à Geyer dans [109] et [110], nous utilisons ici les paramètres du plan plutôt que le point polaire à la quadrique par rapport au plan considéré. Ainsi, l'intersection de trois plans est toujours définie et conduit à un point unique dès lors qu'il n'y a aucun couple de plans confondus.

Pour que cette intersection corresponde à une solution physiquement possible, il faut obtenir un paramètre h qui soit positif et réel. Comme l'intersection a lieu en $(u_0 \ v_0 \ u_0^2 + v_0^2 + 4h^2)$, ceci se traduit par le fait que l'intersection doit se situer au dessus de la courbe définissant le parabolöide Q_v .

3.6.3.3 Unicité de la solution

Pour qu'il y ait unicité de la solution à l'équation (3.36), les différentes lignes de N doivent être linéairement indépendantes, c'est à dire $\det(N) \neq 0$. Chaque ligne de N correspond à la normale à un plan, et donc les trois normales ne doivent pas être dans un même plan. Ceci se traduit sur les équations de cercle dans l'image par le fait que les centres des trois cercles ne doivent pas être alignés. Cela signifie aussi que l'étalonnage ne peut pas être effectué à partir de trois images de droites se projetant en droites, car les normales des plans correspondants sont toutes situées dans le plan $z = 0$. Ces cas correspondent généralement à des droites dans l'espace contenues dans des plans formant un unique faisceau (par exemple des droites parallèles).

3.6.3.4 Exemple

Voici un exemple d'étalonnage avec deux d_i linéaires et un circulaire pour montrer comment elles sont unifiées. Les deux droites sont définies par $-2x + y - 400 = 0$ et $2x + y = 0$. Le cercle est quant à lui défini par $(u_c, v_c) = (330, -100)$ et $r_c = \sqrt{502900}$. Le plan correspondant est $(-660 \ 200 \ 1 \ -384000)$. On obtient le système d'équations linéaires suivant :

$$\begin{bmatrix} 2 & 1 & 0 \\ -2 & 1 & 0 \\ -660 & 200 & 1 \end{bmatrix} \mathbf{x} = \begin{pmatrix} 400 \\ 0 \\ 384000 \end{pmatrix}$$

La solution de ce système est $\mathbf{x} = [100 \ 200 \ 410000]^T$, ce qui correspond aux paramètres $u_0 = 100$, $v_0 = 200$ et $h = 300$.

3.6.3.5 Normalisation des données et dénormalisation du résultat

L'idéal pour la méthode d'étalonnage serait qu'elle permette de minimiser une erreur géométrique entre les p_{j_i} et les d_i dans l'image. L'inconvénient d'une telle approche est qu'elle ne permet pas une formulation linéaire du problème. Geyer propose dans la méthode (G2), d'utiliser une distance algébrique, à la fois pour ajuster les d_i et pour estimer les paramètres ce qui, comme note Zhang dans [144], introduit un biais dans l'estimation. A l'opposé, la formulation proposée ici permet de minimiser une erreur géométrique. Malheureusement cette distance n'est pas exprimée dans l'image, mais à la surface du paraboloïde. Les précautions à prendre pour l'ajustement des plans ont déjà été dictées en 3.3.1.3.

Pour que l'erreur géométrique minimisée lors de l'intersection des Π_i soit proche d'une erreur géométrique dans l'image, il faut que Q_v soit proche de Q_r . Comme Q_r n'est pas connu a priori, il faut projeter sur un Q_v qui en soit assez proche.

Partant de l'hypothèse que les droites observées sont disposées tout autour de la caméra, il est possible de supposer que Q_r est centré par rapport aux données. De même, on peut postuler que la moitié des points sont situés de chaque côté de l'horizon, et donc se trouvent à l'intérieur du cercle fronto-parallèle.

Transformation des points : Plutôt que de changer l'expression de Q_v , il est plus simple de transformer le jeu de points pour les amener en des positions relatives par rapport à Q_v telles que celles qu'ils auraient par rapport à Q_r . Ceci est effectué grâce à un déplacement et une mise à l'échelle isotropique des p_j . Soit $p_j = [u_{pj} \ v_{pj} \ 1]^T$. Les points normalisés p_{nj} sont obtenus grâce à la transformation Γ telle que donnée par l'équation (3.37).

$$\Gamma = \begin{pmatrix} s & 0 & t_x & s \\ 0 & s & t_y & s \end{pmatrix}; p_{nj} = \begin{pmatrix} u_{pnj} \\ v_{pnj} \\ 1 \end{pmatrix} = \Gamma \begin{pmatrix} u_{pj} \\ v_{pj} \\ 1 \end{pmatrix} \quad (3.37)$$

t_x et t_y sont estimés pour amener la position moyenne des p_j en $(0, 0)$. Q_v est un paraboloïde ayant un paramètre $h = 1/4$, son cercle fronto-parallèle a donc un rayon de $1/2$. Le facteur d'échelle s doit être tel que la moitié des points translétés se trouvent à une distance inférieure à $1/2$ de l'origine. Dans les expériences effectuées, il s'est avéré que le facteur d'échelle s tel que l'écart type des points translétés soit égal à $\sqrt{2}$ fournissait un résultat encore meilleur.

Dénormalisation : Une fois l'étalonnage effectué à partir des points normalisés, les paramètres doivent être dénormalisés en appliquant la transformation inverse Γ_{inv} .

$$\Gamma_{inv} = \begin{pmatrix} 1/s & 0 & -t_x \\ 0 & 1/s & -t_y \end{pmatrix} \quad (3.38)$$

Affinages successifs : Éventuellement, il est possible d'itérer plusieurs étalonnages successifs, en utilisant les paramètres obtenus à une itération pour estimer un meilleur Γ pour l'itération suivante.

3.6.4 Résultats sur des données de synthèse

Un logiciel permettant la comparaison (Benchmark) de la méthode proposée avec celle de Geyer (G2) et de Barreto (B) a été développé en langage Matlab. Ce logiciel est mis en ligne sur Internet [145] pour permettre la comparaison avec d'autres méthodes à venir et effectuer des tests dans des conditions différentes.

3.6.4.1 Description du Benchmark

Le Benchmark est paramétrable pour permettre la comparaison dans des configurations diverses. Il permet de générer un certain nombre de points p_{j_i} correspondants à des d_i observés par une caméra paracatadioptrique. Ensuite, l'étalonnage est effectué par les différentes méthodes et les résultats obtenus sont comparés avec les paramètres réels à partir desquels les données ont été générées.

Voici la liste des paramètres réglables pour la caméra virtuelle :

- Les dimensions de l'image
- La focale combinée h
- Le centre de l'image (u_0, v_0)

Les points générés sur les d_i sont échantillonnés aléatoirement sur d_i en utilisant une distribution uniforme. Ces points sont ensuite contaminés par un bruit gaussien d'amplitude variable. Voici la liste des paramètres réglables pour la synthèse des points de d_i :

- Le nombre de d_i
- Le nombre de points échantillonnés sur chaque d_i
- L'amplitude du bruit pour les différents tests
- La façon dont les images de droites sont générées. Dans le cas d'une répartition uniforme, visible sur la figure 3.23 (a), des normales au plans π_i sont aléatoirement choisies.

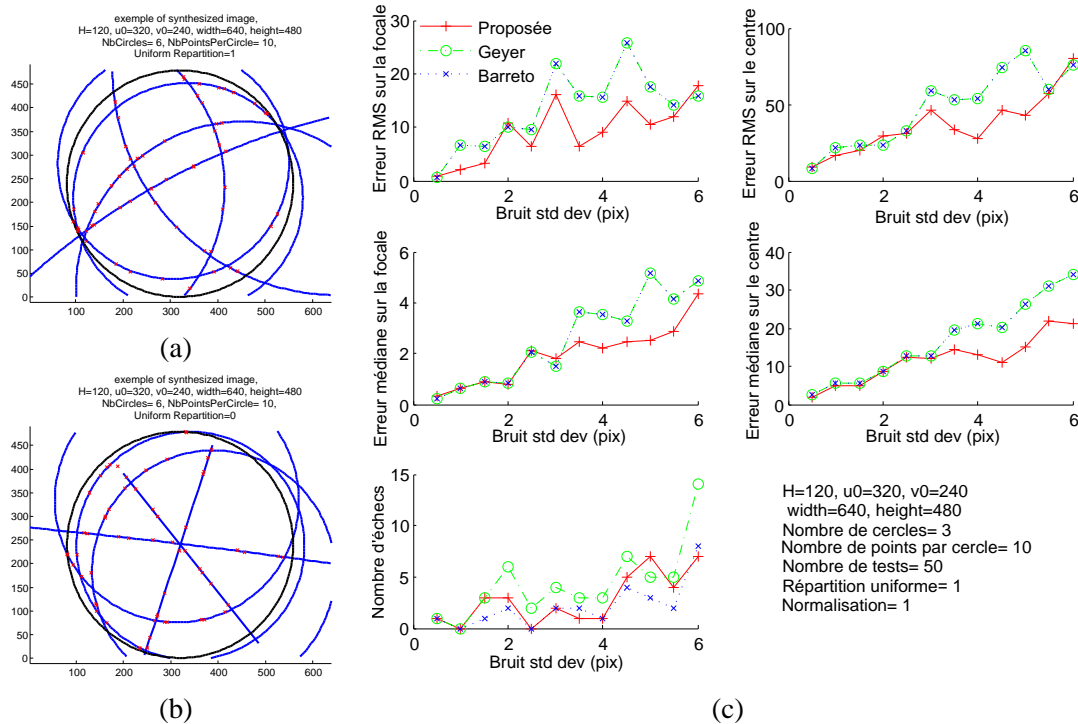


FIG. 3.23 – Exemple d’images de droites utilisées pour la simulation avec (a) une répartition uniforme, (b) une répartition non uniforme. (c) Exemple de résultats du Benchmark.

Dans le cas non uniforme, une répartition bimodale est obtenue en utilisant une distribution gaussienne pour les normales autour des axes horizontaux et verticaux, la moitié des normales appartenant à chacune de ces orientations. Cette répartition est plus proche des cas réels rencontrés, tel que celui de la figure 3.23 (b).

Enfin, différents paramètres de simulations sont configurables :

- Le nombre de tests effectués pour chaque valeur d’amplitude du bruit
- L’activation ou non de la normalisation des données d’entrée : les coordonnées des points peuvent être normalisées pour toutes les méthodes présentées de manière à obtenir une moyenne nulle et un écart type égal à $\sqrt{2}$. Ceci améliore particulièrement les résultats de la méthode proposée. En fait, cela tend à centrer le paraboloïde virtuel par rapport aux données et fait que les plans obtenus sont plus proches des vrais plans définis par le véritable paraboloïde. Cette normalisation a part contre peu d’influence sur les résultats des méthodes de Geyer et de Barreto .
- Le détecteur de cercle de Geyer ou le notre peuvent être utilisés pour la méthode de Barreto.

3.6.4.2 Résultats du Benchmark

La figure 3.23 (c) montre le résultat d'une exécution du Benchmark tel que fourni par le logiciel. Le tracé des courbes d'erreur RMS et médiane pour les différentes méthodes ainsi que le nombre de fois où l'étalonnage a complètement échoué y sont visibles, ainsi que les paramètres de la simulation.

La figure 3.24 montre des résultats du même type que ceux de [82] et [138], le champ de vue étant ici réglé à 180° soit un paramètre $h = 120$ et une hauteur d'image de $4h = 480$ pixels.

Les tests sont effectués dans les conditions suivantes : 500 étalonnages sont réalisés pour chaque niveau de bruit, de 0.5 à 6 pixels d'écart type. Les étalonnages sont effectués sur des jeux de six images de droites chacune échantillonnée en dix points. La normalisation des points est effectuée. La méthode de Barreto utilise le détecteur de cercle proposé par Geyer.

La colonne de gauche (resp. droite) de la figure 3.24 montre l'erreur RMS pour l'estimation de la focale combinée (resp. du centre de l'image, (u_0, v_0)). La nouvelle méthode est notée *Proposée* sur ces graphiques.

La première ligne montre l'erreur RMS en fonction de l'écart type du bruit gaussien appliqué dans le cas d'une répartition uniforme (UR) des normales aux plans contenant les droites.

La seconde ligne correspond à une répartition non uniforme (NUR), comportant deux orientations privilégiées.

Les cas où l'étalonnage a échoué complètement sont automatiquement détectés et éliminés et n'affectent donc pas les courbes RMS.

Dans ces deux cas expérimentaux, la méthode proposée fournit des résultats meilleurs que les deux autres, et ceci est particulièrement vérifié dans le cas d'une répartition non uniforme, grâce à la meilleure paramétrisation des d_i .

La troisième ligne montre l'influence du nombre d'images de droites observées sur l'erreur RMS des paramètres retrouvés. Seuls les résultats de la méthode proposée sont affichés pour 3, 5 et 20 images de droites, chacune comportant vingt points, en utilisant une répartition uniforme (UR). Alors que l'écart est très visible entre trois et cinq d_i , la différence est beaucoup moins marquée entre cinq et vingt. L'utilisation de cinq ou six d_i semble être un bon compromis entre qualité des résultats et coût calculatoire.

La dernière ligne montre l'erreur RMS en fonction de la focale combinée pour la méthode proposée en utilisant une répartition uniforme (UR) et un bruit gaussien d'un pixel d'écart type. Six d_i comportant chacun vingt points sont utilisés. Les résultats sont semblables à ceux de Geyer et montre un minimum d'erreur pour un certain paramètre h . L'erreur est tout de même très faible sur tout l'intervalle étudié.

Quand seulement trois images de droites sont utilisées pour l'étalonnage, les différentes méthodes fournissent approximativement les mêmes résultats, comme visible sur la figure 3.23. Dans cet exemple, la méthode de Barreto utilise le détecteur de cercle de Geyer et fournit donc les mêmes résultats. Les différences avec la méthode proposée viennent du fait que les cercles retrouvés sont différents. Dans cet exemple, dix points sont présents sur chaque d_i , et l'estimation des cercles grâce à la projection sur le paraboloïde conduit globalement à de meilleurs résultats que par la méthode de Geyer. Quand les mêmes cercles sont utilisés, les

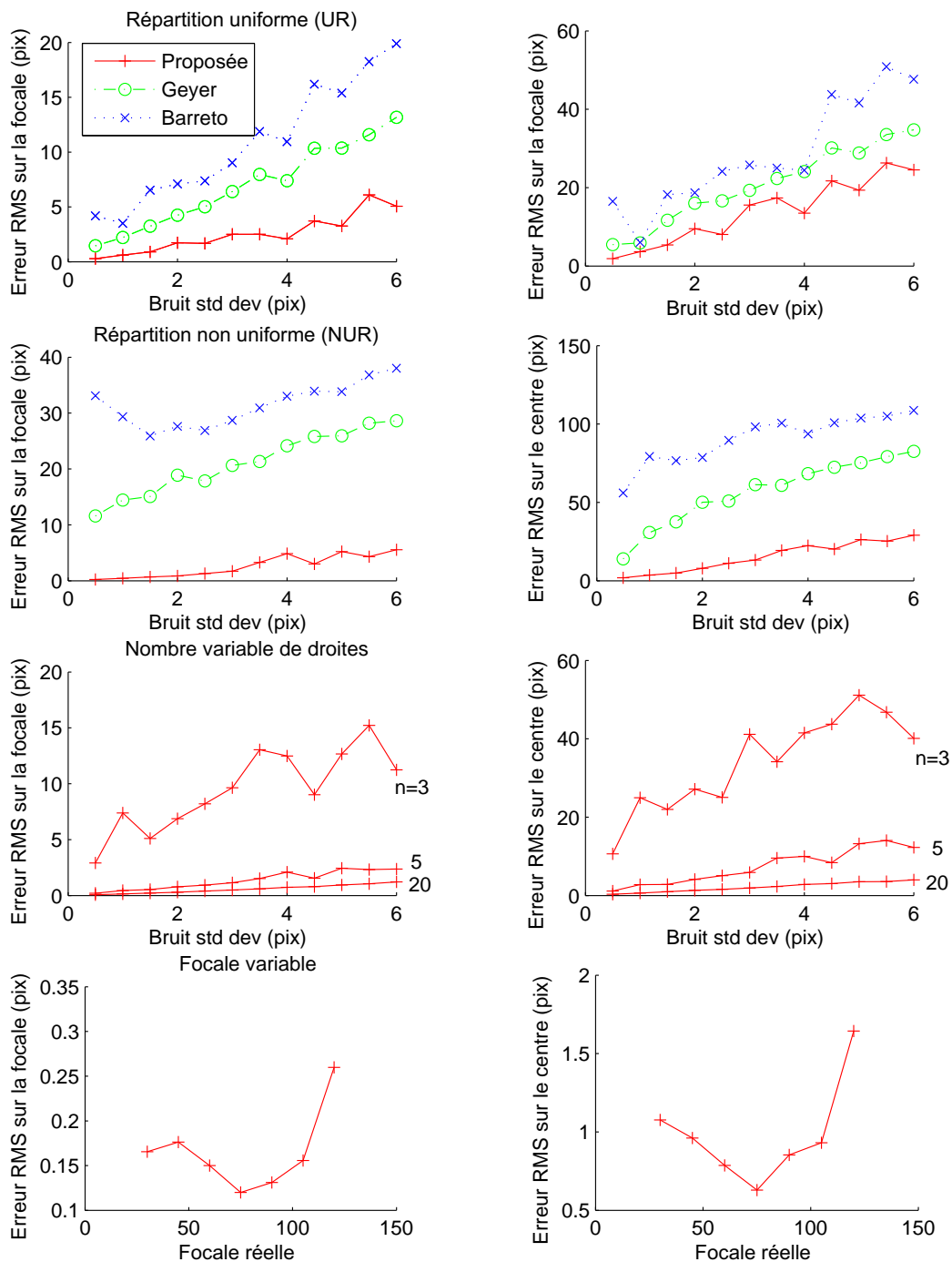


FIG. 3.24 – Erreurs RMS pour différents tests, voir le texte pour les détails.

trois méthodes fournissent évidemment exactement les mêmes résultats.

3.6.5 Résultats sur des images réelles

Il est assez difficile de fournir des résultats concernant l'estimation de paramètres de la caméra réelle, car contrairement au cas de synthèse, les paramètres exacts ne sont pas connus. Pour cette raison, les résultats fournis par les différentes méthodes sont comparés à ceux fournis par une autre méthode, dénommée ici NonLin et censée être plus précise. Cette méthode permet d'estimer itérativement les paramètres h , u_0 et v_0 qui minimisent une erreur géométrique dans l'image. Ceci est effectué en calculant, pour chaque jeu de paramètres testé, les plans π_i correspondant aux d_i . La somme des carrés des distances des p_j au d_i est ensuite minimisée en réévaluant les d_i à chaque itération.

L'image de la figure 3.25 (a) est utilisée pour la comparaison. 16 d_i sont définis dans l'image en utilisant un grand nombre de p_j et la méthode de projection sur le paraboloïde. Trois points sont extraits des d_i obtenus, l'ajustement de cercles par notre méthode ou celle de Geyer fournit alors exactement les mêmes résultats. La méthode NonLin est appliquée, en partant d'une estimée initiale fournie par chacune des trois méthodes testées et en conservant le meilleur résultat.

16 étalonnages sont effectués avec chacune des autres méthodes en utilisant tous les d_i moins un à chaque fois. Le tracé de la figure 3.25 (b) permet de voir les paramètres obtenus en fonction du d_i non pris en compte. Le tableau 3.26 montre les valeurs moyennes et les écarts types relevés pour les différentes méthodes ainsi que les résultats de la méthode NonLin. Sur ces résultats, il est possible de voir que la méthode proposée est moins sensible aux variations de choix de d_i car l'écart type est plus faible.

L'erreur géométrique moyenne dans l'image est présentée en dernière ligne du tableau 3.26. Elle représente la distance moyenne de chaque point à l'image de droite correspondante. La méthode proposée fournit les meilleurs paramètres par rapport aux deux autres méthodes sur ce critère. L'erreur géométrique a été minimisée itérativement avec la méthode NonLin, qui sert ici de référence. L'erreur géométrique faible obtenue avec la méthode proposée traduit le fait que les images de droites sont "mieux" ajustées aux ensembles de points correspondants.

Il est difficile de tirer d'autres conclusions de cette expérience, si ce n'est que les trois méthodes fournissent des résultats assez bons, surtout celle de Barreto et celle proposée.

La tendance de ces résultats a été confirmée sur une dizaine d'images, malheureusement toutes acquises avec la même caméra.

3.7 Détection robuste des droites et étalonnage

Les éléments déjà présentés peuvent être combinés pour obtenir une méthode robuste d'étalonnage à partir de droites, comme celle de Ying et Zha [140] utilisant un double espace paramétrique. Elle permet une formulation plus efficace mais est par contre spécifique aux caméras paracatadioptriques.

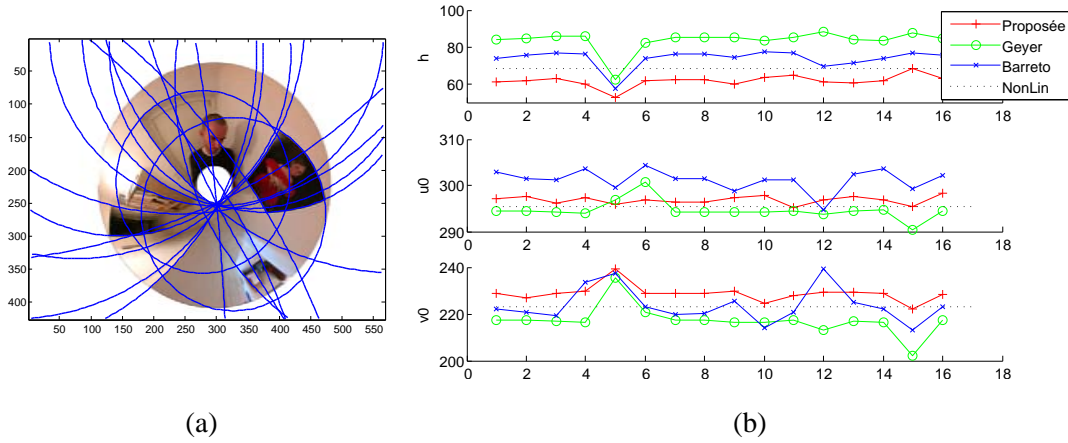


FIG. 3.25 – (a) Image utilisée pour la comparaison des méthodes. (b) Résultats de l'estimation des paramètres pour les différentes méthodes.

Param.	Proposée	Geyer	Barreto	NonLin
h	61.68 ± 3.08	83.59 ± 5.62	73.82 ± 4.70	68.12
u_0	296.84 ± 0.84	294.60 ± 1.96	301.21 ± 2.25	295.42
v_0	228.79 ± 3.35	217.19 ± 6.10	223.67 ± 7.03	223.12
Err.	1.0836	1.4289	1.3725	0.8395

FIG. 3.26 – Résultats comparatifs pour les différentes méthodes.

La tâche est décomposable en quatre étapes. La première consiste en l'estimation des cercles et droites dans l'image grâce à l'élévation sur Q_v et l'ajustement de plans Λ_i . Il faut ensuite en extraire les Π_i qui correspondent à des d_i . L'étalonnage peut ensuite être effectué sur les Π_i conservés. Enfin, les d_i sont affinés en les réestimant grâce à la connaissance des paramètres du modèle. Les deux dernières étapes ne sont pas obligatoires et sont à considérer en fonction de l'application.

3.7.1 Description de la méthode

3.7.1.1 Détection des cercles et droites

Contrairement à la détection des d_i dans le cas où les paramètres de la caméra sont connus, il n'est pas possible de projeter les p_j sur Q_r . Au lieu de cela, ils sont projetés sur Q_v . Des plans Λ_i sont ensuite détectés sur ce jeu de points projetés, pour estimer toutes les droites et cercles. Afin d'éviter d'avoir à tirer aléatoirement les p_j pouvant appartenir à un d_i , il est judicieux d'utiliser la méthode de choix des POIs déjà proposée en 3.3.2.3, en l'adaptant légèrement. Ici, trois points sont nécessaires pour calculer un plan, et ce sont donc trois POIs qui sont extraits de chaque ensemble de pixels connexes à chaque étape de la décomposition. Les deux POIs

des extrémités sont obtenus exactement de la même façon que précédemment et le POI central est simplement le pixel équidistant des deux extrémités sur le squelette.

Ensuite, un score est attribué à chaque triplet de POIs en comptant le nombre de pixels correspondants et les meilleurs triplets sont conservés en employant la même méthode que lorsque les paramètres de la caméra sont connus.

3.7.1.2 Extraction des images de droites

Parmi les plans Λ_i , uniquement les Π_i correspondant aux images de droites doivent être conservés, afin d'éliminer les autres cercles et droites. Un RANSAC est appliqué à des triplets de Λ_i pour déterminer le plus grand ensemble de plans s'intersectant approximativement en un point. Lorsqu'un triplet est tiré au sort, la distance géométrique entre l'intersection correspondante x_t et les Λ_i est obtenue par un simple produit scalaire grâce à la normalisation des normales des plans. Le triplet pour lequel le plus de plans Λ_i est à une distance donnée de x_t est conservé.

Le nombre de tirages aléatoires de triplets à effectuer dépend de la probabilité souhaitée d'obtenir un bon triplet et de la proportion d'inliers a priori ρ . Les inliers sont ici les plans Π_i et les outliers sont les autres Λ_i .

ρ doit être réglée en fonction de la scène observée, une sous-estimation étant préférable à une sur-estimation.

Quand x_t correspond à une valeur proche de x , un cercle ou une droite considéré par erreur comme un inlier ne peut pas décroître dramatiquement les performances de l'estimation des paramètres car son plan Λ_i doit de toute façon être proche de l'intersection x_t pour qu'il soit considéré comme tel.

3.7.1.3 Etalonnage sur le jeu d'images de droites retenues

Les paramètres de la caméra sont estimés précisément en utilisant tous les plans retenus pour calculer l'intersection x .

L'influence d'un plan dans l'estimation de x est pondérée par le nombre de points de contour que l'image de droite contient (voir section 3.6.3.1), afin qu'une image de droite observée sur une petite portion, et donc avec une précision moindre, soit moins influente. Ceci est plus important ici que dans le cas non robuste, car il est possible que l'extraction automatique des Π_i fournisse des d_i représentés par très peu de pixels, alors qu'une personne les étiquetant manuellement a tendance à choisir les d_i les plus importants. Plutôt que de fixer un seuil élevé pour éliminer certains d_i , il est préférable d'utiliser cette pondération.

3.7.1.4 Affinage des images de droites

Pour améliorer la précision de l'estimation des d_i , une fois les paramètres estimés, les points sont projetés sur le Q_r et les plans π_i passant par F sont calculés. Ceci permet de forcer la contrainte qui impose que les images de droites aient uniquement deux degrés de liberté.

3.7.2 Résultats

3.7.2.1 Résultats sur des données de synthèse

Premièrement, uniquement la partie géométrique est testée sur des données synthétiques (détection des plans Π_i et calcul de l'intersection \mathfrak{x}), sans utiliser l'étape d'extraction des contours et des POIs. Un RANSAC est donc nécessaire pour détecter les triplets de points menant à des Λ_i .

Des images telles que celle de la figure 3.27 (a) sont synthétisées. La taille des images est de 640*480 pixels et les paramètres de la caméra sont $h = 120$ et $(u_0, v_0) = (320, 240)$, ce qui correspond à un champ visuel de 180°. Six images de droites sont aléatoirement générées en utilisant une répartition uniforme pour les normales aux plans π_i . Trois droites quelconques ensuite ajoutées pour générer des plans Λ_i outliers. Chacune d'elles est échantillonnée en dix points aléatoires (visible comme \times pour les inliers et o pour les outliers). Finalement, dix points aléatoires uniformément répartis sont ajoutés (visible comme $+$). Un bruit gaussien de deux pixels d'écart type est ensuite ajouté à ces cent points, dont seulement soixante appartiennent à des images de droites.

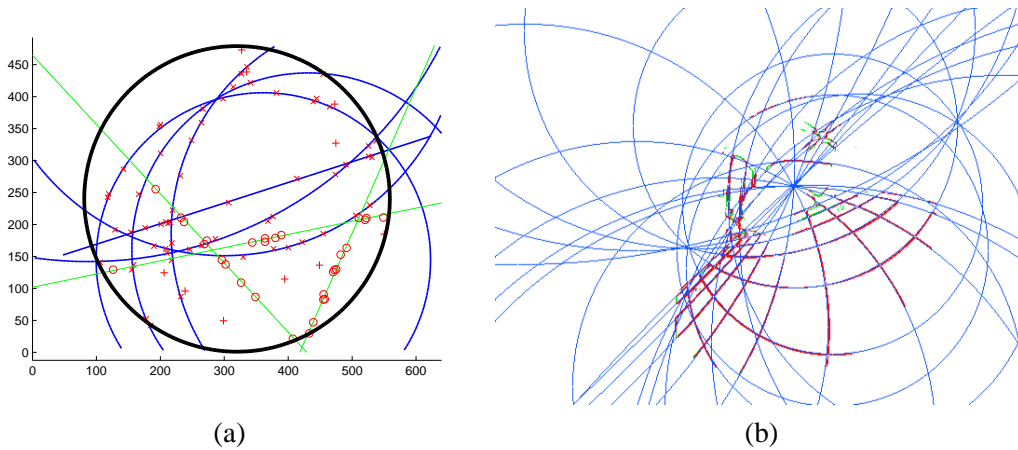


FIG. 3.27 – (a) Exemple d'image synthétisée. (b) Contours et images de droites détectés dans une image réelle.

Sur la figure 3.27 (a), une des droites est en fait une image de droite linéaire. Le cercle en gras est l'image du plan fronto parallèle qui représente ici aussi la limite du champ visuel de la caméra

sc_{min} est réglé à sept points et d_{min} à deux fois l'écart type du bruit gaussien. Le RANSAC est alors utilisé pour détecter les plans Π_i . ρ est réglé à 50%, ce qui est légèrement inférieur aux 60% réels. Le test est effectué 100 fois. Dans 87 cas, les paramètres de la caméra ont été estimés avec une erreur relative inférieure à 10% et les images de droites sont correctement estimées (Les normales des plans π_i correspondant ne dévient pas de plus de 5°). La plupart des cas d'échec proviennent d'une mauvaise classification des points, amenant à des plans Π_i qui ne

correspondent à aucune image de droites. Dans les cas réels, ceci a peu de chance d'arriver, car les plans Π_i testés correspondent à des triplets de POIs. De plus, les plans Π_i correspondant à de fausses images de droites ne contiennent généralement que très peu de points. Cela signifie qu'il y a très peu de cercles et de droites dans l'image qui ne soient pas des images de droites et qui contiennent beaucoup de pixels de contour.

3.7.2.2 Résultats sur des images réelles

Cette fois la détection est effectuée sur des images réelles, en utilisant l'étape de détection de contours et d'extraction des POIs.

La figure 3.27 (b) montre les d_i détectés sur une image de contour. Tous les d_i assez importants sont correctement détectés. L'observation des trois points de fuite visibles montre que les d_i sont assez bien estimés car ils s'intersectent relativement bien.

La méthode est appliquée à l'image montrée sur la figure 3.25 (a). Le paramètre h retrouvé diffère de 5% par rapport à celui obtenu par étiquetage manuel (65.03 au lieu de 61.68). La position du centre optique est estimée à moins de dix pixels ((304.22,222.38) au lieu de (296.84,228.79)). Ces résultats confirment que la méthode d'extraction automatique fonctionne bien car les valeurs obtenues sont dans un intervalle proche de celles obtenues par étiquetage manuel (notamment par rapport à l'écart type observé lors de la non prise en compte d'un d_i).

La robustesse de la méthode par rapport à des occultations de l'image est maintenant étudiée. La méthode est appliquée dix fois sur l'image précédente, après qu'une zone de contour de 100*100 pixels ait été effacée à une position aléatoire dans l'image. La méthode a fonctionné dans quasiment tous les cas et les paramètres retrouvés restent dans une plage de l'ordre de 5% pour h et 15 pixels pour le centre de l'image.

L'étalonnage a ensuite été effectué sur dix images différentes en utilisant la même caméra dans les mêmes conditions. Les paramètres estimés sont toujours une plage de 6% pour h et 20 pixels pour le centre de l'image. Seule une image a posé problème, les contours étant mal répartis dans l'image. Pour cette image le paramètre h a été estimé à 55.3794, ce qui représente un peu plus de 10% d'erreur. Le centre de l'image a par contre été correctement estimé.

3.7.2.3 Implémentation

Notre algorithme est implémenté en langage C et exécuté sur un ordinateur Pentium IV fonctionnant à 1.8Ghz. Le temps d'exécution pour détecter les images de droites et étalonner la caméra est inférieur à deux secondes pour une image de 568*426 pixels sans optimisation particulière du code. L'étape d'estimation du plan utilisant la SVD est remplacé par un calcul du plan principal d'inertie et l'étape de raffinement par régression en distance orthogonale n'est pas effectuée.

Le temps d'exécution de cet algorithme doit être comparé avec celui de Ying et al. [140] utilisant l'espace paramétrique. N'ayant pas implémenté complètement leur méthode et étant donné qu'ils ne fournissent pas d'indication à ce sujet, il est nécessaire d'extrapoler à partir des résultats fournis par Vasseur et al. [135]. Ceux-ci annoncent un temps de trois secondes

pour traiter une image de $768*576$ pixels dans le cas où la caméra est étalonnée. Comme Ying réalisent un grand nombre de fois ce calcul pour différents paramètres de caméras, le temps d'exécution doit être au minimum de l'ordre de plusieurs minutes.

3.8 Conclusions

Une méthode spécifique a été proposée pour la détection des projections de droites dans l'image puis a été utilisée pour estimer l'orientation de la caméra par rapport à la verticale, transformant cette dernière en inclinomètre deux axes. Une nouvelle méthode d'étalonnage géométrique utilisant des images de droites a été développée, fournissant des résultats plus précis et plus répétables que les méthodes existantes pour les tests effectués. De plus, cette méthode emploie une paramétrisation des images de droites pour laquelle il n'y a pas de configuration dégénérée. Finalement, cette méthode d'étalonnage, relativement simple à mettre en œuvre, a été utilisée de manière robuste pour permettre l'étalonnage à partir d'une image contenant des projections de droites, sans nécessiter d'étiquetage manuel.

Les travaux sur l'étalonnage géométrique ont fait l'objet d'une publication, dans [146] et la détection robuste des images de droites avec une caméra non étalonnée dans [147]. La détection de l'orientation de la caméra a été utilisée pour la localisation dans l'article [33], dont une copie est fournie en Annexe G.

Chapitre 4

Caméra catadioptrique non centrale avec miroir quadrique : Un modèle direct pour la projection des points et ses applications

Ce chapitre présente une méthode permettant le calcul de l'image d'un point observé par une caméra catadioptrique avec un miroir en forme de quadrique, ce qui correspond au modèle direct de la caméra. Le chapitre est organisé de la façon suivante :

Tout d'abord, en 4.1, la problématique, les applications, les méthodes existantes et les apports de la nouvelle méthode sont présentées.

La section 4.2 propose une paramétrisation utilisant un ellipsoïde ou un paraboloidé de révolution pour la réflexion respectivement des points à coordonnées finies et des points à l'infini.

Ensuite, en 4.3, les cas simples où la réflexion peut être recherchée dans le plan sont traités.

La méthode est étendue en 3D pour les autres quadriques, en 4.4. Les problèmes d'occlusion, le fait que le miroir ne soit qu'une portion de quadrique, l'existence de réflexions multiples ainsi que les considérations sur la précision des calculs numériques sont discutés.

Des résultats pour les différentes quadriques les plus utiles sont présentés en 4.5, pour les points finis et ceux situés à l'infini. Le cas des caméras axiales est étudié.

La section 4.6 présente des résultats sur les applications en vision par ordinateur. Après avoir montré comment calculer l'image d'un point, le calcul d'une courbe de correspondance stéréo est effectué en projetant une demi-droite échantillonnée, y compris dans le cas où plusieurs réflexions sont visibles. Des Splines sont utilisées pour approximer les projections de droites et accélérer les calculs. La reprojection d'une texture plane est aussi étudiée. Finalement, la méthode est utilisée pour l'étalonnage d'une caméra catadioptrique non centrale afin de minimiser une erreur de reprojection.

Finalement, la section 4.7 conclue le chapitre et propose des perspectives, notamment pour

finaliser l'étalonnage géométrique.

4.1 Motivations et définition du problème

Certains agencements particuliers de caméras et de miroirs permettent d'obtenir un centre de projection unique, si des contraintes strictes sont respectées, comme présenté au [chapitre 1](#). Néanmoins, considérer une caméra comme étant centrale est généralement une approximation, car les contraintes ne sont respectées qu'à une certaine échelle. Dès lors, il est nécessaire, afin de mieux modéliser la caméra, d'utiliser un modèle non central. Un tel modèle permet alors de considérer une famille bien plus large de caméras, en s'affranchissant des contraintes censées permettre la réalisation d'une caméra centrale.

Si l'utilisation de modèles non centraux requiert des traitements plus complexes, il faut noter que les caméras non centrales présentent quelques avantages. Par exemple, il est possible de détecter la position d'une droite (à une transformation près) dans l'espace à partir d'une seule image comme le montrent Caglioti et al. dans [148].

Des tâches telles que la reconstruction 3D de l'environnement peuvent aussi être effectuées avec ces caméras, comme le proposent par exemple Fiala et Basu, dans [149], avec une caméra stéréoscopique utilisant un modèle simple.

4.1.1 Modèles directs et inverses de caméra

4.1.1.1 Le modèle direct d'une caméra

Le modèle direct d'une caméra permet de calculer l'image p d'un point P de la scène. Il est plus ou moins facile à déterminer en fonction du caractère de la caméra. Dans certains cas, il est possible d'avoir plusieurs images pour un même point P .

Trou d'épingle : L'image p d'un point P observé par une caméra en trou d'épingle est facile à obtenir. Comme le montre la figure 4.1 (a), la caméra placée en C ne perçoit que le rayon émis par P passant par C , et p est donc située à l'intersection de ce rayon avec le plan image de la caméra.

Caméra catadioptrique centrale : Dans le cas d'une caméra catadioptrique centrale, les rayons doivent passer par le centre optique de la caméra catadioptrique pour être perçus. Ainsi, dans l'exemple de la figure 4.1 (b), une caméra placée au niveau d'un des foyers d'un miroir ellipsoïdal permet d'avoir un centre de projection unique situé au niveau du second foyer F de l'ellipsoïde. La caméra ne perçoit donc que les rayons passant par F , qui est donc le centre optique de la caméra catadioptrique. La réflexion de ces rayons est triviale à calculer et par conséquent p aussi, obtenu à partir du rayon réfléchi.

Caméra catadioptrique non centrale : Dans le cas d'une caméra non centrale, les rayons avant réflexion ne sont pas contraints de passer par un point unique. Dans l'exemple de la

figure 4.1 (c), deux rayons r_1 et r_2 sont par exemple réfléchis vers le centre optique C de la caméra. La difficulté réside alors en la détermination du ou des rayons qui sont réfléchis vers C . Une fois ces rayons calculés, le calcul de la ou des images de P peut être effectué comme précédemment, en considérant le ou les rayons réfléchis. Nous parlerons de *réflexions multiples* quand un point de la scène possède plus d'une image.

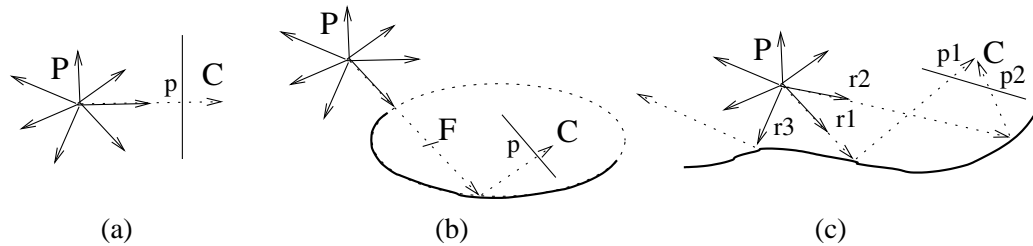


FIG. 4.1 – Le point P vu par une caméra placée en C (a) dans le cas d'une caméra standard, (b) dans le cas d'une caméra catadioptrique centrale, (c) dans le cas d'une réflexion sur une surface quelconque.

4.1.1.2 Le modèle inverse d'une caméra

Le modèle inverse d'une caméra permet de calculer le rayon de projection associé à l'image d'un point.

Le modèle inverse d'une caméra catadioptrique avec un miroir quadrique est relativement simple à définir, comme le montre par exemple Mikusic dans [89]. Connaissant l'image p observée par une caméra située en un point C , le rayon de projection associé l_p après réflexion dans un miroir s'obtient par *lancer de rayon direct* : Un rayon primaire est émis depuis C dans la direction définie par C et p puis est réfléchi à la surface S du miroir en χ dans la direction l_p (voir figure 4.2).

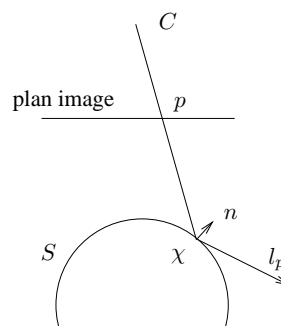


FIG. 4.2 – Le rayon l_p obtenu après réflexion sur la surface S à partir du point p observé par la caméra située en C .

4.1.2 Intérêts et applications du modèle direct de caméra

Le modèle direct d'une caméra permet de nombreuses applications, dont certaines seront décrites plus en détail en section 4.6.

4.1.2.1 Applications en vision

Étalonnage géométrique : Grâce à un modèle direct de la caméra, il est possible de minimiser une erreur de reprojection dans l'image plutôt qu'une distance dans l'espace entre des rayons de projection et des points 3D comme le propose Mikusic dans [89]. La minimisation dans l'espace présente des inconvénients dès lors que les points observés ne sont pas tous à la même distance de la caméra, l'échelle des résidus à minimiser étant alors fonction de la distance des points observés. A l'opposé, l'emploi d'une métrique dans l'image ne présente pas ce problème.

Stéréovision : Le modèle direct d'une caméra permet la projection de rayons de projection, qui permettent de contraindre la zone de recherche des pixels homologues sur des courbes de correspondances stéréoscopiques dans le cadre de la stéréovision. Ceci aide à la reconstruction 3D, comme le fait la géométrie épipolaire dans le cas des caméras centrales.

Reprojection d'images : Le modèle direct d'une caméra peut aussi être utilisé pour échantillonner une surface de l'espace et connaître la projection des échantillons dans l'image, afin de reconstruire la texture de la surface.

4.1.2.2 Applications en synthèse d'images

Un modèle direct trouve aussi des applications dans le domaine de la synthèse d'images, comme proposé par Glaeser dans [150]. Il convient alors de considérer la caméra catadioptrique constituée de la surface réfléchissante à rendre et de la caméra virtuelle pour laquelle l'image doit être rendue.

Rendu d'image par lancer de rayon inverse : Le modèle de caméra permet de calculer directement les reflets des points de la scène sur la surface réfléchissante afin de projeter la scène vers l'image. En utilisant un maillage triangulaire assez fin pour représenter la scène, il est possible d'approximer son image en projetant les coins des triangles et en interpolant linéairement entre ces points, ce que font très efficacement les cartes graphiques accélératrices 3D.

Meilleur rendu du miroir : Alors qu'une approche couramment utilisée consiste en l'approximation du miroir par un ensemble de facettes planes, dans lesquelles la réflexion est simple à calculer, un modèle direct de la caméra intégrant un modèle sans discontinuité du miroir permet un rendu bien plus réaliste.

Rendu de spécularité : En calculant la réflexion d'une source lumineuse ponctuelle puis en effectuant un maillage adapté de la surface réfléchissante (des sommets du maillage doivent être situés à l'endroit de la réflexion de la source lumineuse), il est aussi possible d'obtenir un rendu très rapide de spécularité par interpolation de type Gouraud Shading.

4.1.3 Méthodes existantes pour la détermination du modèle direct de caméra catadioptrique non centrale

Le modèle direct de la caméra catadioptrique permet de prédire la position de p (ou de χ) correspondant à un point P , après réflexion sur S pour un point C fixé. Il correspond au *lancer de rayon inverse*.

Il n'existe pas de solution directe a priori à la détermination de p pour la réflexion sur une surface quelconque. Il est néanmoins possible d'obtenir p à partir du modèle inverse de la caméra, par les deux méthodes suivantes.

Il faut noter que lors du calcul de distance entre rayon de projection et point de la scène, il faut tenir compte du sens du rayon de projection, en utilisant la géométrie projective orientée, présentée par exemple dans [79] par Laveau et al.

4.1.3.1 Modèle inverse et minimisation de distance en espace continu

Micusik et al. dans [89, 52] proposent, pour les caméras catadioptriques, de déterminer l'image d'un point P en minimisant la distance entre un rayon de projection l_p et le point 3D P par lequel il est censé passer (voir figure 4.3 (a)). Ainsi, partant d'une solution initiale p' , un rayon l'_p est calculé. Ensuite la position de p' est ajustée itérativement dans l'image pour minimiser la distance $d(l'_p, P)$ entre l'_p et P , afin de tendre vers p lorsque $d(l_p, P) = 0$.

Cette méthode n'est pas optimale pour plusieurs raisons :

- Tout d'abord, elle nécessite la connaissance d'une valeur p' assez proche de p , obtenue par Micusik grâce à un modèle de caméra centrale approximant la caméra non centrale, qui n'est pas toujours disponible. Toutes les caméras ne peuvent donc pas être utilisées avec cette méthode.
- Ensuite, elle requiert l'emploi de méthodes d'optimisation itératives, dont la convergence n'est pas assurée.
- Finalement, cette méthode ne permet pas la prise en compte simultanée des réflexions multiples.

La réflexion d'un point situé à l'infini ne peut pas non plus être obtenue directement avec cette méthode, la distance d'un rayon l_p à un tel point n'étant pas définie. Il est alors nécessaire de formuler le problème comme la minimisation de l'angle entre l_p et la direction définie par le point à l'infini.

4.1.3.2 Modèle inverse et minimisation de distance en espace discret

Fabrizio, dans [21] propose d'utiliser le lancer de rayon direct pour déterminer des courbes de correspondances stéréoscopiques. Exploitant le fait que l'observation d'un point dans une

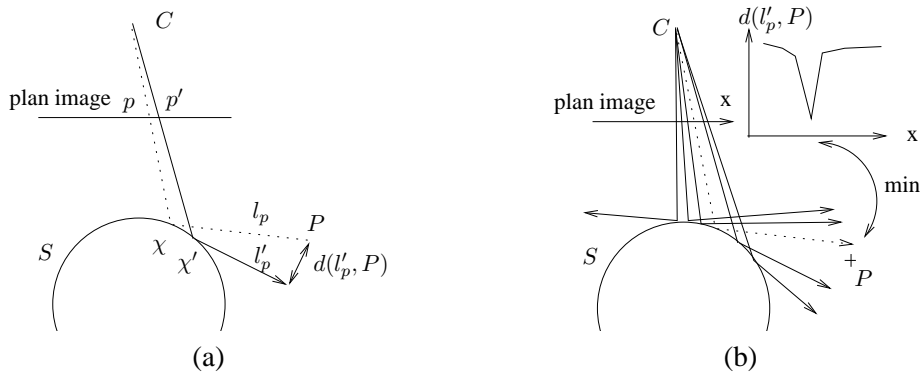


FIG. 4.3 – (a) L'image p d'un point P est calculée par minimisation de la distance entre le rayon de projection l'_p et le point P . (b) Échantillonnage du plan image et recherche du minimum de distance.

image A contraint sa position sur une demi droite d (le rayon de projection associé au point dont on recherche l'homologue), il cherche à calculer la projection de d dans l'image B d'une caméra catadioptrique non centrale.

Ne disposant pas de méthode pour projeter individuellement des points de d , il mesure la distance dans l'espace de d aux rayons de projection correspondants à chacun des pixels de l'image B et stocke le résultat dans une image de distance. En recherchant une vallée dans cette image, il peut ainsi déterminer les pixels correspondants à la projection de d .

Cette méthode, effectuant une minimisation dans l'espace discret de l'image de distance, présente les inconvénients suivants :

- Elle est relativement lourde puisqu'elle nécessite un traitement pour chacun des pixels de l'image B .
- De plus, la courbe obtenue est limitée en précision à l'échelle du pixel, à moins de sur-échantillonner l'image B , ce qui alourdirait encore les calculs.

Cette approche peut être aisément utilisée pour calculer la projection d'un unique point, en stockant dans l'image la distance des différents rayons de projection au point à projeter, comme représenté sur la figure 4.3 (b). Il faut alors rechercher un puits dans l'image de distance. Cette approche permet alors de gérer les réflexions multiples, en recherchant plusieurs puits dans l'image. Elle ne nécessite pas d'initialisation par un modèle central approché et peut donc être utilisée sur tout type de caméra catadioptrique, mais nécessite des calculs **très** lourds pour un seul point.

4.1.4 Nouvelle approche et ses apports

Les deux approches qui viennent d'être présentées sont clairement non optimales.

Une solution pour les miroirs sphériques et cylindriques : Glaeser a proposé dans [150] une solution directe pour la détermination de la position du reflet d'un point sur un miroir

sphérique ou cylindrique, qui sont des quadriques, en exploitant le fait que le problème est alors bidimensionnel dans un plan particulier comme il sera montré en 4.3.

Ces travaux, issus du domaine de la synthèse d'images, peuvent facilement être utilisés pour les caméras catadioptriques. Une fois la position du reflet sur le miroir obtenue, le calcul de son image est trivial. Cette méthode peut donc être utilisée pour calculer le modèle direct d'une caméra catadioptrique non centrale avec miroir sphérique ou cylindrique.

Généralisation pour les miroirs quadriques : La méthode proposée dans cette thèse peut être considérée comme une généralisation de la méthode de Glaeser, avec une formulation différente permettant son extension au calcul de la réflexion sur tout type de quadrique.

Elle permet le calcul efficace et direct de l'image d'un point de la scène après réflexion sur une quadrique. Ce calcul implique la recherche de racines d'un polynôme de degré 4 à 10 en fonction de la quadrique considérée.

Elle permet l'obtention des images correspondant à des réflexions multiples, les différentes images correspondant à différentes racines du polynôme.

Elle permet le calcul de la réflexion de points finis ainsi que ceux situés à l'infini, permettant d'utiliser la notion de point de fuite.

Miroir quadrique : Un très grand nombre de caméras utilisent de tels miroirs. Alors qu'il est difficile de garantir un alignement parfait d'une optique et d'un miroir, il est assez facile d'usiner avec précision le miroir pour qu'il ait une forme de quadrique. La modélisation du miroir par une quadrique n'est donc pas limitative a priori en terme de précision.

Caustique : Berger [151] et Swaminathan et al. [69] ont montré l'existence d'une surface caustique dans le cas où la surface réfléchissante est une quadrique. Les rayons incidents, pour être réfléchis vers la caméra, doivent être tangents à cette caustique. L'expression de la contrainte de tangence d'un rayon à la caustique n'étant pas simple, il n'est pas aisé de déterminer quels rayons sont réfléchis vers la caméra. Dans notre approche, la notion de caustique n'est pas explicitement utilisée.

4.2 Réflexion de la lumière sur une surface

4.2.1 Formulation utilisant les angles

Considérons un rayon incident r_i émis en P (figure 4.4 (a)). Ce rayon intersecte une surface S en un point χ . En ce point, la surface est équivalente à son plan tangent Π et se comporte donc comme un miroir plan dont la normale est n . Ainsi, le reflet du point P en P' est obtenu par symétrie dans le plan Π et le rayon réfléchi r_r est défini par P' et χ . Les deux lois de la réflexion en découlent :

Première loi : Les rayons r_i et r_r ainsi que la normale n sont dans un même plan.

Deuxième loi : L'angle α_i entre r_i et n est égal à l'angle α_r entre r_r et n .

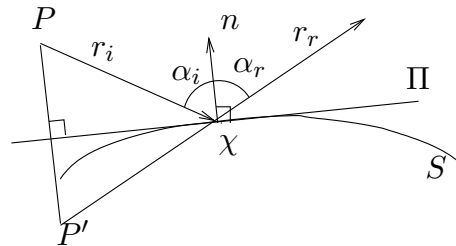


FIG. 4.4 – La réflexion sur une surface S , équivalente à la réflexion sur un plan Π tangent à S .

4.2.2 Formulation utilisant la distance du trajet lumineux

Il est possible de formuler différemment la réflexion. Le principe de Fermat établit que le trajet de la lumière est extremal, soit minimal ou maximal, en fonction du caractère convexe ou concave de la surface sur laquelle il y a réflexion. Ce principe permet aussi d'expliquer la déviation des rayons lumineux lors d'un changement de milieu (réfraction), en considérant le durée du trajet de la lumière plutôt que la distance parcourue.

Points à coordonnées finies : Le trajet emprunté par la lumière pour aller d'un point P à un point C en passant par un autre point peut être paramétré par la distance parcourue. L'ellipsoïde est une quadrique dont la somme des distances des points de la surface à chacun de ses deux foyers est égale à une constante. Pour deux points P et C et une distance parcourue adéquate, on obtient un ellipsoïde $Q_e(P, C, \xi)$ ayant pour foyers P et C et touchant S en un unique point. Cet ellipsoïde définit les trajets possibles pour la lumière dont la longueur est fonction d'un paramètre ξ . Sur la figure 4.5 est visible l'ellipsoïde (coupe en forme d'ellipse) correspondant à un trajet minimal de la lumière pour aller de P et C en passant par un point de S .

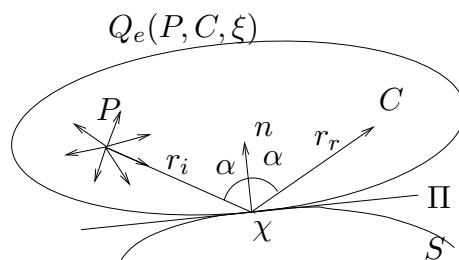


FIG. 4.5 – L'ellipsoïde dont les deux foyers sont P et C et qui est tangent à S en χ .

Il est facile de montrer que la détermination d'un point χ de S , où se réfléchit vers C le rayon provenant de P , est équivalent à déterminer le point de tangence entre l'ellipsoïde $Q_e(P, C, \xi)$ et S . Sachant que $Q_e(P, C, \xi)$ est obtenu par rotation d'une ellipse autour de l'axe passant par P et C , considérons le plan passant par P , C et χ dans lequel nous situons cette ellipse.

La normale à cette ellipse en chacun de ses points est la bissectrice aux deux droites reliant les points aux foyers. Ces deux droites peuvent être interprétées comme le trajet d'un rayon réfléchi en χ par un plan tangent à l'ellipse en les points considérés.

Pour qu'il y ait un unique contact avec S , ce plan doit être confondu avec Π . Or Π est lui-même tangent à S , donc $Q_e(P, C, \xi)$ doit aussi être tangent à S .

Points à l'infini : Soit P_∞ un point à l'infini qui ne définit pas une position mais une direction. Nous utilisons alors une propriété bien connue du paraboloïde de révolution, souvent appelé par abus de langage parabole. Celle-ci concentre les rayons arrivant depuis une direction particulière vers un point. La direction visée est perpendiculaire à la droite directrice de la parabole. L'exemple de la figure 4.6 illustre la réflexion de deux rayons provenant de la direction définie par P_∞ vers le foyer C . Pour les mêmes raisons que précédemment, la tangence entre le paraboloïde de révolution et la surface réfléchissante permet de déterminer la position χ où a lieu la réflexion.

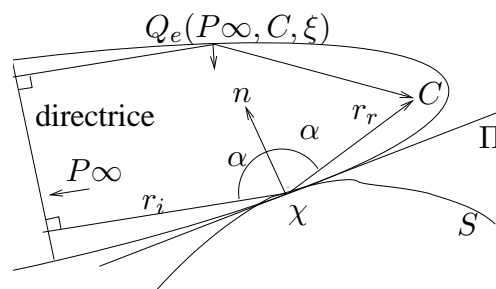


FIG. 4.6 – Le paraboloïde dont le foyer est C , la directrice orthogonale à la direction définie par P_∞ et qui est tangent à S en χ .

La paraboloïde de révolution peut être considéré comme un cas limite de l'ellipsoïde dont un des foyers est situé à l'infini. Il sera donc noté $Q_e(P_\infty, C, \xi)$ comme l'ellipsoïde, même si le rôle du paramètre ξ est alors moins évident car il n'est alors plus réellement lié à une distance.

4.3 Réflexion dans le cas 2D pour les miroirs sphériques et cylindriques

4.3.1 Un problème bidimensionnel

Pour les miroirs sphériques et cylindriques, la recherche du point χ où a lieu la réflexion peut être effectuée dans le plan, comme le montre Glaeser dans [150].

Dans le cas de la sphère, la réflexion a lieu sur un grand cercle (un cercle dont le centre est le centre O de la sphère), car tous les points de la sphère sont sur un grand cercle. La position de réflexion χ est située dans le plan défini par O , P et C .

Dans le cas du cylindre, la réflexion a aussi lieu sur un cercle particulier. Soit O_P (resp. O_C) la projection orthogonale de P (resp. C) sur l'axe de révolution du cylindre. La réflexion a lieu en χ sur un cercle du cylindre dont le centre O' est équidistant de O_P et O_C . En projetant orthogonalement dans le plan de ce cercle les points P et C en P' et C' , il est de nouveau possible de résoudre la réflexion dans le plan, cette fois avec les points O' , P' et C' .

La figure 4.7 montre la réflexion du point P vers C sur une sphère et sur un cylindre. Le plan Π tangent à S en χ est alors assimilable à une droite l_Π située dans le plan contenant O , P et C (ou O' , P' et C').

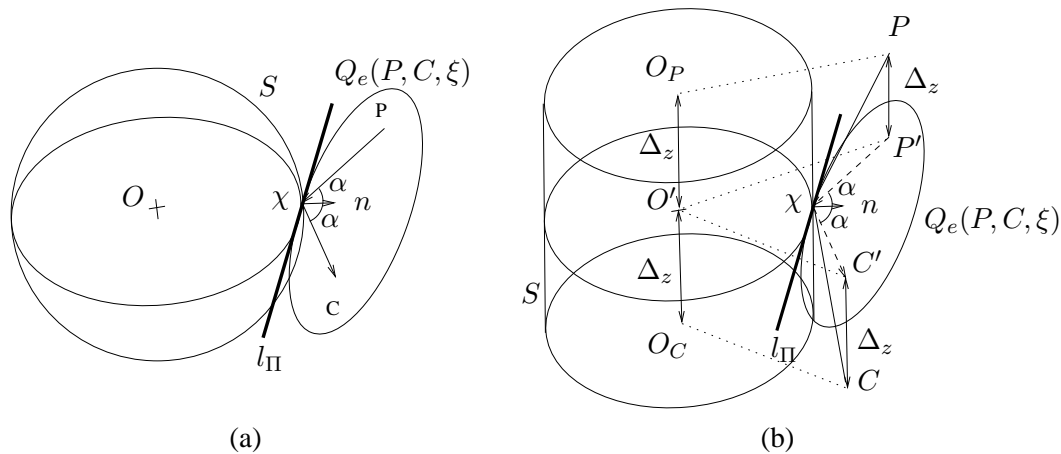


FIG. 4.7 – Réflexion du point P vers C dans le cas où la surface S est (a) une sphère, (b) un cylindre.

Glaeser formule la recherche du point χ sur le cercle comme celle de l'intersection d'une hyperbole avec le cercle considéré. Il en déduit un polynôme de degré quatre, dont les quatre racines fournissent les quatre positions où peut potentiellement se trouver χ . Nous reviendrons plus tard sur le rôle de ces quatre positions.

Le cas des points à l'infini n'est pas étudié par Glaeser. Néanmoins, dans le cas de la sphère, il peut être étudié dans le plan défini par C , O et la direction de P . Ceci n'est pas possible directement avec le cylindre, le choix de O' ne pouvant plus être fait comme dans le cas des points finis.

La formulation proposée ici est différente et exploite la tangence entre deux coniques (le cercle et l'ellipse ou la parabole) dans le cas 2D. Elle fournit des solutions plus complexes à calculer mais menant aux mêmes résultats que celle de Glaeser dans le cas de la sphère et du cylindre. Par contre, elle peut être étendue facilement en 3D, afin de calculer la réflexion sur d'autres quadriques.

4.3.2 Intersection et tangence de deux coniques

4.3.2.1 Les coniques

Les coniques sont une famille de courbes planes correspondants à l'intersection d'un plan avec un cône de révolution. Elles peuvent être classées en trois groupes, en fonction des positions relatives du plan et du cône :

- Si le plan n'est pas perpendiculaire à l'axe du cône, et ne passe pas par son sommet, la conique est alors dite *propre*. Ce groupe de coniques peut encore être subdivisé en trois en fonction de l'angle d'inclinaison du plan α avec l'axe du cône. La conique est une *ellipse* si l'angle α est supérieur à l'angle d'ouverture du cône β . Elle est une *hyperbole* si $\alpha < \beta$ et une *parabole* si $\alpha = \beta$.
- Lorsque le plan est perpendiculaire à l'axe du cône ou que $\alpha = \beta - 45^\circ$, la conique est dite *partiellement dégénérée*. Dans le premier cas, il s'agit d'un *cercle*, dans le second d'une *hyperbole équilatère* (une hyperbole dont les asymptotes sont orthogonales).
- Lorsque le plan contient le sommet du cône, la conique est *totalelement dégénérée*. Elle peut être un *couple de droites sécantes*, si $\alpha < \beta$, une seule *droite* si $\alpha = \beta$ ou un *point* unique si $\alpha > \beta$.

Les coniques sont les courbes planes algébriques dont les coordonnées cartésiennes x et y des points sont solutions d'une équation polynômiale du second degré, de la forme :

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (4.1)$$

Cette équation peut s'écrire sous forme matricielle grâce à une matrice symétrique C de dimension 3×3 . La matrice C est définie à un facteur d'échelle près, car toute multiplication de l'équation (4.1) par un entier non nul ne change pas la conique. Pour cette raison, C est appelée représentation homogène de la conique :

$$C = \begin{pmatrix} a & b/2 & c/2 \\ b/2 & d & e/2 \\ c/2 & e/2 & f \end{pmatrix}; P = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}; P^T C P = 0 \quad (4.2)$$

Grâce à la dualité entre les points et les droites [77], il existe une conique C^* définie tangentiellement dite *duale* (par anglicisme) de celle définie par les coordonnées des points. C^* est la matrice adjointe de C . Si C est inversible, alors $C^* \approx C^{-1}$ (il y a équivalence plutôt qu'égalité à cause du facteur d'échelle libre). C^* définit l'ensemble des droites l tangentes à la conique C :

$$l = (u, v, w)^T; l^T P = 0; l^T C^* l = 0 \quad (4.3)$$

4.3.2.2 Intersection de deux coniques

L'intersection de deux coniques a lieu en quatre points, éventuellement complexes, infinis ou confondus, comme expliqué par Semple et KneeBone dans [152]. Par ces quatre points, six

droites peuvent être obtenues.

La figure 4.8 montre quatre exemples d'intersections réelles de deux ellipses. Les autres cas d'intersections d'ellipses ne sont pas représentés car ils correspondent à des points d'intersection complexes.

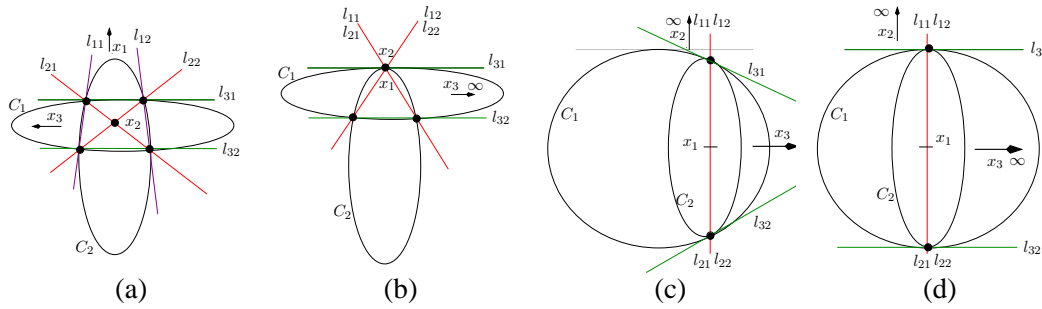


FIG. 4.8 – Intersection de deux coniques et des trois paires de droites correspondantes : (a) C_1 et C_2 s'intersectent en quatre points réels et les trois paires de droites en trois points réels. (b) C_1 et C_2 s'intersectent en quatre points réels dont un double point lorsque les coniques sont tangentes et qu'elles s'intersectent par ailleurs. Parmi les trois paires de droites, deux sont tangentes et qu'elles s'intersectent par ailleurs. Parmi les trois paires de droites, deux sont confondues et s'intersectent à l'endroit de tangence et la dernière paire s'intersecte en un point à l'infini définissant la direction de tangence. (c) C_1 et C_2 s'intersectent en deux doubles points réels lorsque les coniques sont tangentes en deux points. Deux paires de droites relient les points de tangence et la dernière paire s'intersecte en un point fini. (d) Même cas que précédemment sauf que la dernière paire de droite s'intersecte en un point à l'infini définissant la direction de tangence.

Soit $C_{12}(\alpha, \beta)$ le faisceau de coniques ayant pour base les coniques C_1 et C_2 . En exploitant le fait que les coniques sont définies à un facteur d'échelle près, l'expression générale d'une conique de ce faisceau est donnée par l'équation (4.4), avec $\alpha, \beta \in \mathbb{C}$, $\alpha\beta \neq 0$ et $\lambda \in \mathbb{C} \cup \{\infty\}$.

$$C_{12}(\alpha, \beta) = \alpha C_1 + \beta C_2 \approx C_{12}(\lambda) = C_1 - \lambda C_2 \quad (4.4)$$

Dans ce faisceau, trois coniques dégénérées correspondent aux trois paires de droites passant par chaque couple de points d'intersection comme expliqué dans [152]. Les matrices de ces coniques dégénérées $C_{n12}(\lambda) = C_1 - \lambda_n C_2$, $n = 1..3$, sont singulières, de rang strictement inférieur à trois. Le paramètre λ_n est solution de l'équation déterminantielle généralisée :

$$\det(C_1 - \lambda C_2) = 0 \quad (4.5)$$

A condition que C_2 soit inversible, le paramètre λ_n est donc aussi solution de l'équation déterminantielle (simple) :

$$\det(C_1 C_2^{-1} - \lambda I) = 0 \quad (4.6)$$

Selon la formulation utilisée, soit (4.5) ou (4.6), les paramètres λ_n correspondent aux valeurs propres généralisées de (C_1, C_2) ou aux valeurs propres de $C_1 C_2^{-1}$. On peut noter que si v_n est le vecteur propre de $C_1 C_2^{-1}$ associé à λ_n , alors la relation entre v_n et x_n , vecteur propre généralisé de (C_1, C_2) associé à λ_n , s'écrit :

$$v_n \approx C_2 x_n \quad (4.7)$$

Comme ces coniques dégénérées définissent des paires de droites l_1 et l_2 , elles peuvent aussi s'écrire :

$$C_{n12} = l_{n1} l_{n2}^T + l_{n2} l_{n1}^T \quad (4.8)$$

Les trois paires de droites l_1 et l_2 s'intersectent en trois points x_n correspondant aux vecteurs propres de $C_1 C_2^{-1}$ associés aux valeurs propres λ_n . Ces points peuvent aussi être obtenus à partir des paires de droites :

$$x_n = l_{n1} \times l_{n2} \quad (4.9)$$

Les trois paires de droites sont par exemple visibles sur la figure 4.8 (a). L'intersection x_2 est à l'intérieur de l'image alors que x_1 et x_3 en dépassent.

4.3.2.3 Tangence entre deux coniques

Lorsqu'il y a tangence entre C_1 et C_2 , deux paires de droites sont confondues (figure 4.8 (b) et (c) pour les paires 1 et 2). Ceci se traduit par deux valeurs propres λ_n identiques, les deux coniques dégénérées étant identiques.

Dans l'exemple (b), deux vecteurs propres sont identiques et conduisent à x_1 et x_2 confondus et localisés au point de tangence. x_3 étant à l'intersection des droites parallèles l_{31} et l_{32} , il est situé à l'infini dans la même direction.

Dans l'exemple (c), il y a double tangence entre C_1 et C_2 . Ceci génère deux paires confondues de droites elles mêmes confondues l_{11} , l_{12} , l_{21} et l_{22} . Les droites l_{31} et l_{32} sont quant à elles parallèles.

Dans la suite, nous considérons, sans en expliciter la preuve, que si (C_1, C_2) admet une valeur propre généralisée double, alors C_1 et C_2 sont tangentes. En effet, ce cas correspond à un vecteur propre associé à une valeur propre de multiplicité deux, indiquant un point d'intersection double entre C_1 et C_2 .

4.3.2.4 Raisonnement sur les coniques duales

Le même raisonnement peut être mené sur les coniques définies tangentiellement. Au lieu de partager un point en commun, deux coniques définies tangentiellement partagent une droite. La tangence entre deux de ces coniques se définit donc aussi par une valeur propre de multiplicité deux.

4.3.3 Tangence d'un cercle et d'une conique dont les foyers sont fixés

Appliquons maintenant ce raisonnement à la recherche de la tangence entre le cercle dans le plan considéré et une ellipse ou une parabole.

4.3.3.1 L'ellipse recherchée dans le cas d'un point dont les coordonnées sont finies

Soit $C_e^*(P, C, \xi)$ (abrégié C_e^*) la conique définissant tangentiellement l'ensemble des droites de l'ellipse C_e ayant pour foyers deux points P et C . Cette conique possède un degré de liberté, paramétré par ξ . Elle est obtenue à partir des matrices $[CP]^*$ et $[IJ]^*$ suivantes :

$$P = \begin{pmatrix} x_P \\ y_P \\ 1 \end{pmatrix}, C = \begin{pmatrix} x_C \\ y_C \\ 1 \end{pmatrix}, [CP]^* = CP^T + PC^T, [IJ]^* = \text{diag}(1, 1, 0) \quad (4.10)$$

Des coniques ayant les mêmes foyers sont dites *homofocales*. L'ensemble de coniques ayant pour foyers C et P est défini par :

$$C_e^* = [CP]^* + \xi [IJ]^* = \begin{bmatrix} 2x_C x_P + \xi & y_C x_P + x_C y_P & x_P + x_C \\ y_C x_P + x_C y_P & 2y_C y_P + \xi & y_P + y_C \\ x_P + x_C & y_P + y_C & 2 \end{bmatrix} \quad (4.11)$$

En fonction de la valeur de ξ , la conique obtenue est de différent type :

- $\xi < 0$: C_e est une ellipse dont les foyers sont C et P .
- $\xi > 0$: C_e est une hyperbole dont les foyers sont C et P .

Il faut trouver la valeur du paramètre ξ pour que C_e soit tangente à un cercle centré et de rayon unitaire C_c , correspondant à la surface réfléchissante dans le plan considéré, après application d'une transformation de similitude :

$$C_c = \text{diag}(1, 1, -1) \quad (4.12)$$

En exploitant la symétrie axiale autour de l'origine de C_c , il est possible de considérer uniquement le cas $x_F = 0$, obtenu par rotation autour de l'origine. Le déterminant de C_e^* est alors :

$$\det(C_e^*) = 2 * \xi^2 + (2y_C y_P - y_P^2 - y_C^2 - x_P^2) * \xi \quad (4.13)$$

Ce déterminant est nul et C_e est donc dégénérée quand :

$$\det(C_e^*) = 0 \iff \begin{cases} \xi = 0 \\ \xi = (2y_C y_P - y_P^2 - y_C^2 - x_P^2)/2 \end{cases} \quad (4.14)$$

Le second cas correspond à $\xi = -d^2/2$ avec d la distance entre les points P et C .

Le développement du polynôme caractéristique $\Omega(\xi, \lambda) = \det(C_e^* C_c - \lambda I) = 0$ est de degré deux en ξ et trois en λ .

Nous avons vu que la tangence entre deux coniques correspond à une valeur propre λ de multiplicité deux. Afin que C_c et C_e soient tangentes, il faut donc calculer les valeurs de ξ qui conduisent à deux valeurs λ identiques. Ceci est fait grâce à l'analyse du discriminant du polynôme.

4.3.3.2 Le discriminant

Le discriminant d'un polynôme $\Omega(\lambda)$ permet de déterminer si ses racines sont d'une *multiplicité* supérieure à un, c'est à dire qu'elles apparaissent plusieurs fois dans l'expression factorisée du polynôme. Ceci arrive lorsque le discriminant est nul.

Pour un polynôme à une variable de degré général n , dénoté $\Omega(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0$ (le polynôme est normalisé pour que $a_n = 1$.) où les coefficients sont des nombres réels, le discriminant est défini comme le déterminant d'une matrice de dimension $(2n - 1) \times (2n - 1)$. Cette matrice, appelée matrice de Sylvester permet de trouver les racines communes à deux polynômes. Dans le cas du discriminant, on recherche les racines communes à $\Omega(\lambda)$ et à $\Omega'(\lambda)$, correspondantes à des racines doubles de $\Omega(\lambda)$, la fonction et sa dérivée s'annulant en même temps lors d'une racine double.

Par exemple, dans le cas d'un polynôme de degré 4, qui sera utile dans le cas de tangence entre quadriques, $\Omega(\lambda) = \lambda^4 + a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0$, le discriminant est donné par le déterminant de la matrice D définie par l'équation (4.15). Les trois premières lignes de cette matrice correspondent aux coefficients de $\Omega(\lambda)$ et les quatre dernières à ceux de sa dérivée.

$$D = \begin{pmatrix} 1 & a_3 & a_2 & a_1 & a_0 & 0 & 0 \\ 0 & 1 & a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & 0 & 1 & a_3 & a_2 & a_1 & a_0 \\ 4 & 3a_3 & 2a_2 & a_1 & 0 & 0 & 0 \\ 0 & 4 & 3a_3 & 2a_2 & a_1 & 0 & 0 \\ 0 & 0 & 4 & 3a_3 & 2a_2 & a_1 & 0 \\ 0 & 0 & 0 & 4 & 3a_3 & 2a_2 & a_1 \end{pmatrix} \quad (4.15)$$

En développant le déterminant de D , on obtient :

$$\Delta = \det(D) = -27 * a_1^4 + 16 * a_2^4 * a_0 - 4 * a_2^3 * a_1^2 + 256 * a_0^3 - 192 * a_0^2 * a_3 * a_1 + a_2^2 * a_3^2 * a_1^2 - 4 * a_2^3 * a_3^2 * a_0 - 4 * a_3^3 * a_1^3 - 27 * a_3^4 * a_0^2 + 18 * a_3^3 * a_1 * a_2 * a_0 - 128 * a_2^2 * a_0^2 + 144 * a_2 * a_3^2 * a_0^2 + 18 * a_3 * a_1^3 * a_2 - 6 * a_0 * a_3^2 * a_1^2 + 144 * a_0 * a_1^2 * a_2 - 80 * a_3 * a_1 * a_2^2 * a_0$$

4.3.3.3 Les racines ξ

Dans le cas du polynôme de degré trois en λ correspondant à la tangence entre C_c et C_e , on obtient comme discriminant un polynôme de degré quatre en ξ dont les coefficients sont les suivants :

$$\Delta(\xi) = c_4\xi^4 + c_3\xi^3 + c_2\xi^2 + c_1\xi^1 + c_0 \quad (4.16)$$

$$\begin{aligned}
c_0 &= 4(x_P^2 - 1 + y_P^2)(y_C - 1)(y_C + 1)(-y_C^2 - x_P^2 + y_C^2 x_P^2 + 2y_C y_P - y_P^2)^2 \\
c_1 &= 40x_P^2 y_P^3 y_C - 16y_C^5 x_P^2 y_P^3 - 16y_C^5 x_P^4 y_P - 32y_C x_P^2 y_P + 40y_C^2 x_P^2 + 56y_C^2 y_P^2 - 32y_C^3 y_P - \\
& 32y_P^3 y_C + 56y_C^3 y_P^3 - 48y_C^4 y_P^2 - 48y_C^2 y_P^4 + 48y_C^3 x_P^2 y_P - 92y_C^2 x_P^2 y_P^2 + 72y_C^4 x_P^2 y_P^2 + 4y_C^4 + \\
& 4y_P^4 + 20y_C^5 y_P + 20y_P^5 y_C - 44y_C^4 x_P^2 + 32y_C^4 y_P^4 - 16y_C^5 y_P^3 - 16y_C^3 y_P^5 - 4y_C^5 x_P^2 y_P + 8x_P^2 y_P^2 + \\
& 4x_P^4 + 20y_C x_P^4 y_P - 4y_C^3 x_P^4 y_P - 20y_C^3 x_P^2 y_P^3 - 44y_C^2 x_P^4 + 40y_C^4 x_P^4 \\
c_2 &= -20y_C^3 y_P - 20y_P^3 y_C - 52y_C^2 x_P^2 y_P^2 + 2x_P^2 y_P^2 - 20y_C x_P^2 y_P + 16y_C^4 y_P^4 - 8y_C^4 x_P^4 + \\
& 32y_C^3 y_P^3 + 54y_C^2 y_P^2 - 32y_C^2 y_P^4 + y_C^4 + y_P^4 + 50y_C^2 x_P^2 - 20y_C^2 x_P^4 + 32y_C^3 x_P^2 y_P - 32y_C^4 y_P^2 + \\
& 8y_C^4 x_P^2 y_P^2 - 20y_C^4 x_P^2 + x_P^4 \\
c_3 &= 4y_C(6y_C y_P^2 - y_C^2 y_P + 4y_C^2 y_P^3 + 4y_C^2 x_P^2 y_P + 6x_P^2 y_C - y_P^3 - x_P^2 y_P) \\
c_4 &= 4y_C^2(y_P^2 + x_P^2)
\end{aligned}$$

Les quatre racines ξ_m de ce polynôme (correspondant à des valeurs propres λ de multiplicité deux) peuvent être calculées directement par les méthodes de Ferrari ou de Descartes.

La figure 4.9 (a) montre les coniques C_e obtenues pour les quatre ξ_m . Deux sont des ellipses tangentes au cercle C_c . La plus petite, $E1$, correspond au trajet lumineux le plus court et donc à la réflexion sur un miroir sphérique convexe. La plus grande, $E2$, correspond à la réflexion sur un miroir concave. Les deux hyperboles $H1$ et $H2$ sont bien tangentes à C_c mais ne correspondent pas à des solutions physiquement réalisables. En effet, dans ces cas, les rayons réfléchis sont dirigés de l'autre côté de la surface de C_c par rapport aux rayons incidents malgré que les angles de ces rayons par rapport à la normale de la surface soient égaux.

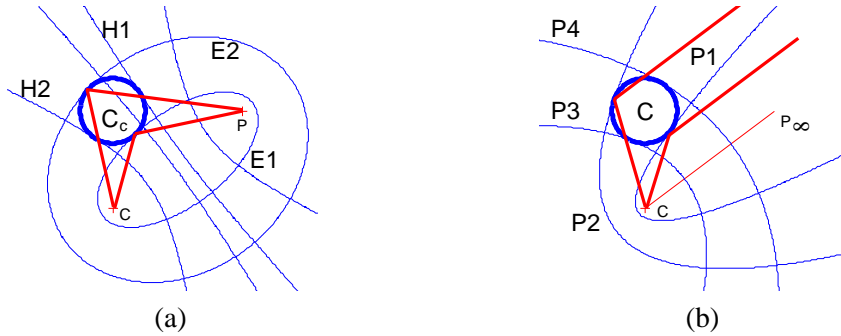


FIG. 4.9 – (a) Les quatre coniques correspondant à la réflexion sur C_c : (a) dans le cas où P est à une position finie. (b) dans le cas où P_∞ est à l'infini.

4.3.3.4 La parabole recherchée dans le cas d'un point à l'infini

Il est facile d'étendre ces résultats pour les points à l'infini. Soit $P_\infty = [x_P \ y_P \ 0]^T$, en remplaçant dans l'équation (4.10), on obtient l'ensemble des paraboles ayant pour foyer C et dont la directrice est orthogonale à la direction P_∞ définie tangentiellement par :

$$C = \begin{pmatrix} x_C \\ y_C \\ 1 \end{pmatrix}, P_\infty = \begin{pmatrix} x_P \\ y_P \\ 0 \end{pmatrix}, C_p^* = \begin{bmatrix} 2x_C x_P + \xi & y_C x_P + x_C y_P & x_P \\ y_C x_P + x_C y_P & 2y_C y_P + \xi & y_P \\ x_P & y_P & 0 \end{bmatrix} \quad (4.17)$$

En fonction des valeurs de ξ , la conique obtenue est de différent type :

- $\xi < 0$: C_p est une parabole dont le foyer est C , la directrice est orthogonale à la direction définie par P_∞ . Le côté 'ouvert' de la parabole pointe dans une direction.
- $\xi > 0$: identique au cas précédent sauf que le côté 'ouvert' de la parabole est dans la direction opposée.

Il est toujours possible de considérer $x_C = 0$ grâce à la symétrie axiale de C_c . Le déterminant de C_p^* est alors :

$$\det(C_p^*) = (-y_P^2 - x_P^2) * \xi \quad (4.18)$$

Ce déterminant est nul et C_p^* dégénère lorsque $P_\infty = [0 \ 0 \ 0]^T$ ou $\xi = 0$.

L'annulation du discriminant du polynôme caractéristique correspond au calcul des racines d'un polynôme de degré quatre en ξ (équation (4.16)) mais dont les coefficients sont les suivants :

$$\begin{aligned} c_0 &= 4(x_P^2 + y_P^2)(y_C - 1)(y_C + 1)(y_C^2 x_P^2 - x_P^2 - y_P^2)^2 \\ c_1 &= -4y_C y_P (x_P^2 + y_P^2)(y_C^2 x_P^2 + 4y_C^4 x_P^2 - 5x_P^2 - 5y_P^2 + 4y_C^2 y_P^2) \\ c_2 &= -(x_P^2 + y_P^2)(20y_C^2 x_P^2 + 8y_C^4 x_P^2 - x_P^2 - 16y_C^4 y_P^2 + 32y_C^2 y_P^2 - y_P^2) \\ c_3 &= 4y_C y_P (x_P^2 + y_P^2)(2y_C - 1)(2y_C + 1) \\ c_4 &= 4y_C^2 (x_P^2 + y_P^2) \end{aligned}$$

La figure 4.9 (b) montre les quatre paraboles obtenues pour les quatre ξ_m . La parabole $P1$ correspond au trajet lumineux le plus court et donc à la réflexion sur un miroir sphérique convexe. La plus grande, $P2$, correspond à la réflexion sur un miroir concave. Les deux autres paraboles $P3$ et $P4$ sont bien tangentes à C_c mais correspondent à des solutions pour une direction de P_∞ opposée.

4.3.3.5 Calcul des coordonnées de la réflexion

Les coordonnées du point $\chi = [x_\chi, y_\chi, 1]^T$ vérifient à la fois $\chi^T C_c \chi = 0$ et $\chi^T C_s \chi = 0$ avec C_s la conique solution considérée (l'une des ellipses ou paraboles). Néanmoins, à cause de la précision finie du codage de la valeur numérique de ξ , l'intersection n'est pas toujours parfaitement définie, le ξ obtenu ne correspond donc pas exactement à un point d'intersection unique. Nous avons vu que deux des paires de droites définies par l'intersection de deux coniques s'intersectent au point de tangence. Dans le cas où il y a *presque* tangence, deux paires de droites *presque* identiques s'intersectent en deux points *presque* confondus. Un de ces deux points ou la moyenne des deux peuvent être utilisés pour définir le point de *presque* tangence. Les coordonnées homogènes de ces points sont données par les vecteurs propres associés à la valeur propre *presque* double de $C_s^* C_c$.

4.4 Réflexion dans le cas 3D pour tous les miroirs quadriques

Nous nous intéressons maintenant à la réflexion à la surface d'une quadrique quelconque. Le problème est alors tridimensionnel et consiste à détecter l'ellipsoïde ou le parabolôïde de révolution tangent à la quadrique modélisant le miroir.

4.4.1 Intersection et tangence de deux quadriques

4.4.1.1 Les quadriques

Les quadriques sont des surfaces quadratiques. Elles sont classées en 17 catégories [153] que nous ne détaillerons pas ici et qui contiennent notamment : hyperboloïde, parabolôïde, ellipsoïde, sphère, cylindre, cône, paire de plans, plan. Elles peuvent être considérées comme une généralisation des coniques en 3D.

Les coordonnées cartésiennes x , y et z des points d'une quadrique sont solutions d'une équation polynômiale du second degré, de la forme :

$$ax^2 + bxy + cxz + dx + ey^2 + fyz + gy + hz^2 + iz + j = 0 \quad (4.19)$$

Cette équation peut s'écrire sous forme matricielle grâce à une matrice symétrique Q de dimension 4×4 définie à un facteur d'échelle près :

$$Q = \begin{pmatrix} a & b/2 & c/2 & d/2 \\ b/2 & e & f/2 & g/2 \\ c/2 & f/2 & h & i/2 \\ d/2 & g/2 & i/2 & j \end{pmatrix}; P = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}; P^T Q P = 0 \quad (4.20)$$

La dualité s'exprime dans l'espace entre des points et des plans [77]. Il existe une quadrique Q^* définie tangentiellement dite *duale* de celle définie par les coordonnées des points. Q^* est la matrice adjointe de Q . Si Q est inversible, alors $Q^* \approx Q^{-1}$ et Q^* définit l'ensemble des plans Π tangents à la quadrique Q :

$$\Pi = (t, u, v, w)^T; \Pi^T P = 0; \Pi^T Q^* \Pi = 0 \quad (4.21)$$

4.4.1.2 Intersection de deux quadriques

Alors que deux coniques s'intersectent en quatre points situés à l'intersection de droites, deux quadriques s'intersectent en une ou plusieurs quartiques (courbes polynômiales de degré 4) situées à l'intersection de cônes [152]. La figure 4.10 montre deux exemples d'intersection de deux quadriques. Comme pour les droites associées à l'intersection de deux coniques, les cônes peuvent être complexes, confondus ou avoir un sommet situé à l'infini. Lazard et al. propose dans [154] un algorithme permettant le calcul de cette intersection dans le cas où les coefficients des quadriques sont réels.

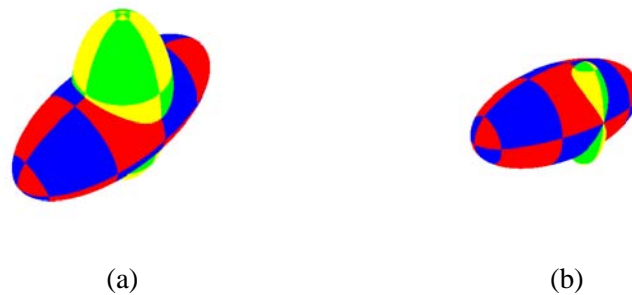


FIG. 4.10 – Exemple d’intersection de deux quadriques : (a) Deux ellipsoïdes non tangents. (b) Deux ellipsoïdes tangents en un point.

Soit $Q_{12}(\alpha, \beta)$ le faisceau de quadriques ayant pour base les quadriques Q_1 et Q_2 . Comme pour les coniques, ce faisceau peut être écrit sous la forme $Q_{12}(\lambda)$:

$$Q_{12}(\alpha, \beta) = \alpha Q_1 + \beta Q_2; \quad Q_{12}(\lambda) \approx Q_1 - \lambda Q_2 \quad (4.22)$$

Parmi ce faisceau, quatre quadriques dégénérées correspondent aux quatre cônes passant par les quartiques à l’intersection de Q_1 et Q_2 . Comme ces quadriques sont dégénérées, elles ne sont pas de rang plein. Comme pour les coniques, selon que Q_2 est inversible ou non, les quadriques dégénérées sont obtenues par analyse des valeurs propres λ_n de $(Q_1 \ Q_2^{-1})$ ou des valeurs propres généralisées de (Q_1, Q_2) pour $n \in \{1, 2, 3, 4\}$. Chaque valeur propre λ_n fournit une quadrique dégénérée qui représente un cône parmi les quatre possibles :

$$Q_{n12} = Q_1 - \lambda_n Q_2 \text{ tq } \begin{cases} \det(Q_1 \ Q_2^{-1} - \lambda_n I) = 0 \\ \text{ou } \det(Q_1 - \lambda_n Q_2) = 0 \end{cases} \quad (4.23)$$

Les sommets x_n des quatre cônes correspondent aux vecteurs propres associés aux valeurs propres λ_n .

Les quatre cônes définis par l’intersection des deux ellipsoïdes de la figure 4.10 (a) sont visibles sur la figure 4.11. Le troisième à partir de la gauche est en fait un cylindre hyperbolique, obtenu à partir d’un sommet x_n situé à l’infini.

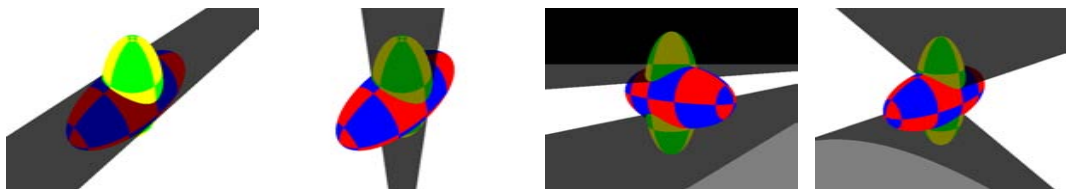


FIG. 4.11 – Les quatre cônes définis par l’intersection des deux quadriques dans le cas de l’intersection montrée sur la figure 4.10 (a).

4.4.1.3 Tangence entre deux quadriques

Quand les deux quadriques Q_1 et Q_2 ont un point de tangence en A , Q_{n12} contient deux cônes confondus ayant pour sommet le point A . Dans l'exemple de la figure 4.10 (b), les Q_{n12} sont visibles sur la figure 4.12. Les troisième et quatrième à partir de la gauche sont confondus.

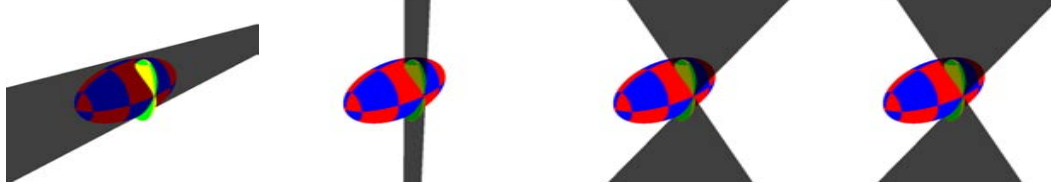


FIG. 4.12 – Les quatre cônes définis par l'intersection des deux quadriques tangentes dans le cas de l'intersection montrée sur la figure 4.10 (b).

Comme pour les coniques, nous considérons, sans en donner ici une preuve rigoureuse, que si (Q_1, Q_2) admet une valeur propre généralisée de multiplicité deux, alors Q_1 et Q_2 sont tangentes. La preuve sera proposée dans une prochaine publication.

4.4.1.4 Raisonnement sur les quadriques définies tangentiellement

Le même raisonnement peut être mené sur les quadriques définies tangentiellement. Au lieu de partager un point en commun, deux quadriques définies tangentiellement partagent un plan. La tangence entre deux de ces quadriques se définit donc aussi par une valeur propre de multiplicité deux.

4.4.2 Tangence entre une quadrique modélisant un miroir et une quadrique dont les foyers sont fixés

4.4.2.1 Ellipsoïde recherché dans le cas d'un point dont les coordonnées sont finies

Soit $Q_e^*(P, C, \xi)$ (abrégé Q_e^*) la quadrique définissant l'enveloppe de plans d'un ellipsoïde Q_e ayant pour foyers les points C et P . Cette quadrique possède un degré de liberté, paramétré par ξ . Elle est obtenue à partir des matrices $[CP]^*$ et $[IJ]^*$ suivantes :

$$C = \begin{pmatrix} x_C \\ y_C \\ z_C \\ 1 \end{pmatrix}, P = \begin{pmatrix} x_P \\ y_P \\ z_P \\ 1 \end{pmatrix}, [CP]^* = CP^T + PC^T, [IJ]^* = \text{diag}(1, 1, 1, 0) \quad (4.24)$$

L'ensemble des quadriques homofocales ayant pour foyers F et G est défini par :

$$Q_e^* = [CP]^* + \xi [IJ]^* \quad (4.25)$$

En développant, on obtient :

$$Q_e^* = \begin{bmatrix} 2x_C x_P + \xi & y_C x_P + x_C y_P & z_C x_P + x_C z_P & x_P + x_C \\ y_C x_P + x_C y_P & 2y_C y_P + \xi & z_C y_P + y_C z_P & y_P + y_C \\ z_C x_P + x_C z_P & z_C y_P + y_C z_P & 2z_C z_P + \xi & z_P + z_C \\ x_P + x_C & y_P + y_C & z_P + z_C & 2 \end{bmatrix} \quad (4.26)$$

Le même raisonnement que pour les coniques peut être mené pour déterminer la forme de Q_e en fonction de la valeur de ξ . Q_e est du type :

- $\xi < 0$: Q_e est un ellipsoïde dont les foyers sont C et P .
- $\xi > 0$: Q_e est un hyperboloïde dont les foyers sont C et P .

Afin de trouver la valeur de ξ telle que Q_e soit tangent à une quadrique Q_M modélisant le miroir, le développement du polynôme caractéristique $\Omega(\xi, \lambda) = \det(Q_e^* Q_M - \lambda I) = 0$ est effectué. Il est au maximum de degré trois en ξ et quatre en λ .

Nous avons vu que la tangence entre deux quadriques correspond à une valeur propre λ de multiplicité deux. Afin que Q_M et Q_e soient tangentes, il faut donc calculer les valeurs de ξ qui conduisent à une valeur propre λ de multiplicité deux. Ceci est fait grâce à l'analyse du discriminant $\Delta(\xi)$ du polynôme, cette fois de degré quatre, dont le développement a déjà été effectué en section 4.3.3.2.

Le discriminant obtenu est un polynôme de degré variable en ξ en fonction de Q_M (voir les résultats en section 4.5 pour le degré correspondant à chaque quadrique).

Il n'existe pas de formule directe pour calculer les racines d'un polynôme de degré supérieur à quatre. Néanmoins, de nombreuses méthodes itératives fournissent des solutions. Dans le cadre des expériences présentées ici, les racines sont calculées comme les valeurs propres de la matrice compagnone du polynôme. On pourra néanmoins noter l'existence d'autres méthodes, telle que celle que Fortune présente dans [155]. Cette méthode itère des calculs de valeurs propres d'une matrice représentant le polynôme. L'auteur compare sa méthode avec d'autres méthodes, notamment en termes de temps de calculs et montre un gain pouvant être assez important.

4.4.2.2 Paraboloïde recherchée dans le cas d'un point à l'infini

Comme dans le cas 2D, il est facile d'étendre ces résultats pour les points à l'infini. Soit $P_\infty = [x_P \ y_P \ z_P \ 0]^T$, en remplaçant dans l'équation (4.24), on obtient l'ensemble des paraboloïdes de révolution ayant pour foyer C et dont la directrice est orthogonale à la direction P_∞ définie tangentiellement par :

$$C = \begin{pmatrix} x_C \\ y_C \\ z_C \\ 1 \end{pmatrix}, P_\infty = \begin{pmatrix} x_P \\ y_P \\ z_P \\ 0 \end{pmatrix} \quad (4.27)$$

$$Q_p^* = \begin{bmatrix} 2x_C x_P + \xi & y_C x_P + x_C y_P & z_C x_P + x_C z_P & x_P \\ y_C x_P + x_C y_P & 2y_C y_P + \xi & z_C y_P + y_C z_P & y_P \\ z_C x_P + x_C z_P & z_C y_P + y_C z_P & 2z_C z_P + \xi & z_P \\ x_P & y_P & z_P & 0 \end{bmatrix} \quad (4.28)$$

En fonction des valeurs de ξ , la quadrique obtenue est de différent type :

- $\xi < 0$: Q_p est une parabolôide de révolution dont le foyer est C et la directrice est orthogonale à la direction définie par P_∞ . Le côté 'ouvert' du parabolôide pointe dans une direction.
- $\xi > 0$: identique au cas précédent sauf que le côté 'ouvert' du parabolôide est dans la direction opposée.

Comme pour l'ellipsoïde, le discriminant obtenu est un polynôme de degré variable en ξ en fonction de Q_M (voir les résultats en section 4.5 pour le degré correspondant à chaque quadrique).

Le code MAPLE permettant d'obtenir les polynômes dans le cas d'un miroir sphérique traité en 3D pour les points finis et infinis est présenté en Annexe E ainsi que les coefficients correspondants.

4.4.2.3 Calcul des coordonnées de la réflexion

Le calcul des coordonnées du point $\chi = [x_\chi, y_\chi, z_\chi, 1]^T$ est effectué comme pour les coniques, en vérifiant à la fois $\chi^T Q_M \chi = 0$ et $\chi^T Q_s \chi = 0$ avec Q_s la quadrique solution considérée (l'une des ellipsoïdes ou parabolôides de révolution). Les coordonnées homogènes du point sont obtenues comme étant celles du vecteur propre associé à la valeur propre *presque* double de $Q_s^* Q_C$.

4.4.3 Considérations annexes

4.4.3.1 Choix des racines

Le polynôme $\Delta(\xi)$ obtenu est de différent degré en fonction de la quadrique considérée. Parmi ses racines ξ , seules les réelles négatives correspondent à des ellipsoïdes pour les points finis. Selon la configuration (position des points et quadrique considérée), un nombre variable de racines ξ sont réelles et négatives. Toutes ou parties de ces racines correspondent à des réflexions visibles à cause des occultations par le miroir et du fait que le miroir ne couvre en fait qu'une portion de la quadrique.

Occultation par le miroir : Selon le caractère convexe ou concave du miroir, les racines ξ correspondantes à une possible réflexion sont différentes. Ceci a déjà été montré avec l'exemple de la sphère sur la figure 4.9.

Sur l'exemple d'un miroir convexe paraboloidal de la figure 4.13, la caméra est placée en C et une racine ξ conduit à un point de tangence situé en χ . Afin de vérifier que le point χ n'est pas occulté par la quadrique, il est nécessaire de vérifier s'il y a intersection du segment (en fait un rayon de projection de la caméra) reliant les points C à χ avec la quadrique.

Soit P_I un point du segment défini par les points C et χ tel que $P_I = C + t(\chi - C)$ pour $0 < t < 1$. En imposant que P_I appartienne à la quadrique Q par substitution dans l'équation (4.21), on obtient un polynôme de degré deux en t . Si il existe une racine t_0 comprise entre 0 et 1 exclus alors le rayon de projection intersecte la quadrique entre les points C et χ et il y a donc occultation par la quadrique en $P_I = C + t_0(\chi - C)$.

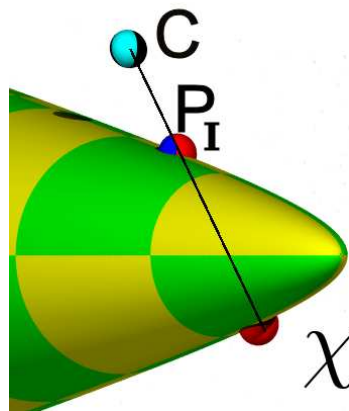


FIG. 4.13 – Exemple d'occultation du point χ vu depuis le point C par la quadrique, il y a intersection du rayon de projection et de la quadrique en P_I .

Le miroir, une portion de quadrique : Le choix des racines ξ correspondants à des réflexions est aussi guidé par le fait que le miroir ne couvre physiquement qu'une partie de la quadrique. Ainsi, par exemple, dans le cas du miroir hyperboloïdal en deux morceaux, seule une portion finie d'un des deux morceaux est généralement couverte par le miroir.

Il est possible de définir des contraintes d'inégalité à base de plans. Dans l'exemple cité, on peut par exemple éliminer les racines correspondant à des réflexions en des points dont les coordonnées z sont inférieures à 0 ou supérieures à un seuil.

Plusieurs réflexions : Dans certaines configurations, un point peut se refléter à différents endroits sur la quadrique vers la caméra. Dans ces cas, plusieurs racines ξ négatives conduisent à des points de tangence compris dans la portion de quadrique couverte par le miroir et qui ne

sont pas occultés par le miroir. Ce cas arrive par exemple à l'intérieur d'un miroir paraboloidal ou sur un paraboloidal hyperbolique comme il sera montré par la suite.

4.4.3.2 Précisions des calculs numériques

Les détails sur la précision du codage des valeurs numériques flottantes en double précision est donné en Annexe E. Les précautions à prendre pour le calcul des coefficients du polynômes y sont discutées.

Dans le cadre de ces recherches, il a été préféré d'utiliser des outils permettant un codage de meilleure précision. Ceci permet de s'affranchir du problème cité et d'obtenir au final des valeurs plus précises qu'avec des flottants en doubles précision. Le logiciel MAPLE est utilisé en précision arithmétique variable afin d'obtenir les coefficients des polynômes ainsi que leurs racines à la précision souhaitée, ceci au prix d'un temps de calcul plus élevé.

4.5 Résultats

4.5.1 Calcul des points de tangence

Un logiciel Matlab incluant du code MAPLE a été créé afin de valider la méthode. La réflexion sur les différentes quadriques jugées utiles a été testée. Le programme permet la génération automatique d'un programme Matlab qui réalise le calcul du polynôme en fonction des coefficients. Ensuite, il calcule les différentes racines qui correspondent aux positions de tangence avec une précision bien supérieure à celle des flottants doubles précision. Il est capable de générer du code pour le logiciel de raytracing Persistence Of Vision qui permet de visualiser la solution en 3D comme le montrent les figures d'exemple suivantes.

La similitude est la transformation la plus riche qui conserve la forme de l'ellipse et du paraboloidal de révolution. Une transformation plus complexe, par exemple affine, ne les conserve pas. Afin d'obtenir les formules les plus simples possibles, les calculs sont effectués sur des quadriques exprimées sous une forme simplifiée. Ainsi, par exemple, la sphère est considérée centrée et de rayon unitaire. Afin de calculer la réflexion sur un miroir réel, il faut tout d'abord appliquer la transformation de similitude qui transforme la quadrique du miroir en sa forme simplifiée, faire les calculs et appliquer la transformation inverse pour obtenir le résultat sur la quadrique réelle.

Dans les figures d'exemple suivantes, la quadrique représentant le miroir est affichée en vert et jaune (plus clair en N/B). L'ellipsoïde ou le paraboloidal de révolution correspondant à la plus petite racine ξ est affiché en rouge et bleu (plus foncé en N/B).

Afin de comparer la complexité des polynômes obtenus pour les différentes quadriques, les caractères ASCII des formules analytiques sont comptés. Cette mesure, notée *Compt*, n'est pas très représentative de la complexité réelle du calcul par l'ordinateur mais fournit une idée de la taille des équations.

4.5.1.1 Sphère

Soit Q_s la sphère centrée de rayon unitaire :

$$Q_s : x^2 + y^2 + z^2 - 1 = 0; Q_s = \text{diag}(1, 1, 1, -1) \quad (4.29)$$

Avec un miroir sphérique, il est possible de considérer x_C et y_C en 0 grâce à la double symétrie de la sphère. Ainsi, seuls z_C , x_P , y_P et z_P interviennent. Le polynôme obtenu est de degré quatre pour les points finis et à l'infini, comme lors de l'analyse 2D. Le détail de la détermination du polynôme est fourni en Annexe E. $Compt = 1529$ dans le cas d'un point fini et $Compt = 519$ dans le cas d'un point à l'infini. La figure 4.14 montre un exemple de tangence.

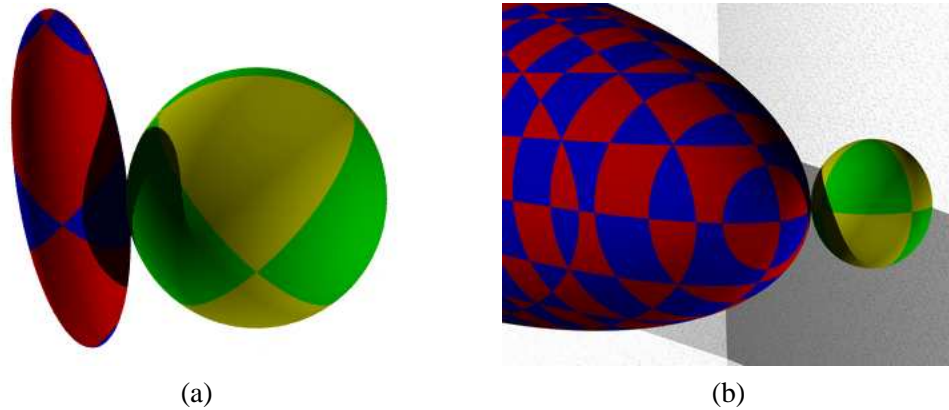


FIG. 4.14 – Exemple de tangence entre le miroir sphérique et l'ellipsoïde (a) ou le paraboloidé de révolution (b) calculé.

4.5.1.2 Cylindre

Soit Q_C le cylindre dont l'axe de révolution est z et de rayon unitaire.

$$Q_C : x^2 + y^2 = 1; Q_C = \text{diag}(1, 1, 0, -1) \quad (4.30)$$

x_C est considéré en 0 grâce à la symétrie autour de l'axe z . Ainsi, seuls y_C , z_C , x_P , y_P et z_P interviennent. Le polynôme obtenu est de degré six pour les points finis, contrairement à celui fourni par la méthode 2D proposée par Glaeser dans [150]. Par contre, parmi les six racines, deux sont complexes (sur tous les tests effectués), et les quatre racines réelles sont bien obtenues comme dans la méthode de Glaeser. Par contre, le polynôme est de degré quatre pour les points à l'infini, comme pour le cas 2D. $Compt = 3918$ dans le cas d'un point fini et $Compt = 347$ dans le cas d'un point à l'infini. La figure 4.15 montre un exemple de tangence.

4.5.1.3 Ellipsoïde

Soit $Q_e(e_1)$ l'ellipsoïde dont les foyers se déplacent le long de l'axe x , et dont l'écart est fonction d'un paramètre e_1 :

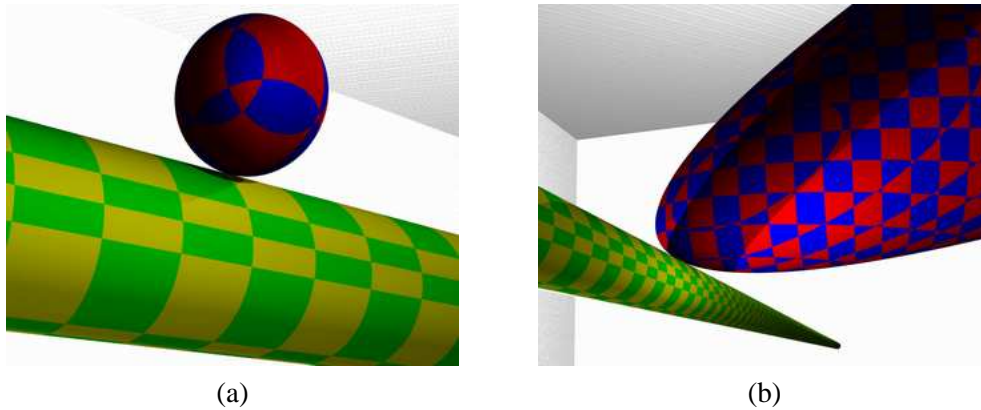


FIG. 4.15 – Exemple de tangence entre le miroir cylindrique et l'ellipsoïde (a) ou le paraboloidé de révolution (b) calculé.

$$Q_e(e_1) : e_1 x^2 + y^2 + z^2 = 1; Q_e(e_1) = \text{diag}(e_1, 1, 1, -1) \quad (4.31)$$

x_C est considéré en 0 grâce à la symétrie autour de l'axe z . Ainsi, seuls y_C, z_C, x_P, y_P et z_P interviennent. Le polynôme obtenu est de degré huit pour les points finis et infinis. $\text{Compt} = 428136$ dans le cas d'un point fini et $\text{Compt} = 97838$ dans le cas d'un point à l'infini. La figure 4.16 montre un exemple de tangence.

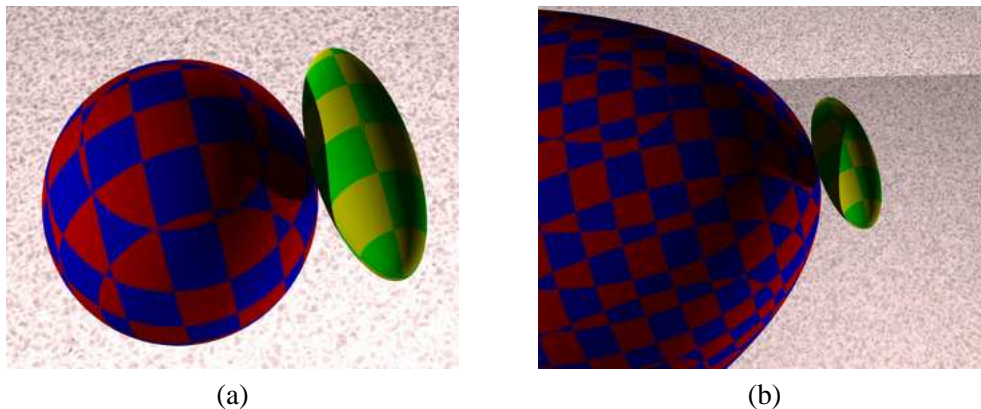


FIG. 4.16 – Exemple de tangence entre le miroir ellipsoïdal et l'ellipsoïde (a) ou le paraboloidé de révolution (b) calculé.

4.5.1.4 Paraboloidé de révolution

Soit $Q_p(p_1)$ le paraboloidé de révolution ayant z comme axe de révolution et paramétrée par p_1 :

$$Q_p(p_1) : p_1(x^2 + y^2) - 2z = 0; Q_p(p_1) = \begin{pmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad (4.32)$$

x_C est considéré en 0 grâce à la symétrie autour de l'axe z . Ainsi, seuls y_C, z_C, x_P, y_P et z_P interviennent. Le polynôme obtenu est de degré sept pour les points finis et huit pour les points à l'infini. $Compt = 80944$ dans le cas d'un point fini et $Compt = 6414$ dans le cas d'un point à l'infini. La figure 4.17 montre un exemple de tangence.

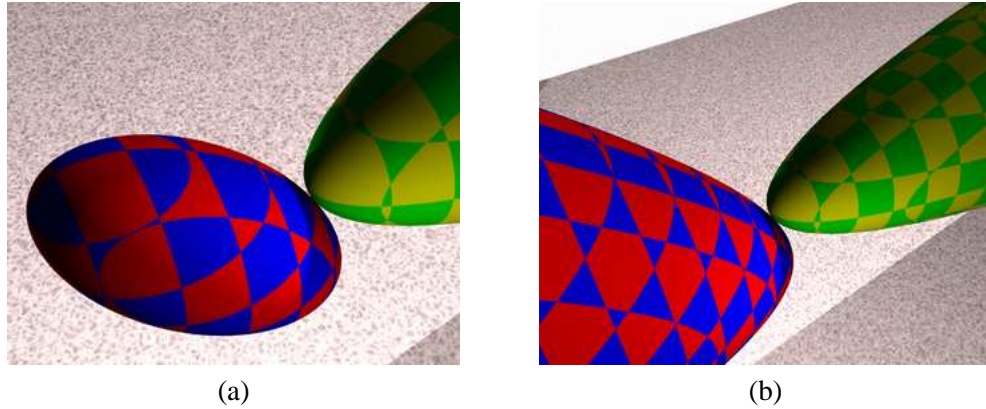


FIG. 4.17 – Exemple de tangence entre le miroir paraboloidal et l'ellipsoïde (a) ou le paraboloidal de révolution (b) calculé.

4.5.1.5 Hyperboloïde

Soit $Q_h(h_1)$ l'hyperboloïde (en un ou deux morceaux) ayant z comme axe de révolution et paramétrée par h_1 :

$$Q_h(h_1) : h_1(x^2 + y^2) - z^2 + 1 = 0; Q_h(h_1) = \text{diag}(h_1, h_1, -1, 1) \quad (4.33)$$

x_C est considéré en 0 grâce à la symétrie autour de l'axe z . Ainsi, seuls y_C, z_C, x_P, y_P et z_P interviennent. Le polynôme obtenu est de degré huit pour les points finis et ceux à l'infini. $Compt = 348141$ dans le cas d'un point fini et $Compt = 106337$ dans le cas d'un point à l'infini. La figure 4.18 montre un exemple de tangence.

4.5.1.6 Cône Droit

Soit $Q_c(c_1)$ le cône droit ayant z comme axe de révolution et paramétré par son ouverture c_1 :

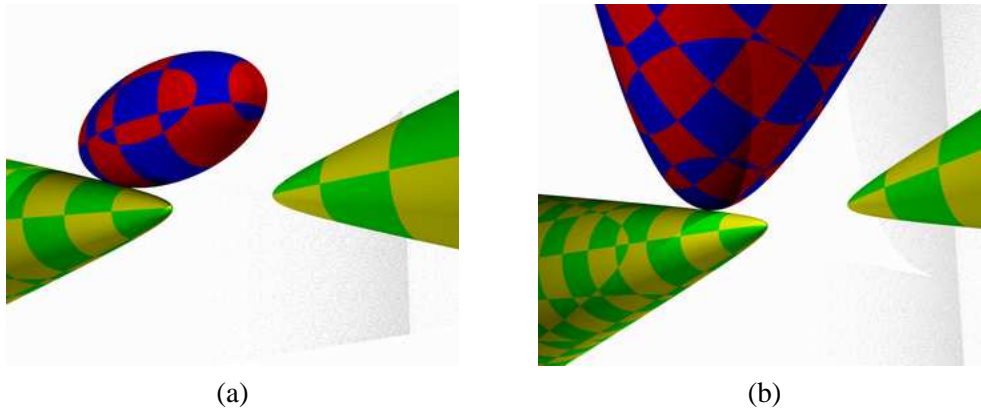


FIG. 4.18 – Exemple de tangence entre le miroir hyperboloïdal et l’ellipsoïde (a) ou le paraboloïde de révolution (b) calculé.

$$Q_c(c_1) : x^2 + y^2 + c_1 z^2 = 0; Q_c(c_1) = \text{diag}(1, 1, c_1, 0) \quad (4.34)$$

x_C est considéré en 0 grâce à la symétrie autour de l’axe z . Ainsi, seuls y_C, z_C, x_P, y_P et z_P interviennent. Le polynôme obtenu est de degré huit pour les points finis et à l’infini. $\text{Compt} = 29723$ dans le cas d’un point fini et $\text{Compt} = 29722$ dans le cas d’un point à l’infini. La figure 4.19 montre un exemple de tangence.

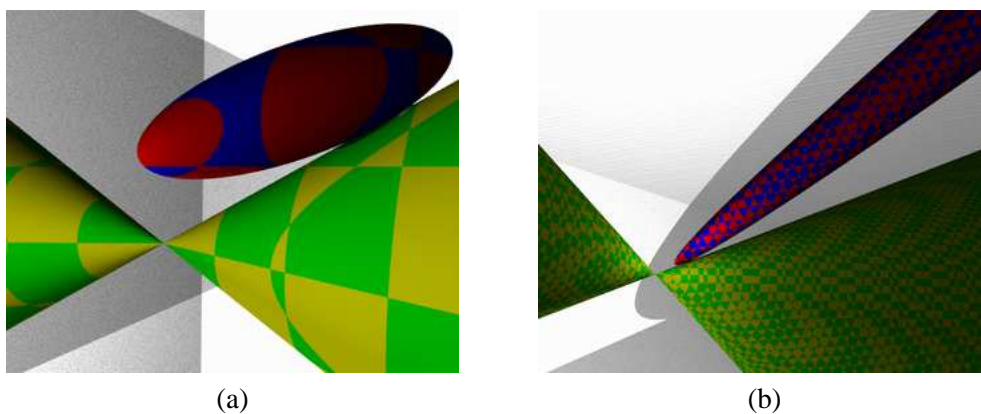


FIG. 4.19 – Exemple de tangence entre le miroir conique et l’ellipsoïde (a) ou le paraboloïde de révolution (b) calculé.

4.5.1.7 Un plan

Soit Q_π la quadrique définissant un plan $z = 0$, qui a donc z comme axe de symétrie :

$$Q_\pi : z = 0; Q_\pi = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.35)$$

x_C est considéré en 0 grâce à la symétrie autour de l'axe z . Ainsi, seuls y_C, z_C, x_P, y_P et z_P interviennent. *Compt* = 96 dans le cas d'un point fini. La figure 4.20 (a) montre un exemple de tangence. Le polynôme obtenu est de degré un pour les points finis mais il n'y a pas de solution pour les points à l'infini avec cette formulation, bien qu'elle soit facile à obtenir d'une autre façon comme le montre la figure 4.20 (b). Parmi l'ensemble des rayons r_r émis depuis la direction P_r (reflet dans le plan de P), celui qui passe par C après réflexion dans le plan l'intersecte en χ .

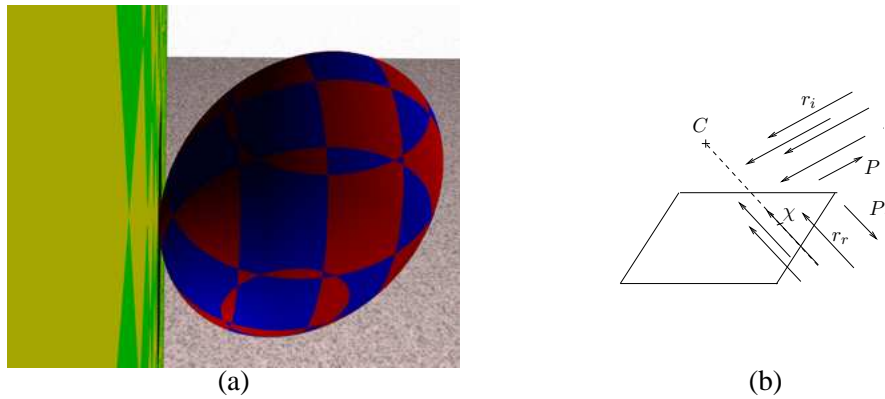


FIG. 4.20 – (a) Exemple de tangence entre le miroir plan et l'ellipsoïde calculé. (b) Obtention de l'image d'un point à l'infini.

4.5.1.8 Une paire de plans

Soit Q_Π la quadrique définissant une paire de plans dont π_1 est $z = 0$ et π_2 est libre. Il n'y a pas d'axe de révolution pour cette quadrique lorsque π_1 et π_2 ne sont pas parallèles.

$$\pi_1 = [0 \ 0 \ 1 \ 0]^T; \pi_2 = [a_2 \ b_2 \ c_2 \ d_2]^T; Q_\Pi = \pi_1 \pi_2^T + \pi_2 \pi_1^T \quad (4.36)$$

$$Q_\Pi = \begin{pmatrix} 0 & 0 & a_2 & 0 \\ 0 & 0 & b_2 & 0 \\ a_2 & b_2 & 2c_2 & d_2 \\ 0 & 0 & d_2 & 0 \end{pmatrix} \quad (4.37)$$

Tous les paramètres x_C, y_C, z_C, x_P, y_P et z_P interviennent ici. Le polynôme obtenu est de degré deux pour les points finis et infinis, dont chacune des deux racines correspondent au reflet sur chacun des plans. $Compt = 357$ dans le cas d'un point fini et $Compt = 319$ dans le cas d'un point à l'infini. La figure 4.21 montre un exemple de tangence.

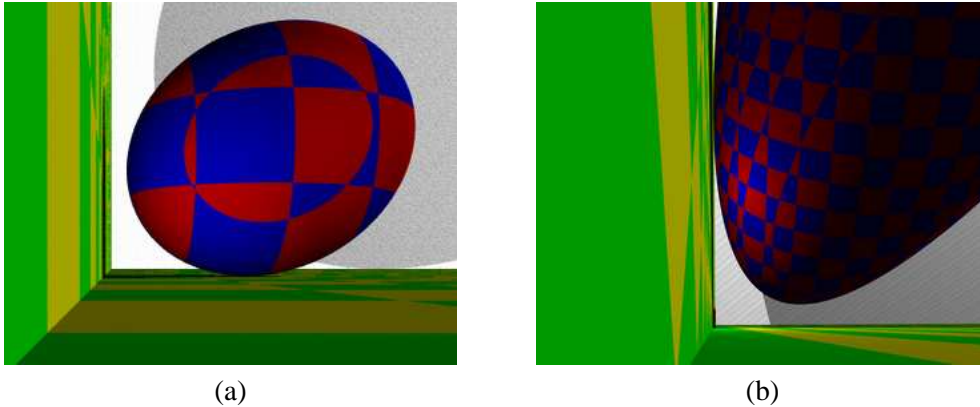


FIG. 4.21 – Exemple de tangence entre un miroir à deux plans et l'ellipsoïde (a) ou le parabolôïde de révolution (b) calculé.

4.5.1.9 Quadrique en général

Il est possible de développer les calculs pour une quadrique quelconque définie par une matrice comme dans l'équation (4.20). Néanmoins, les polynômes obtenus ont des paramètres dont les expressions sont très longues ($Compt \geq 3.5e^6$ pour les points finis par exemples), ce qui les rend difficilement utilisables en pratique. La figure 4.22 montre un exemple de tangence correspondant à une quadrique définie par une matrice. Cette quadrique est ici un hyperboloïde en un morceau et le calcul aurait donc pu être effectué plus efficacement en utilisant la formule spécialisée.

Il est toujours possible d'exprimer la quadrique sous une forme plus simple. Par exemple, pour un parabolôïde hyperbolique défini par $Q_{ph}(p_1, p_2)$:

$$Q_{ph}(p_1, p_2) : -p_1x^2 + p_2y^2 + 2z = 0; Q_{ph}(p_1, p_2) = \begin{pmatrix} -p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.38)$$

Il n'y a pas d'axe de symétrie, donc toutes les coordonnées des points interviennent comme dans le cas du double plan. Le polynôme obtenu est de degré neuf pour les points finis et huit pour les points à l'infini. $Compt = 371337$ dans le cas d'un point fini (c'est dix fois moins de caractères que dans le cas général) et $Compt = 64866$ dans le cas d'un point à l'infini. La figure 4.22 (b) montre un exemple de tangence pour un point dont les coordonnées sont finies.

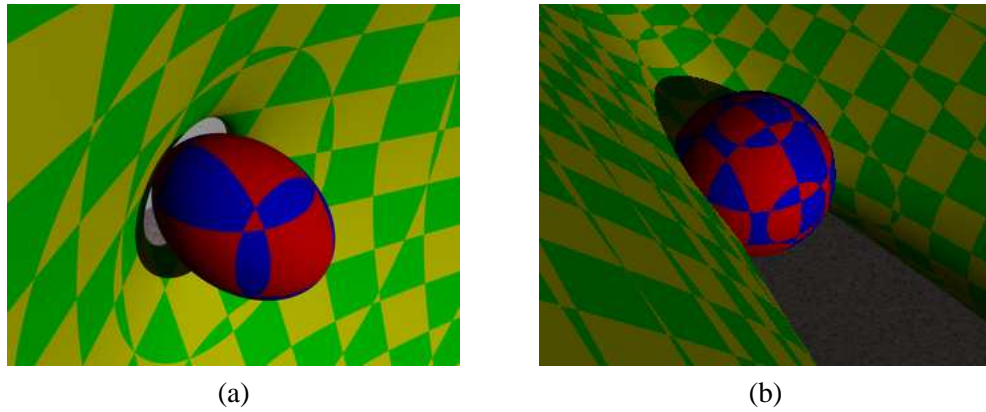


FIG. 4.22 – Exemples de tangence entre l’ellipsoïde calculé et le miroir défini par une quadrique quelconque, (a) un hyperboloïde, et (b) un paraboloïde hyperbolique.

4.5.1.10 Caméras axiales

Ramalingam et al étudient dans [94] les caméras axiales. Ces caméras, à mi-chemin entre les caméras centrales et les caméras non centrales générales, possèdent une droite d que tous les rayons de projection intersectent (contrairement à un point pour une caméra centrale).

Si la détermination de la projection d’un point avec une caméra de ce type peut se faire plus efficacement en 2D dans le plan contenant d et le point considéré (avec une approche utilisant des coniques, comme dans le cas bidimensionnel déjà étudié), nous proposons ici d’utiliser notre approche en 3D afin de la valider.

En général, une caméra axiale peut être obtenue dans le cas où la caméra est alignée avec l’axe de révolution de la quadrique du miroir. Ceci conduit à un point C dont les coordonnées x_C et y_C sont nulles. Par conséquent, l’expression des coefficients des polynômes s’en trouve simplifiée comme le montre le tableau suivant dans lequel $Compt_f$ et deg_f (resp. $Compt_i$ et deg_i) représentent le nombre de caractères ASCII pour les coefficients et le degré du polynôme dans le cas d’un point fini (resp. à l’infini).

Type de quadrique.	$Compt_f$	deg_f	$Compt_i$	deg_i
paraboloïde de révolution	1898	5	497	4
hyperboloïde	5190	6	2364	6
ellipsoïde	80468	8	22111	8
cône droit	697	6	695	6

FIG. 4.23 – Résultats comparatifs pour les réflexions sur différentes surfaces dans le cas où la caméra est alignée avec l’axe de révolution de la quadrique.

4.6 Applications

Le calcul de la tangence des quadriques est maintenant utilisé dans le cadre d'une application de vision. Dans un premier temps, la projection d'un point dans l'image après réflexion est étudiée, en utilisant une caméra modélisée par un trou d'épingle. Ensuite, l'image du contour du miroir est calculée. Le calcul de la projection d'un point est utilisé pour projeter des (segments de) droites puis pour calculer une reprojexion de texture. Finalement, il sert à l'étalonnage de la caméra catadioptrique.

4.6.1 Projection d'un point

4.6.1.1 La caméra trou d'épingle

Considérons maintenant la projection dans l'image de la réflexion d'un point P_T exprimé dans un repère de travail orthonormal ω_T . La figure 4.24 montre les différents repères utilisés.

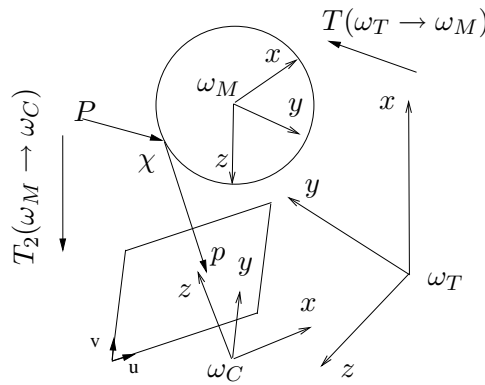


FIG. 4.24 – Projection d'un point P de la scène en p dans l'image.

La quadrique du miroir, dans sa forme simplifiée, est définie dans le repère orthonormal associé ω_M . Soit $S = \text{diag}(s, s, s, 1)$ la matrice de mise à l'échelle entre ω_T et ω_M . Soit t_T et R_T la translation et la matrice de rotation du repère ω_T vers le repère lié au miroir ω_M . La matrice $T(\omega_T \rightarrow \omega_M)$ modélisant le changement de repère entre ω_T et ω_M et permettant d'obtenir les coordonnées du point $P_{(\omega_M)}$ est donnée par l'équation (4.39) :

$$T(\omega_T \rightarrow \omega_M) = S \begin{bmatrix} R_T & -R_T^T t_T \end{bmatrix}; P_{(\omega_M)} = T(\omega_T \rightarrow \omega_M) P_{(\omega_T)} \quad (4.39)$$

Soit Γ la fonction telle que $\chi_{(\omega_M)} = \Gamma(P_{(\omega_M)}, C_{(\omega_M)})$, avec $C_{(\omega_M)} = [x_C y_C z_C]^T$ (resp. $\chi_{(\omega_M)}$) la position de la caméra (resp. la position du reflet à la surface du miroir) exprimée dans ω_M . $\chi_{(\omega_M)}$ n'est pas forcément unique car il peut y avoir plusieurs réflexions. Seules les $\chi_{(\omega_M)}$

visibles depuis C doivent être considérés grâce à un choix adéquat des racines ξ . Ceux qui ne sont pas visibles sont détectés comme énoncé précédemment.

La caméra est de type trou d'épingle, telle que décrite au [chapitre 1](#). Elle définit un repère ω_C centré en le point C dont les axes x et y sont alignés avec les axes des colonnes et des lignes u et v du plan image.

Soit $t_C = -[x_C y_C z_C]^T$ et R_C la translation et la matrice de rotation du repère lié au miroir ω_M vers le repère lié à la caméra ω_C , concaténés dans la matrice de transformation $T_2(\omega_M \rightarrow \omega_C)$:

$$T_2(\omega_M \rightarrow \omega_C) = [R_C \quad -R_C^T t_C] \quad (4.40)$$

Soit K la matrice caméra (définie par l'équation (1.2) du [chapitre 1](#)). La projection dans l'image p du point $\chi_{(\omega_M)}$ s'obtient par l'équation (4.41). Les coordonnées cartésiennes de p sont $[u/w, v/w]^T$.

$$p = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = K [R_C \quad -R_C^T t_C] \chi_{(\omega_M)} \quad (4.41)$$

Ceci permet d'établir l'expression complète de la projection de la (ou des) réflexion(s) du point P_t dans l'image :

$$p = K [R_C \quad -R_C^T F] \Gamma(S [R_T \quad -R_T^T t_T] P_{(\omega_T)}, C_{(\omega_M)}) \quad (4.42)$$

4.6.1.2 Obtention du contour de la quadrique projetée

Cas de la sphère : la sphère est une quadrique très facile à projeter car son image est celle d'un cercle et à donc la forme d'une ellipse. La sphère et le centre optique de la caméra définissent un cône Q_C tangent à la sphère dont l'axe de révolution est représenté par z sur la figure 4.25 (a). Nous considérons dans la suite, afin de décrire une exemple simple, que le centre de la sphère est situé sur l'axe z_C passant par le centre optique C . Soit L un point paramétré par l'angle α sur le cercle dans le quadrant inférieur droit. Les coordonnées du point L sont :

$$L = [x_L \ y_L] = [\cos(\alpha) \ \sin(\alpha)]^T, \quad 0 \leq \alpha < \pi/2 \quad (4.43)$$

Soit $dist$ l'abscisse de l'image d'un point du cercle (dans le plan perpendiculaire à z et à distance unitaire du centre optique de la caméra, ce plan n'étant pas le plan image de la caméra), on recherche L tel que $dist$ soit maximum, quand $\frac{d \ dist}{d \alpha}$ s'annule :



FIG. 4.25 – (a) Projection du contour d’une sphère dans l’image. (b) Projection du contour d’une quadrique dans l’image. Le plan Π_Γ est le plan polaire de C par rapport à la quadrique Q .

$$dist = 1 \frac{y_L}{|z_C| - y_L} \quad (4.44)$$

$$\frac{d \, dist}{d\alpha} = \frac{|z_C| \sin(\alpha) - 1}{z_C^2 - 2 |z_C| \sin(\alpha) + 1 - \cos^2(\alpha)} \quad (4.45)$$

$$\frac{d \, dist}{d\alpha} = 0 \iff |z_C| \sin(\alpha) - 1 = 0 \iff \alpha = \arcsin(1/|z_C|) \quad (4.46)$$

L’expression de $Q_{C(\omega_{M'})}$ dans un repère centré sur C et dont les axes sont orientés selon ω_M est $Q_{C(\omega_{M'})} = \text{diag}(1, 1, -dist^2, 0)$. Pour obtenir $Q_{C(\omega_C)}$, il faut effectuer la rotation par R_C . Soit R_{CH} la matrice de transformation en coordonnées homogènes correspondant à la rotation R'_C , $Q_{C(\omega_C)}$ s’obtient par :

$$R_{CH} = \begin{pmatrix} R'_C & 0 \\ 0 & 1 \end{pmatrix} \quad (4.47)$$

$$Q_{C(\omega_C)} = R_{CH} Q_{C(\omega_{M'})} R'_{CH} \quad (4.48)$$

Le calcul de l’intersection de $Q_{C(\omega_C)}$ avec le plan image normalisé $z = 1$ dans le repère ω_C est effectué en gardant la matrice 3*3 du coin haut gauche, on obtient la conique définie par la matrice C_1 :

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.49)$$

$$C_1 = V Q_{C(\omega_C)} V' \quad (4.50)$$

L’image C_2 de C_1 dans le plan image de la caméra est obtenue par :

$$C_2 = K^{-T} C_1 K^{-1} \quad (4.51)$$

Cas général : Il est possible d'exploiter la notion de plan polaire à une quadrique par rapport à un point pour obtenir l'image du contour d'une quadrique.

Sur la figure 4.25 (b) est visible un exemple avec un ellipsoïde. Le point C et la quadrique Q définissent un plan Π_Γ appelé polaire de Q par rapport au point C . $\Pi_\Gamma = Q C$ et intersecte Q en une conique Γ , génératrice d'un cône Q_C ayant pour sommet le point C . L'équation de ce cône est :

$$Q_C = (C^T Q C) Q - (Q C)(Q C)^T \quad (4.52)$$

L'intersection de ce cône avec le plan image de la caméra conduit à une conique dans l'image.

4.6.2 Projection d'une droite ou d'un segment

A défaut d'une formule directe pour projeter une droite, il est nécessaire de projeter des points de celle-ci, en réalisant un échantillonnage adéquat. Nous nous intéressons ici particulièrement à l'obtention de l'image d'une demi droite dans l'espace, qui peut par exemple être un rayon de projection correspondant à l'observation d'un point de la scène sur l'image d'une caméra. La projection de cette demi droite dans l'image d'une autre caméra (ou la même déplacée) permet de définir une courbe de correspondance stéréoscopique, sur laquelle on peut restreindre la recherche de l'image du même point de la scène.

4.6.2.1 Pour un miroir sphérique convexe

Considérons dans un premier temps un miroir sphérique convexe. Il a été montré que ce type de miroir génère une réflexion unique vers la caméra (une seule des quatre racines ξ est négative et réelle et conduit à un trajet ne passant pas par l'intérieur de la sphère). La figure 4.26 (a) montre un exemple de configuration géométrique. Seule une portion de la demi droite est représentée, la demi droite s'étendant à l'infini vers le haut de la figure. La figure 4.26 (b) montre l'image obtenue par la caméra. Le contour de la sphère est obtenu par la méthode spécifique à la sphère décrite précédemment. Les points de l'image de la demi droite sont quand à eux obtenus grâce à la méthode présentée dans le paragraphe suivant.

4.6.2.2 Échantillonnage des points de la demi droite

Voici l'algorithme permettant de générer l'image de la demi droite D définie par les points $P(t) = \text{Origine} + t \cdot \text{Direction}$. Soit $\text{Proj}(t)$ la fonction qui calcule la projection d'un point de la droite sur l'image. Un seuil d_{max} est défini afin de régler la distance maximale autorisée entre deux points voisins sur l'image de D . Le principe de l'échantillonnage est d'ajouter des points jusqu'à ce que l'image obtenue soit suffisamment dense, c'est à dire lorsque l'écart entre chaque couple de points voisins dans l'image est inférieur ou égal à d_{max} .

Liste contient les coordonnées u et v dans l'image des différents points ainsi que la valeur de t qui leur correspond et un booléen indiquant si le point est visible (pas d'occultation par

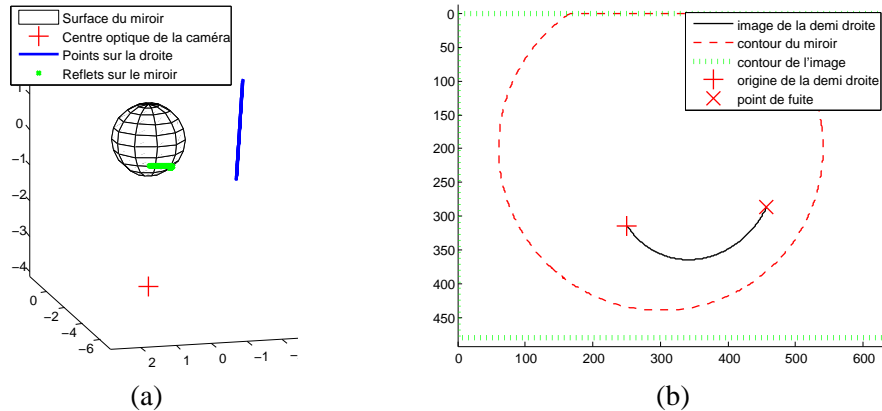


FIG. 4.26 – (a) Le reflet d'une demi droite sur le miroir sphérique. (b) L'image correspondante.

le miroir). Cette liste est ordonnée par valeurs croissantes de t . Les projections des points sont calculées indépendamment de leur visibilité et celle ci est gérée à posteriori lors de l'affichage.

Algorithme de projection d'une demi droite :

```

Liste = [Proj(0), Proj(1)], fin=0
Tant que fin==0
  Pour i=1 à taille(Liste)-1
    Si distance dans l'image ( Liste(i)(u,v), Liste(i+1)(u,v))>dmax
      np = Proj((Liste(i)(t) + Liste(i+1)(t))/2)
      Insérer np dans la liste entre les éléments i et i+1
      np = Proj((Liste(taille(Liste))(t) * 2 + 1))
    Si distance dans l'image ( Liste(taille(Liste))(u,v), np(u,v))>dmax
      Insérer np à la fin de Liste
    Si aucun point n'a été ajouté à cette itération
      ou qu'un nombre trop important d'itérations a été effectué
      fin=1
  np = Proj(∞)
  Insérer np à la fin de Liste
  Afficher les points de Liste qui sont dans l'image et qui sont visibles

```

La norme de *Direction* doit être suffisamment grande pour ne pas provoquer l'arrêt de l'exécution à la première itération, si aucun point n'a été ajouté. Si c'est le cas, $Direction = Direction * 10$ jusqu'à ce qu'il n'y ait plus de blocage.

Sur les figures 4.27 et 4.28, sont visibles les différentes itérations conduisant à l'image de la figure 4.26 (b). Jusqu'à l'itération 6, il y a doublement du nombre de points à chaque itération. Ensuite, des points sont ajoutés uniquement où il y en a besoin. La dernière itération correspond à l'ajout du point de fuite, calculé à partir de l'intersection du paraboloïde de révolution avec

la sphère.

Projection d'une droite : Pour projeter une droite complète, il suffit d'ajouter des points des deux côtés de la liste à chaque itération en générant des points sur D pour lesquels $t < 0$. Un point de fuite devra alors être généré pour chaque direction.

Projection d'un segment : Pour projeter un segment, il suffit d'initialiser la liste avec les deux extrémités du segment et de ne pas ajouter des points des deux côtés de la liste à chaque itération, ni de points de fuite à la fin.

4.6.2.3 Pour un miroir présentant plusieurs réflexions

Dans le cas où la quadrique peut faire apparaître plusieurs réflexions pour un même point de la droite, l'algorithme doit être adapté pour gérer séparément les différents reflets de D . La figure 4.29 montre un exemple avec un miroir en forme de paraboloïde hyperbolique. La demi droite étudiée se prolonge vers le haut de la figure. En (a) et (b) est visible la configuration de l'ensemble depuis deux points de vue différents. En (c) est montrée l'image de la demi droite observée par la caméra.

4.6.2.4 Approximation par Splines

Les images de droites sont assez bien approximées par de Splines cardinaux (Schumaker [156] et de Boor [157]) passant par leurs points de contrôle. Grâce à elles, il n'est pas forcément nécessaire d'effectuer les itérations jusqu'à obtenir un écartement des points projetés dans l'image suffisamment petit, l'interpolation par Splines fournissant les coordonnées de nouveaux points entre des points projetés. Il est possible d'adapter l'algorithme proposé pour tenir compte de cette propriété. Ainsi, partant d'une liste contenant les projections des deux extrémités de la primitive à projeter (des points finis ou à l'infini en fonction du caractère de la primitive, droite, demi droite ou segment), l'ajout de nouveaux points se fait comme précédemment. A chaque itération, une Spline est calculée à partir des points projetés. À l'itération suivante, la distance des nouveaux points à la Spline est calculée et s'ils sont tous suffisamment proches, il est possible d'arrêter les itérations ; la Spline approxime alors suffisamment bien l'image de la primitive projetée.

Sur l'exemple de la figure 4.30, reprenant l'itération 2 de la figure 4.27, il est visible que les points créés à l'itération 3 sont très proches de la Spline obtenue à partir des points créés à l'itération 2. Il est alors possible d'approximer cette courbe par une spline avec 9 points de contrôle au lieu de nécessiter le calcul de la projection des 361 points suivants.

4.6.3 Reprojection de texture

La méthode pour le calcul de la projection d'un point est utilisée ici pour reconstruire une texture plane. L'image de la figure 4.31 (b) est appliquée sur une surface plane et l'image (a) est

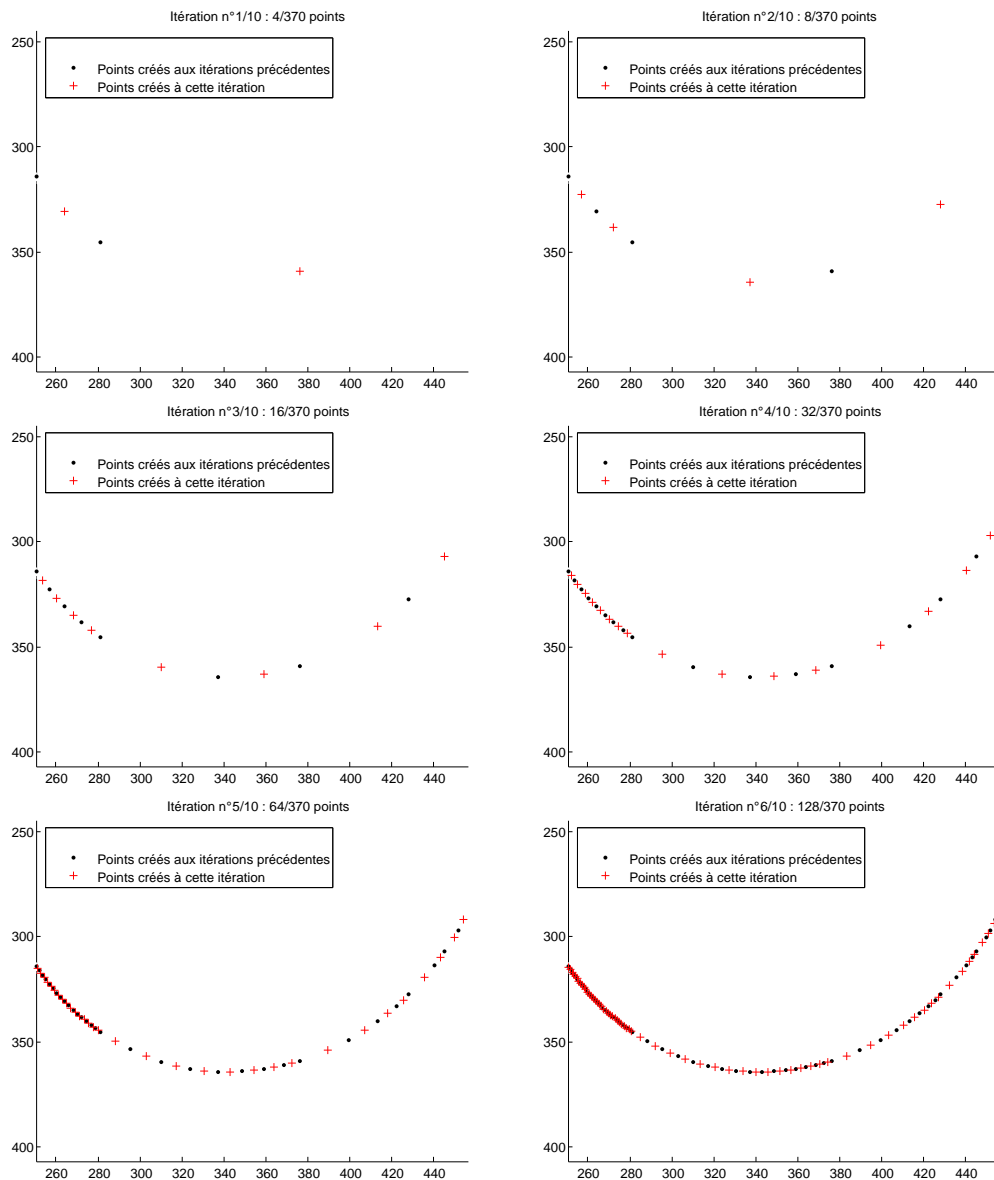


FIG. 4.27 – Itérations 1 à 6 de la projection des points de la demi droite.

synthétisée par raytracing. Cette image correspond au reflet de la scène présentée sur la figure (c), observée après réflexion dans le miroir sphérique.

Le positionnement de la surface plane est ici connu puisque l'image est synthétisée. Dans une application réelle, il serait nécessaire de le déterminer, soit par étalonnage, soit par l'observation des quatre coins suivie d'une estimation de pose par exemple.

La surface dont on souhaite reconstruire la texture est échantillonnée et les points correspondants sont projetés dans l'image observée par la caméra pour déterminer leurs couleurs. La

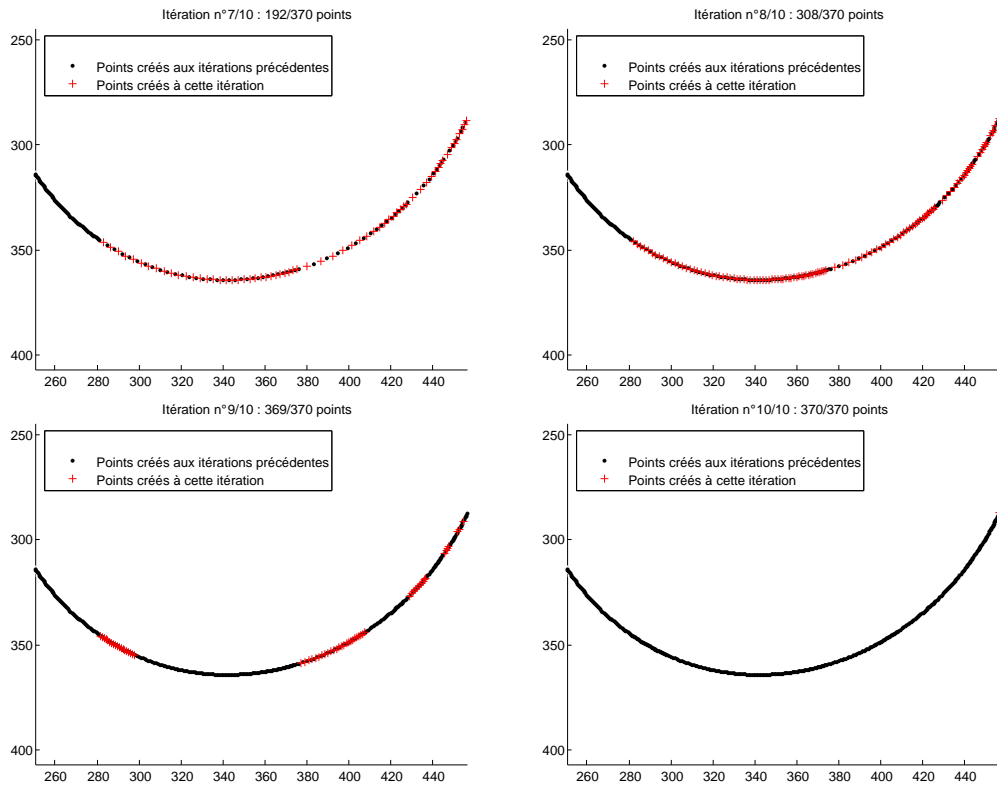


FIG. 4.28 – Itérations 7 à 10 de la projection des points de la demi droite.

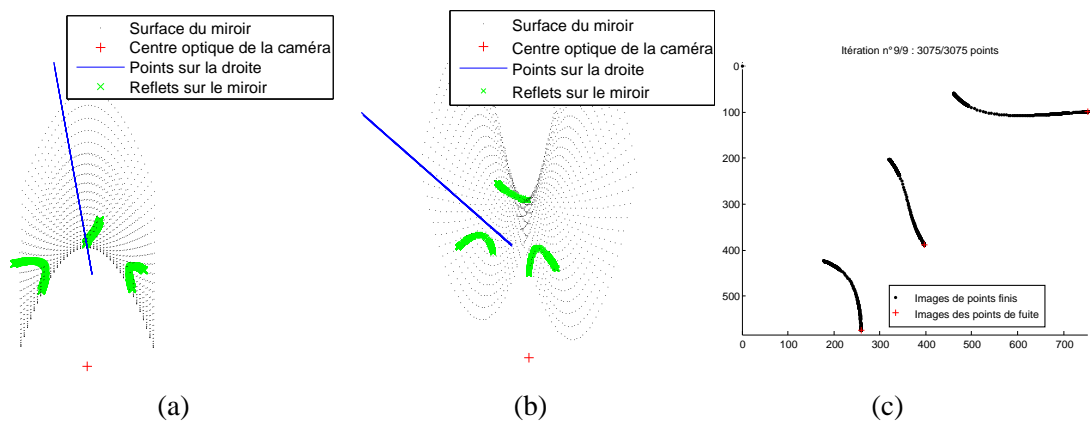


FIG. 4.29 – (a) et (b) Même configuration d’un miroir en forme de paraboloidé hyperbolique et d’une demi droite observée depuis deux points de vue différents. (c) L’image obtenue.

figure (c) représente un échantillonnage du plan sur lequel la texture d’origine a été appliquée de 10×10 points. L’image (d) est reconstruite à partir d’un échantillonnage plus fin de 500×500

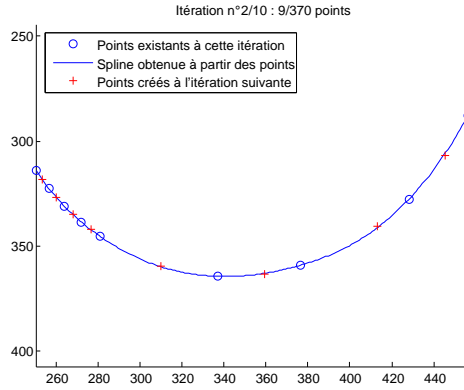


FIG. 4.30 – Approximation par une Spline de l'image d'une demi droite.

points, correspondant à la résolution de l'image (b). Celle de l'image (a) est de 640*480. Certains détails de la texture d'origine ont été perdus lors de la synthèse de l'image (a) et il est donc normal qu'ils n'apparaissent pas sur l'image (d). Les bandes ajoutées autour du logo permettent de vérifier la rectitude de la reprojection.

On pourra noter la luminosité variable sur l'image reconstruite, due à l'éclairage utilisé pour la synthèse de l'image (a).

Finalement, cette méthode peut être adaptée au calcul de la texture d'une surface quelconque. Il suffit pour cela de pouvoir échantillonner des points sur cette surface et de les projeter dans l'image comme dans le cas d'une surface plane.

4.6.4 Etalonnage géométrique

Comme il a été montré au [chapitre 1](#), l'étalonnage géométrique (pour un modèle paramétrique) d'une caméra consiste en la détermination des paramètres d'un modèle de la caméra. Considérant la caméra fixe par rapport au miroir, voici la liste des onze paramètres intrinsèques du modèle de projection d'un point déjà présenté.

- La caméra en trou d'épingle est paramétrée par $\alpha_u, \alpha_v, p_u, p_v$.
- R_C est paramétrée par trois angles d'Euler (précession Ψ_C , nutation Θ_C et rotation propre φ_C).
- La position de la caméra dans ω_M est paramétrée par ses trois coordonnées x_C, y_C et z_C .
- Le facteur d'échelle entre le repère miroir et le repère de travail est paramétré par un facteur s .

Les paramètres extrinsèques sont quant à eux au nombre de six et correspondent à :

- R_T est paramétrée par trois angles d'Euler (Ψ_T, Θ_T et φ_T).
- t_T est paramétré par trois coordonnées t_x, t_y et t_z .

Soit u_o et v_o les coordonnées des observations de projection des points d'une mire sur l'image. L'étalonnage consiste à trouver le jeu de paramètres \mathbf{x} qui minimise l'erreur de repro-

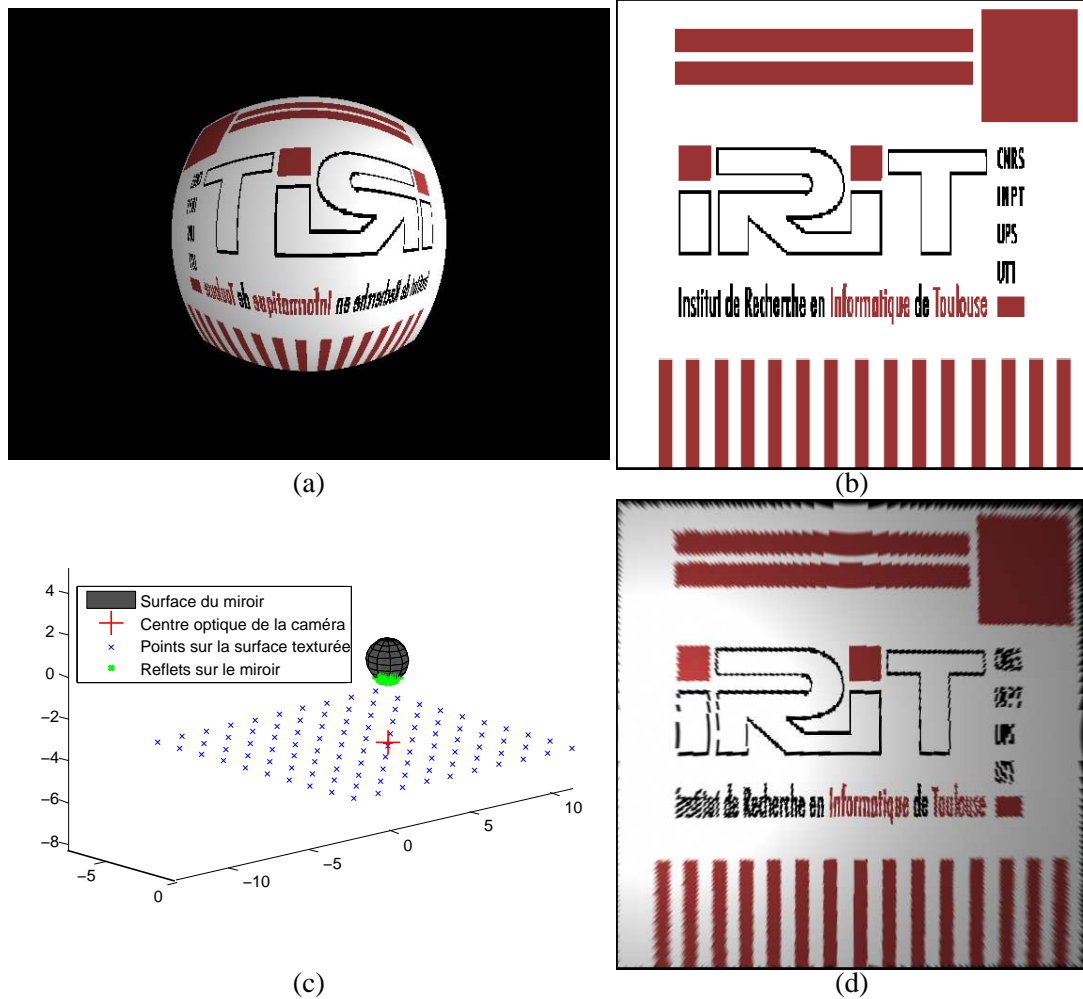


FIG. 4.31 – (a) Le reflet sur un miroir sphérique de l’image (b) observé par une caméra perspective. (c) Configuration géométrique et échantillonnage de points à la surface contenant la texture. (d) La texture reconstruite d’après l’image (a).

jection des points i de la ou des mires j , u et v étant les projections prédites par le modèles, données par l’équation (4.42) tel que :

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \sum_j \sum_i ((u_o - u)^2 + (v_o - v)^2) \quad (4.53)$$

La minimisation de ce critère doit être effectuée de manière itérative, en partant d’une solution initiale \mathbf{x}_0 proche de la véritable solution \mathbf{x}_r . Ce type d’optimisation fait intervenir une approximation de la hessienne à partir de la jacobienne. Étant dans l’incapacité d’obtenir

une expression analytique pour la jacobienne, une estimation est effectuée en chaque jeu de paramètres \mathbf{x} visité par calcul des différentes finies.

Par manque de temps, il n'a pas été possible d'investiguer profondément l'étalonnage d'une caméra réelle, et nous nous contentons donc d'une étude rapide sur des caméras de synthèse, qui permet de montrer sa faisabilité. La caméra catadioptrique testée comporte un miroir sphérique convexe, qui assure une réflexion unique des points d'étalonnage, plus simple à gérer que des réflexions multiples. Avec cette quadrique, il est possible de fixer x_C et y_C à 0. L'angle φ_C correspondant à une rotation autour de l'axe z de ω_C peut aussi être fixé à 0 grâce à la symétrie de révolution de la sphère (avec les angles d'Euler, la rotation autour de l'axe z est la première à être effectuée). Le rayon du miroir peut aussi être mesuré très précisément avec un pied à coulisse et donc s est connu. Ainsi, seuls sept paramètres intrinsèques doivent être estimés pour cette caméra.

4.6.4.1 Étalonnage à partir d'une mire 3D

Afin d'estimer au mieux les paramètres, la projection de la mire doit couvrir la plus grande surface possible dans l'image. À cause du champ visuel large de la caméra catadioptrique, ceci ne peut pas être atteint avec une mire plane unique. Nous proposons donc l'emploi d'une mire 3D dans laquelle la caméra est immergée, semblable à celle utilisée par Vasseur et Mouaddib dans [135]. Une seule observation de la mire suffit alors, et il y a donc un seul jeu de paramètres de pose de la mire à estimer, soit un total de $7 + 6$ paramètres.

La figure 4.32 (a) montre la position relative des points de la mire, de leurs reflets sur le miroir et de la caméra.

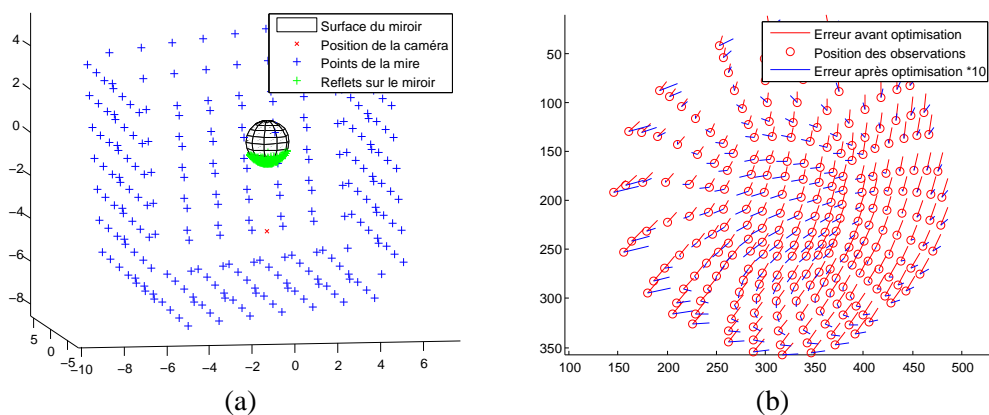


FIG. 4.32 – (a) Configuration de la mire 3D et de la caméra catadioptrique. (b) L'image obtenue, et la reprojction par le modèle avec les paramètres d'initialisation et les paramètres affinés.

Test effectué : Un jeu de paramètres \mathbf{x}_r a été choisi (voir la première colonne de la figure 4.33). Comme s est connu, il est considéré égal à 1 ici pour avoir une unité de longueur (ul)

identique pour tous les paramètres. Ensuite, cent étalonnages sont effectués de la façon suivante :

Les observations des points de la mire sont générées en projetant la mire avec les paramètres \mathbf{x}_r et en appliquant un bruit gaussien d'amplitude $\sqrt{2}/2$ pixels sur chacune de leurs coordonnées.

Les paramètres du modèle sont perturbés par un bruit de répartition uniforme et d'amplitude variable en fonction du paramètre (de l'ordre de huit degrés d'amplitude pour les angles, 50 pixels ou 0.2 unité de longueur) et un jeu \mathbf{x}_0 est obtenu. Ce jeu est utilisé pour initier l'optimisation itérative par l'algorithme de Levenberg Marquardt [158]. Il représente les paramètres qui seraient obtenus par une mesure manuelle peu précise des paramètres, que l'on peut supposer non biaisée.

L'optimisation correspondant à l'équation (4.53) est effectuée et le jeu de paramètres $\hat{\mathbf{x}}$ obtenu.

La figure 4.32 (b) montre la position des observations sur l'image de la caméra pour un étalonnage ainsi que l'erreur pour chaque point avant et après optimisation (l'erreur après optimisation est agrandie dix fois pour être visible).

Parmi les cent résultats d'étalonnage, sept sont supprimés car ils présentent une somme de résidus trop importante, à cause d'une convergence vers un minimum local trop éloigné du jeu de paramètres \mathbf{x}_r . Le tableau 4.33 montre les valeurs moyennes et les écarts types pour les paramètres obtenus avant (\mathbf{x}_0) et après ($\hat{\mathbf{x}}$) optimisation.

Paramètre.	\mathbf{x}_r	\mathbf{x}_0	$\hat{\mathbf{x}}$
α_u (pix)	905.62	905.40 ± 8.16	905.88 ± 8.61
α_v (pix)	905.62	906.49 ± 8.75	905.93 ± 8.31
p_u (pix)	320.00	319.41 ± 8.03	319.36 ± 7.66
p_v (pix)	240.00	239.80 ± 8.36	239.66 ± 8.03
z_C (ul)	-4.1300	-4.1304 ± 0.0268	-4.1324 ± 0.0457
Ψ_C (rad)	0.0500	0.0543 ± 0.0416	0.0494 ± 0.0133
Θ_C (rad)	0.0120	0.0216 ± 0.0421	0.0131 ± 0.0128
Ψ_T (rad)	-0.1571	-0.1595 ± 0.0431	-0.1578 ± 0.0042
Θ_T (rad)	0.1013	0.0981 ± 0.0407	0.1007 ± 0.0046
φ_T (rad)	-0.1164	-0.1183 ± 0.0457	-0.1173 ± 0.0060
t_x (ul)	-0.4155	-0.4144 ± 0.0557	-0.4161 ± 0.0593
t_y (ul)	-0.0321	-0.0258 ± 0.0560	-0.0259 ± 0.0608
t_z (ul)	-7.1234	-7.1250 ± 0.0625	-7.1250 ± 0.0702
$\sum res^2$ (pix ²)	242.43 ± 17.26	388991 ± 246465	616.21 ± 391.92
$moy(res)$ (pix)	0.8824 ± 0.0331	36.4607 ± 12.7925	1.3531 ± 0.3771

FIG. 4.33 – Résultats de l'étalonnage : Les différents paramètres du modèle, la valeur du critère à minimiser et la moyenne des résidus pour \mathbf{x}_r : les paramètres choisis ; \mathbf{x}_0 : les paramètres perturbés ; $\hat{\mathbf{x}}$: les paramètres estimés en partant de \mathbf{x}_0 .

Les valeurs moyennes de \mathbf{x}_0 sont proches de celles de \mathbf{x}_r car le bruit appliqué aux paramètres est de moyenne nulle. C'est donc plutôt la dispersion des valeurs qu'il faut étudier, représentée sous la forme d'un écart type.

La première constatation à faire est que l'optimisation a bien rempli son rôle, en minimisant la somme des carrés des résidus. Même si celle-ci n'atteint pas le minimum global censé être situé en \mathbf{x}_r , sa valeur est relativement proche par rapport à celle de départ. Ramené à une erreur moyenne en pixels dans l'image, l'optimisation permet d'obtenir une erreur moyenne de l'ordre de 1.3 pixels, alors que le bruit gaussien ajouté aux données génère 0.8 pixels d'erreur en moyenne. Ces valeurs doivent être comparées au plus de 36 pixels d'erreur moyenne associée au jeu de paramètres \mathbf{x}_0 et montrent que l'erreur est fortement réduite durant la minimisation.

En regardant attentivement les valeurs des paramètres $\hat{\mathbf{x}}$ obtenus, il est possible de faire quelques remarques :

L'optimisation a largement amélioré les valeurs de Ψ_C , Θ_C , Ψ_T , Θ_T et φ_T en rapprochant leurs valeurs moyennes de leurs valeurs idéales et en diminuant les écarts types. Par contre, les autres paramètres ne présentent pas d'amélioration notable pendant l'optimisation, et pourtant l'erreur pixellique diminue.

Ceci peut traduire une corrélation entre les différents paramètres. Ainsi, par exemple, α_u et α_v sont corrélés avec z_C , un éloignement de la caméra ayant un effet très proche d'une augmentation de la focale. Les paramètres p_u et p_v sont quant à eux corrélés avec les paramètres de position de la mire. Ce phénomène se caractérise par de très faibles déplacements sur certains de ces paramètres, l'optimisation préférant se diriger dans la direction de descente associée à des paramètres qui leur sont corrélés.

Une autre raison qui expliquerait ce phénomène provient de l'ordre de grandeur des différents paramètres. L'incrément calculé par la méthode de Levenberg-Marquardt est ainsi correct sur les paramètres de l'échelle d'un angle mais est très faible pour un paramètre de l'échelle d'une focale. Lors d'une itération, la variation d'angle est donc privilégiée au détriment d'une variation de focale. Ce problème devrait être résolu par une normalisation des paramètres.

4.6.4.2 Étalonnage à partir de plusieurs mires planes

L'étalonnage est aussi possible en utilisant une mire plane sur laquelle figurent des points. La mire est observée dans différentes poses afin de couvrir au mieux la surface du plan image après projection. La figure 4.34 (a) montre un exemple avec cinq acquisitions de la mire, dans une configuration proche de celle la mire 3D déjà étudiée. La figure 4.34 (b) montre un exemple d'image obtenue. La différence par rapport à l'étalonnage utilisant la mire 3D est qu'ici il est nécessaire d'estimer la pose des différentes mires. Ainsi au lieu de devoir estimer $7 + 6$ paramètres, ce sont $7 + 6n$ paramètres qui doivent être retrouvés, n étant le nombre de positions différentes pour la mire.

Cette augmentation du nombre de degrés de libertés induit différents problèmes. Premièrement, la recherche de \mathbf{x} se fait dans un espace de dimension plus grand, et le calcul de la jacobienne est plus lourd. Deuxièmement, les problèmes de corrélation entre les paramètres sont toujours présents et conduisent ici à un nombre très élevé d'itérations due à une conver-

gence très lente. Enfin, il ne faut pas oublier que nous supposons les poses des mires connues approximativement a priori, ce qui sous entend une mesure manuelle à effectuer. Un nombre élevé de poses implique par conséquent un nombre élevé de mesures, ce qui est contraignant.

Les expériences sur les données de synthèse ont montré qu'il est préférable d'utiliser une mire 3D. Ceci ne tient néanmoins pas compte du fait qu'en pratique, une mire 2D peut être fabriquée avec précision beaucoup plus facilement qu'une mire 3D.

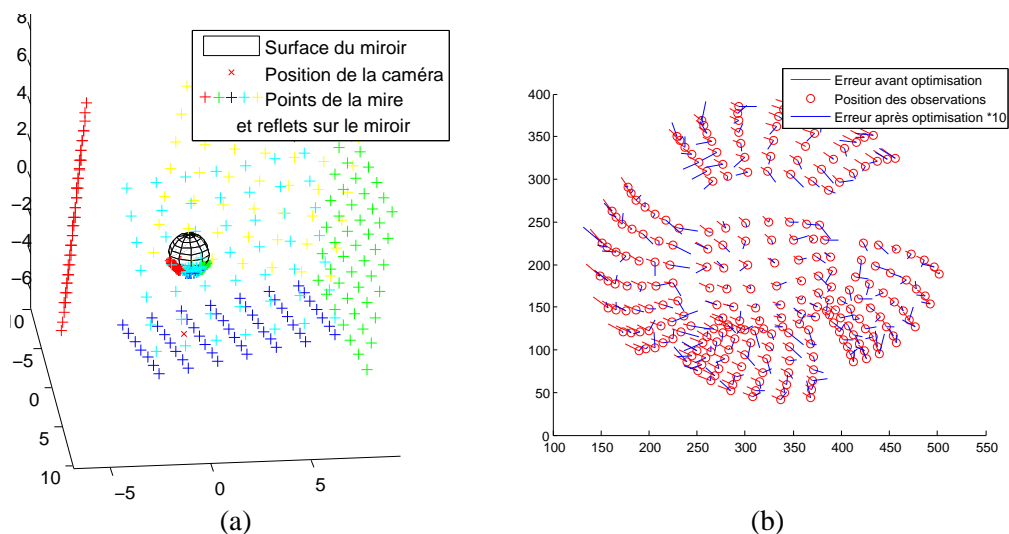


FIG. 4.34 – (a) Configuration des cinq mires 2D et de la caméra catadioptrique. (b) L'image obtenue, et la reprojexion par le modèle avec les paramètres d'initialisation et les paramètres affinés.

4.7 Conclusions

Ce chapitre a proposé une solution à la détermination de l'image d'un point après réflexion sur un miroir en forme de quadrique. En développant une formulation de la réflexion utilisant un ellipsoïde (pour des points à des positions finies) ou un parabolôïde de révolution (pour des points à l'infini, représentant des directions) paramétré, il a été possible d'exprimer une équivalence entre une condition de tangence entre deux quadriques et la réflexion sur une quadrique. La détermination de la position de la ou des réflexions est effectuée par la recherche des racines d'un polynôme de degré variable, en fonction de la quadrique sur laquelle a lieu la réflexion.

Cette approche présente de nombreux avantages par rapport aux alternatives existantes :

- Comparée à la méthode présentée en 4.1.3.1, utilisant le modèle inverse et une minimisation de distance en espace continu, elle ne nécessite pas une estimée initiale fournie par un modèle central approché de la caméra, ne présente pas de problème de convergence lié à la minimisation et est capable de traiter les réflexions multiples.

- Comparée à la méthode présentée en 4.1.3.2, utilisant le modèle inverse et une minimisation de distance en espace discret, elle fournit une solution bien plus rapidement et n'est pas limitée en précision par le pas d'échantillonnage du plan image de la caméra.

En contrepartie, les expressions des coefficients des polynômes dont les racines sont recherchées étant assez longues, il est nécessaire de prendre des précautions lors du calcul des valeurs numériques.

Différents types de quadriques ont été étudiés, et le degré et la complexité des polynômes correspondants ont été comparés. Le cas particulier des caméras axiales a été discuté et mène à des polynômes dont les expressions sont plus simples.

De multiples applications ont été proposées, notamment pour la reconstruction 3D de géométrie et de texture. Des travaux préliminaires sur l'étalonnage géométrique ont été présentés en utilisant des données de synthèse.

4.7.1 Conclusions sur l'étalonnage

Par manque de temps, il n'a pas été possible d'approfondir l'étude de l'étalonnage géométrique et notamment de comparer l'approche proposée avec celle de Micusik et al. Néanmoins, grâce au cadre proposé pour l'obtention de l'image d'un point après réflexion sur une quadrique, il a été possible de minimiser une erreur de reprojection dans l'image, ce qui est préférable, comme le note Lhuillier dans [90].

L'étalonnage fera l'objet de recherches ultérieures, notamment avec une caméra réelle. Comme le principal problème rencontré provient de la corrélation entre les paramètres du modèle, un effort sera porté sur la dissociation de ceux-ci, par exemple en réalisant tout d'abord un étalonnage de la caméra seule, suivi d'un étalonnage des paramètres de pose relative au miroir, par exemple en observant le contour de celui-ci. L'étape de raffinement global, telle que proposée ici, devra donc être appliquée ensuite à partir de paramètres très proches de leur valeurs réelles.

L'étalonnage à partir d'une scène non contrainte, tel que proposé par Lhuillier dans [90] ou Micusik et al dans [52], peut aussi exploiter la méthode de projection des points dans l'image proposée, l'ajustement de faisceaux étant alors effectué en minimisant l'erreur de reprojection.

Chapitre 5

Conception et réalisation d'un capteur 3D omnidirectionnel catadioptrique

Ce chapitre décrit le développement d'un capteur permettant l'acquisition d'un modèle 3D texturé d'une scène à partir d'images de celle-ci. Ce capteur est constitué d'une caméra paracatadioptrique, mise en rotation afin que son centre optique se déplace et que l'intégralité de la scène soit observée depuis la position du capteur. Le chapitre décrit tout d'abord le capteur, en 5.1, et compare son principe avec celui d'autres capteurs 3D existants.

Ensuite, une méthode d'étalonnage est proposée en 5.2. La reconstruction de modèle 3D est étudiée en 5.3, en utilisant la géométrie épipolaire, dans des images rectifiées ou non. Des résultats de reconstruction automatique et une application permettant la reconstruction 3D interactive sont montrés. Des critiques sont finalement formulées en 5.4 et des améliorations possibles sont décrites.

5.1 Description du capteur

Il existe une multitude de moyens différents pour acquérir un modèle 3D d'une scène.

Une simple caméra standard en mouvement peut être utilisée pour cette tâche, comme l'attestent les travaux de Pollefeys [159], mais une telle approche requiert des calculs lourds pour le traitement des images. Les travaux plus récents de Snavely et al. [160] ont ainsi permis la réalisation de l'application Photo Tourism, permettant la reconstruction de modèles de scènes 3D à partir de photos collectées sur Internet puis la navigation interactive entre les différentes photos.

D'un autre côté, un système de triangulation par mesure de temps de vol, employant le laser par exemple, permet un traitement aisé des données et l'obtention d'un modèle 3D très précis mais au prix d'un matériel plus complexe.

L'approche proposée ici se place entre ces deux extrêmes, en réalisant un compromis entre la complexité du matériel et du logiciel.

5.1.1 Principe

Le choix s'est porté sur une caméra paracatadioptrique, fixée sur une platine mise en rotation de manière à ce que le centre optique de la caméra décrive un cercle horizontal. Cette mécanique relativement simple permet d'acquérir des images automatiquement depuis des points de vue différents. Le centre optique de la caméra catadioptrique étant situé approximativement à 25 cm de l'axe de rotation, il y a 50 cm d'écart intra-oculaire entre deux prises de vues opposées, ce qui, associé au fait que plusieurs images sont utilisées, permet d'espérer une précision assez bonne sur la reconstruction de la profondeur. Dans la suite, sauf mention contraire, vingt images sont acquises à incrément angulaire constant lors d'une rotation complète de la platine.

Comme montré au [chapitre 3](#), la caméra paracatadioptrique a un champ visuel omnidirectionnel amputé de deux cônes. Lorsque cette caméra est mise en rotation, la seule zone de la scène qui reste toujours à l'intérieur de ces deux cônes est approximativement une sphère située au niveau de la platine, qui est une partie de la scène vide puisque le capteur y est placé. L'intégralité de la scène est donc observable, le capteur est donc omnidirectionnel.

Le capteur complet est visible sur la figure 5.1 (a). Il est équipé d'un trépied démontable et le tout peut être transporté dans une boîte de dimensions 28*21*61cm.

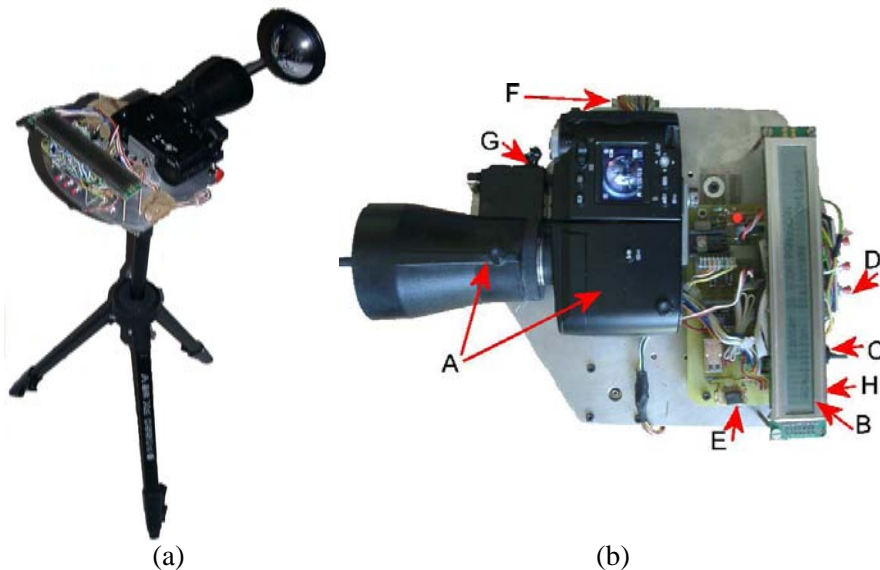


FIG. 5.1 – (a) Le capteur 3D et son trépied. (b) Gros plan sur la platine rotative : A l'appareil photo et l'optique télécentrique, B l'écran à cristaux liquides, C l'interrupteur général, D les 3 boutons de l'interface, E l'accéléromètre 2 axes, F la boussole numérique, G le servo moteur, H le port USB.

Les solutions les plus proches déjà proposées sont les suivantes :

- Kang, dans [78], présente un système comportant une caméra paracatadioptrique en rotation destinée à l'acquisition d'intérieurs de bâtiments. La caméra est orientée différem-

ment : son centre optique décrit un cercle mais la caméra est à l'extérieur de ce cercle. De plus l'axe optique de la caméra est orienté obliquement, ce qui ne permet pas d'avoir un champ de vision couvrant le sol. Ce système est basé sur l'analyse de vidéo, plutôt que d'images, et le capteur est donc équipé d'une caméra haute résolution pouvant acquérir jusqu'à douze images par seconde.

- Bakstein et Pajdla décrivent dans [161] un capteur (similaire à celui de Nayar et Karmarkar [18] p291) permettant la reconstruction 3D omnidirectionnelle assez similaire à celui proposé ici puisqu'il s'agit d'une caméra catadioptrique (miroir conique) ou fish-eye mise en rotation. La différence essentielle tient au fait que la caméra est utilisée pour percevoir uniquement une *tranche* de la scène dans chaque image. Cette tranche correspond à un plan, tangent à la trajectoire de la caméra et perpendiculaire au plan qui la contient. Avec cette approche, un grand nombre d'images (1800) doit être acquis lors d'une rotation du capteur.
- Micusik et Pajdla proposent, dans [52], la mise en rotation de caméras catadioptriques, afin d'avoir une mesure de la précision de leur méthode d'estimation de *Structure From Motion*. Dans l'un des tests effectués, ils utilisent une caméra paracatadioptrique, dans la même configuration que celle proposée ici, mais les auteurs n'ont pas noté les avantages d'un tel agencement.

Schum et al. décrivent, dans [162], le concept de *caméra virtuelle omnivergente*. L'image d'une telle caméra est obtenue par collecte de rayons lumineux captés par une caméra réelle en rotation autour d'un axe ne comprenant pas son centre optique. De bonnes propriétés sont exhibées, notamment au niveau de l'appariement dense, les images de la caméra virtuelle étant rectifiée épipolairement.

5.1.2 Avantages de ce type de capteur

- La mécanique du capteur et de l'optique sont relativement simples pour permettre une production et une maintenance facile.
- Le modèle 3D peut être acquis depuis une position unique du capteur pour éviter d'avoir à le déplacer. Il peut néanmoins être déplacé lorsque la scène le nécessite, de par ses dimensions ou ses formes, présentant des occultations et ne permettant pas l'acquisition depuis un point de vue unique.
- La texture de la scène est acquise en même temps que sa géométrie, afin de pouvoir par la suite synthétiser des images réalistes.
- Le champ de vision du capteur est complètement omnidirectionnel.
- Le modèle 3D obtenu est orienté, c'est à dire qu'il est exprimé dans un repère dont les axes sont contraints par rapport à la scène réelle. Ceci permet par la suite de générer des images dont l'orientation est connue.
- Le capteur est compact, ce qui permet l'acquisition de pièces exigües. Il permet aussi l'acquisition des images de manière autonome, sans nécessiter le transport d'un ordinateur ni la présence d'un opérateur (la personne manipulant l'appareil) lors des prises de vue, évitant ainsi des occultations de la scène. Enfin, il est facilement transportable et son coût n'est pas prohibitif.

5.1.3 La mécanique

La mécanique du capteur est principalement une platine en rotation par rapport à une base fixe. Cette platine est montée sur un axe équipé de deux roulements à billes. Elle est entraînée par un moteur pas à pas (de 400 pas par tour) dont l'axe s'appuie sur un joint torique en latex, monté dans la rainure d'un disque fixe, ce qui permet une démultiplication sans jeu pour obtenir à peu près 0.03 degré par pas du moteur. La figure 5.2 montre différentes vues du modèle CAO de la platine et de la base qui se montent sur le trépied. En (c), le moteur et son axe ainsi que le disque fixe sont visibles. En (d), le disque est effacé. En (e) et (f), on peut voir aussi la batterie avec son système de fixation et le disque fixe équipé du joint torique.

La platine a été dessinée afin d'être la plus compacte possible et pour occulter le moins possible le champ visuel de la caméra. Tous les éléments électriques sont fixés sur la platine, y compris la batterie, ce qui évite les problèmes de rotation avec les fils, tous les éléments pivotant en même temps. Les différents éléments sont positionnés pour équilibrer les masses autour de l'axe de rotation, ce qui est surtout crucial pour les éléments les plus lourds, la caméra et son optique ainsi que la batterie.

Enfin, on pourra noter que l'axe de rotation de la platine est approximativement aligné avec le centre optique de l'appareil photo numérique, ce qui permet, en retirant l'optique paracata-dioptrique, d'acquérir des images panoramiques par mosaïquage.

5.1.4 L'électronique

Comme le capteur doit être autonome, il est piloté par un microcontrôleur. Nous avons choisi un PIC18F252 de la société Microchip car il avait déjà été utilisé pour le capteur d'orientation du chapitre 2 et les outils de développement étaient donc disponibles.

Ce microcontrôleur est utilisé pour gérer l'interface utilisateur du scanner, pilotable à l'aide des trois boutons-poussoirs (en D sur la figure 5.1 (b)) et affichant les informations et menus sur l'écran à cristaux liquide (en B). La mise en marche du capteur est effectuée via l'interrupteur général (en C). Un servomoteur de modélisme (en G) est utilisé pour commander la mise au point et la prise de vue des photographies en appuyant sur le bouton de déclenchement de l'appareil photographique (en A). Enfin, deux capteurs sont utilisés pour mesurer l'orientation du capteur, un double accéléromètre (en E) et une boussole numérique (en F).

Le double accéléromètre remplit le rôle d'inclinomètre et permet à l'opérateur de régler le trépied dans une configuration telle que la platine rotative soit horizontale. Il équivaut donc à un niveau à bulle numérique deux axes, l'inclinaison étant affichée sur l'écran en temps réel. La boussole numérique, sous la forme d'un module CMP03 commercialisé par la société DEVANTECH, permet de commencer l'acquisition des séries d'images en ayant toujours la même orientation. Ce module fonctionne uniquement quand il est tenu dans le plan horizontal, ce qui est le cas ici grâce au réglage préliminaire de l'inclinaison du capteur.

Le microcontrôleur commande également un relais, pilotant l'alimentation d'un appareil externe, tel un projecteur laser, un ensemble de diodes électroluminescentes (LED) ou autre.

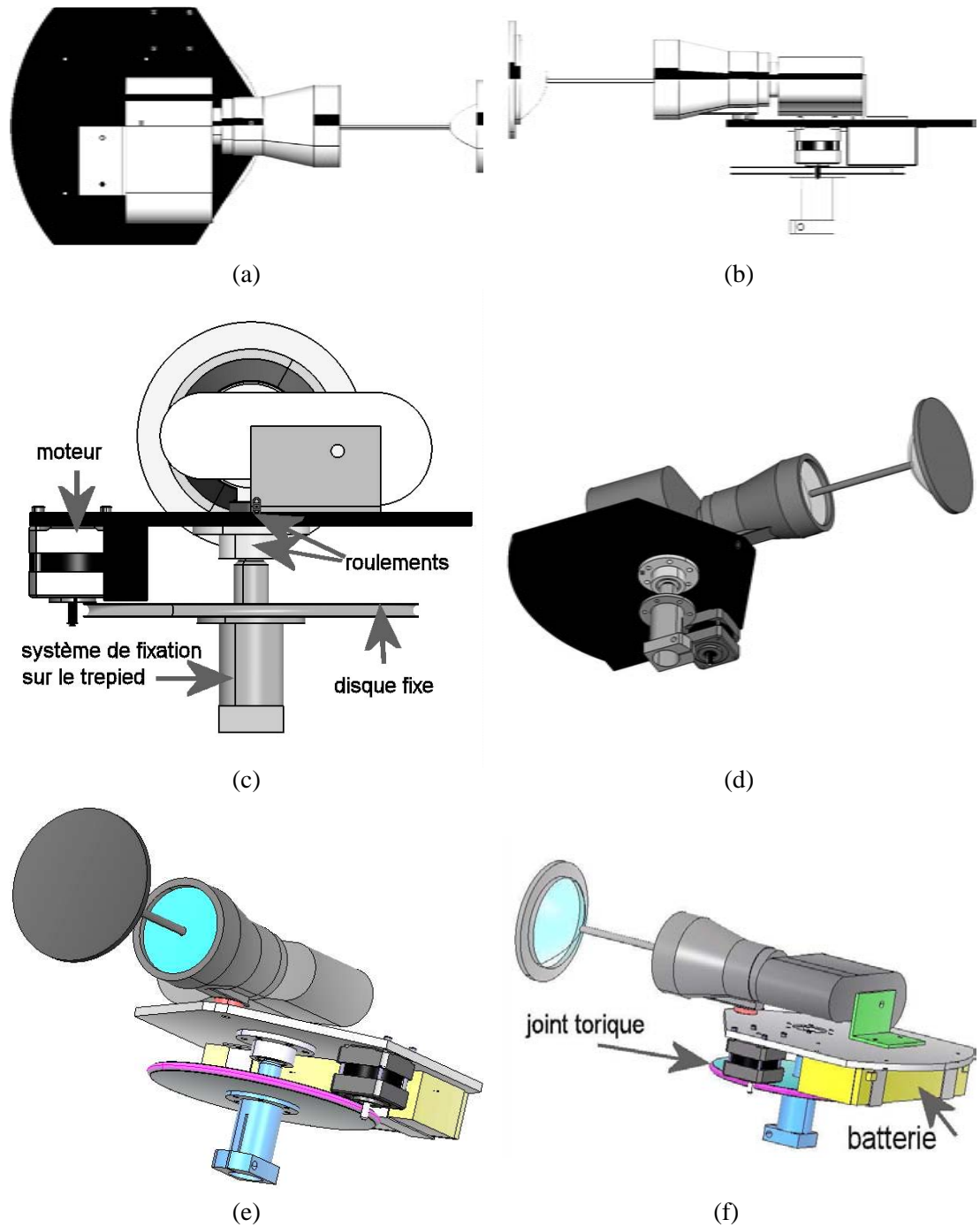


FIG. 5.2 – (a-c) Modèle CAO du capteur. (d) Modèle CAO vue de dessous sans le disque fixe. (e-f) Modèle CAO avec la batterie.

Le moteur pas à pas est piloté par des transistors de puissance fournissant le courant nécessaire à l'obtention d'un couple suffisant. Son accélération et sa décélération sont gérées de manière progressive par le microcontrôleur pour compenser l'inertie de la platine et assurer qu'aucun pas ne soit sauté.

Les différents réglages du capteur sont paramétrables depuis l'interface utilisateur et ils peuvent être mémorisés dans la mémoire EEPROM du microcontrôleur. Le tout est alimenté par une batterie 12V de 2Ah située sous la platine, l'appareil photo disposant de sa propre batterie.

Le capteur est également équipé d'une connexion USB permettant le contrôle depuis un ordinateur. Dans ce cas, il est possible d'envoyer les images de la caméra à l'ordinateur via la sortie vidéo de l'appareil photographique connectée à une carte d'acquisition sur l'ordinateur. Ce mode permet l'acquisition d'un grand nombre d'images en un temps réduit, comme proposé par Kang dans [78], mais au prix d'une baisse de la résolution et de la nécessité de transporter l'ordinateur.

5.1.5 La caméra et l'optique

La caméra utilisée est un appareil photo numérique Nikon CoolPix 4500 équipé d'un capteur DTC d'une résolution de 4 mégapixels effectifs, permettant l'enregistrement d'images de 2272*1704 pixels. Malheureusement, uniquement 1.6 million de pixels sont effectivement utilisés pour les deux raisons suivantes :

Premièrement, seule la portion de l'image comprise entre les deux cercles définis par les contours du miroir et du reflet de la caméra est utilisée.

Ensuite, afin de maintenir les paramètres intrinsèques de la caméra fixes entre les différentes acquisitions et éviter des fastidieux étalonnages, il ne faut pas que la mécanique de mise au point soit activée. Pour ceci, nous utilisons l'appareil dans une position parquée, correspondant à un zoom minimum. Dans cette configuration, le cadrage du miroir n'est pas optimal.

Le capteur d'image de la caméra est couleur. La caméra ne disposant pas de trois capteurs séparés, un pour chacun des canaux Rouge, Vert et Bleu, un unique capteur est utilisé. Il est recouvert d'un filtre, dit de Bayer, constitué de cellules colorées des couleurs primaires. Une image couleur de la même résolution que le capteur monochrome utilisé est reconstruite en utilisant un traitement propre à chaque appareil, et non inversible a priori. Ceci pose des problèmes, par exemple lors de la recherche subpixelique de la position d'un amer de la mire. En effet, l'image de luminance n'est pas réellement disponible et c'est une version dégradée par ce traitement qui est utilisée. Dans la suite, ce problème n'est pas traité, et l'image est considérée comme étant acquise avec une caméra monochrome.

Une carte mémoire Compact Flash de 128 Mo est utilisée pour stocker à peu près 100 photos au format JPEG avec un taux de compression garantissant une qualité d'image suffisante pour les traitements. L'idéal est d'utiliser les images brutes, non compressées et donc non dégradées, mais chacune pèse 12Mo, ce qui limite la quantité d'images enregistrables et surtout allonge le temps d'acquisition, l'enregistrement sur la carte mémoire d'une telle image nécessitant une vingtaine de secondes.

Le complément d'optique paracatadioptrique utilisé est le Oneshot360° de Remote Reality commercialisé par la société Panosmart.

5.1.6 Le logiciel

5.1.6.1 Le logiciel embarqué

Le logiciel embarqué dans le microcontrôleur permet le contrôle des différents éléments. Des paramètres tels que la vitesse du moteur, le nombre d'images à acquérir ou le nombre de pas du moteur par tour du capteur peuvent être réglés via l'interface. L'état de charge de la batterie est aussi contrôlé en permanence.

Une fois les paramètres réglés, l'opérateur règle manuellement l'inclinaison du capteur puis l'acquisition d'une série d'image commence automatiquement dans la bonne orientation après un délai de durée programmable, pour permettre à l'opérateur de sortir du champ de vision du capteur.

L'interface utilisateur permet aussi la sélection d'un mode d'acquisition des images dans lequel deux images sont successivement acquises pour chaque position du capteur, avec une alternance sur la commande du relais, pour permettre l'utilisation de LED ou de projecteur de lumière structurée, comme il sera montré plus loin.

5.1.6.2 Le logiciel non embarqué

Le capteur peut aussi fonctionner en étant contrôlé à distance via un ordinateur. Il utilise pour cela un protocole de communication propre véhiculé par liaison USB. Tout le fonctionnement du capteur est alors piloté par l'ordinateur, les images de la caméra étant alors récupérées en temps réel grâce à une liaison vidéo.

5.2 Étalonnage du capteur

Un modèle géométrique doit être défini et ses paramètres estimés afin de pouvoir utiliser les images du capteur pour effectuer la reconstruction 3D. Le modèle présenté ici est inspiré des travaux de Gonzalez-Barbosa et Lacroix [142], sauf qu'il ne prend pas en compte les erreurs d'alignement entre l'optique et le miroir, qui conduiraient à une caméra non centrale mais que les auteurs approximent par une caméra centrale. La méthode d'étalonnage proposée ici a aussi des points communs avec celle proposée par Mei et Rives dans [143] de par sa prise en compte de la non homogénéité de la résolution dans l'image pour éviter un biais dans l'estimation des paramètres. La gestion des distorsions radiales est étendue et la détermination des paramètres extrinsèques adaptée au mouvement particulier du capteur.

5.2.1 Un modèle géométrique du capteur

5.2.1.1 La caméra paracatadioptrique

Le modèle employé pour la caméra est le même qu'au [chapitre 3](#) avec en plus une gestion des pixels non carrés et une correction radiale.

Une rectification radiale est généralement employée pour les caméras standard afin d'éliminer les distorsions présentes dans les images et générant des effets en coussinet ou barillet. La fonction de correction est en général un polynôme de degré variable tel que, par exemple, r_d étant le rayon associé à une coordonnée pixellique dans l'image avec distorsion et r_p étant le rayon correspondant dans l'image sans distorsion :

$$r_d = (1 + k_1 r_p^2 + k_2 r_p^4) r_p \quad (5.1)$$

Li et Hartley proposent dans [163] un récapitulatif très intéressant sur les différents modèles de distorsion radiale. À titre indicatif, car elle n'est pas utilisée ici, ils présentent une nouvelle méthode permettant d'estimer les paramètres de leur modèle de distorsion (qui n'assume pas un centre des distorsions confondu avec le centre optique) de manière non itérative, à partir de l'observation de deux images d'une même scène. L'intérêt de leur méthode est qu'elle permet de découpler l'estimation de la matrice fondamentale et celle de la distorsion grâce à l'emploi du modèle de caméra radiale de Thirthala et Pollefeys [100, 101].

Pour notre caméra, une nouvelle fonction de correction radiale est proposée ici. Contrairement aux caméras standard, la zone centrale de l'image d'une caméra catadioptrique n'est pas utilisée à cause du reflet de la caméra. La fonction de rectification choisie ne doit donc pas nécessairement conserver un rayon nul, et peut donc posséder un offset, pour être adaptée au mieux aux données à rectifier.

Après analyse des erreurs radiales dans les données d'étalonnage, la fonction retenue est un polynôme de degré trois, telle que :

$$r_d = r_p + (l_0 + l_1 r_p + l_2 r_p^2 + l_3 r_p^3) \quad (5.2)$$

Un autre intérêt d'une telle fonction est que l'on peut calculer, à une itération, les coefficients du polynôme qui minimisent l'erreur radiale. Si les données d'origine étaient déjà rectifiées par une fonction calculée à l'itération précédente, la nouvelle fonction s'obtient simplement par sommation des coefficients des polynômes.

Comme cela est souvent fait pour les caméras standard, le centre des distorsions est assimilé à la projection orthogonale du centre optique sur l'image.

La caméra comporte donc les huit paramètres intrinsèques suivants :

- Les trois paramètres vus au [chapitre 3](#) : u_0 , v_0 et h .
- Un paramètre d'échelle entre la largeur et la hauteur des pixels.
- Les quatre paramètres de correction radiale.

5.2.1.2 Un mouvement contraint

La caméra paracatadioptrique définit un repère caméra $\omega_c = (x_c, y_c, z_c)$, centré en son centre optique et avec les axes x_c et y_c alignés avec les lignes et colonnes du capteur d'image comme vu au [chapitre 3](#).

Le mouvement de ce repère est contraint entre les différentes images et il n'est donc pas nécessaire de modéliser et donc d'estimer les différentes poses indépendamment par rapport au repère de travail $\omega_M = (x_M, y_M, z_M)$. Au lieu de cela, les paramètres extrinsèques utilisés sont tels que montrés sur la figure 5.3. (x_a, y_a) correspond à la position dans le plan $z_M = 0$ de l'axe de rotation. α_1 et β_1 sont les deux angles de rotation d'Euler permettant d'aligner z_M avec l'axe de rotation. θ_0 correspond à l'angle de rotation autour de l'axe pour la première image de la séquence et θ_i est l'incrément angulaire entre chaque image, numérotée par n . h_m est une translation le long de l'axe de rotation. α_2 et β_2 sont les deux angles de rotations d'Euler permettant d'aligner le repère avec celui de la caméra. Une dernière translation de d_m sur l'axe z_x et de x_b sur l'axe x_c permet d'arriver au repère de la caméra.

Le très faible pas pour la rotation de la platine (à peu près 0.03 degré par pas du moteur) permet une grande répétabilité dans les commandes de rotation. Par contre, la valeur exacte d'un incrément angulaire θ_i n'est pas précisément connue par construction, car il correspond à un très grand nombre de pas. C'est pourquoi elle est estimée ici.

Il y a donc au total onze paramètres extrinsèques quel que soit le nombre d'images utilisées :

- x_a, y_a : translation dans le plan $z_M = 0$.
- α_1, β_1 : rotation pour aligner le repère avec l'axe de rotation.
- θ_0, θ_i, h_m : rotation autour de l'axe et translation.
- α_2, β_2 : rotation pour aligner le repère avec celui de la caméra.
- d_m, x_b : translation pour amener le repère au niveau du centre optique de la caméra.

5.2.2 La mire 3D

Afin d'estimer les paramètres du modèle, un ensemble de points est observé et un critère est minimisé. Ce critère utilise une métrique appelée erreur de reprojection qui consiste en la mesure de la distance dans l'image entre la position réelle d'une observation et sa position prédite par le modèle.

L'utilisation exclusive d'images de droites pour l'étalonnage, telle que présentée au [chapitre 3](#) n'est pas adaptée à la détermination des paramètres extrinsèques ni à celle des distorsions radiales. Des points sont donc utilisés ici. Ils permettent l'utilisation d'une métrique simple et significative pour déterminer l'adéquation du modèle et de ses paramètres avec le capteur réel. De plus, il est facile d'obtenir un grand nombre de points et leurs projections dans l'image peuvent être détectées aisément et précisément.

La mire est telle qu'elle se projette sur l'intégralité de la surface du capteur d'image lorsque la caméra est mise en rotation, ce qui permet d'éviter que l'estimation des paramètres du modèle soit effectuée au détriment d'une zone du capteur. La figure 5.6 (a) montre la répartition des points d'étalonnage dans les différentes images.

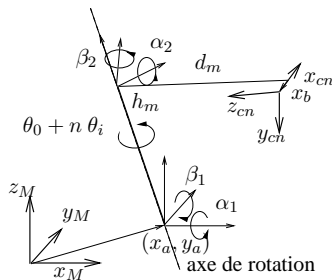


FIG. 5.3 – Les différents paramètres extrinsèques entre le repère de travail ω_M et le repère de la caméra ω_C .



FIG. 5.4 – Exemple d'image de la mire, cinquième de la séquence acquise qui en comporte vingt.

La figure 5.4 montre une image sur laquelle la mire est visible. Elle est constituée de trois plans disposés en U, pour une hauteur d'1.2 mètre et comporte 360 points utilisables. Ces points sont des croix définies par les coins des cases d'un échiquier, utilisé fréquemment pour l'étalonnage des caméras standard et par Mei et al. [143] pour les caméras catadioptriques. La mire est supposée parfaitement construite et les erreurs d'alignement entre les trois plans sont considérées nulles et ne sont donc pas intégrées au modèle pour l'instant.

Des tests ont été effectués pour estimer l'erreur d'alignement des trois plans constituant la mire à partir des images. Les résultats obtenus sont très variables alors que la mire est physiquement inchangée. Nous avons donc préféré considéré la mire comme parfaitement construite plutôt que d'ajouter cette estimation. De plus, cette estimation représente un surcoût calculatoire assez important, car il faut alors estimer 12 paramètres supplémentaires.

5.2.3 Estimation des paramètres du modèle

Les paramètres du modèle sont estimés en plusieurs phases, partant d'une solution initiale grossière qui est ensuite raffinée.

5.2.3.1 Un premier recalage du modèle de mire

Dans un premier temps, afin de permettre un recalage approximatif du modèle de la mire dans les images, un certain nombre de points doivent être détectés dans les images. Ensuite l'appariement d'un maximum de points du modèle de mire avec leurs projections correspondantes dans les images est effectué pour ensuite réaliser un affinage sur un jeu de données plus important. La détection des points dans les images peut être faite de différentes façons :

Manuellement : Quatorze points particuliers de la mire sont localisés manuellement dans chaque image quand ils sont visibles. Ces points sont localisés comme suit : quatre points aux coins de chacun des deux damiers horizontaux et six points disposés verticalement au centre du damier vertical. Les positions des points sont alors raffinées pour obtenir une précision subpixelique en utilisant l'un des outils de la Calibration Toolbox de Bouget [125], réalisant un ajustement de paraboloïde hyperbolique sur le canal de luminance de l'image pour estimer au mieux la position des intersections de cases du damier.

Automatiquement : La localisation de ces points peut aussi être effectuée de manière totalement automatique, en employant des méthodes d'estimation robustes. Néanmoins, pour simplifier la localisation des amers, ce sont des LED plutôt que les positions des intersections des damiers qui sont utilisées, car ces dernières sont en trop grand nombre. Pour détecter les projections des LED dans les images, ce sont deux images qui sont acquises pour chaque position du capteur. Sur l'une des deux images, la carte électronique commande l'allumage d'un jeu de LED. Deux images sont acquises, comme celles montrées sur la figure 5.5 (a) et (b) (sur laquelle n'est représenté qu'un unique damier). Ces images permettent, par simple différenciation, d'obtenir l'image (c). Elle est segmentée et les centres de gravité de groupes de pixels connexes sont utilisés comme position de points d'étalonnage. On pourra noter l'emploi de LED de couleurs différentes pour assister l'appariement automatique des points de l'image avec le modèle 3D de la mire. Ces points ne peuvent par contre pas être utilisés lors de l'étape d'estimation fine car leurs positions dans la mire ne sont pas parfaitement maîtrisées. De plus, les centres de gravité de leurs projections dans l'image ne correspondent pas physiquement aux projections des centres des LED.

Une première estimation d'un sous ensemble de paramètres est effectuée en utilisant l'algorithme de Levenberg-Marquardt pour minimiser la somme des carrés des erreurs de projection.

Le modèle 3D de la mire est ensuite reprojété dans chacune des images en utilisant le jeu de paramètres calculés. Les coins sont détectés dans les images par le détecteur de Harris et appariés avec des points du modèle de mire s'ils sont situés à une distance suffisamment faible de leurs reprojections et qu'aucun autre coin n'a été détecté dans un rayon donné. Les positions des nouvelles intersections de damier dans les images sont raffinées comme précédemment.

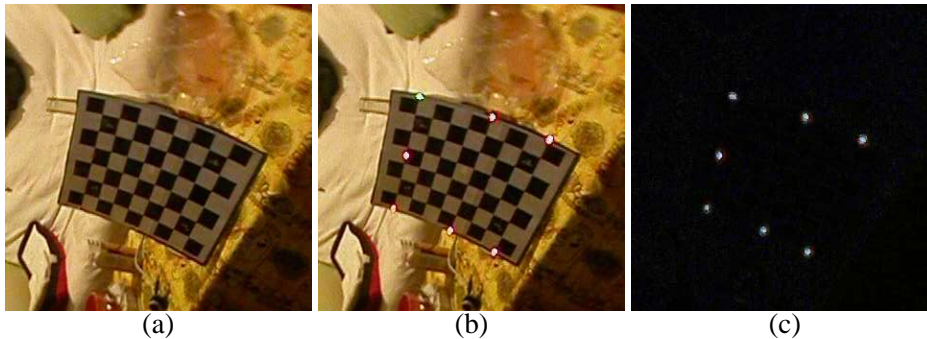


FIG. 5.5 – (a) Image de la mire acquise avec les LED éteintes. (b) Image de la mire acquise avec les LED allumées. (c) Différence des deux images.

5.2.3.2 Affinement des paramètres p_1

Ensuite, les paramètres du modèle sont estimés sur l'ensemble des points appariés en alternant des phases d'estimations de deux jeux de paramètres différents, comme proposé par Zhang dans [106]. Tous les paramètres hormis ceux de rectification radiale sont estimés (ces paramètres sont notés p_1) puis la rectification radiale à appliquer est estimée à son tour (ces paramètres sont notés p_2). Le processus est itéré jusqu'à convergence. Cette décomposition en deux jeux de paramètres a fourni de meilleurs résultats en termes d'erreur de reprojection finale et de vitesse de convergence qu'une estimation simultanée de tous les paramètres.

Cette étape d'affinement des paramètres doit prendre en compte la résolution fortement variable de la caméra, la non homogénéité de la répartition des points d'étalonnage ainsi que la gestion des outliers, dûs à de mauvais appariements. Ceci est effectué par des fonctions de pondération et de saturation décrite ci-dessous.

Une résolution variable : Il a été montré au [chapitre 1](#) que la caméra paracatadioptrique présente une forte variation de la résolution. Comme le proposent Mei et Rives dans [143], il est préférable de minimiser une erreur angulaire entre les rayons de projection qu'une erreur pixellique dans l'image. Cette erreur angulaire doit être la plus constante possible sur toute la surface de l'image. Pour simplifier les calculs, la métrique utilisée est une distance pixelique dans l'image. Dans la suite, la sphère représentant le champ de vision de la caméra est partitionné en coordonnées sphériques en portions k de Δ_ϕ et Δ_θ radians respectivement en élévation et azimuth, l'axe z de la sphère étant confondu avec l'axe z_c . Une approximation est faite par l'emploi d'un facteur multiplicatif $f_1(k)$ entre l'erreur pixelique et l'erreur angulaire, calculé pour chacune des portions k de la sphère de champ de vision. Ce facteur est calculé pour un rayon situé au centre de la portion. Ensuite, l'erreur angulaire correspondant à un point p est approximée en déterminant la portion de sphère k correspondante et en multipliant l'erreur pixelique e_p correspondante par le facteur $f_1(k)$.

Une saturation adaptée pour les outliers : L'influence d'un résidu correspondant à un outlier (un mauvais appariement entre un point du modèle de mire et un coin dans l'image) est

limitée par une fonction de saturation. A cause de la résolution variable, il est impossible de fixer un seuil constant de distance dans l'image pour qu'un point soit considéré comme inlier. En effet, un tel seuil mènerait à considérer à tort des points situés à la périphérie de l'image comme des outliers et des points situés près du centre comme des inliers. Il est par contre possible d'utiliser le facteur f_1 pour avoir une estimation de l'erreur angulaire, qui elle est une mesure constante dans l'image et peut donc être saturée correctement par une fonction $sat(a)$ valant a si $|a| < seuil$ et $seuil$ sinon.

Une répartition non homogène : La figure 5.6 montre en (a) la répartition des points d'étalonnage dans l'image et en (b), la même répartition, mais représentée de manière radiale par rapport à la projection du centre optique dans l'image. Cette répartition est clairement non homogène et il est donc nécessaire de pondérer les résidus pour éviter que certaines zones de l'image ne soient favorisées lors de l'estimation au détriment d'autres. Ceci est effectué grâce à un facteur $f_2(p) = a(p)/n(p)$ associé à chaque portions de la sphère de vue, $n(p)$ étant le nombre de points présents dans la portion considérée et $a(p)$ l'aire de la portion sur la sphère. Chaque portion a une forme correspondant à une partie de zone. Une zone est une portion de sphère comprise entre deux plans parallèles et sa surface est égale à $2\pi rh$, r étant le rayon de la sphère, unitaire ici, et h l'écartement entre les deux plans. La surface correspondant à $a(p)$ est donc égale à $\Delta\theta/2\pi$ de la surface de cette zone dont la hauteur dépend de l'élévation ϕ de la portion considérée :

$$a(p) = \Delta\theta \left(\cos\left(\phi + \frac{\Delta\phi}{2}\right) - \cos\left(\phi - \frac{\Delta\phi}{2}\right) \right) \quad (5.3)$$

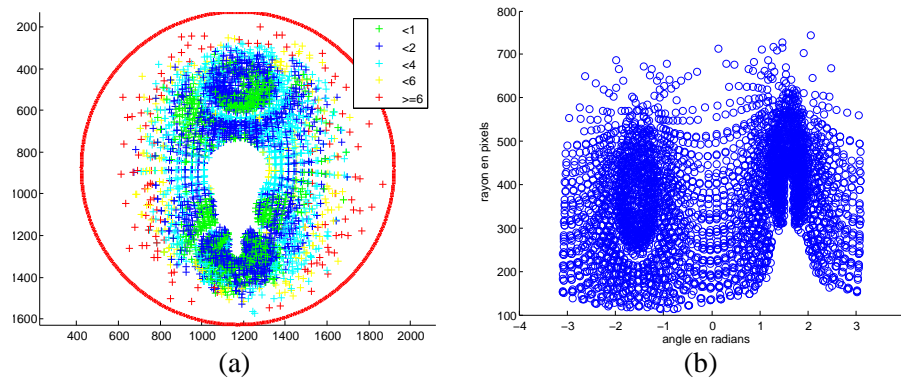


FIG. 5.6 – (a) Répartition des observations et de l'erreur de reprojection dans l'image en pixels. (b) Répartition radiale des observations dans l'image.

L'histogramme visible sur la figure 5.7 (a) montre la répartition radiale, correspondant à différentes élévations sur la sphère de vue, des points d'étalonnages dans l'image. La figure

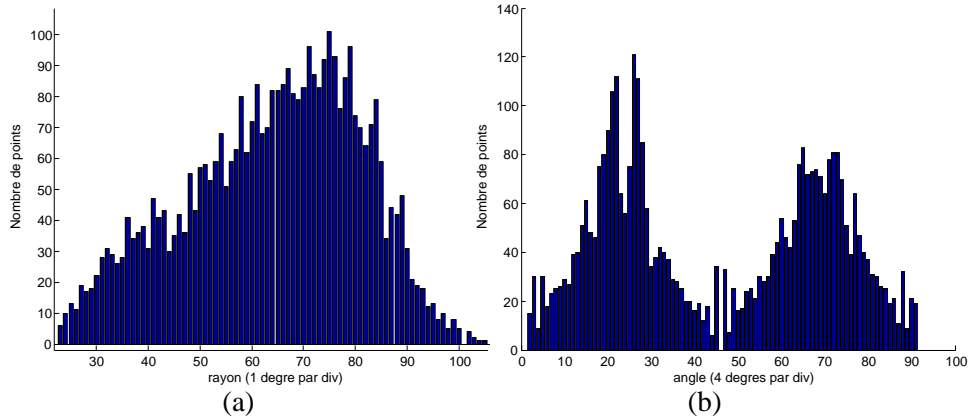


FIG. 5.7 – Histogrammes des positions des points d’étalonnage sur la sphère de vue (a) en élévation (radialement), (b) en azimut (tangentiellement).

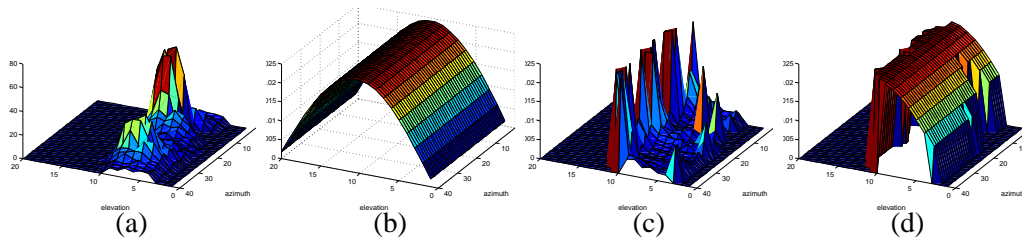


FIG. 5.8 – (a) Histogramme 2D montrant la répartition des observations sur la sphère de vue. (b) Surface des différentes zones correspondant aux portions. (c) La valeur de f_2 pour chaque portion. (d) L’histogramme (a) multiplié par (c).

(b) montre la répartition sur les différents secteurs angulaires dans l’image, correspondant à différents azimuts sur la sphère de vue. La figure 5.8 (a) montre l’histogramme de répartition sur la sphère de vue échantillonnée à la fois en azimut et en élévation. La figure (b) montre l’aire correspondant aux différentes zones. La figure (c) montre la fonction de pondération $f_2(p)$ calculée et la figure (d) montre le produit de l’histogramme (a) par (c), qui doit se rapprocher au plus de celui visible en (b).

Intégration des différentes pondérations et saturations : Finalement les résidus res_p sont pondérés en combinant les différents résultats obtenus et l’estimation des paramètres est effectuée en minimisant $\sum res_p^2$.

$$res_p^2 = f_2(k)^2 sat(f_1(k)^2 e_p^2); p \in portion(k) \quad (5.4)$$

Pour éviter des problèmes de stabilité lors de l’estimation de paramètres et accélérer les calculs, les facteurs $f_1(p)$ et $f_2(p)$ ne sont pas mis à jour à chaque itération, bien que des points puissent changer de portions.

5.2.3.3 Affinement des paramètres p_2

L'affinement des paramètres p_2 fait intervenir un ajustement de polynôme sur un ensemble de points 2D. Le raisonnement concernant la pondération et la saturation des résidus déjà décrit lors de l'estimation des paramètres p_1 est toujours valable. Il est néanmoins simplifié car l'analyse est ici purement radiale, et les histogrammes à utiliser sont donc monodimensionnels tel celui présenté sur la figure 5.7.

5.2.4 Résultats

5.2.4.1 Première estimation

Sur la séquence de vingt images dont celle de la figure 5.4 est extraite, 259 points ont été définis pour le premier recalage. Pour ces points, l'erreur de reprojection moyenne et l'écart type avant (resp. après) minimisation est de 34.40 ± 19.55 (resp. 4.17 ± 1.98) pixels, pour une image correspondant à un cercle de rayon 1600 pixels.

Ce premier recalage a permis l'appariement de 3847 points, dont les positions dans l'image sont situées à moins de 10 pixels de la position obtenue par reprojection du modèle de mire.

5.2.4.2 Seconde estimation

La seconde partie de l'estimation des paramètres, permet d'obtenir une erreur de reprojection de 1.93 ± 1.83 pixels après quinze couples d'estimations alternées des paramètres p_1 et p_2 . La répartition de l'erreur de reprojection pixellique à la fin de l'optimisation est visible sur la figure 5.6 (a), l'amplitude de l'erreur de reprojection en pixel étant représentée par différentes couleurs.

Cette erreur de reprojection est globalement plus grande à la périphérie, ce qui était souhaité puisque la résolution est la plus grande dans cette zone. Ceci provient aussi du fait que les positions des points d'étalonnage sont moins bien détectées à cause du flou dans cette zone de l'image.

La figure 5.9 montre la répartition de l'erreur tangentielle (en (a)) et radiale (en (b), marquée "avant correction") après la première estimation de paramètre p_1 en fonction du rayon en pixel. L'erreur tangentielle vaut 0.092 ± 2.031 pixels, ce qui justifie l'absence de correction tangentielle, la moyenne étant presque nulle.

En (b) figure le tracé du polynôme estimé pour corriger radialement les coordonnées des points. Le choix du degré trois pour le polynôme a été effectué de manière empirique pour s'adapter au mieux aux données tout en étant le plus petit possible.

Le polynôme obtenu après une première estimation des paramètres p_2 a les coefficients suivants : $l_0 = 9.03$, $l_1 = -2.19e^{-2}$, $l_2 = 1.17e^{-4}$, $l_3 = -2.13e^{-7}$.

Il permet de corriger l'erreur radiale qui vaut initialement 4.21 ± 5.11 pour l'amener à une valeur de -0.28 ± 1.70 comme visible sur la figure (b). Il faut noter que ces valeurs ne tiennent pas compte de la pondération des résidus. En effet l'optimisation permet d'obtenir une erreur radiale moyenne nulle sur les données pondérées grâce au paramètre d'offset l_0 .

On retrouve clairement une erreur de reprojection plus grande sur la périphérie de l'image déjà évoquée, et visible dans des proportions semblables sur les tracés (a) et (b) après correction.

Finalement, après quinze itérations d'estimations successives des paramètres p_1 et p_2 , les coefficients du polynôme obtenu sont les suivants : $l_0 = 11.05$, $l_1 = -3.17e^{-2}$, $l_2 = 1.37e^{-4}$, $l_3 = -2.39e^{-7}$.

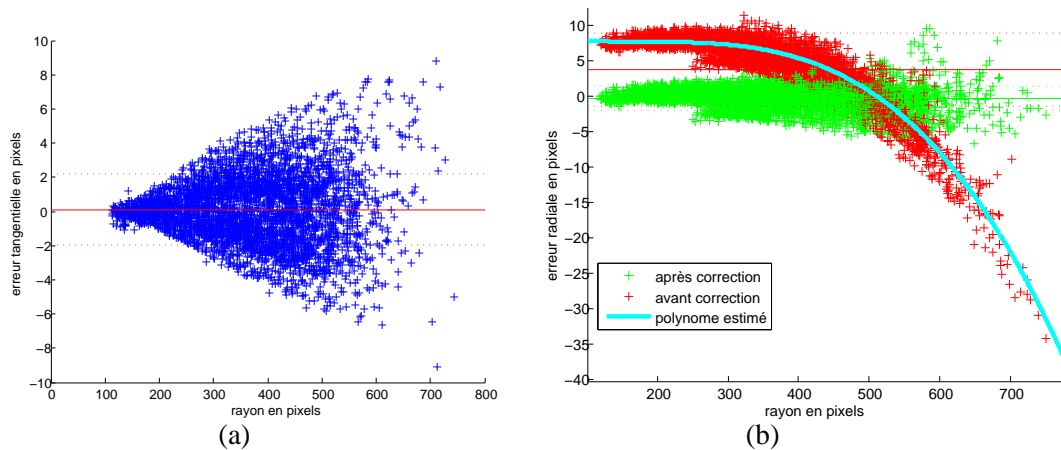


FIG. 5.9 – (a) Erreur tangentielle en fonction du rayon en pixel. (b) Erreur radiale en fonction du rayon en pixel, avant et après correction lors de la première itération.

5.3 Reconstruction 3D

La reconstruction 3D est possible grâce à la mise en correspondance d'observations de même objets de la scène dans des images acquises depuis différentes positions.

5.3.1 Géométrie épipolaire

La géométrie épipolaire est utilisée en stéréoscopie pour contraindre l'espace de recherche des pixels homologues, qui sont les projections d'un même point de la scène dans les différentes images.

Le document de Zhang [164], même s'il est assez ancien, fournit une description intéressante de la géométrie épipolaire pour les caméras standard et de l'incertitude liée à son estimation à partir des images.

Un point P observé par une caméra centrale se trouve sur un rayon de projection reliant la position pixellique à laquelle il est observé et le centre optique de la caméra O_1 . Afin de déterminer la profondeur du point sur ce rayon, une autre image du même point est observée depuis une autre position de la caméra O_2 . Le lieu possible des images du point sur la seconde

image est situé sur l'image du rayon de projection, elle-même située à l'intersection du second plan image et du plan défini par O_1 , O_2 et P .

Dans le cas des caméras standard, une fois l'image rectifiée pour retirer les distorsions optiques, une droite de la scène est projetée en une droite dans l'image. La recherche de l'homologue d'un point observé dans l'image 1 s'effectue donc le long d'une telle droite, dite *épipolaire*, dans l'image 2. Ces droites s'intersectent en un point, appelé épipole, qui est l'image de O_1 . Le même raisonnement est bien sûr valable en inversant les images 1 et 2.

La géométrie épipolaire a été largement étudiée pour les caméras standard, par exemple par Zhang dans [164] ou Hartley et Zisserman dans [77]. La thèse de Svoboda [165] l'étend pour les caméras catadioptriques centrales. Micusik et Padjla proposent un modèle pour la géométrie épipolaire, permettant l'étalonnage de caméras catadioptriques légèrement non centrales dans [52]. Plus particulièrement, pour la caméra paracatadioptrique, la géométrie épipolaire a été étudiée par Geyer et Daniilidis dans [109]. Il faut aussi noter l'existence de travaux de Sturm [111] permettant l'utilisation de la géométrie épipolaire avec des caméras de types différents, paracatadioptriques, perspectives ou affines. Les contraintes linéaires sur trois ou quatre images peuvent aussi être établies grâce à l'emploi de tenseurs trifocaux ou quadrifocaux, qui sont des extensions de la géométrie épipolaire. Ceci a également été étudié pour différentes configurations de caméras par Torii et al. dans [166].

5.3.1.1 Géométrie épipolaire dans les images d'origine sans distorsions

Considérons dans un premier temps deux images obtenues par une caméra paracatadioptrique parfaite en mouvement. Ces images sont obtenues soit par une caméra idéale, telle que présentée au [chapitre 3](#), soit après rectification d'images d'une caméra non idéale.

La rectification d'une image est effectuée en échantillonnant une image rectifiée. Pour chacun de ses pixels p_p , le rayon r_p est calculé. r_d est ensuite déterminé en utilisant l'équation (5.2), fournissant les coordonnées correspondantes de p_d dans l'image de la caméra après application du facteur d'échelle entre la hauteur et la largeur des pixels. La valeur de luminance ou de couleur en cette position est déterminée, soit par celle du plus proche voisin, soit par interpolation, bilinéaire par exemple. Cette valeur est ensuite utilisée dans l'image rectifiée pour le pixel p_p .

Comme il a été montré au [chapitre 3](#), l'image d'une droite et par conséquent d'un rayon de projection est un cercle dans le plan image d'une caméra paracatadioptrique parfaite. La détermination du cercle épipolaire dans l'image 2 correspondant à un point dans l'image 1 est réalisée en calculant l'intersection avec le paraboloïde (correspondant à la caméra 2) du plan contenant le rayon de projection de la caméra 1 et O_2 .

Il est donc très facile de déterminer les cercles épipolaires lorsque les paramètres extrinsèques sont connus.

Comme pour les caméras standard, la géométrie épipolaire peut se formuler sous forme matricielle en tant qu'une contrainte bilinéaire entre les coordonnées des points observés dans les images, élevés sur les paraboloïde correspondants aux caméras. Ceci n'est pas détaillé ici, mais Geyer et Daniilidis proposent dans [109] une étude correspondant au cas où les paramètres

sont inconnus, avec l'emploi d'une matrice *fondamentale*.

5.3.1.2 Géométrie épipolaire dans les images d'origine avec distorsions

Il est souvent préférable de travailler dans les images d'origine fournies par la caméra, pour éviter le rééchantillonnage introduit par la rectification. Ceci peut, par exemple, permettre une localisation plus précise des points homologues.

La géométrie épipolaire dans ces images est plus complexe à appréhender :

Premièrement, connaissant une position pixelique dans l'image 1, il faut calculer les coordonnées correspondantes dans le plan image parfait pour déterminer le rayon de projection. Il n'est pas toujours possible d'obtenir ces coordonnées à partir des coordonnées dans le plan image réel. En effet, le modèle employé pour la correction radiale fournit une solution unique pour la transformation directe (de r_p vers r_d). Pour la transformation inverse, il est nécessaire de calculer les racines d'un polynôme dont l'unicité d'une racine réelle positive n'est pas garantie. La valeur de r_p peut néanmoins être obtenue en partant d'une solution initiale r_{p0} et en minimisant :

$$\hat{r}_p = \arg \min_{r_p} (r_d - (l_0 + l_1 r_p + l_2 r_p^2 + l_3 r_p^3)) \quad (5.5)$$

La valeur de r_{p0} est par exemple obtenue en calculant les valeurs de r_{dj} pour tous les pixels j d'un rayon dans l'image et en choisissant la valeur de r_{pj} qui a la valeur correspondante r_{dj} la plus proche du r_d recherché.

Le seconde raison qui rend compliquée l'utilisation des images d'origine pour la géométrie épipolaire est que le cercle épipolaire calculé dans le plan image parfait à partir du rayon de projection doit être projeté dans le plan image réel. Ceci conduit à une courbe difficilement utilisable en pratique.

Au lieu de cette approche, il est plus simple d'abstraire la géométrie épipolaire et de projeter dans l'image 2 des points échantillonnés le long du rayon de projection de l'image 1 et d'interpoler entre les points. Ceci a déjà été présenté au [chapitre 4](#) avec un modèle de caméra non centrale.

La figure 5.10 illustre ce principe. Considérons le rayon de projection défini par le point fléché dans l'image (a). Des points le long du rayon de projection correspondant sont échantillonnés avec un pas de 5cm pour des distances de 10cm à 5m. Les projections de ces points sont visibles sur les images (b) et (c) correspondant à des orientations relatives du capteur proches de 18° et 180°.

Anecdotiquement, on pourra noter l'existence de travaux sur la géométrie épipolaire entre deux images de caméras standard non rectifiées avec des distorsions radiales simples de type $r_p = (1 + k_1 r_d^2) r_d$, proposés par exemple Zhang dans [167].

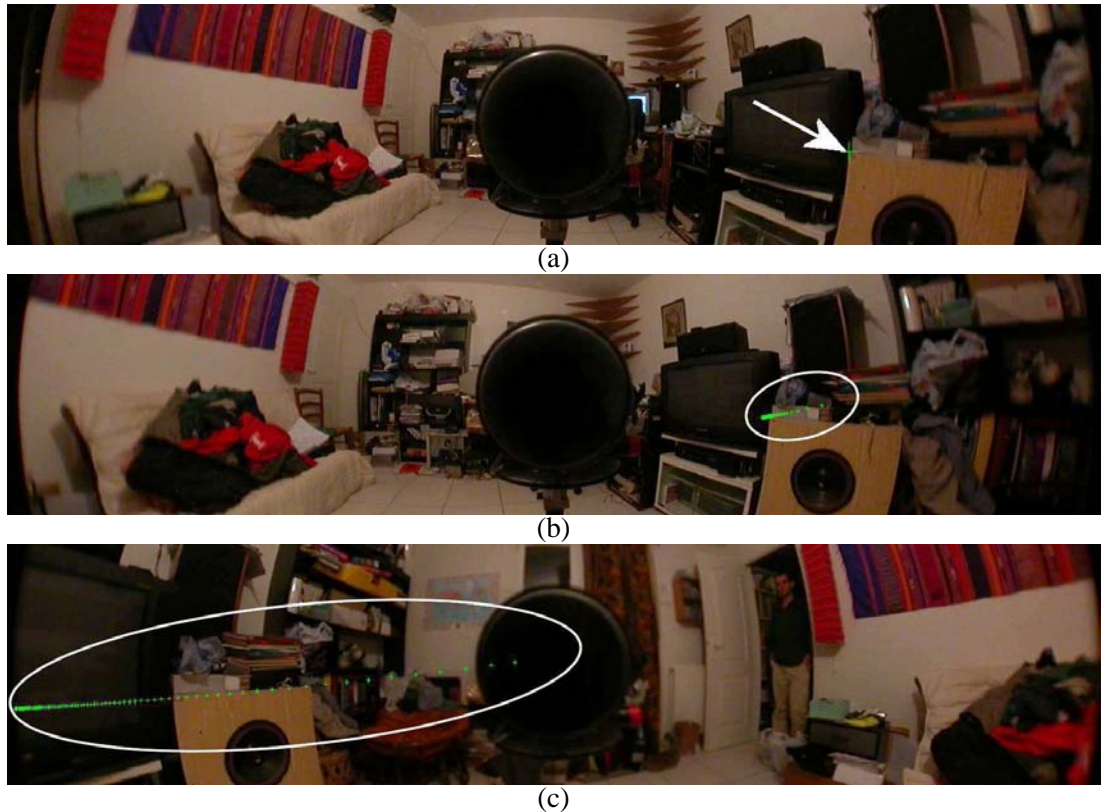


FIG. 5.10 – (a) Une image acquise par la caméra et un rayon de projection marqué par une croix. (b) Une image acquise après rotation de 18° de la platine par rapport à (a) et échantillonnage de points sur le rayon de projection. (c) Une image acquise après rotation de 180° de la platine par rapport à (a) et échantillonnage de points sur le rayon de projection.

5.3.2 Rectification épipolaire

La recherche des homologues dans les couples d'images nécessite la recherche le long des courbes (ou droites selon la caméra) épipolaires. Afin d'accélérer le traitement et permettre la création de cartes de profondeur dites denses, pour lesquelles chaque pixel possède une profondeur, il est nécessaire de rectifier le couple d'images afin que les courbes épipolaires soient rectifiées en des droites parallèles.

La figure 5.11 (a) illustre ce principe. Les repères $\omega_{c1} = (x_{c1}, y_{c1}, z_{c1})$ et $\omega_{c2} = (x_{c2}, y_{c2}, z_{c2})$ sont ceux de la caméra dans les deux prises de vue. Le segment L relie les positions des deux centres optiques et sa longueur représente l'écart intra-oculaire. Soit $\omega_{r1} = (x_{r1}, y_{r1}, z_{r1})$ le repère obtenu par rotation de ω_{c1} par la rotation R_{cr1} telle que l'axe y_{r1} soit aligné avec L . Soit $\omega_{r2} = (x_{r2}, y_{r2}, z_{r2})$ le repère obtenu par rotation de ω_{c2} par la rotation R_{cr2} telle que l'axe y_{r2} soit aussi aligné avec L et que l'axe z_{r2} soit aligné avec z_{r1} .

La rectification épipolaire est effectuée en reprojectant les images des deux caméras sur des plans Π_1 et Π_2 confondus, et en choisissant des repères de lignes et colonnes (u_1, v_1) et

(u_2, v_2) tels que u_1 et u_2 soient alignés.

Un point P définit un plan épipolaire contenant P et les centres optiques O_1 et O_2 des deux caméras. Ce plan intersecte les plans Π_1 et Π_2 en des droites confondues l_1 et l_2 . Grâce à l'échantillonnage adéquat des plans images Π_1 et Π_2 , ces droites correspondent à la même ligne v dans les images. Par conséquent, la recherche des homologues des pixels situés sur l_1 dans l'image 1 doit être effectuée sur l_2 dans l'image 2.

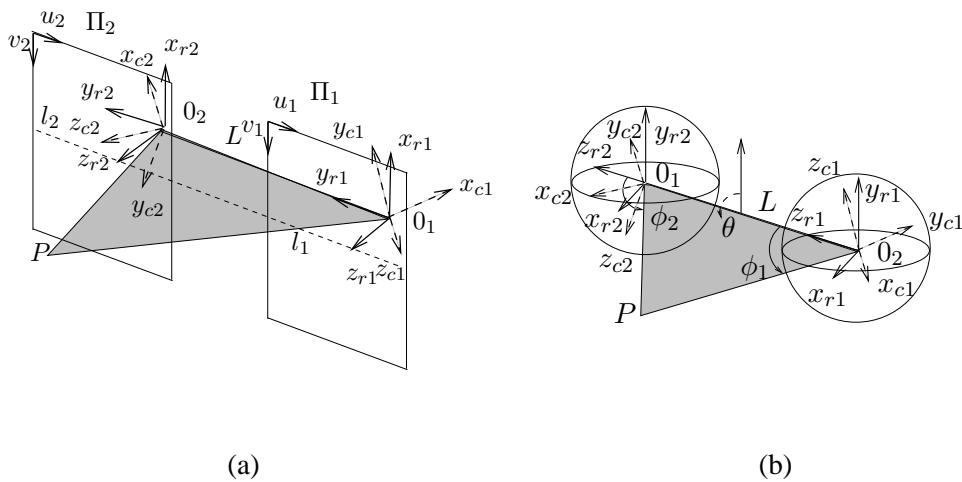


FIG. 5.11 – Rectification épipolaire : (a) pour des caméras standard, en utilisant un plan (b) pour des caméras à large champ de vision, en utilisant deux sphères.

5.3.2.1 Géométrie épipolaire dans des couples sphériques rectifiés

Comme proposé par Geyer et Daniilidis dans [168] et par Gonzalez-Barbosa et Lacroix dans [142], il est possible d'effectuer la rectification épipolaire d'un couple d'images obtenu par une caméra paracatadioptrique. Cette représentation permet d'avoir les points homologues sur les mêmes lignes des images rectifiées, comme lors de la rectification épipolaire pour les caméras standard. Pour représenter le large champ de vision de la caméra paracatadioptrique, le plan sur lequel les images des caméras standard est reprojété n'est pas suffisant, et il faut donc utiliser une surface couvrant une zone plus large. La surface la plus évidente pour cette tâche est la sphère.

La solution proposée par Geyer et Daniilidis pour obtenir les couples rectifiés est relativement complexe à mettre en œuvre. Elle est basée sur une transformation utilisant le calcul de cotangente hyperbolique, assez difficile à appréhender et ne permettant pas un paramétrage aisé de la résolution angulaire des images obtenues.

Gonzalez-Barbosa et Lacroix proposent une solution plus simple basée sur l'échantillonnage des plans épipolaires, qui contiennent les deux centres optiques. Une solution similaire est

proposée ici. Une formulation très légèrement différente permet l'extension à d'autres surface que la sphère, par exemple le cube, comme nous le verrons plus loin.

La figure 5.11 (b) montre la géométrie associée à la rectification épipolaire sur la sphère. Les repères $\omega_{c1} = (x_{c1}, y_{c1}, z_{c1})$ et $\omega_{c2} = (x_{c2}, y_{c2}, z_{c2})$ sont ceux de la caméra dans les deux prises de vue. Soit $\omega_{r1} = (x_{r1}, y_{r1}, z_{r1})$ le repère obtenu par rotation de ω_{c1} par la rotation R_{cr1} telle que l'axe z_{r1} soit aligné avec L . Soit $\omega_{r2} = (x_{r2}, y_{r2}, z_{r2})$ le repère obtenu par rotation de ω_{c2} par la rotation R_{cr2} telle que l'axe z_{r2} soit aussi aligné avec L et que l'axe y_{r2} soit aligné avec y_{r1} .

Le plan épipolaire défini par un point P , O_1 et O_2 est paramétré uniquement par un angle θ par rapport à l'axe y_{r1} . Les coordonnées sphériques de P dans le repère ω_{r1} sont (θ, ϕ_1) . Dans le repère ω_{r2} elles sont (θ, ϕ_2) . Le point P a donc la même coordonnée θ dans les deux repères.

Il apparaît clairement que la rectification épipolaire peut être effectuée en échantillonnant les valeurs de θ , de ϕ_1 et de ϕ_2 . Il suffit donc simplement de reprojeter les images de la caméra en projection sphérique, dans des repères judicieusement choisis. Dans l'algorithme suivant, qui décrit la rectification, $image_{r1}$ et $image_{r2}$ sont les images rectifiées en projection sphérique avec des pas d'échantillonnage de Δ_θ et Δ_ϕ . $image_1$ et $image_2$ sont les images d'origine acquise par la ou les caméras. $proj_1(P)$ et $proj_2(P)$ sont les fonctions de projection de chacune des deux caméras. Elles associent une position pixellique dans l'image à une position tridimensionnelle dans les repères caméra.

```

Algorithme de rectification épipolaire sphérique :
Calculer les matrices de rotation  $R_{cr1}$  et  $R_{cr2}$ 
%Pour tous les pixels de l'image rectifiée
pour  $u=0$  à  $u_{max}$  par pas de 1
  pour  $v=0$  à  $v_{max}$  par pas de 1
    %Calcul des coordonnées sphériques associées au pixel
     $\theta = u \Delta_\theta$ 
     $\phi = v \Delta_\phi$ 
    %Calcul des coordonnées cartésiennes dans les repères  $\omega_{r1}$  et  $\omega_{r2}$ 
     $x = \cos(\theta)\sin(\phi)$ 
     $y = \sin(\theta)\sin(\phi)$ 
     $z = \cos(\phi)$ 
    %Pour chacune des deux images du couple
    pour  $i = 1$  à 2
      %Calcul des coordonnées cartésiennes dans le repère  $\omega_{ci}$ 
       $P_i = R_{cri}^T [x, y, z]^T$ 
      %Calcul de la projection dans les images d'origine
       $(u_i, v_i) = proj_i(P_i)$ 
      %Recopie de la valeur du pixel
       $image_{ri}(u, v) = image_i(u_i, v_i)$ 

```

La recopie de la valeur du pixel de l'image d'origine vers l'image rectifiée faisant intervenir des coordonnées (u_i, v_i) non entières, une interpolation bilinéaire permet la détermination

d'une valeur à utiliser dans l'image d'origine.

Comme vu au [chapitre 1](#), la reprojection d'une image fait intervenir un rééchantillonnage et comporte des risques d'apparition de moiré si le pas d'échantillonnage est trop grand par rapport aux fréquences présentes dans l'image. Pour éviter d'avoir à utiliser des méthodes de lissage coûteuses, il est donc nécessaire de choisir des valeurs de Δ_θ et Δ_ϕ suffisamment petites pour garantir un échantillonnage correct. On pourra néanmoins noter que la résolution théoriquement élevée en périphérie extérieure du champ de vision de la caméra est dégradée par un flou important, qui filtre naturellement les hautes fréquences dans cette zone. Il est donc possible d'échantillonner avec des pas plus petits que la résolution angulaire théorique maximale de la caméra, afin d'éviter une taille d'image rectifiée trop importante. Il faut également noter que l'échantillonnage en coordonnées sphériques réalise un échantillonnage très dense près des pôles (ϕ proche de 0 ou π radians) mais relativement épars dans la région proche de l'équateur (ϕ proche de $\pi/2$ radians). C'est donc cette dernière région qui doit être considérée pour la détermination des pas d'échantillonnage à utiliser.

Il est intéressant de noter que les épipoles sont situés aux pôles des sphères. À cause du champ de vision large de la caméra, les épipoles sont observables en deux positions différentes, correspondant à chacun des deux pôles des sphères. Lors de l'échantillonnage en coordonnées sphériques, les pôles sont "étalés" sur deux lignes complètes (un ϕ fixé et $\theta \in [0, 2\pi[$). Les épipoles n'ont donc ni une position ponctuelle ni une direction correspondant à un point à l'infini dans les couples sphériques rectifiés, comme cela est habituellement le cas.

La figure [5.12](#) montre deux couples rectifiés. Seul l'intervalle utile des θ est représenté. L'axe correspondant aux différents θ (resp. ϕ) est vertical (resp. horizontal). Le pas d'échantillonnage est de $1/4$ de degré par pixel en θ et ϕ . Les homologues sont approximativement alignés sur les mêmes colonnes. Dans l'exemple (a), la platine rotative a tourné d'approximativement 180° entre les deux prises de vue. Dans, le (b), l'écart est de seulement 90° . L'image d'origine utilisée pour générer l'image supérieure de chacun des couples est la même et est visible sur la figure [5.4](#).

Les fonctions de projections $proj_1(P)$ et $proj_2(P)$ peuvent être différentes et il est donc possible de réaliser la rectification même lors d'un changement des paramètres intrinsèques. De même, la rectification est possible pour des types de caméras différents, par exemple une caméra paracatadioptrique et une caméra standard, la seule contrainte étant que les caméras soient centrales.

Comme proposé par Gonzalez-Barbosa et al., il est possible de stocker le résultat de la rectification dans des tables de correspondances réutilisables, à conditions que les paramètres intrinsèques et extrinsèques des caméras soient fixes. Dans notre application, ceci est particulièrement intéressant. En effet, les paramètres intrinsèques sont fixes et les paramètres de pose relative entre deux poses reliées par une rotation donnée de la platine sont aussi fixes. Par exemple, une rotation de 18° de la platine correspond à une rotation et translation fixe, quelle que soit son orientation de départ. Il est donc possible de précalculer les rectifications épipolaires à appliquer à un couple d'images relié par une rotation de 1 à n incréments angulaires θ_i (voir la figure [5.3](#))

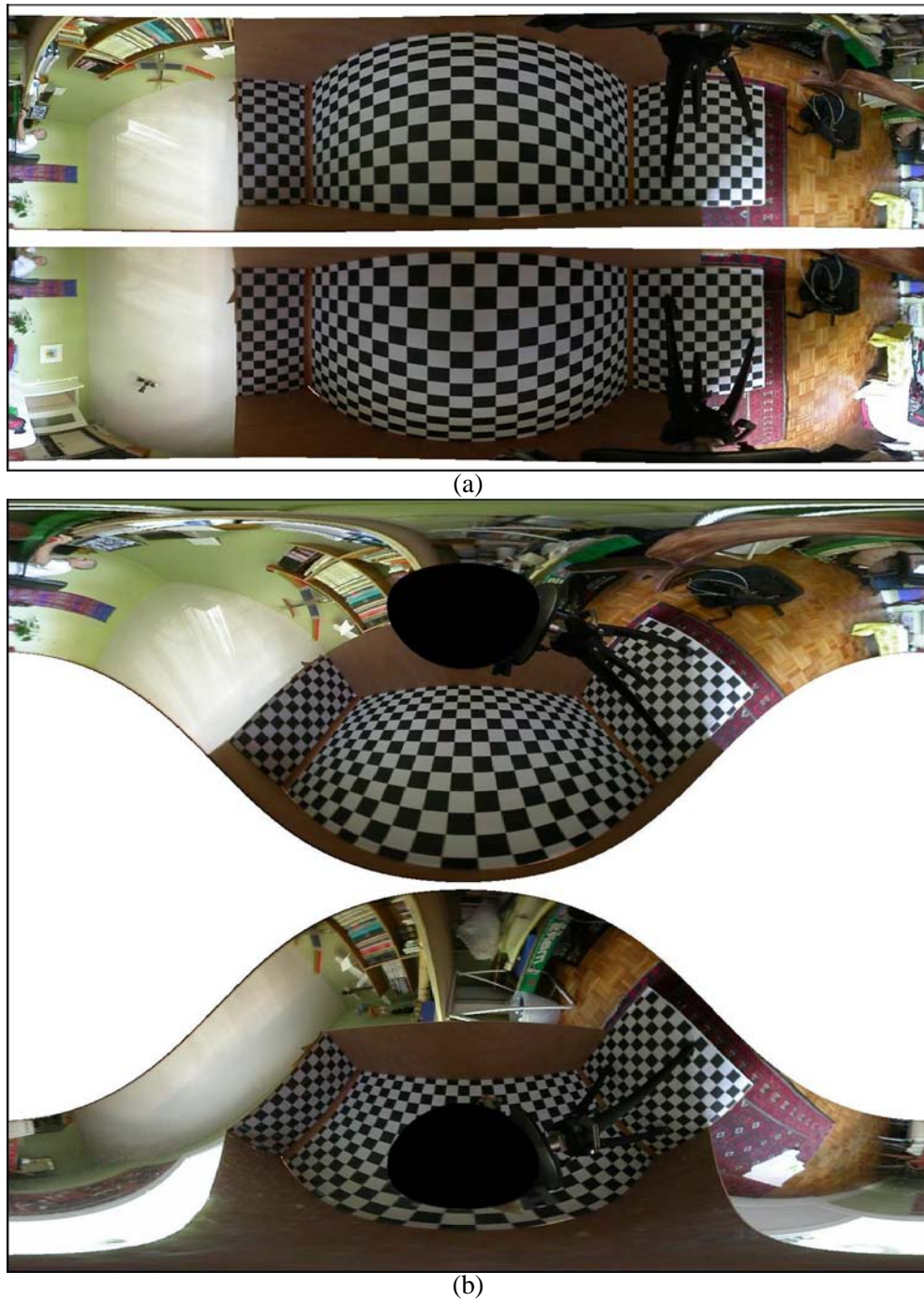


FIG. 5.12 – Couple d'images rectifiées (a) pour deux vues acquises lors d'une rotation de 180° de la platine, (b) pour deux vues acquises lors d'une rotation de 90° de la platine.

Si la rectification épipolaire sur la sphère paraît complexe à première vue, il est intéressant de noter qu'à bien des égards, elle est plus simple à utiliser que la rectification sur un plan. Certes, l'échantillonnage de la sphère fait intervenir des fonctions trigonométriques, alors que la rectification plane ne nécessite que l'application d'une homographie, qui est une application linéaire. Hormis ce fait, la rectification sur la sphère ne nécessite pas de choix approprié de l'orientation de y_{r1} . En effet, une rotation autour de z_{r1} se traduit simplement par un offset dans l'image sphérique, alors qu'une rotation autour de y_{r1} fait apparaître une transformation perspective dans le cas de la rectification sur un plan. De plus, le choix du pas d'échantillonnage sur la sphère est directement lié à une mesure angulaire, et il n'est donc pas nécessaire d'aborder des questions d'échelles, comme c'est le cas lors du choix de la taille de (u_1, v_1) et (u_2, v_2) pour la rectification sur un plan. Enfin, la taille d'une image en projection sphérique est bornée et contient un champ de vision omnidirectionnel. À l'opposé, il est nécessaire d'effectuer un choix sur la taille des images rectifiées lors de la rectification sur un plan, pour que celles-ci contiennent tout le champ visuel correspondant aux images d'origine.

On pourra également noter une autre application à la rectification épipolaire sur la sphère. Elle permet en effet de générer des anaglyphes omnidirectionnels, tel que celui présenté en Annexe F.

5.3.2.2 Géométrie épipolaire dans des couples cubiques rectifiés

Comme il vient d'être montré, la rectification épipolaire correspond à une reprojection des images. Le cube permet, comme la sphère, de représenter un champ de vision omnidirectionnel. S'il paraît à première vue moins adapté pour la reprojection de couples d'images, il présente l'avantage d'utiliser des projections planes pour ses faces, ce qui autorise l'emploi de méthodes destinées aux caméras standard. Il peut donc être utilisé pour reprojeter les images.

La figure 5.13 (a) montre la géométrie associée à la rectification épipolaire sur le cube. Kangni et Laganière ont étudié dans [51] la rectification de couples d'images acquises avec la caméra Ladybug, montrée au chapitre 1. Cette caméra fournit directement les images sous la forme de six projections sur les faces d'un cube et la méthode proposée par les auteurs permet de passer d'une projection sur le cube d'origine à une projection sur un cube aligné comme sur la figure 5.13 (a). Ce principe de reprojection des images est en fait applicable quel que soit le type de projection correspondant aux images d'origine.

Les images d'origine sont reprojétées sur les faces de cubes définis dans les repères ω_{r1} et ω_{r2} par les plans $F_1 : x_r = 1$, $F_2 : y_r = -1$, $F_3 : x_r = -1$, $F_4 : y_r = 1$, $F_5 : z_r = 1$, $F_6 : z_r = -1$. Comme visible sur la figure 5.13 (b), les homologues sont alignés sur les couples d'images obtenues par échantillonnage des faces F_1 à F_4 .

Par rapport à la méthode proposée par Kangni et Laganière, deux améliorations sont proposées : l'élargissement des faces des cubes et l'échantillonnage radial des deux faces F_5 et F_6 contenant les épipoles.

Des faces élargies : Afin d'éviter d'avoir à chercher les homologues sur des faces différentes des cubes et pour assurer que la zone entourant un point d'intérêt soit visible sur une unique

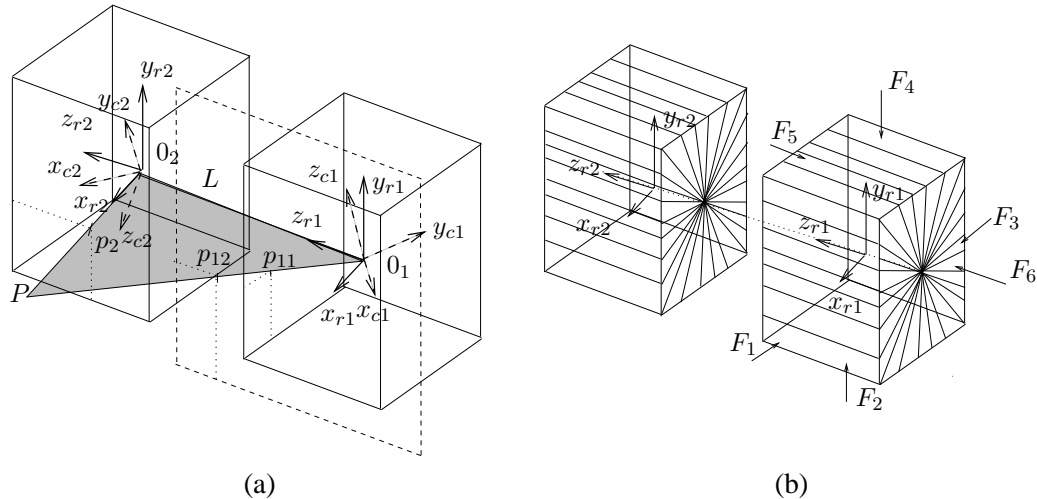


FIG. 5.13 – (a) Rectification épipolaire à la surface de cubes. (b) L'intersection de quelques plans épipolaires avec les deux cubes et numérotation des faces.

image (par exemple pour le calcul d'un descripteur de type SIFT), il est nécessaire de calculer les projections sur les faces du cube en prévoyant un dépassement. Sur la figure 5.13 (a), le point P est projeté en p_2 sur la face F_1 du cube pour la caméra 2. Il est projeté en p_{11} sur la face F_5 du cube pour la caméra 1. En considérant une face F_1 plus grande (marquée en pointillés sur la figure), le point P est observé sur la face F_1 en p_{12} , ce qui permet un appariement efficace avec p_2 , les homologues étant sur les mêmes lignes et mêmes faces.

La figure 5.14 montre le couple de la figure 5.12 (a) projeté sur des cubes dont les faces F_4 , F_1 et F_2 sont représentées. Le dépassement est tel que la largeur et la hauteur des faces sont doublées. Il y a donc redondance entre les différentes faces, par exemple l'avion sur l'étagère, présent sur les images (b) et (d) ou la lampe au plafond sur les images (a) et (c).

Échantillonnage radial : Pour permettre une recherche efficace des points homologues, il est préférable qu'ils soient disposés sur les mêmes lignes, pour les raisons évoquées plus haut. Une légère adaptation de la méthode proposée par Kangni et Laganière permet d'adapter les images correspondant aux faces F_5 et F_6 pour qu'elles respectent cette contrainte. L'échantillonnage doit simplement être fait de manière radiale pour obtenir des homologues alignés. Les centres des faces F_5 et F_6 , qui sont les épipoles, sont utilisés comme centres pour l'échantillonnage radial. Sur la figure 5.15, les deux faces F_5 et F_6 sont représentées avec l'échantillonnage radial. Les lignes (resp. colonnes) correspondent à des angles (resp. rayons) différents. Afin d'assurer un dépassement des faces, le rayon est échantillonné jusqu'à des valeurs supérieures à $\sqrt{2}$. Dans cet exemple, le rayon maximal, correspondant à la colonne de droite dans le couple (a)-(b) et à celle de gauche dans le couple (e)-(f), vaut 2.

Comme pour la projection sphérique, les épipoles sont "étalés" sur deux colonnes avec cette représentation.

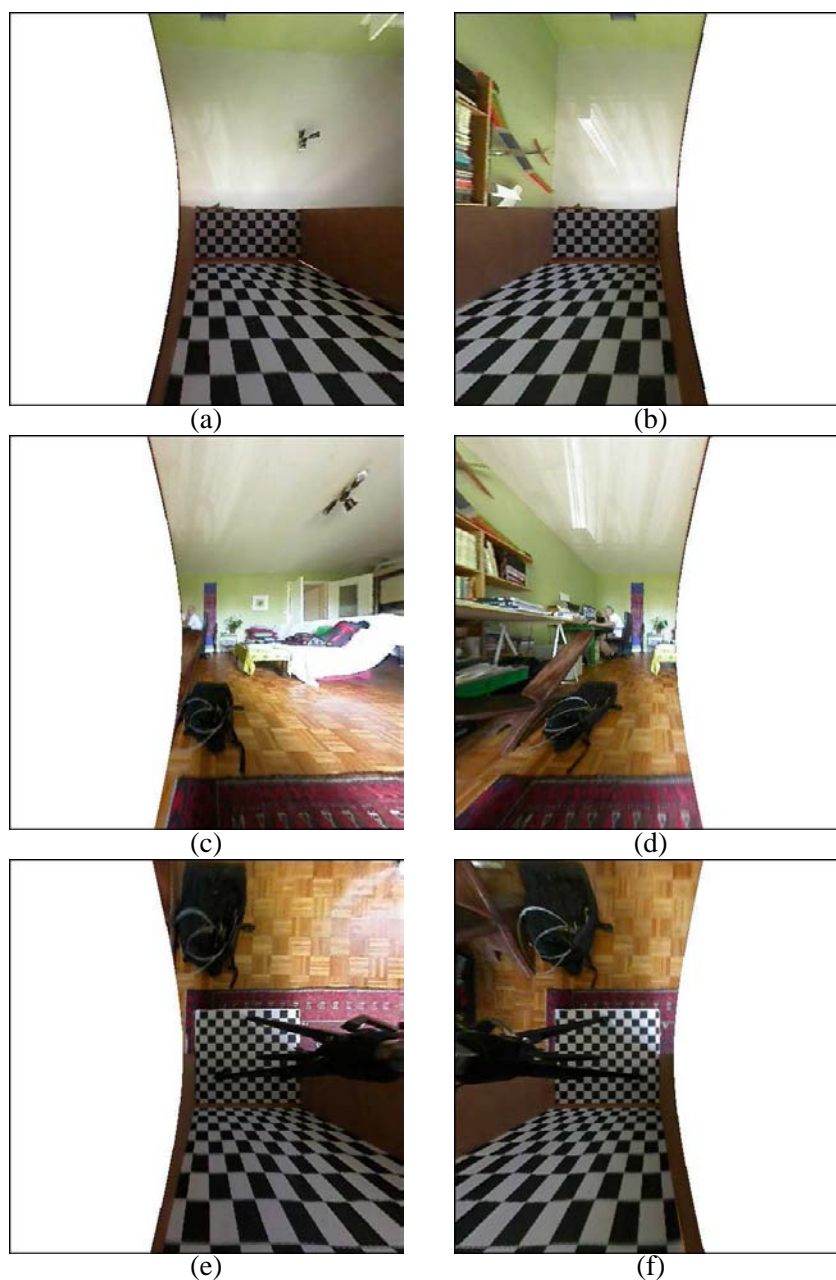


FIG. 5.14 – Rectification épipolaire à la surface d'un cube. (a) et (b) : faces F_4 , (c) et (d) : faces F_1 , (e) et (f) : faces F_2 .

5.3.2.3 Géométrie épipolaire dans des couples cylindrique rectifiés

On pourra noter l'existence des travaux de Bunschoten et Krose [169, 13, 170, 171] destinés à des applications de robotique mobile. Les auteurs projettent les images d'une caméra

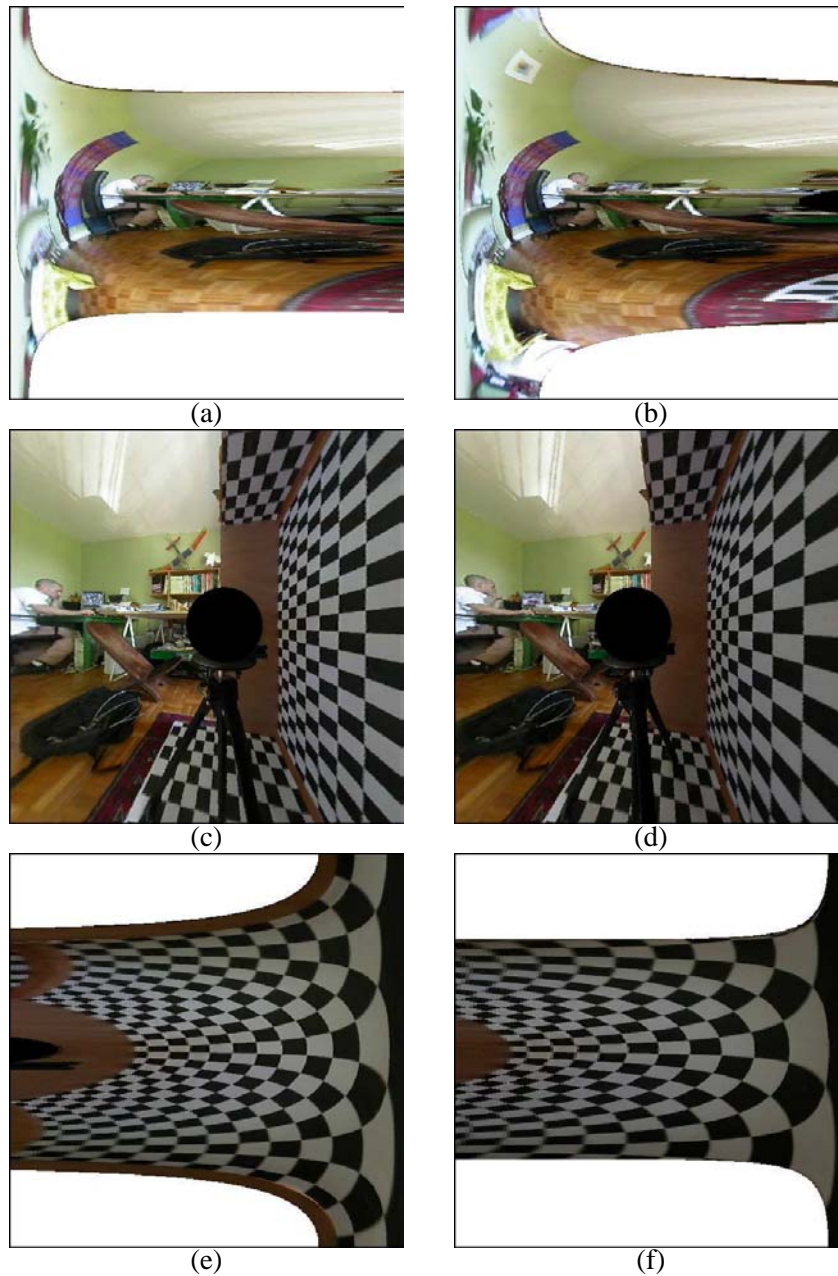


FIG. 5.15 – Rectification épipolaire à la surface d’un cube avec échantillonnage radial sur deux faces. (a) et (b) : faces F_5 , (c) et (d) : faces F_1 , (e) et (f) : faces F_6

catadioptrique verticale sur deux surfaces différentes : Premièrement des cylindres dont l’axe est confondu avec l’axe de révolution du miroir de la caméra et deuxièmement sur un plan perpendiculaire à cet axe. La seconde projection est utilisée comme capteur d’odométrie, afin d’estimer les petits déplacements du robot, alors que la première sert à la reconstruction 3D de

l'environnement. Les auteurs décrivent la géométrie épipolaire dans les couples cylindriques, menant à des courbes épipolaires sinusoidales. Ils présentent des résultats de reconstruction 3D multiscopique très convaincants.

5.3.3 Triangulation des points

La triangulation d'un point de la scène consiste en la détermination de ses coordonnées 3D à partir de plusieurs observations.

5.3.3.1 Triangulation des points dans les images d'origine

Dans le cas où les observations sont effectuées dans les images d'origine (avec ou sans distorsion), il est nécessaire de calculer les rayons de projection correspondants.

Une première estimée de la position du point 3D observée est obtenue en minimisant linéairement une distance algébrique à un ensemble de couples de plans, chacun de ces couples étant défini par le rayon de projection associé à une observation. Cette première mesure est obtenue très rapidement.

Ensuite la position 3D du point est raffinée afin de minimiser une erreur géométrique dans les différentes images. Le capteur ayant une résolution angulaire variable, il est préférable de minimiser une erreur angulaire entre le rayon de projection correspondant au point observé et le rayon correspondant au point reprojété plutôt que de minimiser une erreur de reprojektion pixellique dans l'image.

5.3.3.2 Triangulation des points dans les couples rectifiés

Dans le cas des couples rectifiés, la triangulation des points à partir de deux observations est simplifiée. Elle est effectuée en mesurant la disparité, qui est l'écart en unité pixellique le long de la ligne considérée, entre la projection du même point de la scène dans les deux images. Une fois la disparité mesurée, la triangulation dans le repère lié au couple rectifié est triviale, et l'obtention des coordonnées dans le repère de travail nécessite simplement l'application d'une transformation rigide.

5.3.4 Appariement des homologues

5.3.4.1 Dans les images d'origine non rectifiées

La recherche des homologues dans les images fournies par la caméra présente l'avantage que les images n'ont pas été rééchantillonnées, permettant a priori une recherche plus précise en terme de positionnement. Lhuillier, dans [54], propose d'utiliser des fenêtres rectangulaires entre lesquelles il calcule un score de corrélation ZNCC sans utiliser les contraintes épipolaires a priori. L'utilisation de fenêtre rectangulaire est rendue possible par les conditions d'acquisition des images. Celles-ci sont acquises à intervalle régulier, après une translation perpendiculaire à l'axe de révolution du miroir d'approximativement un mètre.

Dans le cas plus général, y compris celui du capteur présenté ici, il n'est pas possible d'utiliser directement des fenêtres rectangulaires dans l'image car le voisinage d'un point se projette trop différemment dans les différentes images. Svoboda et Pajdla proposent dans [172] l'emploi de fenêtres adaptées. En réalité, ceci revient à rééchantillonner localement l'image comme cela est proposé avec la rectification épipolaire sphérique.

Mikusic, dans [52], utilise le descripteur MSER [40], adapté aux caméras standard pour de grands écarts intra-oculaires, pour détecter les correspondances entre des images de caméras catadioptriques.

Le descripteur SIFT [39] devrait être utilisable dans certaines conditions mais les tests effectués sur nos images n'ont pas été concluants. Ceci est probablement dû au fait que la projection effectuée par la caméra est trop différente de la transformation affine pour laquelle a été développé le descripteur.

On pourra noter les travaux de Sato et Yokoya [50] concernant l'établissement de correspondances entre des points d'intérêt détectés dans des images omnidirectionnelles. Au lieu d'utiliser une mesure de similarité, ils utilisent une fonction de score appelée TNIP. Leur technique, relativement simple, consiste à compter le nombre de points d'intérêt pouvant être l'image d'un même point 3D dans les différentes images. Si ce nombre est important, le point 3D est triangulé à partir des différents points d'intérêt. Cette approche présente deux avantages. Tout d'abord le coût calculatoire est très faible et comme il n'est pas nécessaire d'effectuer de mesure de similarité, il n'est pas nécessaire de rééchantillonner les images.

5.3.4.2 Dans les images d'origine rectifiées

Dans le cas où les distorsions optiques ont été corrigées, la seule différence est que les homologues peuvent être recherchés sur les cercles épipolaires au lieu de courbes plus complexes. La rectification fait par contre perdre l'intérêt d'avoir des images non rééchantillonnées.

5.3.4.3 Dans des couples rectifiés à la surface d'une sphère

Un couple d'images rectifiées épipolairement sur la sphère permet la détection efficace des correspondances. En effet, comme les homologues sont situés sur les mêmes lignes, la recherche est purement 1D. Les homologues sont détectés par de fortes corrélations entre des fenêtres rectangulaires englobant l'image de leur voisinage. Différents critères de corrélations peuvent être employés, par exemple le SSD (somme des différences de valeurs de pixels aux carrés) ou le ZNCC (Corrélation croisée normalisée), plus apte à gérer les variations d'éclairage.

On pourra noter les travaux de Scharstein et Szeliski [173], opérant une comparaison des différentes méthodes d'appariement denses. Les travaux de Chambon et Cruzil [174, 175] présentent aussi un intérêt, notamment [176] où les auteurs proposent une gestion plus approfondie des images couleurs.

La rectification épipolaire permet l'emploi de méthodes de recherches d'homologues pour une ligne complète, plutôt qu'individuellement pour chaque point. La programmation dyna-

mique (par exemple Drouin et al. [177]) est un exemple d'une telle méthode qui tend à fournir une bonne continuité dans la carte de profondeur estimée.

Notons que, par rapport aux couples rectifiés sur des plans (pour les caméras standard), les couples sphériques présentent un caractère cyclique sur l'axe θ . Il faut donc prévoir, lors de la comparaison des fenêtres, de gérer le modulo 2π sur θ pour définir le voisinage.

5.3.4.4 Dans des images à la surface d'un cube

L'intérêt de projeter les images sur les faces planes d'un cube est qu'il est alors possible d'utiliser des descripteur tels que SIFT pour l'appariement épars des homologues, même dans le cas où les cubes n'ont pas été alignés pour la rectification épipolaire. Ceci permet la détection de correspondances entre les différentes images, très utile pour la multiscopie.

Dans le cas où les cubes sont alignés par la rectification épipolaire, la projection sur le cube permet l'appariement dense sur les six faces, comme dans le cas des projections sphériques, en employant des mesures de corrélation sur des fenêtres rectangulaires.

5.3.4.5 Multiscopie

Il est possible de tirer parti des multiples images de la scène pour rendre la reconstruction plus précise et surtout plus robuste. Les approches utilisant des tenseurs trifocaux et quadri-focaux permettent de modéliser les contraintes existantes entre les différentes projections des mêmes points de la scène. Malheureusement, il est difficile d'exprimer ces contraintes pour plus de quatre images.

Les travaux de Okutomi et Kanade [178] présentent une approche applicable pour une caméra standard en translation permettant la détection des correspondances dans un nombre quelconque d'images. Au lieu de considérer un score d'appariement entre deux points dans deux images, un score est attribué pour différentes distances le long du rayon de projection correspondant au pixel considéré dans une des images (La droite D associée au pixel p_1 dans la figure d'exemple 5.16). Pour une certaine distance, la somme des scores de corrélation est optimale et la position du point est obtenue directement (P dans la figure d'exemple).

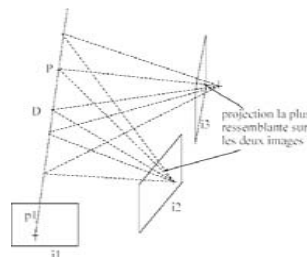


FIG. 5.16 – Triangulation du point P par multiscopie.

Dans la configuration étudiée par Okutomi et Kanade, la disparité est liée à l'inverse de la distance le long du rayon de projection et il en découle un algorithme très efficace.

Cette méthode peut être simplement adaptée pour notre application. Il faut néanmoins prendre soin de comparer les voisinages de points de façon adéquate, en utilisant soit les descripteurs proposés par Svoboda et Pajdla dans [172] ou Ieng et al. dans [179], soit en traitant les images rectifiées épipolairement. Dans ce dernier cas, la mesure de corrélation entre un point p dans une image 1 et des points dans des images 2, 3 et 4 par exemple, doit être effectuée en considérant les différents couples rectifiés, 1-2, 1-3 et 1-4, dans lesquels le voisinage de p dans l'image 1 est rectifié différemment.

5.3.5 Reconstruction de texture

La reconstruction de la texture correspond ici à la détermination d'une image à plaquer sur une facette triangulaire plane.

5.3.5.1 Le cas simple

Dans le cas le plus simple, la facette est entièrement vue sur une image, sans occultation ni de la scène, ni du capteur. Dans ce cas, l'image contient toute l'information nécessaire à la détermination de la texture. La résolution de l'image de texture est choisie puis la couleur de chacun de ses pixels est déterminée de la façon suivante :

Trois coins de l'image de texture correspondent aux sommets de la facette. Les coordonnées de chacun des pixels de cette image s'expriment sous la forme d'une somme vectorielle de deux vecteurs définis par deux couples de points, pondérés par deux facteurs, appelés coordonnées barycentriques. Les coordonnées 3D correspondantes dans la scène s'obtiennent à partir de la même somme vectorielle, cette fois appliquée aux coordonnées 3D des coins de la facette. Une fois la position d'un point obtenu dans la scène, il suffit de le projeter dans l'image de la caméra par le modèle pour obtenir sa couleur.

La figure 5.17 (a) montre un exemple de reprojection de texture sur deux facettes triangulaires.

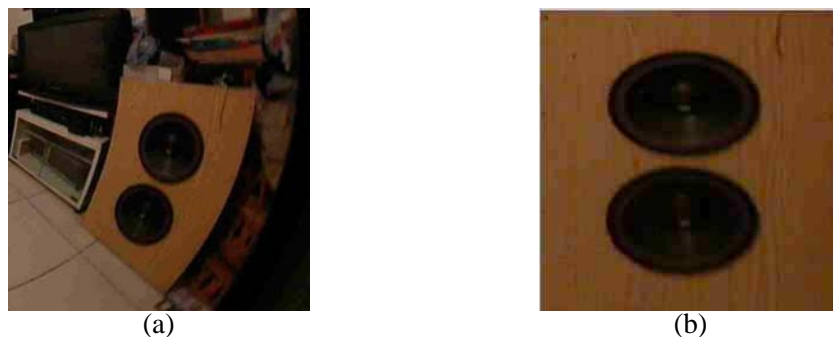


FIG. 5.17 – (a) Portion d'une image de la caméra. (b) Exemple de reprojection de texture

5.3.5.2 Le cas avec occultation

Assez souvent, lorsque la facette est trop grande pour tenir sur une unique image, elle est occultée. Elle peut l'être par la scène elle-même à cause de la parallaxe ou bien par le capteur.

Dans ce cas, il est nécessaire soit de partitionner la facette afin que chaque portion soit observable intégralement sur une image, soit calculer une texture en intégrant les données des différentes images. C'est cette dernière approche qui a été préférée ici. Considérons deux cas distincts :

Dans le premier, l'objet occultant n'est pas connu, c'est à dire qu'il n'a pas été reconstruit en 3D et la facette observée est intégralement observée dans toutes les images. Dans ce cas, la reconstruction de la texture doit s'appuyer uniquement sur une étude statistique. Une texture est générée pour chaque image acquise.

L'exemple de la figure 5.18 montre le damier de la mire situé sous le trépied et occulté par celui-ci. Les images (a) à (d) montrent la texture calculée à partir de quatre images différentes parmi les vingt. À cause de la parallaxe, le trépied n'est pas projeté aux mêmes positions dans ces différentes textures.

Le traitement suivant est effectué individuellement pour chaque pixel de la texture. La valeur médiane des différentes textures calculées est utilisée pour générer une nouvelle image, sur laquelle les extrema sont donc éliminés. Ce traitement très simple permet par exemple d'éliminer le trépied sur l'image (e) de la figure 5.18.

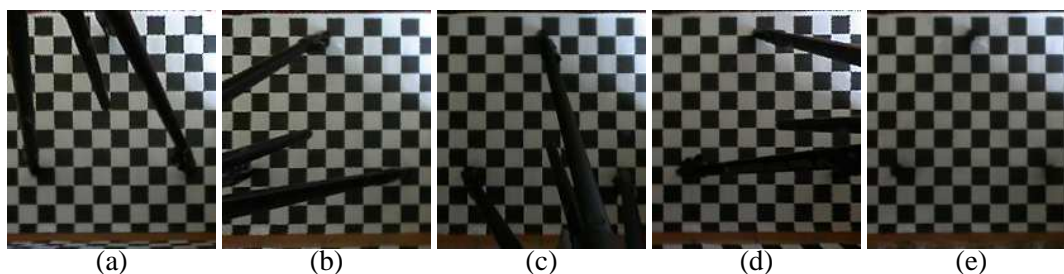


FIG. 5.18 – (a-d) Quatre textures générées à partir de différentes images. (e) La texture calculée à partir des 20 images, seule la zone située sous les branches du trépied (3 cercles) ne sont vues sur aucune image.

Dans le second cas, plus complexe, l'occultation peut être due au champ visuel limité de la caméra ou à un objet déjà reconstruit. La reprojexion de la texture depuis les différentes images laisse donc apparaître des zones non renseignées, soit parce qu'elles sortent du champ de vision, soit parce que l'objet reconstruit est positionné devant (Ceci est détecté grâce à une image associant une profondeur à chaque pixel de l'image, appelée Z-BUFFER). Le calcul de la valeur médiane doit alors être effectué uniquement à partir des images contenant une information pour le pixel considéré.

Il est important de noter que la modélisation du capteur est particulièrement sensible pour le calcul de texture à partir de plusieurs images. En effet, si l'erreur de reprojexion est im-

portante, les différentes textures correspondant à une même facette et obtenues à partir des différentes images ne se superposent pas. Il en résulte une texture calculée floue. L'exemple de la figure 5.18 fonctionne particulièrement bien car il est dans une zone contenant des points d'étalonnage, et donc dotée d'une bonne précision de reconstruction, comme il sera discuté plus loin.

La continuité de texture entre deux facettes voisines est importante pour obtenir un rendu réaliste. Les travaux de Niem et Broszio [180] traitent de ce sujet, et proposent une méthode de choix des images d'origine pour les différentes facettes afin de garantir un rendu convenable. Ils proposent aussi la synthèse de texture pour les facettes non visibles en utilisant une combinaison des textures des facettes voisines.

On pourra finalement noter l'existence de méthodes permettant la complétion de texture, par exemple celle de Criminisi et al. dans [181].

5.3.6 Résultats

5.3.6.1 Appariement automatique et triangulation

Les coins sont tout d'abord détectés dans les différentes images non rectifiées en utilisant le détecteur de Harris. Les coordonnées correspondantes sont calculées dans les couples sphériques rectifiés (exemple sur la figure 5.19 (a), les points de Harris reprojétés sont montrés par des croix sur l'image en projection sphérique, tournée à 90° par rapport aux précédentes).

Les homologues de ces points sont recherchés dans un couple d'images numéroté (4-5), l'image 4 étant celle dans laquelle les points de Harris ont été détectés et l'image 5 celle correspondant à une rotation d'approximativement 18° de la platine. La disparité correspondant au meilleur score de corrélation (ZNCC ici) dans des fenêtres rectangulaires de dimensions $7*7$ est utilisée pour trianguler le point.

Les homologues ont dû être recherchés non pas sur la même ligne mais dans une zone de 7 pixels autour de la même ligne. Ceci est lié à une mauvaise modélisation du capteur qui sera discutée à la section suivante. Comme les deux homologues peuvent être situés sur des lignes différentes, la valeur θ utilisée pour la triangulation est calculée par moyennage des valeurs pour les deux images du couple.

Cette méthode permet la mise en correspondance des points menant à la reconstruction 3D de la mire présentée sur la figure 5.19 (b) sous différents angles de vue, le repère étant lié au couple sphérique rectifié (4-5). Des segments ont été ajoutés sur les figures afin de juger du parallélisme et de l'orthogonalité des plans reconstruits.

L'image numéro 3 (correspondant à une rotation d'approximativement 18° de la platine dans le sens inverse) est utilisée pour vérification. Les coordonnées triangulées à partir du couple (4-5) sont reprojétés dans le couple d'images numéroté (4-3) et les fenêtres rectangulaires sont utilisées dans ce nouveau couple pour obtenir un nouveau score de corrélation. Si ce second score de corrélation est bien plus mauvais que le premier, le point est rejeté. La figure 5.19 (c) présente le résultat de la reconstruction après validation en utilisant le couple (4-3). Les points de Harris sont affichés par une croix bleue dans l'image (a) si ils ont pu être triangulés et validés, sinon ils sont affichés par une croix rouge.

Les points correctement reconstruits sur la mire sont compris dans une zone de l'ordre de 4cm autour des trois plans définis par la mire.

Il est important de noter que la reconstruction de la mire est compliquée par le fait qu'elle présente des motifs répétitifs, raison pour laquelle une troisième image doit être utilisée pour validation.

5.3.6.2 Développement d'une application d'acquisition multiscopique omnidirectionnelle

Il n'est pas facile de calculer un modèle 3D constitué de facettes triangulaires texturées à partir d'un nuage de points tel que celui obtenu par l'appariement automatique. Pour cette raison, une application sous Windows a été développée afin de permettre la reconstruction interactive à partir d'une série d'images acquises par le capteur.

Le système proposé est assez semblable à celui de Doubek et Svoboda, présenté dans [182], qui consiste à définir des correspondances entre des points dans les images. La différence principale tient au fait que le mouvement particulier du capteur est exploité pour permettre une reconstruction totalement omnidirectionnelle. De plus, le nombre de projections de points utilisées pour la triangulation n'est limité que par le nombre d'images acquises.

Gaspar et al. proposent dans [183] une méthode de reconstruction 3D basée sur la définition de points dans une image obtenue avec une caméra catadioptrique avec miroir sphérique. Leur approche est monoscopique et l'utilisateur doit donc définir des contraintes d'alignement et de coplanarité entre les points 3D à reconstruire. Ceci n'est pas nécessaire avec notre approche, grâce à la multiscopie. Sturm propose le même type d'approche pour la caméra paracatadioptrique dans [184].

L'interface du logiciel développé est principalement basée sur la définition de points et de facettes dans les images. L'utilisateur est aidé dans sa tâche grâce à l'affichage des cercles épipolaires pour la recherche des homologues. La création de facettes triangulaires est effectuée grâce à la sélection de trois points. La texture de chacune des facettes peut être, au choix, calculée à partir d'une image donnée ou bien à partir de toutes pour gérer les occultations.

La figure 5.20 montre deux captures d'écran de l'application développée. Sur la gauche est visible l'image de la caméra en cours de traitement, dans laquelle l'utilisateur définit des points et des facettes triangulaires. Sur la droite est visible le modèle 3D reconstruit en temps réel à partir des données fournies par l'utilisateur. La caméra virtuelle peut être déplacée et orientée à souhait. En bas à droite figure une fenêtre informant l'utilisateur sur les données en cours d'édition, par exemple la position renseignée du point dans une image et sa position triangulée dans la scène.

La figure 5.21 montre quatre reconstructions 3D partielles d'une scène.

La précision de la triangulation est très variable en fonction de la profondeur des points observés. Elle est très bonne (de l'ordre du cm) pour les points à une distance correspondante à celle des points de la mire (de 30cm à 1m) mais se dégrade au delà (erreur de l'ordre de 20 cm pour des points situés à 3m).

La même remarque est vérifiée concernant la reconstruction des textures. Des erreurs dans

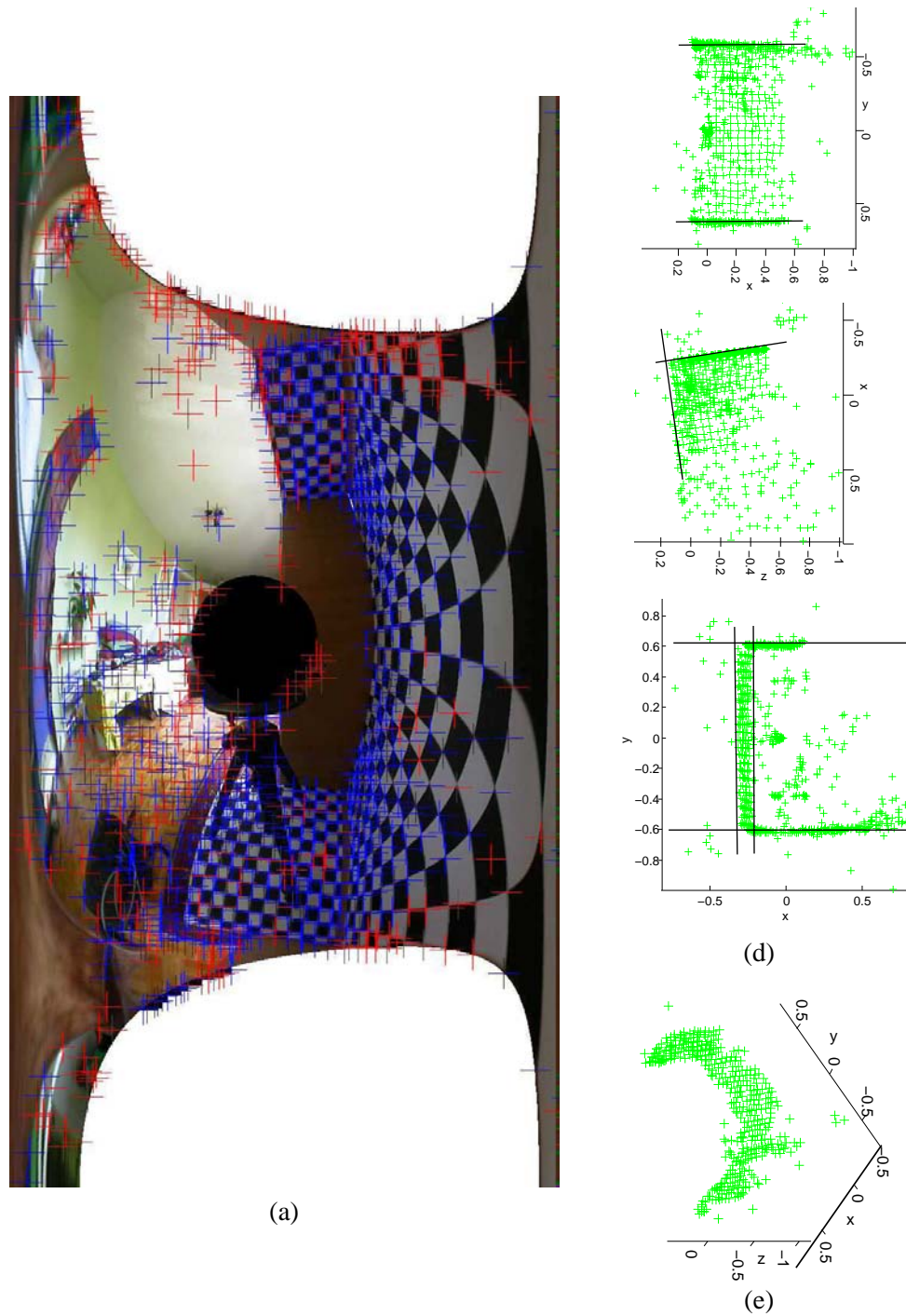


FIG. 5.19 – (a) Image en projection sphérique et points de Harris reprojétés depuis les images d'origine. (b) La reconstruction 3D effectuée à partir d'un couple, observée sous différents angles. (c) La reconstruction 3D effectuée à partir de deux couples.

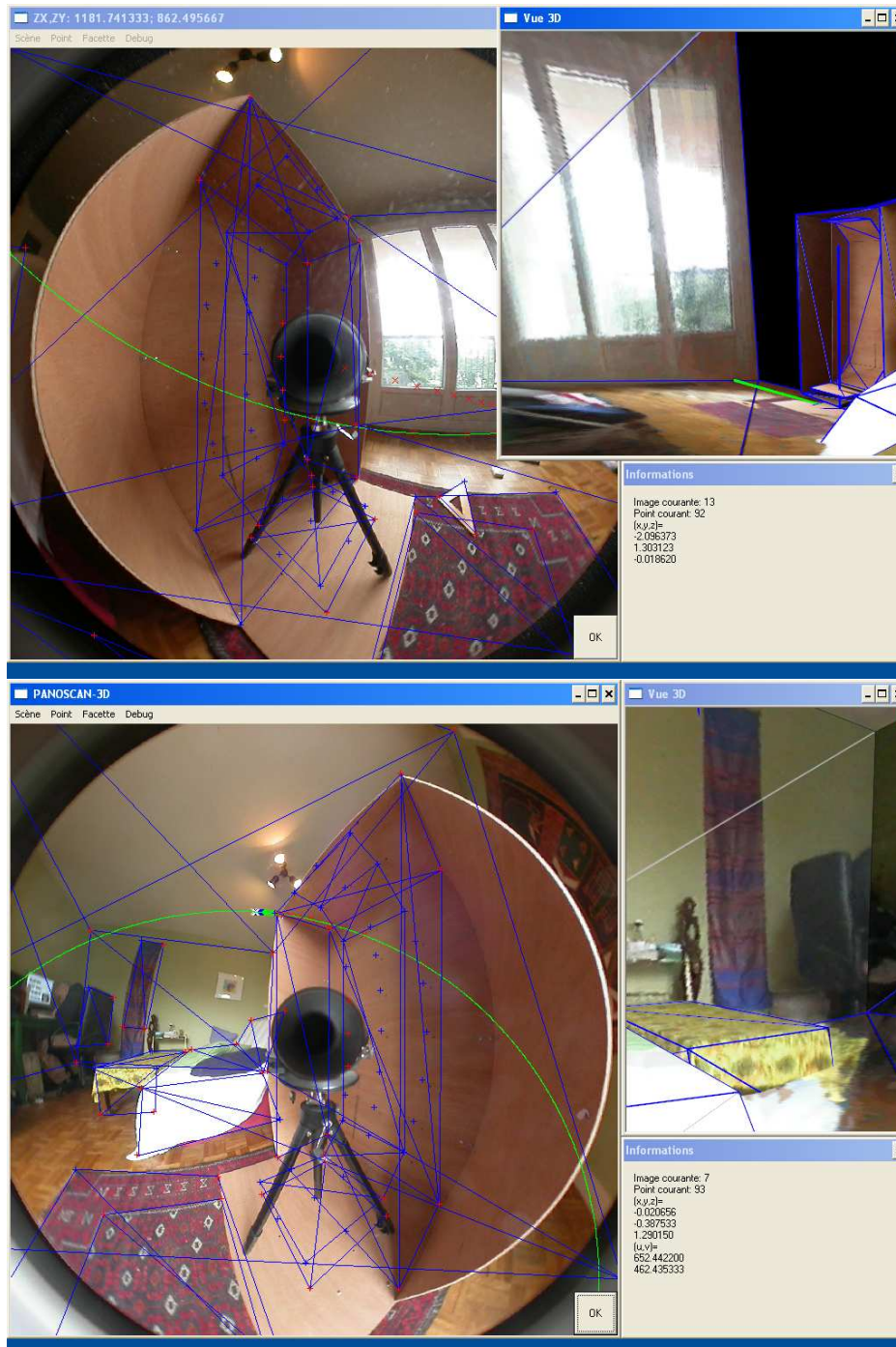


FIG. 5.20 – Deux captures d'écran de l'interface de l'application développée.

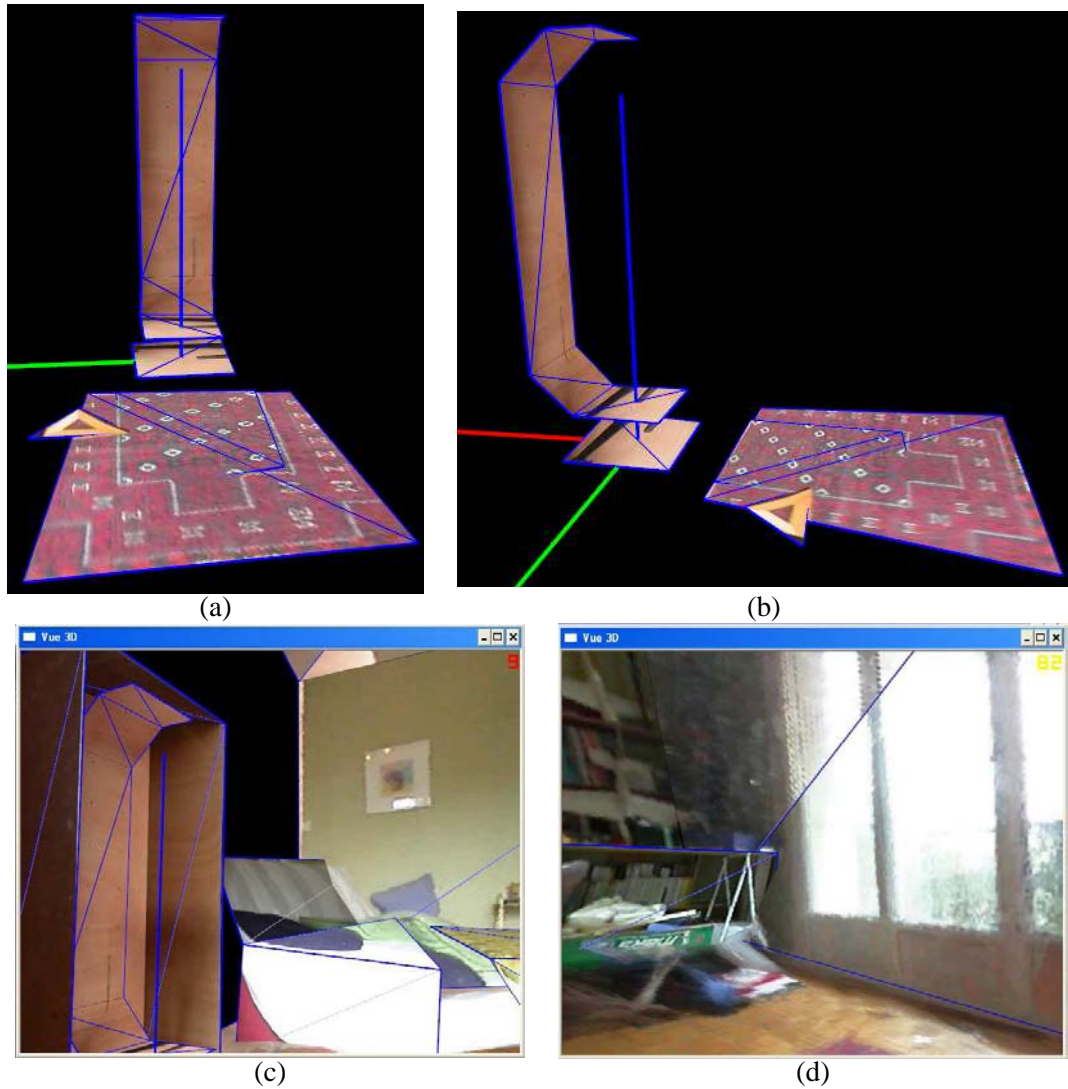


FIG. 5.21 – Parties du modèles 3D reconstruit observé sous différentes poses.

le calcul des textures ne sont pas très gênantes si l'on considère individuellement des facettes dont la texture est calculée à partir d'une image unique. Par contre dès qu'elle doit être calculée à partir de plusieurs images, ou que l'on considère des facettes voisines dont la texture provient d'images différentes (par exemple, la fenêtre, sur la figure 5.21 (d)), les erreurs deviennent visibles.

5.4 Critiques et perspectives

Les résultats obtenus pour la reconstruction 3D sont peu satisfaisants. Ceci est dû à plusieurs raisons qui sont détaillées ici, et auxquelles diverses solutions sont proposées.

5.4.1 Vers un meilleur modèle pour la caméra paracatadioptrique

Le premier problème est lié au fait que la caméra paracatadioptrique n'est pas parfaite. Comme noté par Lhuillier dans [54], la taille de la caustique n'est pas forcément négligeable devant la taille du miroir, auquel cas la caméra doit donc être considérée comme non centrale. Ceci est dû au fait que l'optique, qui est supposée télécentrique, ne l'est pas vraiment et aussi que l'alignement entre le miroir et l'optique n'est pas parfait.

C'est pour cette raison que les recherches décrites au chapitre 4 ont été effectuées. Malheureusement, comme il a déjà été expliqué, il n'a pas été possible, par manque de temps, d'utiliser les méthodes proposées sur des images réelles pour étalonner la caméra et donc utiliser les résultats pour ce capteur 3D.

Si l'on considère le caractère non central de la caméra, il apparaît judicieux de reconsidérer la conception du capteur 3D, car beaucoup des avantages escomptés grâce au caractère central de la caméra ne sont plus présents.

Ainsi, par exemple, pour la reconstruction à partir des couples sphériques rectifiés, présentées sur la figure 5.19, c'est la mauvaise modélisation par une caméra centrale qui oblige à rechercher les homologues sur différentes lignes.

Pour la reconstruction multiscopique, les erreurs importantes pour les points éloignés sont dues au fait que l'étalonnage a fourni les paramètres pour un modèle central minimisant l'erreur de reprojection pour les observations disponibles. Dans le cas d'une caméra vraiment centrale, la distance des points observés n'aurait eu pas d'influence, la minimisation étant effectuée sur un critère angulaire. Par contre, comme la vraie caméra est non centrale, et qu'elle est modélisée par un modèle central, il est nécessaire d'avoir des observations aux différentes distances, pour obtenir une approximation de modèle central valable pour les différentes distances. Ceci peut être vu comme une répartition de l'erreur due à l'approximation, pour qu'elle soit valable sur une plage de distance plus large. Le modèle central estimé peut par la suite servir à l'estimation du modèle non central, comme proposé par Micusik et al. dans [52].

Il est pour cela nécessaire d'utiliser la scène elle-même pour étalonner la caméra, la création d'une mire de la taille de la scène n'étant pas réalisable en pratique. Les coordonnées de ces points doivent donc être retrouvées en même temps que les paramètres de la caméra, en utilisant des méthodes de *Structure From Motion*. Ensuite, l'ajustement de faisceaux permet d'affiner le tout comme montré dans [52, 54].

On pourra rappeler que les recherches du chapitre 4 peuvent être utilisées pour le calcul de textures observées par une caméra non centrale.

Finalement, une meilleure modélisation du capteur d'image permettrait d'estimer plus précisément les coordonnées des projections des amers dans les images, en tenant compte du fait que le capteur n'est pas monochrome. Pour cela, il faudrait générer une image de luminance en

inversant les traitements qui ont été effectués pour obtenir l'image couleur à partir de l'image monochrome et du filtre de Bayer.

5.4.2 Vers un meilleur modèle pour représenter la scène

Le choix d'un modèle triangulaire texturé pour représenter la scène acquise n'est pas forcément le plus adapté à une méthode d'acquisition automatique. Deux autres approches sont proposées.

5.4.2.1 Le rendu basé image (IBR)

Le capteur a été développé afin de créer des modèles 3D destinés à la création d'images d'une caméra virtuelle se déplaçant dans le modèle. Il existe des méthodes permettant la création de telles images sans nécessiter la création du modèle 3D. Ces méthodes portent le nom d'*Image Based Rendering*, car le rendu d'images est basé sur d'autres images.

L'intérêt de cette approche est qu'elle permet le rendu réaliste indépendamment de la complexité de la scène. Elle nécessite par contre l'acquisition d'un grand nombre d'images pour échantillonner le plus de rayons de projection dans la scène (approches *Light Field*).

On peut donner la description très simplifiée suivante de cette méthode. La scène, au lieu d'être modélisée par sa géométrie est représentée sous la forme d'un ensemble de rayons de projection perçus par la caméra qui est utilisée pour l'acquisition. Une image peut alors être générée depuis une nouvelle position, en choisissant parmi les rayons acquis ceux qui correspondent à la caméra virtuelle associée à cette image.

Le lecteur pourra découvrir ce qu'est l'IBR grâce à l'article de Shum et Kang [185] et trouver plus d'informations dans les travaux de Bakstein et Pajdla [186, 187, 188, 189], de Hiroshi et al. [190]. Ng et al. décrivent dans [191, 190] leurs travaux sur la synthèse d'images par IBR à partir d'un échantillonnage de la scène avec une caméra catadioptrique, ce qui est assez proche de notre objectif de départ. Aliaga et al. traitent dans [192, 193] de la même problématique et investiguent particulièrement les aspects de compression des données (des milliers d'images) et de rendu temps réel.

Le livre de Shum et al. [194], bientôt disponible, sera consacré entièrement à l'IBR.

On pourra finalement noter l'existence des travaux de Komodakis et al [195], traitant des mosaïques déformables, qui sont à mi-chemin entre les méthodes IBR et les méthodes basées sur un modèle 3D de la scène.

5.4.2.2 Une modélisation et un rendu basé sur des plans avec zones transparentes

L'utilisation des facettes triangulaires texturées pour modéliser la scène nécessite un grand nombre de facettes pour modéliser correctement les objets à la géométrie complexe.

Le problème a été résolu dans les jeux vidéo pour le rendu, en utilisant des facettes avec des zones transparentes. Ainsi, un arbre par exemple, peut être modélisé par deux ou trois plans texturés sur lesquels sont mappés la texture de l'arbre. Un canal Alpha, utilisé en plus

des canaux Rouge, Vert et Bleu, permet d'encoder la transparence correspondant à chaque pixel. La géométrie complexe du contour de l'objet est ainsi ramenée dans sa texture, sous une forme bien plus facile à gérer.

Cette modélisation de la scène permet aussi d'éviter d'avoir à calculer un modèle à facettes à partir d'un nuage de points triangulés, ce qui est complexe, notamment à cause du bruit dans les mesures 3D.

En supposant les images acquises avec une caméra centrale, il est possible de détecter des homographies entre des parties d'images acquises par la caméra. Ceci a été effectué notamment par Vincent et Lagagnière [196, 197], Bartoli [198] ou Alon et al. [199] pour des images de caméras standard, mais il est possible d'appliquer les mêmes principes aux caméras catadioptriques centrales.

Une fois les différentes homographies détectées, les plans correspondants dans la scène peuvent être reconstruits et leurs textures calculées par reprojection des images comme montré précédemment. Si il y a des différences entre les mêmes pixels des différentes reprojctions, c'est qu'il y a soit occultation soit une disparité liée au fait que la surface reprojctée n'est pas dans le plan considéré. Dans ce dernier cas, la texture doit être transparente au niveau des pixels concernés.

Une scène complexe peut ainsi être modélisée par un nombre limité de plans texturés.

On pourra également noter l'existence de travaux de Sturm et al. [200] et Wilczkowiak et al. [201] sur la reconstruction monoscopique à base de plans et de parallélépipèdes.

5.4.3 Vers un plus grand nombre d'images pour l'acquisition

L'augmentation du nombre d'images acquises permettrait d'améliorer les résultats de reconstruction.

5.4.3.1 Diminution de l'écart angulaire entre les prises de vue

La diminution de l'incrément angulaire θ_i entre les positions des différentes acquisitions d'images permettrait de se rapprocher du système décrit par Kang [78], et d'effectuer un suivi des points dans la séquence d'images, ce qui améliorerait l'établissement des correspondances au prix d'une augmentation de la durée d'acquisition.

5.4.3.2 Translation du capteur

L'acquisition de la scène depuis une position unique du capteur ne permet sa reconstruction que si elle ne présente pas d'occultation par elle-même depuis la position du capteur. En pratique ce cas s'est révélé très rare et il est donc nécessaire d'acquérir la scène depuis différentes positions du capteur. Ceci, en plus de permettre l'acquisition de zones normalement occultées, permet une augmentation de l'écart intraoculaire, qui améliore la précision de reconstruction 3D. Par contre, contrairement au mouvement de rotation de la platine, qui peut être étalonné une fois pour toutes, les différentes positions du capteur devraient être estimées à chaque utili-

sation à partir des images acquises. L'orientation fournie par le capteur embarqué pourra alors être mise à profit pour simplifier l'estimation de pose relative entre les différentes acquisitions.

5.4.3.3 Combinaison de caméras de différents types

Pour augmenter la qualité de la reconstruction 3D, il serait judicieux d'utiliser des types de caméras différents. Il est par exemple possible d'utiliser conjointement des images acquises par une caméra standard et des images acquises par la caméra paracatadioptrique, comme proposé par Sturm dans [111].

Dans cette optique, le capteur 3D présenté ici serait utilisé pour estimer un modèle approximatif de la scène. Une caméra standard déplacée manuellement permettrait ensuite l'acquisition des détails et des zones occultées. L'estimation de la pose de la caméra standard serait alors effectuée à partir du modèle 3D approximatif de la scène et pourrait être aidée par le capteur d'orientation présenté au [chapitre 2](#).

Finalement, une étape d'ajustement de faisceaux sur toutes les images des différentes caméras permettrait de raffiner le modèle 3D de la scène.

5.4.4 Vers l'acquisition active

Les scènes d'intérieur sont souvent constituées de murs et sols unis, qu'il est difficile de reconstruire automatiquement à cause du manque de texture.

Bien que le capteur soit de type passif, il a été prévu qu'il puisse être équipé d'un système de projection de lumière permettant l'éclairage de la scène. Deux approches différentes ont été envisagées. Pour chacune, deux images sont acquises pour chaque position du capteur. L'une d'elle est sans éclairage, afin d'acquérir la texture de la scène et l'autre est avec éclairage, afin d'acquérir sa géométrie.

5.4.4.1 Utilisation d'un projecteur de ligne laser

Un projecteur de ligne laser a été fixé sur la platine rotative, dirigé à 90° par rapport à l'axe de révolution du miroir, de sorte que la ligne verticale soit verticale. Le projecteur est fixé à une distance d'approximativement 24 cm du centre optique de la caméra, ce qui génère une parallaxe permettant l'estimation de distance. Le projecteur utilisé a les caractéristiques suivantes : puissance de 3,6mWatt, bande des 650nm (rouge visible) et angle de projection de 45° .

Contrairement au système proposé par Orghidan et al. dans [202, 203, 204], le laser ne permet pas ici de reconstruire une coupe horizontale de la scène avec un champ visuel de 360° . La figure [5.22](#) montre le positionnement du laser sur la platine rotative du capteur développé. Il permet la projection d'une ligne laser verticale dans la scène telle que celle visible sur la figure [5.23](#) (a).



FIG. 5.22 – Le projecteur de ligne laser monté sur la platine.

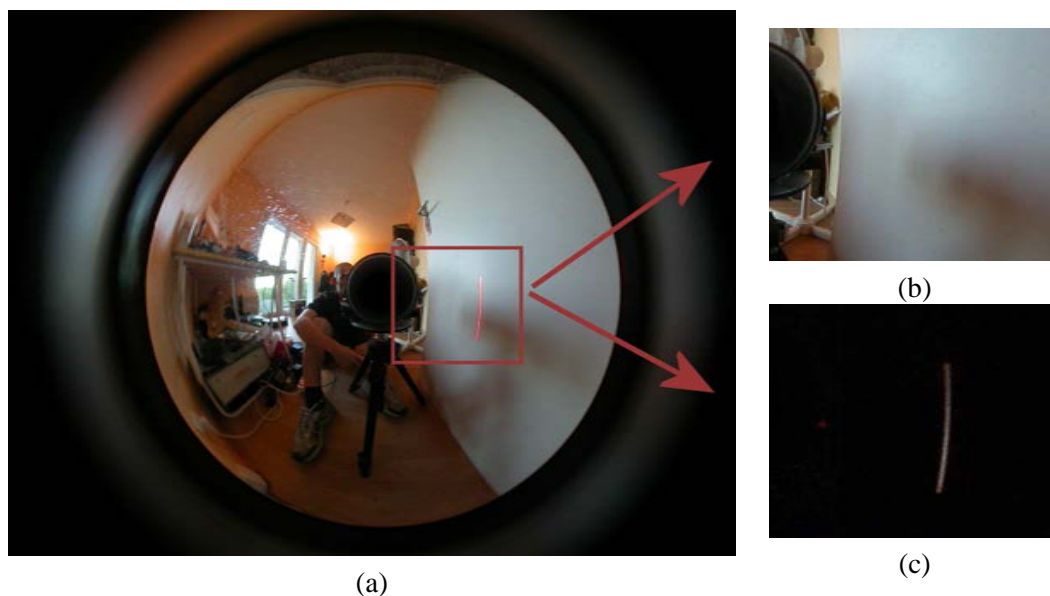


FIG. 5.23 – (a) Image de la scène éclairée par la ligne laser. (b) Zoom sur la portion de la scène éclairée et acquise sans le laser. (c) Différence entre l'image (b) et (a) pour la zone montrée en (b).

Le placement relatif du laser et de l'optique est étalonné une fois pour toute. Ensuite, un grand nombre d'images est acquis avec un incrément angulaire θ_i faible. L'image d'une seule ligne laser est obtenue par soustraction des images acquises avec le laser alternativement allumé (figure 5.23 (a)) ou éteint (figure 5.23 (b)) ce qui permet de segmenter facilement sa projection dans l'image de différence (figure 5.23 (c)), comme lors de l'étalonnage avec localisation de la mire par des LED.

Des essais pour acquérir une partie de scène à une distance de l'ordre de 40 cm du capteur ont permis de valider le principe de fonctionnement. La zone plane sur la porte de la figure 5.23 (a) a été reconstruite à partir de 20 bandes, sur une surface de 20cm par 35cm. Un plan a été ajusté sur le nuage de points obtenu et l'erreur maximale est de l'ordre du centimètre. Néanmoins, le projecteur utilisé s'avère trop peu puissant pour acquérir des scènes d'intérieur à une distance plus grande, car la puissance du laser est alors répartie sur une surface trop importante.

Pour améliorer les résultats de reconstruction, il serait donc souhaitable :

- D'utiliser un laser de plus forte puissance (au moins dix fois).
- D'utiliser un projecteur laser permettant la projection sur au moins 180°, afin de profiter du champ de vision large de la caméra paracatadioptrique et permettre une reconstruction omnidirectionnelle.
- D'écarter encore plus le laser du centre optique de la caméra, afin d'obtenir une plus grande parallaxe et d'augmenter la précision lors de l'acquisition d'objets lointains.

Anecdotiquement, on pourra noter l'existence des travaux de Mei et Rives [205] présentant l'utilisation conjointe d'une caméra catadioptrique et d'un système de mesure de distance utilisant un laser et basé sur le temps de vol. Ce type d'approche pourrait aider à la détermination de la géométrie de la scène, en réalisant des mesures ponctuelles, qui seraient ensuite complétées par l'approche purement basée vision.

5.4.4.2 Utilisation d'un spot omnidirectionnel

L'approche utilisant le projecteur de ligne laser nécessite l'acquisition d'un grand nombre d'images, deux par tranche à acquérir ce qui requiert un temps important.

Une autre approche a été envisagée pour combler cette lacune. Elle consiste à utiliser un spot lumineux omnidirectionnel placé dans la scène et équipé d'un cache permettant la projection de motifs quelconques. Ce projecteur permet d'ajouter de la texture aux zones unies de la scène et comme le projecteur est fixe, la texture ne varie pas entre les images, ce qui permet l'appariement.

5.5 Conclusions

Ce chapitre a décrit la conception, l'étalonnage et l'utilisation d'un capteur omnidirectionnel permettant la modélisation 3D d'intérieurs de bâtiments. L'intérêt majeur de ce capteur est son champ visuel complètement omnidirectionnel, qui permet l'acquisition en une seule fois des scènes simples. Ce capteur a finalement été utilisé comme plateforme d'expérimentation pour la caméra paracatadioptrique, afin d'étudier différents modèles géométriques. Finalement, cette étude a révélé la nécessité d'utiliser un modèle de caméra non centrale pour obtenir une reconstruction 3D précise aux différentes distances, raison pour laquelle les travaux du [chapitre 4](#) ont été effectués. Une application interactive de reconstruction 3D d'environnements intérieurs a été développée. Elle pourra servir de base pour la reconstruction utilisant d'autres capteurs, qu'ils soient omnidirectionnels ou non. Des extensions ont finalement été proposées pour améliorer les performances du capteur.

Conclusions et perspectives

Les travaux présentés dans cette thèse s'inscrivent dans l'étude et la réalisation d'un système de guidage par l'image pour les personnes. Dans ce contexte, il a été établi qu'un champ de vision large contribuait à améliorer la localisation. Cette thèse s'est donc principalement concentré sur la thématique des caméras à large champ de vision et a permis quelques avancées dans ce domaine.

Contributions

Contributions théoriques : Les chapitres 3 et 4 ont présenté des apports théoriques de la thèse respectivement pour la caméra paracatadioptrique, supposée centrale et pour les caméras catadioptriques non centrales utilisant des miroirs en forme de quadrique :

- Le [chapitre 3](#) a fourni des outils spécifiques pour la caméra paracatadioptrique, essentiellement axés sur la détection robuste des droites, avec application à l'estimation de l'orientation de la caméra et des paramètres de son modèle géométrique.
- Le [chapitre 4](#) a permis l'obtention d'un modèle direct pour les caméras catadioptriques à base de miroirs quadriques. Ce modèle, général, permet la projection des points, des (segments de) droites (par échantillonnage adéquat de points) et la reprojection de texture. Des résultats pratiques ont été proposés pour ces applications et des résultats préliminaires sur l'étalonnage géométrique de caméra non centrale ont été présentés.

Contributions pratiques : Les chapitres 2 et 5 ont présenté les apports plus pratiques et techniques de la thèse pour le développement respectivement d'un capteur d'images et d'un capteur 3D, tous deux omnidirectionnels :

- Le [chapitre 2](#) a présenté un nouveau type de caméra omnidirectionnelle, constituée d'une caméra standard et d'un capteur d'orientation. Elle a été utilisée pour acquérir en temps réel des mosaïques omnidirectionnelles orientées qui sont ensuite affinées par la méthode de l'ajustement de faisceaux.
- Finalement, le [chapitre 5](#) a montré un capteur permettant l'acquisition 3D omnidirectionnelle en cours de développement. Ce capteur utilise une caméra paracatadioptrique et des méthodes de multiscopie pour la modélisation 3D de la géométrie et de la texture d'un environnement d'intérieur de bâtiment. Le modèle de caméra centrale utilisé n'a pas permis une reconstruction 3D suffisamment précise pour l'application visée, ce

qui a conduit aux recherches du [chapitre 4](#), qui n'ont pas encore pu être appliquées à ce capteur.

Perspectives

- Les travaux présentés au [chapitre 3](#) permettent l'estimation de l'attitude d'une caméra paracatadioptrique située dans une scène contenant des droites verticales contrairement à ceux de Demonceaux et Vasseur [16] utilisant l'horizon. Alors que ces derniers travaillent sur l'estimation d'attitude d'un drone en milieu extérieur, où l'horizon est visible, les travaux présentés ici permettent par exemple d'étendre cette estimation pour les scènes d'intérieur.

Les travaux effectués sur l'étalonnage robuste de la caméra paracatadioptrique pourront par exemple être utilisés pour des caméras disposant d'un zoom, afin de faire varier le champ visuel et la résolution. Dans ce cadre, l'étalonnage automatique permet de réestimer dynamiquement les paramètres en fonction du changement de focale.

- Nous pensons que le [chapitre 4](#) est celui qui laisse entrevoir le plus de perspectives, de par le champ très large d'applications possibles. En effet, les caméras catadioptriques non centrales peuvent être très simple à fabriquer, les contraintes strictes d'alignement entre l'optique et le miroir ou l'emploi d'une optique télécentrique parfaite n'étant pas requis. Ces caméras ont connu un intérêt croissant, mais néanmoins freiné par l'inexistence d'un modèle direct.

La preuve de l'implication d'une tangence entre deux quadriques dans le cas d'une valeur propre de multiplicité deux sera établie dans une prochaine publication.

La solution proposée pourra aussi être appliquée dans le cadre de la synthèse d'image pour le lancé de rayons inverse. Dans ce cas, le reflet dans un miroir en forme de quadrique quelconque d'un objet représenté par son modèle 3D à base de facettes triangulaires peut être approximé pour être rendu en temps réel. Ceci permettrait d'étendre la méthode proposée par Roger et Holzschuch dans [206], restreinte à la réflexion sur un miroir sphérique. La solution proposée trouve aussi comme application le calcul des spécularités, qui sont les reflets des sources lumineuses ponctuelles.

Enfin, il est possible d'envisager des applications par exemple pour l'étalonnage de projecteurs vidéo sur des écrans dont la surface est une quadrique, comme le proposent Baar, Raskar et al. dans [207] et [208]. L'emploi d'écrans courbés en forme de quadriques est, en effet, de plus en plus courant dans le contexte de la visualisation immersive.

La détermination des racines des polynômes menant au calcul des quadriques tangentes mériterait une attention particulière, avec l'emploi de méthodes optimisées, telle que celle de Fortune, présentée dans [155].

Les perspectives pour les chapitres 2 et 5 ont déjà été présentées et seules les grandes lignes sont reprises ici :

- Le capteur présenté au [chapitre 2](#) mériterait d'être intégré avec une unité de calcul permettant le traitement temps réel des images. Ainsi, il deviendrait un appareil photo panoramique portable, capable d'acquérir des images en haute dynamique. Les travaux concer-

- nant les descripteurs de points d'intérêt orientés devrait aussi être étendus, notamment pour la localisation par l'image utilisant des points d'intérêt, telle que décrite plus loin.
- Finalement, pour améliorer le capteur présenté au [chapitre 5](#), de nombreuses voies ont été proposées. L'emploi d'un modèle non central pour la caméra permettra une reconstruction plus précise de la géométrie de la scène. Un choix de modèle différent pour la représentation de la scène a aussi été proposé. L'amélioration du capteur peut aussi passer par l'emploi de méthodes actives, utilisant la projection de lumière, structurée ou non, sur la scène afin de faciliter la reconstruction 3D automatique.

Vers la localisation basée image

Les apports de la thèse, plus particulièrement pour l'application de localisation présentée dans l'introduction, sont ici passés en revue et des perspectives sont proposées.

Le [chapitre 3](#) a permis d'étudier la caméra paracatadioptrique. Elle peut être utilisée pour la localisation d'un utilisateur. Son orientation est alors calculée à partir de la projection des droites verticales de la scène. Une application de localisation utilisant ce principe a été proposée dans [33]. Ce chapitre a aussi présenté une modélisation simple de la caméra utilisée dans le capteur omnidirectionnel 3D présenté au [chapitre 5](#). Ce capteur est utilisable pour l'acquisition de modèle 3D de scènes d'intérieur, même si l'acquisition de la géométrie souffre actuellement d'une précision limitée, liée à une mauvaise modélisation de la caméra.

Des représentations omnidirectionnelles sont alors générées à l'intérieur du modèle 3D de la scène en utilisant la méthode de synthèse proposée en Annexe B. Des représentations omnidirectionnelles du même type peuvent être obtenues à partir d'images de la caméra paracatadioptrique, par reprojection d'images, telle que présenté au [chapitre 1](#). L'appariement de ces représentations de l'environnement avec celles obtenues depuis la caméra de l'utilisateur n'a pas été présenté ici. Quelques résultats ont été publiés dans [35], en utilisant des images en projection paracatadioptrique comme base de représentation. Dans cet article, les images de la base de données avaient été acquises avec la caméra paracatadioptrique et les images à localiser étaient fournies par une caméra standard équipée du capteur d'orientation présenté au [chapitre 42](#).

Le large champ visuel, obtenu grâce au mosaïquage avec la caméra présentée au [chapitre 2](#), permet de générer des représentations de l'environnement plus riches, plus à même de permettre une localisation robuste.

Trois ans après le début de cette thèse, d'autres méthodes ont permis de gros progrès pour la localisation et il apparaît que la localisation utilisant l'apparence n'est pas forcément la mieux adaptés au guidage des personnes.

Argyros et al. décrivent dans [4] une méthode de localisation dynamique basée sur le suivi de points qui sont des projections de coins détectées dans les images d'une caméra catadiop-

trique. Grâce à l'emploi de descripteurs pour décrire les points d'intérêt détectés, tels que le SIFT, ces points peuvent être utilisés pour la localisation absolue, ce que font Silpa-Anan et al. dans [209]. De nombreux travaux utilisant ce types d'approches ont été présentés par différents chercheurs et ont montré des résultats très convaincants.

Les différents travaux réalisés durant la thèse trouvent aussi une utilité dans un tel cadre :

L'étalonnage et la modélisation des caméras centrales et non centrales présentés aux chapitres 3 et 4 permettent par exemple le calcul de la projection des points d'intérêt dans les images ainsi que la triangulation de leurs positions dans la scène.

La caméra présentée au [chapitre 2](#) permet d'obtenir des points d'intérêt dont les rayons de projection sont orientés dans la scène et répartis tout autour de la caméra. Il est donc nécessaire d'estimer seulement la position de la caméra, ce qui simplifie grandement le problème de l'estimation de pose et la rend plus robuste, notamment aux occultations. Avec une telle approche, seule la position des points d'intérêt doit être affinée pendant l'ajustement de faisceaux car la synthèse de l'image mosaïquée avec post-traitements permettant la comparaison avec des images de la base de données n'est pas nécessaire.

Dans le même contexte, le capteur 3D présenté au [chapitre 5](#) n'aurait plus à fournir un modèle permettant la synthèse d'images réalistes. Il serait alors seulement nécessaire d'acquérir les positions 3D des points d'intérêt et de les stocker dans la base de données avec les descripteurs correspondants, ce qui est nettement plus facile.

Un des aspects pour lesquels la localisation utilisant l'apparence avait été préférée au début de la thèse est que les intérieurs de bâtiments sont souvent constitués de grandes zones non texturées (murs, sols, plafonds). Ces zones sont difficilement prises en compte dans des approches utilisant uniquement des points d'intérêt. Une solution optimale pour la localisation dans ces conditions est peut être une approche hybride de localisation combinant l'apparence, les droites de la scène et les points d'intérêt, ce qui laisse le champ libre à de nombreuses recherches futures

Annexe A

Quelques définitions

Quelques un des termes utilisés dans le document, et dont la compréhension est nécessaire, sont définis ici.

Une *caméra* est un dispositif permettant la projection d'une scène vers une surface sensible sur laquelle se forme l'*image* de la scène. Cette projection s'effectue via des droites, appelées *rayons de projection*. Si les différents rayons de projection perçus par la caméra s'intersectent en un point unique, la caméra est dite *centrale*, l'exemple le plus courant étant la caméra modélisée par un trou d'épingle et un plan. Sinon, la caméra est dite *non centrale*.

La *caustique* d'une caméra est définie comme étant la surface tangente à l'ensemble des rayons de projection de la caméra. Dans le cas d'une caméra centrale, la caustique est réduite à un unique point.

Le terme *catadioptrique* est employé pour une caméra si elle comporte à la fois des éléments optiques (lentilles) et réfléchifs (miroirs). Cette définition n'est pas assez restrictive car nombreuses sont les caméras comportant des éléments réfléchifs ne modifiant pas le champ de vision de la caméra (par exemple, des miroirs sont employés dans les appareils photos numériques pour diminuer l'encombrement des objectifs). Le terme catadioptrique sera donc utilisé ici pour les caméras dans lesquelles le(s) miroir(s) permet(tent) d'accroître le champ de vision.

La *modélisation 3D* ou *reconstruction 3D* est une opération correspondant à la détermination de la géométrie 3D d'une scène ou d'un objet. Elle peut éventuellement être obtenue à une transformation près, à partir d'une ou plusieurs images issues d'une ou plusieurs caméras, l'éclairage pouvant être ou non contrôlé. Les mêmes termes que pour les champs de vision sont employés ici pour les reconstructions 3D. Ainsi, une reconstruction 3D omnidirectionnelle correspond à des données 3D situées tout autour du capteur.

L'*étalonnage géométrique* de caméra, souvent appelé *calibrage*, est l'opération qui consiste à estimer les paramètres d'un modèle de la caméra afin de pouvoir associer un rayon de projection dans l'espace à une position dans l'image (*modèle inverse* de la caméra) ou inversement prédire la projection dans l'image d'un point de la scène (*modèle direct* de la caméra).

Les termes *outliers* et *inliers* sont utilisés dans le cadre des problèmes d'estimation robuste pour nommer respectivement les données aberrantes et les autres.

Annexe B

Synthèse d'images pour des caméras centrales à large champ de vision

Il est parfois nécessaire de générer des images à partir d'un modèle, par exemple pour tester des algorithmes sur des données de synthèse ou pour synthétiser de nouvelles images à partir d'images existantes.

Le rendu d'images par une projection centrale autre que perspective plane ou orthographique n'est pas pris en charge nativement par les cartes accélératrices 3D présentes dans les ordinateurs. Ces dernières calculent les images par projection de primitives graphiques depuis un repère de travail vers l'image très rapidement. Ainsi des facettes triangulaires texturées peuvent être rendues très efficacement mais la projection doit pouvoir être représentée sous forme matricielle pour que le pipeline graphique puisse l'effectuer.

Dès lors, la méthode généralement employée pour rendre les images avec des projections différentes utilise le lancer de rayons [88]. Cette méthode consiste à lancer un rayon dans la scène virtuelle pour chaque pixel de l'image à rendre. Le trajet du rayon est calculé en tenant compte des intersections avec les objets et des éventuelles réflexions, en émettant récursivement des rayons secondaires depuis les positions d'intersections successives. La couleur du pixel à afficher est alors déterminée par les propriétés des matériaux des différents objets rencontrés pendant le trajet, les conditions d'éclairage etc. . . Dans sa version simplifiée, le Raycasting, on ne cherche pas à calculer les rayons secondaires, la couleur est donc déterminée uniquement par le matériau au niveau du point d'impact du rayon. Cette méthode de rendu serait suffisante pour notre application si l'on considère, pour les caméras catadioptriques, que les rayons primaires sont ceux déjà réfléchis par le miroir. Le problème est que le raycasting n'est pas intégré nativement dans les fonctionnalités des cartes 3D.

La méthode proposée pour rendre des images omnidirectionnelles consiste à décomposer la projection de la scène vers l'image à rendre en deux projections successives. La scène est tout d'abord projetée en projection perspective plane sur les six faces d'un cube. Ce cube est ensuite projeté par lancer de rayons vers une image en utilisant le modèle souhaité.

L'intérêt de cette décomposition est que la projection du cube vers l'image finale peut être précalculée et stockée dans une table de correspondances si l'on considère que le cube est rigi-

dement liée à la caméra virtuelle avec laquelle on synthétise l'image. Le rendu des six images à la surface du cube peut ensuite être effectué par la carte graphique, en utilisant la librairie de synthèse d'image OpenGL [210] par exemple. L'image à rendre est alors calculée, pixel par pixel, en allant chercher dans les six images à la surface du cube les valeurs correspondantes.

La figure B.1 montre un exemple de synthèse. La scène, représentée par (1) est tout d'abord rendue en (2) sur les six images du cube dont les faces sont représentées dépliées. Ensuite la projection paracatadioptrique vers l'image panoramique est effectuée en projetant le cube sur le miroir (3) par projection centrale de centre F puis en projetant le résultat sur le plan de l'image panoramique (4) par projection orthographique.

Avec cette approche, il est possible de rendre plusieurs projections différentes depuis le même point de vue en utilisant les mêmes images à la surface du cube. Ainsi, sur l'exemple de la figure B.2, ce sont deux images panoramiques en projection paracatadioptrique qui sont synthétisées à partir des six images planes.

De même, la figure 1 de l'introduction, présente des images en projection cylindrique et sphérique obtenues avec cette méthode.

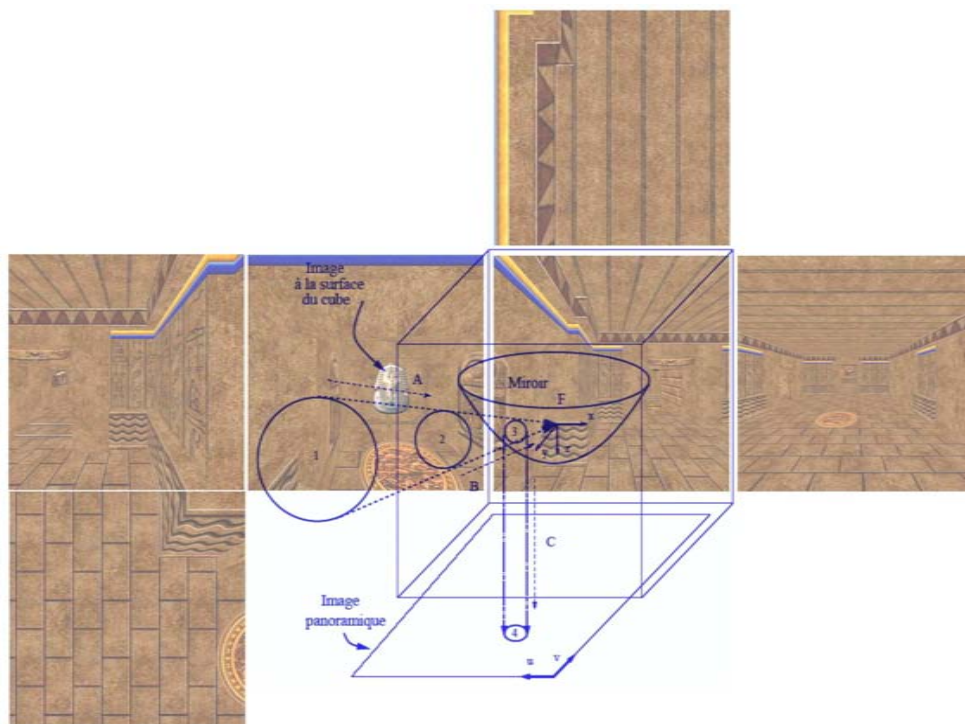


FIG. B.1 – Exemple de synthèse d'image optimisée. La scène (1) est projetée (A) sur les six faces d'un cube en (2) avant d'être projetée (B) vers l'image panoramique en (3) puis (4) par projection (C) orthographique.

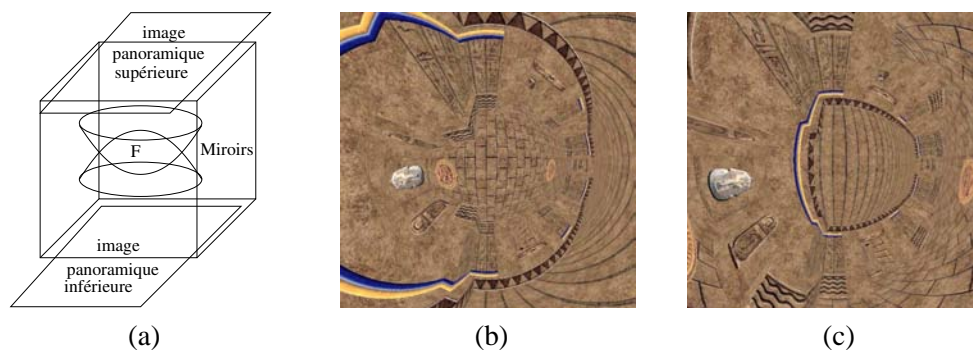


FIG. B.2 – (a) Projection du cube vers deux images panoramiques en projection paracatadioptrique. (b) L’image panoramique inférieure obtenue. (c) L’image panoramique supérieure.

Gestion de l’antialiasage

Le rendu en deux temps effectué par la méthode proposée pose un problème de rééchantillonnage pouvant faire apparaître un phénomène de *moiré* ou *aliasage*. Les cartes 3D possèdent la faculté d’éviter le moiré lors du rendu en projection perspective en utilisant des fonctionnalités de filtrage. Il est donc intéressant de pouvoir conserver cet avantage pour le rendu de nos images.

Le moiré apparaît lors de la projection du cube sur l’image à rendre car les images à la surface du cube définissent des signaux 2D qui possèdent des fréquences qui peuvent être supérieures à la demi fréquence à laquelle les images sont rééchantillonnées lorsque le cube est projeté vers l’image à rendre. Ainsi, sur la figure B.3, on peut voir en (a), les rayons de projection correspondant à deux pixels voisins dans l’image à rendre. Ces deux rayons intersectent l’image à la surface du cube en deux pixels noirs. Le théorème de Shannon garantissant un échantillonnage correct n’est alors plus respecté, car le pixel blanc du milieu n’est pas échantillonné. La solution d’un filtrage coupant les hautes fréquences appliqué sur toute l’image n’est pas souhaitable car ceci génère alors un flou dans certaines zones disposant d’une meilleure résolution.

Une solution consiste à utiliser un *cône de projection* plutôt qu’un rayon, comme proposé par Hasenfratz [211] et par Amanatides [212] dans le cadre de la synthèse d’images par Raytracing.

Nous choisissons ici une ouverture de ce cône en fonction de l’angle maximal entre le rayon de projection pour le pixel considéré et les rayons de projection correspondant à ses pixels voisins ((b) sur la figure B.3).

L’intersection du cône avec l’un des plans image du cube forme une ellipse. En analysant la surface occupée par chaque pixel de l’image à la surface du cube dans cette ellipse, il est possible de déterminer l’influence de chacun d’eux.

Ainsi, au lieu de calculer une table de correspondances comportant seulement des positions dans les images du cube, la table calculée contient des ensembles de positions (tous les pixels

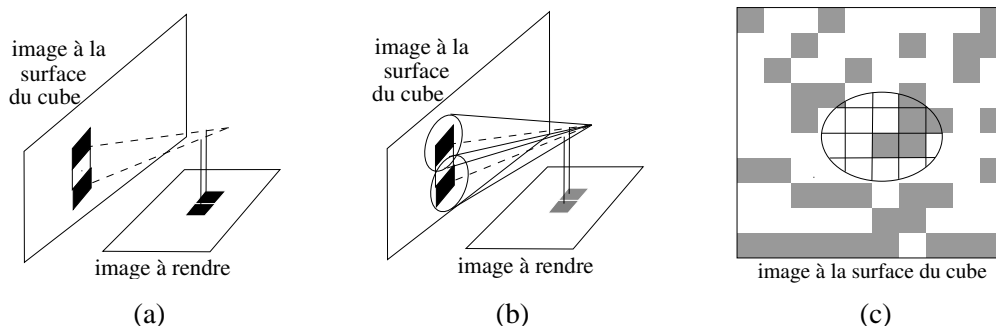


FIG. B.3 – (a) Mauvais rééchantillonnage de l'image à la surface du cube. Le pixel central blanc n'est pas pris en compte et il y a apparition de moiré dans l'image à rendre. (b) Bon rééchantillonnage de l'image grâce à l'utilisation des cônes de projection. Les deux pixels de l'image à rendre sont calculés à partir d'un ensemble de pixels couvert par l'intersection des cônes avec l'image à la surface du cube. (c) Gros plan sur une image à la surface du cube et les vingt pixels couverts par une ellipse.

ayant une surface non nulle sous l'ellipse pour un rayon de projection donné) assorties de facteurs de pondération. Dans l'exemple (c) de la figure B.3, c'est une liste de vingt positions et poids qu'il faut stocker pour le cône de projection générant l'ellipse montrée.

Dans les zones où l'image à la surface du cube est rééchantillonnée finement, la taille de l'ellipse peut être inférieure à un pixel. Dans ce cas, la valeur du pixel sur lequel se trouve l'ellipse est directement utilisée.

Certains rayons de projection génèrent des ellipses qui dépassent des faces du cube. Ceci arrive fréquemment lorsque l'on considère les rayons qui intersectent le cube près de ses bords. Le cône intersecte alors plusieurs faces distinctes. Pour éviter d'avoir à gérer ce cas plus complexe, il suffit de rendre des images à la surface du cube avec un peu plus de 90° de champ visuel. Ces images dépassant du cube, il est alors possible de calculer l'intersection du cône avec une seule face.

Ce traitement permet de gérer le flou localement et permet d'éviter le moiré, en propageant le filtrage déjà effectué par la carte 3D pour le rendu des images en projection perspective. La figure B.4 montre le résultat de ce traitement lors de la projection vers une image paracata-dioptrique d'un cube dont l'une des faces est une grille de pixels noirs et blancs. Il apparaît clairement en (b) que le traitement a supprimé le moiré.

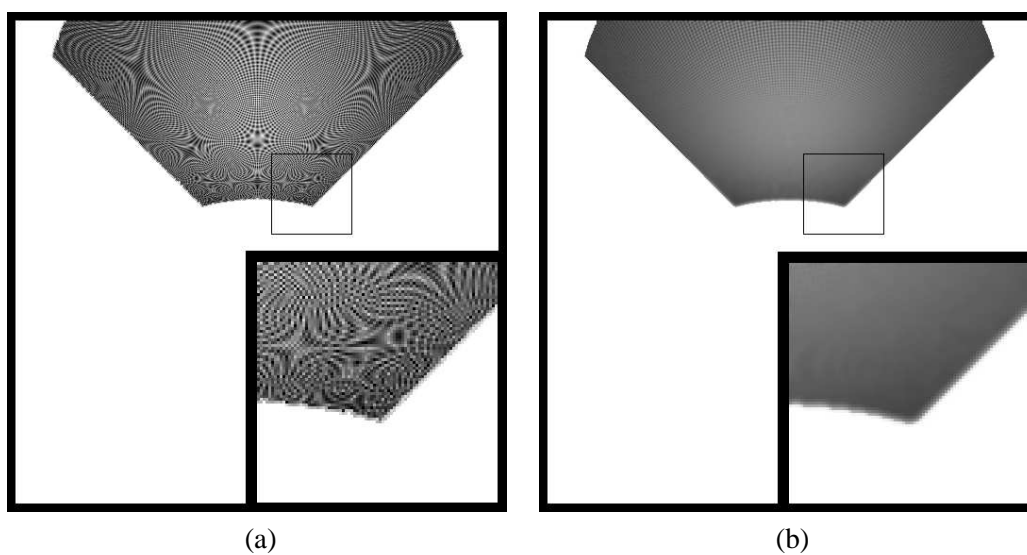


FIG. B.4 – Reprojection d'une grille de pixels sur une face du cube avec grossissement de facteur trois sur la zone de basse résolution qui est encadrée : (a) sans gestion de l'antialiasage, (b) avec la méthode proposée.

Annexe C

Calcul du quaternion correspondant à une matrice de rotation

Le listing suivant est un programme MATLAB permettant d'obtenir le quaternion Q à partir d'une matrice de rotation de dimension 3*3.

```
function [Q]=MatrixToQuaternion(mat)

T=1 + mat(1,1) + mat(2,2) + mat(3,3);
S = sqrt(T) * 2;
X = ( mat(3,2) - mat(2,3) ) / S;
Y = ( mat(1,3) -mat(3,1) ) / S;
Z = ( mat(2,1) - mat(1,2) ) / S;
W = 0.25 * S;
if ( T > 0.00000001 )
    S = sqrt(T) * 2;
    X = ( mat(3,2) - mat(2,3) ) / S;
    Y = ( mat(1,3) - mat(3,1) ) / S;
    Z = ( mat(2,1) - mat(1,2) ) / S;
    W = 0.25 * S;
else
    if ( mat(1,1) > mat(2,2) && mat(1,1) > mat(3,3) )
        S = sqrt( 1.0 + mat(1,1) - mat(2,2) - mat(3,3) ) * 2;
        X = 0.25 * S;
        Y = (mat(2,1) + mat(1,2) ) / S;
        Z = (mat(1,3) + mat(3,1) ) / S;
        W = (mat(3,2) - mat(2,3) ) / S;
    else if ( mat(2,2) > mat(3,3) )
        S = sqrt( 1.0 + mat(2,2) - mat(1,1) - mat(3,3) ) * 2;
        X = (mat(2,1) + mat(1,2) ) / S;
        Y = 0.25 * S;
```

```
Z = (mat(3,2) + mat(2,3) ) / S;  
W = (mat(1,3) - mat(3,1) ) / S;  
else  
  S = sqrt( 1.0 + mat(3,3) - mat(1,1) - mat(2,2) ) * 2;  
  X = (mat(1,3) + mat(3,1) ) / S;  
  Y = (mat(3,2) + mat(2,3) ) / S;  
  Z = 0.25 * S;  
  W = (mat(2,1) - mat(1,2) ) / S;  
end;  
end;  
end;  
Q=[X Y Z W];
```

Annexe D

Simplification de l'algorithme d'étalonnage proposé par Barreto

Sous l'hypothèse d'une caméra avec des pixels carrés, ou du moins de géométrie connue, la seconde étape de l'algorithme de Barreto [139] p.90 peut facilement être simplifiée. Nous adoptons ici les notations de Barreto. Soit (Ω_i, Ω_j) une paire de cercles dans le plan image. L'étape 2 de l'algorithme consiste à calculer la droite $\mu_{ij} = \mathbf{f}_{ij} \times \mathbf{g}_{ij}$ partagée par les deux points fins \mathbf{f}_{ij} et \mathbf{g}_{ij} en lesquels les cercles Ω_i et Ω_j s'intersectent. Comme μ_{ij} contient le point principal, à partir de $n \geq 2$ paires de cercles, le point principal est défini comme l'intersection des différentes μ .

Au lieu de calculer explicitement les quatre points communs à Ω_i et Ω_j (elles passent aussi par les points cycliques) et d'en déduire \mathbf{f}_{ij} et \mathbf{g}_{ij} , il est possible d'opérer de la manière suivante : Considérons que tous les Ω sont normalisés tel que $\Omega_{(1:2,1:2)} = \mathbf{I}_{2 \times 2}$. Il est facile de voir que :

$$\Delta_{ij} = \Omega_i - \Omega_j = \begin{bmatrix} 0 & 0 & d_{ij}^1 \\ 0 & 0 & d_{ij}^2 \\ d_{ij}^1 & d_{ij}^2 & 2d_{ij}^3 \end{bmatrix} = \mathbf{l}_\infty \mathbf{d}_{ij}^T + \mathbf{d}_{ij} \mathbf{l}_\infty^T,$$

où $\mathbf{l}_\infty = (0 \ 0 \ 1)^T$, $\mathbf{d}_{ij} = (d_{ij}^1 \ d_{ij}^2 \ d_{ij}^3)^T$. Donc Δ_{ij} est une conique dégénérée de rang 2 constituée de deux droites passant nécessairement par les quatre points communs de Ω_i et Ω_j (voir la section 4.3 pour plus de détails). Une de ces droites est clairement la ligne à l'infini \mathbf{l}_∞ et l'autre est $\mu_{ij} = \mathbf{d}_{ij}$. Ceci permet d'obtenir un algorithme de Barreto modifié, dans lequel le calcul μ_{ij} depuis Ω_i et Ω_j est direct (les étapes 3 à 7 sont par contre inchangées).

Cette remarque n'est pas cruciale au niveau des considérations numériques, mais elle mérite d'être mentionnée, car en plus de l'approfondissement géométrique, les calculs sont bien plus simples et stables.

Annexe E

Détails sur les calculs du chapitre 3 et exemples

Code MAPLE pour obtenir le polynôme pour la réflexion sur la sphère

Par rapport aux notation du chapitre 3, F_z correspond dans la suite à x_C et G_x, G_y, G_z correspondent respectivement à x_P, y_P et z_P .

```
with( LinearAlgebra ):
assume( Fz, real ):
assume( Gx, real ):
assume( Gy, real ):
assume( Gz, real ):
%position de la caméra
Fe := <0,0,Fz,1>;
%position d'un point fini
Ge := <Gx,Gy,Gz,1>;
%ou position d'un point à l'infini
Ge := <Gx,Gy,Gz,0>;
%ellipsoïde ou paraboloid de révolution correspondant
FG := MatrixAdd(OuterProductMatrix(Fe,Ge),
                 OuterProductMatrix(Ge,Fe))
Q := < <1,0,0,0> | <0,1,0,0> | <0,0,1,0> | <0,0,0,0> >
assume( xi, real ):
Ellipsoid := MatrixAdd( FG, ScalarMultiply(Q,xi))
%La sphère sur laquelle à lieu la réflexion
S:= ScalarMultiply(MatrixInverse(<<1,0,0,0> | <0,1,0,0> |
                                <0,0,1,0> | <0,0,0,-1> >),1)
%Calcul des valeurs propres lambda pour lequel determinant=0
lambdaI4 := ScalarMultiply( < <1,0,0,0> | <0,1,0,0> |
                             <0,0,1,0> | <0,0,0,1> >,lambda)
```

```

mat := MatrixAdd(MatrixMatrixMultiply(S, Ellipsoid), -lambdaI4)
s := Determinant( mat )
%Les valeurs propres doubles sont recherchées,
%elles correspondent à un discriminant nul
dis := expand(discrim(s, lambda)/xi^2)
%Récupère les coefficients du polynôme
scoef=coeffs(dis, xi, 'tts')
tts

```

Polynôme pour un point à coordonnées finies

$$c_0 = 4 * Fz^4 * (Gx^2 + Gy^2)^2 * (Gx^2 + Gy^2 + Gz^2 - 1) * (Fz - 1) * (Fz + 1) * (-Fz^2 + Fz^2 * Gx^2 - Gx^2 + Fz^2 * Gy^2 - Gy^2 + 2 * Fz * Gz - Gz^2)^2$$

$$c_1 = -4 * Fz^4 * (Gx^2 + Gy^2)^2 * (-14 * Fz^2 * Gz^2 - Fz^4 + 8 * Fz^5 * Gx^2 * Gy^2 * Gz - 2 * Gx^2 * Gz^2 + Fz^3 * Gy^4 * Gz + 4 * Fz^5 * Gy^4 * Gz + Gx^4 * Fz^3 * Gz + 4 * Gx^4 * Fz^5 * Gz - 5 * Gy^4 * Fz * Gz + 22 * Fz^2 * Gx^2 * Gy^2 + 11 * Gx^4 * Fz^2 + 11 * Fz^2 * Gy^4 - 10 * Fz^4 * Gy^4 - 2 * Gx^2 * Gy^2 - 10 * Fz^4 * Gx^4 - Gy^4 + 8 * Fz * Gy^2 * Gz - 10 * Fz^2 * Gy^2 + 8 * Gz^3 * Fz + 8 * Fz^3 * Gz - 14 * Fz^3 * Gz^3 + 12 * Gz^4 * Fz^2 + 12 * Fz^4 * Gz^2 + 11 * Fz^4 * Gx^2 - 10 * Fz^2 * Gx^2 + 11 * Fz^4 * Gy^2 - Gz^4 - 5 * Gz^5 * Fz - 5 * Fz^5 * Gz - 12 * Fz^3 * Gx^2 * Gz + Fz^5 * Gx^2 * Gz - 12 * Fz^3 * Gy^2 * Gz + Fz^5 * Gy^2 * Gz + 8 * Gx^2 * Fz * Gz + 23 * Gz^2 * Fz^2 * Gx^2 + 5 * Gz^3 * Fz^3 * Gx^2 - 18 * Gz^2 * Fz^4 * Gx^2 + 4 * Gz^3 * Fz^5 * Gx^2 + 23 * Fz^2 * Gy^2 * Gz^2 - 18 * Fz^4 * Gy^2 * Gz^2 + 4 * Fz^5 * Gz^3 - 2 * Gy^2 * Gz^2 + 4 * Gz^5 * Fz^3 + 5 * Fz^3 * Gy^2 * Gz^3 - 10 * Gx^2 * Gz^3 * Fz - 10 * Gz^3 * Gy^2 * Fz - 8 * Fz^4 * Gz^4 + 4 * Fz^5 * Gy^2 * Gz^3 - 5 * Gx^4 * Fz * Gz - Gx^4 - 20 * Fz^4 * Gx^2 * Gy^2 + 2 * Fz^3 * Gx^2 * Gy^2 * Gz - 10 * Gx^2 * Fz * Gy^2 * Gz)$$

$$c_2 = -Fz^4 * (Gx^2 + Gy^2)^2 * (-54 * Fz^2 * Gz^2 - Fz^4 - 2 * Gx^2 * Gz^2 + 40 * Fz^2 * Gx^2 * Gy^2 + 20 * Gx^4 * Fz^2 + 20 * Fz^2 * Gy^4 + 8 * Fz^4 * Gy^4 - 2 * Gx^2 * Gy^2 + 8 * Fz^4 * Gx^4 - Gy^4 + 20 * Fz * Gy^2 * Gz - 50 * Fz^2 * Gy^2 + 20 * Gz^3 * Fz + 20 * Fz^3 * Gz - 32 * Fz^3 * Gz^3 + 32 * Gz^4 * Fz^2 + 32 * Fz^4 * Gz^2 + 20 * Fz^4 * Gx^2 - 50 * Fz^2 * Gx^2 + 20 * Fz^4 * Gy^2 - Gz^4 - 32 * Fz^3 * Gx^2 * Gz - 32 * Fz^3 * Gy^2 * Gz + 20 * Gx^2 * Fz * Gz + 52 * Gz^2 * Fz^2 * Gx^2 - 8 * Gz^2 * Fz^4 * Gx^2 + 52 * Fz^2 * Gy^2 * Gz^2 - 8 * Fz^4 * Gy^2 * Gz^2 - 2 * Gy^2 * Gz^2 - 16 * Fz^4 * Gz^4 - Gx^4 + 16 * Fz^4 * Gx^2 * Gy^2)$$

$$c_3 = 4 * Fz^5 * (Gx^2 + Gy^2)^2 * (-Gz * Gx^2 + 6 * Gx^2 * Fz + 4 * Gx^2 * Fz^2 * Gz + 4 * Fz^2 * Gy^2 * Gz - Fz^2 * Gz + 6 * Fz * Gz^2 - Gz^3 + 4 * Fz^2 * Gz^3 - Gy^2 * Gz + 6 * Fz * Gy^2)$$

$$c_4 = 4 * Fz^6 * (Gx^2 + Gy^2)^2 * (Gx^2 + Gz^2 + Gy^2)$$

Polynôme pour un point à l'infini

$$c_0 = 4 * Fz^4 * (Gx^2 + Gy^2)^2 * (Gx^2 + Gz^2 + Gy^2) * (Fz - 1) * (Fz + 1) * (-Gx^2 + Fz^2 * Gx^2 - Gy^2 + Fz^2 * Gy^2 - Gz^2)^2$$

$$c_1 = -4 * Fz^5 * Gz * (Gx^2 + Gy^2)^2 * (Gx^2 + Gz^2 + Gy^2) * (Fz^2 * Gx^2 - 5 * Gx^2 + 4 * Fz^4 * Gx^2 - 5 * Gz^2 + 4 * Fz^2 * Gz^2 + Fz^2 * Gy^2 + 4 * Fz^4 * Gy^2 - 5 * Gy^2)$$

$$c_2 = -Fz^4 * (Gx^2 + Gy^2)^2 * (Gx^2 + Gz^2 + Gy^2) * (8 * Fz^4 * Gx^2 - Gx^2 + 20 * Fz^2 * Gx^2 - Gz^2 + 8 * Fz^4 * Gy^2 + 20 * Fz^2 * Gy^2 - Gy^2 + 32 * Fz^2 * Gz^2 - 16 * Fz^4 * Gz^2)$$

$$c_3 = 4 * Fz^5 * Gz * (Gx^2 + Gy^2)^2 * (Gx^2 + Gz^2 + Gy^2) * (2 * Fz - 1) * (2 * Fz + 1)$$

$$c_4 = 4 * Fz^6 * (Gx^2 + Gy^2)^2 * (Gx^2 + Gz^2 + Gy^2)$$

Précisions des calculs numériques

Les valeurs flottantes en double précision

Le standard IEEE 754 fournit un codage des valeurs flottantes en double précision. Les valeurs numériques sont codées sur 64 bits avec la répartition suivante :

- 1 bit pour le signe S (0-1)
- 11 bits pour l'exposant E (0-2047)
- 52 bits pour la mantisse M (0-4.503.599.627.370.495)

La valeur numérique n correspondant à un triplet S , E et M est la suivante :

$$n = (-1)^S (1 + M) 2^{E-1023} \quad (\text{E.1})$$

Le codage approché des valeurs fait apparaître des problèmes de précision lors d'opérations sur des valeurs numériques dont l'ordre de grandeur est différent. Considérons par exemple le problème d'*absorption* : Soit $n_1 = 1e^{16}$ et $n_2 = n_1 + 1$. Comme le plus grand entier que l'on peut représenter exactement par une valeur flottante en double précision est $2^{53} - 2 = 9.007.199.254.740.992$, n_1 et n_2 sont codés de manière approchée et de la même façon. Ainsi l'incrément correspondant à n_2 n'a aucune influence. Ceci n'a pas forcément une grosse répercussion si l'on considère que 1 est négligeable devant 1^{16} . Néanmoins, lors d'opérations successives, ceci peut conduire à des erreurs bien plus importantes. Considérons, par exemple, que l'on souhaite ajouter 1^{16} fois la valeur 1 à n_1 . Si l'on exécute l'opération $n_{i+1} = n_i + 1$ en partant de n_1 , le résultat obtenu sera n_1 , et l'erreur de 100%. A l'opposé, si les incréments sont stockés dans une valeur séparée qui est ensuite ajoutée à n_1 , l'erreur sera très faible.

Ce type de problème est traité, par exemple, dans certaines bibliothèques d'optimisation telle que MINPACK. Lorsqu'un calcul d'une somme de carré de résidus est souhaité, les résidus doivent tout d'abord être triés. Les sommes de sous ensembles de résidus ayant le même ordre de grandeur sont ensuite effectuées, suivies finalement de la somme globale. Ceci évite qu'un très grand nombre de résidus relativement petits soient "noyés" devant un très faible nombre de résidus relativement grands.

Le codage des coefficients des polynômes

Lors du calcul des coefficients des polynômes dont on cherche les racines ξ , il est souhaitable d'avoir la meilleure précision possible. Comme les expressions des coefficients sont relativement longues, le nombre de calculs successifs à effectuer est important. Pour éviter les erreurs de précision telles que citées précédemment, il serait souhaitable de décomposer les opérations.

Annexe F

Anaglyphe sphérique



FIG. F.1 – Anaglyphe sphérique, à visionner avec des lunettes avec filtre rouge sur l'œil gauche et cyan sur l'œil droit, montrant l'auteur de ce document en activité dans son environnement de travail.

Annexe G

Articles sur la localisation basée image

REALTIME LOCALIZATION OF A CENTRAL CATADIOPTRIC CAMERA USING VERTICAL LINES

Bertrand Vandepoortaele, Michel Cattoen, Philippe Marthon

IRIT & LEN7

Enseeiht, 2 rue Camichel, 31071 Toulouse, France

bvdp@enseeiht.fr, cattoen@enseeiht.fr, marthon@enseeiht.fr

Keywords: Omnidirectional vision, localization, orientation, lines.

Abstract: Catadioptric sensors are used in mobile robot localization because of their panoramic field of view. However most of the existing systems require a constant orientation of the camera and a planar motion, and thus the localization cannot be achieved in general for persons handling a camera. In this paper, we use the images of the vertical lines of indoor environment to localize in realtime the central catadioptric camera orientation and the 2D position. The pose detection is done in two steps. First, a two axes absolute rotation is computed to bring the vertical line images in vertical position on the viewing sphere. Then the 2D pose is estimated using a 2D map of the site.

1 INTRODUCTION

Our goal is to localize in realtime a central catadioptric camera held by a person inside a known building. Benosman and Kang gave in (Benosman 2001) a detailed description of these cameras. They are able to acquire instantaneously some panoramic images and have a single viewpoint. They are frequently used in robotic applications where the provided 360° field of view is useful for image based localization and 3D reconstruction. Image based localization methods can be divided in two categories:

A: methods requiring *a priori* knowledge about the place where to localize the robot (for example a database of appearance images corresponding to different locations) (Padjla 2001). These methods require the acquisition of many images and constrain the orientation of the sensor. Moreover, the robust realtime matching of occluded images is a relatively complex problem, even if the panoramic field of view make it easier than in the perspective case.

B: methods using Simultaneous Localization And Mapping (Shakernia 2003). They do not require an acquisition of the database before the localization. Invariant points from the scene are often detected in many images acquired at different locations using Scale Invariant Feature Transform, KLT or Harris detector, and then Structure From Motion is performed in order to detect the position of the camera in the re-

constructed scene.

We propose a new method to achieve the localization of a central catadioptric camera using the vertical lines images. We use this method to localize persons. In this application, the accuracy is not very important (0.3m is sufficient) but the sensor can be held in different orientations and many occlusions can occur.

First, the vertical lines provide information about the absolute orientation of the sensor around two axes thanks to the common vanishing points. Second, they can be used for localization using a simple 2D map because they are invariant to translations about the vertical axis. Vertical lines are numerous inside buildings and difficult to occlude completely. Thus, vertical lines based method particularly fits our application.

Our method allows a realtime matching with varying orientation around every axes. We can use *a priori* knowledge consisting in a 2D map which is easy to acquire. Thus, the camera to localize do not have to acquire images continuously from a known starting position. Moreover, vertical lines images are well detected in varying poses and this compensates the fact that they can be less numerous than corners in the image.

In this paper, we first give a method to retrieve the two axes orientation using the vertical lines images acquired with a calibrated camera. Then we show how to determine the 2D pose using a simple 2D map of the scene and present some experimental results.

2 THE VERTICAL LINES IMAGES DETECTION

The catadioptric camera is calibrated using the Geyer & Daniilidis method (Geyer 1999) in order to know the correspondence between the viewing sphere and the image. The detection of a central catadioptric line image dl_i can be accomplished by finding the orientation of the plane π_i containing dl_i (the line image dl_i lifted on the viewing sphere) and the viewpoint F as shown on fig. 1. Ying & Hu (Ying 2004) and Vasseur & Mouaddib (Vasseur 2004) proposed robust methods to detect the line images using a two parameters Hough transform. For our experiments, we use our own realtime line image detector which is based on robust least squares fitting of planes on the image points projected on the mirror surface. It provides the same kind of results than the Hough based methods but with increased speed.

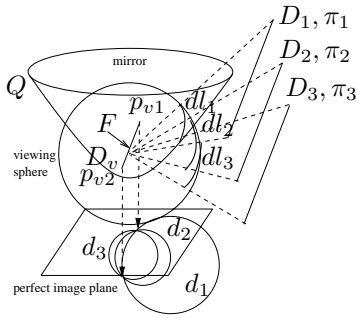


Figure 1: Central catadioptric projection of parallel lines to portions of great circles on the viewing sphere and the paracatadioptric projection on the image.

The parallel lines can be detected from the image by finding the two vanishing points located at the intersection of parallel lines images which are conics (Geyer 2001). In this case, the detection of vanishing points is difficult because the conics parameters are not accurately estimated from small conic sections on noisy data and thus parallel lines images do not intersect precisely in the same points (see fig 6.g for an example).

We propose a method to gather the potentially parallel lines in space based on the detection of the line D_v which is common to a set of planes π_i defined by parallel lines D_i in space. D_v defines a bundle of planes. In practice, the detected planes π_i do not intersect exactly in D_v . Nevertheless, our method allows the gathering from a criterion based on an angular measurement.

The intersection of two planes π_i and π_j respectively with normals n_i and n_j is a line in the direction $h(\pi_i, \pi_j)$ passing through F . We have chosen to use $g(\pi_i, \pi_j)$, a normalized vector in the top hemisphere

defined by: $g(\pi_i, \pi_j) = \frac{n_i \wedge n_j}{\|n_i \wedge n_j\|} \cdot \text{sign}((n_i \wedge n_j)_z)$.

We give a criterion which measures how close to a single line three planes π_i , π_j and π_k intersect. It uses the angular measurement $\omega(\pi_i, \pi_j, \pi_k)$ between $g(\pi_i, \pi_j)$ and $g(\pi_i, \pi_k)$: $\omega(\pi_i, \pi_j, \pi_k) = \text{acos}(g(\pi_i, \pi_j) \cdot g(\pi_i, \pi_k))$.

To deal with planes which have a normal near the axis z , a "close" intersection between three planes π_i, π_j, π_k is detected in the following two cases: $\omega(\pi_i, \pi_j, \pi_k) < \omega_{thres}$ or $\omega(\pi_i, \pi_j, \pi_k) > \pi - \omega_{thres}$. The parameter ω_{thres} is an angular threshold and has to be set in accordance with the accuracy of the sensor. On the figure 2, a small angle between $g(\pi_1, \pi_2)$, $g(\pi_1, \pi_3)$ and $g(\pi_1, \pi_4)$ indicates that π_1, π_2, π_3 and π_4 almost define a bundle of planes so we can conclude that D_1, D_2, D_3 and D_4 are potentially parallel in space.

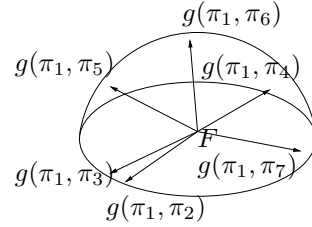


Figure 2: Angles between planes intersections.

A score is given to each gathered solution by summing the scores sc_i corresponding to planes π_i inside the bundle. The score sc_i is computed for each line image by counting the number of corresponding pixels. So the gathered solution having the best score fits the greatest number of line points and thus corresponds to the main orientation inside the scene.

This method is robust to erroneous detection of vertical lines because the normal of such lines do not fit inside the threshold and thus are not taken in account. Once the planes have been gathered, a more precise intersection is computed using iterative Levenberg Marquardt (LM) algorithm. Let α and β be respectively the azimuth and elevation of the line $P_{(\alpha, \beta)}$ corresponding to the best intersection. Its corresponding 3D vector coordinates are $P_{3D(\alpha, \beta)} = [P_x, P_y, P_z]$ and $P_x = (\cos(\alpha) \cdot \cos(\beta))$, $P_y = (\sin(\alpha) \cdot \cos(\beta))$ and $P_z = \sin(\beta)$. The intersection of two random planes from the gathered set is used as an initialization value. The different planes are weighted by their scores sc_i and $P_{3D(\alpha, \beta)}$ is computed to be the most possibly perpendicular line to all the weighted normalized normals of planes n_i by minimizing the following criteria:

$$C(\alpha, \beta) = \sum sc_i \left| \text{acos}(P_{3D(\alpha, \beta)} \cdot n_i) - \frac{\pi}{2} \right|^2 \quad (1)$$

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{(\alpha, \beta)} C(\alpha, \beta) \quad (2)$$

As the vertical lines are often the most numerous inside buildings, this method is used to detect their relative orientation and thus to detect the sensor's orientation around two axes. To increase the robustness of the vertical line detection, we can restrict the potential orientations by asking the user to hold the camera in an interval of angular values ($\pm 45^\circ$). It is thus easy to discard the candidate orientations that cannot be vertical. The two axes orientation of the sensor is obtained by computing the 3D rotation $R(\hat{\alpha}, \hat{\beta})$ that brings $P_{3D(\alpha, \beta)}$ to a vertical line.

3 THE 2D POSE DETECTION

Once the rotation $R(\hat{\alpha}, \hat{\beta})$ has been detected, the localization can be achieved in 2D. The altitude is not estimated as it can vary between different users and it is not an useful data in our application where only an approximate localization of the camera inside the 2D map is needed. For buildings with multiple floors, a single 2D map can be created by lifting the different floors maps to different locations on the same plane. A 2D map contains the positions of vertical lines (x_i, y_i) and the occlusive segments joining these points.

In the noiseless and non degenerate cases (no more than two map points and camera position lie on the same line), 3 points are sufficient to localize the camera. However, due to the limited accuracy of the vertical line detector, more points should be used to achieve an accurate localization.

The figure 3 shows an example of a simple 2D map made of 8 points and 8 occlusive segments.

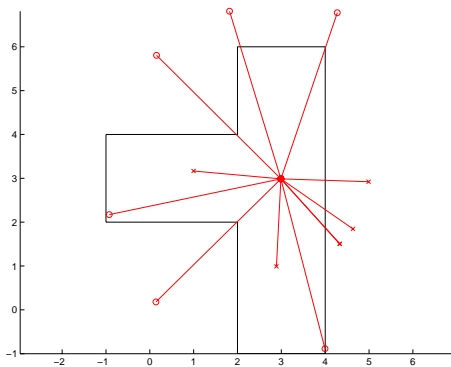


Figure 3: The 2D map (in black), inlier lines images (long red lines with circles) and outliers (short red lines with cross) after localization (the sensor is located at the intersection of all the red lines). The scale is in meter unit.

Each plane π_i corresponding to a detected vertical

line is defined by an value γ_i . γ_i is the angle around the vertical axis corresponding to the intersection of the plane π_i with the horizontal plane. The plane π_i is not sufficient to know on which side of the viewing sphere lie a vertical line D_i . Let p_i be the image points corresponding to the line image d_i . These points are lifted to the viewing sphere in P_i and then rotated to $RP_i = R(\hat{\alpha}, \hat{\beta}).P_i$ in order to align the viewing sphere with the vertical. As the p_i correspond to a vertical line of the scene, the RP_i lie on the same meridian on the viewing sphere. We use the center of mass of RP_i to know on which side of the sphere the line is located and thus can compute the corresponding angular value γ_i in the range $[0, 2.\pi]$. If the RP_i lie on the two sides of the sphere, then we detect a line on each side of the camera. A descriptor of a location is defined by a vector of different γ_i .

Let us first consider that the correspondences between every γ_i and 2D scene point i (corresponding to a vertical line located in (x_i, y_i)) are known.

Let the pose be defined by three parameters. x_c and y_c are the 2D position of the camera in the map and γ_c is the rotation.

Let γ_a be the angular direction corresponding to the point i in the map viewed from the pose (x_c, y_c, γ_c) :

$$\gamma_a = \text{atan2}(y_i - y_c, x_i - x_c) - \gamma_c \quad (3)$$

The correct pose best fits the different angular values γ_a for every γ_i . The angular error has to be expressed in the interval $[-\pi, \pi]$ in order to be near zero for small deviations in the two directions. Finally $E(i, x_c, y_c, \gamma_c)$ corresponds for each point i to the squared angular difference:

$$E(i, x_c, y_c, \gamma_c) = \left| \gamma_i - \gamma_a \right|_{[-\pi, \pi]}^2 \quad (4)$$

The pose is then estimated using Levenberg Marquardt to minimize the following criteria:

$$(\hat{x}_c, \hat{y}_c, \hat{\gamma}_c) = \arg \min_{(x_c, y_c, \gamma_c)} \sum V_i \quad sc_i \quad E(i, x_c, y_c, \gamma_c) \quad (5)$$

where V_i is equal to 1 if the point i is not occluded by segments of the map from (x_c, y_c) and 0 else. sc_i is the score corresponding to the plane π_i as described in the previous section.

In practice, all the V_i are equal to 1 if the correspondences are all correct because only the non occluded lines are detected on the image. If we do not deal with the occlusions and noise and use a sufficient number of points (> 4), the convergence is generally obtained for any initialization value.

Let us now consider the more complex problem of detecting inliers and outliers γ_i and finding the correspondences with points from the map. A simple RANSAC scheme would result in long process-

ing time even if only a few of the possible correspondences have to be tested, as at least triplets (3 correspondences) of couples (1 map point and one γ_i) are needed to compute a pose P_R .

In order to reduce the computational complexity, we propose to achieve a Delaunay triangulation of the 2D map (as shown on the fig 4). We then use the three corners of each triangle as triplets of points to match, as they are never occluded (by map segments) from inside the triangle. The correspondences between the triangle corners points i and γ_i have to respect the order constraint. If there are N map points, M values γ_i and P Delaunay triangles, the number of all possible solutions to test is reduced from $(N.M)^3$ to less than $P.M^3$ as there are less than M^3 combinations of correspondences verifying the order constraint.

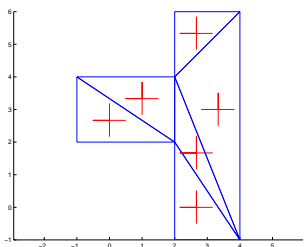


Figure 4: The Delaunay triangulation of the 2D map and the corresponding initialization value (red crosses).

We now show how to discard quickly an invalid triplet of correspondences. Two points i and j and a relative angle $\gamma_i - \gamma_j$ define a circle of potential locations for (x_c, y_c) as shown on fig 5. The third point k should lie on a line related to $(x_i, y_i) - (x_c, y_c)$ by $\gamma_k - \gamma_i$ and to $(x_j, y_j) - (x_c, y_c)$ by $\gamma_j - \gamma_k$. So, if the two extremum positions are considered for the camera position (x_c, y_c) , (x_k, y_k) should lie inside the zone between the two lines L_i and L_j defined by the tangents of the circle at point i (resp. j) and the angles $\gamma_k - \gamma_i$ (resp. $\gamma_j - \gamma_k$). Thus, if (x_k, y_k) is inside the circle or outside the zone delimited by L_i and L_j , the triplet is discarded.

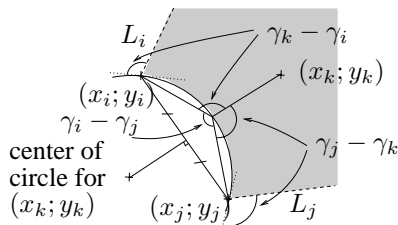


Figure 5: Fast detection of valid triplets (i,j,k).

The non linear optimization of the Equ. 5 is applied on the retained triplets of points in order to es-

timate the pose. The center of the Delaunay triangle is used as initial value (fig. 4). The process is applied to several different triplets and the pose P_R which fits the greatest numbers of γ_i within an angular threshold $thres_\gamma$ and without occlusion due to occluding segments is kept.

Once the best pose P_R has been computed from three points, the optimization is achieved on all the detected inliers using the pose P_R as initialization value.

The figure 3 shows the localization of a pose in the 2D map. The inliers are shown in red long line finishing with a circle and the outliers are shown in red short line finishing with a cross.

Once the camera has been localized, it is easy to track it inside the map if three lines images are visible on the two successive images. Each vertical line projection γ_i is tracked individually to keep the correspondence information and the new pose is estimated using the last pose as initialization value. Some correspondences are removed and others are added during the process.

4 RESULTS

In our experiments, we use 568*426 pixels images, a paracatadioptric sensor [4] and a Pentium IV based computer running at 1.8 Ghz. The figure 1 shows parallel lines imaged by this sensor. In a perfect image plane, with zero skew and square pixels, these projections are circles having two common points (the projections of the vanishing points p_{v1} and p_{v2}).

In order to validate the two axes orientation detection, the camera is mounted on a rotating table moved by a high accuracy step motor (0.03 °/step). We apply the method to images of a scene containing two main sets of parallel lines disposed on the 3D pattern shown on the figure 6.a, parameter ω_{thres} being set to 4 °. In this example, 45 lines images have been detected. The best set of potentially parallel lines contains 23 lines and is shown on the figure 6.b. The figure 6.e shows the normals of planes π_j and the corresponding computed orientation (α, β) in 3D. The figures 6.f and 6.g show the two estimated vanishing points on the paracatadioptric image plane. The orientation of the sensor is closely related to the relative position of the vanishing points around the projection of the viewpoint.

We apply the method for different rotations and plot the orientation error as a function of the real orientation on the figure 6.c. 10 tests are done on different images for 9 different values of rotation from -60 ° to 60 ° using a 15 ° step. The plot shows the maximum, minimum and mean error. The figure 6.d shows the result for the other axis obtained by rotating the

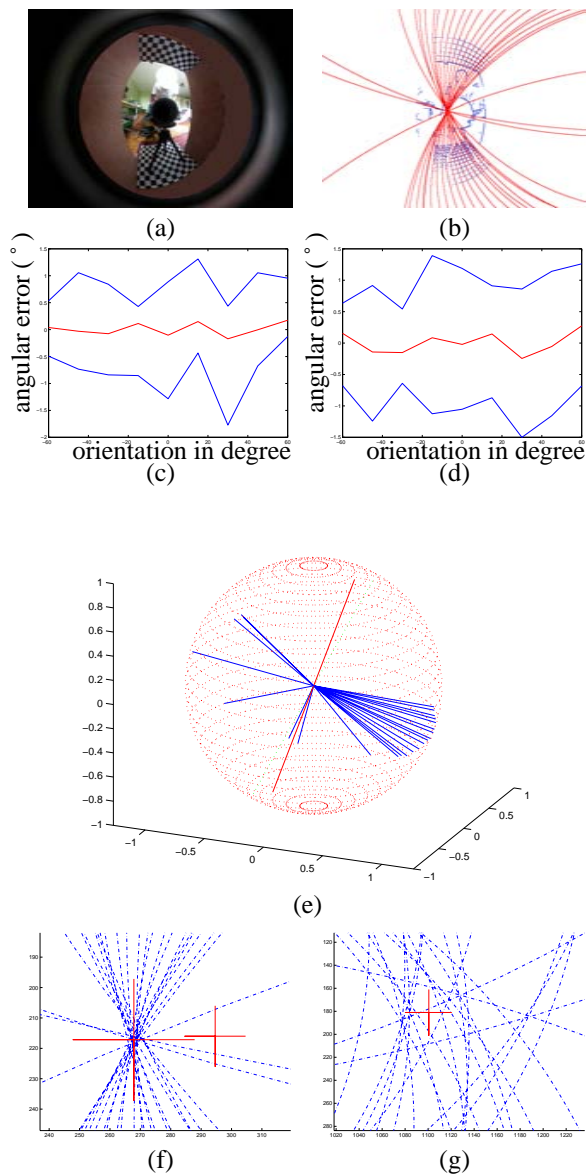


Figure 6: (a) An image of the 3D pattern. (b) Detected lines images. (c) Maximum, mean and minimum error in degree versus ground trust orientation. (d) Same plotting than (c) around the other axis of rotation. (e) The normalized normals to the different planes of the bundle and the retrieved orientation (orthogonal line). (f) One computed vanishing point (Big cross) and projection of the sphere center on the image plane (Small cross). (g) The second vanishing point (Big cross.)

3D pattern by $\pi/2$. These curves show that the estimation is not biased and that the absolute maximum deviation is less than 2° .

Once the orientation around the two axes has been determined, for validation purpose, a rectification of the image is achieved in order to obtain an image from

the same viewpoint but with a fixed sensor's orientation. The figure 7.a shows an image acquired by our camera projected to a cylinder whose axis of rotation is the revolution axis of the paraboloidal mirror. The figure 7.b shows the same image projected to a vertical cylinder, whose relative orientation is estimated using the vertical lines projections. The images of the vertical lines of the scene are approximately vertical in the cylindrical projection, as the orientation of the sensor has been well estimated. A completely rotation invariant image can be generated by shifting the cylindrical image by the detected orientation $\hat{\gamma}_c$.

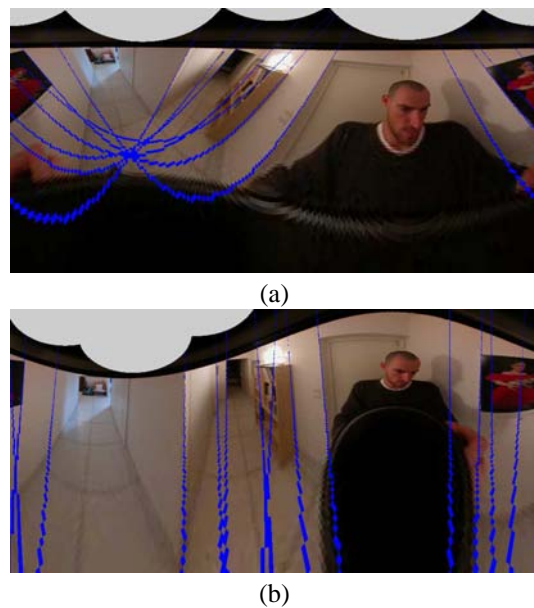


Figure 7: The cylindrical projection of the original (a) and rectified (b) image and some of the detected vertical lines.

We now validate the 2D pose detection using the synthetic map of the fig 3. 100 random poses are generated inside this map and 6 random inliers are kept while 5 random outliers are added. The two first rows of table 1 shows the mean and max absolute deviation for the pose parameters without noise. On the first part of the table, the γ_i are noised. For a random angular maximum deviation of 2° (which correspond to the accuracy of our vertical lines detector), the max pose error is less than 10cm and 2.3° . The second part of the table shows the influence of errors in the 2D map (positions of the vertical lines) using a 2° deviation for γ_i . Errors of about 20cm do not degrade very much the accuracy of the pose estimation. The last column shows how many poses have been completely erroneously estimated because of a bad choice of inliers. Theses false matches are not taken into account in the calculus of the deviations.

The method is now used to localize a real camera

Table 1: Varying deviations of γ_i or (x_i, y_i) over 100 tests and their influence on the absolute deviation of pose estimation parameters. (sub row 1=mean, sub row 2=max). The right column shows the number of images that have been erroneously localized.

$\Delta\gamma_i$	Δx_c (m)	Δy_c (m)	$\Delta\gamma_c(^{\circ})$	bad
0°	0.0007	0.0060	0.0917	0
	0.0677	0.0597	1.7762	
2°	0.0110	0.0127	1.0485	1
	0.0523	0.0998	2.3090	
5°	0.0294	0.0305	2.4981	3
	0.1743	0.1163	4.2628	
10°	0.0501	0.0647	4.8415	4
	0.1820	0.2830	7.3625	
$\Delta(x_i, y_i)$	Δx_c (m)	Δy_c (m)	$\Delta\gamma_c(^{\circ})$	bad
0.2m	0.0467	0.0516	1.1115	2
	0.1895	0.1645	3.6326	
0.5m	0.1321	0.1455	1.6100	4
	0.7256	0.4124	7.7750	

inside three different rooms (approx. surface: 60m²) containing 63 main vertical lines (wall corners, windows and doors borders, racks and desks) whose positions are measured by hand. The fig 7.b shows a part of the indoor environment used in this experiment. We processed 40 images randomly selected from a sequence of 150 frames acquired at 15fps. 32 images were correctly localized, the position of the camera being estimated inside a 20cm tolerance (We use the tiled floor to localize approximately the camera as ground thrust). γ_c is compared with the angle given by an electronic compass whose accuracy is about 2°. The maximum detected orientation error for γ_c in the 32 images is less than 4°. The 8 images which have been erroneously localized can geometrically correspond to different locations due to outliers. However when we use the complete sequence to track the pose and the correspondences between vertical lines and γ_i , all the 150 images are correctly localized.

The detection of the lines images takes about 40 ms (mean time for 150 images for approx 20000 contour points and up to 100 lines to detect). The computation of $(\hat{\alpha}, \hat{\beta})$ is generally achieved in less than 1 ms. The 2D pose estimation time greatly depends of the complexity of the 2D map and the number of the detected vertical lines. In our experiments, the localization of the first image of the sequence has needed 1.2 sec. The tracking of the following poses, however, has been achieved in a few ms. During the first two seconds, the computer processes the first pose and caches the incoming images. Then it processes the cached images more quickly than the acquisition rate and thus can localize in realtime at 15 fps after about

three seconds of initialization.

5 CONCLUSION

In this paper, we have proposed an original method to detect the pose of a central catadioptric camera from an image of a indoor environment containing vertical lines. The two axes orientation detection which is first applied can be used in others applications to detect arbitrary sets of parallel lines. The 2D pose estimation, in spite of its apparent simplicity has exhibited an high computational complexity due to the presence of outliers and unknown matches. We have proposed improvements allowing to achieve the pose estimation in realtime using a smart selection of the correspondences between the lines in the 2D map and the detected vertical lines. Realtime is also obtained thanks to a caching of the images and a tracking of the correspondences inside the sequence. As future work, we plan to integrate colorimetric information to avoid false detections that are geometrically correct and to accelerate the search of the correspondences by discarding incompatible matches. Methods based on 1D Panoramas (Briggs 2005) will also be investigated. Then, experiments on entire buildings will be achieved to validate the approach at a wide scale.

REFERENCES

X. Ying, Z. Hu (2004). Catadioptric Line Features Detection using Hough Transform. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pp 839-842, 2004.

P. Vasseur and E. M. Mouaddib (2004). Central Catadioptric Line Detection. In *BMVC*, Kingston, Sept 2004.

T.Pajdla, V.Hlavac (2001). Image-based self-localization by means of zero phase representation in panoramic images. In *Proceedings of the 2nd International Conference on Advanced Pattern Recognition*, March 2001.

C. Geyer and K. Daniilidis (2001). Catadioptric Projective Geometry. In *International Journal of Computer Vision*, 45(3), pp. 223-243, 2001.

C. Geyer and K. Daniilidis (1999). Catadioptric Camera Calibration. In *Proceedings of the 7th International Conference on Computer Vision*, volume 1, p. 398, 1999.

R. Benosman and S. B. Kang (2001). *Panoramic Vision*. Springer, 2001.

A. Briggs, Y. Li and D. Scharstein (2005). Feature Matching Across 1D Panoramas. In *Proceedings of the OMNIVIS*, 2005.

O. Shakernia, R. Vidal, S. Sastry (2003). Structure from Small Baseline Motion with Central Panoramic Cameras. In *The Fourth Workshop on Omnidirectional Vision*, 2003

ORIENT-CAM, a camera that knows its orientation and some applications

Bertrand Vandepoortae^{1,2}, Christophe Dehais¹, Michel Cattoen², and Philippe Marthon¹

¹ Institut de Recherche en Informatique de Toulouse, France,
bvdvp@enseeiht.fr, <http://www.enseeiht.fr/~bvdvp/>

² Laboratoire d'Electronique de l'ENSEEIHHT, Toulouse, France

Abstract. We introduce a new type of smart cameras. These cameras have an embedded orientation sensor which provides an estimate of the orientation of the camera. In this paper, we describe our prototype orientation sensor and propose some methods for the calibration of the whole camera. We then show two applications. First, the camera is used to create oriented spherical panoramas. Second, it is used for image based localization, in which only the position of the camera has to be retrieved.

1 Introduction

Many researchers are working on methods providing the camera pose (orientation and translation) from the image. These methods use either some knowledge about the observed scene (for example, a 3D model) or more than one image in order to determine the camera pose and a partial model of the scene at the same time. In the first case, the recovered pose is relative to the 3D model. In the second one, the different poses are generally expressed as a function of the first pose and are defined up to a unknown scale.

In this paper, we propose to use an electronic device fastened to a camera in order to determine its orientation. This absolute orientation is defined in a fixed orthonormal world coordinates system in which two axes correspond to the vertical and magnetic north directions.

Some commercial digital cameras contain an embedded orientation sensor. This sensor is not of the same type that the one described here, as it only allows to determine if the captured image is in portrait or landscape orientation. Our sensor, on the other hand, allows to know in which direction each pixel of the camera is pointing.

In [7], a system combining a camera and inertial sensors is proposed to detect the vertical reference. The authors of [8] proposed a more advanced solution, using both inertial sensors and gyroscopes. The magnetic sensors we use do not deviate in time as gyroscopes do, and we can so expect to obtain the correct absolute orientation whereas they can only expect a relative one.

The paper is organized as follows. First, we describe the camera and the orientation sensor. Second, we propose some methods for the calibration of the

complete sensor. Third, we use the camera to create automatically omnidirectional oriented panoramas. Finally, we show how this camera can be used for image based localization and conclude with some perspectives.

2 Description of the camera

Our aim was to create an handheld and low cost smart camera. The figure 1 shows the resulting prototype whose dimensions are 100*60*22mm and cost is about 150 USD. Both can be greatly reduced if the sensor is mass produced.



Fig. 1. The complete camera whose thickness is about 2cm.

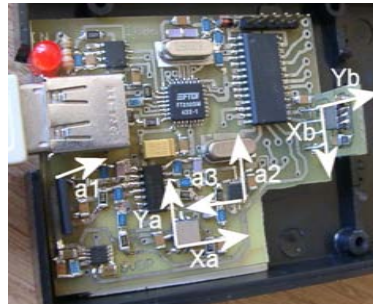


Fig. 2. The orientation sensor and its different axes of sensor. Both a_1 and Y_b are orthogonal to the main board.

2.1 The orientation sensor

The figure 2 shows the orientation sensor prototype we have built and integrated inside the camera. Its dimensions are 45*60*10mm. It is mainly composed of low cost integrated Micro Electro Mechanical Systems (iMEMS), namely accelerometers and magnetometers.

We use the magnetic sensors HONEYWELL HMC1051Z and HMC1052 in order to have a 3D measurement of the magnetic field (on the axes a_1, a_2 and a_3 shown on the figure 2). We also use two double axis accelerometers MEMSIC MXD3334UL mounted orthogonally to sense the gravity. Each MXD3334UL includes 2 orthogonal accelerometers. A single chip would lead to inaccuracies when one of the two sensor's axis is near the gravity, because each measurement corresponds to the sinus of the angle between the sensor axis and the horizontal plane. When this angle is near 90 degrees, a change of orientation of a few degrees cannot be detected as its sinus changes very few. Using at least 3 sensors allows to always have some sensor axes far away from the gravity axis and hence ensure more accuracy. Finally, the acceleration is sensed on the axes X_a, Y_a, X_b and Y_b as shown on the figure 2.

A Microchip PIC 18F252 microcontroller gathers the different sensors' measures and send them to the computer using a USB connexion thanks to the

FTDI FT232BM circuit. The complete set of measures is sent at 100Hz allowing a realtime orientation estimation in our computer vision task.

It is noticeable that the sensor provides redundant information as only 3 parameters (the rotations) have to be retrieved from 7 measurements. This allows to compute a more accurate orientation and to detect some incoherences such as parasite accelerations due to movements or magnetic field perturbations. Hence a confidence measure in the retrieved orientation is provided.

2.2 The camera

The imaging sensor can be either a digital camera or a Webcam. In our prototype, we use the electronic parts of a Logitech QuickCam Pro 4000 at a resolution of 352*288 pixels and at 15 fps. This camera provides the ability to control the shutter speed and gain. Moreover, our model has very low radial distortions, and so can be well approximated by a simple pinhole model.

3 Calibration

3.1 The orientation sensor

The data provided by the different sensors have to be corrected in order to determine the orientation. The adequate correction is computed thanks to a calibration procedure that we describe hereafter.

Each magnetometer provide an analogous output. Offsets and gains have to be applied to each channel to ensure that the norm of the magnetic field is approximately constant when the sensor is rotated in any direction. They are estimated by acquiring a sequence of measurements while the sensor is rotating. A cost function whose value is minimal when the variance of the corrected norm is the lowest is then minimized to find the best set of parameters.

Oppositely, the accelerometers needs to be calibrated in static positions to avoid parasite accelerations. The orientation sensor is held in different known orientations and many measurements are done on each channel to compute mean values corresponding to extremal orientations (vertical and horizontal on each axis). The outputs of these circuits are Pulse Width Modulated with a duty cycle between 20 and 70%. The calibration consists in the estimation of a rectification function that provides a duty cycle of 20 or 70% when the sensor is held vertically and of 50% when the sensor is held horizontally.

The next calibration step is optional and only provides better accuracy. It consists in the estimation of the 3D direction of the earth magnetic field at the current location. The field has a different vertical component depending on the latitude on earth. The angle between the magnetic field and the horizontal is estimated by holding the sensor horizontally and deducing the angle directly from the measures given by the sensor.

Once the calibration has been achieved, a first estimate of the orientation is computed using only two accelerometers and a 2D projection of the magnetic

field to the estimated horizontal plane. The orientation is then refined by taking into account the data from the 7 sensors. The orientation is parameterized with a quaternion to avoid discontinuity problems and gimbal lock caused by Euler angles parametrization.

Because of the noise in the measurements, we apply a temporal filter to increase the stability and accuracy. Our first idea was to implement a Kalman filter as proposed in [2], but as we have no idea of the movement of the sensor at a given time, we would have modeled it as static. Under this condition, the Kalman filter is equivalent to a standard least squares estimator that we have preferred to implement. Let w defines the size of a temporal window in which the measures of the sensors previously acquired between times $t - w$ and t are used to compute the orientation at instant t . The samples in this windows are weighted differently in order to make the samples which are close in time to the last acquired sample more influent. Increasing (resp. decreasing) w makes the detected orientation more stable (resp. reactive). Values of w between 10 and 20 have provided good results in our applications.

3.2 The camera and its relative orientation

Once the orientation sensor has been calibrated, it is necessary to calibrate the whole camera, *ie.* estimate its internal parameters and orientation relatively to the orientation sensor.

This is achieved by acquiring pictures I_j of a checkerboard pattern whose one axis is oriented to the magnetic north. We then use the Jean-Yves Bouguet's Complete Camera Calibration Toolbox for Matlab [5] to estimate the pose of the camera for each image (Ri_j, Ti_j) and its intrinsic parameters. Using the orientation given by the orientation sensor for each image Rs_j , we then compute a rotation R_Δ which minimizes $\sum_{j=1}^n (R_\Delta Ri_j - Rs_j)^2$ over the n images. This provides the relative orientation of the camera much more accurately than the measure from only one view. Moreover, in order to avoid bias in the estimation of R_Δ , it is better to sample the complete space of possible orientations.

The final detected standard deviation between $R_\Delta Ri_j$ and Rs_j is about 1.5 degree and the max deviation is about 4 degrees.

4 Oriented panoramas

4.1 Their inherent advantages

The most obvious applications for our camera is the automatic stitching of images to create panoramas. It consists in generating a panoramic image from the different images acquired by the camera without needing to detect some common points in the different images, as Brown and Lowe do in [9]. Our approach has mainly four main advantages, thanks to the orientation sensor.

First, the reconstructed panoramas are oriented, so the horizon of the scene is projected on a straight horizontal line and the first column of the panoramic image can point to the magnetic north for example.

Second, as the orientation of the camera is known a priori, homologous points detection is not needed to stitch the images. This allows to deal with panoramas which contains big portions of non textured areas whose images cannot be automatically stitched by standard methods.

Third, the knowledge of the orientation simplifies greatly the creation of complete spherical panoramas. The problem of closings usually appears when one try to stitch images under a full 360 degrees rotation of the camera. Because of the accumulated errors, it is quite hard to stitch the last image and the first one. This problem can be avoided by using bundle adjustment to estimate the whole solution for the stitching instead of stitching iteratively new images to the panorama as proposed in [1] but this method requires a good initial guess for the whole correspondances.

Finally, our camera allows to deal very efficiently with scenes containing moving object which are outliers in the panorama reconstruction. Methods generally used to compute panoramas of such scenes involve a background/foreground segmentation in order to detect the camera orientation relatively to the background. With our sensor, this is done very simply by detecting zones in the images whose movements are incoherent with the orientation provided by the orientation sensor. The pixels of these zones are labeled as foreground and are discarded in the panorama reconstruction.

4.2 How to create the panoramas

We recall that a panorama can be created from images if the viewpoint of the camera does not move (or at least, its displacement should be very small compared with the distance to the observed scene), id. that the images are only related by a rotation of the camera. In this paper, we consider that the intrinsic parameters of the camera do not change between the images.

Using the orientation provided by the orientation sensor, it is possible to project each image of the camera to the right location on the panorama. However, due to imprecisions in the orientations of the different images, the different images do not exactly connect together. We first show how to obtain an approximate panorama in real-time and then propose to use the bundle adjustment method to improve its quality.

The resulting panorama we create is spherical, id. each pixel of the resulting image corresponds to a point on the sphere whose spherical coordinates are related to the pixel position. Any surface could be used to sample the image but the sphere allows an omnidirectional field of view.

Let the point P_i be one 3D position on the sphere corresponding to a pixel i . Let the quaternion corresponding to the orientation of the image j be defined by $Q_j = [abcd]^T$. The j indices are omitted in the quaternion parameters for readability purpose. The corresponding rotation matrix R_j is defined in the equation (1).

Let the camera matrix K be defined in the equation (2), α_u (resp. α_v) being the horizontal (resp. vertical) focal length and $[p_u p_v]^T$ being the position of the principal point. The points $P_i = [x_i y_i z_i]^T$ of the sphere is projected

in the image j to the homogeneous point $p_{Hij} = [u_{ij} \ v_{ij} \ w_{ij}]^T$ whose inhomogeneous coordinates are $p_{ij} = [u_{ij}/w_{ij} \ v_{ij}/w_{ij}]^T$ obtained by the function $p_{ij} = D(p_{Hij})$. The image p_{ij} of the point P_i in the image j is obtained as shown in the equation (2).

$$R_j = \begin{bmatrix} 1 - 2b^2 - 2c^2 & 2ab - 2cd & 2ac + 2bd \\ 2ab + 2cd & 1 - 2a^2 - 2c^2 & 2bc - 2ad \\ 2ac - 2bd & 2bc + 2ad & 1 - 2a^2 - 2b^2 \end{bmatrix} \quad (1)$$

$$K = \begin{pmatrix} \alpha_u & 0 & p_u \\ 0 & \alpha_v & p_v \\ 0 & 0 & 1 \end{pmatrix}; p_{ij} = D(KR_jP_i) \quad (2)$$

If p_{ij} lies inside the image boundaries and if the z component of R_jP_i is positive (id. the camera is facing the point), then the point P_i is viewable on the image j and the RGB values of p_{ij} are used in the panorama for the pixel i .

This computation is achieved for the whole sphere at each frame in order to update the panorama. The figure 3 shows an image from the camera and the reconstructed low resolution (1024*512 pixels) oriented panorama obtained in real time from a sequence of approximately 10 seconds. No post processing is applied; the panorama only results from the projections of the different images to the sphere. The left column of the reconstructed panorama points to the north and the horizon of the scene corresponds to an horizontal line in the panorama. Note the narrow field of view of the camera. The scene contains large un-textured areas on the roof, walls and a over exposed area at the window. The images of these parts would be impossible to stitch automatically as they do not contain any discriminant points.



Fig. 3. Left: An image from the camera. Right: The reconstructed oriented panorama.

4.3 Refinement of the panoramas

The bundle adjustment is used to improve the estimation of the different images orientations. This has already been proposed in [1] but they were not using the a priori knowledge provided by the orientation sensor. The reader can find more information about the bundle adjustment techniques in [6]. This problem can be stated as an iterative minimization of a cost function f defined in equation (3). A detected projection of the point i in the image j is denoted $\widetilde{p}_{ij} = [\widetilde{u}_{ij} \ \widetilde{v}_{ij}]^T$ and is called an observation. Each observation of a point projection provides 2 residuals (res_u and res_v).

$$f = \sum_{j=1}^n \left(\sum_{i=1}^m (\|D(KR_j P_i) - [\widetilde{u}_{ij} \ \widetilde{v}_{ij}]^T\|^2) \right) = \sum_{j=1}^n \left(\sum_{i=1}^m (res_u^2 + res_v^2) \right) \quad (3)$$

Not all the images (the camera is acquiring images at 15fps) are needed to refine the panorama, so the set of images have to be decimated in order to avoid useless computations. The decimation is very simple and consists in selecting images such that they have at least half their surface in common with the other images. The approximate orientation from the sensor is accurate enough to compute the set of images to keep.

The computations cannot be achieved on every pixels of the retained images and so some of them have to be selected. We use the famous Harris corner detector [3] to detect the projections of corners. Thanks to the knowledge of the approximate orientation of each image, there is no need to use descriptors to detect correspondences between the points in the different images. Instead of this, we simply project the detected Harris points of the different images to the sphere in order to detect the correspondences. A correspondence is established if some projected points from different images are closer than a threshold and if there is no other Harris points in the same images too close in their neighborhood.

Once the correspondences have been detected, the cost function is iteratively minimized. This way, we refine the intrinsic parameters, the quaternions Q_j and the P_i . The convergence is very fast in general and the computation can be achieved very efficiently thanks to the sparse structure of the jacobian. As the accuracy of the orientation sensor is known, we define some limits on the possible values for the quaternion parameters during the iterations. If a quaternion exceed these limits, we take it back to the limit. This ensure that the orientations of the different images cannot deviate too much during the bundle adjustment.

Some images contain no Harris point at all. These images will not have any observation and will not be related in the jacobian matrix. So their orientation will not change. Hopefully, the images which do not contains Harris point are likely to be untextured, so small misalignments due to the orientation sensor inaccuracies are not usually noticeable.

Finally, a post processing is applied in order to smooth the discontinuities at the image boundaries.

The figure (4) shows the results of the bundle adjustment and post processing on a part of the panorama. Only five iterations were necessary to adjust the

whole panorama . Only four images are rendered here in order to see the details. Note that the discontinuities are attenuated in the refined image.

We have not deal with radial distortions because our camera does not require it. However it can be included very easily in the model, both in the calibration and bundle adjustment steps.



Fig. 4. (a) A part of the panorama from 4 images before the bundle adjustment and post processing. (b) The same part after refinement and post processing.

5 Image Based localization

Our camera can also be used as an image based substitute for the GPS inside buildings, where the satellite signals cannot be received. In our previous works, we were using catadioptric camera in order to achieve this task. However, these cameras are quite expensive and fragile. Their size do not really allows to use the term "handheld" and special care must be taken in order to avoid scratches and dust that reduce the image quality. Finally, their resolution is lower than the one available from stitched images and the exposure has to be set for the whole scene, resulting in over or under exposed areas. Their main advantage is their ability to acquire more than an hemisphere of the scene in only one shot. Another advantage is that they allow to retrieve the orientation relatively to the horizon using the projection of vertical lines which are numerous inside buildings. However, the orientation to the magnetic north obviously cannot be determined automatically from the image.

In this section, we propose a very simple system to illustrate the ability of our camera. A database B of images I_P of the building in which we want to localize our camera (from its image I_C) is acquired using a catadioptric camera with a parabolic mirror. The acquired area is sampled at a given altitude and every 30cm in an horizontal plane. The paracatadioptric camera is held horizontally and with the columns of the image sensor pointing to the north (using a compass) so the database is at least approximately oriented.

The localization of I_C is achieved after computing a rectified image I_R (which is of the same type than I_P) and finding the closest image I_P in B . Instead of the sphere (as in the section 4), the surface of the paracatadioptric image is sampled. This way I_C is projected to a portion of a paracatadioptric image I_R and can be matched with images from B . Thanks to the orientation knowledge, I_R should superpose to a I_P acquired from the same location. A color rectification is also applied in order to make the images matchable.

In order to avoid costly comparisons of I_R with all the I_P , a first detection is achieved using corners detected by the Harris corner detector. The database is processed offline. Local descriptors are computed to describe every corner points. A famous descriptor widely used in the computer vision community is the Scale Invariant Feature Transform (SIFT) from David Lowe [4]. It provides very good results for matching image points related by affine transformations. However, thanks to the orientation sensor, such a complex descriptor is useless, as the points are observed in similar orientations and from very close positions (less than 20cm). In this experiment, we only use square windows of pixels to describe the points. These descriptors are simpler to compute and are also more discriminant, as they do not give good matching scores to couples which are related by affine transformation.

When an image I_C has to be localized, the Harris points H_C are detected in I_C using a variable size for the gaussian filter in order to deal with the varying resolution of the paracatadioptric camera. I_C and H_C are then projected to the paraboloid and the descriptors corresponding to the projected H_C are computed in I_R .

In order to match the images efficiently, we use a decomposition of the images in annular portions of sectors. The potential Harris points correspondences between I_R and the whole set of I_P are only compared if the points lie in the same or one of the 8 neighbor portions. The computed score is a ZNCC measure between the descriptors, which allows to deal with light variations. Harris points which do not have any correspondence get the worst score. The partial correlation score computed for an image I_P is the sum of the scores of the different points.

A final verification step is applied in order to check the correspondences which have the highest scores, and to decide between different solutions if needed (when many correspondences have nearly the same scores). This step consists in trying to match every pixels of I_R with one pixel in I_P . The ZNCC score is computed for each pixel of the I_R with pixels from I_P in a centered window neighborhood. The lowest score is kept for each pixel, allowing small displacement in the images. The resulting score for each image correspondence is the sum of the pixels' scores.

During our preliminary experiments, we have observed that this simple localization is robust to occlusions up to 30% of the surface of the image. When we integrate many images I_R to create a paracatadioptric panorama from the current location, the robustness of the localization is greatly increased. Obviously, the localization fails if the camera is pointed only to a region which is not discriminant, such as an uniform wall for example.

The figure 5 shows an example of the method. Less than 5 seconds are necessary to retrieve the correct I_P from a database of more than 200 images using a pentium 4 computer. The panoramic images were divided in 10 angular sectors, each one divided in 4 portions.

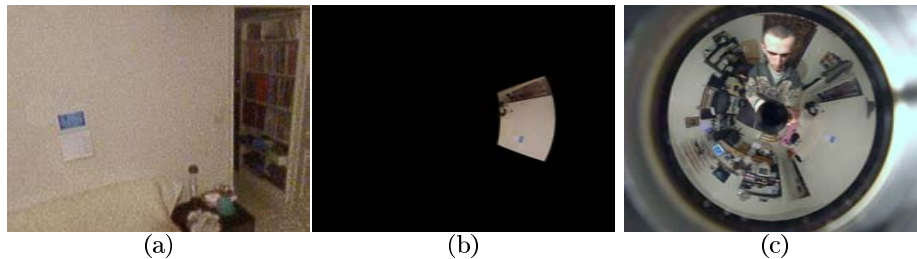


Fig. 5. (a) The image I_C from the camera to be localized. (b) The projection I_R of the image to the surface of the paracatadioptric mirror. (c) The closest panoramic image I_P found in the database.

6 Conclusions and perspectives

As far as we know, this paper presents the first handheld camera containing a complete absolute orientation sensor. We have shown that this camera is compact, low cost, practical and quite easy to calibrate. This sensor could be improved to obtain more accurate measures but it is sufficient for the proposed applications. We believe that this kind of camera have many other applications and should interest many researchers. Future work will focus on improving the image based localization system in order to avoid the dense sampling of panoramic images while acquiring the database. We are trying to use either a reconstructed 3D model of the scene or a sparse set of panoramic images.

References

1. Philip F. Mc Lauchlan and Allan Jaenicke: Image Mosaicing using Sequential Bundle Adjustment, BMVC 2000.
2. E.H. Shin: A Quaternion-based unscented Kalman filter for the integration of GPS and MEMS, ION GNSS 2004.
3. C. Harris and M.J. Stephens: A combined corner and edge detector, In Alvey Vision Conference, pages 147-152, 1988.
4. David G. Lowe: Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.
5. Jean-Yves Bouguet, Complete Camera Calibration Toolbox for Matlab, <http://www.vision.caltech.edu/bouguetj/>
6. C. Slama: Manual of Photogrammetry, American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.
7. Jorge Lobo, Jorge Dias: Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference, PAMI, December 2003 (Vol. 25, No. 12), pp. 1597-1608.
8. Thomas Schon and Fredrik Gustafsson: Integrated Navigation of Cameras for Augmented Reality, IFAC 2005.
9. M. Brown, D. G. Lowe: Recognising Panoramas, In Proceedings of the 9th International Conference on Computer Vision (ICCV2003).

Bibliographie

- [1] E. M. Mouaddib, "La Vision Omnidirectionnelle," *Conférence invitée aux Journées Nationales de la Recherche en Robotique (JNRR 2005), Octobre 2005.*
- [2] J.-J. Gonzalez-Barbosa, "Vision panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images," *Thèse de doctorat, LAAS.*
- [3] T.Pajdla et V.Hlavac, "Image-based self-localization by means of zero phase representation in panoramic images," *In Proceedings of the 2nd International Conference on Advanced Pattern Recognition, volume 2013 of Lecture Notes in Computer Science pages 24-33, Heidelberg, Germany, March 2001, IAPR, Springer-Verlag.*
- [4] A. A. Argyros, K. E. Bekris, et S. C. Orphanoudakis, "Robot Homing based on Corner Tracking in a Sequence of Panoramic Images," *IEEE Computer Society's Computer Vision and Pattern Recognition (CVPR 2001) Hawaii, Dec 2001.*
- [5] L. Paletta, S. Frintrop, et J. Hertzberg, "Robust Localization Using Context in Omnidirectional Imaging," *Proc. IEEE International Conference on Robotics and Automation, ICRA 2001, pp. 2072-2077.*
- [6] M. Fiala, "Panoramic Computer Vision," *Thèse de doctorat.*
- [7] D. Bradley, A. Brunton, et M. Fiala, "Image-Based Navigation in Real Environments Using Panoramas," *IEEE International Workshop on Haptic Audio Environments and their Applications (HAVE'2005), Octobre 2005.*
- [8] C. Gonzalez-Buesa et J. Campos, "Solving the mobile robot localization problem using string matching algorithms," *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04).*
- [9] J. A. da Cruz Pinto Gaspar, "Omnidirectional Vision for Mobile Robot Navigation," *Thèse de doctorat, décembre 2002.*
- [10] I. Ulrich et I. Nourbakhsh, "Appearance-Based Place Recognition for Topological Localization," *IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000, pp. 1023-1029. Best Vision Paper Award.*
- [11] T. Pajdla, "Robot Localization Using Shift Invariant Representation of Panoramic Images," *Research report, K335-CMP-1998-170.*
- [12] M. Fiala et A. Basu, "Robot Navigation Using Panoramic Landmark Tracking," *Society of Manufacturing Engineers (SME). Article RPOS-100. Dearborn, Michigan, États-Unis. 2003. NRC 47136.*

- [13] R. Bunschoten, "Mapping and Localization from a Panoramic Vision Sensor," *Thèse de doctorat, Novembre 2003*.
- [14] J. Y. Zheng et S. Tsuji, "Panoramic Representation for Route Recognition by a Mobile Robot," *International Journal of Computer Vision*, 9 :1, pp 55-76, 1992.
- [15] S. Hrabar et G. S. Sukhatme, "Omnidirectional Vision for an Autonomous Helicopter," *In proceedings of the IEEE International Conference on Robotics and Automation (ICRA03)*, pp 3602-3609, avril 2004.
- [16] C. Demonceaux et P. Vasseur, "Mesure d'Attitude pour les Drones par Vision Catadioptrique Centrale," *RFIA 2006*.
- [17] S. Ikeda, T. Sato, et N. Yokoya, "Panoramic Movie Generation Using an Omnidirectional Multi-camera System for Telepresence," *Proc. 13th Scandinavian Conf. on Image Analysis (SCIA2003)*, pp. 1074-1081, July 2003.
- [18] R. Benosman et S. B. Kang, "Panoramic Vision : Sensors, theory and applications," *Ed Springer Verlag, 2001*.
- [19] G. Cielniak, M. Miladinovic, D. Hammarin, L. Goranson, A. Lilienthal, et T. Duckett, "Appearance-based Tracking of Persons with an Omnidirectional Vision Sensor," *In Proceedings of the Fourth IEEE Workshop on Omnidirectional Vision (Omnivis 2003)*.
- [20] H. N. Charif et S. J. McKenna, "Tracking the activity of participants in a meeting," *Machine Vision and Applications Journal, Springer Berlin, Heidelberg, Volume 17, Number 2, May 2006*, pp 83-93.
- [21] J. Fabrizio, "Localisation d'obstacles coopératifs par systèmes de vision classiques et panoramiques.," *Thèse de doctorat, décembre 2004*.
- [22] M. Douze, "Estimation d'homographies inter-images, Cas des mosaïques et du suivi en temps réel, Applications en réalité augmentée," *Thèse de doctorat, Institut National Polytechnique de Toulouse, Novembre 2004*.
- [23] J. J. Gonzalez-Barbosa et S. Lacroix, "Localisation d'un robot mobile par indexation d'images panoramiques," *RFIA 2002*.
- [24] J.-J. Gonzalez-Barbosa et S. Lacroix, "Rover localization in natural environments by indexing panoramic images," *ICRA 2002*, pp 1365-1370.
- [25] M. Jogan et A. Leonardis, "Robust Localization Using Panoramic View-Based Recognition," *In 15th ICPR, volume 4, pages 136-139. IEEE Computer Society, September 2000*.
- [26] E. Menegatti, M. Zoccarato, E. Pagello, et H. Ishiguro, "Image-Based Monte-Carlo Localisation with Omnidirectional Images," *Robotics and Autonomous Systems, Elsevier Volume 48, Issue 1 , 31 August 2004, Pages 17-30*.
- [27] A. Remazeilles, F. Chaumette, et P. Gros, "3D navigation based on a visual memory," *Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, Floride, USA, Mai 2006*.
- [28] A. J. Briggs, C. Detweiler, P. C. Mullen, et D. Scharstein, "Scale-Space Features in 1D Omnidirectional Images," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2004), April, 2004*.

- [29] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, et L. Iftode, "Indoor Localization Using Camera Phones," *7th IEEE Workshop on Mobile Computing Systems and Applications, 2006. WMCSA '06. Proceedings.*
- [30] C. Cauchois, E. Brassart, B. Marhic, et C. Drocourt, "An Absolute Localization Method using a Synthetic Panoramic Image Base," *Proc. of the IEEE Workshop on Omnidirectional Vision (OMNIVIS'02) at European Conference on Computer Vision (ECCV 2002), Copenhagen, Denmark, June 2, 2002, p. 128-135.*
- [31] B. Vandeportaele, "Un système de guidage de personnes utilisant des images omnidirectionnelles," *Congrès jeunes chercheurs en Vision par ordinateur, ORASIS'2005, Clermont-Ferrand, France, Mai 2005.*
- [32] B. Vandeportaele, M. Cattoen, et P. Marthon, "Image Rectification and Synthesis for an Image Based Automatic Guidance System in a 3D Modelled Site," *PSIP 2005 : Physics in Signal and Image Processing.*
- [33] B. Vandeportaele, M. Cattoen, et P. Marthon, "Realtime Localization of a Central Catadioptric camera Using Vertical Lines," *VISAPP 2006 : International Conference on Computer Vision Theory and Applications Setubal, Fevrier 2006.*
- [34] B. Marhic, E. Mouaddib, D. Fofi, et E. Brassart, "Localisation absolue par le capteur omnidirectionnel SYCLOP," *Dans la revue Traitement du Signal, Vol. 17, N°3, 2000.*
- [35] B. Vandeportaele, C. Dehais, M. Cattoen, et P. Marthon, "ORIENT-CAM, A camera that knows its orientation and some applications," *CIARP 2006 : 11th Iberoamerican Congress on Pattern Recognition.*
- [36] C. Fermüller et Y. Aloimonos, "Geometry of Eye Design : Biology and Technology," *Theoretical Foundations of Computer Vision 2000 : pp 22-38.*
- [37] C. Harris et M. Stephens, "A combined corner and edge detector," *In Alvey Vision Conference, pages 147-152, 1988.*
- [38] C. Tomasi et T. Kanade, "Detection and Tracking of Point Features," *Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.*
- [39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision, 60, 2, (2004), pp. 91-110.*
- [40] J. Matas, O. Chum, M. Urban, et T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *BMVC 2002.*
- [41] M. Irani et P. Anandan, "About direct methods," *In Workshop on Vision Algorithms : Theory and Practice, 1999.*
- [42] A. Bartoli, "Groupwise Geometric and Photometric Direct Image Registration," *BMVC'06 - In Proceedings of the Seventeenth British Machine Vision Conference, September 2006.*
- [43] M. Brown et D. G. Lowe, "Recognising Panoramas," *In Proceedings of the 9th International Conference on Computer Vision (ICCV2003), pages 1218-1225, Nice, France, October 2003.*

- [44] Y. Wexler et D. Simakov, "Space-Time Scene Manifolds," *International Conference on Computer Vision, 2005, volume 1, pages 858-863*.
- [45] A. Rav-Acha, Y. Pritch, D. Lischinski, et S. Peleg, "Dynamosaics : Video Mosaics with Non-Chronological Time," *International Conference on Computer Vision and Pattern Recognition - CVPR 2005, San Diego, CA, USA, June 20-26 2005*.
- [46] R. Benosman, T. Maniere, et J. Devars, "Multidirectional stereovision sensor, calibration and scenes reconstruction," *Proceedings of the 13th International Conference on Pattern Recognition, Volume 1, Aug. 1996, pp :161-165*.
- [47] O. ROMAIN, T. EA, C. GASTAUD, et P. GARDA, "Un capteur multi-spectral de vision panoramique 3D," *Actes des journées francophones des jeunes chercheurs en analyse d'images et perception visuelle, ORASIS'2001, Cahors, Juin 2001*.
- [48] A. Waggershauser, "Combining Full Spherical Depth and HDR Images to Implement a Virtual Camera," *Proceedings of the ISPRS working group, 'Panoramic Photogrammetry Workshop', Berlin, Germany, 24-25 February 2005*.
- [49] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L. wei He, A. Colburn, Z. Zhang, Z. Liu, et S. Silverberg, "Distributed Meetings : A Meeting Capture and Broadcasting System," *ACM Multimedia, 2002*.
- [50] T. Sato et N. Yokoya, "Omni-directional Multi-baseline Stereo without Similarity Measures," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [51] F. Kangni et R. Laganière, "Epipolar Geometry for the Rectification of Cubic Panoramas," *Third Canadian Conference on Computer and Robot Vision, CRV 2006*.
- [52] B. Micusik et T. Pajdla, "Structure from Motion with Wide Circular Field of View Cameras," *IEEE Transaction on Pattern Analysis and Machine Intelligence, VOL. 28, NO. 7, JULY 2006*.
- [53] S. Gourichon, J. Meyer, S. H. Ieng, L.Smadja, et R. Benosman, "Estimating ego-motion using a panoramic sensor : Comparison between a bio-inspired and a camera-calibrated method," *Symposium on Biologically Inspired Vision, Theory and Application (AISB 03)*.
- [54] M. Lhuillier, "Automatic Structure and Motion using a Catadioptric Camera," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [55] P. Sturm et S. Ramalingam, "Géométrie d'images multiples pour des modèles de caméras généraux," *Traitement du Signal, 2005, Vol. 22, N° 5, p. 483-495*.
- [56] O. Shakernia, R. Vidal, et S. Sastry, "Structure from Small Baseline Motion with Central Panoramic Cameras," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003*.
- [57] C. Geyer, S. Sastry, et R. Bajcsy, "Euclid meets Fourier : Applying harmonic analysis to essential matrix estimation in omnidirectional cameras," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2004), April, 2004*.

- [58] D. Slater, "Panoramic Photography with Fisheye Lenses," *Published in the International Association of Panoramic Photographers Journal in 1996.*
- [59] K.-H. Jeong, J. Kim, et L. P. Lee, "Biologically inspired artificial compound eyes," *Science*, 312, p. 557-561 (2006).
- [60] D. Diderot et J. le Rond d'Alembert, "Encyclopédie des Sciences, des Arts et des Métiers," 1751-1772.
- [61] D. Rees, "Panoramic television viewing system," *United States Patent No. 3,505,465, 1970.*
- [62] Y. Yagi et S. Kawato, "Panoramic scene analysis with conic projection," *IROS 1990.*
- [63] S. Baker et S. K. Nayar, "A Theory of Catadioptric Image Formation," *In Proceedings of ICCV, pages 35-42, Bombay, India, January 1998.*
- [64] J. Gluckman et S. K. Nayar, "Rectified Catadioptric Stereo Sensors," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2000 - CVPR'2000 South Carolina, USA, June 2000.*
- [65] J. Gluckman et S. K. Nayar, "Rectified Catadioptric Stereo Sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, February 2002 p.224.*
- [66] T. Nishimura, T. Mukai, et R. Oka, "Spotting recognition of gestures performed by people from a single time-varying image," *International Conference on Robots and Systems 1997, pp 967-972.*
- [67] M. Wachs, D. Crispell, et G. Taubin, "Calibrating a Catadioptric Light Field Array," *Siggraph 2005.*
- [68] S. Lin et R. Bajcsy., "True single view point cone mirror omni-directional catadioptric system.," *In ICCV01, pages II :102-107, 2001.*
- [69] R. Swaminathan, M. D. Grossberg, et S. K. Nayar, "Non-Single Viewpoint Catadioptric Cameras : Geometry and Analysis," *International Journal of Computer Vision, Vol 66, Issue 3, March 2006, pages : 211-229.*
- [70] S. Ieng et R. Benosman, "Les surfaces caustiques par la géométrie - application aux capteurs catadioptriques," *Revue Traitement du signal TS ,Numéro spécial Vision omnidirectionnelle 2005.*
- [71] S. Gaechter, T. Pajdla, et B. Micusik, "Mirror Design for an Omnidirectional Camera with a Space Variant Imager," *Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing Workshop. Budapest, August 2001.*
- [72] F. Berry, P. Martinet, Y. Mezouar, F. Chaumette, F. Combi, O. Strauss, et M. Mouadib, "OMNIBOT - De la vision omnidirectionnelle à la commande de robot mobile," *ROBEA 2005.*
- [73] M. Ollis, H. Herman, et S. Singh, "Analysis and Design of Panoramic Stereo Vision Using Equi-Angular Pixel Cameras," *rapport CMU-RI-TR-99-04, The Robotics Institute of Carnegie Mellon University, January 1999.*

- [74] J. Chahl et M. Srinivasan, "Reflective surfaces for panoramic imaging," *Applied Optics*, 1997.
- [75] R. A. Hicks et R. Bajcsy, "Catadioptric Sensors that Approximate Wide-angle Perspective projections," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2000 - CVPR'2000 USA, June 2000*.
- [76] R. A. Hicks et R. K. Perline, "Equirectangular Catadioptric Sensors," *Applied Optics* 44, 6108-6114 (2005).
- [77] R. Hartley et A. Zisserman, "Multiple View Geometry in Computer Vision," *Cambridge University Press, July 31, 2000*.
- [78] S. B. Kang, "Catadioptric Self-Calibration," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2000 - CVPR'2000*.
- [79] S. Laveau et O. Faugeras, "Oriented Projective Geometry For Computer Vision," *ECCV 1996, pp. 147-156*.
- [80] C. Geyer et K. Daniilidis, "A Unifying Theory for Central Panoramic Systems and Practical Implications," *ECCV 2000*.
- [81] X. Ying et Z. Hu, "Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model," *In Proc. European Conference on Computer Vision (ECCV 2004), Vol. I, pp.442-455, Prague, Czech, 2004*.
- [82] C. Geyer et K. Daniilidis, "Paracatadioptric Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 24, N5, may 2002*.
- [83] F. Comby, C. C. D. Kerleau, et O. Strauss, "Étalonnage de caméras catadioptriques hyperboloïdes," *Revue Traitement du signal TS, Numéro spécial Vision omnidirectionnelle 2005, 2005*.
- [84] H. Bakstein et T. Pajdla, "An overview of non-central cameras," *In Proceedings of Computer Vision Winter Workshop, pages 223-233, Ljubljana, Slovenian Pattern Recognition Society, Slovenia, February 2001*.
- [85] T. PAJDLA, "Stereo with Oblique Cameras," *IJCV 2002 j'ai pas compris ce que c'est mais apparemment un modele general non central et il decrit la geometrie stereo pour ce type de camera*.
- [86] R. Pless, "Using Many Cameras as One," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'03, volume II : pp 587-593, June 2003*.
- [87] T.-H. Ho, C. C. Davis, et S. D. Milner, "Using Geometric Constraints for Fisheye Camera Calibration," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [88] A. S. Glassner, A. Arvo, R. L. Cook, E. Haines, P. Hanrahan, P.S. Heckbert, et P. Hanrahan, "An Introduction to Ray Tracing," *Harcourt Brace Jovanovich, 1989*.
- [89] B. Micusik, "Two View Geometry of Omnidirectional Cameras, PhD Thesis," *TR No. CTU-CMP-2004-07, Phd Thesis, Center for Machine Perception, Czech Technical University in Prague, 2004*.

- [90] M. Lhuillier, "Reconstruction 3D automatique avec une camera catadioptrique," *RFIA 2006*.
- [91] N. Goncalves et H. Araujo, "Estimating Parameters of Non-Central Catadioptric Systems Using Bundle Adjustment," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [92] C. Cauchois, E. Brassart, L. Delahoche, et A. Clerentin, "3D Localization with Conical Vision," *In the Fourth Workshop on Omnidirectional Vision, 2003*.
- [93] C. Pinciroli, A. Bonarini, et M. Matteucci, "Robust Detection of 3D Scene Horizontal and Vertical Lines in Conical Catadioptric Sensors," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [94] S. Ramalingam, P. Sturm, et S. K. Lodha, "Theory and Calibration for Axial Cameras," *Proceedings of the Asian Conference on Computer Vision, Hyderabad, India, Volume I, pp 704-713, jan 2006*.
- [95] D. G. Aliaga, "Accurate Catadioptric Calibration for Real-time Pose Estimation in Room-size Environments," *ICCV 2001*.
- [96] G. Jang, S. Kim, et I. Kweon, "Single Camera Catadioptric Stereo System," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [97] S.-S. Lin et R. Bajcsy, "Single Cone Mirror Omni-Directional Stereo," *Technical Report MS-CIS-01-03, 2001. Philadelphia, PA, USA, Computer and Information Science Department, University of Pennsylvania*.
- [98] S.-S. Lin et R. Bajcsy, "High Resolution Catadioptric Omni-Directional Stereo Sensor for Robot Vision," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September 14-19, 2003*.
- [99] J. Fabrizio et J. Devars, "Estimation numérique des courbes épipolaires pour les capteurs omnidirectionnels," *Revue Traitement du Signal (TS05), Special Issue on Omnidirectional Vision, Volume 22, Number 5, pages 527-535, December 2005*.
- [100] S. R. Thirthala et M. Pollefeys, "Multi-view geometry of 1D radial cameras and its application to omnidirectional camera calibration," *ICCV 2005*.
- [101] S. Thirthala et M. Pollefeys, "The radial trifocal tensor : A tool for calibrating the radial distortion of wide-angle cameras," *In Proc CVPR-2005, June 2005*.
- [102] J.-P. Tardif et P. Sturm, "Calibration of Cameras with Radially Symmetric Distortion," *OMNIVIS - 6th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras, Beijing, China, 44-51, 2005*.
- [103] J.-P. Tardif, P. Sturm, et S. Roy, "Self-Calibration of a General Radially Symmetric Distortion Model," *ECCV - 9th European Conference on Computer Vision, Graz, Austria, Vol. 4, 186-199, 2006*.
- [104] S. Ramalingam, P. Sturm, et S. K. Lodha, "Towards Complete Generic Camera Calibration," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, California, Volume 1, page 1093-1098 - june 2005*.

- [105] P. Sturm et S. Ramalingam, "A Generic Calibration Concept - Theory and Algorithms," *Rapport Technique INRIA N°5058, December 2003*.
- [106] Z. Zhang, "A Flexible New Technique for Camera Calibration," *Technical report March 25, 1999*.
- [107] F. Huang, S.-K. Wei, et R. Klette, "Comparative Studies of Line-based Panoramic Camera Calibration," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003*.
- [108] M. Fiala et C. Shu, "Fully Automatic Camera Calibration Using Self-Identifying Calibration Targets," *published as NRC/ERB-1130. November 2005. 26 pages. NRC 48306*.
- [109] C. Geyer et K. Daniilidis, "Properties of the Catadioptric Fundamental Matrix," *ECCV 2002*.
- [110] C. Geyer et K. Daniilidis, "Structure and Motion from Uncalibrated Catadioptric Views," *IEEE Computer Society's Computer Vision and Pattern Recognition (CVPR 2001), Hawaii, Dec 2001*.
- [111] P. Sturm, "Mixing Catadioptric and Perspective Cameras," *OMNIVIS - Workshop on Omnidirectional Vision, 2002*.
- [112] C. Geyer et K. Daniilidis, "Catadioptric Camera Calibration," *ICCV 1999*.
- [113] J. P. Barreto et H. Araujo, "Geometric Properties of Central Catadioptric Line Images," *PAMI August 2005 (Vol. 27, No. 8), pp. 1327-1333*.
- [114] X. Ying et Z. Hu, "Catadioptric Camera Calibration Using Geometric Invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 26, NO. 10, OCTOBER 2004*.
- [115] F. Devernay et O. Faugeras, "Automatic calibration and removal of distortion from scene of structured environments," *in SPIE Medical Imaging : Image Processing, vol. 2567, San Diego, CA, July 1995*.
- [116] H. Zhang, G. Zhang, et K.-Y. K. Wong, "Camera Calibration With Spheres : Linear Approaches," *ICIP 2005*.
- [117] X. Ying et H. Zha, "Linear Catadioptric Camera Calibration from Sphere Images," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [118] T. Mashita, Y. Iwai, et M. Yachida, "Calibration Method for Misaligned Catadioptric Camera," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005*.
- [119] C. Sun et S. Peleg, "Fast Panoramic Stereo Matching Using Cylindrical Maximum Surfaces," *IEEE Transactions on Systems, Man, and Cybernetics, vol. 34, no. 1, February 2004*.
- [120] J. Lobo et J. Dias, "Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference," *PAMI, December 2003 (Vol. 25, No. 12), pp. 1597-1608*.

- [121] T. Schon et F. Gustafsson, "Integrated Navigation of Cameras for Augmented Reality," *IFAC 2005*.
- [122] H. A. Information, "3-AXIS COMPASS SENSOR SET," *Honeywell*.
- [123] E. Shin, "A Quaternion-based unscented Kalman filter for the integration of GPS and MEMS," *ION GNSS 2004*.
- [124] G. Welch et G. Bishop, "An Introduction to the Kalman Filter," <http://www.cs.unc.edu/welch/kalman/index.html>.
- [125] J.-Y. Bouguet, "Complete Camera Calibration Toolbox for Matlab," <http://www.vision.caltech.edu/bouguetj/>.
- [126] P. F. M. Lauchlanand et A. Jaenicke, "Image Mosaicing using Sequential Bundle Adjustment," *BMVC 2000*.
- [127] P. F. McLauchlan et A. Jaenicke, "Image Mosaicing using Sequential Bundle Adjustment," *BMVC 2000*.
- [128] M. Fischler et R. Bolles, "Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of the ACM, Vol 24, pp 381-395, 1981*.
- [129] B. Triggs, P. McLauchlan, R. Hartley, et A. Fitzgibbon, "Bundle Adjustment A Modern Synthesis," *Vision Algorithms : Theory and Practice, Lecture Notes in Computer Science, volume 1883, pages 298-372, Springer-Verlag, 2000*.
- [130] C. Slama, "Manual of Photogrammetry," *American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980*.
- [131] M. Pollefeys, "Visual 3D Modeling from Images," *Tutorial Notes*.
- [132] A. Mihal, "enblend, Image Blending with Multiresolution Splines," <http://enblend.sourceforge.net/>.
- [133] J. P. Barreto et H. Araujo, "Direct Least Square Fitting of Paracatadioptric Line Images," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003*.
- [134] R. Fruhwirth, A. Strandlie, W. Waltenberger, et J. Wroldsen, "A review of fast circle and helix fitting," *Nuclear Instruments and Methods in Physics Research A 502 (2003) 705-707*.
- [135] P. Vasseur et E. M. Mouaddib, "Central Catadioptric Line Detection," *BMVC 2004*.
- [136] P. H. and, "Machine Analysis of Bubble Chamber Pictures," *International Conference on High Energy Accelerators and Instrumentation, CERN, 1959*.
- [137] X. Ying et Z. Hu, "Catadioptric Line Features Detection using Hough Transform," *ICPR 2004*.
- [138] J. Barreto et H. Araujo, "Fitting Conics to Paracatadioptric Projections of Lines," *Computer Vision and Image Understanding, Volume 101, Issue 3, 1 March 2006, Pages 151-165*.
- [139] J. P. de Almeida Barreto, "General Central Projection Systems ; Modeling, Calibration and Visual Servoing," *Phd Thesis, University of Coimbra October 2003*.

- [140] X. Ying et H. Zha, "Simultaneously Calibrating Catadioptric Camera and Detecting Line Features using Hough Transform," *International Conference on Intelligent Robots and Systems 2005*.
- [141] L. Quan et R. Mohr, "Determining perspective structures using hierarchical Hough transform," *Pattern Recognition Letters*, 9 :279-286 (1989).
- [142] J.-J. Gonzalez-Barbosa et S. Lacroix, "Un algorithme rapide de stéréo vision panoramique dense," *rapport interne LAAS-CNRS*, 2003.
- [143] C. Mei et P. Rives, "Calibrage non biaisé d'un capteur central catadioptrique," *RFIA 2006*.
- [144] Z. Zhang, "Parameter Estimation Techniques : A Tutorial with Application to Conic Fitting," *Rapport N° 2676 Octobre 1995*.
- [145] B. Vandepoortaele et P. Gurdjos, "Paracatadioptric camera calibration benchmark for Matlab," <http://www.enseiht.fr/bvdp/calib/index.htm>.
- [146] B. Vandepoortaele, M. Cattoen, P. Marthon, et P. Gurdjos, "A New Linear Calibration Method for Paracatadioptric Cameras," *ICPR 2006 : 18th International Conference on Pattern Recognition*.
- [147] B. Vandepoortaele, M. Cattoen, et P. Marthon, "A Fast Detector of Line Images Acquired by an Uncalibrated Paracatadioptric Camera," *ICPR 2006 : 18th International Conference on Pattern Recognition*.
- [148] V. Caglioti et S. Gasparini, "Localization of 3D lines from single images using off-axis catadioptric cameras," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005)*, October, 2005.
- [149] M. Fiala et A. Basu, "Panoramic Stereo Reconstruction Using Non-SVP Optics," *Proceedings. 16th International Conference on Pattern Recognition, 2002. Volume 4, 2002 Page(s) :27 - 30 vol.4*.
- [150] G. Glaeser, "reflections on spheres and cylinders of revolution," *Journal for Geometry and Graphics*, Vol. 3, No. 2, pp. 121-139, 1999.
- [151] M. Berger, "Seules les quadriques admettent des caustiques," *Bulletin de la S.M.F.*, tome 123, n°1, 1995, pp. 107-116.
- [152] J. Semple et G. Kneebone, "Algebraic Projective Geometry," *Oxford Classic Texts in physical sciences*, 1952, réédité en 1998.
- [153] W. H. Beyer, "CRC Standard Mathematical Tables," 28th ed. Boca Raton, FL : CRC Press, pp. 210-211, 1987.
- [154] S. Lazard, L. M. Peñaranda, et S. Petitjean, "Intersecting Quadrics : An Efficient and Exact Implementation," *Proc. 20th ACM Symp. on Comput. Geom.*, pp. 419-428, 2004.
- [155] S. Fortune, "Polynomial Root Finding Using Iterated Eigenvalue Computation," *International Conference on Symbolic and Algebraic Computation*, 2001.
- [156] L. L. Schumaker, "Spline Functions : Basic Theory," *Wiley Interscience*, 1980, 553 pp. Reprinted by Krieger, Malabar, Fla, 1993.

- [157] C. de Boor, "A practical guide to splines," *volume 27 of Applied Mathematical Sciences. Springer-Verlag, New York, revised edition, 2001.*
- [158] D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *SIAM J. Appl. Math. 11, 431-441, 1963.*
- [159] R. Koch, M. Pollefeys, et L. J. V. Gool, "Robust Calibration and 3D Geometric Modeling from Large Collections of Uncalibrated Images," *DAGM-Symposium, pp 413-420, 1999.*
- [160] N. Snavely, S. M. Seitz, et R. Szeliski, "Photo Tourism : Exploring Photo Collections in 3D," *Proc. ACM SIGGRAPH 2006.*
- [161] H. Bakstein et T. Pajdla, "3D Reconstruction from 360x360 Mosaics," *IEEE Computer Society's Computer Vision and Pattern Recognition (CVPR 2001) Hawaii, Dec 2001.*
- [162] H.-Y. Shum, A. Kalai, et S. M. Seitz, "Omnivergent Stereo," *7th IEEE Int. Conference on Computer Vision, Sept. 1999.*
- [163] H. Li et R. Hartley, "A Non-iterative Method for Correcting Lens Distortion from Nine Point Correspondences," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005), October, 2005.*
- [164] Z. Zhang, "Determining the Epipolar Geometry and its Uncertainty : A Review," *Rapport N° 2927, Juillet 1996.*
- [165] T. Svoboda, "Central Panoramic Cameras : Design, Geometry, Egomotion," *thèse de doctorat, Faculty of Electrical Engineering of Czech Technical University, septembre 1999.*
- [166] A. Torii, A. Sugimoto, et A. Imiya, "Mathematics of a Multiple Omni-Directional System," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003.*
- [167] Z. Zhang, "On the Epipolar Geometry Between Two Images With Lens Distortion," *Proceedings of the 13th International Conference on Pattern Recognition, 1996. Volume 1, 25-29 Aug. 1996 Page(s) :407 - 411 vol.1.*
- [168] C. Geyer et K. Daniilidis, "Conformal Rectification of Omnidirectional Stereo Pairs," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003.*
- [169] R. Bunschoten et B. Krose, "Robust Scene Reconstruction from an Omnidirectional Vision System," *IEEE Transactions on Robotics and Automation, 2002.*
- [170] R. Bunschoten et B. Krose, "3-D Scene Reconstruction from Multiple Panoramic Images," *In 9th International Symposium on Intelligent Robotic Systems, pages 199-205, 2001.*
- [171] R. Bunschoten et B. Krose, "3-D Scene Reconstruction from Cylindrical Panoramic Images," *Robotics and Autonomous Systems, Volume 41, Number 2, 30 November 2002, pp. 111-118(8).*
- [172] T. Svoboda et T. Pajdla, "Matching in Catadioptric Images with Appropriate Windows and Outliers Removal," *Proc. of the 9th International Conference on Computer*

- Analysis of Images and Patterns*. Springer Verlag. Warsaw, Poland, 5–7 September, 2001.
- [173] D. Scharstein et R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” *IJCV* 47 :7-42, April-June 2002.
- [174] S. Chambon, “Mise en correspondance stéréoscopique d’images couleur en présence d’occultations,” *Thèse de doctorat*, décembre 2005.
- [175] S. Chambon et A. Crouzil, “Dense matching using correlation : new measures that are robust near occlusions,” *In British Machine Vision Conference - BMVC 2003, volume 1, pages 143-152, Norwich, United Kingdom, September 2003.*
- [176] S. Chambon et A. Crouzil, “Color stereo matching using correlation measures,” *In Complex Systems Intelligence and Modern Technological Applications - CSIMTA 2004, p. 520-525, Cherbourg, France, September 2004 .*
- [177] M. A. Drouin, M. Trudeau, et S. Roy, “Fast Multiple-baseline Stereo with Occlusion,” *3DIM 2005.*
- [178] M. Okutomi et T. Kanade, “A Multiple-Baseline Stereo,” *IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 15, No. 4, April, 1993, pp. 353-363.*
- [179] S. hoï Ieng, R. Benosman, et J. Devars, “An Efficient Dynamic Multi-Angular Feature Points Matcher for Catadioptric Views,” *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003.*
- [180] W. Niem et H. Broszio, “Mapping Texture from Multiple Camera Views onto 3D-Object Models for Computer Animation,” *Proceedings of the International Workshop on Stereoscopic and Three Dimensional Imaging, September 1995, Greece.*
- [181] A. Criminisi, P. Perez, et K. Toyama, “Object removal by exemplar-based inpainting,” *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 721-728, June 2003.*
- [182] P. Doubek et T. Svoboda, “Reliable 3D Reconstruction from a few Catadioptric Images,” *Omnivis Workshop, June 2002, Copenhagen, Denmark.*
- [183] J. Gaspar, E. Grossmann, et J. Santos-Victor, “Interactive Reconstruction from an Omnidirectional Image,” *VisLab-TR 02/2001 - 9th International Symposium on Intelligent Robotic Systems - SIRS2001 - Toulouse, France, July 2001.*
- [184] P. Sturm, “A Method for 3D Reconstruction of Piecewise Planar Objects from Single Panoramic Images,” *OMNIVIS - IEEE Workshop on Omnidirectional Vision, Hilton Head Island, USA, 119-126, 2000.*
- [185] H.-Y. Shum et S. B. Kang, “A Review of Image-based Rendering Techniques,” *IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000, pp. 2-13, Perth, June 2000.*
- [186] H. Bakstein et T. Pajdla, “Visual Fidelity of Image Based Rendering,” *Computer Vision Winter Workshop 2004.*
- [187] H. Bakstein et T. Pajdla, “Omnidirectional Image-based Rendering,” *Computer Vision Winter Workshop 2006.*

- [188] H. Backstein et T. Pajdla, "Ray space volume of omnidirectional 180*360° images," *Proceedings of the 8th Computer Vision Winter Workshop. Computer Vision-CVWW'03, Prague, January 2003*.
- [189] H. Bakstein et T. Pajdla, "Rendering novel views from a set of omnidirectional mosaic images," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003), June, 2003*.
- [190] H. Ishiguro, K. C. Ng, R. Capella, et M. M. Trivedi, "Omnidirectional image-based modeling : three approaches to approximated plenoptic representations," *Machine Vision and Applications (2003) 14 : pp94-102, Springer-Verlag 2003*.
- [191] K. C. Ng, M. Trivedi, et H. Ishiguro, "Generalized Multiple Baseline Stereo and Direct Virtual View Synthesis Using Range-Space Search, Match, and Render," *International Journal of Computer Vision, Special Issue on Multicamera Stereo, Volume 47, Numbers 1/2/3. April-June 2002 pp. 131-147*.
- [192] D. G. Aliaga, T. Funkhouser, D. Yanovsky, et I. Carlbom, "Sea of Images : A Dense Sampling Approach for Rendering Large Indoor Environments," *Computer Graphics and Applications November/December 2003 (Vol. 23, No. 6) pp. 22-30*.
- [193] D. G. Aliaga, D. Yanovsky, T. Funkhouser, et I. Carlbom, "Interactive Image-Based Rendering Using Feature Globalization," *Proceedings of the 2003 symposium on Interactive 3D graphics, Session 7, pp : 163-170, 2003*.
- [194] H. Shum, S. Chan, et S. Kang, "Image-Based Rendering," *Ed Springer Verlag, 2006*.
- [195] N. Komodakis, G. Pagonis, et G. Tziritas, "Interactive walkthroughs using morphable 3D mosaics," *3DPVT 2004*.
- [196] E. Vincent et R. Laganiere, "Detecting Planar Homographies in an Image Pair," *ISPA 2001*.
- [197] E. Vincent, "On feature point matching, in the calibrated and uncalibrated contexts, between widely and narrowly separated images," *Thèse de doctorat, janvier 2004*.
- [198] A. Bartoli, "Piecewise Planar Segmentation for Automatic Scene Modeling," *CVPR'01 - IEEE International Conference on Computer Vision and Pattern Recognition, Hawaii, PP. 283-289, VOL. 2, DECEMBER 2001*.
- [199] J. Alon et S. Sclaroff, "An Integrated Approach for Segmentation and Estimation of Planar Structures," *Boston University Computer Science Technical Report No.2000-23, Dec.2000*.
- [200] P. Sturm et S. J. Maybank, "A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images," *BMVC 1999*.
- [201] M. Wilczkowiak, E. Boyer, et P. Sturm, "Camera Calibration and 3D Reconstruction from Single Images Using Parallelepipeds," *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, Volume 1, page 142-148 - Jul 2001*.
- [202] R.Orghidan, E.M.Mouaddib, et J.Salvi, "Systeme de vision 3D omnidirectionnelle," *Congrès jeunes chercheurs en Vision par ordinateur, ORASIS'2005 Clermont-Ferrand, France, Mai 2005*.

- [203] R. Orghidan, J. Salvi, et E. M. Mouaddib, "Calibration of a Structured Light-Based Stereo Catadioptric Sensor," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2003)*, June, 2003.
- [204] R. Orghidan, E. M. Mouaddib, et J. Salvi, "Omnidirectional Depth Computation from a Single Image," *IEEE International Conference on Robotics and Automation, ICRA2005, Barcelona, Spain, April 18-22, 2005*.
- [205] C. Mei et P. Rives, "Calibration between a Central Catadioptric Camera and a Laser Range Finder for Robotic Applications," *ICRA 2006*.
- [206] D. Roger et N. Holzschuch, "Réflexions spéculaires en temps réel sur des surfaces lisses," *Journées AFIG 2005*.
- [207] J. van Baar, T. Willwacher, S. Rao, et R. Raskar, "Seamless Multi-Projector Display on Curved Screens," *Proceedings of the workshop on Virtual environments, ACM International Conference Proceeding Series, vol. 39, 2003*.
- [208] R. Raskar, J. van Baar, et T. Willwacher, "Quadric Transfer for Immersive Curved Display," *Eurographics, August 2004*.
- [209] C. Silpa-Anan et R. Hartley, "Visual localization and loop-back detection with a high resolution omnidirectional camera," *IEEE Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 2005)*, October, 2005.
- [210] M. Segal et K. Akeley, "The OpenGL Graphics System : A Specification (Version 1.3)," *Editor : Jon Leech*.
- [211] J.-M. Hasenfratz, "Lancer de Faisceaux en Synthèse d'Images," *Thèse de doctorat, Université de Limoges, 1998*.
- [212] J. Amanatides, "Ray Tracing with Cones," *Computer Graphics, (Proceedings of SIGGRAPH'84)*, July 1984, pp. 129-135.

Contributions à la vision omnidirectionnelle: Étude, conception et étalonnage de capteurs pour l'acquisition d'images et la modélisation 3D

Bertrand VANDEPORTAELE

Résumé:

Cette thèse présente des contributions au domaine de la vision omnidirectionnelle. Les travaux ont été effectués dans le contexte de la localisation par l'image, dans lequel une caméra est utilisée comme appareil portatif pour localiser et guider une personne. Des apports théoriques sont proposés et deux capteurs omnidirectionnels sont développés et décrits pour l'acquisition respectivement de mosaïques d'images et de modèles 3D d'intérieurs de bâtiments.

Le premier chapitre présente le domaine de l'imagerie à large champ de vision et les différents moyens d'obtenir un champ visuel important. Quelques modèles géométriques sont décrits pour différentes caméras et les spécificités pour l'étalonnage sont discutées. Les caméras sont séparées en deux familles, selon qu'elles possèdent un centre de projection unique (caméras centrales) ou non (caméras non centrales). Finalement, la reprojection des images entre différents modèles de caméra est présentée.

Le second chapitre présente une nouvelle approche de la vision omnidirectionnelle utilisant une caméra standard dotée d'un capteur électronique d'orientation pour l'obtention automatique de mosaïques d'images orientées. Ces images possèdent une résolution plus importante que celles provenant d'une caméra catadioptrique. Le champ de vision couvert par notre caméra est omnidirectionnel et son encombrement et son coût sont moins élevés que ceux d'un ensemble de plusieurs caméras. Elle permet la création de mosaïques en temps réel, qui peuvent être affinées par la méthode de l'ajustement de faisceaux.

Le troisième chapitre traite de la caméra paracatadioptrique, qui est centrale. Une méthode est proposée pour la détection des projections de droites dans l'image, dans le cas où la caméra est étalonnée, puis est utilisée pour estimer l'orientation de la caméra par rapport à la verticale. Une nouvelle méthode d'étalonnage géométrique utilisant des images de droites est développée, fournissant des résultats plus précis et répétables que les méthodes existantes et utilisant une meilleure paramétrisation des images de droites. Cette méthode est robuste et permet l'étalonnage automatique à partir d'une image contenant des projections de droites.

Le quatrième chapitre propose une méthode pour calculer l'image (ou les images) d'un point observé par une caméra catadioptrique non centrale avec un miroir en forme de quadrique. Cette méthode est validée pour différentes formes de quadriques puis est utilisée pour calculer la projection de droites (ou segments de droites), la reprojection de texture et enfin permettre l'étalonnage géométrique de la caméra.

Le dernier chapitre décrit un nouveau capteur, doté d'un champ de vision omnidirectionnel, utilisé pour acquérir des modèles 3D d'intérieurs de bâtiments à partir d'une caméra paracatadioptrique. Le développement et l'étalonnage du capteur sont présentés, et les premiers résultats de reconstruction 3D du modèle de l'environnement sont montrés, pour différents niveaux d'automatisation.

Mots clefs: Vision omnidirectionnelle, caméras catadioptriques, mosaïquage d'image, capteurs, modélisation 3D.

Abstract:

This thesis presents some contributions to the omnidirectional vision field. It has been achieved in the context of the image based localization for persons. In this application, a camera is used as a hand-held device to localize and guide a person in a known place using image analysis. Theoretical contributions are presented and two omnidirectional sensors are developed and described, respectively for the acquisition of image mosaics and 3D models of building interiors.

The first chapter presents the omnidirectional vision domain and describes the existing different approaches to increase the field of view of a camera. The main geometric camera models are described and some specific facts about camera calibration are exhibited. The cameras are separated in two different groups: the central cameras that have a single viewpoint and the non central cameras. Finally, the projection of images between different camera models is presented.

The second chapter presents a new kind of omnidirectional camera. Using a home made electronic sensor that provides the orientation of a standard camera to which it is connected, an omnidirectional oriented image mosaic is created. This image has a higher resolution than one acquired by a catadioptric camera and the proposed camera is less voluminous and expensive than a group of many standard cameras that provides the same field of view. The mosaics are computed in real-time, and can be improved using the bundle adjustment algorithm.

The third chapter deals with the paracatadioptric camera, which is central. A specific method for the line images detection is presented, in the case where the camera parameters are known. It is then used to estimate the orientation of the camera related to the vertical. A new calibration method using a better parameterization of the line images is derived, providing more precise and repeatable results than previous methods. This method is robust and allows the automatic calibration from an image containing line images.

The fourth chapter derives an innovative and efficient approach to obtain the projection(s) of a 3D point observed by a non central catadioptric camera with a quadric mirror. This method is validated for different quadric shapes. It is used to compute the images of complete or segments of lines, to retrieve a texture from its image and for calibration purposes.

The last chapter describes a new omnidirectional 3D sensor using a paracatadioptric camera to acquire 3D models of building interiors. The development and methods for the calibration of the sensor are shown. Preliminary results of 3D model acquisition are also presented.

Keywords: Omnidirectional Vision, Catadioptric Cameras, Image Mosaicing, Sensors, 3D model acquisition.