



**HAL**  
open science

# Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions.

Rémi Boutteau

► **To cite this version:**

Rémi Boutteau. Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions.. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université de Rouen, 2010. Français. NNT: . tel-00711922v2

**HAL Id: tel-00711922**

**<https://theses.hal.science/tel-00711922v2>**

Submitted on 24 Nov 2017

**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.

pour obtenir le titre de  
**DOCTEUR EN SCIENCES**

Spécialité

**Génie Informatique, Automatique et Traitement du Signal**

de l'Université de ROUEN

préparée à  
l'Institut de Recherche en Systèmes Électroniques Embarqués  
**IRSEEM – EA 4353**

présentée et soutenue publiquement par

**Rémi BOUTTEAU**

**Reconstruction tridimensionnelle de  
l'environnement d'un robot mobile, à partir  
d'informations de vision omnidirectionnelle,  
pour la préparation d'interventions.**

Le 19 avril 2010 devant le jury composé de :

Président :

M. **François CHAUMETTE**, Directeur de Recherche, IRISA / INRIA Rennes

Rapporteurs :

M. **Peter STURM**, Directeur de Recherche, INRIA Grenoble - Rhône-Alpes

M. **Didier AUBERT**, Directeur Adjoint du LIVIC, INRETS

Examineurs :

M. **Richard GRISEL**, Professeur des Universités, Université de Rouen

M. **Bélahcène MAZARI**, Directeur de l'IRSEEM, Directeur de Thèse

M. **Xavier SAVATIER**, Enseignant-chercheur IRSEEM / ESIGELEC, Co-encadrant

M. **Noël MOISAN**, Directeur technique, IPSIS

Invité :

M. **Jean-Yves ERTAUD**, Enseignant-chercheur IRSEEM / ESIGELEC, Co-encadrant

# Remerciements

Je tiens tout d'abord à remercier mon directeur de thèse, **Bélahcène Mazari**, pour m'avoir fait confiance en acceptant de diriger cette thèse, et, en tant que directeur de l'IRSEEM, pour m'avoir accueilli dans son laboratoire.

Je suis particulièrement reconnaissant à **Jean-Yves Ertaud** et **Xavier Savatier**, d'avoir encadré cette thèse. Ils ont su me conseiller, me soutenir, et m'encourager durant ces trois années.

Je remercie sincèrement **Peter Sturm** et **Didier Aubert** d'avoir accepté d'être rapporteurs, malgré la distance et leur emploi du temps chargé. Leurs remarques ont contribué à l'amélioration de ce manuscrit, et ils ont su soulever des perspectives intéressantes à mes travaux.

Mes remerciements vont également à **François Chaumette** de m'avoir fait l'honneur de présider ce jury, et à **Richard Grisel** d'avoir examiné mes travaux.

Un grand merci à tous les membres de l'équipe IIS (Instrumentation Informatique et Systèmes) pour nos échanges fructueux : **Nicolas Ragot**, **Pierre Merriaux**, **Romain Rossi**, **Jean-François Layerle**, **Fengchun Dong**, **Yohan Dupuis**, et à tous les stagiaires et post-doctorants qui ont passé quelques temps au sein de notre équipe.

Je veux exprimer ma gratitude à **Noël Moisan**, directeur technique de la société IPSIS, sans qui cette thèse n'aurait pas vu le jour puisqu'il s'agit d'une thèse CIFRE. Je remercie également mes chefs de projets, **Olivier Adam** et **Jonathan Letessier** qui ont su m'apporter leur professionnalisme dans les domaines de la gestion de projet et de la qualité.

J'adresse mes remerciements aux membres de la DGA qui sont à l'origine de l'étude et qui ont suivi ce projet : **Gaël Désilles**, **Georges Lamy Au Rousseau**, **Benjamin De Coutard**, **Sébastien Le Bouter** et **Francis Celeste**.

Je remercie également **Jean-Jacques Delarue**, **Christian Dureisseix** et **Habib Baldé** de m'avoir fait confiance en me laissant intégrer l'équipe pédagogique de l'ESIGELEC. Ce fut une expérience très enrichissante qui m'a permis de découvrir le métier d'enseignant.

Je remercie tous les professeurs qui ont contribué à mon apprentissage, et qui ont su me donner le goût des sciences et de la recherche. Je pense en particulier à **Alain Cousin**, **Jacques Boonaert**, **Philippe Desodt** et **Stéphane Lecoecuche**.

J'exprime ma reconnaissance à toutes les personnes qui ont contribué à la fastidieuse tâche de relecture et de correction de ce mémoire.

Enfin, je remercie particulièrement ma famille et mes amis, pour leur soutien.

# Table des matières

<b>Remerciements.....</b>	<b>3</b>
<b>Table des figures.....</b>	<b>9</b>
<b>Liste des tableaux.....</b>	<b>13</b>
<b>Notations.....</b>	<b>15</b>
<b>Introduction.....</b>	<b>17</b>
<b>Chapitre 1 Conception d'un capteur d'acquisition 3D.....</b>	<b>21</b>
1.1.Introduction.....	23
1.2.Techniques d'acquisition 3D sans contact.....	24
1.2.1.Systèmes actifs.....	24
1.2.1.1.Télémétrie.....	24
1.2.1.2.Triangulation active.....	27
1.2.2.Systèmes passifs.....	29
1.2.2.1.Shape-From-Shading.....	29
1.2.2.2.Shape-From-Photometric Stereo.....	30
1.2.2.3.Shape-From-Scattering.....	30
1.2.2.4.Shape-From-Texture.....	31
1.2.2.5.Shape-From-Contour.....	31
1.2.2.6.Shape-From-Focus/Defocus.....	32
1.2.2.7.Shape-From-Silhouettes.....	32
1.2.2.8.Shape-From-Consistency.....	33
1.2.2.9.Shape-From-Motion.....	36
1.2.2.10.Shape-From-Stereo.....	36
1.2.3.Synthèse sur les techniques d'acquisition 3D.....	37
1.3.Les systèmes de vision.....	38
1.3.1.Vision classique.....	38
1.3.1.1.Le sténopé.....	38
1.3.1.2.La caméra réelle.....	41
1.3.1.3.Les défauts de la caméra réelle.....	42
1.3.2.Vision omnidirectionnelle.....	44
1.3.2.1.Fusion de plusieurs images.....	45
1.3.2.2.Utilisation d'un objectif grand angle.....	47
1.3.2.3.Utilisation d'un capteur catadioptrique.....	48
1.4.Conception d'un système de vision 3D.....	55
1.4.1.Introduction.....	55
1.4.2.Choix du capteur catadioptrique.....	55
1.4.2.1.Choix des miroirs.....	55
1.4.2.2.Choix des caméras et des optiques.....	58
1.4.2.3.Respect de la contrainte du point de vue unique.....	59
1.4.3.Étude de la configuration des deux capteurs.....	60
1.4.3.1.Géométrie épipolaire.....	61
1.4.3.2.Étude de la résolution du système stéréoscopique catadioptrique.....	64
1.4.3.3.Vue d'ensemble du système.....	65
1.5.Conclusion.....	66
<b>Chapitre 2 Calibrage du système de stéréovision omnidirectionnelle.....</b>	<b>69</b>
2.1.Introduction.....	71



2.2.Modélisation d'un système de stéréovision omnidirectionnelle.....	72
2.2.1.Modélisation d'un système de vision omnidirectionnelle.....	73
2.2.1.1.Modèles ad hoc.....	73
2.2.1.2.Modèles polynomiaux.....	74
2.2.1.3.Modèles unifiés.....	75
2.2.2.Modélisation d'un système stéréoscopique.....	79
2.3.Calibrage d'un système de stéréovision omnidirectionnelle.....	80
2.3.1.Calibrage des systèmes de vision.....	80
2.3.2.Calibrage des systèmes stéréoscopiques.....	80
2.4.Application au calibrage du système de stéréovision omnidirectionnelle.....	81
2.4.1.Estimation des paramètres intrinsèques du système.....	82
2.4.1.1.Initialisation des paramètres.....	83
2.4.1.2.Minimisation de l'erreur de reprojection.....	89
2.4.2.Estimation de la pose relative des deux capteurs.....	90
2.4.2.1.Principe de la méthode.....	91
2.4.2.2.Méthode d'estimation de la pose relative.....	91
2.5.Résultats expérimentaux.....	92
2.5.1.Évaluation de l'estimation des paramètres intrinsèques.....	93
2.5.1.1.Évaluation sur des données de simulation.....	93
2.5.1.2.Évaluation sur des données réelles.....	94
2.5.1.3.Évaluation qualitative.....	94
2.5.2.Évaluation de l'estimation de la pose relative.....	95
2.5.2.1.Évaluation sur des données de simulation.....	95
2.5.2.2.Évaluation sur des donnée réelles.....	96
2.5.2.3.Évaluation qualitative.....	96
2.6.Conclusion et perspectives.....	97
<b>Chapitre 3 Reconstruction 3D statique.....</b>	<b>99</b>
3.1.Introduction.....	101
3.2.Reconstruction 3D dense.....	101
3.2.1.Les méthodes locales.....	102
3.2.1.1.Mesures de corrélation.....	103
3.2.1.2.Utilisation de la couleur.....	104
3.2.1.3.Méthodes de rejet des mauvais appariements.....	105
3.2.1.4.Résultats expérimentaux.....	106
3.2.2.Prise en compte des contraintes globales.....	108
3.2.2.1.Définition des contraintes globales.....	109
3.2.2.2.La programmation dynamique.....	111
3.2.2.3.Résultats expérimentaux.....	118
3.2.3.Triangulation de points.....	120
3.2.3.1.Méthode du point milieu.....	120
3.2.3.2.Résultats expérimentaux.....	121
3.3.Reconstruction 3D de droites.....	123
3.3.1.Détection de droites dans les images omnidirectionnelles.....	123
3.3.2.Triangulation de droites.....	127
3.4.Conclusion et perspectives.....	129
<b>Chapitre 4 Reconstruction 3D avec déplacement.....</b>	<b>131</b>
4.1.Introduction.....	133
4.2.Initialisation du déplacement et des points 3D.....	134
4.2.1.Mise en correspondance des images.....	135
4.2.2.Estimation de la géométrie épipolaire.....	136
4.2.2.1.Géométrie épipolaire des capteurs centraux.....	136
4.2.2.2.Algorithme des huit points pour les capteurs centraux.....	138

4.2.2.3.Renforcement des contraintes de la matrice essentielle.....	139
4.2.2.4.Estimation robuste de la matrice essentielle.....	140
4.2.3.Détermination du déplacement.....	143
4.2.3.1.Décomposition de la matrice essentielle.....	143
4.2.3.2.Détermination du facteur d'échelle.....	145
4.3.Ajustement de faisceaux.....	146
4.3.1.Fonctionnelle à minimiser.....	146
4.3.2.Fonction de projection stéréoscopique.....	147
4.3.3.Structure de la jacobienne.....	148
4.4.Résultats expérimentaux.....	149
4.4.1.Phase d'initialisation.....	149
4.4.1.1.Mise en correspondance des images.....	149
4.4.1.2.Estimation de la géométrie épipolaire .....	151
4.4.1.3.Détermination du déplacement.....	151
4.4.2.Ajustement de faisceaux.....	152
4.4.2.1.Estimation des déplacements.....	152
4.4.2.2.Estimation de la trajectoire.....	153
4.4.2.3.Recalage des cartes locales.....	154
4.5.Conclusion et perspectives.....	156
<b>Conclusion et perspectives.....</b>	<b>159</b>
Conclusion.....	159
Perspectives.....	160
<b>Bibliographie.....</b>	<b>161</b>
<b>Annexes.....</b>	<b>169</b>

# Table des figures

Figure 1: Objectif de l'étude.....	18
Figure 2: Principe de la télémétrie.....	25
Figure 3: Principe du radar.....	25
Figure 4: Radar Autocruise ACC.....	26
Figure 5: Lidar Sick LD-PDS.....	26
Figure 6: Télémètre à ultrasons.....	27
Figure 7: Principe de la triangulation active.....	27
Figure 8: Triangulation laser point.....	28
Figure 9: Profilométrie.....	28
Figure 10: Projection de lumière structurée.....	29
Figure 11: Reconstruction 3D d'un visage par Shape-From-Shading.....	30
Figure 12: Reconstruction 3D d'un buste par stéréophotométrie.....	30
Figure 13: Exemple de reconstruction 3D d'un guépard par Shape From Texture.....	31
Figure 14: Application du Shape-From-Contour à la rectification d'images.....	31
Figure 15: Résultat obtenu par Shape-From-Defocus.....	32
Figure 16: Principe du Shape-From-Silhouettes.....	33
Figure 17: Discrétisation de l'espace en volumes élémentaires.....	34
Figure 18: Principe de la photo-consistance.....	34
Figure 19: Exemple de reconstruction 3D par photo-consistance.....	35
Figure 20: Modèles 3D obtenus par SFM.....	36
Figure 21: Principe de la stéréovision.....	37
Figure 22: Reconstruction 3D d'un visage par stéréovision.....	37
Figure 23: Formation d'une image dans un sténopé.....	38
Figure 24: Modèle géométrique du sténopé.....	39
Figure 25: Comparaison entre une caméra perspective (à gauche) et une caméra orthographique (à droite).....	42
Figure 26: Illustration de l'impact d'une faible profondeur de champ.....	42
Figure 27: Aberrations sphériques.....	43
Figure 28: Aberrations chromatiques.....	43
Figure 29: Exemple de distorsions.....	44
Figure 30: Champ de vue des systèmes de vision (représenté en gris).....	44
Figure 31: Capteur cylindrique.....	45
Figure 32: Image acquise par un capteur cylindrique et zoom sur une région.....	46
Figure 33: Principe du système multi-caméras.....	47
Figure 34: Systèmes multi-caméras.....	47
Figure 35: (a) Objectif fish-eye, (b) Image acquise par une caméra fish-eye.....	48
Figure 36: Exemple de capteur catadioptrique.....	48
Figure 37: Miroir plan.....	50
Figure 38: Miroir conique.....	51
Figure 39: Miroir ellipsoïde.....	52
Figure 40: Miroir sphérique.....	52
Figure 41: Miroir hyperboloïde.....	53
Figure 42: Miroir parabolöide.....	54
Figure 43: Encombrement d'un système stéréoscopique utilisant des miroirs parabolöides.....	56
Figure 44: Paramètres d'une hyperbole.....	57

Figure 45: Comparaison de différents champs de vue.....	57
Figure 46: Formation des images avec un miroir hyperboloïde.....	59
Figure 47: Ajustement de la position du miroir pour respecter la contrainte du point de vue unique.....	60
Figure 48: Géométrie épipolaire d'un système stéréoscopique classique.....	61
Figure 49: Géométrie épipolaire d'un système stéréoscopique catadioptrique.....	62
Figure 50: Exemple de courbes épipolaires obtenues par déplacement d'un capteur catadioptrique.....	63
Figure 51: Configuration particulière de capteurs panoramiques simplifiant la géométrie épipolaire.....	63
Figure 52: Paire d'images stéréoscopiques obtenue avec une baseline de 50cm.....	64
Figure 53: Résolution spatiale du capteur de stéréovision omnidirectionnelle.....	65
Figure 54: Système d'acquisition 3D embarqué sur un robot Pioneer.....	66
Figure 55: Interface du logiciel d'acquisition des données.....	66
Figure 56: Configuration du système de stéréovision omnidirectionnelle.....	72
Figure 57: Modèle de projection d'un capteur à base de miroir hyperboloïde.....	74
Figure 58: Principe du modèle polynomial.....	75
Figure 59: Modèle de projection unifié utilisé.....	76
Figure 60: Convention des axes.....	77
Figure 61: Modélisation d'un système stéréoscopique.....	79
Figure 62: Interfaces des logiciels de calibrage développés.....	82
Figure 63: Principe de la méthode de détection de cercle.....	84
Figure 64: Contours détectés par l'algorithme de Canny.....	84
Figure 65: Représentation de l'accumulateur.....	85
Figure 66: Représentation de l'accumulateur après le seuillage.....	86
Figure 67: Résultat de l'extraction automatique du miroir (en vert). .....	86
Figure 68: Exemple de points utilisés pour estimer la focale généralisée (en rouge).....	87
Figure 69: Résultat de l'extraction automatique des points de contrôle (en jaune).....	89
Figure 70: Structure de la jacobienne pour le calibrage.....	90
Figure 71: Points de contrôle extraits dans les images des deux capteurs (croix jaunes).....	92
Figure 72: Extrait des images de synthèse utilisées pour le calibrage.....	93
Figure 73: Représentation de la sphère d'équivalence grâce aux paramètres intrinsèques estimés.....	95
Figure 74: Paires d'images utilisées pour l'estimation de la pose relative.....	95
Figure 75: Extraits de la séquence d'images utilisée pour l'estimation des déplacements.....	96
Figure 76: Courbes épipolaires (en vert) correspondant aux pixels sélectionnés (en rouge).....	97
Figure 77: Principe de la corrélation.....	102
Figure 78: Principe de l'algorithme de Bresenham.....	105
Figure 79: Comparaison d'un appariement en niveaux de gris (a) et d'un appariement en couleur (b).....	107
Figure 80: Comparaison d'un appariement en RGB (a) et d'un appariement en H1H2H3 (b).....	107
Figure 81: Carte de disparité obtenue avec les méthodes locales sans rejet.....	108
Figure 82: Carte de disparité obtenue par les méthodes locales avec rejet.....	108
Figure 83: La contrainte d'unicité.....	110
Figure 84: La contrainte d'ordre.....	111
Figure 85: Extraction des rayons dans les images haute et basse.....	113
Figure 86: Exemple de matrice de recherche et de chemin optimal.....	114
Figure 87: Pseudo-code du calcul des coefficients de la matrice de recherche.....	115
Figure 88: Pseudo-code de la recherche du chemin optimal.....	116
Figure 89: Chemin optimal avec mise en correspondance valides (en vert) et non valides (en rouge).....	117
Figure 90: Structure de la matrice de recherche avec la contrainte de limite de la disparité.....	118
Figure 91: Évolution du temps de calcul par rapport à la limite de la disparité.....	118
Figure 92: Paire d'images stéréoscopique utilisée pour l'évaluation de la programmation dynamique.....	119
Figure 93: Cartes de disparité obtenues.....	119

Figure 94: Mise en valeur des objets de la carte de disparité.....	120
Figure 95: Triangulation par la méthode du point milieu.....	120
Figure 96: Paire d'images stéréoscopiques utilisée pour l'évaluation de la triangulation.....	122
Figure 97: Position des points des cinq mires (croix) comparée à la vérité terrain (lignes bleues et rouges).....	122
Figure 98: Projection d'une droite sur la sphère d'équivalence.....	124
Figure 99: (a) Image omnidirectionnelle - (b) Contours détectés - (c) Chaînes de pixels connexes.....	125
Figure 100: Caractérisation d'une droite omnidirectionnelle par une normale.....	125
Figure 101: Pseudo-code pour l'estimation des droites omnidirectionnelles par la méthode du RANSAC.....	127
Figure 102: Segments de droites omnidirectionnelles détectés.....	127
Figure 103: Triangulation d'une droite.....	128
Figure 104: Images omnidirectionnelles basse et haute utilisées pour la triangulation de droites.....	129
Figure 105: Reconstruction statique d'un couloir.....	129
Figure 106: Géométrie épipolaire des capteurs centraux.....	137
Figure 107: Pseudo-code pour l'estimation robuste de la matrice essentielle.....	141
Figure 108: Calcul de l'erreur dans le cas d'une caméra perspective.....	141
Figure 109: Calcul de l'erreur dans le cas d'un capteur omnidirectionnel.....	142
Figure 110: Test des différentes combinaisons de pose possibles.....	144
Figure 111: Appariements nécessaires pour retrouver le facteur d'échelle.....	145
Figure 112: Transformation rigide entre les deux capteurs du système stéréoscopique.....	147
Figure 113: Structure de la jacobienne pour l'ajustement de faisceaux.....	149
Figure 114: Illustration de la mise en correspondance entre deux instants.....	150
Figure 115: Courbes épipolaires obtenues à partir de la matrice essentielle.....	151
Figure 116: Environnement utilisé pour l'évaluation de l'ajustement de faisceaux.....	152
Figure 117: Erreur commise sur l'estimation des déplacements d'une séquence de 25 images.....	153
Figure 118: Trajectoire estimée par odométrie.....	153
Figure 119: Trajectoire estimée par le lidar.....	154
Figure 120: Trajectoire estimée par ajustement de faisceaux.....	154
Figure 121: Fusion des cartes locales avec les déplacements estimés par le lidar.....	155
Figure 122: Fusion des cartes locales avec les déplacements estimés par odométrie.....	155
Figure 123: Fusion des cartes locales avec les déplacements estimés par ajustement de faisceaux.....	156
Figure 124: Pseudo-code de l'algorithme de Levenberg-Marquardt.....	176
Figure 125: Estimation de l'équation d'une droite par le RANSAC.....	177
Figure 126: Pseudo-code pour l'estimation robuste de la matrice essentielle.....	178
Figure 127: Environnement de la mission 1.....	184
Figure 128: Images omnidirectionnelles de la séquence utilisée pour la mission 1.....	184
Figure 129: Validation qualitative des données de calibrage.....	185
Figure 130: Environnement de la mission 2.....	186
Figure 131: Images omnidirectionnelles de la séquence utilisée pour la mission 2.....	186
Figure 132: Images omnidirectionnelles utilisées pour la reconstruction 3D du cube.....	187
Figure 133: Sélection du cube dans les deux images omnidirectionnelles.....	187
Figure 134: Reconstruction 3D du cube comparé à la vérité terrain (en vert).....	187
Figure 135: Environnement de la mission 3.....	188
Figure 136: Images omnidirectionnelles de la séquence utilisée pour la mission 3.....	189
Figure 137: Environnement de la mission 4.....	189
Figure 138: Images omnidirectionnelles de la séquence utilisée pour la mission 4.....	190
Figure 139: Environnement de la mission 5.....	190
Figure 140: Images omnidirectionnelles de la séquence utilisée pour la mission 5.....	191
Figure 141: Cartes des profondeurs pour deux positions de la mission 5.....	191
Figure 142: Décision sur la possibilité de poser un hélicoptère.....	192



# ***Liste des tableaux***

Tableau 1: Paramètres du miroir hyperboloïde.....	58
Tableau 2: Résultats du calibrage sur des images de synthèse.....	93
Tableau 3: Résultats du calibrage sur des images réelles.....	94
Tableau 4: Résultats de l'estimation de la pose relative sur des images de synthèse.....	95
Tableau 5: Résultats de l'estimation des déplacements.....	96
Tableau 6: Résultats de la triangulation des points des mires.....	122
Tableau 7: Nombre de tirages à effectuer pour différentes proportions d'outliers.....	142
Tableau 8: Résultats de la mise en correspondance sur la paire d'image de la figure 114.....	150
Tableau 9: Déplacements estimés par décomposition de la matrice essentielle.....	152
Tableau 10: Distance estimée du cube par rapport à la vérité terrain.....	188
Tableau 11: Erreur sur le dimensionnement du cube.....	188

# *Notations*

$a$	Scalaire (minuscule, italique)
$\mathbf{v}$	Vecteur $\mathbf{v}$ (minuscule, gras)
$\mathbf{M}$	Matrice $\mathbf{M}$ (majuscule, sans-serif)
$\mathbf{v}^T, \mathbf{M}^T$	Transposée du vecteur $\mathbf{v}$ ou de la matrice $\mathbf{M}$
$\mathbf{M}^{-1}$	Inverse de la matrice $\mathbf{M}$
$\mathbf{I}_n$	Matrice identité de dimension $n \times n$
$\mathbf{P}, \mathbf{p}$	Point 3D $\mathbf{P}$ et son correspondant 2D $\mathbf{p}$ sur le plan image (même caractère, minuscule)
$F$	Fonction (majuscule, italique)
$\mathbf{x} \cdot \mathbf{y}$	Produit scalaire de $\mathbf{x}$ et $\mathbf{y}$
$\mathbf{x} \times \mathbf{y}$	Produit vectoriel de $\mathbf{x}$ et $\mathbf{y}$



# Introduction

L'intérêt pour la robotique mobile n'a de cesse d'augmenter depuis quelques années, notamment pour remplacer l'homme dans des environnements à risque. Ces environnements, dits « hostiles », peuvent être naturels (fonds marins, planètes, zones de catastrophes naturelles) ou liés à l'action humaine (centrales nucléaires, champs de bataille). Pour intervenir dans ce type d'environnements, de nombreux robots ont vu le jour, mais ceux-ci sont généralement télé-pilotés par un opérateur qui reçoit les données nécessaires par l'intermédiaire de capteurs (capteurs à ultrasons, caméras, etc), qui les interprète, et envoie ensuite des ordres au robot en fonction de ces différents éléments. La communication entre le robot et le poste de commande pose alors de réels problèmes, qu'il s'agisse d'une transmission filaire (longueur des câbles, impossibilité de refermer une porte hermétique derrière le robot, *etc*) ou sans fil (temps de propagation des ondes sur de longues distances, présence d'un système de brouillage, *etc*).

La résolution de ces problèmes passe par la conception de robots capables de s'adapter seuls et en temps réel à leur environnement. Le robot doit alors être en mesure d'accomplir deux principales tâches : percevoir son environnement, et pouvoir se localiser et naviguer au sein de celui-ci. Nos travaux concernent le premier point soulevé, à savoir la perception de l'environnement d'un robot mobile sans connaissance *a priori*.

La perception de l'environnement peut être réalisée grâce à de nombreux capteurs, plus ou moins performants en fonction des technologies utilisées. Nos travaux sont centrés sur les systèmes de vision car ils permettent d'obtenir des informations sans aucune émission d'énergie, ce qui les rend plus difficilement détectables. Si une image fournit une grande quantité d'informations sur la scène observée en un temps très court, elle n'est pas suffisante pour reconstruire une scène en trois dimensions. Cette impossibilité est due au processus de formation des images, qui consiste à projeter la scène 3D sur un plan image bidimensionnel, et qui entraîne la perte de l'information de profondeur. Pour contourner ce problème, plusieurs points de vue sont nécessaires. Ceux-ci peuvent être obtenus soit en déplaçant la caméra, soit en ajoutant une deuxième caméra pour former un système stéréoscopique et permettre, par triangulation, de retrouver la notion de profondeur.

Les travaux de thèse présentés dans ce mémoire s'inscrivent dans le cadre d'un marché du LRBA<sup>1</sup> (Laboratoire de Recherches Balistiques et Aérodynamiques) portant sur l'étude d'un moyen de reconstruction tridimensionnelle de l'environnement, à partir d'informations de vision,

---

<sup>1</sup> Centre d'Expertise de la DGA (Délégation Générale pour l'Armement)

pour la préparation d'interventions. Cette étude a été menée conjointement par l'IRSEEM (Institut de Recherche en Systèmes Électroniques Embarqués) et IPSIS (Ingénierie pour Signaux et Systèmes). L'objectif de l'étude, schématisé par la figure 1, est de concevoir un système de vision adapté aux problématiques de la robotique mobile, et de développer des algorithmes permettant au robot d'obtenir un modèle 3D de la scène à partir des images acquises par ce système.

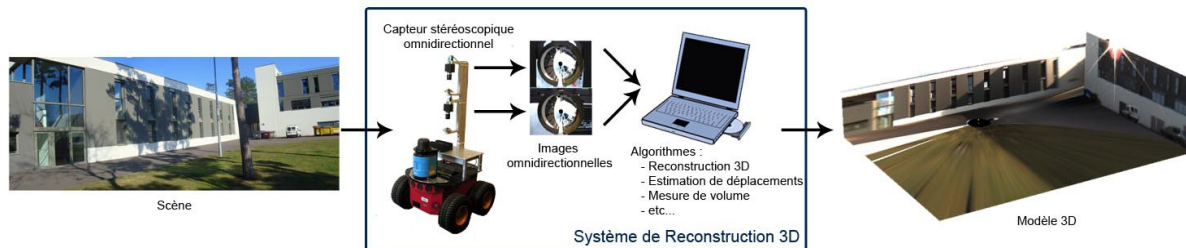


Figure 1: Objectif de l'étude.

La spécification technique de besoin de cette étude était exprimée par six missions opérationnelles<sup>2</sup> ayant un intérêt pour la défense :

- **Préparation de la progression en milieu urbain** : relevé d'architecture le long de la façade d'un bâtiment. La plate forme mobile progressant le long d'une façade devra être capable de repérer les ouvertures et de dimensionner ces dernières (portes, fenêtres, garages, soupiraux).
- **Préparation en de la progression en milieu fermé (bâtiment)** : relever la configuration (portes, fenêtres) de l'intérieur d'un bâtiment le long d'un couloir.
- **Préparation de la progression en milieu naturel** : le porteur suivant un axe (route), le système de vision permet de localiser les obstacles locaux (arbres, mouvements de terrain, construction).
- **Préparation d'intervention d'aide à la mobilité** : estimation du volume d'obstacles placés sur la voie (gravats, véhicules).
- **Préparation d'intervention d'aide au déploiement** : identification de zone d'héliport (évaluation de volume).
- **Surveillance du champ de bataille** : discrimination des obstacles fixes et mobiles (lors d'un déplacement). Estimation des distances et vitesses de déplacements de mobiles.

Dans cette introduction, nous avons présenté le contexte de nos travaux et soulevé les problématiques liées à la reconstruction 3D d'environnements inconnus *a priori* par des systèmes de vision.

<sup>2</sup> Les objectifs détaillés de ces missions, les séquences d'images utilisées, et les résultats obtenus sont disponibles en Annexe E

Le **chapitre 1** dresse un état de l'art des techniques d'acquisition 3D susceptibles de répondre à la problématique de la reconstruction 3D de scènes complexes. La contrainte de fertilité liée au projet nous a conduit à considérer plus particulièrement les systèmes de vision qui sont des systèmes passifs. Parmi les systèmes de vision, nous présentons plus spécifiquement les systèmes de vision omnidirectionnelle qui sont de plus en plus présents en robotique de par leurs nombreux avantages (champ de vue plus important, *etc*). Enfin, nous expliquons dans ce chapitre comment nous avons conçu notre système pour répondre au mieux à notre problématique.

Le **chapitre 2** concerne le calibrage du capteur que nous avons développé, étape nécessaire à tout système de vision dès lors qu'il est destiné à dimensionner son environnement. Calibrer le capteur consiste à trouver la relation mathématique liant les coordonnées des points 3D et leurs projections sur le plan image, ce qui suppose dans un premier temps le choix d'un modèle adapté. Nous présentons dans ce chapitre les modèles utilisés en vision omnidirectionnelle, et nous justifions notre choix. Nous présentons ensuite notre contribution sur le calibrage des systèmes stéréoscopiques omnidirectionnels et nous illustrons les performances obtenues par des résultats expérimentaux réalisés à la fois sur des images de synthèse et sur des images réelles.

Le **chapitre 3** est dédié aux méthodes de reconstruction 3D statiques, c'est-à-dire sans déplacement du capteur, qui sont possibles grâce à la structure stéréoscopique de notre système. Après une présentation des méthodes classiques de reconstruction 3D dense, nous présentons notre principale contribution dans ce domaine qui concerne l'adaptation de la programmation dynamique à un système stéréoscopique omnidirectionnel. Enfin, pour les environnements peu structurés tels que les bâtiments, nous proposons un nouvel algorithme de détection de droites dans les images omnidirectionnelles, droites qui sont ensuite triangulées pour obtenir la position des murs.

Nous proposons dans le **chapitre 4** d'exploiter les déplacements du système pour enrichir la reconstruction 3D. Les informations fournies par le robot n'étant pas suffisamment précises, nous proposons une méthode permettant de les estimer à partir d'informations visuelles uniquement. Notre contribution porte d'une part sur la phase d'initialisation en exploitant la géométrie épipolaire des capteurs sphériques, et d'autre part sur un algorithme d'ajustement de faisceaux pour les systèmes stéréoscopiques omnidirectionnels.

Dans la dernière partie de cette thèse, nous résumons les différents points clés abordés, et nous proposons des perspectives aux problèmes soulevés.

# Chapitre 1

## Conception d'un capteur d'acquisition 3D

### Sommaire

---

<b>1.1.Introduction.....</b>	<b>23</b>
<b>1.2.Techniques d'acquisition 3D sans contact.....</b>	<b>24</b>
1.2.1.Systèmes actifs.....	24
1.2.1.1.Télémétrie.....	24
1.2.1.2.Triangulation active.....	27
1.2.2.Systèmes passifs.....	29
1.2.2.1.Shape-From-Shading.....	29
1.2.2.2.Shape-From-Photometric Stereo.....	30
1.2.2.3.Shape-From-Scattering.....	30
1.2.2.4.Shape-From-Texture.....	31
1.2.2.5.Shape-From-Contour.....	31
1.2.2.6.Shape-From-Focus/Defocus.....	32
1.2.2.7.Shape-From-Silhouettes.....	32
1.2.2.8.Shape-From-Consistency.....	33
1.2.2.9.Shape-From-Motion.....	36
1.2.2.10.Shape-From-Stereo.....	36
1.2.3.Synthèse sur les techniques d'acquisition 3D.....	37
<b>1.3.Les systèmes de vision.....</b>	<b>38</b>
1.3.1.Vision classique.....	38
1.3.1.1.Le sténopé.....	38
1.3.1.2.La caméra réelle.....	41
1.3.1.3.Les défauts de la caméra réelle.....	42
1.3.2.Vision omnidirectionnelle.....	44
1.3.2.1.Fusion de plusieurs images.....	45
1.3.2.2.Utilisation d'un objectif grand angle.....	47
1.3.2.3.Utilisation d'un capteur catadioptrique.....	48
<b>1.4.Conception d'un système de vision 3D.....</b>	<b>55</b>
1.4.1.Introduction.....	55
1.4.2.Choix du capteur catadioptrique.....	55
1.4.2.1.Choix des miroirs.....	55
1.4.2.2.Choix des caméras et des optiques.....	58
1.4.2.3.Respect de la contrainte du point de vue unique.....	59
1.4.3.Étude de la configuration des deux capteurs.....	60
1.4.3.1.Géométrie épipolaire.....	61
1.4.3.2.Étude de la résolution du système stéréoscopique catadioptrique.....	64
1.4.3.3.Vue d'ensemble du système.....	65
<b>1.5.Conclusion.....</b>	<b>66</b>



## 1.1. Introduction

Le premier objectif de nos travaux de recherche concerne la conception d'un capteur de perception adapté aux problématiques de la robotique mobile. Cette étude est particulièrement importante puisque l'architecture matérielle du capteur de perception conditionne la capacité du robot à reconstruire son environnement et à le dimensionner pour obtenir les informations lui permettant de planifier sa trajectoire. Le robot doit également pouvoir intervenir dans des environnements intérieurs ou extérieurs (bâtiments, milieu naturel, *etc*) et potentiellement dynamiques (présence d'un ou plusieurs mobiles).

Une architecture matérielle inadaptée augmenterait considérablement la complexité des algorithmes mis en œuvre dans la suite du projet. Nous avons donc étudié les différents systèmes de numérisation 3D, industriels ou universitaires, pour dresser la liste des différentes méthodes susceptibles de répondre à la problématique de la reconstruction 3D. Les méthodes spécifiques à un domaine, par exemple au domaine médical ou à la microscopie, ont été volontairement écartées de l'étude. Un inventaire exhaustif peut néanmoins être trouvé dans les articles de Laugeau [Laugeau97a] et Laurent [Laurent97a].

Les différentes technologies permettant de numériser un objet en trois dimensions peuvent être classées en deux principales catégories : la numérisation 3D par contact et la numérisation 3D sans contact.

La **numérisation par contact**, encore appelée numérisation par palpation, consiste à toucher la surface de l'objet à numériser avec une tête de mesure. L'identification de la position de la tête de mesure dans un référentiel fixe permet alors de déterminer les coordonnées du point 3D en contact avec celle-ci. La détermination de la position de la tête peut être obtenue par un procédé mécanique, en montant la tête sur une machine à commande numérique ou sur un bras articulé, ou par un procédé optique, en utilisant des cibles facilement identifiables. Cette technique de numérisation permet d'obtenir une bonne précision, mais la vitesse d'acquisition est très faible puisque la mesure est réalisée point par point. L'objet à numériser doit de plus être indéformable et de faible dimension. Ce type de numérisation n'est par conséquent pas envisageable dans le cadre de la reconstruction 3D de l'environnement d'un robot mobile.

La **numérisation sans contact** utilise l'émission d'une onde ou des méthodes optiques pour obtenir des informations sur la surface de l'objet à numériser. Cette technologie permet de numériser des volumes très importants en un temps d'acquisition très court puisque le système n'a pas besoin de venir en contact avec chacun des points à reconstruire. Les technologies de numérisation sans contact sont donc plus appropriées à la reconstruction 3D de l'environnement

d'un robot.

La première partie de ce chapitre dresse un état de l'art des techniques d'acquisition 3D sans contact potentiellement applicables à la reconstruction tridimensionnelle de l'environnement d'un robot mobile. La seconde partie du chapitre décrit les systèmes de vision qui sont particulièrement intéressants pour la reconstruction 3D puisqu'ils sont capables de fournir instantanément une grande quantité d'informations sur la scène. Enfin, nous justifions l'architecture matérielle que nous avons développée pour répondre aux problèmes de reconstruction et de navigation, et nous décrivons la conception de ce système.

## **1.2. Techniques d'acquisition 3D sans contact**

Les techniques d'acquisition 3D sans contact peuvent être classées en deux familles technologiques. La première comporte les **systèmes actifs**, c'est-à-dire les systèmes émettant de l'énergie (onde acoustique, électromagnétique, *etc*) pour obtenir des informations sur l'environnement. Cette énergie peut être de la lumière visible ou invisible, une onde électromagnétique, une onde acoustique, *etc*.

La seconde famille regroupe les **systèmes passifs** qui ne font que recevoir des informations de l'environnement sans aucune émission. Ces méthodes utilisent généralement des systèmes de vision pour effectuer les mesures.

### **1.2.1. Systèmes actifs**

Un système actif est un système qui agit sur la scène, par exemple en l'éclairant ou en émettant un signal connu [Laurgeau97a]. Pour obtenir une reconstruction 3D dense de l'environnement, ces techniques nécessitent un système de balayage puisqu'elles fournissent généralement les coordonnées d'un unique point de l'espace. La distance entre le capteur et le point de la scène « éclairé » peut être évaluée de deux façons : par **télémétrie** ou par **triangulation**.

#### **1.2.1.1. Télémétrie**

La télémétrie est une technique de mesure d'une distance par l'émission d'une onde rétrodiffusée par la surface d'un objet (*cf* figure 2). Lorsque l'onde émise par un émetteur rencontre la surface d'un objet, elle est en partie réfléchiée vers un récepteur qui mesure une grandeur physique (temps de vol, déphasage, *etc*), ce qui permet, après traitement, de déterminer la distance parcourue par l'onde. La télémétrie ne fournit qu'une mesure dans la direction d'émission de l'onde, mais l'utilisation d'un système de balayage permet de reconstruire une scène en trois dimensions.

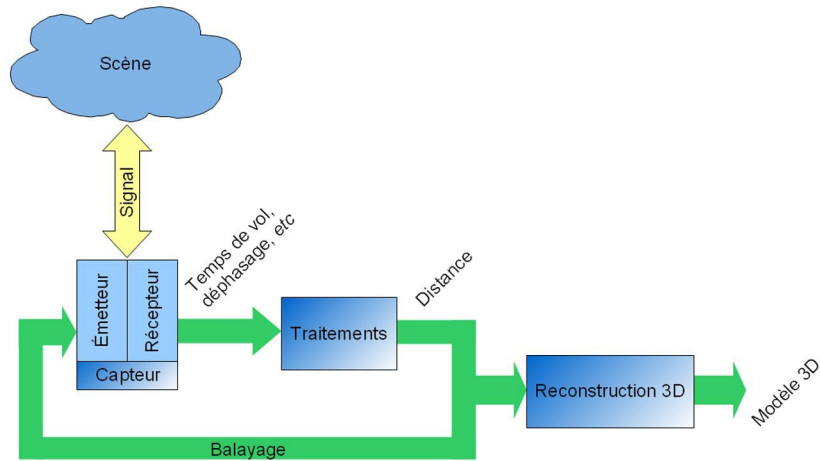


Figure 2: Principe de la télémétrie.

### Radar

Le radar (RAdio Detection And Ranging) est un système qui utilise les ondes radio-électriques pour déterminer la distance des objets. Les radars émettent des ondes qui, lorsqu'elles rencontrent un objet, sont réfléchies et forment un écho-radar comme l'illustre la figure 3. La mesure du temps de vol ou du déphasage de cette onde entre l'émetteur et le récepteur permet de déterminer la distance de l'objet.

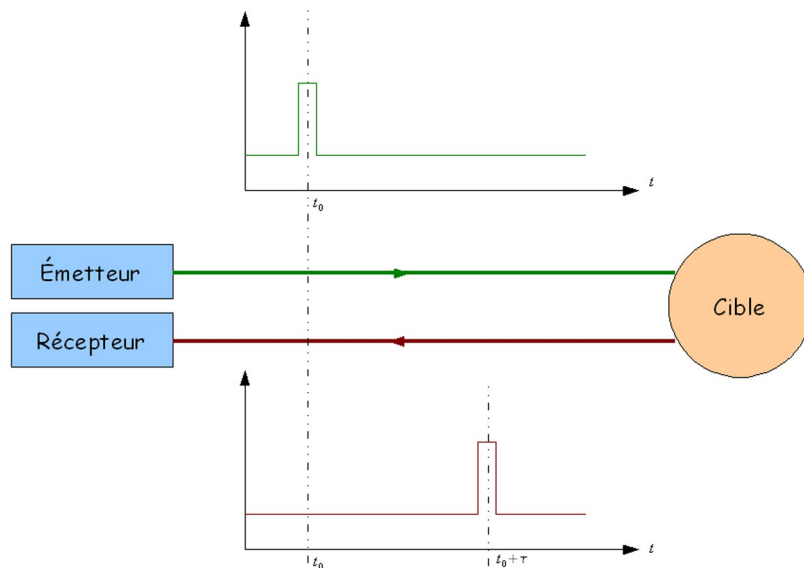


Figure 3: Principe du radar.

Les radars (cf figure 4) ont l'avantage d'avoir une portée très importante, mais la longueur d'onde des ondes électromagnétiques leur confère une faible résolution, surtout à courte distance, et ne sont par conséquent pas adaptés aux environnements confinés.





Figure 4: Radar Autocruise ACC.

### Lidar

Le fonctionnement du lidar (LIght Detection And Ranging) est le même que celui du radar. La seule différence avec le radar est le domaine spectral dans lequel il travaille : le radar fonctionne dans le domaine des ondes radio-électriques alors que le lidar couvre le domaine du visible, de l'ultraviolet et de l'infrarouge. Les lidars ont une bonne directivité car ils utilisent un faisceau laser alors que les radars utilisent un faisceau électromagnétique classique (non polarisé).

Les lidars (*cf* figure 5) sont des systèmes coûteux car ils nécessitent des circuits électroniques de grande précision pour mesurer des temps de vol très courts.



Figure 5: Lidar Sick LD-PDS.

### Télémètre à ultrasons

Le télémètre à ultrasons utilise le temps de vol d'une onde acoustique ultrasonore pour déterminer la distance. La vitesse des ultrasons (environ 340 m/s) est beaucoup plus faible que celle de la lumière, ce qui demande une électronique moins précise et plus abordable que celle des lidars.

Les télémètres à ultrasons (*cf* figure 6) ont une mauvaise résolution angulaire (environ 30°) et une portée relativement faible (une dizaine de mètres), ce qui restreint leur utilisation à la détection d'obstacles.

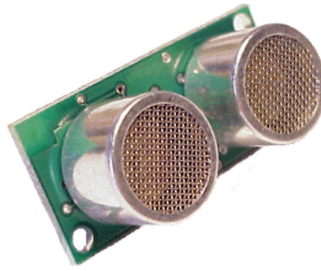


Figure 6: Télémètre à ultrasons.

### 1.2.1.2. Triangulation active

La triangulation active consiste à projeter sur la surface de l'objet un motif lumineux et à observer cette surface avec une caméra décalée de l'émetteur (cf figure 7). La connaissance de la géométrie du système (direction d'émission du rayon, position relative de l'émetteur et du récepteur, position d'observation) permet de calculer les coordonnées des points par trigonométrie. La source lumineuse peut être un rayon laser, un plan laser ou de la « lumière structurée », c'est-à-dire un motif connu.

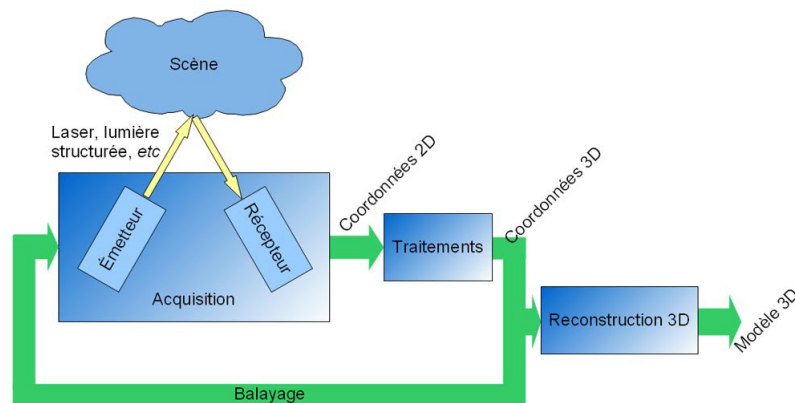


Figure 7: Principe de la triangulation active.

De nombreux capteurs industriels fondés sur la triangulation active ont été développés car ils fonctionnent sur des objets uniformes et sont beaucoup moins sensibles aux conditions d'éclairage que les systèmes passifs.

#### Projection d'un rayon laser

La projection d'un faisceau laser (point laser) sur la surface de l'objet est la technique de triangulation active la plus simple. En effet, un seul point étant projeté, celui-ci est facilement identifiable sur l'image et un simple calcul trigonométrique permet de calculer ses coordonnées, comme l'illustre la figure 8.

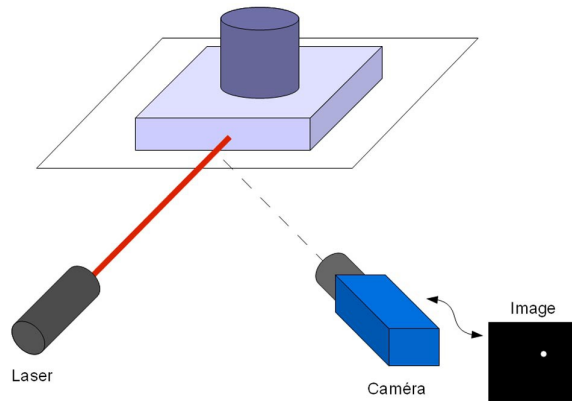


Figure 8: Triangulation laser point.

L'inconvénient majeur de cette technique est le temps d'acquisition car il est nécessaire d'utiliser un balayage autour de deux axes pour aboutir à un modèle 3D global.

### Projection d'un plan laser

Un plan lumineux, obtenu en associant une lentille cylindrique à une diode laser, est projeté sur la surface de l'objet à numériser. La caméra relève le profil formé par ce plan lumineux sur la pièce. Les profils sont ensuite traduits en coordonnées 3D grâce au principe de la triangulation comme l'illustre la figure 9.

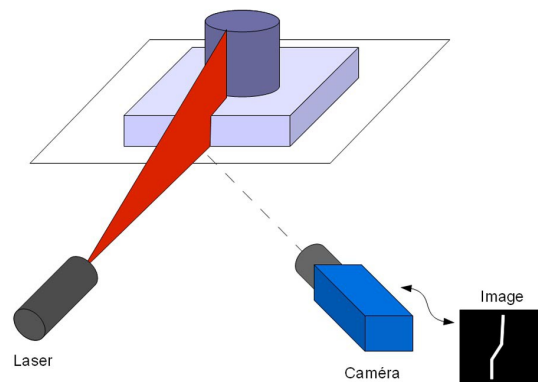


Figure 9: Profilométrie.

Cette technique, également connue sous le nom de profilométrie, est plus rapide que la projection d'un point laser puisqu'à chaque point du profil dans l'image correspond un point 3D dans l'espace. Un balayage est nécessaire mais autour d'un seul axe.

### Projection de lumière structurée

Pour s'affranchir du système de balayage, il est possible de projeter une figure bi-dimensionnelle sur l'objet grâce à un projecteur de lumière structurée (*cf* figure 10). Ces figures peuvent être des grilles, des franges binaires, des franges codées en niveaux de gris, *etc.* Le temps d'acquisition est alors grandement amélioré puisqu'il n'y a plus besoin de mécanisme de

déplacement du système.

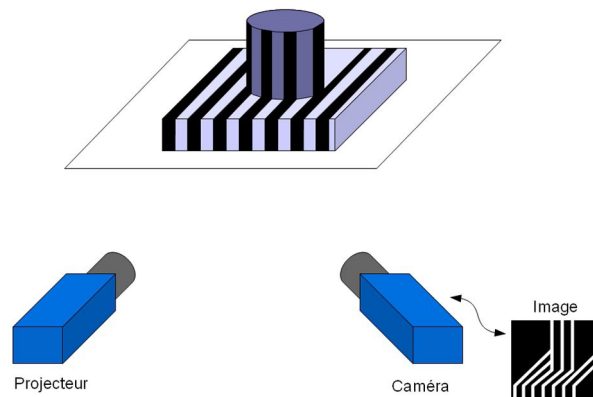


Figure 10: Projection de lumière structurée.

## 1.2.2. Systèmes passifs

Les systèmes de reconstruction 3D passifs cherchent à retrouver la structure 3D de la scène sans émettre d'onde ou de lumière spécifique. Les systèmes passifs utilisent des capteurs de vision (CCD ou CMOS) pour retrouver la forme 3D des objets à partir d'une ou plusieurs images. En vision par ordinateur, ces techniques de reconstruction 3D à partir d'images sont appelées techniques de « *Shape-From-X* », où X peut être « *Stereo* », « *Motion* », « *Shading* », etc...

### 1.2.2.1. Shape-From-Shading

Le *Shape-From-Shading* (SFS) consiste à calculer la forme tridimensionnelle d'une surface à partir des variations graduelles de son ombre dans une seule image 2D. La modélisation du processus de formation des images permet en effet d'exprimer le niveau de gris d'un pixel de l'image en fonction de la direction de la source de lumière et de la normale à la surface de l'objet. Le but du SFS est donc, étant donné un niveau de gris de l'image, de retrouver la source de lumière et la normale à la surface pour chaque pixel de l'image.

Horn [Horn75a] a été le premier, dans les années 70, à montrer qu'il était possible de reconstruire une forme en se basant sur l'ombre, mais les hypothèses sur lesquelles il s'appuyait à l'origine étaient irréalistes et le SFS n'a pas donné de résultats intéressants pendant près de trente ans.

Le problème du SFS a été reformulé récemment par Prados et Faugeras [Prados04a] [Prados06a]. Les résultats obtenus avec cette nouvelle modélisation sont assez concluants, comme le montre la figure 11, et ont relancé l'intérêt de la communauté scientifique pour le *Shape-From-Shading*. Cette méthode ne fonctionne cependant que sur des objets uniformes et avec un éclairage contrôlé, ce qui est inadapté à la reconstruction 3D de scènes complexes.

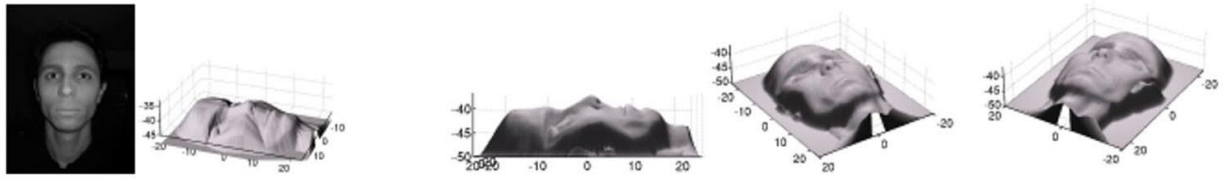


Figure 11: Reconstruction 3D d'un visage par Shape-From-Shading.  
Source [Prados06a].

### 1.2.2.2. Shape-From-Photometric Stereo

Cette technique, connue sous le nom de stéréophotométrie, a été introduite par Woodham en 1980 [Woodham80a]. Elle consiste à utiliser plusieurs images d'une scène en gardant le même point de vue mais en modifiant la direction de l'éclairage entre les clichés. Bien que similaire sur le principe au *Shape-From-Shading*, la stéréophotométrie est une technique plus aboutie qui permet d'obtenir de meilleurs résultats comme l'illustre la figure 12.

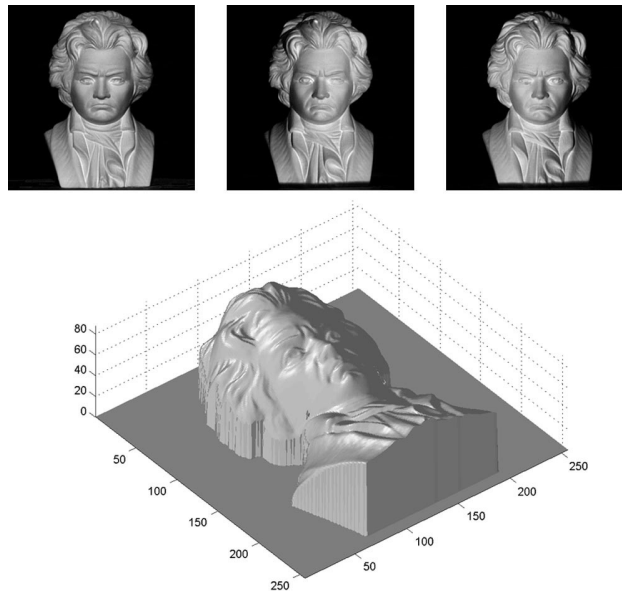


Figure 12: Reconstruction 3D d'un buste par stéréophotométrie.  
Source [Durou07a].

En revanche, la quantité d'informations nécessaire pour la stéréophotométrie est plus importante que le SFS qui ne nécessite qu'une image, elle ne fonctionne pas en environnement dynamique et les conditions opératoires sont très contraignantes puisqu'il faut changer la direction de la source d'éclairage sans modifier la position de la caméra.

### 1.2.2.3. Shape-From-Scattering

Le Shape-From-Scattering exploite la diffusion atmosphérique pour obtenir la carte des profondeurs de la scène. Il existe en effet une relation liant la luminance et la distance entre

l'objet et l'observateur [Cozman97a].

Cette technique donne de bons résultats dans les environnements extérieurs et a l'avantage de fonctionner là où les autres techniques sont vouées à l'échec : en présence de brouillard. Elle n'est cependant pas la plus adaptée à la reconstruction 3D d'environnements intérieurs car les distances entre les objets et le nombre de particules en suspension dans l'air ne sont pas suffisants pour mesurer des variations de luminance.

#### 1.2.2.4. Shape-From-Texture

Le *Shape-From-Texture* (SFT) consiste à étudier les distorsions des textures dues à la projection du monde tridimensionnel sur un plan image bidimensionnel.

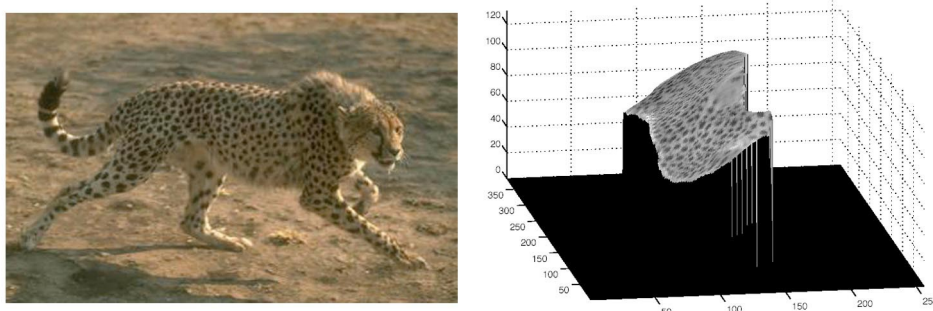


Figure 13: Exemple de reconstruction 3D d'un guépard par Shape From Texture.  
Source [Forsyth02b].

Le principe même de cette méthode repose sur la déformation d'une texture et n'est par conséquent pas applicable dans des environnements peu texturés tels que les bâtiments.

#### 1.2.2.5. Shape-From-Contour

La déformation des contours d'un objet dans l'image fournit des informations sur la forme de sa surface. Ainsi, les auteurs de [Courteille07a] proposent d'appliquer le *Shape-From-Contour* (SFC) pour calculer les déformations d'un livre et aplatir son image dans le but de faciliter sa numérisation (cf figure 14).

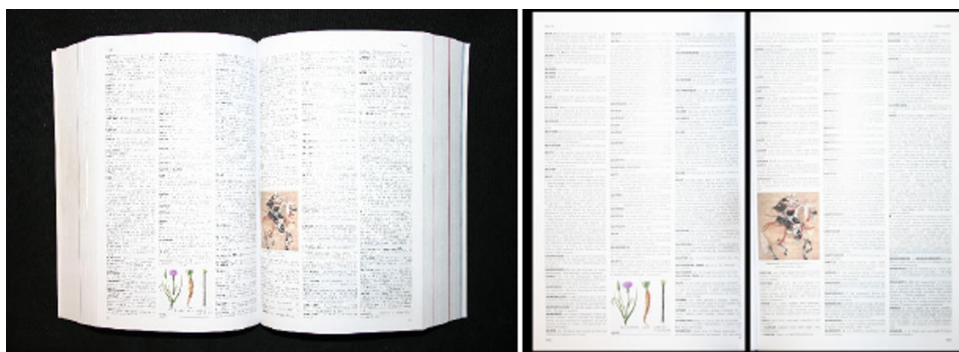


Figure 14: Application du Shape-From-Contour à la rectification d'images.  
Source [Courteille07a].

Cette méthode n'est cependant pas applicable à la reconstruction 3D de scènes complexes et/ou inconnues.

### 1.2.2.6. *Shape-From-Focus/Defocus*

L'utilisation d'une caméra réelle (*cf* section 1.3.1.2) ayant une profondeur de champ limitée permet de retrouver la notion de distance des objets. En effet, si la profondeur de champ de cette caméra est très faible, seule la petite tranche de la scène située au focus apparaît nette dans l'image. Le *Shape-From-Focus/Defocus* exploite cette caractéristique pour estimer la surface 3D d'un objet de la scène à partir d'un jeu d'images capturées du même point de vue mais avec paramètres de caméra différents (focale, ouverture, *etc*).

Contrairement à la plupart des algorithmes de vision par ordinateur qui modélisent les caméras par le modèle *pinhole* (*cf* section 1.3.1.1), le *Shape-From-Focus/Defocus* exploite un modèle de caméra réelle faisant intervenir les lois de l'optique géométrique, et par conséquent les paramètres de focale, de distance entre la lentille et le plan de la scène au focus, et de distance entre la lentille et le plan image.

La différence entre les problèmes de *Shape-From-Focus* et de *Shape-From-Defocus* vient de la possibilité dans le premier cas de modifier dynamiquement les paramètres de la caméra pendant le processus d'estimation de la surface [Nayar94a], ce qui n'est pas autorisé dans le second cas [Favaro05a].

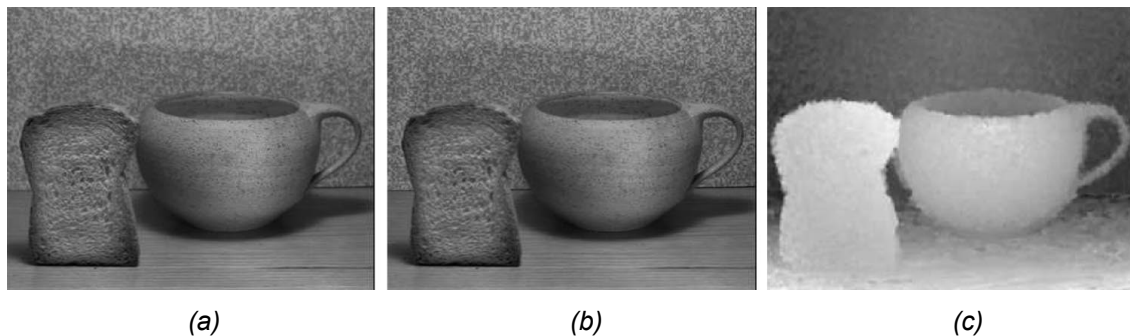


Figure 15: Résultat obtenu par *Shape-From-Defocus*.

(a) et (b) Images acquises avec des paramètres de focus différents - (c) Carte de profondeur obtenue.

Source [Favaro05a]

### 1.2.2.7. *Shape-From-Silhouettes*

Le *Shape-From-Silhouettes*, introduit par Baumgart [Baumgart74a], est une méthode permettant de reconstruire automatiquement le modèle 3D d'un objet à partir d'une séquence d'images de cet objet prise sous de multiples points de vue. Les images sont binarisées afin de ne conserver que la silhouette de l'objet qui est la seule information utilisée par cette méthode. De

cette façon, la valeur des pixels indique s'il s'agit ou non d'un point appartenant à l'objet. L'extraction des silhouettes est donc une étape décisive qui peut être grandement simplifiée si un fond uniforme est placé derrière l'objet à numériser.

La forme de l'objet, appelée enveloppe visuelle, est ensuite obtenue par l'intersection des cônes formés par les centres optiques des caméras et les silhouettes de l'objet dans les plans images. Comme le montre la figure 16, le modèle 3D obtenu par cette méthode (en gris) ne correspond pas tout à fait à l'objet réel (en bleu). Il s'agit du principal inconvénient du *Structure From Silhouettes* qui ne permet pas de reconstruire les concavités d'un objet. De plus, cette méthode de reconstruction 3D n'utilise pas les propriétés photométriques des images qui amélioreraient ses performances.

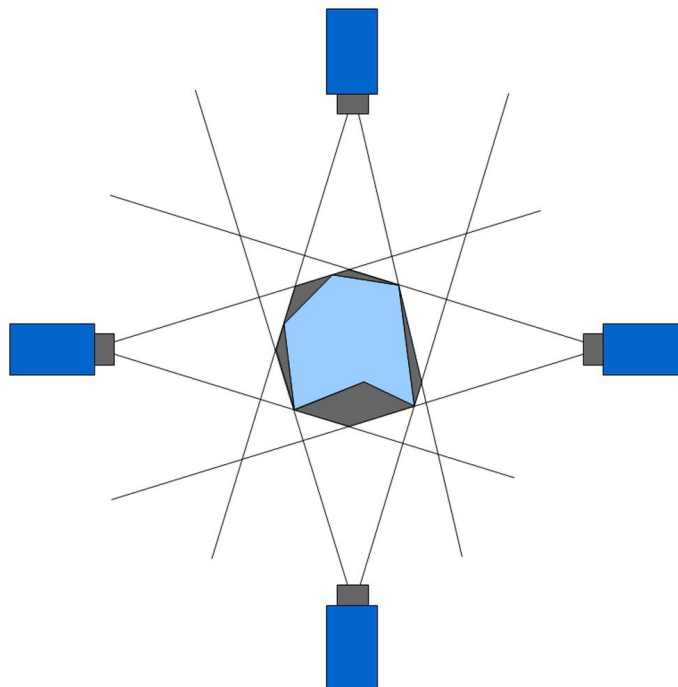


Figure 16: Principe du Shape-From-Silhouettes.

#### 1.2.2.8. Shape-From-Consistency

Les techniques de *Shape-From-Consistency* sont des méthodes de reconstruction dites volumétriques. Contrairement à la plupart des méthodes qui consistent à mettre en correspondance les pixels des images puis à les trianguler pour retrouver la structure de la scène (passage 2D→3D), les méthodes volumétriques sont fondées sur des calculs permettant d'obtenir les volumes ou les surfaces qui sont consistants avec les images (passage 3D→2D).

L'espace 3D est discrétisé en volumes élémentaires appelés « voxels », ce qui suppose que le volume dans lequel se trouvent les objets à reconstruire est connu et borné comme l'illustre la figure 17.



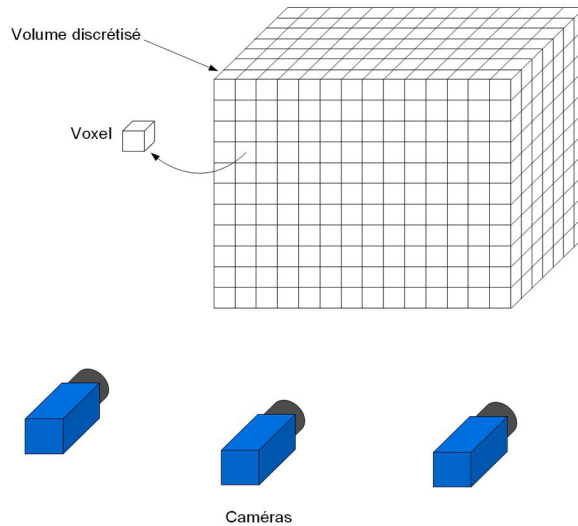


Figure 17: Discretisation de l'espace en volumes élémentaires.

La reconstruction par photo-consistance consiste à étudier les propriétés photométriques des projections d'un voxel dans les images pour déterminer si le voxel appartient à la surface d'un objet (voxel photo-consistant) ou si le voxel est vide (voxel non photo-consistant). Ce principe est illustré par la figure 18. Sur la figure 18 a, le voxel n'est pas photo-consistant puisque la caméra de gauche observe un voxel bleu alors que la caméra de droite observe un voxel orange : ce voxel est par conséquent « vide » ou « transparent ». Sur la figure 18 b, la projection du voxel dans les deux images a la même couleur : il est donc considéré comme « plein » ou « opaque ».

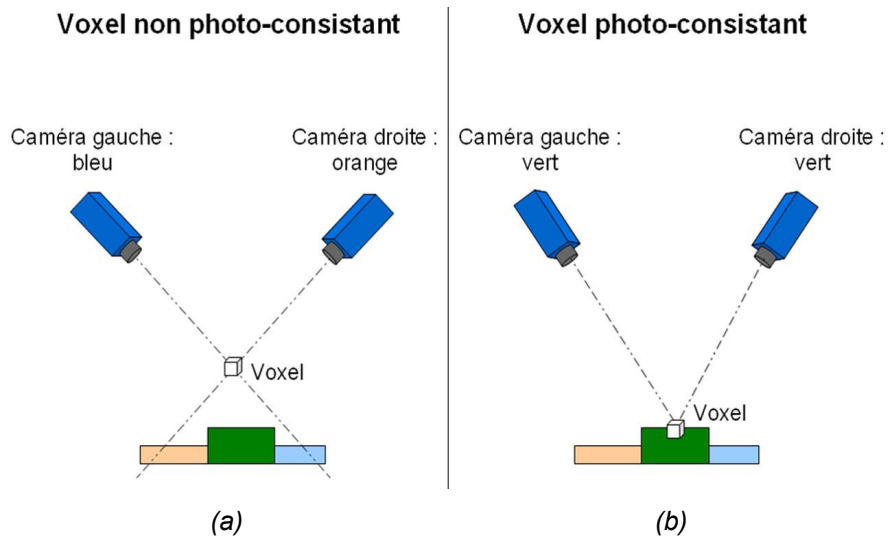


Figure 18: Principe de la photo-consistance.  
 (a) Voxel non photo-consistant – (b) Voxel photo-consistant.

Le modèle 3D obtenu est donc l'ensemble des voxels de l'espace discrétisé auxquels une valeur indiquant leur état est attribuée. Cette valeur peut être binaire (voxel plein/opaque ou vide/transparent), ternaire (plein/opaque, vide/transparent, non vu) ou une valeur indiquant le

degré de probabilité que le voxel soit plein [Broadhurst01a].

La reconstruction volumétrique par photo-consistance peut être déclinée en deux approches. La première approche, connue sous le nom de *voxel coloring* [Seitz97a], consiste à supposer que le volume à reconstruire est initialement composé de voxels transparents, c'est-à-dire que le volume à reconstruire est initialement vide. La photo-consistance des voxels est ensuite testée et les voxels qui passent le test de photo-consistance avec succès sont marqués opaques. Les voxels sont examinés dans un ordre particulier pour prendre en compte la contrainte de visibilité des voxels. La prise en compte de l'ordre de la visibilité permet de reconstruire la scène en une seule passe mais impose des contraintes sur la disposition des caméras.

La seconde approche, le *space carving* [Kutulakos00a], est une généralisation du principe du *voxel coloring* à des configurations de caméras quelconques. Dans une telle configuration, les voxels ne peuvent plus être classés en fonction de leur visibilité et l'algorithme en une seule passe ne peut plus être appliqué. La photo-consistance des voxels est donc testée itérativement et comme chaque passe est susceptible de modifier la visibilité ou la propriété (transparent ou opaque) des voxels, le processus doit être répété jusqu'à ce qu'il n'y ait plus aucun changement. Contrairement au *voxel coloring*, le *space carving* considère que le volume à reconstruire est initialement plein et les voxels qui échouent au test de photo-consistance sont retirés, d'où le nom de « creusage d'espace ».

Les méthodes de reconstruction 3D par photo-consistance donnent d'excellents résultats pour la reconstruction 3D d'objets (voir figure 19), en particulier lorsqu'il est possible de différencier l'objet de l'arrière plan et lorsque les prises de vue couvrent la totalité de l'objet. L'application de ces méthodes à une caméra placée à l'intérieur d'une scène complexe est donc plus délicate.



Figure 19: Exemple de reconstruction 3D par photo-consistance.  
Source [Kutulakos00a].

### 1.2.2.9. Shape-From-Motion

Le *Shape-From-Motion* (SFM) ou *Structure-From-Motion* consiste à utiliser une seule caméra en mouvement pour retrouver la structure géométrique de la scène. Cette méthode impose soit de connaître les déplacements entre deux prises de vues [Carceroni06a], soit de les estimer à partir des images [Wang92a].

Les techniques de SFM ont acquis une certaine maturité ces dernières années et permettent désormais d'obtenir des modèles 3D très proches de la réalité (*cf* figure 20). Les travaux les plus aboutis sur le sujet sont ceux de Pollefeys [Pollefeys04a], Nister [Nister01a] et Rodriguez [Rodriguez03a].

Les algorithmes de SFM nécessitent cependant de traiter toutes les images de la séquence simultanément. De plus, le modèle 3D obtenu n'est généralement défini qu'à un facteur d'échelle près puisqu'aucun étalon de distance n'est disponible. Enfin, les images n'étant pas acquises simultanément, les algorithmes de SFM ne fonctionnent pas dans des environnements dynamiques. Ces inconvénients rendent les algorithmes de SFM inadaptés à la robotique mobile puisqu'un robot qui explore son environnement a besoin d'informations sur le monde qui l'entoure en continu pour pouvoir planifier sa trajectoire. Il existe néanmoins une variante du SFM, appelée SLAM (*Simultaneous Localization and Mapping*), qui a pour but la localisation et la cartographie de l'environnement en temps réel, et qui est donc plus adaptée à la robotique mobile (*cf* chapitre 4).

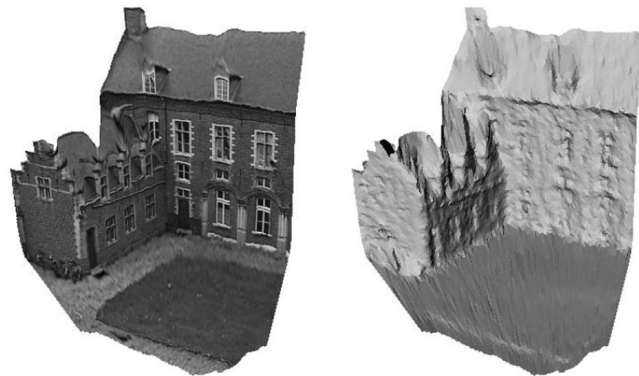


Figure 20: Modèles 3D obtenus par SFM.  
Source [Pollefeys04a]

### 1.2.2.10. Shape-From-Stereo

Le *Shape-From-Stereo*, ou **stéréovision**, repose sur le même principe que le *Shape-From-Motion* mis à part que les images sont obtenues par plusieurs caméras et non par le mouvement d'une même caméra. La principale difficulté de la stéréovision réside dans la mise en correspondance des pixels, c'est-à-dire la détermination dans les images des projections  $\mathbf{m}_1$  et

$m_2$  d'un même point  $M$ . Une fois les pixels mis en correspondance, les coordonnées du point  $M$  sont obtenues par triangulation des points  $m_1$  et  $m_2$  à partir des centres optiques  $C_1$  et  $C_2$  des caméras, comme l'illustre la figure 21.

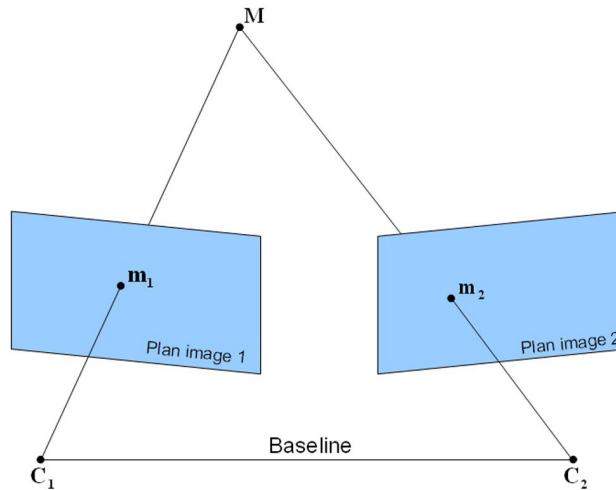


Figure 21: Principe de la stéréovision.

Bien que deux caméras soient suffisantes pour retrouver l'information de profondeur par triangulation, l'utilisation de caméras supplémentaires permet d'ajouter des contraintes qui facilitent la mise en correspondance et augmentent la fiabilité de l'algorithme [Ayache89a]. Cependant, la multiplication des caméras, outre le coût matériel qu'elle engendre, augmente le temps de calcul et réduit la taille de la zone de reconstruction puisque celle-ci correspond au champ de vue commun de toutes les caméras. Le choix du nombre de caméras à utiliser est donc soumis à un compromis. L'ajout d'une troisième caméra est courant car elle apporte un gain significatif sur les résultats mais un nombre de caméras supérieur rend les inconvénients cités ci-dessus trop contraignants.

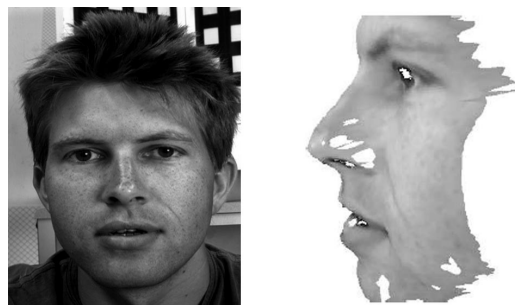


Figure 22: Reconstruction 3D d'un visage par stéréovision.

Source [Devernay97a]

### 1.2.3. Synthèse sur les techniques d'acquisition 3D

Les techniques d'acquisition 3D passives présentées dans la section 1.2.2. utilisent

différentes méthodes pour aboutir au modèle 3D, mais le capteur utilisé est toujours un système de vision. Ce type de capteur est particulièrement intéressant puisqu'il permet d'obtenir une grande quantité d'informations tout en étant peu coûteux, léger et peu gourmand en énergie.

Les données fournies par un tel capteur ne sont cependant pas exploitables directement car il ne s'agit pas de données sur les distances mais uniquement sur la luminance des pixels. Le robot doit donc être capable de comprendre et d'interpréter ces données pour modéliser son environnement en trois dimensions. Toute la difficulté de la **vision par ordinateur** (ou vision artificielle) réside donc dans les algorithmes d'interprétation à mettre en œuvre pour permettre au robot de percevoir et de comprendre son environnement.

### 1.3. Les systèmes de vision

#### 1.3.1. Vision classique

Un système de vision classique est un système de vision utilisant une seule caméra qui observe directement son environnement. Les algorithmes fondés sur l'utilisation de caméras classiques (perspectives ou orthographiques) sont plus aboutis que ceux fondés sur la vision omnidirectionnelle, celle-ci ne datant que de quelques années.

Cette section a pour objectif de décrire le fonctionnement de la vision classique et d'introduire le modèle de projection le plus couramment utilisé en vision par ordinateur : le sténopé. Ces notions sont en effet nécessaires pour comprendre le fonctionnement des systèmes omnidirectionnels qui sont plus complexes mais qui sont fondés sur les mêmes principes.

##### 1.3.1.1. Le sténopé

Le sténopé (*pinhole* en anglais) est un dispositif optique sans lentille datant du XV<sup>ème</sup> siècle. Il est composé d'une boîte dont une des faces est percée d'un minuscule trou qui laisse entrer la lumière (*cf* figure 23). L'image vient se former sur la face opposée au trou et celle-ci peut être capturée en y plaçant un support photosensible (papier photographique).

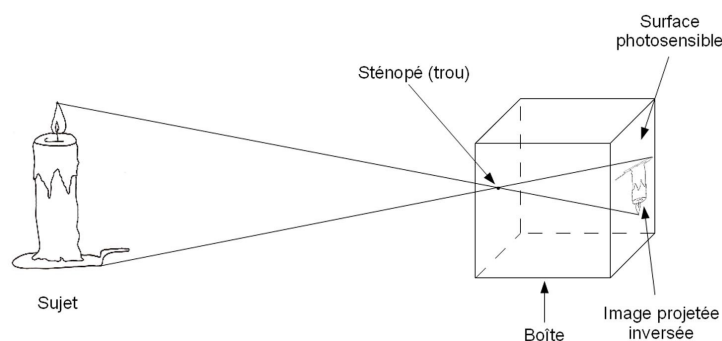


Figure 23: Formation d'une image dans un sténopé.

Le sténopé est un appareil très simple mais sa petite ouverture ne laisse passer qu'une très faible quantité de lumière. Le temps d'exposition nécessaire pour marquer le papier photographique doit par conséquent être très élevé (jusqu'à plusieurs heures). Il n'est donc pas concevable d'utiliser un tel système de vision de nos jours, mais son modèle géométrique permet de décrire la formation des images, c'est-à-dire la transformation permettant de passer de l'espace tridimensionnel du monde à l'espace bidimensionnel du plan image.

Une caméra perspective peut en effet être modélisée grâce au modèle du sténopé comme l'illustre la figure 24. La caméra est représentée par un **plan image**, ou **rétine**, correspondant à la surface photosensible (capteur CCD ou CMOS) et un **centre de projection C**, aussi appelé **centre optique**, situé à la **distance focale  $f$**  de ce plan. L'**axe optique** est la droite passant par le centre de projection et perpendiculaire au plan image. L'intersection de l'axe optique avec le plan image est appelé **point principal**.

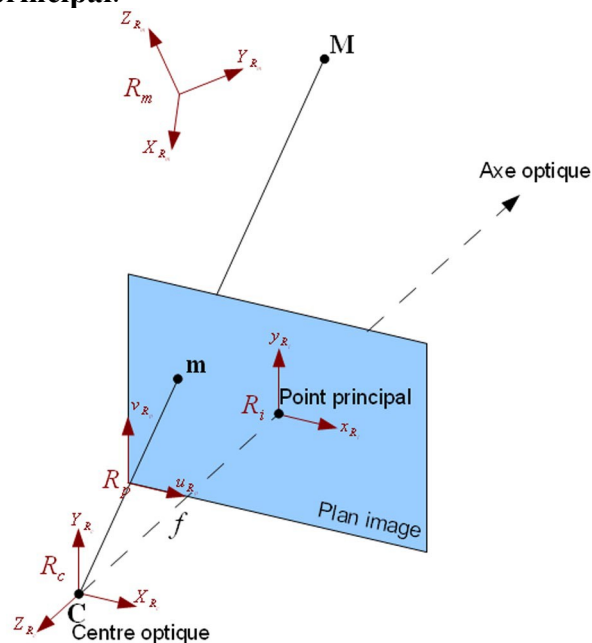


Figure 24: Modèle géométrique du sténopé.

La projection  $\mathbf{m}$  d'un point de l'espace  $\mathbf{M}$  est l'intersection du rayon optique ( $\mathbf{CM}$ ) avec le plan image. Cette projection peut être décomposée en trois transformations élémentaires successives.

Les coordonnées du point  $\mathbf{M} = [X_m \ Y_m \ Z_m \ 1]^T$  sont en général exprimées dans un repère arbitraire, appelé repère monde (cf figure 24). La première transformation est donc une transformation rigide  $\mathbf{T}$  qui permet d'exprimer ces coordonnées dans le repère caméra  $R_c$ . Le repère caméra est le repère dont l'origine se situe sur le centre optique  $C$  (cf figure 24). Les coordonnées  $[X_c \ Y_c \ Z_c \ 1]^T$  du point  $\mathbf{M}$  dans le repère caméra sont donc obtenues par la

relation suivante :

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} + \mathbf{t} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{3 \times 1}^T & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (1)$$

où  $\mathbf{t}$  est le vecteur de translation entre le repère monde et le repère caméra et  $\mathbf{R}$  est la matrice de rotation entre ces deux repères. Les paramètres de cette transformation sont appelés **paramètres extrinsèques** de la caméra.

La deuxième transformation relie le repère caméra  $R_c$  au repère du plan image  $R_i$ . Il s'agit d'une projection perspective  $\mathbf{P}$  qui transforme un point 3D de coordonnées  $[X_c \ Y_c \ Z_c \ 1]^T$  en un point image de coordonnées  $[x \ y \ 1]^T$  en unité métrique. En utilisant les coordonnées homogènes, cette transformation peut s'écrire :

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \sim \mathbf{P} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2)$$

où  $f$  désigne la distance focale.

La dernière transformation est une transformation affine  $\mathbf{A}$  qui permet de convertir les coordonnées images métriques  $[x \ y \ 1]^T$  en coordonnées images discrètes  $[u \ v \ 1]^T$  :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & -k_u \cot \theta & u_0 \\ 0 & k_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

où :

- $u_0$  et  $v_0$  sont les coordonnées (en pixels) du point principal,
- $k_u$  et  $k_v$  correspondent au nombre de pixels par unité de longueur du capteur suivant les directions  $u$  et  $v$  ( $k_u = k_v$  dans le cas de pixels carrés),
- $\theta$  traduit la non orthogonalité des lignes et des colonnes de l'image.

En pratique, on considère souvent que les lignes et les colonnes de l'image sont perpendiculaires, et donc que  $\theta = \pi/2$ . L'équation (3) peut alors se simplifier :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

Le produit  $\mathbf{AP}$  des matrices correspondant aux deux dernières transformations (projection perspective  $\mathbf{P}$  et projection affine  $\mathbf{A}$ ) peut être pré-calculé puisqu'il est indépendant de la

position de la caméra et de la structure de la scène :

$$\mathbf{AP} = \begin{bmatrix} f k_u & 0 & u_0 & 0 \\ 0 & f k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5)$$

En posant  $\alpha_u = f k_u$  et  $\alpha_v = f k_v$ , l'équation (5) devient :

$$\mathbf{AP} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [\mathbf{K} \mid \mathbf{0}] \quad (6)$$

Ce produit contient toutes les grandeurs physiques de la caméra : la focale exprimée en pixels dans les directions  $\mathbf{u}$  et  $\mathbf{v}$  ( $\alpha_u$  et  $\alpha_v$ ), les coordonnées du point principal ( $u_0$  et  $v_0$ ) et éventuellement le paramètre d'obliquité  $\theta$  si celui-ci n'a pas été négligé. Ce sont ces paramètres, appelés **paramètres intrinsèques**, qui doivent être estimés lors du calibrage. Ils sont regroupés dans la matrice des paramètres intrinsèques  $\mathbf{K}$ , telle que :

$$\mathbf{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

En coordonnées homogènes, la projection  $\mathbf{m} = [u \ v \ 1]^T$  du point  $\mathbf{M} = [X_m \ Y_m \ Z_m \ 1]^T$  est donc obtenue par la relation :

$$\mathbf{m} \sim [\mathbf{K} \mid \mathbf{0}] \mathbf{T} \mathbf{M} \sim \mathbf{P}_{\text{proj}} \mathbf{M} \quad (7)$$

où  $\mathbf{P}_{\text{proj}}$  est appelée **matrice de projection**.

### 1.3.1.2. La caméra réelle

Le modèle mathématique du sténopé permet donc de fournir une approximation acceptable du processus de prise d'images mais le système physique n'est pas utilisable en pratique à cause du peu de lumière que laisse pénétrer l'ouverture. Pour résoudre ce problème, les caméras sont équipées de lentilles qui permettent de concentrer les rayons lumineux sur le plan image. Deux principales familles de lentilles existent : les lentilles perspectives et les lentilles télécentriques.

Les lentilles perspectives sont les lentilles les plus couramment utilisées. Elles permettent d'obtenir des images perspectives identiques à celles obtenues par un sténopé et son modèle mathématique est par conséquent bien adapté. La principale caractéristique d'un tel objectif est sa focale.

Les lentilles télécentriques permettent de projeter chaque point du champ de vue suivant une direction parallèle à l'axe optique ; elles réalisent donc une projection **orthographique** qui



peut aussi être vue comme une projection perspective de distance focale infinie. Elles éliminent par conséquent les effets de perspective, ce qui est particulièrement intéressant pour faire l'acquisition d'images d'objets en trois dimensions où les effets de perspective peuvent être mal interprétés (*cf* figure 25).

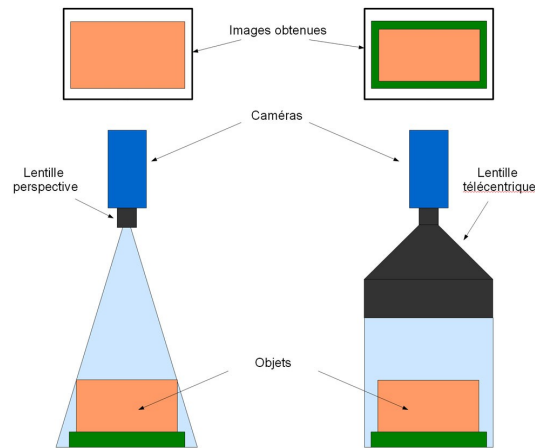


Figure 25: Comparaison entre une caméra perspective (à gauche) et une caméra orthographique (à droite).

Les lentilles télécentriques sont cependant plus coûteuses et surtout plus encombrantes que les lentilles perspectives car elles nécessitent que le champ observé soit inscrit dans le diamètre de la pupille d'entrée de l'objectif.

### 1.3.1.3. Les défauts de la caméra réelle

Les lentilles permettent donc de corriger le problème de la luminosité du sténopé, mais elles introduisent par la même occasion des défauts dans les images. Ces défauts doivent être pris en compte dès la conception d'un système de vision pour choisir les éléments optiques les plus appropriés, puis lors des traitements pour améliorer la précision du système.

#### Réduction de la profondeur de champ

La profondeur de champ correspond à la zone de l'espace dans lequel doit se situer l'objet pour que son image soit nette. Elle est liée à la distance de mise au point et à l'ouverture du diaphragme. L'ouverture du sténopé étant minuscule, sa profondeur de champ est très grande (presque infinie). L'utilisation de la lentille réduit donc cette profondeur de champ, ce qui a un impact particulièrement important lors de la conception d'un capteur catadioptrique (voir section 1.4).

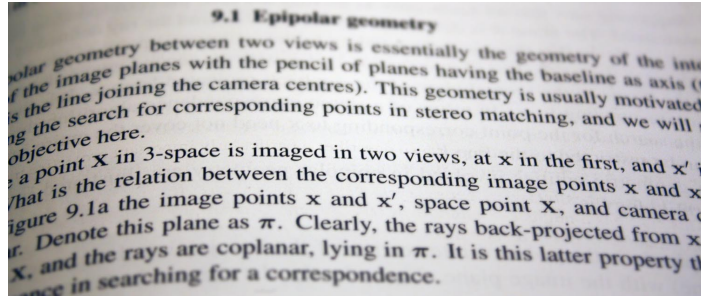


Figure 26: Illustration de l'impact d'une faible profondeur de champ. Seuls les points au focus sont nets.

### Aberrations sphériques

Les rayons frappant le bord de la lentille sont plus fortement réfractés que ceux passant à proximité de l'axe optique. Une partie des rayons ne passe donc pas par le point focal (cf figure 27), ce qui provoque une perte de netteté en périphérie de l'image.

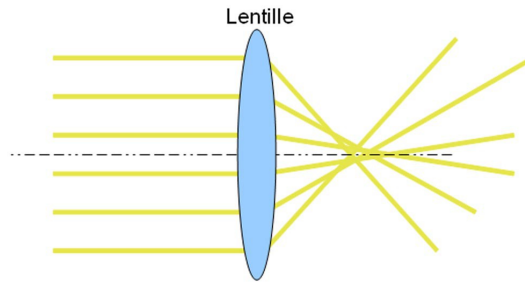


Figure 27: Aberrations sphériques.

### Aberrations chromatiques

Ce défaut est dû à la variation de l'indice de réfraction du matériau composant la lentille en fonction de la longueur d'onde de la lumière qui la traverse (cf figure 28). La mise au point ne peut donc pas être effectuée simultanément pour toutes les couleurs du spectre, ce qui provoque en général l'apparition de franges violettes sur les contours des objets dans l'image.

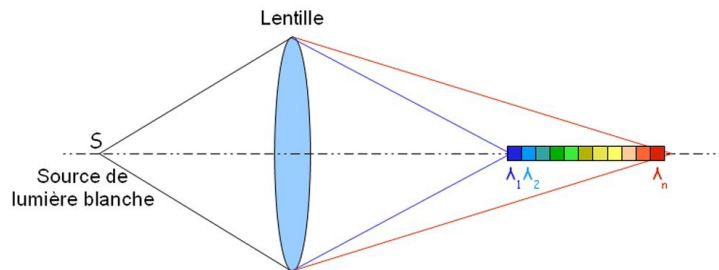


Figure 28: Aberrations chromatiques.

### Distorsions

Les distorsions sont des aberrations géométriques inhérentes aux objectifs qui se traduisent

par la courbure des lignes droites comme l'illustre la figure 29. Deux types de distorsions peuvent apparaître dans les images [Brand93a] :

- les distorsions tangentielles qui apparaissent lorsque les lentilles ne sont pas parfaitement orthogonales à l'axe optique,
- les distorsions radiales qui sont dues aux courbures des lentilles.

Pour améliorer la précision du modèle de projection, il est souvent nécessaire d'ajouter au modèle du sténopé des coefficients de distorsions. La modélisation des distorsions et l'estimation des coefficients de distorsions sont abordées dans le chapitre 2.

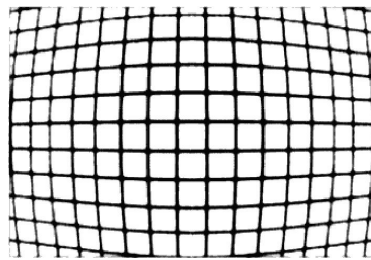


Figure 29: Exemple de distorsions.

### 1.3.2. Vision omnidirectionnelle

Le principal inconvénient d'un système de vision classique, c'est-à-dire utilisant une caméra qui observe directement la scène, est son champ de vue limité (cf figure 30 a). Dans des applications telles que la robotique mobile et la télésurveillance, il est particulièrement important de pouvoir observer l'environnement dans toutes les directions et donc de supprimer le concept de direction du regard. C'est pour pallier cette limitation que la vision omnidirectionnelle a vu le jour puisqu'elle a pour but d'augmenter le champ de vue d'un système de vision pour le rendre le plus grand possible, voire même atteindre un angle de vue de 360°.

Les termes vision omnidirectionnelle et vision panoramique sont généralement employés indifféremment alors qu'il existe une différence entre ces deux concepts. La vision omnidirectionnelle est un procédé de vision qui doit être capable de fournir une image de l'environnement dans toutes les directions, le champ de vue doit donc être sphérique comme l'illustre la figure 30 c. La vision panoramique (figure 30 b) a un champ de vue un peu plus limité car elle couvre un champ de vue de 360° autour d'un axe vertical mais elle ne couvre pas la totalité de l'environnement dans l'autre sens. Dans la pratique, un système omnidirectionnel est extrêmement difficile à réaliser car le capteur d'image obstrue généralement une partie du champ de vue et les termes omnidirectionnel et panoramique sont tous deux employés pour qualifier un système de vision couvrant un champ de 360° autour d'un axe vertical.

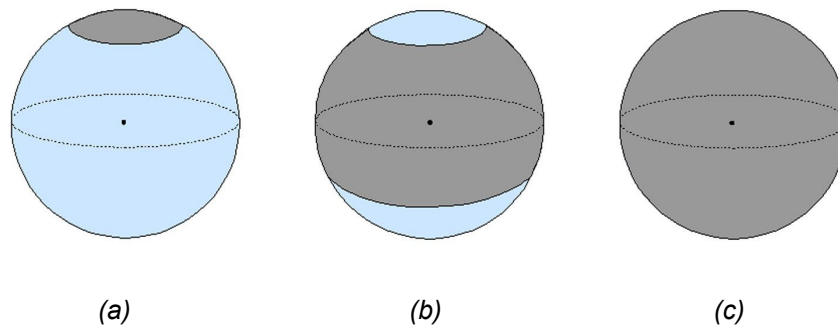


Figure 30: Champ de vue des systèmes de vision (représenté en gris).

(a) Vision classique – (b) Vision panoramique - (c) Vision omnidirectionnelle

La vision omnidirectionnelle est un domaine de recherche très récent. Bien qu'un système de vision omnidirectionnelle fut breveté dès 1970 par Rees [Rees70a], il faut attendre les travaux de Yagi [Yagi90a] et ceux de Nayar et Baker [Nayar97a] [Baker99a] dans les années 90 pour que la vision omnidirectionnelle connaisse un véritable essor. De nombreux travaux ont d'ores et déjà été publiés mais beaucoup d'aspects de la vision omnidirectionnelle restent à découvrir ou à approfondir.

Les solutions pour augmenter le champ de vue d'un système de vision au point de le rendre omnidirectionnel peuvent être classées en trois catégories [Benosman01], [Yagi99a], [Nayar97b] [Barbosa04a] :

- l'utilisation de plusieurs images pour former un panorama,
- l'utilisation d'objectifs grands angles,
- l'utilisation d'un miroir.

### 1.3.2.1. Fusion de plusieurs images

L'utilisation de plusieurs images pour former une image panoramique est une idée intuitive qui peut sembler simple, mais qui en pratique pose de nombreux problèmes. Deux techniques peuvent être mises en œuvre pour l'acquisition des images à fusionner : la rotation d'une caméra autour d'un axe et l'utilisation de plusieurs caméras.

#### Rotation d'une caméra

La rotation d'une caméra autour d'un axe vertical permet d'obtenir une image panoramique assez simplement. Bien qu'il soit possible d'utiliser une caméra matricielle, la caméra utilisée dans un tel système est en général linéaire puisqu'il suffit alors de concaténer les lignes verticales pour former l'image panoramique [Benosman96a]. Ce type de capteur est appelé capteur cylindrique car il peut être modélisé par un cylindre sur lequel sont projetés les points 3D de la scène comme l'illustre la figure suivante :

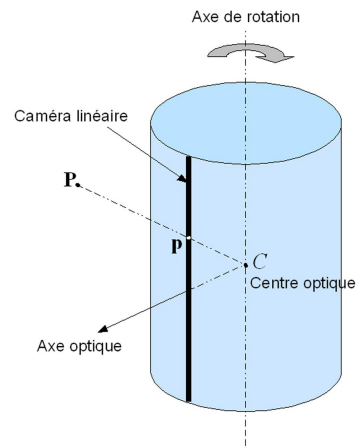


Figure 31: Capteur cylindrique.

Le principal avantage du capteur cylindrique est la résolution de l'image panoramique qu'il est possible d'obtenir, car celle-ci dépend du pas de rotation et non de la caméra elle-même. Si le système mécanique et le système de contrôle de la rotation sont précis, il est possible d'obtenir des images de haute résolution. La figure 32 représente une image acquise par un tel capteur, ainsi qu'un zoom effectué sur une petite région pour mettre en évidence la haute résolution des images acquises par ce type de capteur.



Figure 32: Image acquise par un capteur cylindrique et zoom sur une région.

La résolution de cette image est de 9 000 x 59 000 pixels.

Source [Panoscan]

L'utilisation d'un tel système est cependant limitée à des scènes statiques et à des applications ne fonctionnant pas en temps réel, car l'acquisition est séquentielle et nécessite un temps important. En effet, le temps nécessaire pour effectuer un pas, l'attente pour stabiliser le système, et le temps d'acquisition de la ligne par le capteur, multipliés par le nombre de pas à

effectuer pour couvrir les 360°, conduisent à des temps d'acquisition de l'image panoramique de l'ordre de la minute, voire plusieurs minutes en fonction du pas de rotation choisi.

### Systèmes multi-caméras

L'augmentation du nombre de caméras permet d'obtenir une acquisition du panorama instantanément, contrairement à la solution présentée précédemment. Le recouvrement des champs de vue des différentes caméras permet en effet d'obtenir une image panoramique de l'environnement comme l'illustre la figure 33.

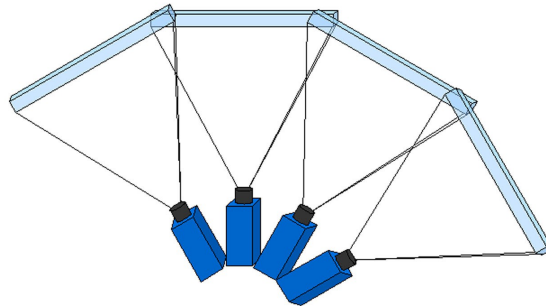


Figure 33: Principe du système multi-caméras.

Quelques systèmes fondés sur ce principe sont commercialisés : le capteur Ladybug2, développé par *Point Grey Research* et la série de capteurs *Astro* mis au point par la société *ViewPlus*.

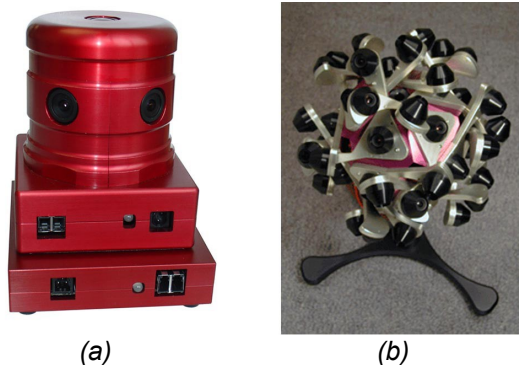


Figure 34: Systèmes multi-caméras.  
(a) Ladybug2 - (b) Astro Sensor

Ces systèmes multi-caméras sont cependant difficiles à mettre en œuvre pour plusieurs raisons : l'importance du flux de données à traiter, la difficulté de synchroniser toutes les caméras, et la calibration du système délicate à effectuer. De plus, la multiplication du nombre de caméras a un impact direct sur le prix du système.

### 1.3.2.2. Utilisation d'un objectif grand angle

Un objectif grand angle est un objectif doté d'une courte focale et qui par conséquent couvre un grand angle de champ. Il n'y a pas de définition précise de l'angle de champ minimal à partir duquel un objectif est qualifié de « grand angle », mais il est courant de prendre comme valeur limite celle de la vision humaine : environ  $50^\circ$ . Ce type d'objectif est utilisé en photographie pour permettre un cadrage large d'objets rapprochés ou pour photographier un paysage dans sa globalité. Les objectifs grand angles peuvent atteindre un angle de champ de  $122^\circ$ , mais au dessus de cette limite les défauts optiques deviennent trop importants pour être corrigés (profondeur de champ restreinte, fortes distorsions, *etc...*).

La vision ne peut être qualifiée d'omnidirectionnelle que si l'angle de champ couvre au moins  $180^\circ$ , ce qui correspondrait à un objectif grand angle de focale nulle. Un tel objectif n'est évidemment pas réalisable, puisqu'il ne s'agit que d'un concept mathématique. Cependant, certains objectifs permettent d'obtenir un angle de champ de  $180^\circ$  : il s'agit des objectifs *fish-eye* (œil de poisson).

Un objectif *fish-eye* est un objectif qui permet l'acquisition d'une image hémisphérique par déviation des rayons lumineux. Pour obtenir un tel angle de champ, les images obtenues ne sont pas corrigées contre les distorsions et sont donc fortement déformées (*cf* figure 35). L'analyse de ces images est alors plus difficile qu'avec une caméra perspective munie d'un objectif classique, notamment pour la détection de droites. La caméra *fish-eye* (association d'une caméra perspective et d'un objectif *fish-eye*) pose des problèmes majeurs en vision par ordinateur car elle ne possède pas de centre de projection unique et est très difficile à modéliser.



(a)



(b)

Figure 35: (a) Objectif *fish-eye*, (b) Image acquise par une caméra *fish-eye*.

### 1.3.2.3. Utilisation d'un capteur catadioptrique

Un capteur catadioptrique (*cf* figure 36) est l'association d'une caméra et d'un miroir ayant en général un axe de révolution. Le nom de ce système est issu de *catoptrique* qui fait référence à la réflexion de la lumière sur le miroir, et de *dioptrique* qui désigne la réfraction de la lumière dans la lentille de la caméra.





Figure 36: Exemple de capteur catadioptrique.

Les rayons lumineux provenant de toutes les directions sont d'abord réfléchis sur le miroir puis projetés sur le capteur de la caméra. On obtient alors une image de l'environnement sur 360°, d'où une vision qualifiée d'omnidirectionnelle. Ces capteurs permettent de plus une acquisition instantanée de tout l'environnement, ce qui les destine aux applications devant fonctionner dans des environnements dynamiques.

Le premier capteur catadioptrique est apparu dans un brevet déposé par Rees en 1970 [Rees70a]. Le système qui y est décrit est composé d'une lentille perspective et de deux miroirs, l'un hyperboloïde et l'autre ellipsoïde, afin de fournir une vue à 360° de l'environnement à un opérateur situé dans un véhicule.

Il faudra cependant attendre les années 90 pour que les capteurs catadioptriques deviennent un sujet de recherche actif, notamment grâce aux articles de Nayar et Baker [Nayar97a] [Baker99a] qui expliquent la théorie de la formation des images avec ce type de capteur. La recherche dans ce domaine s'est alors accélérée du fait des nombreux avantages que présentent ces capteurs dans des domaines tels que la surveillance et la navigation.

### **La contrainte du point de vue unique (SVP)**

Lors de la conception d'un système de vision, il est fortement recommandé d'avoir un seul point de vue effectif, encore appelé **centre de projection**, pour pouvoir obtenir des images géométriquement correctes, c'est-à-dire purement perspectives [Rees70a] [Baker99a]. Dans ce cas, chaque pixel du plan image mesure la quantité de lumière qui passe par ce point unique et dans une direction donnée, la connaissance de la géométrie du capteur permettant de calculer cette direction. Les systèmes qui ne possèdent ainsi qu'un seul centre de projection sont appelés **capteurs centraux**.

La formation des images d'un capteur catadioptrique central s'appuie sur la théorie du point de vue unique (*Single View-Point*, SVP) développée par Nayar et Baker [Baker99a]. Dans cet



article, ils utilisent les lois de la réflexion sur un miroir ponctuel pour exprimer la contrainte du point de vue unique. La résolution de cette contrainte permet d'aboutir à deux solutions générales, pour lesquelles il est possible de déterminer des solutions particulières. Ces dernières définissent les formes des miroirs et les contraintes de positionnement à respecter entre la caméra et le miroir pour obtenir ce point de vue unique.

Les formes de miroirs respectant la contrainte du point de vue unique sont les miroirs plans, coniques, ellipsoïdes, sphériques, hyperboloïdes et paraboloides. Cependant, les contraintes de positionnement à respecter entre la caméra et le miroir produisent parfois des configurations impossibles à utiliser. Il faut alors modifier la position de la caméra, ce qui fait perdre la propriété de projection centrale.

### Le miroir plan

Le miroir plan possède un point de vue unique situé sur l'axe optique et à une distance égale à celle séparant le miroir du centre optique de la caméra (*cf* figure 37).

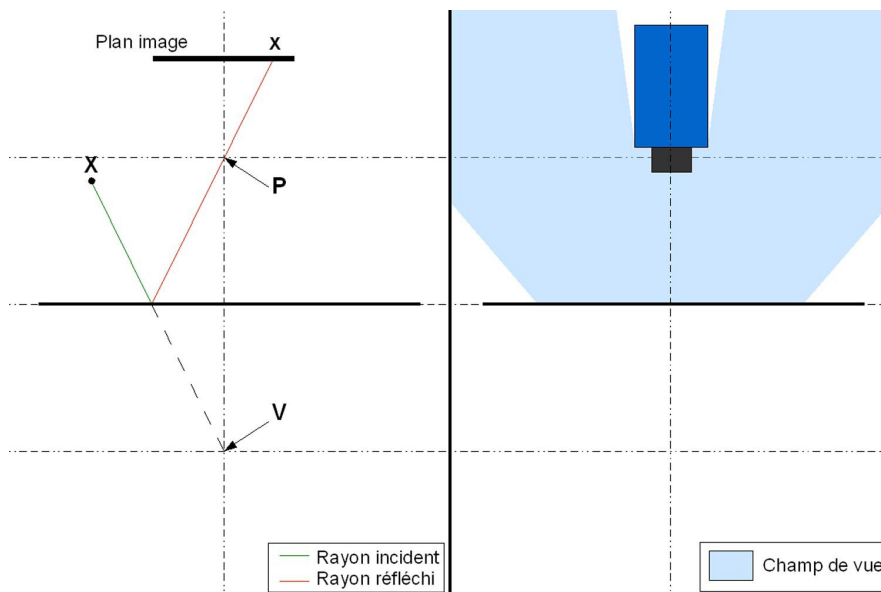


Figure 37: Miroir plan.

Le miroir plan ne permet cependant pas d'augmenter l'angle de champ puisque celui-ci sera identique à celui de la caméra. De plus, il est impossible d'associer plusieurs miroirs plans devant une seule caméra perspective pour augmenter l'angle de champ, car dans ce cas l'unicité du point de vue n'est plus respectée. L'utilisation de plusieurs miroirs plans et d'une seule caméra perspective présente néanmoins des avantages puisque cette configuration permet d'obtenir plusieurs points de vue d'une même scène avec une unique caméra. Un tel système est alors

particulièrement intéressant pour obtenir une vue stéréoscopique avec une seule caméra, ce qui permet de s'affranchir des problèmes de synchronisation et de réponse spectrale des caméras [Duvieubourg06a].

Pour obtenir une vision omnidirectionnelle tout en conservant un point de vue unique, il est possible de multiplier le nombre de miroirs à condition d'associer une caméra à chaque miroir. Ainsi, Nalwa [Nalwa96a] [Nalwa99a] a conçu le premier système de vision panoramique fondé sur cette idée. Son système est composé de quatre miroirs plans disposés en pyramide et de quatre caméras CDD pour obtenir une image panoramique ayant un angle de champ de  $360^\circ \times 50^\circ$ . L'avantage de la multiplication du nombre de caméras est l'augmentation de la résolution des images mais elle introduit également de fortes contraintes de positionnement et de synchronisation des caméras.

### Le miroir conique

La contrainte du point de vue unique impose de positionner le centre optique de la caméra **P** sur le sommet du cône (cf figure 38), ce qui est impossible à réaliser. Dans la pratique, la caméra est éloignée du sommet du miroir conique, ce qui conduit à la perte du point de vue unique mais rend le système exploitable.

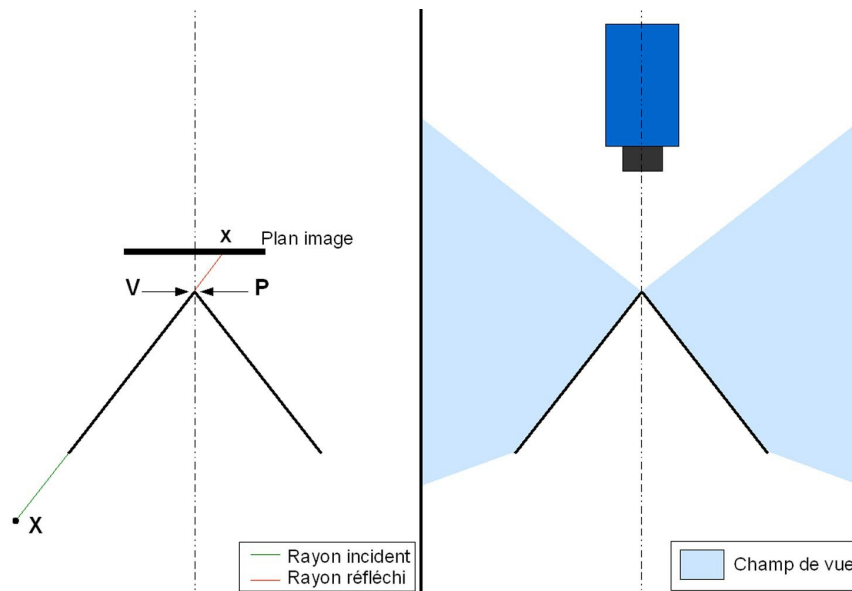


Figure 38: Miroir conique.

Le miroir conique a cependant été utilisé dans de nombreux systèmes de vision car la contrainte du point de vue unique n'est pas toujours nécessaire pour certaines applications comme la navigation autonome [Yagi90a] [Yagi91a] [Caglioti05a]. Ce miroir étant facile à

usiner, il a, de plus, l'avantage d'avoir un coût très faible.

### Le miroir ellipsoïde

Le miroir ellipsoïde respecte la contrainte du point de vue unique lorsque le centre optique **P** de la caméra est situé sur le second foyer de l'ellipsoïde, le premier étant le point de vue **V** (cf figure 39).

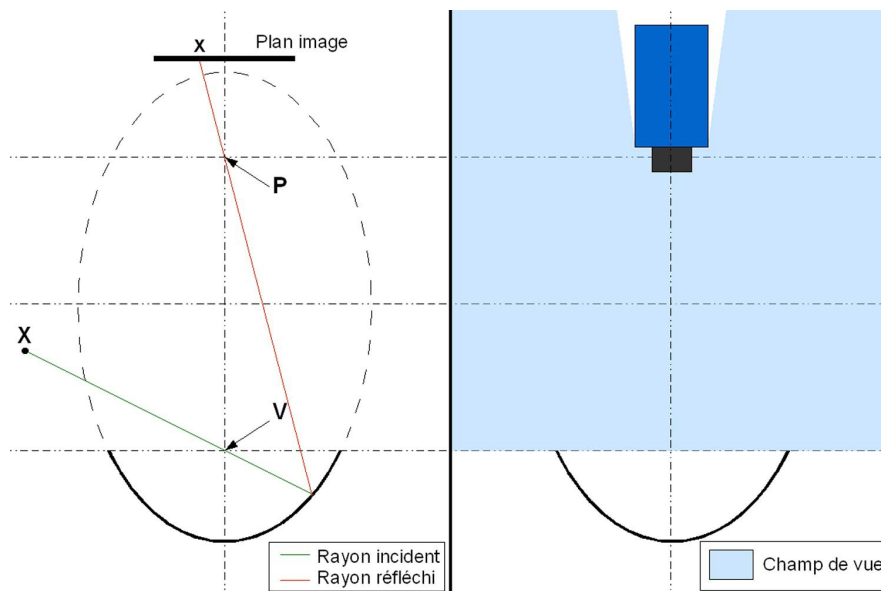


Figure 39: Miroir ellipsoïde.

Ce miroir permet d'augmenter le champ de vue significativement puisque celui-ci est hémisphérique lorsque le miroir correspond à la section d'ellipse située sous le point de vue **V**.

Ce miroir est cependant très peu utilisé car son angle de champ est plus faible que d'autres formes de miroir (comme les miroirs hyperboloïdes et paraboloides), et sa forme concave le rend plus difficile à usiner.

### Le miroir sphérique

Le point de vue unique **V** et le centre optique **P** sont confondus au centre de la sphère (cf figure 40). Dans cette configuration, ce système n'a pas d'intérêt car la caméra n'observe que sa propre image.

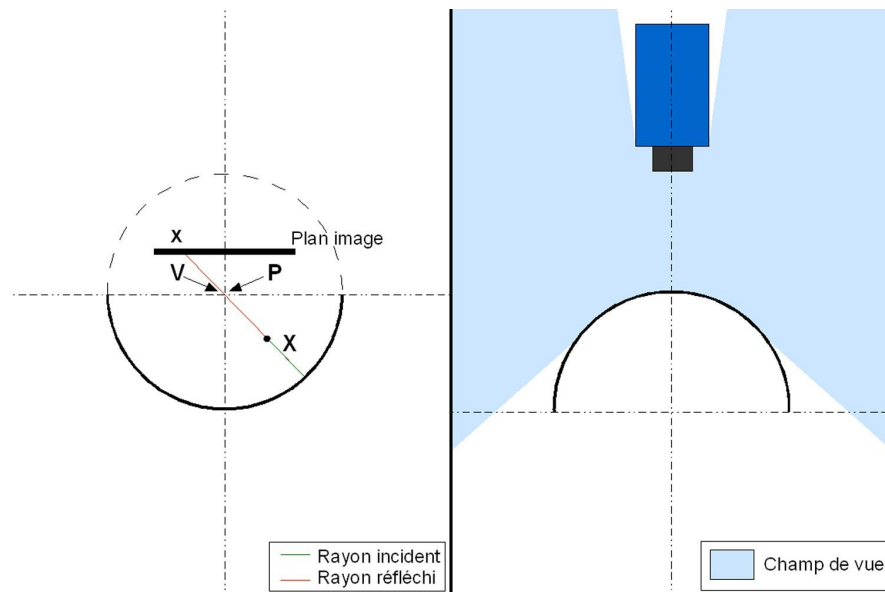


Figure 40: Miroir sphérique.

Le miroir sphérique peut néanmoins être utilisé en éloignant la caméra du centre de la sphère pour obtenir un angle de champ très important, mais le système perd alors son point de vue unique.

Le miroir sphérique peut également être utilisé pour construire un système orthographique sans utiliser de lentille télécentrique (voir la description du miroir parabolöide p54).

#### **Le miroir hyperboloïde :**

Le miroir hyperboloïde respecte la contrainte du point de vue unique lorsque le centre optique **P** de la caméra est situé sur le second foyer de l'hyperbole, le premier étant le point de vue **V** (cf figure 41). Cette condition nécessite donc de faire coïncider le centre optique de la caméra avec le second foyer de l'hyperbole, et de superposer l'axe optique de la caméra et l'axe de révolution du miroir.

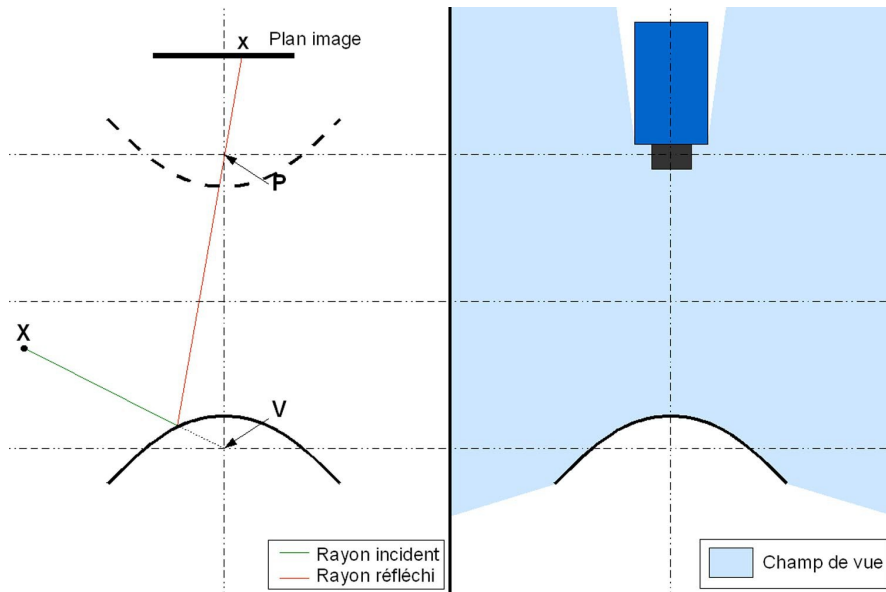


Figure 41: Miroir hyperboloïde.

Ce miroir a été réalisé pour la première fois par Rees [Rees70a] et a ensuite été utilisé dans de nombreux systèmes dédiés à la navigation [Yamazawa93a] [Yachida98a] [Bunschoten03a].

**Le miroir parabololoïde**

Le miroir parabololoïde réalise une projection orthographique, c'est-à-dire que tous les rayons passant par le point de vue **V** sont réfléchis parallèlement à l'axe optique (cf figure 42). Il faut alors utiliser une caméra orthographique, c'est-à-dire mettre une lentille télécentrique devant la caméra pour projeter ces rayons sur le plan image.

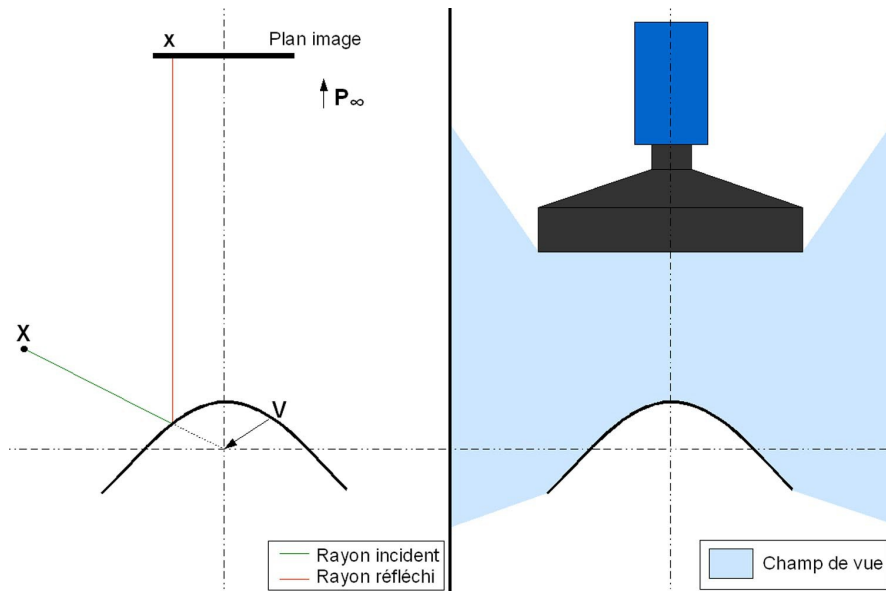


Figure 42: Miroir parabololoïde.

L'utilisation d'un miroir parabololoïde associé à une lentille télécentrique permet de contourner la contrainte de positionnement de la caméra par rapport au miroir car il y a

invariance des images par translation le long de l'axe de révolution [Nayar97c]. Cette propriété a conduit au développement de nombreux systèmes fondés sur l'utilisation d'un miroir parabolöide et d'une lentille télécentrique [Mouaddib05c] [Geyer02a].

L'association d'un miroir parabolöide et d'une lentille télécentrique permet donc d'alléger les contraintes de positionnement, mais ceci au détriment de l'encombrement. Une lentille télécentrique a en effet une taille et une distance minimale de travail plus élevées qu'une lentille classique. Pour contourner ce problème, il est possible d'associer un miroir parabolöide à une caméra perspective en utilisant un miroir sphérique ou un miroir parabolöide concave supplémentaire [Barbosa04a] [Orghidan05b]. Le problème de positionnement ressurgit alors car il faut placer avec précision le second miroir par rapport au premier et à la caméra.

### **Les miroirs non-centraux**

L'article de Baker [Baker99a] a permis de démontrer qu'il n'existe pas d'autre forme de miroir possédant un point de vue unique. Cependant, de nombreux autres profils de miroirs ont été proposés pour répondre à certaines propriétés (résolution verticale constante, résolution centrale plus importante, *etc*), ou pour répondre à des applications spécifiques, comme l'observation simultanée de la scène routière et du visage du conducteur décrite par Layerle [Layerle08a].

Nous ne nous intéresserons pas à ces familles car ces miroirs ne sont pas les plus adaptés à la reconstruction 3D en temps réel de par la complexité de leur modélisation et l'impossibilité de trianguler les points à partir des centres de projection (ce point est abordé plus en détails dans la section 1.4.3.1).

## **1.4. Conception d'un système de vision 3D**

### **1.4.1. Introduction**

Cette étude étant destinée à une application militaire, il était important de développer un système qui soit le plus discret possible, l'idéal étant d'être indétectable. Cette contrainte exclut l'utilisation d'un capteur actif qui émet de l'énergie (onde sonore, lumineuse, *etc*) pour obtenir des informations sur son environnement. L'utilisation d'un système de vision, capable de fournir des informations sans agir sur son environnement, a donc été privilégiée.

Parmi les systèmes de vision, les systèmes omnidirectionnels, et plus particulièrement les capteurs catadioptriques, présentent des caractéristiques particulièrement intéressantes pour la reconstruction 3D et la navigation. Leur champ de vue omnidirectionnel permet en effet de suivre des points d'intérêts sur une longue séquence d'images alors que ceux-ci seraient

rapidement sortis du champ de vue d'une caméra classique. Le flot optique est souvent analysé pour estimer les mouvements du système, mais il existe une ambiguïté entre la translation et la rotation dans le cas de la vision classique qui peut être levée lors de l'utilisation d'un système omnidirectionnel [Mouaddib05a]. Enfin, ils permettent l'acquisition instantanée de la totalité de l'environnement, et peuvent par conséquent fonctionner dans un environnement dynamique.

L'utilisation d'une seule méthode de reconstruction 3D (*cf Shape-From-X*, section 1.2.2) permet difficilement de retrouver la structure 3D complète de la scène. Pour cette raison, nous proposons de combiner les techniques de *Shape-From-Stereo* et de *Shape-From-Motion*, qui sont les méthodes les plus adaptées à la robotique mobile et à la reconstruction 3D de scènes complexes.

Nous avons développé un capteur de stéréovision omnidirectionnelle, composé de deux capteurs catadioptriques, pour tirer partie des avantages de la vision omnidirectionnelle et de la stéréovision.

### **1.4.2. Choix du capteur catadioptrique**

#### **1.4.2.1. Choix des miroirs**

Nous avons présenté dans la section 1.3.2.3 les différentes formes de miroir respectant la contrainte du point de vue unique. Le respect de cette contrainte est important pour s'assurer que le capteur soit central, et par conséquent facilement modélisable et permettant la triangulation.

De nombreuses formes de miroirs respectent en théorie la contrainte du point de vue unique, mais le positionnement de la caméra par rapport au miroir n'est pas toujours réalisable : caméra au centre du miroir sphérique, sur le sommet du miroir conique, *etc.* Seules deux configurations sont utilisables en pratique : le miroir parabolöide et le miroir hyperbolöide.

Le dispositif composé d'un miroir parabolöide et d'une lentille télécentrique est particulièrement intéressant puisqu'il permet de s'affranchir de certaines contraintes de positionnement entre la caméra et le miroir. Les lentilles télécentriques ont cependant l'inconvénient d'être plus encombrantes et d'avoir une distance minimale de travail plus importante que les lentilles perspectives. Un système stéréoscopique composé de deux capteurs catadioptriques à miroirs parabolöides serait très difficilement embarquable sur un robot mobile, comme l'illustre la figure 43.

Nous avons préféré privilégier la compacité du capteur au détriment de la facilité d'assemblage en utilisant un miroir hyperbolöide. Pour respecter la contrainte du point de vue unique, il faut alors placer la caméra précisément par rapport au foyer du miroir comme ceci est

expliqué à la section 1.4.2.3.

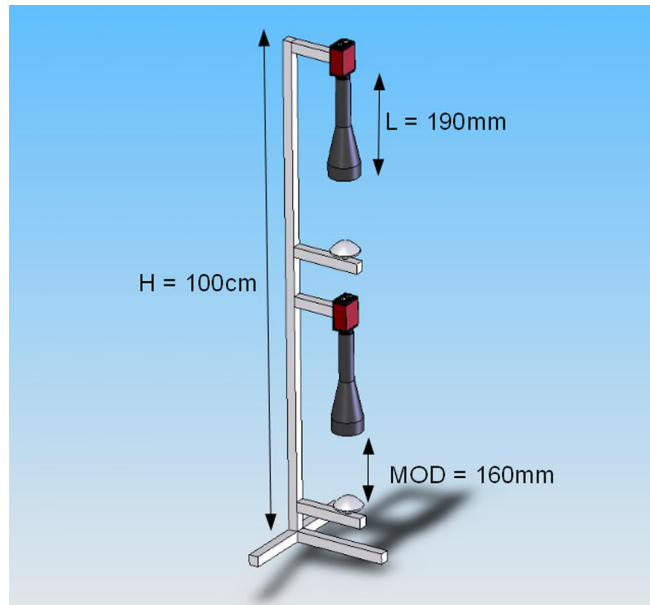


Figure 43: Encombrement d'un système stéréoscopique utilisant des miroirs paraboloides.  $L$  est la taille de la lentille télécentrique,  $MOD$  est la distance minimale de travail, et  $H$  est la hauteur totale du système stéréoscopique.

Un miroir hyperboloïde est caractérisé par ses semi-axes  $a$  et  $b$ , et donc indirectement par son excentricité  $e = \sqrt{a^2 + b^2}$  (voir figure 44). L'équation de la surface du miroir hyperboloïde, dans un repère centré en  $\mathbf{O}$ , est donnée par :

$$\frac{z^2}{a^2} - \frac{x^2 + y^2}{b^2} = 1 \quad (8)$$

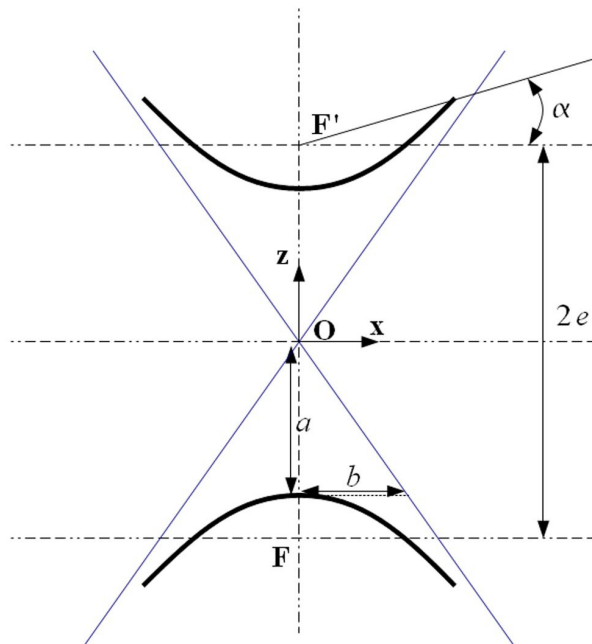


Figure 44: Paramètres d'une hyperbole.



Les paramètres  $a$  et  $b$  permettent de faire varier l'angle de vue  $\alpha$  représenté sur la figure 44. L'angle de vue  $\alpha$  est à choisir en fonction de l'application ciblée : dans le cas de la navigation autonome pure, il est préférable d'observer au maximum le sol pour détecter les obstacles locaux alors que pour la modélisation d'un environnement en trois dimensions, la zone d'intérêt est située au dessus du robot (portes, fenêtres, limites murs/plafond, etc). La figure 45 donne un aperçu des images obtenues pour différents angles de vue. Ces images sont issues d'un capteur Accowle<sup>3</sup>, et pour une configuration telle que le miroir est situé au dessus de la caméra ; le miroir correspond donc à la nappe supérieure de l'hyperbole de la figure 44.

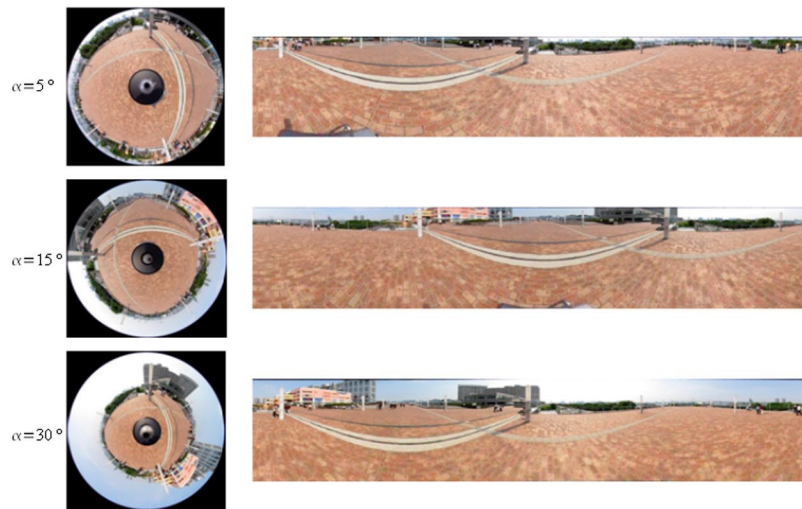


Figure 45: Comparaison de différents champs de vue.  
Images obtenues par des capteurs Accowle.

Le capteur étant dédié à la fois à des applications de navigation et de reconstruction 3D, la configuration retenue est une caméra placée au dessus du miroir (cf figure 36) pour observer la partie supérieure, et un angle  $\alpha$  de l'ordre de  $15^\circ$  sous l'horizontale pour observer les obstacles locaux autour du robot mobile.

Le miroir retenu est un miroir H3S provenant de la société Neovision<sup>4</sup>, qui a un champ de vue de  $360^\circ \times 106^\circ$  ( $\alpha = 16^\circ$ ), et dont les paramètres sont spécifiés dans le tableau 1.

Semi-axe $a$	28,0950 mm
Semi-axe $b$	23,4125 mm
Excentricité $e$	36,5715 mm
Diamètre	60 mm

Tableau 1: Paramètres du miroir hyperboloïde.

<sup>3</sup> <http://www.accowle.com/english/>

<sup>4</sup> <http://www.neovision.cz/>

### 1.4.2.2. Choix des caméras et des optiques

Le choix de la caméra et de l'optique doit se faire en adéquation avec le miroir utilisé afin d'obtenir la meilleure qualité d'image possible. La caméra retenue pour notre application est une caméra CCD couleur GC1380C de Prosilica<sup>5</sup>, qui a une résolution de 1360x1024 pixels, et la particularité d'avoir un capteur 2/3". La taille du capteur a un impact sur celle des photo-sites, et par conséquent sur la sensibilité de la caméra.

L'optique doit être choisie en fonction des paramètres du miroir et de ceux de la caméra. Le premier paramètre à déterminer est la longueur focale  $f$  de l'optique afin d'optimiser la taille du miroir dans l'image. Soient  $a$ ,  $b$  et  $e$  les paramètres de l'hyperbole,  $R$  le rayon du miroir, et  $h$  la hauteur du CCD, la focale optimale est alors donnée par :

$$f = \frac{\frac{h}{2} \left( e + \sqrt{a^2 \left( 1 + \frac{R^2}{b^2} \right)} \right)}{R} \quad (9)$$

Avec les paramètres de notre miroir (cf tableau 1) et de notre caméra ( $h=6,6$  mm pour un capteur 2/3"), la focale optimale est  $f=9,05$  mm, soit une focale standard de 9 mm.

Le second paramètre à prendre en compte lors du choix d'une optique est sa distance minimale de travail (MOD<sup>6</sup>), c'est-à-dire la distance entre l'optique et l'objet à partir de laquelle l'image sera nette. Nous avons donc choisi une optique Fujinon<sup>7</sup> HF9HA-1B qui a l'avantage d'avoir une distance minimale de travail faible, de l'ordre de 10 cm, et une mise au point réglable.

La distance entre le miroir et la caméra est faible, ce qui a pour conséquence de réduire considérablement la profondeur de champ. Pour obtenir des images nettes sur la totalité du miroir, il est nécessaire de réduire l'ouverture (le diaphragme) de l'objectif, et donc la quantité de lumière atteignant le CCD. La grande sensibilité du capteur 2/3" permet alors de conserver un temps d'intégration raisonnable.

### 1.4.2.3. Respect de la contrainte du point de vue unique

L'utilisation d'un capteur catadioptrique à base de miroir hyperboloïde impose de fortes contraintes d'assemblage. Pour respecter la contrainte du point de vue unique, le centre optique de la caméra, c'est-à-dire le lieu de convergence de tous les rayons lumineux, doit coïncider avec le foyer  $F'$  de la nappe virtuelle de l'hyperbole, comme l'illustre la figure 46. Le centre optique de la caméra doit donc être situé sur l'axe de révolution du miroir, et à une distance  $2e$  du foyer

5 <http://www.prosilica.com/>

6 MOD = *Minimum Of Distance*

7 <http://www.fujinon.com/>

F du miroir.

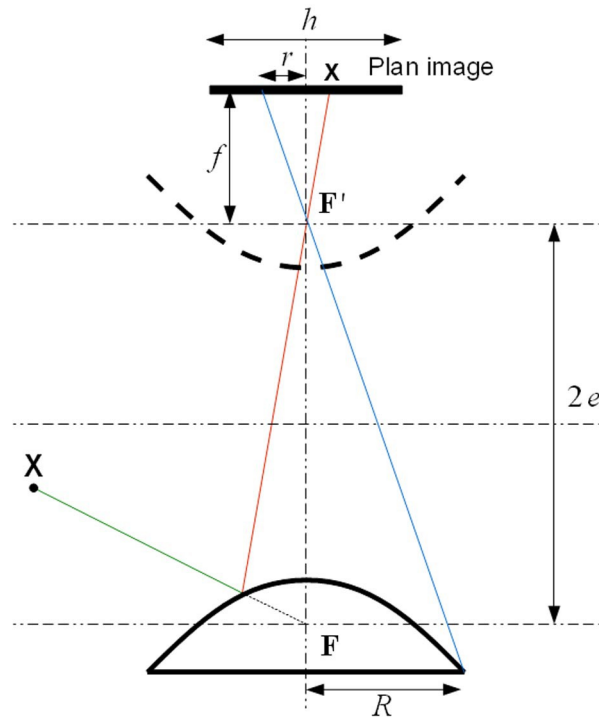


Figure 46: Formation des images avec un miroir hyperboloïde.

Dans la pratique, il est très difficile d'identifier la position du foyer du miroir et le centre optique de la caméra. Nous avons donc mis en place une procédure pour respecter au mieux cette contrainte. Notre méthode consiste à déterminer la projection théorique du miroir sur le plan image lorsque la contrainte du point de vue unique est respectée. Dans ce cas, la projection du miroir sur le plan image est un cercle de rayon  $r$ , qui peut être déterminé grâce à la connaissance des paramètres de la caméra et de ceux du miroir :

$$r = \frac{k f R}{e + \sqrt{a^2 \left(1 + \frac{R^2}{b^2}\right)}} \quad (10)$$

où  $k$  est le nombre de pixels par unité de longueur du capteur de la caméra,  $f$  sa focale,  $a$  et  $b$  les semi-axes du miroir,  $e$  son excentricité et  $R$  son rayon. Les paramètres du miroir sont donnés par le fabricant ; ceux de la caméra sont obtenus en utilisant un outil de calibrage tel que celui de Bouguet<sup>8</sup>.

La projection théorique du miroir est alors tracée, puis la position de la caméra est ajustée pour superposer le bord extérieur du miroir avec le cercle théorique, comme le montre la figure 47.

<sup>8</sup> [http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html)

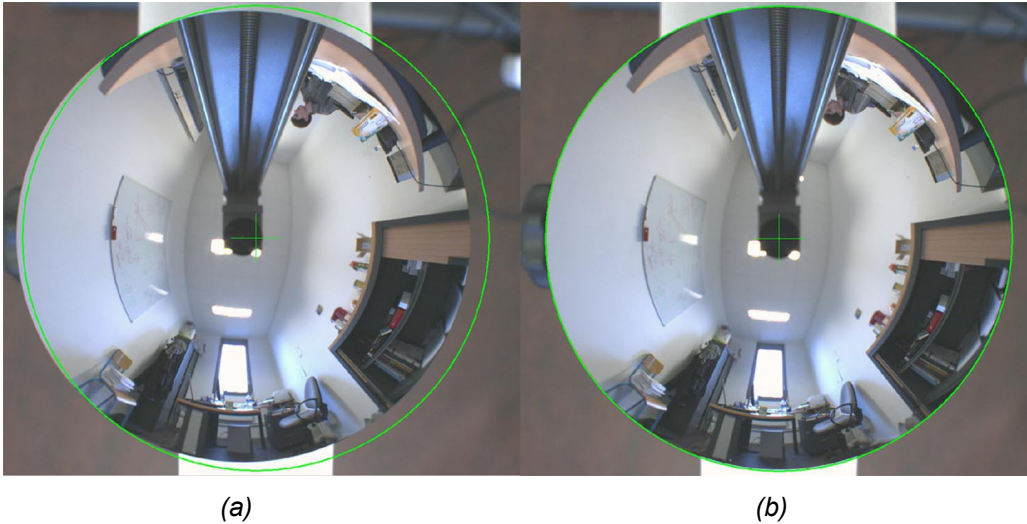


Figure 47: Ajustement de la position du miroir pour respecter la contrainte du point de vue unique.  
 (a) Contrainte du point de vue unique non respectée – (b) Contrainte du point de vue unique respectée.

### 1.4.3. Étude de la configuration des deux capteurs

Les deux capteurs catadioptriques sont maintenant parfaitement définis et assemblés de sorte à respecter au mieux la contrainte du point de vue unique, ce qui permet d'obtenir une projection centrale. Les deux capteurs doivent maintenant être positionnés l'un par rapport à l'autre pour former un capteur stéréoscopique. La position relative des deux capteurs doit être choisie judicieusement pour simplifier les algorithmes de reconstruction 3D et pour obtenir la meilleure précision possible.

Le cadre théorique de la configuration d'un capteur stéréoscopique est celui de la **géométrie épipolaire**. Nous présentons dans un premier temps le principe et la terminologie de la géométrie épipolaire pour les capteurs stéréoscopiques classiques, puis nous généralisons celle-ci aux capteurs catadioptriques.

#### 1.4.3.1. Géométrie épipolaire

##### Géométrie épipolaire d'un capteur stéréoscopique classique

Soient  $\mathbf{M}$  un point de l'espace,  $\mathbf{m}$  sa projection sur le premier plan image et  $\mathbf{m}'$  sa projection sur le second plan image. Comme l'illustre la figure 48, le point  $\mathbf{M}$ , ses projections  $\mathbf{m}$  et  $\mathbf{m}'$ , et les centres des caméras  $\mathbf{C}$  et  $\mathbf{C}'$  sont coplanaires. Cette propriété, appelée **contrainte de coplanarité**, est à la base de la géométrie épipolaire. La géométrie épipolaire joue un rôle fondamental dans la stéréovision puisqu'elle permet de faciliter l'étape la plus critique : la mise en correspondance des pixels.

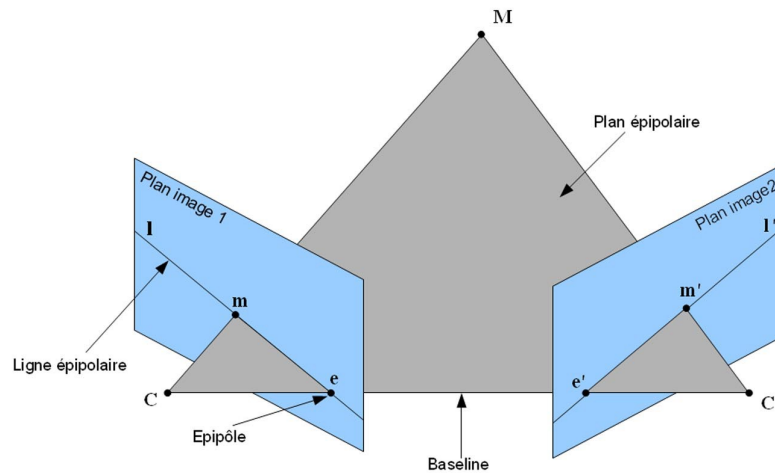


Figure 48: Géométrie épipolaire d'un système stéréoscopique classique.

La connaissance de la géométrie des deux capteurs permet en effet de contraindre la zone de recherche lors de la mise en correspondance des pixels. Étant donné un point  $m$  sur la première image, les coordonnées de son correspondant  $m'$  ne peuvent être déterminées car la position du point  $M$  ne peut être connue. Cependant,  $m$  étant la projection de  $M$  sur le premier plan image,  $M$  appartient à la droite  $(Cm)$ . Le point  $m'$  appartient par conséquent à la projection de cette droite sur la seconde image : il s'agit de la droite  $l'$  de la figure 48. La recherche des stéréo-correspondants devient alors un problème unidimensionnel et non plus bidimensionnel.

Un **épipôle**  $e$  est le point d'intersection de la *baseline*, c'est-à-dire la droite reliant les centres des caméras, avec le plan image. Un **plan épipolaire** est un plan contenant la *baseline*. Une **ligne épipolaire**  $l$  est l'intersection d'un plan épipolaire avec le plan image. Toutes les lignes épipolaires se coupent à l'épipôle  $e$ .

### Géométrie épipolaire d'un capteur stéréoscopique catadioptrique

La géométrie épipolaire des capteurs stéréoscopiques catadioptriques est plus complexe que celle des caméras classiques. Les premiers travaux portant sur l'adaptation de la géométrie épipolaire aux capteurs catadioptriques dans une configuration quelconque ont été publiés par Svoboda, Pajdla, et Hlavac [Svoboda98a]. Des travaux antérieurs existent, mais ceux-ci sont consacrés à des configurations particulières, comme ceux Benosman et al. qui proposent un capteur de stéréovision panoramique composé de deux caméras linéaires en rotation autour d'un axe vertical [Benosman96a]. Cette configuration permet d'obtenir une géométrie épipolaire simple, puisque les lignes épipolaires sont alors les colonnes des paires d'images stéréoscopiques.

La géométrie épipolaire peut être transposée aux capteurs catadioptriques en raisonnant

d'abord sur les miroirs puis en projetant les entités (points, droites, courbes, *etc*) sur les plans images. Ainsi, les centres de projection ne sont plus les centres des caméras mais les foyers  $F$  et  $F'$  des miroirs, qui forment avec le point  $M$  un plan épipolaire. La *baseline*, qui est cette fois la droite séparant les deux foyers, coupe chaque miroir en deux points : les **épipôles**. Les lignes épipolaires deviennent des **courbes épipolaires** qui correspondent à l'intersection des plans épipolaires avec les miroirs.

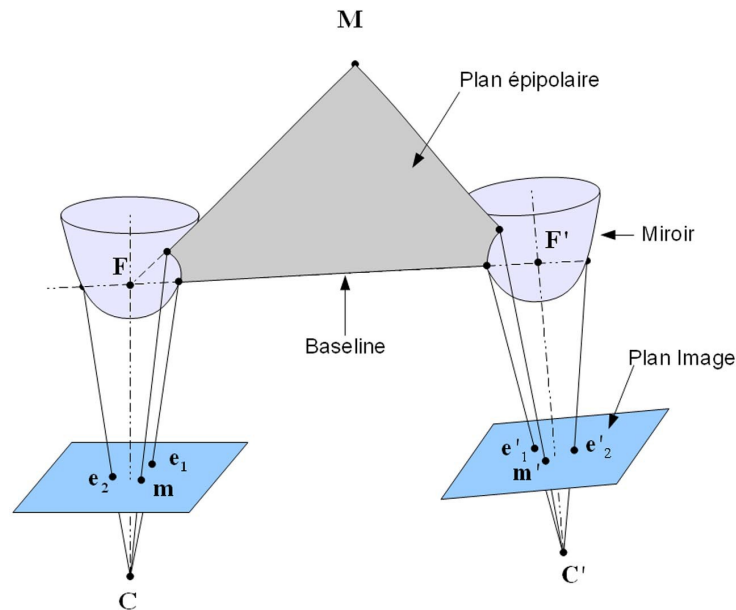


Figure 49: Géométrie épipolaire d'un système stéréoscopique catadioptrique.

Dans une configuration quelconque, la géométrie épipolaire des capteurs catadioptriques est plus complexe que celle des caméras classiques puisque les lignes épipolaires qui permettaient de simplifier la recherche des pixels homologues deviennent des courbes épipolaires, comme l'illustre la figure 50.

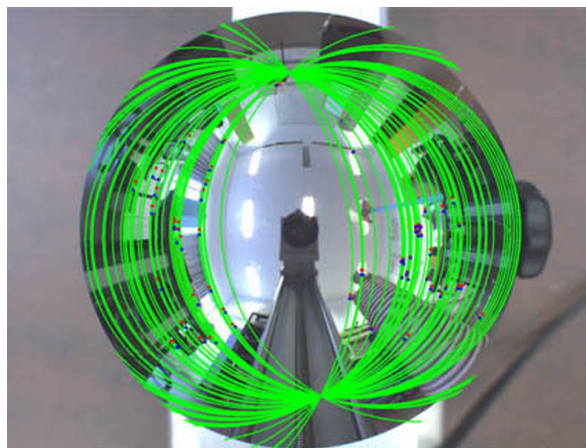


Figure 50: Exemple de courbes épipolaires obtenues par déplacement d'un capteur catadioptrique.

Les capteurs panoramiques peuvent cependant être placés dans une configuration particulière dans le but de simplifier leur géométrie épipolaire. Cette configuration consiste à aligner les axes de révolution des miroirs et des caméras comme l'illustre la figure 51 pour transformer les courbes épipolaires en droites radiales.

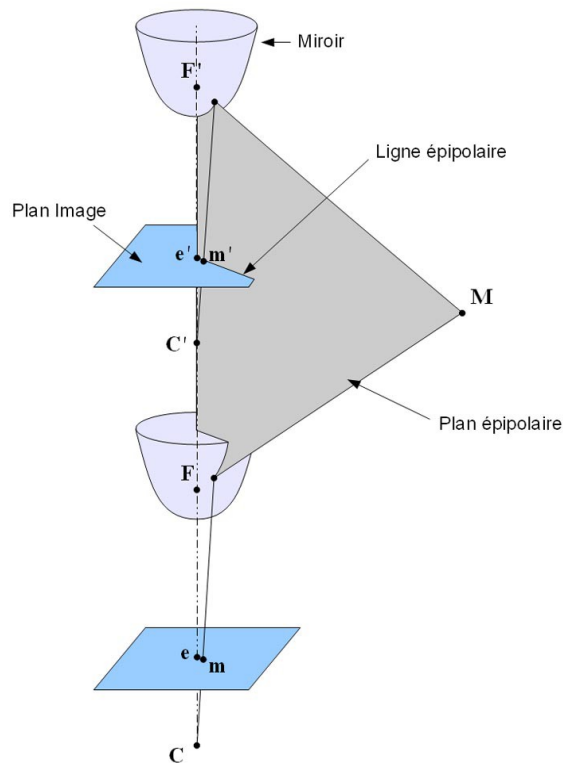


Figure 51: Configuration particulière de capteurs panoramiques simplifiant la géométrie épipolaire.

Nous avons privilégié cette configuration afin de faciliter la mise en correspondance des pixels des deux images de la paire stéréoscopique. L'extraction des pixels appartenant à une droite épipolaire est en effet réalisable plus simplement grâce à des algorithmes tels que celui de Bresenham [Bresenham65a].

#### 1.4.3.2. Étude de la résolution du système stéréoscopique catadioptrique

La résolution du système stéréoscopique catadioptrique est liée d'une part aux différents éléments des capteurs catadioptriques (caméras, miroirs, etc), et d'autre part à la distance séparant ces deux capteurs. Le choix de cette distance, la *baseline*, est donc particulièrement important puisqu'il influe directement sur les performances du système.

L'augmentation de la *baseline* permet de réduire l'erreur commise lors de la triangulation, mais une distance trop grande rend la mise en correspondance plus difficile. En effet, les deux images sont alors prises sous deux points de vues trop différents pour réussir à mettre les points en correspondance. La *baseline* doit donc être choisie judicieusement pour assurer une précision



suffisante sans dégrader l'étape clé de la stéréovision : la mise en correspondance.

Nous avons choisi notre *baseline* en fonction de deux éléments : d'une part la compacité du système, et d'autre part en réalisant des images avec différentes *baseline* pour observer l'influence sur le champ de vue commun des deux caméras et sur la mise en correspondance. Ainsi, avec une *baseline* de 50cm, le champ commun entre les deux caméras devient trop limité (*cf* figure 52), et les points de vues trop différents pour permettre une bonne mise en correspondance. Le capteur du bas (figure 52a) observe par exemple le dessous du bureau situé à gauche alors que la caméra du haut (figure 52b) observe le dessus.

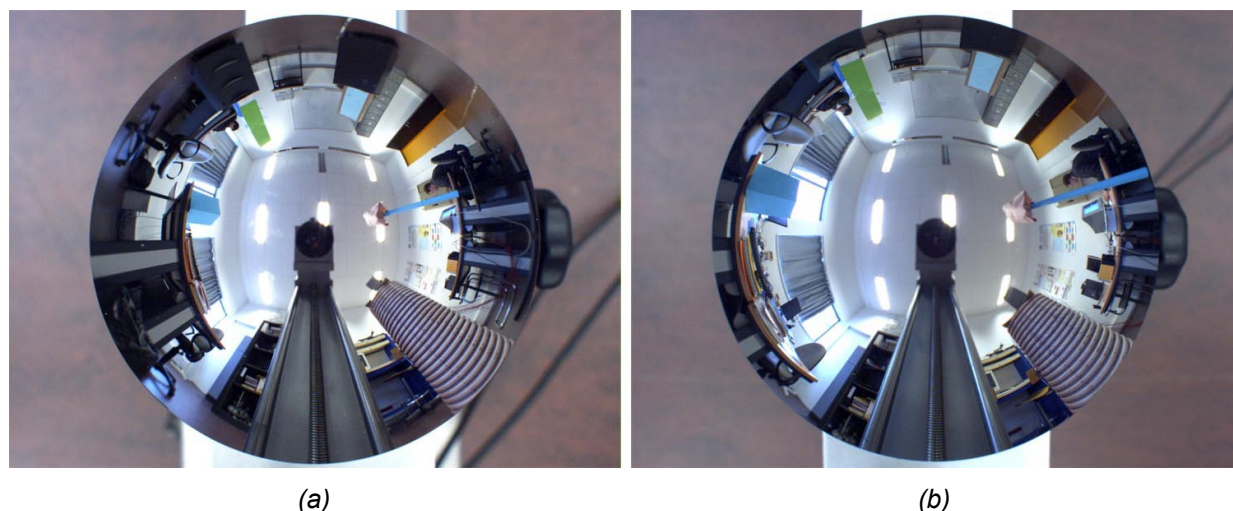


Figure 52: Paire d'images stéréoscopiques obtenue avec une *baseline* de 50cm.  
 (a) Image issue du capteur bas – (b) Image issue du capteur haut.

Le système stéréoscopique étant destiné à être embarqué sur un robot mobile, nous avons privilégié la compacité du capteur en réduisant la *baseline* au minimum. Cette *baseline* minimale, de l'ordre de 22cm, reste tout de même suffisamment importante pour avoir un capteur précis et elle peut être modifiée en fonction du contexte : dans un environnement extérieur avec des objets à reconstruire éloignés, celle-ci peut être augmentée.

La résolution spatiale du capteur a ensuite été évaluée dans une section verticale grâce aux caractéristiques du système (résolution de la caméra, équation du miroir, *baseline*, etc). Chaque pixel de la première image a ainsi été triangulé avec tous les pixels de la seconde image pour obtenir le jeu de points 3D qu'il est possible de reconstruire représenté sur la figure 53. La figure 53a met en évidence la résolution non uniforme du capteur, résolution qui se dégrade rapidement avec la distance. La figure 53b est un zoom sur l'espace proche du capteur qui illustre le champ de vue commun à proximité du capteur ainsi que la bonne résolution à faible distance.



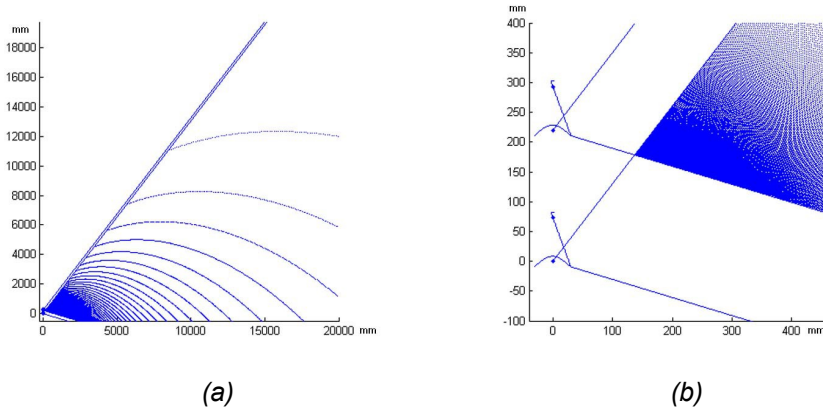


Figure 53: Résolution spatiale du capteur de stéréovision omnidirectionnelle.  
 (a) Vue globale de la résolution spatiale – (b) Zoom sur l'espace proche du capteur.

### 1.4.3.3. Vue d'ensemble du système

Le système d'acquisition 3D complet est présenté sur la figure 54. Il est composé des deux capteurs catadioptriques montés dans une configuration verticale. La *baseline* minimale est d'environ 22cm mais elle peut être augmentée en fonction de l'environnement à reconstruire. La hauteur totale du système est d'environ 40cm. Un lidar a également été embarqué sur le robot dans le but de fournir la vérité terrain qui permettra d'évaluer les performances du système de reconstruction 3D.

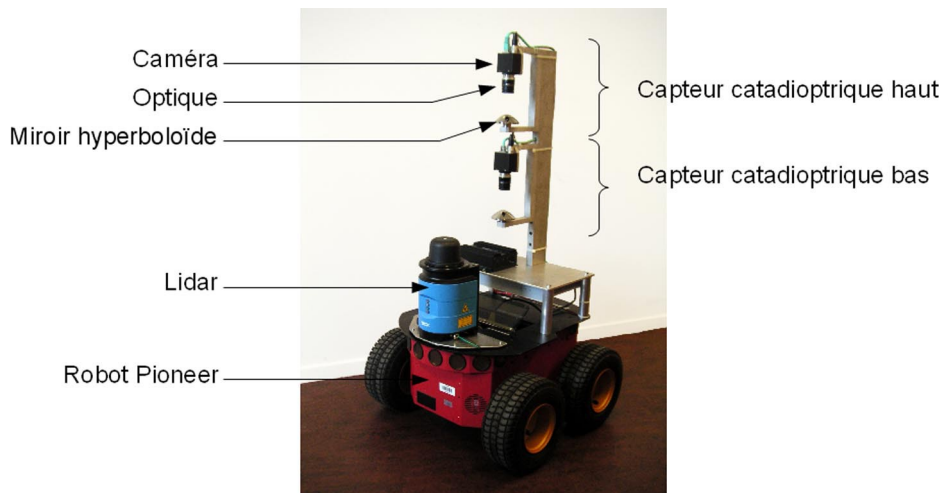


Figure 54: Système d'acquisition 3D embarqué sur un robot Pioneer.

Un logiciel d'acquisition (voir figure 55) des données a également été développé afin de recueillir et de synchroniser les différentes informations provenant du robot :

- les deux images omnidirectionnelles (dont l'une est dépliée pour faciliter l'interprétation de l'environnement par l'utilisateur),
- la carte de l'environnement issue du lidar,
- les données odométriques.

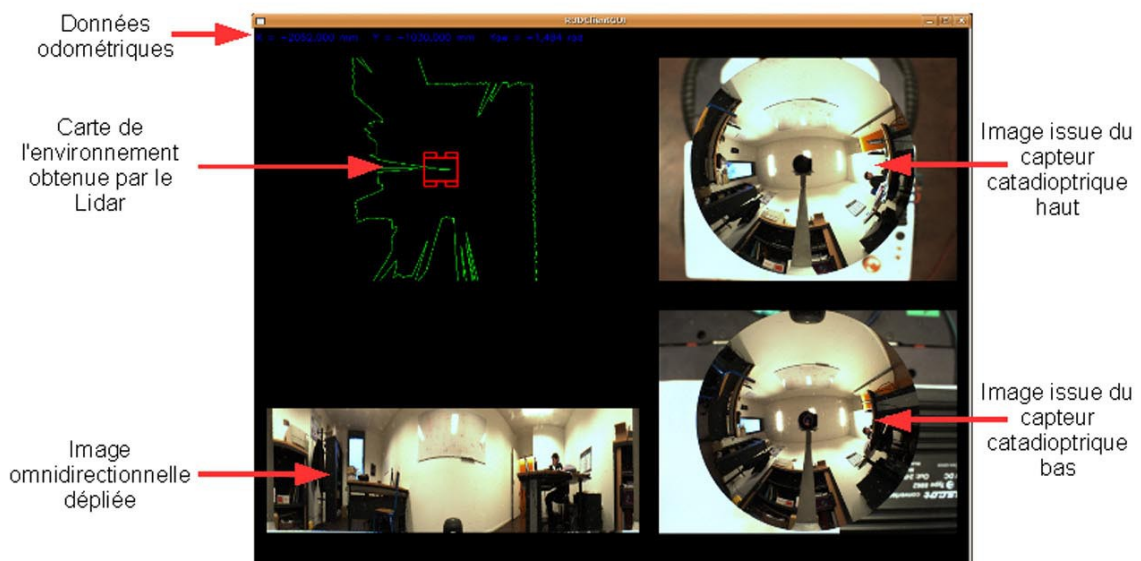


Figure 55: Interface du logiciel d'acquisition des données.

## 1.5. Conclusion

Dans ce chapitre 1, nous avons dressé un état de l'art des systèmes permettant l'acquisition de données 3D. Parmi les différentes technologies, les systèmes passifs, c'est-à-dire les systèmes de vision, sont les seuls à pouvoir répondre aux exigences de furtivité impliquées dans ce projet. Les systèmes de vision sont de plus parfaitement adaptés à la robotique mobile de par leurs nombreux avantages : grande quantité d'informations fournies, prix modéré, consommation faible, *etc.*

Les systèmes de vision « classiques » souffrent néanmoins d'un champ de vue limité qui ne leur permet pas de numériser l'intégralité d'une scène 3D instantanément. C'est pour pallier cette limitation qu'ont vu le jour les systèmes de vision omnidirectionnelle. Parmi les différentes solutions permettant d'augmenter le champ de vue d'une caméra pour le rendre omnidirectionnel, seuls les systèmes catadioptriques, c'est-à-dire l'association d'une caméra et d'un miroir, permettent de conserver une acquisition instantanée de la scène. La conception d'un tel système impose de fortes contraintes sur la position relative de la caméra par rapport au miroir pour respecter la contrainte du point de vue unique, qui permet d'obtenir des images perspectives utilisables avec la plupart des algorithmes de vision par ordinateur. Nous avons donc exposé une méthode permettant d'assembler la caméra et le miroir tout en respectant au mieux la contrainte du point de vue unique.

Parmi les différentes méthodes de reconstruction 3D passives, deux approches sont

particulièrement intéressantes pour la reconstruction de scènes complexes et pour la navigation autonome. La première, connue sous le nom de stéréovision, consiste à utiliser deux caméras pour retrouver la structure géométrique de la scène par triangulation. La seconde, le *Shape-From-Motion*, exploite le déplacement d'une seule caméra dans la scène pour obtenir les différents points de vue nécessaires à la reconstruction 3D.

Le système d'acquisition 3D a été conçu pour répondre à la problématique de la reconstruction 3D, mais aussi à celle de la navigation autonome. Notre démarche a abouti à la réalisation d'un capteur de stéréovision omnidirectionnelle utilisant deux capteurs catadioptriques dont les axes de révolution sont confondus pour simplifier la mise en correspondance des pixels.

## Chapitre 2

# Calibrage du système de stéréovision omnidirectionnelle

### Sommaire

---

<b>2.1.Introduction.....</b>	<b>71</b>
<b>2.2.Modélisation d'un système de stéréovision omnidirectionnelle.....</b>	<b>72</b>
2.2.1.Modélisation d'un système de vision omnidirectionnelle.....	73
2.2.1.1.Modèles ad hoc.....	73
2.2.1.2.Modèles polynomiaux.....	74
2.2.1.3.Modèles unifiés.....	75
2.2.2.Modélisation d'un système stéréoscopique.....	79
<b>2.3.Calibrage d'un système de stéréovision omnidirectionnelle.....</b>	<b>80</b>
2.3.1.Calibrage des systèmes de vision.....	80
2.3.2.Calibrage des systèmes stéréoscopiques.....	80
<b>2.4.Application au calibrage du système de stéréovision omnidirectionnelle.....</b>	<b>81</b>
2.4.1.Estimation des paramètres intrinsèques du système.....	82
2.4.1.1.Initialisation des paramètres.....	83
2.4.1.2.Minimisation de l'erreur de reprojection.....	89
2.4.2.Estimation de la pose relative des deux capteurs.....	90
2.4.2.1.Principe de la méthode.....	91
2.4.2.2.Méthode d'estimation de la pose relative.....	91
<b>2.5.Résultats expérimentaux.....</b>	<b>92</b>
2.5.1.Évaluation de l'estimation des paramètres intrinsèques.....	93
2.5.1.1.Évaluation sur des données de simulation.....	93
2.5.1.2.Évaluation sur des données réelles.....	94
2.5.1.3.Évaluation qualitative.....	94
2.5.2.Évaluation de l'estimation de la pose relative.....	95
2.5.2.1.Évaluation sur des données de simulation.....	95
2.5.2.2.Évaluation sur des donnée réelles.....	96
2.5.2.3.Évaluation qualitative.....	96
<b>2.6.Conclusion et perspectives.....</b>	<b>97</b>

## 2.1. Introduction

Le système d'acquisition 3D présenté dans le chapitre 1 permet au robot d'obtenir des mesures de la scène dans laquelle il se trouve. Ces mesures, obtenues par les deux caméras du système, sont de deux natures différentes : la luminance d'un pixel est liée à l'intensité lumineuse du rayon ayant frappé ce pixel et ses coordonnées fournissent des informations sur la géométrie de la scène. Les caméras, comme tout capteur chargé d'effectuer des mesures, doivent impérativement être étalonnées pour s'assurer de la justesse des informations qu'elles fournissent.

L'information de luminance peut être corrigée grâce un **calibrage radiométrique** pour s'assurer que chaque pixel a un comportement identique du point de vue photométrique [Tarel95a]. Dans le domaine de la vision par ordinateur, les travaux portant sur ce thème sont peu nombreux, d'une part car la qualité des capteurs CCD et CMOS ne cesse de s'améliorer et d'autre part car l'information la plus importante pour la reconstruction 3D est la géométrie de la scène. Le lecteur intéressé par ce thème pourra néanmoins se référer aux travaux de Tarel [Tarel95a] et à ceux de Lin [Lin04a].

Le **calibrage géométrique** est quant à lui nécessaire pour pouvoir extraire des informations métriques à partir des images 2D. Il consiste à établir la relation entre les points 3D de la scène et les points 2D (les pixels) de l'image. Le calibrage géométrique des systèmes de vision est un sujet largement abordé dans la littérature et il existe par conséquent un grand nombre de méthodes que nous classerons en deux principales catégories : les méthodes **paramétriques** et les méthodes **non-paramétriques**.

Le calibrage **paramétrique** consiste à modéliser le capteur et à estimer les paramètres du modèle choisi. Il s'agit de la méthode la plus couramment utilisée, les différences entre les différents travaux résidant principalement dans le choix du modèle (*cf* section 2.2.1) et dans le type de primitives utilisées pour calibrer (damier, droites, cercles, *etc*).

Le calibrage **non-paramétrique** consiste à associer à chaque pixel de l'image un rayon lumineux (origine et direction). Les principaux travaux dans ce domaine sont ceux de Ramalingam, Sturm et Lodha ([Ramalingam06a] [Ramalingam05b] [Sturm04a]) dont l'approche est particulièrement intéressante de par sa généralité : elle fonctionne aussi bien avec des caméras *pinhole* qu'avec des capteurs catadioptriques et même avec des systèmes multi-caméras. Les travaux de Ragot [Ragot09a] ont quant à eux été principalement consacrés à l'utilisation d'une méthode non-paramétrique pour des capteurs centraux : l'origine est la même pour tous les rayons donc seule la direction reste à estimer. La contrepartie des méthodes non-paramétriques est que l'absence de modèle de projection peut être limitante pour certains algorithmes

couramment utilisés en vision par ordinateur, notamment pour ceux nécessitant la minimisation d'un critère (Gauss-Newton, descente du gradient, Levenberg-Marquardt, *etc*). Pour cette raison, nous avons privilégié l'approche paramétrique dans nos travaux.

La première partie du chapitre 2 est consacrée à une revue des différents modèles de capteurs catadioptriques existants, puis au choix de l'un d'entre-eux. La méthode utilisée pour estimer les paramètres intrinsèques du modèle retenu est ensuite exposée. Notre système étant stéréoscopique, nous présentons dans la section suivante une méthode permettant d'estimer la pose relative des deux capteurs qui est une donnée nécessaire pour retrouver la structure géométrique de la scène. Les résultats expérimentaux de la phase de calibrage sont ensuite exposés et permettent de conclure quant à la pertinence de notre procédure de calibrage.

## 2.2. Modélisation d'un système de stéréovision omnidirectionnelle

La modélisation du système a pour but de déterminer la relation mathématique liant un point 3D de la scène à sa projection sur le plan image de chacune des caméras. Le système que nous avons développé est un système de stéréovision omnidirectionnelle composé de deux capteurs catadioptriques disposés idéalement selon une configuration axiale. Dans la pratique, l'assemblage mécanique introduit un désalignement entre les deux capteurs catadioptriques (*cf* figure 56) qui, même s'il est très faible, doit être estimé et pris en compte pour améliorer la précision des résultats.

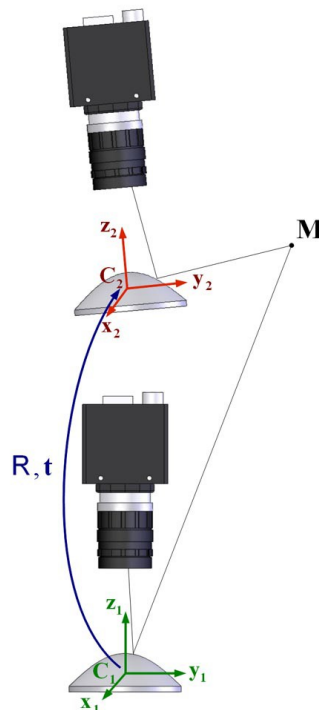


Figure 56: Configuration du système de stéréovision omnidirectionnelle.

Les projections du point  $\mathbf{M}$  dans les images peuvent être obtenues grâce à la connaissance de la pose relative  $(\mathbf{R}, \mathbf{t})$  des deux capteurs et du modèle de projection de ces capteurs. Le repère du capteur bas étant choisi comme référentiel, la pose relative est nécessaire pour exprimer les coordonnées de  $\mathbf{M}$  dans les repères locaux des capteurs catadioptriques. Le modèle de projection de chacun des capteurs catadioptriques permet alors de calculer les projections de  $\mathbf{M}$  dans les images. La modélisation du système de stéréovision omnidirectionnelle requiert donc de modéliser chacun des capteurs catadioptriques puis d'estimer leur pose relative.

### **2.2.1. Modélisation d'un système de vision omnidirectionnelle**

L'approche paramétrique du calibrage a été retenue pour s'affranchir des limitations d'un modèle de type « boîte noire » (non-paramétrique) citées dans l'introduction de ce chapitre. Par définition, une telle approche impose le choix un modèle pour décrire la fonction de projection réalisée par le système. Ce choix est particulièrement important puisqu'il conditionnera la précision des résultats et la complexité des algorithmes qui l'utiliseront. Le modèle idéal doit être capable d'offrir une grande précision tout en étant peu coûteux en temps de calcul, robuste aux imperfections du système (distorsions induites par les lentilles, désalignement possible entre le miroir et la caméra, *etc*), et si possible assez générique pour pouvoir utiliser d'autres types de capteurs sans modifier fondamentalement les algorithmes.

Il existe dans la littérature un grand nombre de modèles plus ou moins complexes pour exprimer la fonction de projection des capteurs catadioptriques, mais nous limiterons notre étude aux **modèles centraux**, l'importance de cette centralité ayant été soulignée dans la section 1.3.2.3. Nous renvoyons aux travaux de Ieng et Benosman [Ieng05a] pour la modélisation des capteurs non-centraux via l'estimation de la surface caustique qui correspond aux points de convergence des rayons lumineux.

#### **2.2.1.1. Modèles ad hoc**

La première méthode pour modéliser un capteur catadioptrique consiste à décrire le processus physique de formation des images en utilisant les lois de la réflexion sur la surface du miroir puis en utilisant le modèle de la caméra. Le modèle mathématique issu d'une telle démarche n'est alors valable que pour le type de miroir pour lequel il a été conçu, c'est pourquoi ces modèles sont qualifiés de **modèles ad hoc**.

Pour notre capteur composé d'un miroir hyperboloïde et d'une caméra perspective, le modèle dépendra des paramètres de l'hyperbole et de la matrice des paramètres intrinsèques  $\mathbf{K}$  de la caméra. En reprenant les notations introduites par Svoboda (*cf* figure 57 et [Svoboda99a]),

la projection  $\mathbf{q}$  d'un point  $\mathbf{X}$  est donnée par l'équation :

$$\mathbf{q} = \mathbf{K} \frac{1}{z_C} \mathbf{R}_C (\lambda \mathbf{R}_M (\mathbf{X} - \mathbf{t}_M) - \mathbf{t}_C) \quad (11)$$

où  $\mathbf{R}_M$  est la matrice de rotation et  $\mathbf{t}_M$  le vecteur de translation permettant de passer du repère monde au repère miroir,  $\mathbf{R}_C$  est la rotation et  $\mathbf{t}_C$  la translation de la caméra par rapport au miroir. Les paramètres du miroir apparaissent dans le facteur  $\lambda$  et  $z_C$  est un facteur de normalisation permettant de travailler en coordonnées homogènes. La démonstration de cette équation ainsi que les expressions complètes des facteurs  $\lambda$  et  $z_C$  sont décrites dans [Svoboda99a].

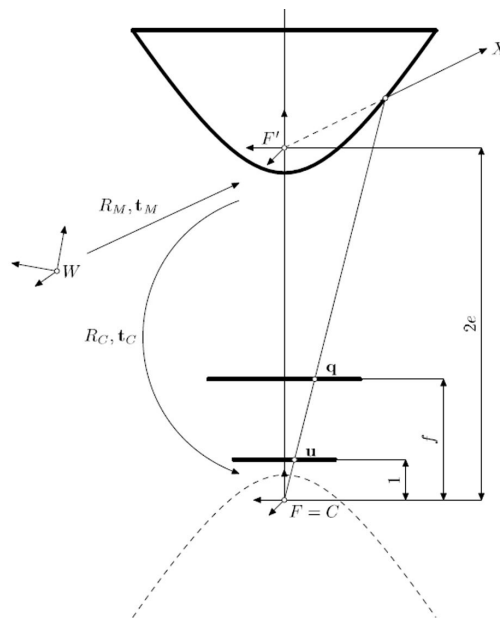


Figure 57: Modèle de projection d'un capteur à base de miroir hyperboloïde.  
Source [Svoboda99a].

Le grand nombre de paramètres à estimer augmente le risque de convergence vers un minimum local lors du calibrage. En pratique, ce modèle est très sensible aux imperfections mécaniques du système, puisque si la contrainte du point de vue unique n'est pas parfaitement respectée, le processus de formation des images ne correspond plus aux équations utilisées.

Les modèles *ad hoc* d'autres configurations (miroir parabolöide, association de plusieurs miroirs, etc), ainsi que des modèles plus complets introduisant des coefficients de distorsion et des erreurs d'alignement sont exposés dans les travaux de Barbosa [Barbosa04a].

### 2.2.1.2. Modèles polynomiaux

La multitude des configurations matérielles a poussé la communauté scientifique à s'intéresser à des modèles plus généraux. Ainsi, Scaramuzza propose d'approximer la fonction de



projection par un polynôme [Scaramuzza06b]. Le modèle sera alors indépendant du système mis en œuvre à condition toutefois qu'il soit central.

Le système est également supposé symétrique par rapport à l'axe vertical pour que le seul paramètre à estimer soit l'angle d'élévation du rayon par rapport à l'horizontale (cf figure 58). Ainsi, connaissant les coordonnées métriques du pixel  $\mathbf{p}=[u \ v]^T$ , il est possible de déterminer le vecteur directeur  $\mathbf{P}$  du rayon lumineux :  $\mathbf{P}=[u \ v \ f(u,v)]^T$ , où  $f$  est la fonction approximée par le polynôme de degré 4 :

$$f(\rho)=a_0+a_1\rho+a_2\rho^2+a_3\rho^3+a_4\rho^4 \quad (12)$$

avec  $\rho=\sqrt{(u^2+v^2)}$ .

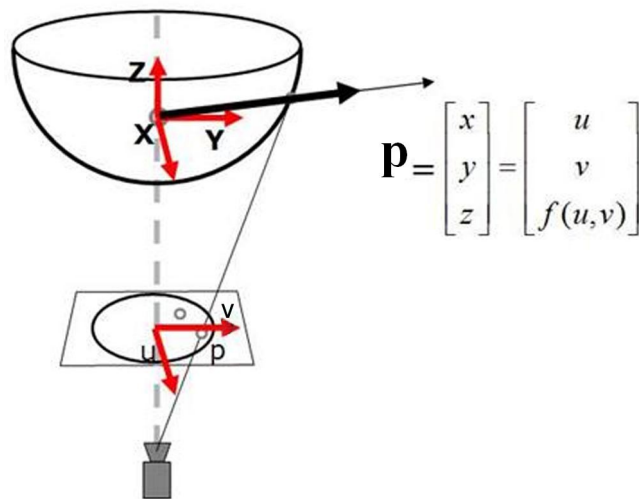


Figure 58: Principe du modèle polynomial.  
Source [Scaramuzza]

Le calibrage consiste alors à estimer les coefficients du polynôme défini par l'équation (12). Avec ce modèle, il est très difficile d'obtenir des valeurs initiales des coefficients du polynôme, ce qui complique l'automatisation du processus de calibrage car il est alors impossible d'estimer la pose de la mire pour en déduire la projection de ses points dans l'image.

### 2.2.1.3. Modèles unifiés

Dans le même esprit de généralisation, Geyer [Geyer01a] et Barreto [Barreto06a] ont proposé une théorie pour l'unification des modèles de capteurs centraux. Cette théorie repose sur l'équivalence entre la projection sur une surface quadrique et la projection sur une sphère. Ainsi, Geyer a démontré qu'il est possible de modéliser les capteurs centraux en utilisant une projection sur une sphère suivie d'une projection sur un plan à partir d'un point dépendant de la forme et des paramètres du miroir [Geyer00a]. Ce théorème de **l'équivalence projective**, a donné son nom à la sphère unitaire correspondante : la **sphère d'équivalence**.

Le modèle initialement proposé par Geyer et Barreto a ensuite été repris et légèrement amélioré. Ainsi, le modèle de Mei [Mei07a] (*cf* figure 59) est particulièrement intéressant puisqu'il ajoute au modèle initial des coefficients de distorsion pour corriger les imperfections des objectifs et de l'assemblage caméra/miroir. La philosophie du modèle est légèrement différente : Mei considère le système catadioptrique comme un unique capteur, et non plus comme l'association de deux éléments (le miroir et la caméra) en introduisant la notion de focale généralisée qui dépend des paramètres de ces deux entités. Celle-ci permet de réduire le nombre de paramètres à estimer et limite par conséquent le risque de convergence vers des minima locaux lors du calibrage. Ces raisons nous ont conduits à utiliser ce modèle dans nos travaux. Après une présentation des conventions utilisées, le modèle de la sphère d'équivalence adopté dans nos travaux est décrit en détail.

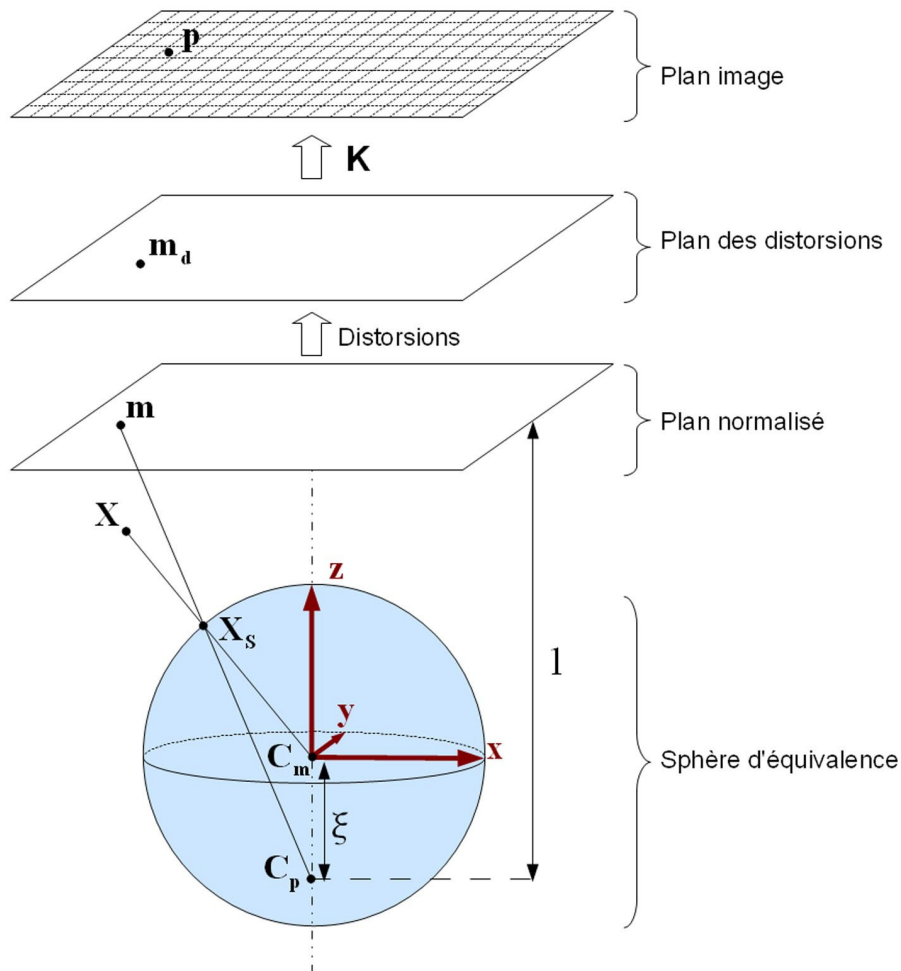
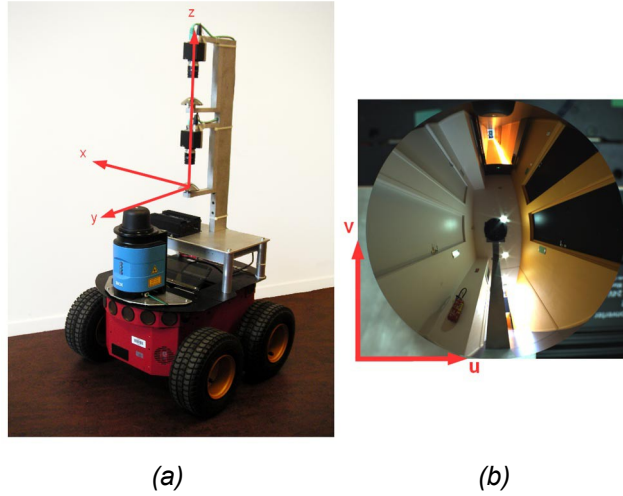


Figure 59: Modèle de projection unifié utilisé.

Nous avons choisi de travailler avec la convention des axes représentée sur la figure 60. Le vecteur  $z$  pointe vers le haut et le vecteur  $y$  vers l'avant du robot. Les coordonnées des pixels

sont données par rapport au coin inférieur gauche de l'image et non pas par rapport au coin supérieur qui est généralement choisi comme origine.



(a) (b)  
Figure 60: Convention des axes.  
(a) Pour le capteur - (b) Pour l'image

La projection  $\mathbf{p}=[u \ v]^T$  d'un point 3D  $\mathbf{X}=[X_m \ Y_m \ Z_m]^T$  dont les coordonnées sont exprimées dans un repère monde arbitraire est illustrée par la figure 59 et peut être obtenue en appliquant les étapes suivantes :

- Les coordonnées du point  $\mathbf{X}$  sont exprimées dans le repère associé au capteur grâce à une transformation rigide notée  $W$ . Cette transformation dépend de sept paramètres contenus dans le vecteur  $\mathbf{V}_1=[q_w \ q_x \ q_y \ q_z \ t_x \ t_y \ t_z]^T$  : les quatre premiers éléments sont ceux de la rotation  $\mathbf{R}$  caractérisée par un quaternion (cf Annexe A) et les trois suivants sont ceux de la translation  $\mathbf{T}$ . Les coordonnées de  $\mathbf{X}$  dans le repère du miroir sont alors données par :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} + \mathbf{T} \quad (13)$$

$$\text{où } \mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 + q_y^2 - q_x^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 + q_z^2 - q_x^2 - q_y^2 \end{bmatrix}.$$

- Le point  $\mathbf{X}$  de coordonnées  $[X \ Y \ Z]^T$  dans le repère du miroir est projeté sur la sphère unitaire en  $\mathbf{X}_S=[X_S \ Y_S \ Z_S]^T$ . Ce point subit une seconde projection sur le plan normalisé à partir d'un point situé à une distance  $\xi$  du centre de la sphère. Ces deux transformations (projection sur la sphère puis projection sur le plan normalisé) sont regroupées dans la fonction  $H$  qui ne dépend que d'un paramètre :  $\mathbf{V}_2=[\xi]$ . La

projection de  $\mathbf{X}_S$  sur le plan normalisé, notée  $\mathbf{m}=[x \ y]^T$  est donc obtenue par :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{X_S}{Z_S + \xi} \\ \frac{Y_S}{Z_S + \xi} \end{bmatrix}, \text{ avec } \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix} = \begin{bmatrix} \frac{X}{\sqrt{X^2 + Y^2 + Z^2}} \\ \frac{Y}{\sqrt{X^2 + Y^2 + Z^2}} \\ \frac{Z}{\sqrt{X^2 + Y^2 + Z^2}} \end{bmatrix} \quad (14)$$

- Les distorsions sont ensuite ajoutées au point  $\mathbf{m}$  en appliquant la fonction de distorsion  $D$  qui fait intervenir cinq coefficients : trois coefficients de distorsions radiales ( $k_1$ ,  $k_2$  et  $k_5$ ) et deux coefficients de distorsions tangentielles ( $k_3$  et  $k_4$ ). Soient  $\mathbf{V}_3=[k_1 \ k_2 \ k_3 \ k_4 \ k_5]^T$  le vecteur de paramètres contenant ces coefficients,  $\rho = \sqrt{x^2 + y^2}$  et  $\mathbf{m}_d=[x_d \ y_d]^T$  le point résultant, ses coordonnées sont obtenues par l'équation :

$$\begin{aligned} \mathbf{m}_d &= \mathbf{m} + D(\mathbf{m}) \\ \mathbf{m}_d &= \begin{bmatrix} x(1 + k_1\rho^2 + k_2\rho^4 + k_5\rho^6) + 2k_3xy + k_4(\rho^2 + 2x^2) \\ y(1 + k_1\rho^2 + k_2\rho^4 + k_5\rho^6) + 2k_4xy + k_3(\rho^2 + 2y^2) \end{bmatrix} \end{aligned} \quad (15)$$

- La projection finale est obtenue grâce à la matrice de projection généralisée  $\mathbf{K}$ . Cette dernière contient 5 paramètres : la focale généralisée  $\gamma_u$  et  $\gamma_v$ , les coordonnées du point principal  $u_0$  et  $v_0$  et le paramètre d'obliquité  $\alpha$ . Soient  $\mathbf{K}$  cette fonction de projection et  $\mathbf{V}_4=[\alpha \ \gamma_u \ \gamma_v \ u_0 \ v_0]^T$  le vecteur de paramètres dont elle dépend, la projection  $\mathbf{p}=[u \ v \ 1]^T$  du point  $\mathbf{X}$  sur le plan image, est alors donnée par :

$$\begin{aligned} \mathbf{p} &= \mathbf{K} \begin{bmatrix} \mathbf{m}_d \\ 1 \end{bmatrix} \\ \mathbf{p} &= \begin{bmatrix} \gamma_u & \gamma_u \alpha & u_0 \\ 0 & \gamma_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{m}_d \\ 1 \end{bmatrix} \end{aligned} \quad (16)$$

Soit  $\mathbf{V}$  le vecteur contenant les 18 paramètres du modèle, c'est-à-dire  $\mathbf{V}=[\mathbf{V}_1^T \ \mathbf{V}_2^T \ \mathbf{V}_3^T \ \mathbf{V}_4^T]^T$ . La fonction de projection globale d'un point  $\mathbf{X}$ , notée  $P(\mathbf{V}, \mathbf{X})$ , est obtenue par composition des différentes fonctions présentées précédemment comme indiqué par l'équation (17).

$$P(\mathbf{V}, \mathbf{X}) = \mathbf{K} \circ D \circ H \circ W(\mathbf{V}, \mathbf{X}) \quad (17)$$

Ces étapes permettent d'obtenir la projection sur le plan image d'un point 3D connaissant ses coordonnées dans l'espace. Dans le cadre de la reconstruction 3D, il est nécessaire de pouvoir effectuer l'opération inverse, c'est-à-dire de pouvoir calculer la direction du rayon lumineux

associé à un pixel. Cette étape revient à calculer les coordonnées du point  $\mathbf{X}_s$  appartenant à la sphère d'équivalence étant donné les coordonnées  $\mathbf{m}=[x \ y]^T$  d'un point 2D du plan normalisé. Cette opération de **rétro-projection**, aussi appelée *lifting*, peut être accomplie grâce à la formule (18) qui correspond à la transformation inverse de la fonction de projection  $H$ .

$$\mathbf{X}_s = H^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} x \\ \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} y \\ \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} - \xi \end{bmatrix} \quad (18)$$

La formule (18) permet d'obtenir un vecteur directeur normalisé du rayon lumineux associé au point  $\mathbf{m}$ . Dans le cas où la normalisation du vecteur importe peu, une équivalence projective est donnée par :

$$H^{-1}(\mathbf{m}) \sim \begin{bmatrix} x \\ y \\ 1 - \xi \frac{x^2 + y^2 + 1}{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}} \end{bmatrix} \quad (19)$$

### 2.2.2. Modélisation d'un système stéréoscopique

La configuration géométrique du système stéréoscopique est modélisée par une matrice de rotation  $\mathbf{R}$  et un vecteur de translation  $\mathbf{t}$  représentant la position et l'orientation du capteur du haut par rapport au capteur du bas, comme l'illustre la figure 61.

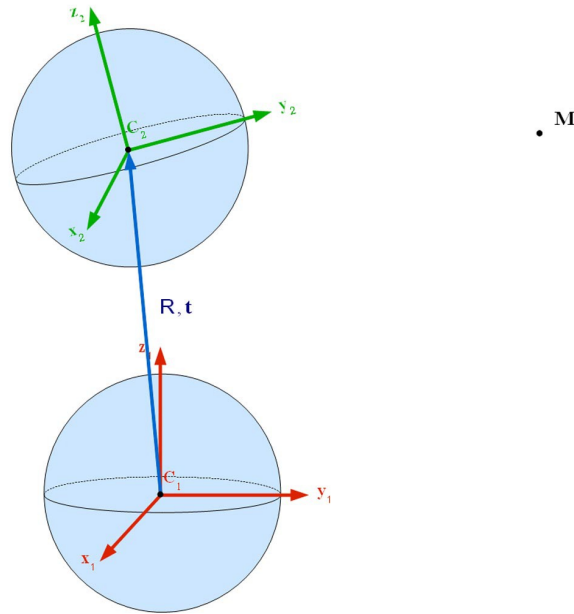


Figure 61: Modélisation d'un système stéréoscopique.

## 2.3. Calibrage d'un système de stéréovision omnidirectionnelle

### 2.3.1. Calibrage des systèmes de vision

Le calibrage d'un système de vision consiste à estimer les paramètres du modèle de projection qui sera utilisé par les algorithmes de reconstruction 3D. Le calibrage est un sujet qui a été très abordé pour les caméras perspectives, mais beaucoup moins pour les systèmes de vision omnidirectionnelle. Les méthodes existantes pour calibrer un système de vision peuvent être classées en deux catégories principales : le calibrage fort et l'auto-calibrage.

Les méthodes d'**auto-calibrage** exploitent le fait que les paramètres intrinsèques du système restent constants entre les différentes prises de vues. Le déplacement de la caméra dans une scène statique suffit à imposer les contraintes nécessaires à l'estimation des paramètres intrinsèques, sans connaissance *a priori* sur la scène. Les résultats obtenus par ces méthodes sont assez instables à cause du grand nombre de paramètres à estimer [Devernay97a] [Zhang99a].

Le **calibrage fort** consiste à utiliser un objet de calibrage, appelé mire, contenant des points de contrôle dont les positions dans l'espace sont connues avec une grande précision [Zhang98a]. Ces points de contrôle doivent être facilement identifiables dans les images ; les primitives telles que les coins, les points, les cercles, les intersections de droites sont donc de bons candidats. Étant données les coordonnées 3D des points de la mire et leurs projections dans l'image, le calibrage consiste à trouver les paramètres de la fonction de projection qui font correspondre au mieux les projections estimées par cette fonction et les projections réelles, c'est-

à-dire celles extraites de l'image.

Parmi les méthodes de calibrage fort, celles reposant sur l'utilisation d'une mire plane sont les plus souvent retenues ([Zhang00a], [Bouguet], [Scaramuzza06a], [Mei07a]) car elles sont simples à mettre en œuvre tout en offrant un degré de précision élevé. Leur popularité est due à la possibilité d'utiliser une simple feuille de papier sur laquelle est imprimée la mire plutôt que d'avoir à se procurer une mire 3D, c'est-à-dire une mire composée de plusieurs plans orthogonaux, qui est très coûteuse. Ces méthodes sont de plus particulièrement intéressantes car le mouvement de la mire peut être quelconque et ne nécessite pas d'être connu. Ces méthodes, initialement développées pour les caméras perspectives ([Zhang00a], [Bouguet]), ont récemment été adaptées aux capteurs catadioptriques centraux, notamment aux travers des travaux de Mei [Mei07a] et de Scaramuzza [Scaramuzza06a].

### 2.3.2. Calibrage des systèmes stéréoscopiques

Pour calibrer un système stéréoscopique, c'est-à-dire déterminer la relation inter-caméras, il existe principalement deux approches.

La première approche consiste à réaliser un **calibrage fort** (*cf* section 2.3.1) de chaque caméra indépendamment, puis à estimer la relation inter-camera grâce à un objet de dimensions connues, par exemple une mire de calibrage représentant un damier [Zhang96a]. L'objet doit alors être visible simultanément par les deux caméras, ce qui peut être difficile à mettre en œuvre car si la mire est placée trop loin du capteur, le calibrage sera moins précis, mais si elle est placée trop près, elle ne sera pas visible par les deux caméras.

La seconde approche, connue sous le nom de **calibrage faible** [Ponce96a] [Zhang98b], consiste à calibrer le système stéréoscopique sans connaissance *a priori* de la scène observée (celle-ci doit néanmoins être statique). Les deux images d'un système stéréoscopique permettent en effet de retrouver sa géométrie épipolaire décrite par la matrice fondamentale ou par la matrice essentielle. La matrice fondamentale facilite la mise en correspondance des points et permet d'obtenir une reconstruction 3D projective de la scène [Hartley92a]. La matrice essentielle est un cas particulier de matrice fondamentale pour lequel les caméras sont calibrées. Elle a été initialement introduite par Longuet-Higgins [Longuet-Higgins81a] avant la matrice fondamentale. La matrice essentielle est particulièrement intéressante puisque sa décomposition fournit la pose relative des deux capteurs, mais la translation ne peut être estimée qu'à un facteur d'échelle près [Bunshoten03a]. Le calibrage faible est donc intéressant pour simplifier la mise en correspondance des points mais il ne permet d'obtenir la structure géométrique de la scène qu'à un facteur d'échelle près.

Bien que dans certaines applications, notamment pour la reconstruction 3D à des fins de visualisation, le facteur d'échelle importe peu, sa connaissance est nécessaire pour la préparation d'interventions et la navigation. Pour réaliser ces tâches, la taille des objets et leur distance par rapport au robot doit pouvoir être déterminée, et la pose relative des deux capteurs doit être estimée en portant une attention particulière à l'obtention du facteur d'échelle.

## 2.4. Application au calibrage du système de stéréovision omnidirectionnelle

Nous avons développé des outils permettant d'une part l'estimation des paramètres intrinsèques des deux capteurs catadioptriques, et d'autre part l'estimation de leur pose relative. Les logiciels développés<sup>9</sup> (cf figure 62) ne nécessitent pas l'achat de licence puisqu'ils ont été entièrement développés en C++ et ils n'utilisent que des bibliothèques libres de droits.

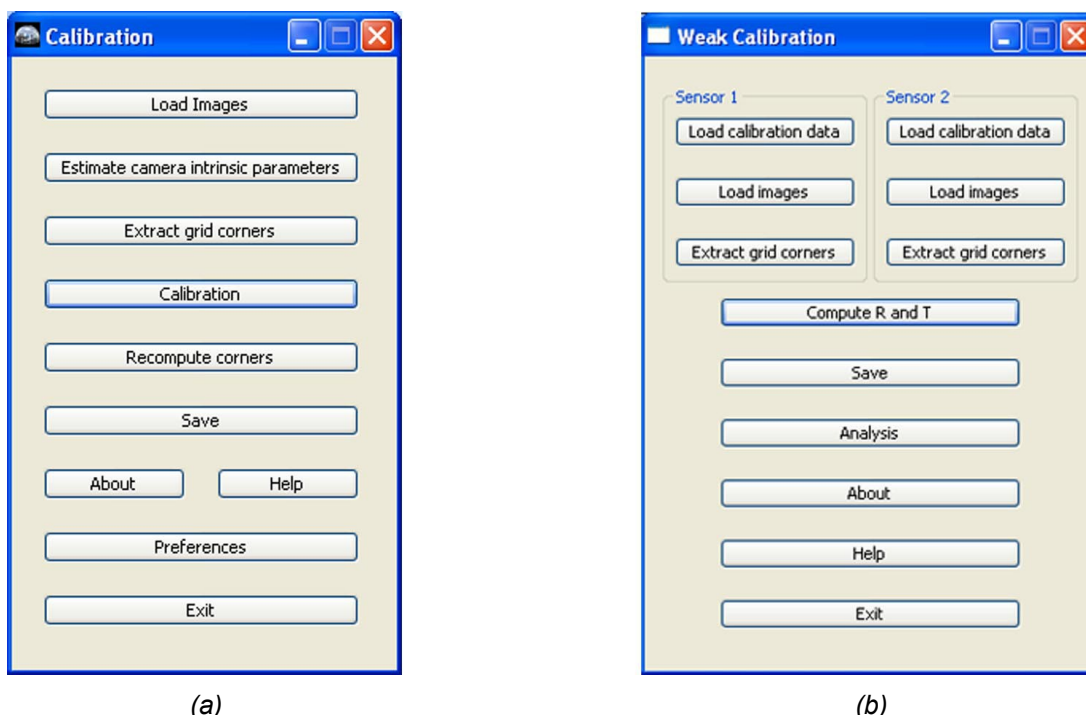


Figure 62: Interfaces des logiciels de calibrage développés.  
(a) Estimation des paramètres intrinsèques – (b) Estimation de la pose relative.

### 2.4.1. Estimation des paramètres intrinsèques du système

Nous avons développé un outil de calibrage des capteurs centraux qui permet d'estimer les paramètres du modèle de la sphère d'équivalence présenté dans la section 2.2.1.3. Il s'appuie sur la méthode proposée par Mei [Mei07a] avec des améliorations dans la détection de l'empreinte du miroir et dans l'optimisation du temps de calcul.

<sup>9</sup> Téléchargeables à l'adresse : <http://omni3d.esigelec.fr/doku.php/thesis/r3d/start>



Cette méthode de calibrage consiste à minimiser l'erreur de reprojection, c'est-à-dire la distance euclidienne entre les projections estimées des points de la mire et leurs projections mesurées dans l'image. Ainsi, si  $n$  est le nombre de points 3D  $\mathbf{X}_i$ ,  $\mathbf{x}_i$  leurs projections dans les images, et  $P$  la fonction de projection, nous cherchons le vecteur de paramètres  $\mathbf{V}$  qui minimise la fonction coût  $E(\mathbf{V})$  :

$$E(\mathbf{V}) = \frac{1}{2} \sum_{i=1}^n [P(\mathbf{V}, \mathbf{X}_i) - \mathbf{x}_i]^2 \quad (20)$$

La fonction  $E(\mathbf{V})$  est une fonction non-linéaire qu'il faut par conséquent minimiser grâce à un algorithme itératif. Nous avons implanté l'algorithme de Levenberg-Marquardt (*cf* Annexe B), qui est souvent utilisé en vision par ordinateur car il permet d'obtenir une bonne estimation du vecteur de paramètres en un temps de calcul assez faible.

#### **2.4.1.1. Initialisation des paramètres**

La méthode de calibrage est fondée sur une minimisation non-linéaire qui, pour être correcte, nécessite une initialisation des paramètres. Certains paramètres du modèle peuvent être initialisés sans calcul, notamment :

- le paramètre dépendant de la forme du miroir :  $\xi = 1$  car ce paramètre a une faible influence sur la convergence [Mei07a],
- les coefficients de distorsions :  $k_1 = k_2 = k_3 = k_4 = k_5 = 0$ , ce qui revient à considérer qu'il n'y a pas de distorsion,
- le paramètre d'obliquité :  $\alpha = 0$ , cela suppose que les lignes et les colonnes des pixels sont perpendiculaires.

Les autres paramètres du modèle sont à déterminer, mais certaines approximations peuvent d'ores et déjà être faites :

- $\gamma_u = \gamma_v = \gamma$ , ce qui revient à supposer que les pixels sont carrés,
- les coordonnées du point principal sont initialisées avec les coordonnées du centre du miroir.

#### **Initialisation du point principal**

Le point principal est initialisé avec les coordonnées du centre du miroir. La solution la plus précise consiste à effectuer une extraction automatique des contours du miroir pour en déterminer le centre car la sélection manuelle du centre d'un cercle n'est pas aisée.

Mei [Mei07a] propose une procédure de type RANSAC (*cf* Annexe C) ; il ne conserve que les points de contours autour d'une estimation initiale du bord du miroir, puis il tire aléatoirement

des points pour créer une liste de centres et de rayons possibles. La valeur médiane est alors prise pour initialiser les coordonnées du point principal. Cependant, cette procédure d'extraction automatique n'a pas fourni de résultats satisfaisants sur nos images, probablement à cause de la grande quantité d'informations autour du miroir. La zone située en dehors du miroir comporte en effet de nombreux contours puisque la caméra n'observe pas un fond uni mais une partie du robot et du sol.

Nous avons donc développé une méthode alternative, qui s'appuie sur le principe de la transformée de Hough [Yuen90a], pour détecter de façon plus robuste le contour du miroir. L'idée principale consiste à déterminer les droites perpendiculaires aux contours extraits puisque leur point d'intersection correspond théoriquement au centre du cercle, comme l'illustre la figure 63.

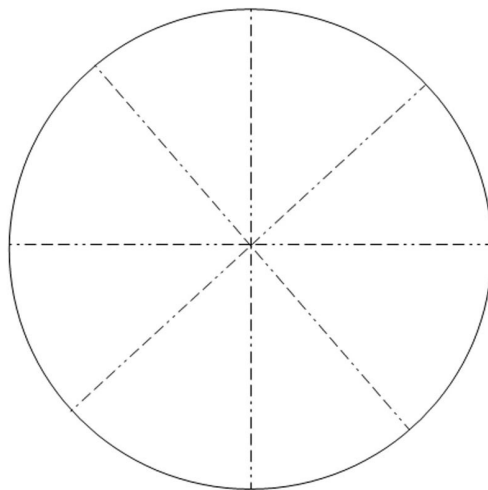


Figure 63: Principe de la méthode de détection de cercle.

La détection automatique du miroir est réalisée grâce aux six étapes présentées ci-dessous. L'utilisateur doit seulement fournir une première estimation de l'emplacement du miroir en sélectionnant manuellement le centre et le bord de celui-ci.

#### **Étape 1 : Filtrage**

En premier lieu, l'image est filtrée par un filtre gaussien 3x3 afin d'éliminer les pixels isolés (bruit) lors de la détection des contours.

#### **Étape 2 : Détection des contours**

La seconde étape consiste à détecter les contours qui serviront pour la détermination du centre du cercle. Pour cela, l'algorithme de Canny est appliqué à l'image précédemment filtrée (cf figure 64 a). Seuls les contours les plus proches de l'estimation fournie par l'utilisateur sont conservés pour la suite des traitements. Les contours situés au delà de  $\pm 10\%$  du rayon fourni par l'utilisateur sont donc rejetés (cf figure 64 b).



Figure 64: Contours détectés par l'algorithme de Canny.  
 (a) Totalité des contours – (b) Contours conservés.

### Étape 3 : Calcul de l'orientation des contours

Pour calculer l'équation des droites perpendiculaires aux contours, les gradients d'intensité ( $\mathbf{G}_x$  et  $\mathbf{G}_y$ ) sont calculés en chaque point en convoluant les noyaux suivants à l'image en niveaux de gris  $\mathbf{I}$  :

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{I} \quad \text{et} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{I} \quad (21)$$

La droite  $\mathbf{d}$ , perpendiculaire au point de contour  $\mathbf{x} = [i \ j]^T$ , a alors pour équation paramétrique :

$$\mathbf{d} = \begin{bmatrix} i \\ j \end{bmatrix} + \lambda \cdot \begin{bmatrix} \mathbf{G}_x(i, j) \\ \mathbf{G}_y(i, j) \end{bmatrix} \quad (22)$$

### Étape 4 : Accumulation

Le centre du cercle correspond théoriquement à l'intersection de toutes les droites perpendiculaires aux contours mais en pratique, ces droites ne s'intersectent pas en un point unique. Une recherche du point appartenant à un maximum de droites est effectuée par le biais d'un accumulateur.

L'accumulateur est une matrice de dimension identique à l'image et ne comportant initialement que des valeurs nulles. Pour chaque point de la droite, et pour chaque droite, la valeur du pixel correspondant dans l'accumulateur est incrémentée. De cette façon, les pixels de l'accumulateur ayant les plus fortes valeurs (représentées en blanc sur la figure 65) sont les centres les plus probables.

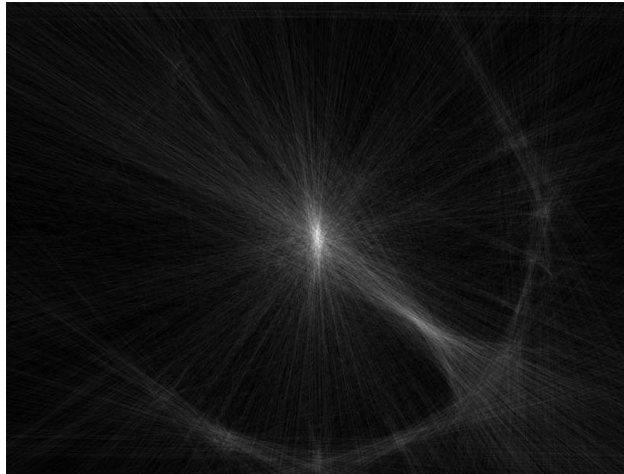


Figure 65: Représentation de l'accumulateur.

### Étape 5 : Calcul du centre

La détermination du centre du cercle par la recherche du maximum de l'accumulateur ne fournit pas le résultat escompté ; le cercle estimé par ce principe est en effet légèrement décalé par rapport à la réalité. Notre méthode consiste donc à calculer un centre de gravité plutôt qu'un maximum après avoir rejeté les valeurs de l'accumulateur les plus faibles (*cf* figure 66), c'est-à-dire celles qui sont inférieures à la moitié de la valeur maximale de celui-ci. Les coordonnées de ce centre de gravité sont alors utilisées pour initialiser le point principal.

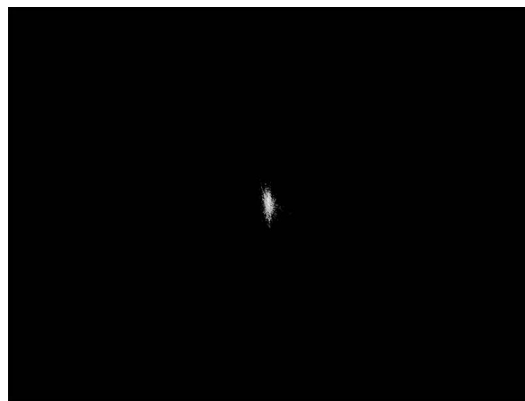


Figure 66: Représentation de l'accumulateur après le seuillage.

### Étape 6 : Détermination du rayon

Bien que le rayon du miroir ne soit pas un paramètre nécessaire au modèle de projection, cette donnée est intéressante d'une part pour tracer une estimation des bords du miroir et ainsi valider l'estimation du point central du capteur (*cf* figure 67), et d'autre part pour définir une région d'intérêt sur les pixels appartenant au miroir uniquement.

Le rayon optimal est recherché dans un intervalle de  $\pm 10$  pixels autour de l'estimation fournie par l'utilisateur lors de la sélection du centre et du bord du miroir. Pour chaque valeur de cet intervalle, le nombre de points de contour appartenant au cercle correspondant est calculé. La

valeur du rayon pour laquelle ce nombre de points est maximal fournit une bonne estimation du rayon du miroir, comme l'illustre la figure 67.

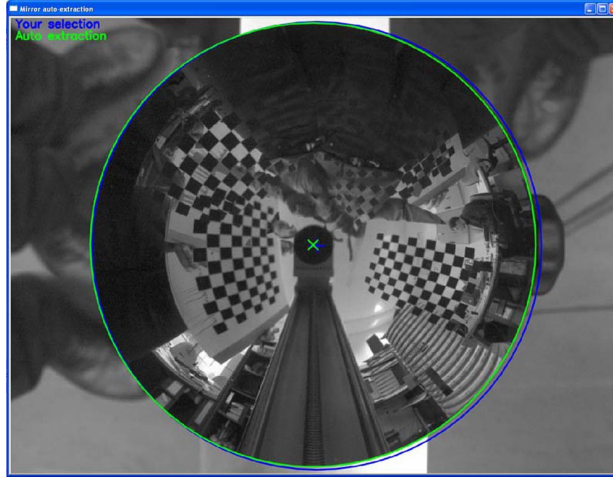


Figure 67: Résultat de l'extraction automatique du miroir (en vert).  
Le cercle bleu correspond à l'estimation du miroir fournie par l'utilisateur.

### Initialisation de la focale généralisée

La sélection de points appartenant à une même droite de l'espace 3D (voir l'exemple de la figure 68) permet de déterminer la valeur de la focale généralisée.

Si les points sélectionnés appartiennent à la droite définie par le plan de normale  $\mathbf{n}=[n_x \ n_y \ n_z]^T$  (cf figure 100), ils vérifient l'équation :

$$[H^{-1}(\mathbf{m})]^T \mathbf{n} = 0 \quad (23)$$

Avec la valeur initiale de  $\xi$  ( $\xi=1$ ), la formule de rétro-projection (19) devient :

$$H^{-1}(\mathbf{m}) = \begin{bmatrix} x \\ y \\ \frac{1}{2} - \frac{1}{2}(x^2 + y^2) \end{bmatrix} \quad (24)$$

Soient  $\mathbf{p}=[u \ v]^T$  un point du plan image, et  $\mathbf{p}_c=[u_c \ v_c]^T=[u-u_0 \ v-v_0]^T$  ce même point exprimé en coordonnées centrées, l'équation (24) devient :

$$H^{-1}(\mathbf{m}) = \begin{bmatrix} u_c \\ v_c \\ \frac{\gamma}{2} - \frac{1}{2\gamma}(u_c^2 + v_c^2) \end{bmatrix} \quad (25)$$

Chaque point de la droite permet alors de former l'équation :

$$\left[ H^{-1}(\mathbf{m}) \right]^T \mathbf{n} = 0 \quad \Leftrightarrow \quad n_x u_c + n_y v_c + \frac{\gamma n_z}{2} - \frac{n_z}{\gamma} \left( \frac{u_c^2 + v_c^2}{2} \right) = 0 \quad (26)$$

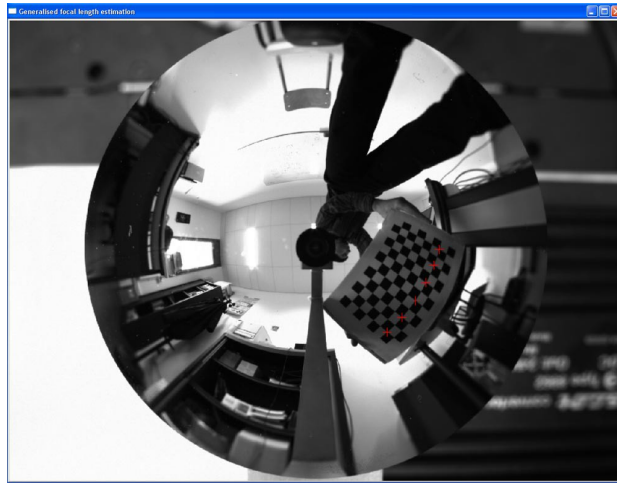


Figure 68: Exemple de points utilisés pour estimer la focale généralisée (en rouge).

La sélection d'au moins trois points appartenant à une même droite permet de résoudre l'équation (26) et d'obtenir la valeur initiale de la focale généralisée  $\gamma$ .

### Initialisation des paramètres extrinsèques des mires

L'initialisation des paramètres extrinsèques des mires a un double intérêt : elle permet d'une part d'assurer la convergence lors de la minimisation finale de l'erreur de reprojection (voir section 2.4.1.2), et d'autre part d'automatiser l'extraction des points de contrôle.

L'extraction automatique des points de contrôle est une opération très délicate dans le cas d'images omnidirectionnelles. En effet, les algorithmes développés pour les caméras perspectives ne sont pas applicables, puisqu'ils se basent en général sur le fait que les points d'intérêt de la mire sont situés sur des droites. Dans le cas omnidirectionnel, les points de contrôle appartiennent à des courbes et non plus à des droites.

Il est cependant possible de faciliter la tâche de l'utilisateur pour qu'il n'ait pas à cliquer sur chaque point de la mire. L'estimation des poses des mires à partir de ses quatre coins permet en effet de prédire la projection de la totalité de ses points de contrôle dans l'image. Un détecteur de coins, tel que le détecteur de Harris [Harris88a], permet ensuite d'extraire la projection réelle de ces points dans l'image.

L'estimation de la pose d'une mire connaissant la projection de  $N$  points de contrôle sur le plan image est un problème bien connu en vision par ordinateur sous le nom de P-N-P (*Perspective-N-Points*) [Fabrizio08a]. La sélection des coins de la mire par l'utilisateur permet de connaître la projection de quatre points dans le plan image qui seront utilisés pour chercher les

paramètres extrinsèques de la mire.

Tous les paramètres intrinsèques du modèle de projection ont été initialisés grâce aux méthodes présentées dans la section 2.4.1.1, et les coordonnées des coins de la mire sont connues dans un repère associé à la mire. Il ne reste donc que les sept paramètres extrinsèques de la mire à estimer pour pouvoir réaliser la projection de la totalité des points de contrôle. Le vecteur  $\mathbf{V}_1$  des paramètres extrinsèques est initialisé avec les valeurs correspondant à une rotation et à une translation nulles :  $\mathbf{V}_1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ . L'algorithme de Levenberg-Marquardt (cf Annexe B) est ensuite utilisé pour minimiser l'erreur entre la projection estimée des quatre coins de la mire et la projection mesurée dans l'image, c'est-à-dire sélectionnée par l'utilisateur. La jacobienne  $\mathbf{J}$  de la fonction de projection  $P$  intervenant dans cet algorithme est alors de dimension  $2N \times 7$  puisque les paramètres intrinsèques sont supposés connus :

$$\mathbf{J} = \begin{bmatrix} \frac{\partial P}{\partial \mathbf{V}_1} \end{bmatrix}_{2N \times 7} \quad (27)$$

avec  $N=4$  puisque quatre points sont connus.

Une fois la pose de la mire déterminée, la totalité des points de contrôle est projetée sur le plan image ; les projections réelles sont alors extraites par un détecteur de coins. La figure 69 présente le résultat de ce processus sur une image omnidirectionnelle.

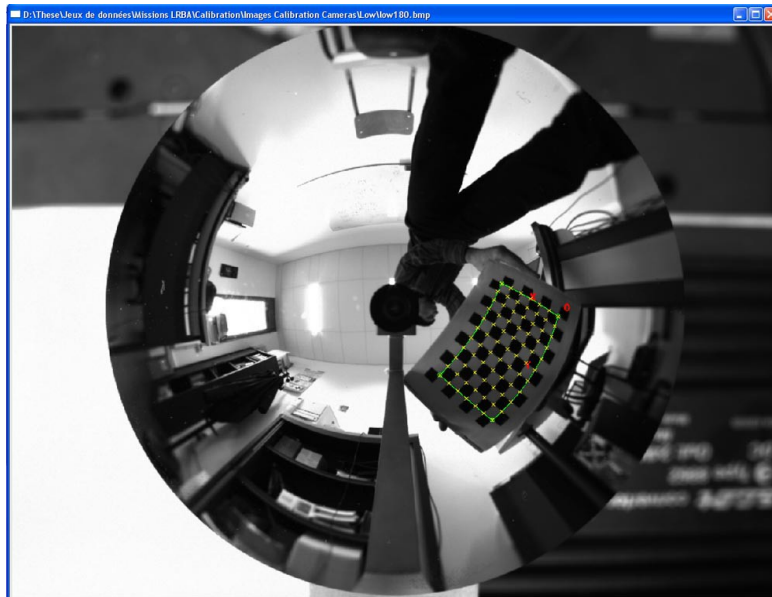


Figure 69: Résultat de l'extraction automatique des points de contrôle (en jaune).

#### 2.4.1.2. Minimisation de l'erreur de reprojection

La principale étape du calibrage consiste à minimiser l'erreur de reprojection définie par l'équation (20) grâce à l'algorithme de Levenberg-Marquardt (cf Annexe B). Les valeurs initiales

des paramètres ont été fixées par les méthodes de la section 2.4.1.1 pour les paramètres intrinsèques et par l'estimation de la pose des mires présentée dans la section précédente pour les paramètres extrinsèques. La minimisation globale est obtenue en faisant varier les paramètres intrinsèques et extrinsèques

L'algorithme de Levenberg-Marquardt nécessite la résolution de l'équation normale augmentée :

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \Delta = -\mathbf{J}^T \mathbf{e} \quad (28)$$

avec  $\lambda$  un réel qui varie d'itération en itération,  $\mathbf{I}$  la matrice identité,  $\Delta$  le vecteur de correction des paramètres,  $\mathbf{e}$  le vecteur d'erreur, et  $\mathbf{J}$  la jacobienne de la fonction de projection.

La résolution de cette équation passe par le calcul de la jacobienne de la fonction de projection. Le calibrage du système revient à estimer tous les paramètres intrinsèques du système ainsi que les paramètres extrinsèques des mires, c'est-à-dire tous les paramètres du vecteur  $\mathbf{V} = [\mathbf{V}_1^T \ \mathbf{V}_2^T \ \mathbf{V}_3^T \ \mathbf{V}_4^T]^T$ . La jacobienne de la fonction de projection s'écrit alors :

$$\frac{\partial P}{\partial \mathbf{V}} = \begin{bmatrix} \frac{\partial P}{\partial \mathbf{V}_1} & \frac{\partial P}{\partial \mathbf{V}_2} & \frac{\partial P}{\partial \mathbf{V}_3} & \frac{\partial P}{\partial \mathbf{V}_4} \end{bmatrix} \quad (29)$$

La fonction de projection  $P = K \circ D \circ H \circ W$  étant la composition de plusieurs fonctions, la jacobienne peut s'exprimer sous la forme d'un produit de dérivées partielles dont les expressions sont données en Annexe D :

$$\frac{\partial P}{\partial \mathbf{V}} = \begin{bmatrix} \frac{\partial K}{\partial D} \left[ \frac{\partial D}{\partial H} \left[ \frac{\partial H}{\partial W} \frac{\partial W}{\partial \mathbf{V}_1} \quad \frac{\partial H}{\partial \mathbf{V}_2} \right] \quad \frac{\partial D}{\partial \mathbf{V}_3} \right] \quad \frac{\partial K}{\partial \mathbf{V}_4} \end{bmatrix} \quad (30)$$

Si  $n$  est le nombre de points de contrôle par mire, et  $m$  le nombre de mires, le vecteur de paramètres a alors pour dimension  $11 + 7 \cdot m$  puisque chaque vue introduit 7 paramètres extrinsèques en plus des 11 paramètres intrinsèques du modèle. De même, la jacobienne  $\mathbf{J}$  de la fonction de projection  $P$  utilisée dans l'algorithme de Levenberg-Marquardt est cette fois de dimension  $(2n \cdot m) \times (11 + 7m)$  et a pour structure celle présentée sur la figure 70. La jacobienne contient de nombreuses valeurs nulles, représentées par des zones grisées sur la figure 70, puisque chaque mire n'est visible que dans une seule image.



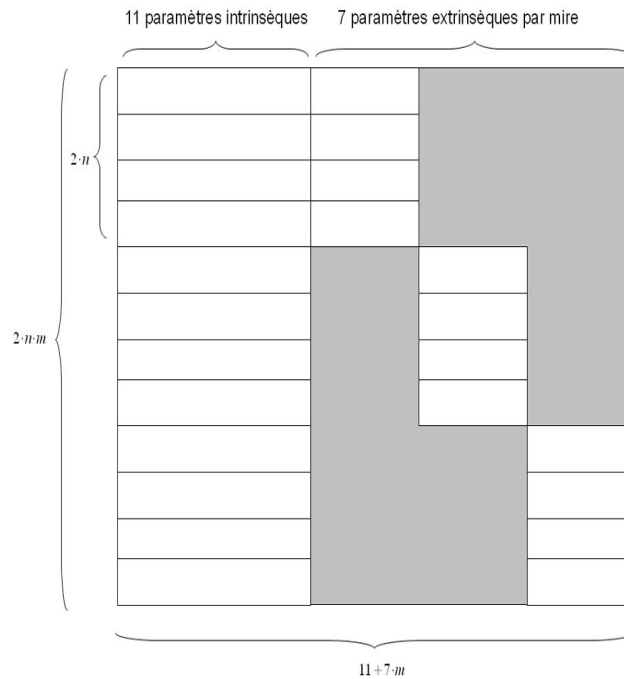


Figure 70: Structure de la jacobienne pour le calibrage. Les zones grisées correspondent à des valeurs nulles.

## 2.4.2. Estimation de la pose relative des deux capteurs

L'estimation des paramètres intrinsèques présentée précédemment permet d'établir la relation entre les points 3D et les pixels pour chacun des deux capteurs du système stéréoscopique. Afin de pouvoir obtenir des informations métriques de la scène, par exemple par triangulation, la pose relative de ces deux capteurs doit être déterminée.

### 2.4.2.1. Principe de la méthode

L'estimation de la pose relative de deux capteurs de vision nécessite une connaissance partielle de l'environnement pour pouvoir connaître le facteur d'échelle. Pour cette raison, nous proposons une méthode fondée sur l'utilisation d'une mire de calibrage dont les dimensions sont connues et qui doit être visible simultanément par les deux capteurs.

Soient  $(C_1, x_1, y_1, z_1)$  et  $(C_2, x_2, y_2, z_2)$  les repères associés aux deux capteurs du système stéréoscopique, et  $\mathbf{M}$  un point 3D (cf figure 61).

Le point  $\mathbf{M}$  de coordonnées  $[x_2 \ y_2 \ z_2 \ 1]^T$  dans le repère associé au second capteur a pour coordonnées dans le repère du premier capteur :

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} \quad (31)$$

où la translation  $\mathbf{t}=[t_x \ t_y \ t_z]^T$  et la rotation  $\mathbf{R}=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$  correspondent à la pose du

second capteur par rapport au premier.

Avec  $n$  points de contrôle, l'équation (31) permet de former le système (32) dont la résolution aboutit aux paramètres de la pose relative des deux capteurs.

$$\begin{bmatrix} x_2^1 & y_2^1 & z_2^1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_2^1 & y_2^1 & z_2^1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_2^1 & y_2^1 & z_2^1 & 0 & 0 & 1 \\ x_2^2 & y_2^2 & z_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_2^2 & y_2^2 & z_2^2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_2^2 & y_2^2 & z_2^2 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^n & y_2^n & z_2^n & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_2^n & y_2^n & z_2^n & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_2^n & y_2^n & z_2^n & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x_1^1 \\ y_1^1 \\ z_1^1 \\ x_1^2 \\ y_1^2 \\ z_1^2 \\ \vdots \\ x_1^n \\ y_1^n \\ z_1^n \end{bmatrix} \quad (32)$$

### 2.4.2.2. Méthode d'estimation de la pose relative

Le principe de l'estimation de la pose consiste à calculer la transformation rigide liant les deux capteurs, connaissant les coordonnées de points de contrôle dans les deux repères. Les coordonnées des points ne sont pas connues directement puisque la position de la mire est quelconque. Dans un premier temps, la pose de la mire doit être déterminée dans les repères associés aux deux capteurs grâce à la méthode présentée précédemment. L'estimation de la pose de la mire permet aussi une extraction automatique des points d'intérêt dans les deux images comme l'illustre la figure 71.

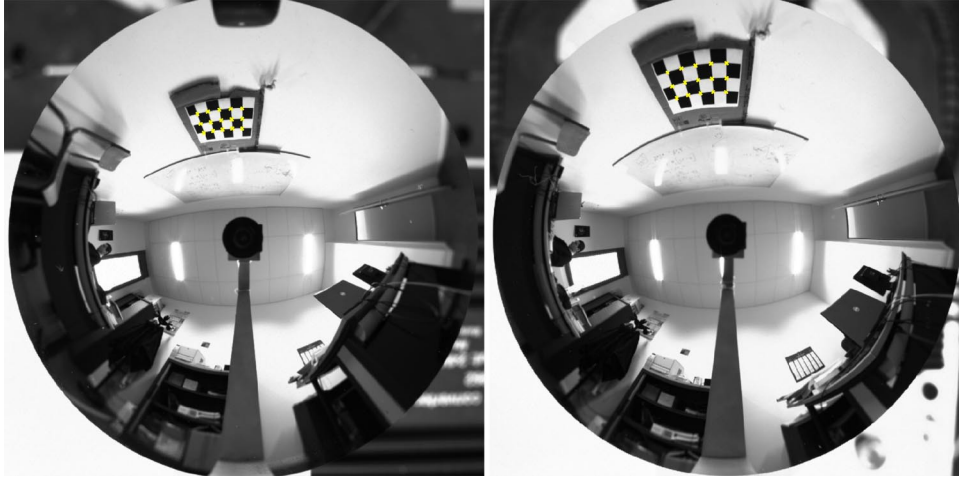


Figure 71: Points de contrôle extraits dans les images des deux capteurs (croix jaunes).

La pose de la mire étant connue, les coordonnées de ses points de contrôle sont exprimées dans chacun des deux repères et le système d'équations (32) est construit. La résolution de ce système au sens des moindres carrés permet d'obtenir le vecteur de paramètres  $\mathbf{X} = [r_{11} \ r_{12} \ r_{13} \ r_{21} \ r_{22} \ r_{23} \ r_{31} \ r_{32} \ r_{33} \ t_x \ t_y \ t_z]^T$  décrivant la pose relative des deux capteurs.

L'estimation par les moindres carrés de la matrice de rotation peut aboutir à une matrice qui ne respecte pas tout à fait les propriétés d'une matrice de rotation. Pour pallier ce problème, la matrice de rotation estimée est orthogonalisée *a posteriori*.

Enfin, la matrice essentielle peut être calculée grâce à l'équation (33) puisque la pose relative est maintenant connue. Cette matrice essentielle permet de calculer les courbes épipolaires et ainsi vérifier les résultats obtenus (*cf* section 2.5.2.3).

$$\mathbf{E} = \mathbf{R} \cdot \mathbf{S}, \text{ avec } \mathbf{S} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (33)$$

## 2.5. Résultats expérimentaux

Les deux étapes de notre méthode de calibrage, c'est-à-dire l'estimation des paramètres intrinsèques d'une part et la détermination de la pose relative d'autre part, ont été évaluées expérimentalement sur des images de synthèse et sur des images réelles.

### 2.5.1. Évaluation de l'estimation des paramètres intrinsèques

#### 2.5.1.1. Évaluation sur des données de simulation

L'estimation des paramètres intrinsèques du système est évaluée en calculant l'erreur entre la projection estimée et la projection réelle d'un ensemble de points 3D. La connaissance des

coordonnées d'un point 3D dans le repère du capteur étant une donnée impossible à obtenir précisément dans la pratique, cette évaluation a été réalisée dans un environnement de simulation.

Le capteur a été modélisé dans le logiciel de *ray-tracing* POV-Ray<sup>10</sup> en utilisant les paramètres du système réel pour le miroir et la caméra. La distance entre le foyer du miroir et le centre de la caméra a été définie afin de répondre parfaitement à la contrainte du point de vue unique. Une mire similaire à celle utilisée en pratique a ensuite été déplacée à différentes positions autour du capteur et les images omnidirectionnelles correspondantes ont été générées. La figure 72 présente quelques unes des dix images générées.

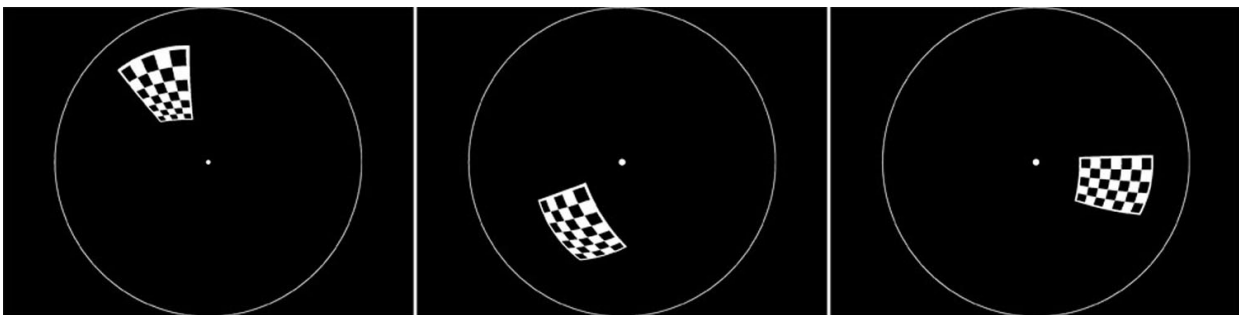


Figure 72: Extrait des images de synthèse utilisées pour le calibrage.

Ces images ont ensuite été utilisées par le logiciel de calibrage développé pour estimer les onze paramètres du modèle de la sphère d'équivalence. Le modèle de projection théorique étant connu dans le cas où la contrainte du point de vue unique est respectée [Svoboda99a], les projections des points 3D de référence peuvent être calculées et comparées à celles obtenues par le modèle estimé. Le tableau 2 résume l'erreur moyenne et l'écart-type obtenus pour un jeu de 150 points répartis dans tout l'espace 3D.

Erreur moyenne (pixels)	0,24
Écart type (pixels)	0,11

Tableau 2: Résultats du calibrage sur des images de synthèse.

La faible erreur obtenue (inférieure au pixel) permet de valider le processus de calibrage. Ces résultats sont cependant à nuancer puisqu'ils sont obtenus avec un capteur idéal, la contrainte du point de vue unique pouvant être parfaitement respectée en simulation.

### 2.5.1.2. Évaluation sur des données réelles

Le calibrage a ensuite été évalué sur des données réelles pour étudier son comportement

<sup>10</sup> <http://www.povray.org/>

sur un système ne respectant pas parfaitement la contrainte du point de vue unique. Pour cela, un ensemble d'images omnidirectionnelles contenant des mires a été séparé en deux jeux. Le premier jeu d'images a été utilisé pour calibrer le capteur, c'est-à-dire pour estimer les paramètres intrinsèques du modèle. Le second jeu a été utilisé pour calculer l'erreur entre la projection estimée des points de contrôle de la mire et la projection extraite des images.

La méthode d'évaluation consiste dans un premier temps à estimer les poses des mires du second jeu par la méthode décrite dans la section 2.4.1.1 en utilisant les paramètres intrinsèques obtenus grâce au premier jeu. La connaissance de la pose d'une mire permet ensuite de projeter tous ses points de contrôles dans le plan image : il s'agit de la projection estimée. Un détecteur de coins est alors appliqué pour obtenir les projections réelles utilisées dans le calcul des erreurs présentées dans le tableau 3.

Erreur moyenne (pixels)	0,44
Écart type (pixels)	0,26

*Tableau 3: Résultats du calibrage sur des images réelles.*

L'erreur moyenne est un peu plus importante que sur les images de synthèse, mais celle-ci reste inférieure au demi-pixel et peut donc être considérée comme très satisfaisante. Ces résultats mettent en évidence une bonne estimation des paramètres malgré l'utilisation de données bruitées puisqu'il s'agit d'images réelles issues d'un capteur qui n'est pas parfait. Ceci peut s'expliquer par les distorsions qui ont été introduites dans le modèle et qui permettent de compenser les erreurs de positionnement éventuelles entre la caméra et le miroir.

### **2.5.1.3. Évaluation qualitative**

Les données fournies par le logiciel de calibrage, c'est-à-dire l'estimation des paramètres intrinsèques, rendent possible le passage du plan image au monde tridimensionnel. En effet, les pixels du plan image peuvent alors être rétro-projetés sur la sphère d'équivalence grâce à l'équation (18). La représentation de la sphère de la figure 73 permet d'évaluer qualitativement les résultats tout en illustrant le concept de la sphère d'équivalence.

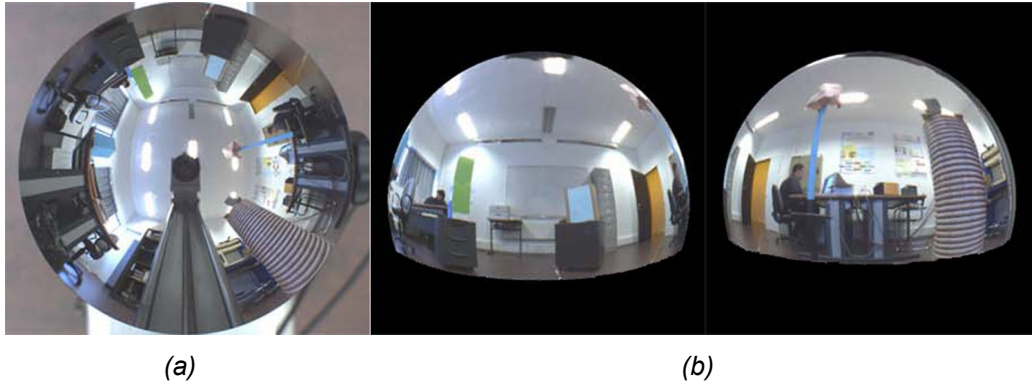


Figure 73: Représentation de la sphère d'équivalence grâce aux paramètres intrinsèques estimés. (a) Image omnidirectionnelle – (b) Vues de la projection de l'image sur la sphère d'équivalence.

## 2.5.2. Évaluation de l'estimation de la pose relative

### 2.5.2.1. Évaluation sur des données de simulation

L'estimation de la pose relative a été faite sur des images de synthèse pour une configuration verticale des deux capteurs, c'est-à-dire pour une configuration similaire au système réel. La pose relative a été estimée par la méthode présentée dans la section 2.4.2 à partir de plusieurs paires d'images contenant des mires (cf figure 74). La distance entre les deux capteurs est fixée à 10cm, et les capteurs sont supposés parfaitement alignés. Le résultat de l'estimation de la pose avec ces images est présenté dans le tableau 4.

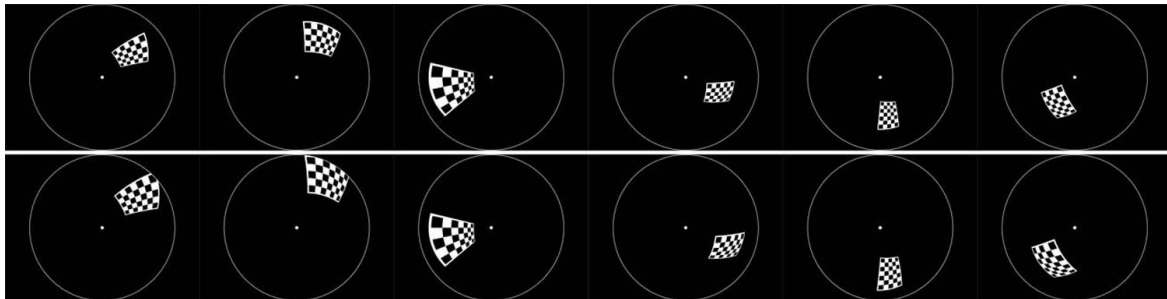


Figure 74: Paires d'images utilisées pour l'estimation de la pose relative.

	Rotation	Translation
Pose théorique	$\mathbf{R} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$	$\mathbf{T} = \begin{bmatrix} 0.000 \\ 0.000 \\ 100.000 \end{bmatrix}$
Pose estimée	$\mathbf{R} = \begin{bmatrix} 1.0000 & -0.0001 & 0.0006 \\ 0.0001 & 1.0000 & 0.0005 \\ -0.0006 & -0.0005 & 1.0000 \end{bmatrix}$	$\mathbf{T} = \begin{bmatrix} -0,413 \\ 0,142 \\ 99,948 \end{bmatrix}$

Tableau 4: Résultats de l'estimation de la pose relative sur des images de synthèse.

La rotation correspond quasiment à une matrice identité, ce qui est normal car les capteurs sont parfaitement alignés. La norme du vecteur de translation, c'est-à-dire la distance séparant les deux capteurs, est évaluée à 99,95mm au lieu des 100mm, soit une erreur de 0,05%. L'estimation de la pose relative grâce à des mires de calibrage est donc très précise.

### 2.5.2.2. Évaluation sur des données réelles

L'estimation de la pose relative a ensuite été faite sur des images réelles. Le capteur a été placé sur un rail gradué et a été déplacé par pas de 10cm. A chaque position, une image omnidirectionnelle a été prise dans le but de calculer les déplacements du capteur par rapport à la position d'origine. Les déplacements sont estimés par la méthode présentée dans la section 2.4.2 en utilisant cinq mires placées dans l'environnement du capteur, comme l'illustre la figure 75. Les résultats obtenus sont récapitulés dans le tableau 5.

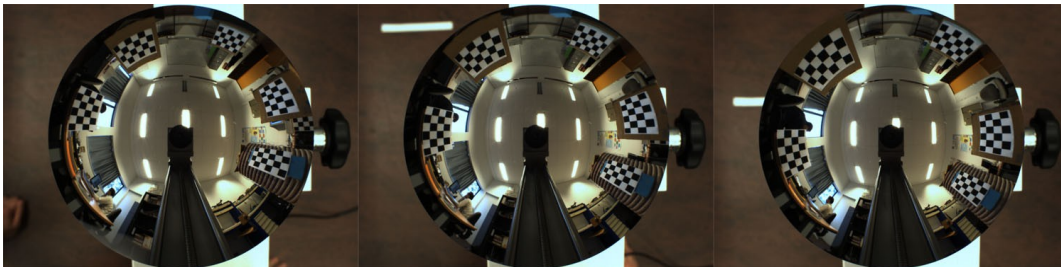


Figure 75: Extraits de la séquence d'images utilisée pour l'estimation des déplacements.

Déplacement (mm)	100	200	300	400	500	600
Estimation (mm)	100,13	200,77	296,48	393,66	494,7	594,6
Erreur (%)	0,13	0,39	1,17	1,59	1,06	0,9

Tableau 5: Résultats de l'estimation des déplacements.

L'erreur est plus élevée qu'avec les données de simulation, mais celle-ci reste en moyenne inférieure à 0,9%. L'augmentation de cette erreur s'explique évidemment par le bruit présent dans les images réelles, mais également par une accumulation des erreurs. En effet, il ne faut pas oublier que l'estimation de la pose relative nécessite préalablement l'estimation des paramètres intrinsèques.

### 2.5.2.3. Évaluation qualitative

La pose relative des deux capteurs fournit des informations sur la géométrie épipolaire du système puisque celle-ci permet de calculer la matrice essentielle du système [Longuet-Higgins81a]. L'étude des propriétés de la géométrie épipolaire permet par conséquent de vérifier la cohérence de la pose estimée.

Pour chaque pixel sélectionné sur une image, la courbe épipolaire correspondante, c'est-à-dire la courbe sur laquelle doit se trouver le pixel homologue, peut être déterminée sur l'autre image. La figure 76 met en évidence le respect de cette propriété puisque pour chaque pixel sélectionné (en rouge) la courbe épipolaire correspondante est tracée sur la deuxième image et il est possible de vérifier que le pixel homologue appartient bien à cette courbe.

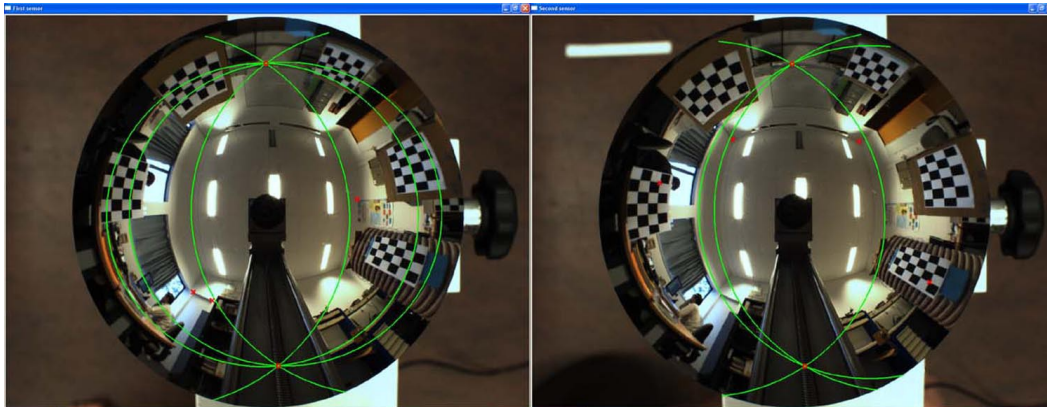


Figure 76: Courbes épipolaires (en vert) correspondant aux pixels sélectionnés (en rouge)

## 2.6. Conclusion et perspectives

Nous avons présenté dans ce chapitre une méthode permettant de calibrer un système de stéréovision omnidirectionnelle. L'utilisation d'une mire plane et d'un modèle générique rend la phase de calibrage particulièrement facile à mettre en œuvre et compatible avec tout type de capteur, à condition toutefois qu'il soit central. Le calibrage s'effectue en deux temps : les paramètres intrinsèques des deux capteurs sont d'abord estimés, puis leur pose relative est déterminée. Nos travaux sur le calibrage ont abouti à des logiciels libres, puisque développés en langage C++ avec des bibliothèques *open-source*, et disponibles gratuitement sur notre site<sup>11</sup>

Les résultats expérimentaux ont mis en évidence que le calibrage fournit une bonne estimation des paramètres du système, ce qui est indispensable pour obtenir une reconstruction 3D précise. Grâce à l'estimation de la distance séparant les deux capteurs, il est de plus possible d'obtenir une reconstruction 3D euclidienne, là où la majorité des systèmes stéréoscopiques ne fournit qu'une reconstruction à un facteur d'échelle près, ou en mesurant celui-ci.

Certains aspects du processus de calibrage méritent néanmoins d'être améliorés. Il serait par exemple intéressant d'automatiser entièrement le calibrage en développant une méthode de détection automatique des mires comme il en existe pour les caméras perspectives [Ha09a]. Un algorithme de détection fonctionnant à la cadence vidéo permettrait de plus de ne sauvegarder que les images sur lesquelles la mire serait entièrement visible et ses points correctement

<sup>11</sup> <http://omni3d.esigelec.fr/doku.php/thesis/r3d/start>



déTECTÉS.

Dans l'implantation actuelle, la pose relative est déterminée par la résolution au sens des moindres carrés d'un système d'équations linéaires. Pour obtenir la linéarité de ce système, la rotation a été exprimée par une matrice, ce qui n'est pas la représentation la plus judicieuse. Une piste intéressante serait donc d'utiliser les quaternions pour exprimer la pose, et de résoudre le système par une approche itérative de type Levenberg-Marquardt.

## Chapitre 3

# Reconstruction 3D statique

### Sommaire

---

<b>3.1.Introduction.....</b>	<b>101</b>
<b>3.2.Reconstruction 3D dense.....</b>	<b>101</b>
3.2.1.Les méthodes locales.....	102
3.2.1.1.Mesures de corrélation.....	103
3.2.1.2.Utilisation de la couleur.....	104
3.2.1.3.Méthodes de rejet des mauvais appariements.....	105
3.2.1.4.Résultats expérimentaux.....	106
3.2.2.Prise en compte des contraintes globales.....	108
3.2.2.1.Définition des contraintes globales.....	109
3.2.2.2.La programmation dynamique.....	111
3.2.2.3.Résultats expérimentaux.....	118
3.2.3.Triangulation de points.....	120
3.2.3.1.Méthode du point milieu.....	120
3.2.3.2.Résultats expérimentaux.....	121
<b>3.3.Reconstruction 3D de droites.....</b>	<b>123</b>
3.3.1.Détection de droites dans les images omnidirectionnelles.....	123
3.3.2.Triangulation de droites.....	127
<b>3.4.Conclusion et perspectives.....</b>	<b>129</b>

### 3.1. Introduction

Les chapitres 1 et 2 ont été consacrés à la conception et au calibrage du système de reconstruction 3D. Ce système a été conçu de sorte à faciliter la reconstruction 3D à partir d'une paire d'images stéréoscopiques acquises simultanément. Cette propriété confère au système la capacité à reconstruire son environnement sans déplacement au sein de celui-ci, c'est-à-dire dans une configuration statique.

Les méthodes de reconstruction 3D applicables à la reconstruction 3D passive et sans connaissance *a priori* peuvent être classées en deux principales catégories. La première catégorie regroupe les méthodes volumétriques, c'est-à-dire les méthodes de *space carving* et de *voxel coloring* (cf section 1.2.2.8). Ces méthodes donnent d'excellents résultats pour la reconstruction d'objets de taille limitée, en particulier lorsqu'il est possible de les différencier de l'arrière plan et d'avoir des prises de vue couvrant la totalité de ceux-ci [Seitz97a] [Kutulakos00a]. L'application de ces méthodes à la reconstruction de scènes complexes est délicate et peu de travaux y ont été consacrés [Ragot09a] [Rossi09a].

La seconde catégorie regroupe les méthodes de stéréovision, qui consistent à mettre en correspondance des entités présentes dans les paires d'images stéréoscopiques, et à les trianguler pour obtenir leur position dans l'espace. Les entités peuvent être les pixels eux-mêmes, ou des primitives plus élaborées telles que des droites ou des plans. L'étape clé des algorithmes de stéréovision est la phase de mise en correspondance puisqu'elle conditionne le nombre et la justesse des entités reconstruites.

Les travaux présentés dans ce chapitre sont consacrés dans un premier temps à la reconstruction 3D dense de l'environnement. Celle-ci a pour but de mettre en correspondance un maximum de pixels pour obtenir le modèle 3D le plus complet possible. Deux phases sont nécessaires : la mise en correspondance des pixels, par des méthodes locales ou globales, puis la triangulation qui permet d'obtenir les coordonnées des points 3D correspondants. La seconde partie du chapitre concerne la reconstruction 3D de droites, dont la réussite dépend fortement de leur détection dans les images omnidirectionnelles. Des résultats expérimentaux réalisés sur des données réelles sont présentés tout au long de ce chapitre.

### 3.2. Reconstruction 3D dense

La reconstruction 3D dense consiste à essayer de mettre en correspondance tous les pixels des images de la paire stéréoscopique afin d'obtenir une carte de disparité la plus complète possible. Cette carte, aussi appelée carte de profondeur, illustre la distance entre le système et les

objets, et permet d'aboutir au modèle 3D de l'environnement. L'intérêt de la reconstruction dense réside dans la suppression de la phase d'extraction des primitives puisqu'un stéréo-correspondant est recherché pour chaque pixel, ce qui limite le risque de fausse détection dans l'une ou l'autre des images.

### 3.2.1. Les méthodes locales

Le principe des méthodes locales est de déterminer, pour chaque pixel et son voisinage de la première image, quel est le pixel (et son voisinage) le plus ressemblant dans la deuxième image (cf figure 77). Ces méthodes utilisent la corrélation pour déterminer la « ressemblance » des pixels issus des deux images [Zhang93a]. Cette approche repose sur l'hypothèse principale de la constance de la luminosité, c'est-à-dire que les pixels correspondants à un même point 3D sont supposés avoir la même couleur (ou luminance dans le cas d'images en niveaux de gris). Cette hypothèse limite le fonctionnement de ces algorithmes à des surfaces lambertienne, i.e. des surfaces dont la luminance émise est égale dans toutes les directions.

La recherche ne s'effectue généralement pas dans toute l'image, mais uniquement dans une zone autour du point d'origine. Lorsque la géométrie épipolaire du système stéréoscopique est connue, la recherche peut être contrainte à une droite dans le cas d'images classiques, et à une courbe dans le cas d'images omnidirectionnelles.

Le résultat recherché lors de la mise en correspondance est la **disparité**  $\mathbf{d} = [d_x \ d_y]^T$ , c'est-à-dire la différence de position des pixels représentant le même point 3D dans les deux images. Ainsi, lorsque cette disparité est calculée pour chaque pixel, la carte des disparités, qui permet d'évaluer la distance séparant les objets du système stéréoscopique, peut être visualisée.

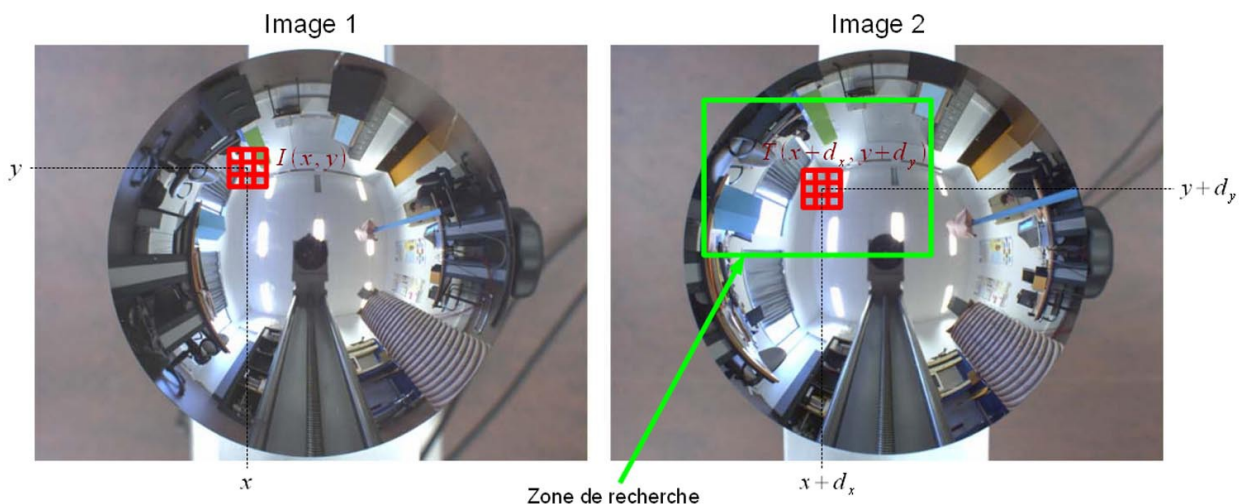


Figure 77: Principe de la corrélation.

### 3.2.1.1. Mesures de corrélation

Une mesure de corrélation est, par définition, une mesure qui permet de quantifier la similarité entre deux pixels et leur voisinage. De nombreuses mesures de corrélation ont été proposées dans la littérature et il est par conséquent très difficile de déterminer la « mesure optimale » [Chambon03a]. Si certaines de ces techniques sont plus performantes que d'autres au niveau du résultat obtenu, le temps de calcul doit également être pris en compte car de grandes différences existent entre les mesures. Nous avons choisi de travailler avec deux scores de corrélations différents : le score SAD pour son temps de calcul limité et le score ZNCC pour ses performances reconnues.

#### SAD

Le score de corrélation SAD (*Sum of Absolute Differences*) a été choisi pour le faible temps de calcul qu'il requiert. Il est en effet très rapide à calculer de par sa simplicité : il consiste à prendre la somme des valeurs absolues des différences entre la luminance des pixels. La formule (34) permet de calculer le score SAD pour le pixel de coordonnées  $[x \ y]^T$  de la première image  $\mathbf{I}$  et le pixel de coordonnées  $[x+d_x \ y+d_y]^T$  dans la seconde image  $\mathbf{T}$ .

$$SAD(x, y, \mathbf{d}) = \sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} |\mathbf{I}(x+i, y+j) - \mathbf{T}(x+d_x+i, y+d_y+j)| \quad (34)$$

où  $w$  est la demi-taille de la fenêtre de corrélation.

#### ZNCC

La corrélation ZNCC (*Zero mean Normalized Cross-Correlation*) procure de meilleurs résultats que les méthodes simples, telles que SAD et SSD (*Sum of Squared Differences*). Cette méthode a l'avantage d'être moins sensible aux changements d'illumination que le score SAD, et de fournir un score borné, compris entre -1 et 1, qui peut servir d'indice de confiance. En effet, lorsque le score de corrélation est supérieur à un certain seuil (par exemple 0,8), il est fort probable que l'appariement soit correct. Ce score repose sur le calcul d'un produit scalaire, donné par la formule (35).

$$ZNCC(x, y, \mathbf{d}) = \frac{\sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} (\mathbf{I}(x+i, y+j) - \overline{I_w(x, y)}) (\mathbf{T}(x+d_x+i, y+d_y+j) - \overline{T_w(x+d_x, y+d_y)})}{\sqrt{\sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} (\mathbf{I}(x+i, y+j) - \overline{I_w(x, y)})^2} \sqrt{\sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} (\mathbf{T}(x+d_x+i, y+d_y+j) - \overline{T_w(x+d_x, y+d_y)})^2}} \quad (35)$$

où  $\overline{I_w(x, y)}$  et  $\overline{T_w(x+d_x, y+d_y)}$  sont les moyennes de la luminance dans les fenêtres centrées autour des pixels d'intérêt.

### 3.2.1.2. Utilisation de la couleur

Les mesures de corrélation sont généralement utilisées sur des images en niveaux de gris. Or, les caméras couleurs se sont démocratisées et sont de plus en plus utilisées dans les applications de vision par ordinateur.

De récents travaux ont mis en évidence l'apport de la couleur par rapport à l'utilisation des niveaux de gris pour la mise en correspondance par corrélation [Chambon04a], notamment en présence d'ambiguïtés telles que les zones d'occultations. L'utilisation de la couleur implique d'une part de généraliser les scores de corrélation à trois dimensions, et d'autre part de choisir un espace de couleur adapté.

Soient  $c$  le canal de l'image, et  $\mathbf{I}_c(x, y)$  la valeur de la composante  $c$  du pixel de coordonnées  $[x \ y]^T$  dans l'image  $\mathbf{I}$ . Par exemple, dans l'espace de couleurs RVB, le canal  $c=1$  correspond à la couleur rouge, le canal  $c=2$  à la couleur verte, et le canal  $c=3$  à la couleur bleue.

Le score de corrélation SAD, précédemment défini pour les images en niveaux de gris, devient, dans un espace de couleur à trois composantes :

$$SAD(x, y, \mathbf{d}) = \sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} \sum_{c=1}^{c=3} |\mathbf{I}_c(x+i, y+j) - \mathbf{T}_c(x+d_x+i, y+d_y+j)| \quad (36)$$

Le score de corrélation ZNCC est quant à lui défini par :

$$ZNCC \begin{pmatrix} x \\ y \\ \mathbf{d} \end{pmatrix} = \frac{\sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} \sum_{c=1}^{c=3} (\mathbf{I}_c(x+i, y+j) - \overline{\mathbf{I}_c^w(x, y)}) (\mathbf{T}_c(x+d_x+i, y+d_y+j) - \overline{\mathbf{T}_c^w(x+d_x, y+d_y)})}{\sqrt{\sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} \sum_{c=1}^{c=3} (\mathbf{I}_c(x+i, y+j) - \overline{\mathbf{I}_c^w(x, y)})^2} \sqrt{\sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} \sum_{c=1}^{c=3} (\mathbf{T}_c(x+d_x+i, y+d_y+j) - \overline{\mathbf{T}_c^w(x+d_x, y+d_y)})^2}} \quad (37)$$

Le choix de l'espace de couleur repose sur la conclusion des travaux de Chambon [Chambon04a] qui ont montré que le système  $H_1 H_2 H_3$  fournit les meilleurs résultats. Cet espace de couleurs, introduit par Braquelaire et Brun [Braquelaire97a], met en évidence les trois directions privilégiées par la vision humaine : blanc/noir, rouge/vert, bleu/jaune. Cet espace aboutit à des résultats proches du  $La^*b^*$  (un paramètre de luminance et deux paramètres de chrominance) mais le temps de calcul nécessaire pour faire la conversion est beaucoup plus rapide car il s'agit d'une combinaison linéaire d'entiers. Les composantes du système  $H_1 H_2 H_3$  peuvent être calculées, à partir de l'espace de couleurs RVB, par :

$$\begin{aligned} H_1 &= R + V, & H_1 &\in [0; 510] \\ H_2 &= R - V, & H_2 &\in [-255; 255] \\ H_3 &= B - \frac{R+V}{2}, & H_3 &\in [-255; 255] \end{aligned} \quad (38)$$

### 3.2.1.3. Méthodes de rejet des mauvais appariements

Les méthodes de mise en correspondance locales ne sont pas suffisamment performantes pour fournir un modèle fidèle à la réalité. En effet, ces méthodes aboutissent à de nombreux mauvais appariements, c'est-à-dire à des points qui ont été mis en correspondance alors qu'il ne s'agissait pas des projections d'un même point 3D. Nous avons donc, pour limiter ces erreurs, introduit des critères supplémentaires que nous présentons dans cette section.

#### Recherche selon les rayons des images omnidirectionnelles

La structure géométrique de notre système a été étudiée de sorte à simplifier la géométrie épipolaire. La configuration axiale des capteurs catadioptriques permet en effet d'obtenir des épipolaires qui correspondent aux rayons des images omnidirectionnelles. Cette propriété peut être exploitée pour limiter la zone de recherche le long de ces rayons, plutôt que sur la totalité de l'image. Le risque de commettre un mauvais appariement se retrouve ainsi diminué, et le temps de calcul fortement amélioré.

Pour extraire les pixels appartenant à un même rayon, l'algorithme de Bresenham [Bresenham65a], un algorithme de tracé de segment courant en infographie, est utilisé. Cet algorithme a été implanté sur des microprocesseurs très anciens et est par conséquent particulièrement optimisé pour n'utiliser ni multiplication, ni division et pour ne nécessiter que très peu de mémoire. Le but de cet algorithme est de déterminer quels pixels doivent être « tracés » pour représenter au mieux une droite, comme l'illustre la figure 78.

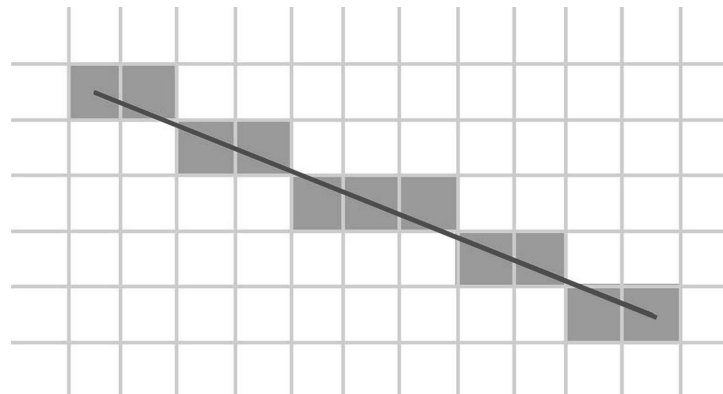


Figure 78: Principe de l'algorithme de Bresenham.

Grâce à l'implantation de cet algorithme, il est possible de définir quels sont les pixels appartenant à un même rayon, et par conséquent à une ligne épipolaire.

#### Corrélation retour

La corrélation retour est une technique utilisée en stéréovision classique pour rejeter un maximum de mauvais appariements. Elle consiste à permuter les deux images et à refaire une

corrélation pour vérifier que l'appariement aboutit au même résultat dans les deux sens. Si la corrélation retour aboutit à un appariement différent, le pixel correspondant n'est pas apparié.

Une version différente de la corrélation retour a été proposée par Werth [Werth00a] afin de limiter le temps de calcul, qui est doublé avec la méthode précédente. Werth propose de ne conserver que les pixels ayant les meilleurs scores de corrélation et de calculer les scores de corrélation de ces pixels grâce à une autre méthode. Si les deux corrélations aboutissent au même résultat, l'appariement est conservé, sinon le point n'est pas apparié.

L'utilisation de la corrélation retour est la raison pour laquelle nous avons présenté deux mesures de corrélation différentes dans la section 3.2.1.1. La corrélation ZNCC est utilisée pour la corrélation « aller », car elle présente de bonnes performances, et la corrélation SAD est utilisée pour la corrélation retour car elle permet de vérifier un appariement rapidement.

### **Distance maximale entre deux rayons**

Les pixels mis en correspondance sont ensuite triangulés pour calculer les coordonnées du point 3D correspondant. Or, comme cela est expliqué dans la section 3.2.3, les rayons lumineux correspondant à deux pixels appariés ne s'intersectent pas nécessairement. Un test sur la distance minimale les séparant permet donc de rejeter les appariements aberrants. Cette méthode de rejet ne s'effectue par conséquent pas au niveau de l'appariement, mais à celui de la triangulation.

#### **3.2.1.4. Résultats expérimentaux**

Les résultats expérimentaux présentés dans cette section ont pour but de justifier les choix que nous avons fait en matière de corrélation. Ces choix ont été guidés par les résultats de travaux existants et sont confortés par nos expérimentations. Sur les figures 79 et 80, les appariements sont illustrés par un trait vert reliant la projection d'un même point 3D sur les deux images. Les grands segments verts représentent des erreurs de mise en correspondance, le déplacement entre les deux acquisitions étant faible.

### **Niveaux de gris vs RGB**

Pour illustrer l'apport de l'utilisation de la couleur dans le processus de mise en correspondance, nous avons comparé les résultats d'un appariement en niveaux de gris et d'un appariement en couleur en utilisant l'espace RGB sur plusieurs types de scènes. Les figures 79 a et 79 b illustrent les résultats obtenus sur une scène intérieure. Sur les 420 points de Harris détectés, seuls 110 points ont été appariés (après la corrélation retour) en niveaux de gris contre 227 points en couleur. L'utilisation de la couleur améliore donc considérablement le processus de mise en correspondance.



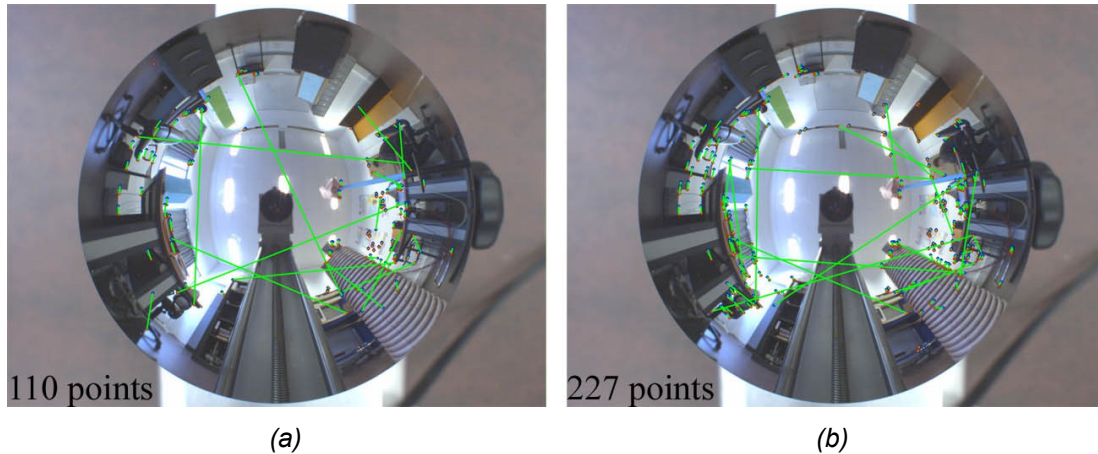


Figure 79: Comparaison d'un appariement en niveaux de gris (a) et d'un appariement en couleur (b).  
Le nombre d'appariements corrects est multiplié par deux en utilisant l'information couleur.

### Comparaison entre les espaces de couleurs RGB et $H_1H_2H_3$

Le choix de l'espace de couleurs utilisé pour la corrélation a également un impact sur les performances. Le nombre de points considérés comme correctement appariés passe de 227 pour le RGB à 221 pour le  $H_1H_2H_3$ . Les six points qui n'ont pas été appariés avec le  $H_1H_2H_3$  correspondaient à des mauvais appariements avec le RGB. L'espace de couleurs  $H_1H_2H_3$  est donc meilleur que l'espace RGB car il y a autant d'appariements corrects et moins de mauvais appariements.

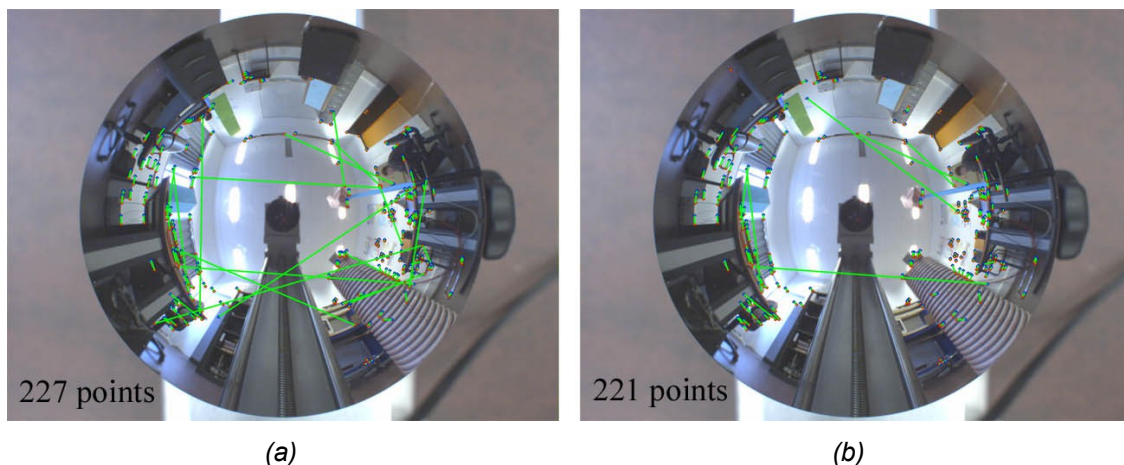


Figure 80: Comparaison d'un appariement en RGB (a) et d'un appariement en  $H_1H_2H_3$  (b).  
L'utilisation de l'espace couleur  $H_1H_2H_3$  réduit le nombre de mauvais appariements.

### Rejet des mauvais appariements

L'objectif final de la mise en correspondance est l'obtention de la carte de disparité reflétant la distance entre les objets et la caméra. Les objets les plus proches de la caméra sont représentés par les couleurs les plus chaudes, et les objets plus lointain par des couleurs froides.

La figure 81b représente la carte de disparité obtenue par une corrélation ZNCC dans l'espace de couleurs  $H_1H_2H_3$ , mais sans appliquer les méthodes de rejet des mauvais appariements. La carte de disparité est très complète, mais fortement bruitée à cause de la présence de nombreux mauvais appariements.

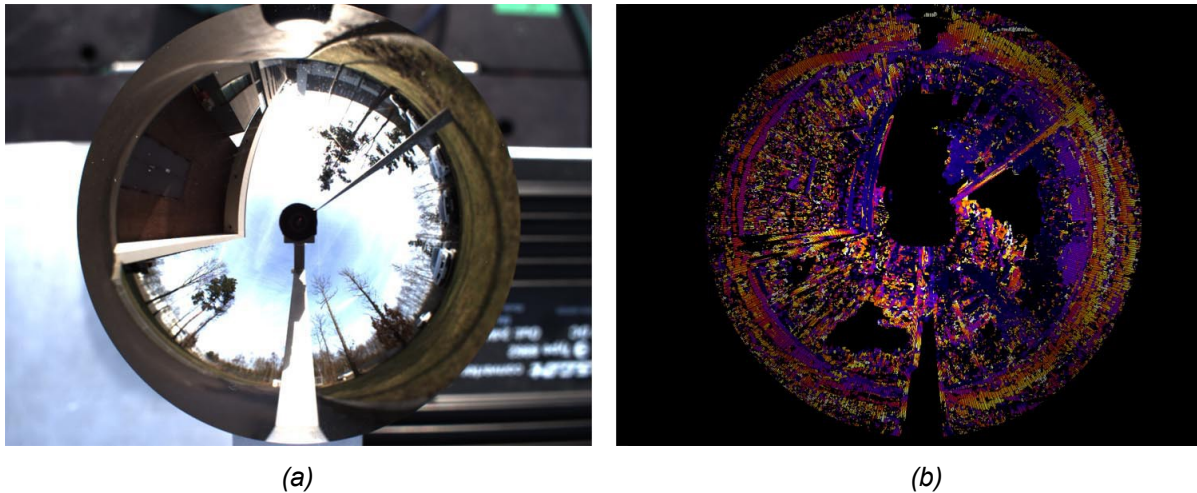


Figure 81: Carte de disparité obtenue avec les méthodes locales sans rejet.  
(a) Image omnidirectionnelle – (b) Carte de disparité

Lorsque les méthodes de rejet des mauvais appariements présentées dans cette section sont mises en œuvre, la carte de disparité obtenue présente peu d'erreurs de mise en correspondance, comme l'illustre la figure 82 b. La carte n'est cependant plus aussi dense, ce qui conduit à un modèle 3D composé de trop peu de points.

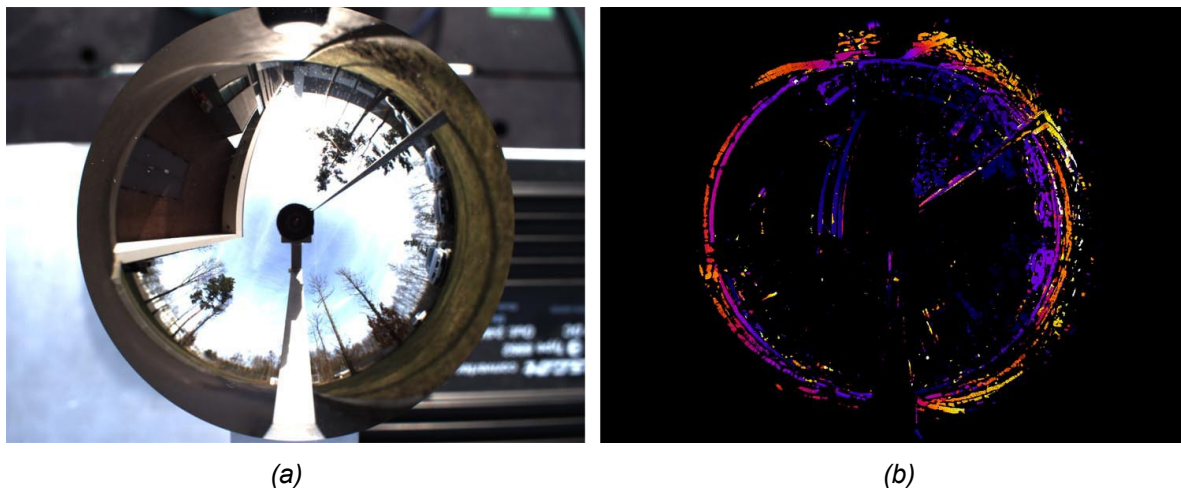


Figure 82: Carte de disparité obtenue par les méthodes locales avec rejet.  
(a) Image omnidirectionnelle – (b) Carte de disparité

### 3.2.2. Prise en compte des contraintes globales

La carte de disparité obtenue par des méthodes locales présente de nombreux artefacts qui

conduisent à un modèle 3D erroné si des techniques de rejet des mauvais appariements ne sont pas mises en œuvre. Ces artefacts sont dus aux ambiguïtés provoquées par une absence de texture, par des variations d'illumination ou encore par la présence d'occultations. Si les techniques de rejet sont utilisées, la carte de disparité n'est plus assez dense pour fournir un modèle 3D interprétable.

Pour améliorer les algorithmes de mise en correspondance, il est indispensable d'imposer des contraintes supplémentaires, appelées contraintes globales car elles s'appliquent à l'ensemble des appariements. La fonction coût qui permettra de valider l'appariement est alors évaluée sur l'ensemble de l'image ou de la ligne épipolaire, et non plus uniquement sur un pixel et son voisinage.

Nous présenterons dans un premier temps les contraintes globales les plus couramment utilisées que nous illustrerons pour le cas du capteur sphérique, puis nous présenterons la méthode que nous avons retenue et adaptée à la vision omnidirectionnelle pour prendre en compte ces contraintes lors de la phase de mise en correspondance.

### **3.2.2.1. Définition des contraintes globales**

Nous avons vu que l'utilisation des critères de ressemblance n'est en général pas suffisante pour mettre en correspondance des primitives sans ambiguïté. Une primitive de la première image peut en effet ressembler à plusieurs primitives de la deuxième image, notamment lorsque la scène ne présente pas une texture suffisante ou lorsque cette texture est répétitive.

Pour lever ces ambiguïtés, il est nécessaire de prendre en compte des contraintes globales. Il existe dans la littérature de nombreuses contraintes globales ([Ayache89a] [Horaud95a] [Pollefeys00a]), mais nous ne présenterons que celles qui apportent une amélioration significative, à savoir la **contrainte d'unicité**, la **contrainte d'ordre** et la **contrainte de continuité de la disparité**. A notre connaissance, ces contraintes ont toujours été présentées pour les caméras perspectives. Nous proposons donc d'illustrer ces contraintes pour les capteurs sphériques.

Il est important de signaler que les contraintes globales qui vont être présentées ci-dessous ne sont pas toujours vraies. En effet, il existe des contre-exemples qui violent chacune de ces règles mais, dans la majorité des cas, les résultats obtenus en imposant ces contraintes sont meilleurs qu'une mise en correspondance fondée sur les méthodes locales.

#### **La contrainte d'unicité**

La contrainte d'unicité exprime le fait qu'une primitive d'une image a au plus un correspondant dans l'autre image, et inversement. Ainsi, sur la figure 83, le point  $X_{s1}$  de l'image

Il n'a qu'un homologue  $X_{S2}$  dans l'image 2. Cette contrainte suppose qu'il n'existe aucun couple de points 3D alignés avec le centre optique d'un des capteurs.

La contrainte d'unicité peut se justifier en supposant l'opacité des points, ce qui conduit à ne considérer que le point physique le plus proche de la caméra ( $X_1$  sur la figure 83) lors d'un alignement de plusieurs points avec le centre optique [Ayache89a].

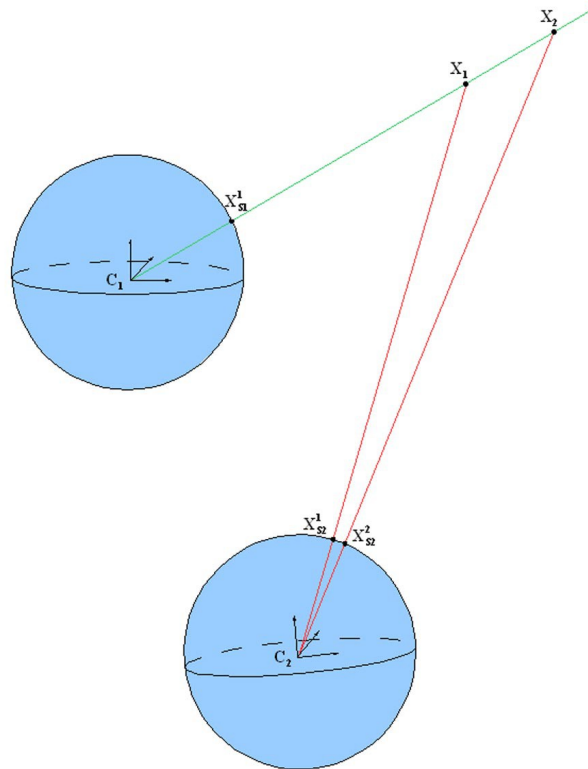


Figure 83: La contrainte d'unicité.

### La contrainte d'ordre

La contrainte d'ordre suppose la conservation de l'ordre des points homologues le long de deux épipolaires [Ayache89a].

La figure 84 illustre cette contrainte. Si trois points  $X_1$ ,  $X_2$  et  $X_3$  se projettent respectivement en  $X_{S1}^1$ ,  $X_{S1}^2$  et  $X_{S1}^3$  sur la première image, les projections de ces points dans la seconde image conserveront le même ordre :  $X_{S2}^1$ ,  $X_{S2}^2$  et  $X_{S2}^3$ .

La contrainte d'ordre est violée lorsque la scène observée contient des objets ayant des surfaces transparentes fortement inclinées par rapport au plan image [Horaud95a].

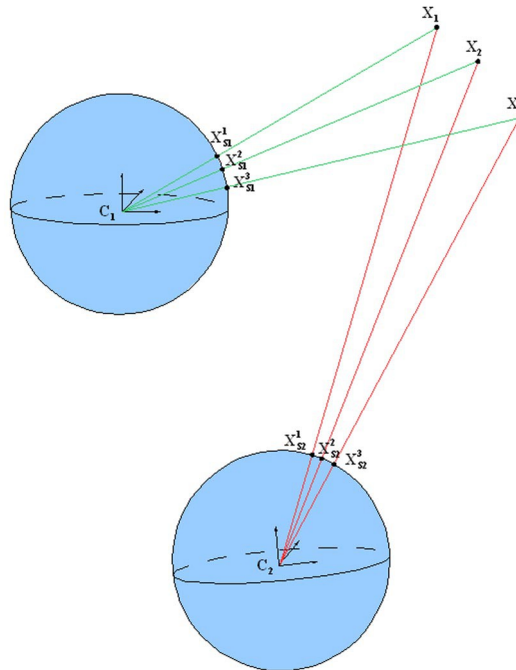


Figure 84: La contrainte d'ordre.

### La contrainte de continuité de la disparité

La contrainte de continuité de la disparité suppose que la scène observée est composée d'objets physiques dont on peut approximer la structure spatiale par des primitives, ce qui signifie que la disparité d'un même objet varie de façon continue [Ayache89a]. Les discontinuités de la disparité apparaissent aux frontières des objets physiques.

#### 3.2.2.2. La programmation dynamique

La programmation dynamique est une technique bien connue en vision classique pour prendre en compte les contraintes globales lors de la phase d'appariement. Il s'agit d'une technique d'optimisation sous contrainte inventée par Richard Bellman en 1953, qui consiste à déduire la solution optimale d'un problème à partir des solutions optimales de sous problèmes plus simples.

Cette technique peut être appliquée au problème de la mise en correspondance en cherchant un chemin de coût maximum à travers une matrice composée de tous les appariements possibles. La mise en correspondance tient alors compte des contraintes globales, puisque la fonction coût est évaluée sur l'ensemble de la séquence à appairer, et non plus sur les primitives indépendamment.

Les premiers travaux sur l'application de la programmation dynamique au problème de la

mise en correspondance sont ceux de Baker et Binford [Baker81a], suivis par ceux de Ohta et Kanade [Ohta85a]. La mise en correspondance ne se fait alors pas sur les pixels mais sur les contours, ce qui ne permet pas d'obtenir une reconstruction 3D dense de l'environnement. [Cox96a] et [Bobick99a] proposent d'utiliser la programmation dynamique pour effectuer une mise en correspondance dense, c'est-à-dire en utilisant les pixels comme primitives d'appariement, et obtiennent des résultats comportant beaucoup moins d'erreurs que dans le cas des méthodes locales.

Le nombre de travaux sur l'application de la programmation dynamique aux images omnidirectionnelles est en revanche très réduit. Ceci peut s'expliquer par le fait que la mise en correspondance doit être restreinte par la contrainte épipolaire (le problème doit être bi-dimensionnel). Or, la géométrie épipolaire des capteurs catadioptriques est plus complexe que celle des caméras perspectives car les épipolaires ne sont plus des lignes mais des coniques.

[Wei00a] applique la programmation dynamique à un capteur catadioptrique utilisant un miroir hyperboloïde en déplacement. Comme le déplacement est supposé quelconque, la recherche doit s'effectuer le long des courbes épipolaires qui sont, dans le cas général, des coniques. L'extraction des valeurs des pixels de la courbe épipolaire et de celles de leurs voisins est donc une étape complexe. Des résultats intéressants sont soulignés mais une carte de disparité dense n'est pas présentée.

[Lee01a] utilise la programmation dynamique pour appairer les droites verticales d'un environnement structuré (portes, arrêtes des murs, *etc*). Sa méthode consiste à déplacer un miroir conique mais il ne prend en compte que le cas d'une translation sans rotation, ce qui est difficilement réalisable avec un robot mobile. De plus, il suppose les déplacements du robot connus pour effectuer la triangulation, ce qui pose des problèmes de précision lorsque ces informations sont obtenues par odométrie.

Enfin, [Correa05a] propose un système stéréoscopique fondé sur un miroir hyperboloïde à deux lobes. Cette configuration a l'avantage de simplifier la recherche le long des épipolaires, mais les résultats de l'appariement utilisant la programmation dynamique ne sont présentés que sur des images de synthèse d'un environnement extrêmement simple puisqu'il n'est composé que de deux cubes.

Nous proposons donc d'appliquer la programmation dynamique à nos images omnidirectionnelles en prenant en compte la spécificité de la configuration de notre capteur qui simplifie la géométrie épipolaire. Nous utilisons également les résultats présentés précédemment (*cf* 3.2.1) pour améliorer les performances (utilisation de la couleur, changement d'espace de couleurs, choix d'un score de corrélation adapté, *etc*).



### Extraction des pixels des lignes épipolaires

La programmation dynamique ne peut s'appliquer qu'à des problèmes bi-dimensionnels. Il est par conséquent nécessaire de restreindre la recherche des pixels homologues par la contrainte épipolaire : le problème se réduit alors à la mise en correspondance des pixels d'une ligne de la première image avec ceux d'une ligne de la deuxième image.

Le géométrie de notre capteur stéréoscopique permet de déterminer rapidement les lignes épipolaires car elles correspondent aux rayons des images omnidirectionnelles. Le traitement des deux images se fera donc en balayant l'image angulairement et en extrayant les rayons définis par l'angle  $\theta$  (cf figure 85). Le pas de variation de l'angle  $\theta$  est déterminé en fonction de la résolution et/ou du temps de calcul désirés.

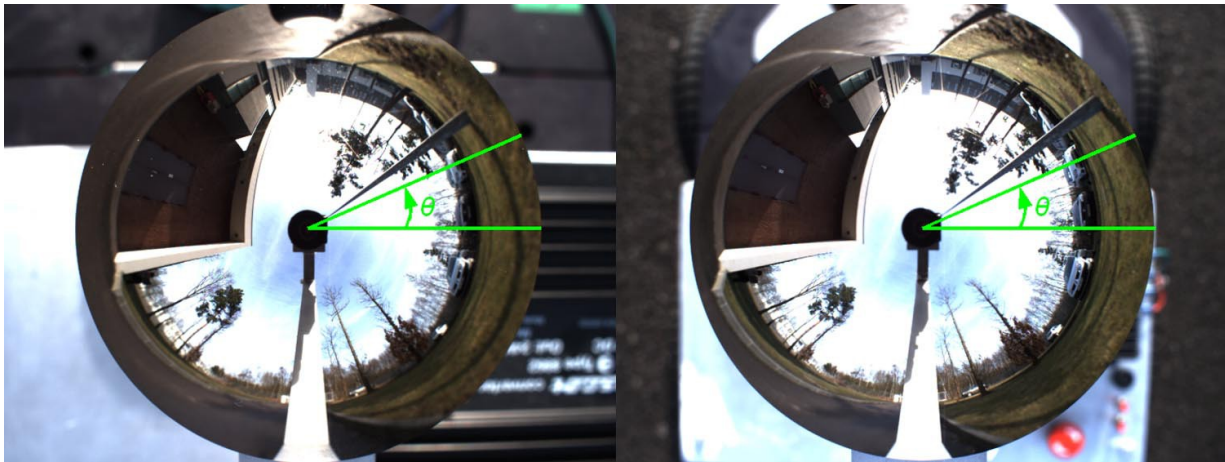


Figure 85: Extraction des rayons dans les images haute et basse.

Pour chaque valeur de  $\theta$ , les coordonnées des  $m$  pixels du rayon de l'image basse et des  $n$  pixels du rayon de l'image haute sont déterminées par l'algorithme de Bresenham [Bresenham65a].

### Construction de la matrice de recherche

Le principe de la programmation dynamique est de trouver un chemin de coût optimal (minimal ou maximal en fonction de la mesure de similarité utilisée) dans une grille appelée matrice de recherche (cf figure 86). Un nœud de la grille est une paire de candidats qui peuvent avoir une correspondance, et le chemin optimal est composé d'une séquence de nœuds.

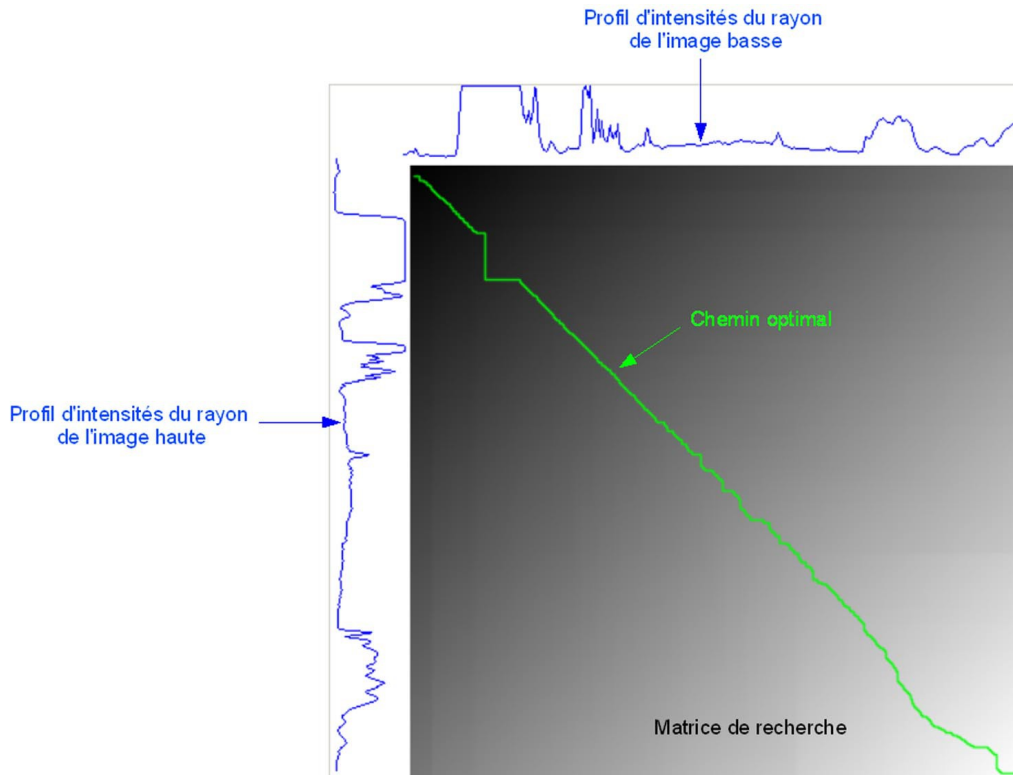


Figure 86: Exemple de matrice de recherche et de chemin optimal.

La première étape consiste donc à déterminer les coefficients de cette grille. Ceux-ci sont calculés du premier élément  $A[0,0]$  vers le dernier élément  $A[m,n]$ . Le coût du chemin étant défini comme la somme des coûts des nœuds du chemin, l'élément  $A[i,j]$  est calculé de la façon suivante :

- le maximum des trois coefficients précédents est recherché :

$$maximum = \max( A[i-1, j], A[i, j-1], A[i-1, j-1] )$$

- à ce maximum est ajouté le coût local qui correspond au score de corrélation de la combinaison de points  $(i, j)$  :

$$A[i, j] = maximum + Correlation( PixelBas( i ), PixelHaut( j ) )$$

Nous utilisons les conclusions que nous avons tirées de la partie traitant des méthodes locales pour améliorer nos résultats. En effet, les images sont converties dans l'espace de couleurs  $H_1 H_2 H_3$ . De plus, la mesure de corrélation utilisée donne de bons résultats, puisqu'il s'agit d'une corrélation croisée normalisée (NCC, *Normalized Cross Correlation*) alors qu'une autre mesure plus rapide aurait pu être choisie mais au détriment des résultats. Enfin, le score de corrélation est calculé sur une fenêtre autour du pixel, et non pas sur le pixel uniquement, nous utilisons dans cet exemple une fenêtre de taille  $3 \times 3$  pour limiter le temps de calcul.

Le remplissage des coefficients de la matrice de recherche peut donc se résumer par le



pseudo-code suivant :

```

A[0,0]=0
For  $i=0$  To  $i=m$  with step of 1
  For  $j=0$  To  $j=n$  with step of 1
     $maximum = \max(A[i-1,j], A[i,j-1], A[i-1,j-1])$ 
     $A[i,j] = maximum + Correlation(PixelBas(i),$ 
       $PixelHaut(j))$ 
  End of for-loop on  $j$ 
End of for-loop on  $i$ 

```

*Figure 87: Pseudo-code du calcul des coefficients de la matrice de recherche.*

### Recherche du chemin optimal

La recherche du chemin optimal est l'étape clé de la programmation dynamique car c'est à ce niveau qu'une partie des contraintes globales énoncées ci-dessus (cf 3.2.2.1) sont imposées.

La recherche du chemin optimal débute par le dernier élément de la matrice de recherche :  $A[m,n]$ . Pour chaque nœud du chemin  $(i,j)$ , nous cherchons à déterminer successivement parmi ses voisins l'élément qui permettra d'atteindre un coût total maximal, le coût du chemin étant la somme des coûts des nœuds.

La contrainte d'**ordre** est imposée en forçant le chemin à être monotone. Ainsi, la recherche de l'élément successif au nœud  $(i,j)$  se fait parmi les trois voisins  $(i-1,j)$ ,  $(i,j-1)$  et  $(i-1,j-1)$ , ce qui interdit tout retour en arrière. L'ordre des primitives homologues sera donc respecté le long des deux lignes épipolaires.

La recherche du chemin optimal peut donc être résumée par le pseudo-code de la figure 88 et un exemple de chemin optimal est donné sur la figure 86.

```

i = m
j = n
While (i > 0 AND j > 0)
    Up = A[i, j-1]
    Left = A[i-1, j]
    UpLeft = A[i-1, j-1]
    maximum = max(Up, Left, UpLeft)
    Case maximum of
        Up : j = j-1
        Left : i = i-1
        UpLeft : i = i-1 , j = j-1
    End of case
End of while-loop

```

Figure 88: Pseudo-code de la recherche du chemin optimal.

### Calcul de la disparité

La disparité est obtenue en faisant la différence entre  $i$  et  $j$  pour chaque nœud du chemin. Cependant, cette disparité ne peut pas être valide pour chaque nœud car les transitions horizontales ou verticales du chemin reviennent à associer à un pixel de l'image basse plusieurs pixels de l'image haute (et *vice versa*), ce qui est en contradiction avec la contrainte d'unicité. Pour imposer la contrainte d'**unicité**, seules les mises en correspondance obtenues lors de transitions diagonales du chemin sont considérées comme valides.

La contrainte de **continuité de la disparité** est également imposée en rejetant les appariements qui introduisent un fort gradient de disparité, c'est-à-dire de longues transitions horizontales ou verticales.

La figure 89 est un exemple de chemin optimal obtenu où les mises en correspondance valides sont représentées en vert et où les mises en correspondance non valides sont représentées en rouge. Les transitions horizontales ou verticales du chemin optimal, c'est-à-dire les mises en correspondance non valides, apparaissent lorsqu'il y a des occultations sur l'une ou l'autre des images.

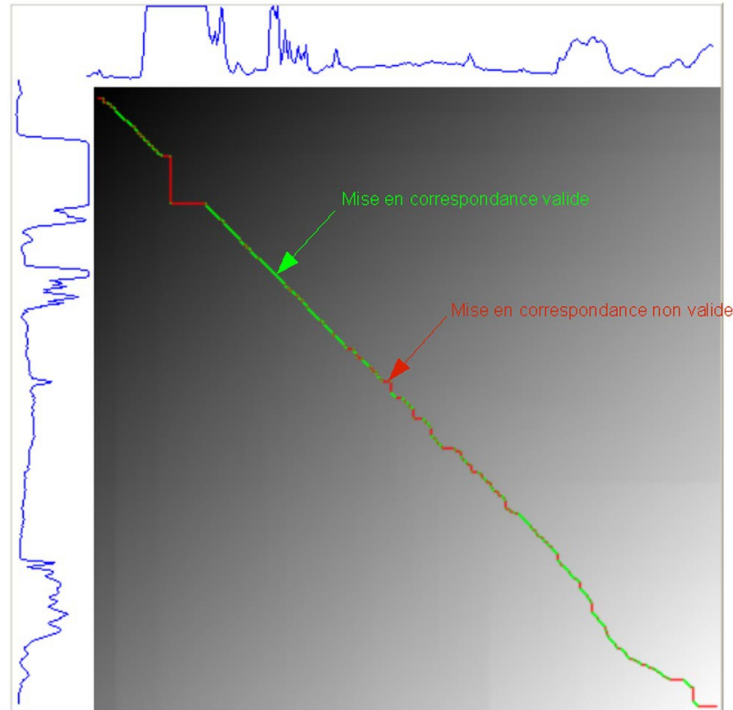


Figure 89: Chemin optimal avec mise en correspondance valides (en vert) et non valides (en rouge).

### Amélioration du temps de calcul

Le principal inconvénient de la programmation dynamique est le temps de calcul nécessaire pour remplir les coefficients de la matrice de recherche. En effet, pour chaque couple possible, il est nécessaire de calculer un score de corrélation qui est en général assez complexe. Le nombre de coefficients à calculer peut cependant être limité en imposant une contrainte supplémentaire connue sous le nom de **limite de la disparité**. Cette contrainte exprime le fait que la disparité est comprise entre deux limites, la scène à reconstruire ayant une profondeur bornée.

Avec cette contrainte, les coefficients de la matrice de recherche ne sont calculés que pour une bande autour de sa diagonale comme l'illustre la figure 90.

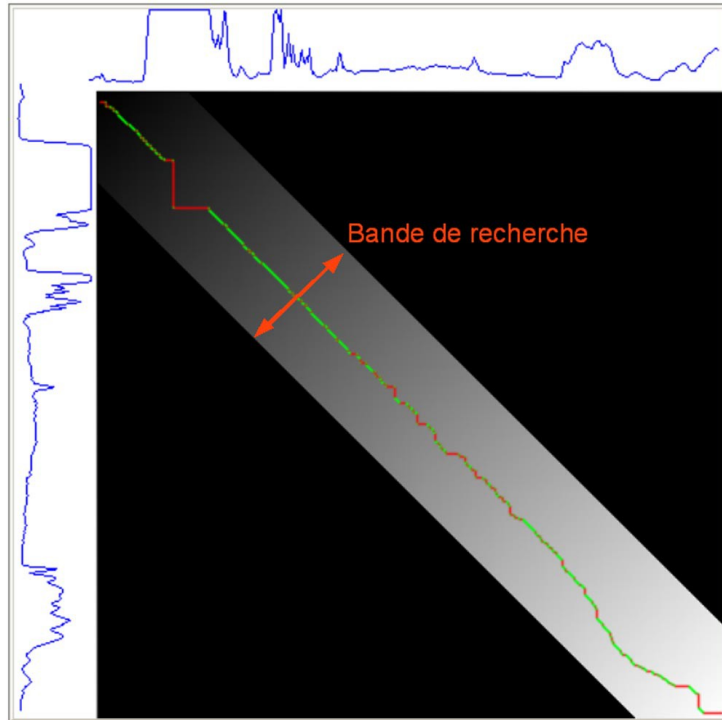


Figure 90: Structure de la matrice de recherche avec la contrainte de limite de la disparité.

Le principal intérêt de cette méthode est de diminuer considérablement le nombre de calculs de scores de corrélation qui est l'étape qui demande le plus de temps de traitement en programmation dynamique. Le temps global est par conséquent fortement diminué comme l'illustre la figure 91. Sur cette figure, les temps de traitement sont évalués en pourcentage par rapport au temps nécessaire lorsque la contrainte de limite de la disparité n'est pas imposée.

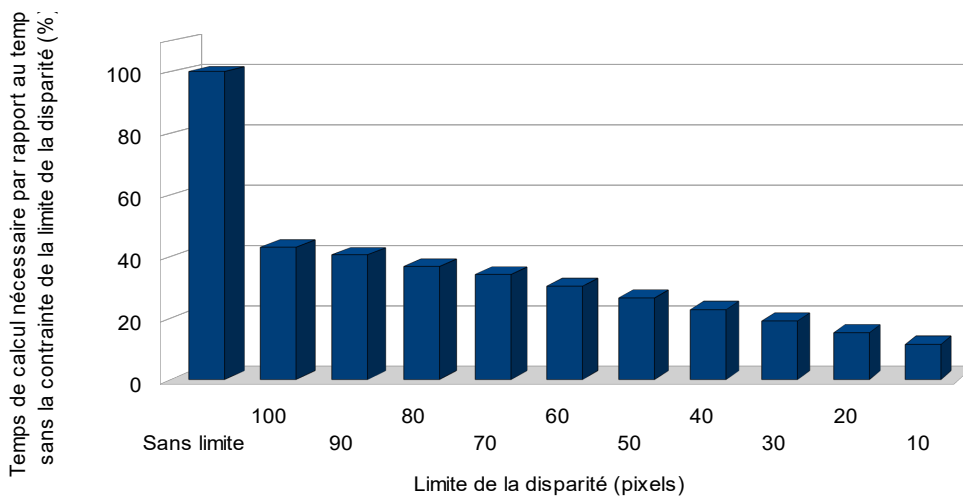


Figure 91: Évolution du temps de calcul par rapport à la limite de la disparité.

### 3.2.2.3. Résultats expérimentaux

Notre méthode a été évaluée sur la paire d'images présentée sur la figure 92. L'environnement à reconstruire est très diversifié puisqu'il présente à la fois des éléments

naturels (arbres, pelouse) et des éléments très structurés comme le bâtiment.

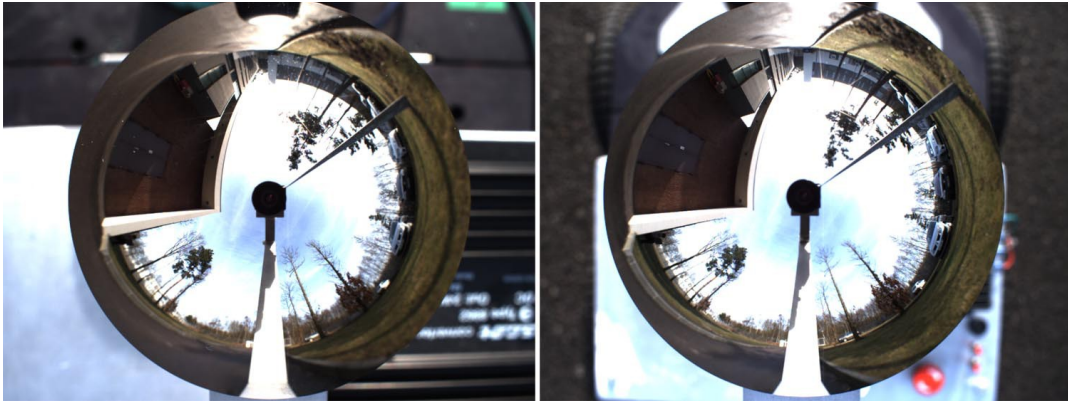


Figure 92: Paire d'images stéréoscopiques utilisées pour l'évaluation de la programmation dynamique.

Nous avons évalué l'apport de la programmation dynamique en comparant la carte de disparité obtenue par les méthodes locales (93 a) et en imposant les contraintes globales (93 b) par la méthode décrite précédemment.

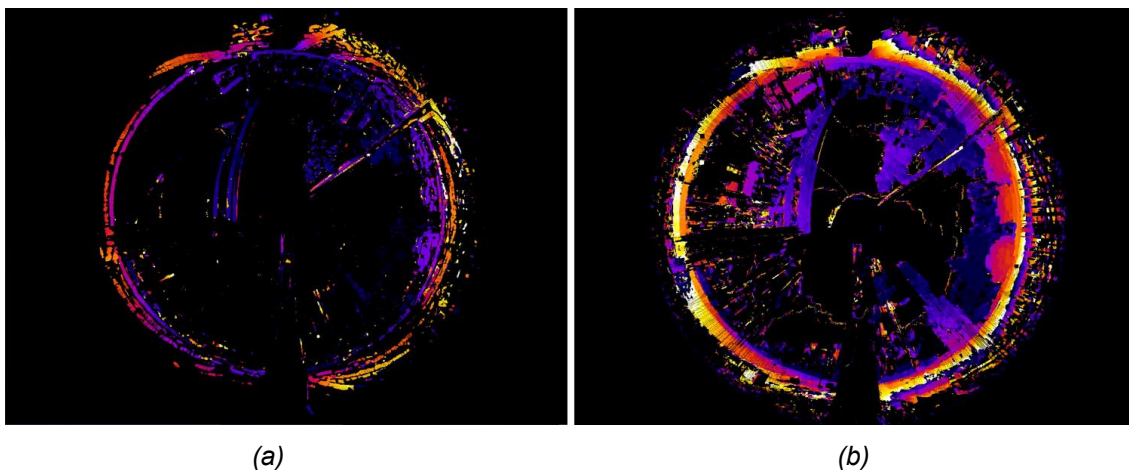


Figure 93: Cartes de disparité obtenues.  
(a) Méthode locale - (b) Méthode globale.

La carte de disparité obtenue par la programmation dynamique est bien plus complète que celle fournie par les méthodes locales. Par exemple, les arbres mis en valeur en rouge sur la figure 94 n'étaient pas reconstruits avec les méthodes locales. Il en est de même pour la structure du bâtiment (en vert sur la figure) pourtant faiblement texturée. La mise en correspondance des pixels de ce bâtiment a été possible grâce à la programmation dynamique qui permet l'appariement de régions faiblement texturées délimitées par des contours remarquables.

Certains objets qui étaient déjà visibles sur la carte de disparité obtenue avec les méthodes locales, par exemple les voitures mises en valeur en bleu sur la figure 94, présentent des surfaces beaucoup plus complètes et sans artefact grâce à la contrainte de continuité de la disparité.



Figure 94: Mise en valeur des objets de la carte de disparité.

Avec l'utilisation de la méthode globale, la reconstruction du bâtiment (en vert), des arbres (en rouge) et des voitures (en bleu) est plus complète.

### 3.2.3. Triangulation de points

#### 3.2.3.1. Méthode du point milieu

La triangulation est l'opération qui consiste à calculer, à partir de deux pixels appariés, les coordonnées du point 3D correspondant. Il s'agit de l'aboutissement du processus de reconstruction 3D, dans le sens où c'est la triangulation qui génère le modèle 3D.

Dans le cas parfait, le principe de la triangulation est très simple : le point 3D correspondant à deux pixels appariés est l'intersection des deux rayons lumineux associés à ces pixels.

Dans la pratique, de petites erreurs pouvant intervenir à tous les niveaux du processus (acquisition, calibration, appariement, *etc*) conduisent à deux rayons qui ne s'intersectent pas parfaitement, comme l'illustre la figure 95.

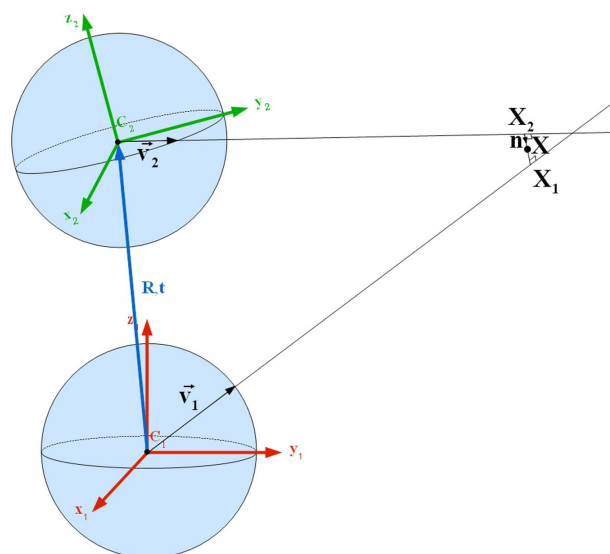


Figure 95: Triangulation par la méthode du point milieu.

Le principe de la méthode du point milieu est de calculer le barycentre des points  $\mathbf{X}_1$  et  $\mathbf{X}_2$  appartenant aux deux rayons telle que  $\|\overrightarrow{\mathbf{X}_1\mathbf{X}_2}\|$  soit minimale.

La distance minimale  $\|\overrightarrow{\mathbf{X}_1\mathbf{X}_2}\|$  correspond à la longueur du segment de la perpendiculaire commune aux deux rayons. Ce segment de droite a, par conséquent, pour vecteur directeur :

$$\mathbf{n} = \frac{\mathbf{v}_1 \wedge \mathbf{v}_2}{\|\mathbf{v}_1 \wedge \mathbf{v}_2\|} \quad (39)$$

Les coordonnées des points  $\mathbf{X}_1$  et  $\mathbf{X}_2$  peuvent être exprimées en utilisant les équations paramétriques des deux droites :

$$\begin{cases} \mathbf{X}_1 = \mathbf{C}_1 + \alpha \cdot \mathbf{v}_1 \\ \mathbf{X}_2 = \mathbf{C}_2 + \beta \cdot \mathbf{v}_2 \end{cases} \quad (40)$$

Avec ces notations, le vecteur  $\overrightarrow{\mathbf{X}_2\mathbf{X}_1}$  peut s'écrire :

$$\overrightarrow{\mathbf{X}_2\mathbf{X}_1} = \lambda \mathbf{n} \quad (41)$$

ou encore  $\mathbf{C}_1 - \mathbf{C}_2 + \alpha \mathbf{v}_1 - \beta \mathbf{v}_2 = \lambda \mathbf{n}$ , qui permet de former le système d'équations :

$$\begin{cases} -\lambda n_x + \alpha v_{1x} - \beta v_{2x} + C_{1x} - C_{2x} = 0 \\ -\lambda n_y + \alpha v_{1y} - \beta v_{2y} + C_{1y} - C_{2y} = 0 \\ -\lambda n_z + \alpha v_{1z} - \beta v_{2z} + C_{1z} - C_{2z} = 0 \end{cases} \quad (42)$$

qui est équivalent, sous forme matricielle, à :

$$\begin{bmatrix} -n_x & v_{1x} & -v_{2x} \\ -n_y & v_{1y} & -v_{2y} \\ -n_z & v_{1z} & -v_{2z} \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} C_{1x} - C_{2x} \\ C_{1y} - C_{2y} \\ C_{1z} - C_{2z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (43)$$

La résolution de ce système permet d'obtenir les coordonnées des points  $\mathbf{X}_1$  et  $\mathbf{X}_2$  en réinjectant les valeurs de  $\alpha$  et  $\beta$  dans le système (40). Le point 3D résultant de la triangulation correspond au barycentre de ces deux points  $\mathbf{X}_1$  et  $\mathbf{X}_2$ .

### 3.2.3.2. Résultats expérimentaux

Nous avons évalué la précision de la reconstruction 3D en estimant la position de points 3D de cinq mires disposées sur trois murs d'une pièce (cf figure 96).

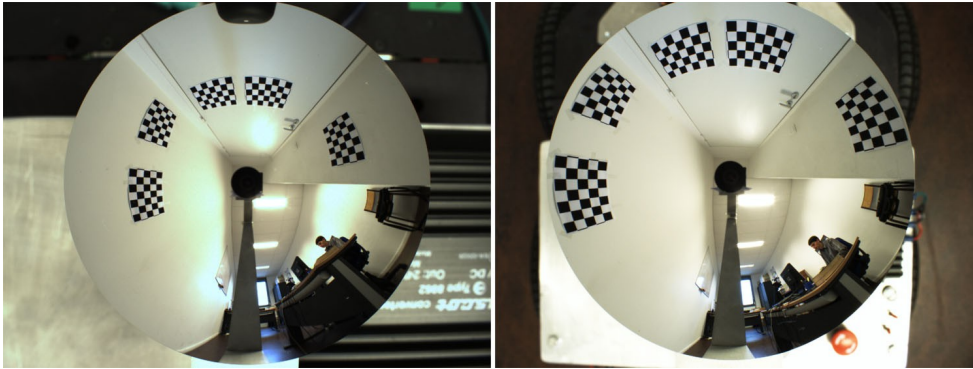


Figure 96: Paire d'images stéréoscopiques utilisée pour l'évaluation de la triangulation.

Pour chaque mire, les quatre coins ont été sélectionnés manuellement sur les deux images et un détecteur de coins automatique a été utilisé pour extraire tous les coins de la mires. La position 3D de ces points a été évaluée par triangulation et est illustrée sur la figure 97. La position de ces points est comparée à la vérité terrain obtenue par télémétrie laser. Une régression linéaire a été appliquée sur les données brutes du laser (en bleu) pour obtenir la position des murs (lignes rouges).

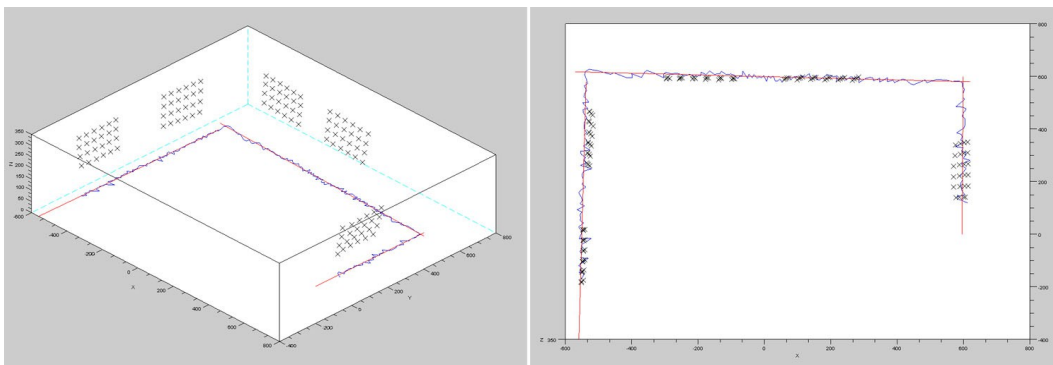


Figure 97: Position des points des cinq mires (croix) comparée à la vérité terrain (lignes bleues et rouges).

Pour chaque mire, l'erreur moyenne et l'écart type de la position de ses points ont été calculés et sont résumés ci-dessous :

Grille n°	1	2	3	4	5
Erreur moyenne (mm)	5,843	15,529	13,182	3,794	12,872
Ecart type (mm)	2,616	4,50	2,420	2,504	7,485

Tableau 6: Résultats de la triangulation des points des mires.

L'estimation de la position des points de la mire est très précise puisque ramenée à la distance des points (situés à environ 1 mètre), l'erreur est de l'ordre de 1%. Cette erreur augmente cependant en fonction de la distance à cause de la résolution spatiale de notre capteur.



### **3.3. Reconstruction 3D de droites**

La reconstruction 3D dense s'applique bien aux environnements naturels car il existe dans ce type d'environnement une forte texture et très peu de motifs répétitifs. Or, dans les environnements créés par l'homme, le manque de texture conduira les algorithmes présentés précédemment à l'échec, en particulier pour les façades de bâtiments ou les scènes intérieures.

Pour remédier à ce problème, et pour avoir un modèle 3D plus représentatif que le nuage de points, il est particulièrement intéressant de reconstruire les environnements créés par l'homme par des primitives plus élaborées, telles que les droites et les plans. En effet, dans un tel environnement, la plupart des objets peut être reconstruit sous forme de polyèdres avec des contours horizontaux et verticaux (murs, portes, *etc*).

Nous présentons dans cette section un nouvel algorithme de détection de droites dans les images omnidirectionnelles, ainsi que la méthode de triangulation permettant de calculer la position d'une droite dans l'espace 3D connaissant ses projections dans les images.

#### **3.3.1. Détection de droites dans les images omnidirectionnelles**

La détection de droites dans les images omnidirectionnelles est beaucoup moins aisée que dans les images perspectives car la projection des droites sur le plan image ne donne plus des droites mais des coniques. Le problème classique de détection de lignes dans les images perspectives devient donc un problème complexe de détection de coniques dans les images omnidirectionnelles.

Il existe quelques cas particuliers qui permettent de simplifier la détection des lignes dans les images omnidirectionnelles. [Yagi91a] et [Delahoche97a] se sont intéressés à la détection de droites projetées radialement sur un miroir conique. Cela impose que le capteur soit en position verticale et qu'il y ait suffisamment de lignes verticales dans l'environnement. [Fiala02a] a proposé une méthode pour extraire des segments horizontaux et verticaux, mais pas des droites quelconques. Cette limitation est due à l'utilisation de capteurs non-centraux qui rend la détection de lignes quelconques extrêmement complexe.

Comme nous disposons d'un capteur central, nous exploitons cette propriété pour faciliter la détection de lignes. Pour cette famille de capteurs, les méthodes d'estimation de lignes peuvent être classées en deux principales catégories. La première catégorie cherche à détecter des coniques dans l'image et à en estimer les cinq paramètres de sorte que la conique s'ajuste aux données extraites [Zhang97a]. Le temps de calcul est généralement important et ces algorithmes ne sont pas efficaces quand seules de petites portions de coniques sont visibles. De plus, ces méthodes ne sont souvent applicables qu'au cas particulier du capteur paracatadioptrique,

association d'un miroir parabolöide et d'une caméra orthographique car les projections des lignes deviennent alors des arcs de cercles [Barreto03b] [Barreto06b] [Vandeportaele06a].

La seconde catégorie consiste en une détection des grands cercles sur la sphère d'équivalence. Il a en effet été démontré dans [Geyer01a] que la projection d'une ligne 3D sur la sphère est un grand cercle  $C$  comme l'illustre la figure 98. Ces méthodes ne sont applicables qu'aux capteurs calibrés car il faut pouvoir calculer la projection des pixels sur la sphère d'équivalence.

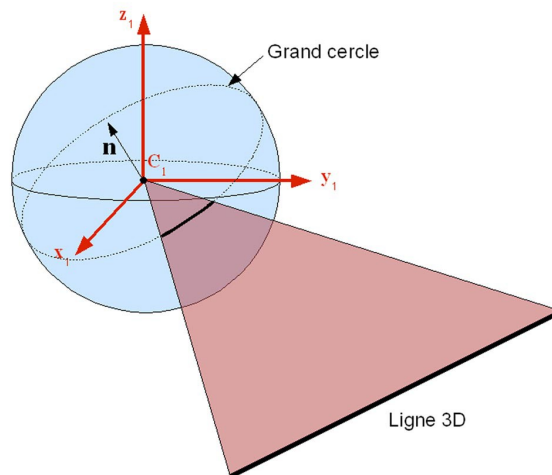


Figure 98: Projection d'une droite sur la sphère d'équivalence.

Le principal avantage de la détection des grands cercles sur la sphère d'équivalence est que l'algorithme sera utilisable pour tous les capteurs centraux puisqu'il s'agit d'un modèle unifié.

La détection des grand cercles est en général réalisée par une transformée de Hough [Hough62a], la différence entre les différents algorithmes est l'espace dans lequel les traitements sont effectués. Dans [Vasseur04a], les grands cercles sur la sphère sont définis par leur normale caractérisée par les angles d'élévation et d'azimut. Un espace de Hough fondé sur ces angles est alors créé et la détection des maxima dans l'espace de Hough permet d'obtenir les paramètres des droites. Il existe d'autres espaces de Hough pour caractériser les droites, il est par exemple possible d'utiliser la normale directement [Mei07b] ou les coordonnées cubiques [Yamazawa00a].

La transformée de Hough introduit cependant quelques problèmes comme le choix du meilleur pas d'échantillonnage de l'espace de Hough qui devrait de plus tenir compte de la résolution non uniforme des capteurs, et la détection des maxima de l'espace de Hough qui est une étape délicate.

Nous proposons donc une nouvelle méthode de détection et d'estimation de droites dans les images omnidirectionnelles, valide pour tous les capteurs centraux, et fondée sur l'algorithme

d'estimation robuste RANSAC.

La première phase de notre méthode consiste à extraire les contours présents dans l'image omnidirectionnelle (figure 99 a) en appliquant l'algorithme de Canny. De très nombreux contours sont détectés parmi lesquels certains n'appartiennent pas réellement à des contours physiques comme le montre la figure 99 b.

Des chaînes de pixels connexes sont ensuite créées avec les contours extraits précédemment. Pour éviter la détection d'une multitude de petites lignes, les chaînes trop courtes sont rejetées. Le résultat de cette étape est présenté sur la figure 99 c. Sur cette figure, les chaînes de pixels connexes sont représentées par différentes couleurs et les chaînes inférieures à 50 pixels ont été rejetées.

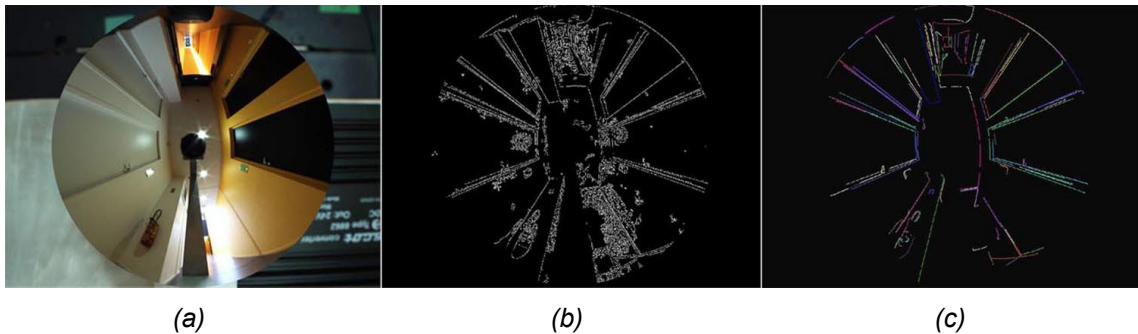


Figure 99: (a) Image omnidirectionnelle - (b) Contours détectés - (c) Chaînes de pixels connexes.

Chaque chaîne ainsi créée est une droite omnidirectionnelle potentielle. Il faut donc statuer pour déterminer s'il s'agit bien d'une droite et, dans ce cas, estimer ses paramètres. Les droites omnidirectionnelles seront caractérisées par la normale du plan qui contient la droite et qui passe par le centre de la sphère d'équivalence (*cf* figure 100).

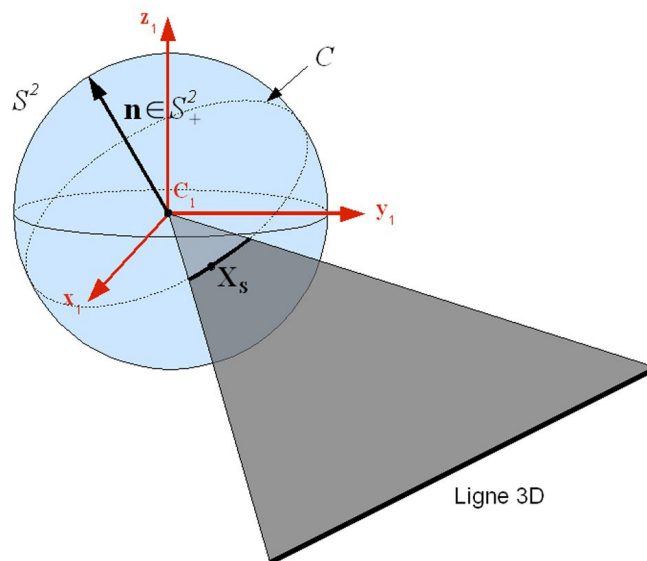


Figure 100: Caractérisation d'une droite omnidirectionnelle par une normale.

Nous proposons de travailler directement sur la sphère unitaire  $S^2 \in \mathbb{R}^3$ . Nous caractériserons une droite par la normale  $\mathbf{n} \in S_+^2$  du plan formé par la droite et le centre de projection.

Tous les pixels des chaînes sont projetés sur la sphère d'équivalence. Cette étape, appelée *lifting*, nécessite la connaissance des paramètres intrinsèques du capteur et suppose donc que le capteur est calibré. Tous les traitements suivants sont réalisés avec les points de la sphère.

Pour estimer les paramètres de la droite qui correspond le mieux aux points de la chaîne, une méthode d'estimation robuste, appelée RANSAC (RANDOM SAMPLE CONSENSUS [Fischler81a]), est utilisée. Son pseudo-code est donné sur la figure 101.

Pour chaque chaîne, deux points  $\mathbf{X}_s^1$  et  $\mathbf{X}_s^2$  sont tirés aléatoirement et la normale du plan passant par ces deux points et par le centre de la sphère est calculée :

$$\mathbf{n} = \lambda \frac{\mathbf{X}_s^1 \times \mathbf{X}_s^2}{\|\mathbf{X}_s^1 \times \mathbf{X}_s^2\|} \quad \text{où } \lambda \in \{-1, 1\} \text{ permet de s'assurer que } \mathbf{n} \in S_+^2. \quad (44)$$

Les autres points de la chaîne sont ensuite testés pour statuer sur leur appartenance au grand cercle  $C$  donné par :

$$C = \{ \mathbf{X}_s \mid \mathbf{n}^T \mathbf{X}_s = 0, \quad \|\mathbf{X}_s\| = 1 \} \quad (45)$$

Dans la pratique, des petites erreurs dues au bruit, à la discrétisation de l'image ou à des erreurs de calibrage conduisent à fixer un seuil d'erreur pour statuer sur l'appartenance ou non du point  $\mathbf{X}_s$  au grand cercle  $C$ . Tous les points appartenant au grand cercle  $C$  défini par  $\mathbf{n}$  sont alors stockés en mémoire.

Si le nombre de points appartenant au cercle  $C$  est supérieur à la longueur minimale autorisée pour définir une droite, alors la chaîne de pixels correspond bien à une droite dont la normale est ré-estimée grâce à tous les points appartenant au grand cercle.

Il est alors possible de définir des segments plutôt que des droites en prenant le premier et le dernier pixel de la chaîne appartenant au grand cercle.

```

k = 0
Tant que k < MAX_ITERATIONS
  maybe_inliers = 2 points sélectionnés aléatoirement
  maybe_droite = n estimée avec ces 2 points
  consensus_set = maybe_inliers
  Pour chaque point pas dans maybe_inliers
    Si le point respecte le modèle avec une erreur
      plus petite que MAX_ERROR alors ajouter le
      point à consensus_set
  Si le nombre de points dans consensus_set > seuil alors il
  s'agit bien d'une droite.
  Ré-estimation de n avec tous les points dans consensus_set

```

Figure 101: Pseudo-code pour l'estimation des droites omnidirectionnelles par la méthode du RANSAC.

Toutes ces étapes permettent de détecter les segments de droites omnidirectionnelles comme l'illustre la figure suivante :

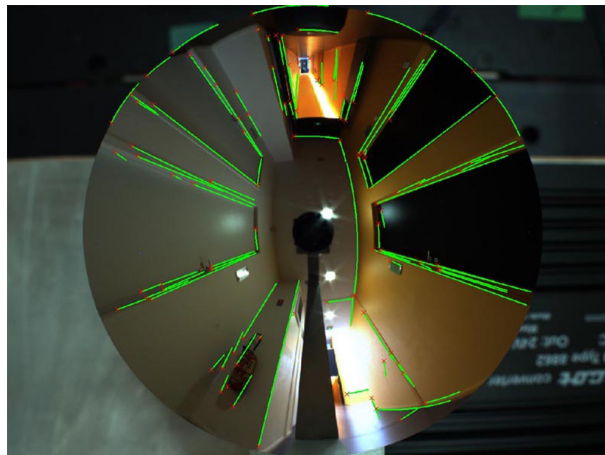


Figure 102: Segments de droites omnidirectionnelles détectés.

### 3.3.2. Triangulation de droites

La détection des droites omnidirectionnelles évoquée ci-dessus permet d'aboutir à la paramétrisation de la droite dans le repère local du capteur, la droite étant représentée par la normale au plan contenant la droite et le centre du capteur. Tous les paramètres sont dans un premier temps exprimés dans le même repère de référence : celui du capteur bas. Ce changement de repère dépend des paramètres de la pose relative des deux capteurs obtenus par calibrage.

Soient  $\mathbf{C}_1 = [0 \ 0 \ 0]^T$  et  $\mathbf{C}_2 = [c_{2x} \ c_{2y} \ c_{2z}]^T$  les centres des capteurs,  $\mathbf{n}_1 = [n_{1x} \ n_{1y} \ n_{1z}]^T$

et  $\mathbf{n}_2 = [n_{2x} \ n_{2y} \ n_{2z}]^T$  les normales caractérisant la droite dans le repère  $(C_1, x_1, y_1, z_1)$  (cf figure 103). L'objectif est de déterminer l'équation paramétrique de la ligne 3D  $L$  correspondante :  $L = \mathbf{A} + \lambda \mathbf{u}$ , où  $\mathbf{A}$  est un point de la droite et  $\mathbf{u}$  le vecteur directeur de celle-ci.

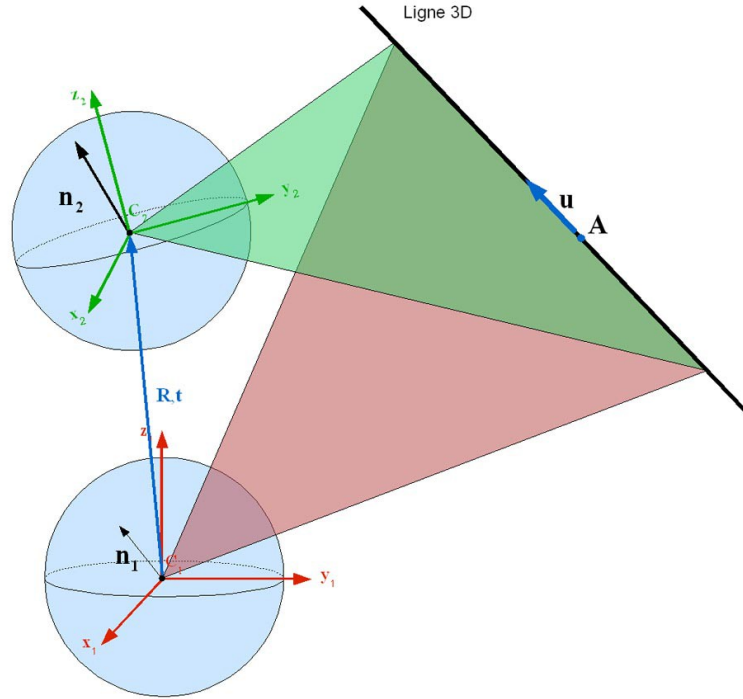


Figure 103: Triangulation d'une droite.

La droite vérifie l'équation des deux plans, donc :

$$\begin{cases} n_{1x} \cdot x + n_{1y} \cdot y + n_{1z} \cdot z & = & 0 \\ n_{2x} \cdot x + n_{2y} \cdot y + n_{2z} \cdot z + d & = & 0 \end{cases} \text{ avec } d = -n_{2x} \cdot c_{2x} - n_{2y} \cdot c_{2y} - n_{2z} \cdot c_{2z}. \quad (46)$$

En  $z=0$ , la ligne  $L$  passe donc par le point  $\mathbf{A}$  de coordonnées

$$\mathbf{A} = \begin{bmatrix} \frac{-n_{1y} \cdot n_{1x} \cdot d}{n_{1x} \cdot (n_{2x} n_{1y} - n_{1x} n_{2y})} & \frac{n_{1x} \cdot d}{(n_{2x} n_{1y} - n_{1x} n_{2y})} & 0 \end{bmatrix}^T.$$

Le vecteur directeur  $\mathbf{u}$  de la droite est obtenu directement en calculant le produit vectoriel des normales des deux plans définissant cette droite :

$$\mathbf{u} = \frac{\mathbf{n}_1 \times \mathbf{n}_2}{\|\mathbf{n}_1 \times \mathbf{n}_2\|} \quad (47)$$

La triangulation de droites a été évaluée sur la paire d'images suivante :

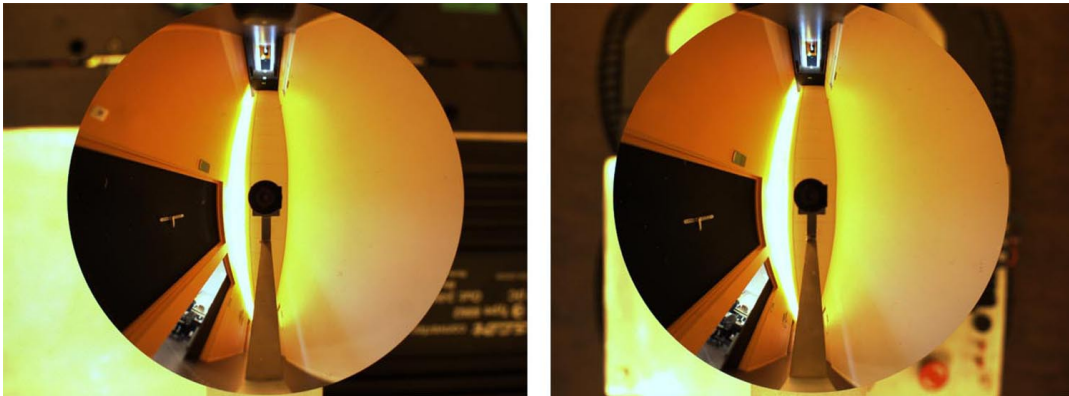


Figure 104: Images omnidirectionnelles basse et haute utilisées pour la triangulation de droites.

La droite correspondant au bas du mur a été triangulée et comparée à la vérité terrain obtenue par télémétrie laser. La figure ci-dessous met en évidence que la position estimée de la droite (en vert) correspond bien à la vérité terrain obtenue par le laser (en jaune).

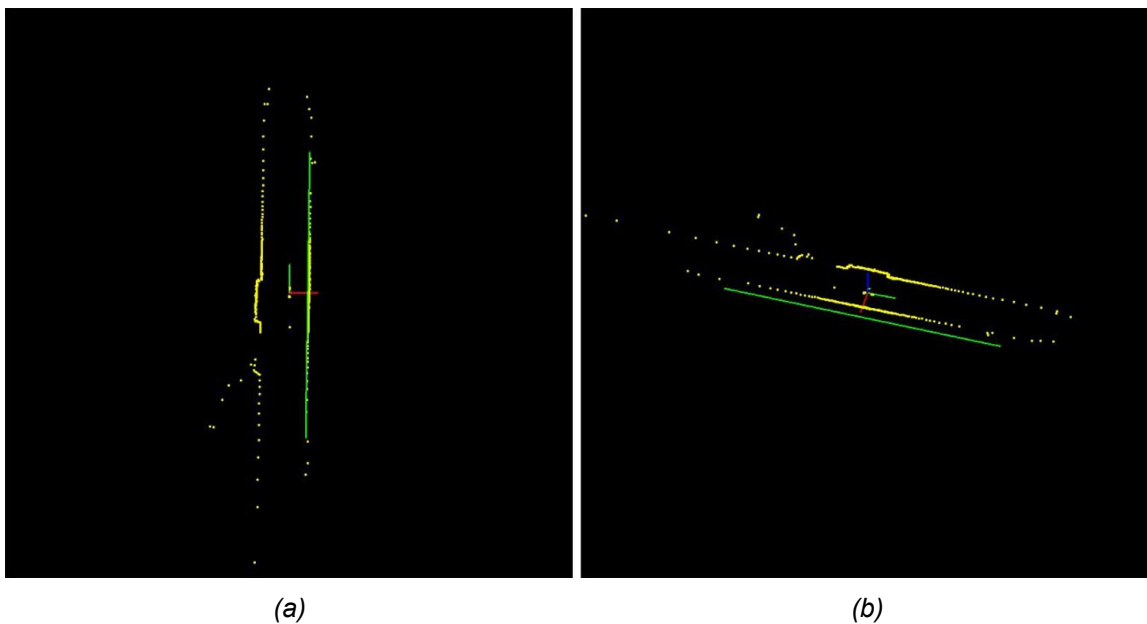


Figure 105: Reconstruction statique d'un couloir.  
La droite triangulée est représentée en vert et la vérité terrain en jaune.  
(a) Vue de dessus – (b) Vue de côté (b)

### 3.4. Conclusion et perspectives

Le chapitre 3 a été consacré aux méthodes permettant la reconstruction 3D de l'environnement sans déplacement du capteur. Les méthodes locales, couramment utilisées en stéréovision, ont vite montré leurs limites sur les images omnidirectionnelles en niveaux de gris. De nombreuses améliorations ont été apportées par rapport aux algorithmes de corrélation classique pour obtenir des résultats moins bruités : utilisation de l'espace de couleurs  $H_1 H_2 H_3$ ,

méthodes de rejet des appariements, *etc.* Ces méthodes ont l'avantage de limiter le nombre de mauvais appariements mais la carte de disparité produite est alors moins complète.

Nous avons ensuite proposé, pour obtenir une carte de disparité la plus complète possible, un algorithme de programmation dynamique adapté aux images omnidirectionnelles et exploitant les résultats soulignés par les méthodes locales. Cet algorithme a permis d'améliorer grandement la densité de la carte de disparité, tout en garantissant un taux de mauvais appariements relativement faible.

La reconstruction 3D dense pourrait encore être améliorée en utilisant une fenêtre de corrélation adaptée à la géométrie du capteur. En effet, la fenêtre de corrélation « carrée » utilisée sur les images classiques n'a plus de sens sur des images omnidirectionnelles. Une approche intéressante dédiée au filtrage et à la détection de contours, consistant à travailler directement sur la sphère équivalente, a été proposée par Bigot-Marchand [Bigot-Marchand08a] et pourrait être adaptée au problème de la corrélation.

Dans des environnements peu texturés tels que les couloirs d'un bâtiment, les méthodes fondées sur la corrélation ne fonctionnent plus. Une alternative à ce problème consiste à travailler avec des primitives plus élaborées telles que les droites. La détection de droites est un problème relativement simple avec une caméra classique, mais beaucoup plus complexe avec des images omnidirectionnelles. Nous avons donc proposé un nouvel algorithme permettant de détecter tout type de droites dans les images omnidirectionnelles. Après triangulation, la position de la droite dans l'espace fournit des informations intéressantes telles que la position d'un mur avec une bonne précision.

Dans les environnements structurés, une reconstruction 3D plus complète pourrait être obtenue en imposant des contraintes supplémentaires et en travaillant sur la reconstruction 3D de plans. Cela permettrait par la suite d'utiliser des méthodes de plaquage de textures pour fournir un modèle photo-réaliste.



## Chapitre 4

# Reconstruction 3D avec déplacement

### Sommaire

---

<b>4.1.Introduction.....</b>	<b>133</b>
<b>4.2.Initialisation du déplacement et des points 3D.....</b>	<b>134</b>
4.2.1.Mise en correspondance des images.....	135
4.2.2.Estimation de la géométrie épipolaire.....	136
4.2.2.1.Géométrie épipolaire des capteurs centraux.....	136
4.2.2.2.Algorithme des huit points pour les capteurs centraux.....	138
4.2.2.3.Renforcement des contraintes de la matrice essentielle.....	139
4.2.2.4.Estimation robuste de la matrice essentielle.....	140
4.2.3.Détermination du déplacement.....	143
4.2.3.1.Décomposition de la matrice essentielle.....	143
4.2.3.2.Détermination du facteur d'échelle.....	145
<b>4.3.Ajustement de faisceaux.....</b>	<b>146</b>
4.3.1.Fonctionnelle à minimiser.....	146
4.3.2.Fonction de projection stéréoscopique.....	147
4.3.3.Structure de la jacobienne.....	148
<b>4.4.Résultats expérimentaux.....</b>	<b>149</b>
4.4.1.Phase d'initialisation.....	149
4.4.1.1.Mise en correspondance des images.....	149
4.4.1.2.Estimation de la géométrie épipolaire.....	151
4.4.1.3.Détermination du déplacement.....	151
4.4.2.Ajustement de faisceaux.....	152
4.4.2.1.Estimation des déplacements.....	152
4.4.2.2.Estimation de la trajectoire.....	153
4.4.2.3.Recalage des cartes locales.....	154
<b>4.5.Conclusion et perspectives.....</b>	<b>156</b>

## 4.1. Introduction

Le chapitre précédent a été consacré aux techniques permettant d'obtenir une reconstruction 3D statique, c'est-à-dire sans déplacement du capteur. Lorsque l'environnement à reconstruire est important, une telle reconstruction ne suffit plus. Deux solutions peuvent alors être envisagées : l'estimation du déplacement du capteur pour fusionner les différentes reconstructions 3D locales, ou la mise en place de techniques permettant d'obtenir une reconstruction 3D grâce au déplacement.

La reconstruction 3D d'une scène grâce au déplacement d'un capteur dans celle-ci fait intervenir deux problématiques : la **localisation** et la **cartographie**. Le problème de la localisation (*localization*) consiste à estimer la trajectoire du robot dans une carte connue *a priori* [Thrun00a]. La cartographie (*mapping*), dont une revue des algorithmes est présentée par Thrun [Thrun02a], a quant à elle pour but la création de la carte de l'environnement à partir de mesures obtenues par différents types de capteurs (sonars, lidars, caméras), connaissant la trajectoire du robot. Dans le cas de figure où ni la trajectoire du robot, ni la carte de l'environnement ne sont connues, les problèmes de localisation et de cartographie doivent être traités simultanément : il s'agit de la problématique du **SLAM** (*Simultaneous Localization And Mapping*).

Les techniques de *Structure From Motion* (SFM) sont très proches de la problématique du SLAM puisqu'elles ont pour but de déterminer la position de la caméra et d'estimer la structure géométrique de la scène [Pollefeys04a]. Les algorithmes de SFM ont cependant été conçus pour fonctionner hors-ligne et nécessitent généralement que toutes les images de la séquence soient traitées simultanément, ce qui n'est pas applicable à la robotique mobile. L'exploration d'un environnement inconnu par le robot nécessite en effet des algorithmes fonctionnant en ligne pour construire la carte de l'environnement et s'y localiser pendant l'exploration du monde.

Le SLAM est l'un des sujets de recherche les plus actifs dans la communauté robotique car la résolution de ce problème permettrait de conférer au robot une réelle autonomie sans avoir à instrumenter son environnement ou à lui fournir une connaissance *a priori* du monde qui l'entoure. Des résultats intéressants dans ce domaine ont été obtenus ces dernières années, mais principalement avec l'utilisation de lidars (qui fournissent directement l'information de distance) et pour la construction de cartes 2D d'environnements intérieurs. Une revue de ces algorithmes est présentée dans les tutoriaux de Durrant-Whyte et de Bailey [Durrant-Whyte06a] [Bailey06a].

La première approche pour résoudre le problème du SLAM consiste à supposer que le déplacement du robot est connu (par odométrie ou par la loi de commande utilisée) même si

cette dernière est entachée d'erreurs. La position des amers<sup>12</sup> peut donc être **prédite** suite à ce déplacement. Le robot dispose également de capteurs, généralement des lidars, lui permettant de faire des **observations** de son environnement. Ces observations sont alors utilisées pour **mettre à jour** le modèle contenant les coordonnées des amers et les positions du robot. Ces trois étapes (prédiction/observation/mise à jour) sont effectuées récursivement grâce au filtre de Kalman [Kalman60a] ou à l'un de ses dérivés.

La seconde approche consiste à optimiser la géométrie de la scène et les positions du capteur par la méthode appelée « ajustement de faisceaux » (*Bundle Adjustment*). L'ajustement de faisceaux donne de meilleurs résultats que le filtre de Kalman [Mouragnon09a], mais nécessite en contre-partie un temps de calcul plus élevé. Une synthèse des algorithmes d'ajustement de faisceaux a été réalisée par Triggs [Triggs00b]. Dans la majorité des applications, cet algorithme est utilisé hors ligne pour affiner les résultats et obtenir un modèle 3D très précis, mais il est également possible de l'appliquer itérativement. Bien que l'ajustement de faisceaux soit un algorithme couramment utilisé avec des caméras classiques, il n'existe que très peu de travaux portant sur son adaptation aux capteurs omnidirectionnels. Les principaux travaux dans ce domaine sont ceux de Lhuillier [Lhuillier05a] et Mouragnon [Mouragnon09a], qui proposent de retrouver la structure de la scène et les mouvements du capteur catadioptrique par un ajustement de faisceaux local, c'est-à-dire en ne prenant en compte que trois images à chaque itération, suivi par un ajustement de faisceaux global pour affiner les résultats. Leurs travaux ont mis en évidence la difficulté d'estimer le facteur d'échelle de la scène, bien que cela soit théoriquement possible avec un capteur non-central.

Nous proposons donc dans ce chapitre d'appliquer un algorithme d'ajustement de faisceaux à un capteur stéréoscopique omnidirectionnel préalablement calibré, afin de lever l'indétermination sur le facteur d'échelle. L'ajustement de faisceaux étant une technique fondée sur une minimisation non-linéaire d'un critère, la première section de ce chapitre est consacrée à l'obtention d'une estimation initiale la plus proche possible de la réalité. L'algorithme d'ajustement de faisceaux à proprement parler est ensuite décrit, puis des résultats expérimentaux sur des séquences d'images réelles sont présentés. Enfin, une conclusion clôt ce chapitre et des perspectives sur le SLAM sont présentées.

## **4.2. Initialisation du déplacement et des points 3D**

L'ajustement de faisceaux consiste à minimiser l'erreur de reprojection des points de la scène, en estimant simultanément la pose du capteur et les coordonnées de ces points 3D. La

---

<sup>12</sup> Les amers sont des points remarquables utilisés comme repères pour la navigation. Ce terme fait référence à la navigation maritime.

fonction de projection étant non-linéaire (cf équation 17), une première estimation des paramètres doit être obtenue pour assurer la convergence. Les algorithmes présentés dans cette section permettent d'obtenir cette première estimation, et peuvent de plus être utilisés pour connaître le mouvement du système (*ego-motion*) si le facteur d'échelle n'a pas besoin d'être déterminé.

### 4.2.1. Mise en correspondance des images

Pour déterminer les mouvements d'un système de vision, qu'il s'agisse d'une caméra perspective ou d'un capteur catadioptrique, il est nécessaire de mettre en correspondance les images issues de ces déplacements. Nos travaux portant sur l'estimation itérative des déplacements, la mise en correspondance est effectuée entre deux instants : l'instant présent  $k$  et l'instant précédent  $k-1$ .

La mise en correspondance consiste à localiser dans les images les projections d'un même point 3D de la scène. Cette étape est décisive puisque la précision de l'estimation des déplacements dépend de la qualité de cet appariement. Nous avons mis en évidence dans le chapitre précédent la difficulté d'apparier des points quelconques, notamment en l'absence de texture. Pour cette raison, il est préférable de travailler sur des points remarquables qui seront alors plus faciles à identifier dans les images.

De nombreux travaux utilisent le détecteur de Harris [Harris88a] pour détecter les amers [Bjorkman00a]. Ce détecteur a en effet de bonnes performances pour les changements de luminosité ou de point de vue, mais ce dernier est très sensible aux changements d'échelle. Dans le cas d'un déplacement assez important, l'échelle des objets est modifiée, ce qui peut aboutir à un échec de la détection ou de l'appariement des points d'intérêt. Nous avons néanmoins essayé de mettre en correspondance les images avec ce détecteur et les techniques de corrélation présentées dans le chapitre précédent : la corrélation est effectuée dans l'espace de couleur  $H_1H_2H_3$  grâce à la mesure ZNCC (*Zero-mean Normalized Cross Correlation*) et une corrélation retour est effectuée en SAD (*Sum of Absolute Differences*) pour rejeter les mauvais appariements. Cette méthode donne de bons résultats, mais il est possible de les améliorer sensiblement comme le montre la section 4.4.1.1.

Lowe a proposé une approche très intéressante pour la détection et l'appariement de zones d'intérêts : la *Scale Invariant Feature Transform*, connue sous le nom de SIFT [Lowe99a], [Lowe04a]. Le principe du SIFT est de détecter non plus des points mais des zones d'intérêt et de leur associer un descripteur qui est invariant au changement d'échelle et à la rotation. Celui-ci est également robuste aux changements d'illumination, à l'ajout de bruit, et à des changements de

point de vue de la caméra [Lowe04a]. Le descripteur est un vecteur de 128 éléments qui correspond à l'histogramme des orientations du gradient de la région. La mise en correspondance des images par l'algorithme SIFT s'effectue donc en trois étapes : détection des régions d'intérêt, création des descripteurs et appariement des régions.

La détection des régions d'intérêt et la création de leur descripteur sont effectuées grâce au programme mis à disposition par Lowe<sup>13</sup>. L'appariement est quant à lui réalisé comme il le préconise : la mesure de vraisemblance est la distance euclidienne des descripteurs. Pour chaque point de la première image, les distances euclidiennes  $d_1$  et  $d_2$  des deux meilleurs candidats sont comparées et l'appariement n'est considéré comme valide que si  $d_1 < 0,6d_2$ .

## **4.2.2. Estimation de la géométrie épipolaire**

### **4.2.2.1. Géométrie épipolaire des capteurs centraux**

Nous avons vu dans la section 1.4.3.1 le principe de la géométrie épipolaire. Notre système de vision a en effet été conçu pour simplifier celle-ci par une configuration verticale des deux capteurs catadioptriques, qui permet d'obtenir des lignes épipolaires correspondant aux rayons de l'image. Cette géométrie est particulièrement intéressante puisqu'elle simplifie grandement la recherche des pixels homologues comme cela a été souligné dans le chapitre 3.

Dans cette partie, l'accent est mis non plus sur la géométrie épipolaire des deux capteurs du système stéréoscopique, mais sur la géométrie épipolaire d'un même capteur en mouvement. Ainsi, entre deux instants  $k-1$  et  $k$ , le capteur fournit deux images qui peuvent être utilisées pour déterminer la géométrie épipolaire, et qui permettent par conséquent d'obtenir des informations sur le déplacement effectué.

Plusieurs travaux sont consacrés à l'estimation de la géométrie épipolaire pour les capteurs catadioptriques. Ainsi, [Pajdla01a], [Svoboda99a], [Barbosa05a] et [Mariottini05a] proposent des solutions analytiques à l'estimation des courbes épipolaires. Leurs méthodes nécessitent cependant de faire intervenir les équations du miroir, les solutions proposées sont par conséquent spécifiques au type de capteur utilisé.

D'autres travaux, notamment ceux de Bunschoten [Bunschoten03a] et Negishi [Negishi04a], sont fondés sur l'utilisation d'images panoramiques (images omnidirectionnelles dépliées), et considèrent les courbes épipolaires comme l'intersection du plan épipolaire et du cylindre représentant le plan image. Cette approche est intéressante car l'idée sous-jacente est de généraliser la notion de géométrie épipolaire aux capteurs panoramiques.

Nous proposons d'aller encore plus loin dans la généralisation de la géométrie épipolaire

---

<sup>13</sup> <http://people.cs.ubc.ca/~lowe/keypoints/>

aux capteurs centraux grâce au modèle de la sphère d'équivalence. Grâce à ce modèle, la contrainte de coplanarité définie initialement pour les caméras perspectives [Longuet-Higgins81a] [Zhang98b] peut être transposée aux capteurs centraux. Comme le montre la figure 106, si  $\mathbf{X}_{S1}$  et  $\mathbf{X}_{S2}$  correspondent aux projections d'un même point physique  $\mathbf{X}$  sur les deux sphères d'équivalences des capteurs, alors les points  $\mathbf{C}_1$ ,  $\mathbf{C}_2$ ,  $\mathbf{X}_{S1}$ ,  $\mathbf{X}_{S2}$  et  $\mathbf{X}$  appartiennent à un même plan.

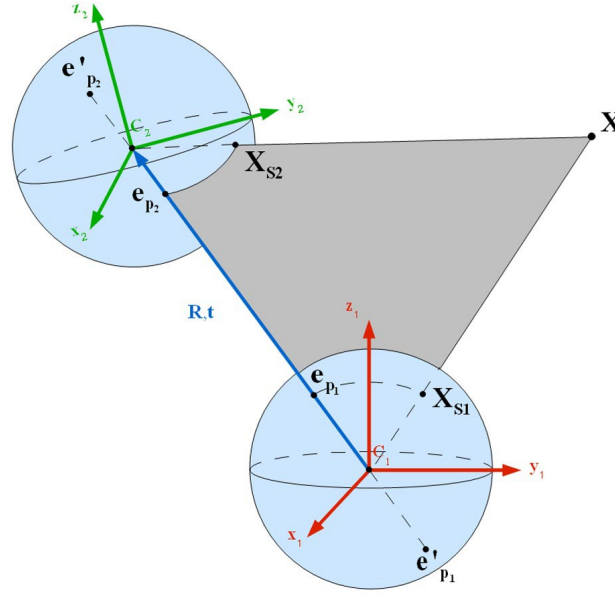


Figure 106: Géométrie épipolaire des capteurs centraux

Soit  $\mathbf{t}$  le vecteur de translation du capteur et  $\mathbf{R}$  la matrice de rotation, la normale du plan épipolaire peut être exprimée dans le repère associé au premier capteur par l'équation (48) et dans le repère associé au second capteur par l'équation (49).

$$\mathbf{N}_1 = \mathbf{t} \times \mathbf{X}_{S1} \quad (48)$$

$$\mathbf{N}_2 = \mathbf{R} \mathbf{N}_1 = \mathbf{R} (\mathbf{t} \times \mathbf{X}_{S1}) \quad (49)$$

Dans le repère associé au second capteur, le point  $\mathbf{X}_{S2}$  appartient au plan épipolaire si la contrainte définie par l'équation (50) est respectée.

$$\mathbf{X}_{S2}^T \mathbf{N}_2 = 0 \quad (50)$$

En combinant les équations (49) et (50), la contrainte de coplanarité est alors définie par :

$$\mathbf{X}_{S2}^T \mathbf{R} (\mathbf{t} \times \mathbf{X}_{S1}) = 0 \quad (51)$$

L'équation (51), c'est-à-dire la contrainte de coplanarité, peut être exprimée sous forme matricielle en introduisant la matrice essentielle  $\mathbf{E}$  :

$$\mathbf{X}_{S2}^T \mathbf{E} \mathbf{X}_{S1} = 0 \quad (52)$$

où  $E=RS$  et  $S$  est une matrice antisymétrique qui caractérise la translation :

$$S = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (53)$$

La notion de ligne épipolaire peut également être généralisée, mais il s'agira désormais d'une courbe épipolaire. Dans le cas d'une caméra perspective, la ligne épipolaire correspond à l'intersection du plan épipolaire avec le plan image. Par analogie, la courbe épipolaire est obtenue par l'intersection du plan épipolaire avec la sphère d'équivalence. Ainsi, pour tout point  $X_{S1}$  sur la première sphère d'équivalence, la courbe épipolaire correspondante sur la seconde sphère est caractérisée par le plan de normale  $N_2$ , dont l'expression est donnée par la formule (54). De même, pour tout point  $X_{S2}$  sur le second capteur, la courbe épipolaire correspondante sur le premier capteur est caractérisée par la normale  $N_1$  définie par l'équation (55).

$$N_2 = E X_{S1} \quad (54)$$

$$N_1 = E^T X_{S2} \quad (55)$$

Les épipôles correspondent, comme pour les caméras perspectives, à l'intersection de toutes les lignes épipolaires, ou encore à l'intersection de la *baseline* avec les plans images. Par analogie, toutes les courbes épipolaires passent par les épipôles qui correspondent cette fois à l'intersection de la *baseline* avec les sphères. Comme l'illustre la figure (106), la *baseline* coupe deux fois la sphère, ce qui fait apparaître un épipôle supplémentaire.

Toutes les courbes épipolaires passent par les épipôles, autrement dit, pour chaque point  $X_{S1}$  de la première sphère, le plan épipolaire défini par la normale  $N_2$  (cf équation 54) contient l'épipôle  $e_{p_2}$ . Par conséquent,  $e_{p_2}$  vérifie l'équation  $e_{p_2}^T E X_{S1} = 0$  pour tout  $X_{S1}$ , ce qui permet d'en déduire que  $e_{p_2}^T \cdot E = 0$ . La solution est obtenue par décomposition en valeurs singulières de  $E$  :  $E = UDV^T$ .  $e_{p_2}$  est alors la troisième colonne de  $U$ . De même, les coordonnées de l'épipôle  $e_{p_1}$  sont données par la troisième colonne de  $V$  [Hartley04a]. Pour chaque capteur, les seconds épipôles  $e'_{p_1}$  et  $e'_{p_2}$  correspondent aux points de la sphère diamétralement opposés aux premiers.

#### 4.2.2.2. Algorithme des huit points pour les capteurs centraux

La matrice essentielle est estimée grâce à l'algorithme des huit points qui a initialement été introduit par Longuet-Higgins [Longuet-Higgins81a]. La mise en correspondance décrite dans la section 4.2.1 permet d'obtenir des couples de pixels homologues entre les instants  $k-1$  et  $k$ . Ces pixels représentant la projection d'un même point 3D, leurs correspondants

$\mathbf{X}_{S1} = [x_1 \ y_1 \ z_1]^T$  et  $\mathbf{X}_{S2} = [x_2 \ y_2 \ z_2]^T$  sur la sphère vérifient la contrainte de coplanarité (52).

En posant  $\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$ , l'équation (52) devient, pour chaque paire de points

appariés :

$$x_2 x_1 e_{11} + x_2 y_1 e_{12} + x_2 z_1 e_{13} + \dots + z_2 x_1 e_{31} + z_2 y_1 e_{32} + z_2 z_1 e_{33} = 0 \quad (56)$$

Avec  $n$  points mis en correspondance, il est par conséquent possible de former le système d'équations linéaires :

$$\mathbf{A} \mathbf{e} = \begin{bmatrix} x_2^1 x_1^1 & x_2^1 y_1^1 & x_2^1 z_1^1 & y_2^1 x_1^1 & y_2^1 y_1^1 & y_2^1 z_1^1 & z_2^1 x_1^1 & z_2^1 y_1^1 & z_2^1 z_1^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^n x_1^n & x_2^n y_1^n & x_2^n z_1^n & y_2^n x_1^n & y_2^n y_1^n & y_2^n z_1^n & z_2^n x_1^n & z_2^n y_1^n & z_2^n z_1^n \end{bmatrix} \mathbf{e} = 0 \quad (57)$$

où  $\mathbf{e} = [e_{11} \ e_{12} \ e_{13} \ e_{21} \ e_{22} \ e_{23} \ e_{31} \ e_{32} \ e_{33}]^T$  est le vecteur contenant les coefficients de la matrice essentielle.

La matrice essentielle ne peut être estimée qu'à un facteur d'échelle près [Hartley04a], et huit points sont par conséquent suffisants à la résolution du système (57), d'où le nom de l'algorithme. Si  $n > 8$ , la solution  $\mathbf{e}$  aux moindres carrés correspond à la dernière colonne de  $\mathbf{V}$  de la décomposition en valeurs singulières de  $\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T$ , c'est-à-dire à la colonne correspondant à la plus petite valeur singulière.

L'algorithme des huit points a souvent été critiqué pour sa sensibilité au bruit et de nombreuses techniques alternatives ont par conséquent été proposées [Luong93a] [Nister04a] [Li06a]. Ces algorithmes sont généralement des algorithmes itératifs considérablement plus compliqués que l'algorithme des huit points. Hartley [Hartley97b] a cependant démontré que l'algorithme des huit points fournit des résultats proches des meilleurs algorithmes itératifs tout en étant vingt fois plus rapide et beaucoup plus facile à implanter. Il souligne également dans cet article l'importance de travailler avec des coordonnées normalisées pour obtenir de bons résultats. Notre méthode a l'avantage d'utiliser des points appartenant à la sphère d'équivalence, ce qui assure la normalisation des coordonnées.

#### 4.2.2.3. Renforcement des contraintes de la matrice essentielle

La matrice essentielle estimée par l'algorithme des huit points est donc obtenue par la résolution du système (57) au sens des moindres carrés. Il est par conséquent possible que la matrice essentielle estimée ne respecte pas tout à fait les propriétés d'une matrice essentielle



parfaite, qui sont [Huang89a]:

- la nullité de son déterminant,
- l'égalité de ses deux valeurs singulières non-nulles.

Une étape de renforcement de contraintes est donc nécessaire pour obtenir une « vraie » matrice essentielle. Les contraintes sont donc renforcées grâce à la propriété mise en évidence par Hartley [Hartley04a]. Ainsi, soit  $E$  une matrice  $3 \times 3$  dont la décomposition en valeurs singulières est  $E = UDV^T$ , avec  $D = \text{diag}(a, b, c)$  et  $a \geq b \geq c$ . La matrice essentielle la plus proche de  $E$  en terme de norme de Frobenius est donnée par  $\hat{E} = U\hat{D}V^T$ , où  $\hat{D} = \text{diag}((a+b)/2, (a+b)/2, 0)$ .

L'impact du renforcement des contraintes peut être observé visuellement. Les courbes épipolaires se coupent dorénavant en un seul point (aux épipôles), alors qu'avant le renforcement elles se coupaient en une région de quelques pixels.

#### 4.2.2.4. Estimation robuste de la matrice essentielle

La normalisation des coordonnées et le renforcement des contraintes de la matrice essentielle ne suffisent cependant pas à obtenir de bons résultats dans le cas d'un mauvais appariement. L'estimation de la matrice essentielle étant réalisée au sens des moindres carrés, la présence d'un ou plusieurs *outliers* dans le système (57) peut en effet fausser totalement le résultat.

L'algorithme des huit points est donc couplé à un algorithme d'estimation robuste appelé « *RANdom SAMple Consensus* » (cf Annexe C). Le RANSAC [Fischler81a] a fait ses preuves pour l'estimation de la matrice essentielle dans le cas de caméras perspectives ([Faugeras01a], [Hartley04a]), et nous proposons par conséquent une adaptation de cet algorithme aux capteurs centraux.

L'estimation de la matrice essentielle en couplant l'algorithme des huit points et le RANSAC peut être résumée par le pseudo-code de la figure 107. Elle consiste à tirer aléatoirement huit points parmi les couples de points appariés pour estimer la matrice essentielle. Il s'agit en effet du nombre de points minimal nécessaire pour pouvoir résoudre le système d'équations (57). Les autres points sont alors testés pour déterminer s'ils respectent ou non la géométrie épipolaire induite par la matrice essentielle calculée, c'est-à-dire que pour chaque point de la première image, son homologue doit appartenir à la courbe épipolaire calculée à partir de  $E$  (cf équations (54) et (55)). Les points respectant la contrainte épipolaire sont alors sauvegardés. Ce processus est répété plusieurs fois, et la meilleure matrice essentielle est la matrice pour laquelle un maximum de points respectent la contrainte épipolaire. L'ensemble de

ces points est alors utilisé pour recalculer la matrice essentielle au sens des moindres carrés grâce au système d'équations (57).

```

k = 0
Tant que k < MAX_ITERATIONS
  maybe_inliers = 8 points sélectionnés aléatoirement
  maybe_essentielle = E estimée avec ces 8 points
  consensus_set = maybe_inliers
  Pour chaque point pas dans maybe_inliers
    Si le point respecte le modèle avec une erreur
    plus petite que MAX_ERROR alors ajouter le
    point à consensus_set
  k=k+1
La meilleure matrice essentielle est celle qui contient
le plus de points dans consensus_set.
Ré-estimation de E avec tous les points contenus dans
consensus_set

```

Figure 107: Pseudo-code pour l'estimation robuste de la matrice essentielle.

Le principe est donc très similaire à celui de l'algorithme utilisé pour les caméras classiques, la principale différence provenant du calcul de l'erreur. En effet, les points n'appartiennent généralement pas parfaitement à la droite épipolaire et il faut autoriser un seuil d'erreur sous lequel le point est considéré comme acceptable.

Dans le cas d'une caméra perspective (cf figure 108), la ligne épipolaire  $\mathbf{l}$  est calculée grâce à la matrice essentielle et aux coordonnées du point  $\mathbf{X}_1$  sur la première image. L'erreur est alors donnée par la distance minimale  $d$  entre le point homologue  $\mathbf{X}_2$  et la droite épipolaire  $\mathbf{l}$ .

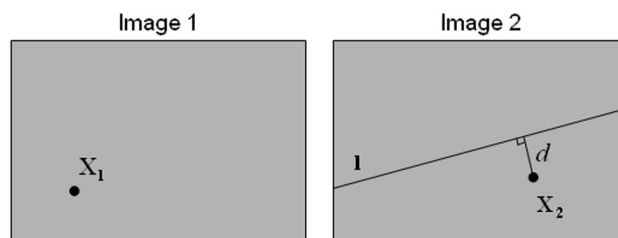


Figure 108: Calcul de l'erreur dans le cas d'une caméra perspective.

Le cas du capteur omnidirectionnel est relativement plus compliqué puisqu'il s'agit alors d'une courbe épipolaire et non plus d'une droite, comme le montre la figure 109. La distance minimale entre le point  $\mathbf{X}_2$  et la courbe épipolaire ne peut pas être obtenue de façon aussi

triviale que dans le cas d'une caméra classique.

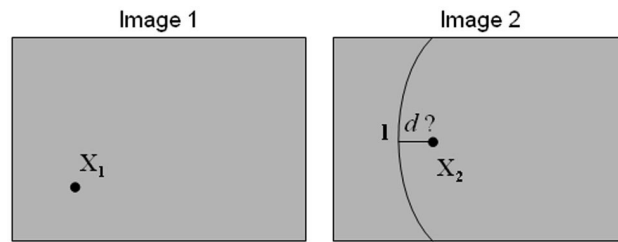


Figure 109: Calcul de l'erreur dans le cas d'un capteur omnidirectionnel.

Pour calculer l'erreur dans le cas d'un capteur omnidirectionnel central, nous proposons donc de raisonner sur la sphère d'équivalence, et d'utiliser la contrainte de coplanarité définie par l'équation (52) et illustrée par la figure 106. En effet, étant donné un point  $X_{s1}$  sur la première sphère, la normale du plan épipolaire correspondant est donnée, dans le repère associé au second capteur, par :  $N_2 = E X_{s1}$ . Dans le cas parfait, les vecteurs  $X_{s2}$  et  $N_2$  sont orthogonaux. L'erreur  $e$  peut donc être qualifiée par une erreur angulaire donnée par le produit scalaire :

$$e = X_{s2}^T N_2 \quad (58)$$

Une fois le critère de mesure de l'erreur défini, il faut déterminer avec attention le nombre de tirages aléatoires de l'algorithme du RANSAC. La totalité des combinaisons possibles ne peut en effet pas être testée car le temps de calcul serait alors trop conséquent. Le nombre de tirages doit donc être choisi judicieusement pour ne pas dégrader les performances du RANSAC.

Le nombre de tirages  $N$  est choisi suffisamment grand pour assurer avec une probabilité  $p$ , qu'au moins l'un des tirages aléatoires de  $s$  points ne contiendra aucun *outlier* [Hartley04a]. Le nombre de tirages à effectuer  $N$  est alors donné par la formule :

$$N = \frac{\log(1-p)}{\log(1-(1-o)^s)} \quad (59)$$

où  $o$  est la proportion d'*outliers*.

Pour notre application, l'algorithme des huit points impose  $s=8$ , et la probabilité  $p$  est choisie très grande ( $p=0,99$ ) pour s'assurer qu'au moins un des tirages ne contiendra pas d'*outlier*. Avec ces données, le nombre de tirages à effectuer peut être calculé grâce à la formule (59) pour différentes proportions d'*outliers*. Quelques valeurs sont données à titre d'exemple dans le tableau 7.

Proportion d' <i>outliers</i> $o$	5%	10%	20%	25%	30%	40%	50%
Nombre de tirages $N$	5	9	26	44	78	272	1177

Tableau 7: Nombre de tirages à effectuer pour différentes proportions d'*outliers*.

La proportion d'*outliers* est déterminée empiriquement grâce aux résultats de la mise en correspondance des images. Grâce aux algorithmes mis en œuvres, et notamment grâce à l'utilisation du SIFT, cette proportion est très faible, de l'ordre de 1%, et est dans tous les cas inférieure à 5%. Une marge de sécurité importante a néanmoins été prise pour s'assurer au maximum de l'absence d'*outliers*. Le nombre de tirages a en effet été fixé à 100, ce qui correspond à une proportion d'*outliers* comprise entre 30% et 40%.

### 4.2.3. Détermination du déplacement

#### 4.2.3.1. Décomposition de la matrice essentielle

Nous avons présenté dans le section 4.2.2 une méthode pour estimer la géométrie épipolaire d'un système omnidirectionnel entre deux instants  $k-1$  et  $k$ . La géométrie épipolaire est en effet décrite par la matrice essentielle  $E$  puisque celle-ci correspond au produit de la matrice de rotation  $R$  et à la matrice antisymétrique  $S$  qui contient les coefficients du vecteur de translation  $\mathbf{t}$  :

$$E=RS, \text{ avec } S=\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (60)$$

La matrice essentielle  $E$  ne fournit donc pas directement le déplacement effectué par le système. Elle doit en effet être décomposée pour obtenir les valeurs de la matrice de rotation  $R$  et de la translation  $\mathbf{t}$ .

La matrice essentielle estimée peut, grâce au renforcement des contraintes présenté dans la section 4.2.2.3, s'écrire  $E=U\text{diag}(1,1,0)V^T$ . Soient  $\mathbf{u}_i$  la  $i^{\text{ème}}$  colonne de  $U$  et  $\mathbf{v}_i$  la  $i^{\text{ème}}$  colonne de  $V$ , quatre solutions sont alors possibles pour la matrice de rotation  $R$  et deux solutions pour le vecteur de translation  $\mathbf{t}$  [Wang00a] :

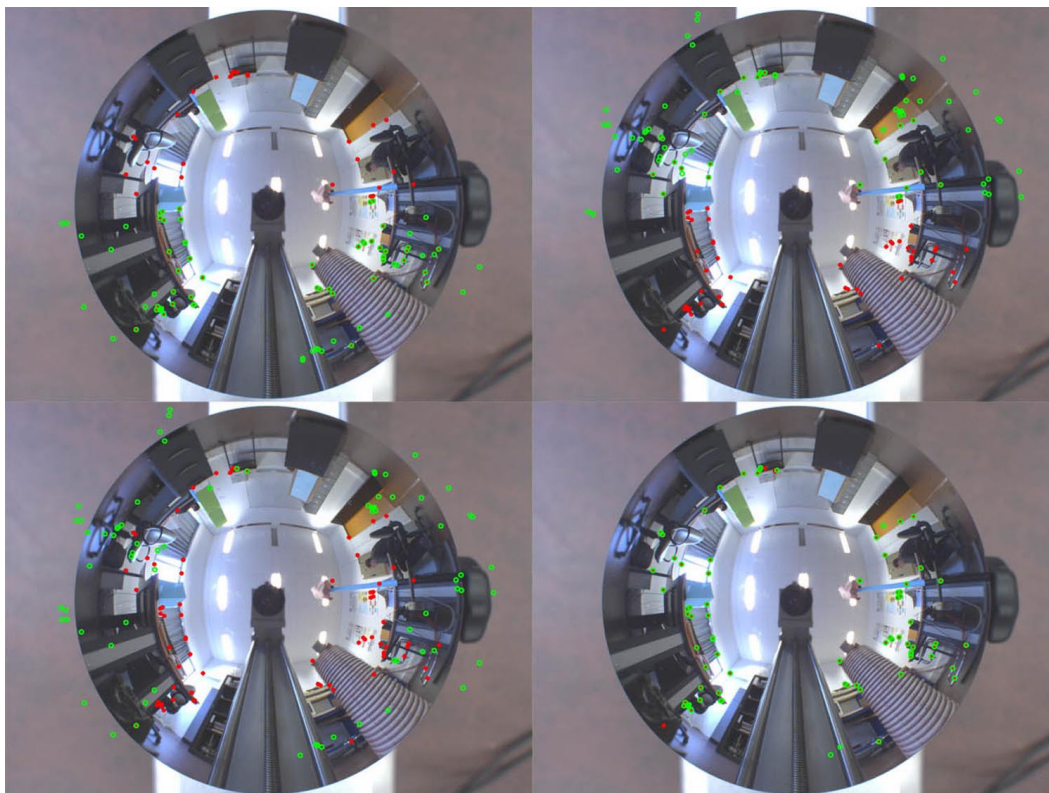
$$\begin{aligned} R_1 &= (-\mathbf{u}_2, \mathbf{u}_1, \mathbf{u}_3) V^T \\ R_2 &= (-\mathbf{u}_2, \mathbf{u}_1, -\mathbf{u}_3) V^T \\ R_3 &= (\mathbf{u}_2, -\mathbf{u}_1, \mathbf{u}_3) V^T \\ R_4 &= (\mathbf{u}_2, -\mathbf{u}_1, -\mathbf{u}_3) V^T \\ \mathbf{t}_1 &= \mathbf{v}_3 \\ \mathbf{t}_2 &= -\mathbf{v}_3 \end{aligned} \quad (61)$$

Deux des solutions de la matrice de rotation peuvent être facilement éliminées puisqu'elles ne respectent pas la contrainte  $\det(R)=1$  d'une matrice de rotation [Bjorkman00a]. Il reste donc quatre combinaisons possibles. Dans le cas d'une caméra perspective, il est trivial de trouver la bonne combinaison puisqu'il suffit de trianguler le point 3D pour les différentes configurations et

de déterminer celle pour laquelle le point 3D se situe devant les deux caméras [Longuet-Higgins81a] [Hartley04a].

Dans le cas de la vision omnidirectionnelle, il n'y a plus de notion de « devant » la caméra, puisque des points provenant de tout l'espace 3D peuvent avoir une projection sur le plan image. Une solution alternative doit par conséquent être trouvée. La solution la plus triviale consisterait à utiliser les données odométriques issues du porteur pour déterminer le sens de la translation et de la rotation, mais il est préférable de n'utiliser que des données visuelles.

Nous proposons donc de tester les quatre combinaisons restantes en triangulant les points pour chacune de ces combinaisons, en reprojétant les points triangulés sur le plan image, et en mesurant l'erreur entre les points 2D reprojétés et les points 2D extraits des images. La combinaison correcte est celle dont l'erreur est minimale, puisque les autres combinaisons conduisent à des projections aberrantes, comme l'illustre la figure 110. Sur cette figure, les points verts correspondent à la reprojektion des points triangulés pour chacune des combinaisons possibles, et les points rouges sont les projections réelles des points remarquables extraits de l'image. Pour ces images, seule la dernière combinaison permet de faire coïncider les points réels et les points reprojétés.



*Figure 110: Test des différentes combinaisons de pose possibles.  
Les points verts correspondent aux points reprojétés, les points rouges aux points remarquables extraits de l'image*

### 4.2.3.2. Détermination du facteur d'échelle

Les algorithmes présentés précédemment permettent d'aboutir à la connaissance du déplacement du système entre les instants  $k-1$  et  $k$ , mais uniquement à un facteur d'échelle près. En effet, bien que la matrice essentielle soit parfaitement définie par l'équation (60), son estimation par l'algorithme des huit points (équation 57) introduit une indétermination sur le facteur d'échelle.

Dans la très grande majorité des travaux, cette ambiguïté est levée en utilisant les données odométriques [Lhuillier08a] [Bunschoten03a]. Cette approche est évidemment intéressante car assez directe, mais il est préférable de n'utiliser que les données visuelles pour s'affranchir de toute communication avec le porteur. L'odométrie optique présente également l'avantage d'être précise même en cas de glissement.

La structure stéréoscopique du système utilisée pour la reconstruction 3D statique prend alors tout son intérêt pour l'estimation des déplacements. En effet, la *baseline* étant connue grâce au calibrage du système (*cf* section 2.4.2), celle-ci peut être utilisée pour lever l'indétermination sur le facteur d'échelle.

Les étapes présentées jusqu'à maintenant ne nécessitaient qu'un seul capteur. Le second capteur doit donc être pris en compte, et les images doivent être mises en relation non seulement entre les instants  $k-1$  et  $k$ , mais aussi entre le capteur haut et le capteur bas du système stéréoscopique, comme l'illustre la figure 111. Les techniques utilisées pour cette mise en relation sont les mêmes que celles présentées dans la section 4.2.1.

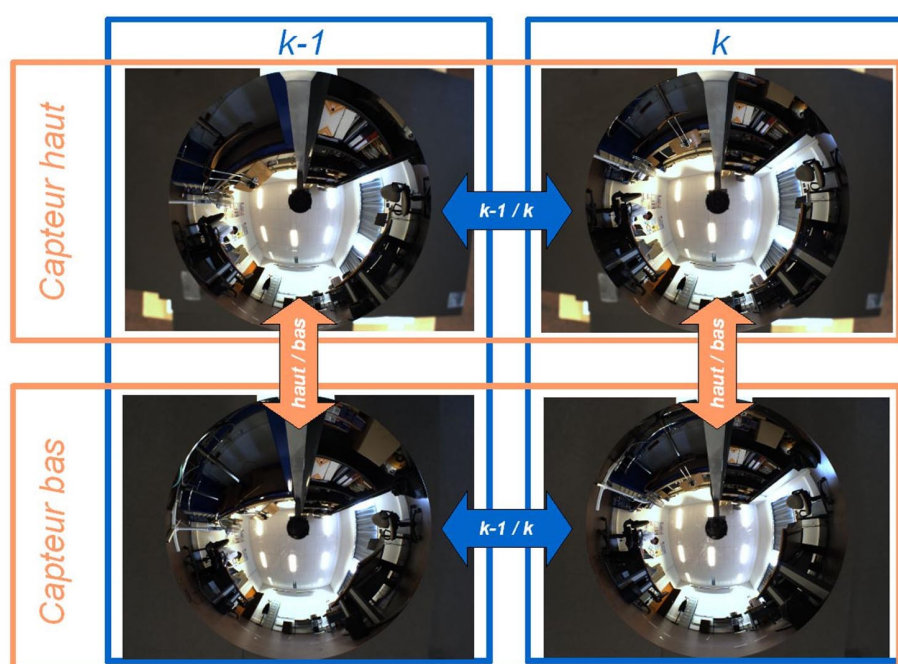


Figure 111: Appariements nécessaires pour retrouver le facteur d'échelle.

Les points appariés entre le capteur du bas et le capteur du haut sont alors triangulés pour aboutir à un jeu de points 3D entre les instants  $k-1$  et  $k$ . Connaissant les coordonnées des points 3D, le facteur d'échelle peut être retrouvé grâce à la norme du vecteur  $\mathbf{t}$ , tel que :

$$\mathbf{t} = \mathbf{X}_k - \mathbf{R} \cdot \mathbf{X}_{k-1} \quad (62)$$

### 4.3. Ajustement de faisceaux

Les algorithmes présentés dans la section précédente permettent l'obtention de valeurs initiales pour le déplacement de la caméra et les coordonnées des points 3D. Les valeurs obtenues ne sont cependant pas assez précises pour pouvoir être utilisées directement. En effet, une erreur trop importante dans l'estimation des déplacements et/ou dans les coordonnées des points 3D conduit à une accumulation des erreurs, et donc à une reconstruction 3D erronée. Une étape d'optimisation supplémentaire, l'ajustement de faisceaux, est donc nécessaire. Cette méthode a atteint une certaine maturité pour les caméras classiques ([Rodriguez03a], [Triggs00b], [Pollefeys04a], [Nister01a]) mais peu de travaux ont été consacrés aux capteurs omnidirectionnels. Nous proposons donc dans cette section un algorithme d'ajustement de faisceaux prenant en compte les spécificités de notre capteur : son champ de vue omnidirectionnel et sa structure stéréoscopique.

#### 4.3.1. Fonctionnelle à minimiser

L'ajustement de faisceaux consiste à minimiser une fonctionnelle correspondant à l'erreur entre la projection estimée de points 3D sur le plan image et leur projection réelle. Soient  $m$  le nombre de positions du système et  $n$  le nombre de points 3D, cette fonctionnelle peut donc s'écrire :

$$F(\mathbf{V}, \mathbf{X}) = \frac{1}{2} \sum_{j=1}^m \sum_{i=1}^n [P(\mathbf{V}_j, \mathbf{X}_i) - \mathbf{x}_{ij}]^2 \quad (63)$$

où :

- $\mathbf{V}_j$  est le vecteur des paramètres de la  $j^{\text{ème}}$  caméra ( $m$  positions),
- $\mathbf{X}_i$  est le vecteur des coordonnées du point 3D  $i$  ( $n$  points 3D) dans le repère monde,
- $\mathbf{x}_{ij}$  est la projection du point 3D  $i$  dans l'image de la caméra  $j$ ,
- $P(\mathbf{V}_j, \mathbf{X}_i)$  est la prédiction de la projection du point 3D  $i$  dans la vue de la caméra  $j$ .

### 4.3.2. Fonction de projection stéréoscopique

L'utilisation d'un capteur stéréoscopique permet de retrouver le facteur d'échelle lors de l'estimation de la structure de la scène, ce qui est impossible avec un capteur monoculaire entre deux instants  $k-1$  et  $k$ . La fonction de projection est donc modifiée par rapport à celle présentée dans la section 2.2.1.3 (équation 17) pour prendre en compte la structure stéréoscopique du système. La précédente fonction de projection permettait d'obtenir, pour un point  $\mathbf{X}$  donné, les coordonnées  $u$  et  $v$  de sa projection sur le plan image. La fonction de projection stéréoscopique fournit maintenant quatre éléments : les coordonnées  $u_{bas}$  et  $v_{bas}$  de la projection du point  $\mathbf{X}$  dans l'image du capteur du bas, et les coordonnées  $u_{haut}$  et  $v_{haut}$  de sa projection sur le plan image du capteur du haut. Le système est donc considéré comme un unique capteur et non plus comme deux capteurs indépendants. Des paramètres intrinsèques supplémentaires sont donc ajoutés au modèle ; il s'agit de la pose  $(\mathbf{R}, \mathbf{t})$  du capteur du haut par rapport au capteur du bas, comme l'illustre la figure 112. Ces paramètres sont ceux qui ont été estimés par la méthode présentée dans la section 2.4.2.

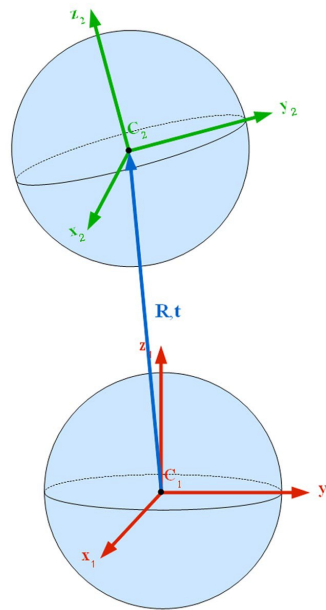


Figure 112: Transformation rigide entre les deux capteurs du système stéréoscopique.

Les nouveaux paramètres intrinsèques sont stockés dans le vecteur  $\mathbf{V}_5 = [q_{w12} \ q_{x12} \ q_{y12} \ q_{z12} \ t_{x12} \ t_{y12} \ t_{z12}]^T$ , la rotation étant caractérisée par un quaternion. Un changement de repère supplémentaire, noté  $C(\mathbf{V}_5)$  est ajouté à la fonction de projection initiale pour exprimer les coordonnées des points soit dans le repère du capteur du bas, soit dans le repère du capteur du haut :

$$P(\mathbf{V}, \mathbf{X}) = K \circ D \circ H \circ C \circ W(\mathbf{V}, \mathbf{X}) \quad (64)$$



avec :

$$\begin{aligned} \mathbf{V} &= [\mathbf{V}_1^T \ \mathbf{V}_2^T \ \mathbf{V}_3^T \ \mathbf{V}_4^T \ \mathbf{V}_5^T]^T \\ \mathbf{V}_1 &= [q_w \ q_x \ q_y \ q_z \ t_x \ t_y \ t_z] \\ \mathbf{V}_2 &= [\xi] \\ \mathbf{V}_3 &= [k_1 \ k_2 \ k_3 \ k_4 \ k_5] \\ \mathbf{V}_4 &= [\alpha \ \gamma_u \ \gamma_v \ u_0 \ v_0] \\ \mathbf{V}_5 &= [q_{w12} \ q_{x12} \ q_{y12} \ q_{z12} \ t_{x12} \ t_{y12} \ t_{z12}] \\ \mathbf{X} &= [X \ Y \ Z]^T \end{aligned}$$

### 4.3.3. Structure de la jacobienne

La minimisation de l'erreur de reprojection définie par l'équation (63) est effectuée par l'algorithme de Levenberg-Marquardt (*cf* Annexe B). L'étape clé de cet algorithme réside dans la résolution de l'équation normale augmentée :

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \Delta = -\mathbf{J}^T \mathbf{e} \quad (65)$$

où  $\lambda$  est un réel qui varie d'itération en itération et  $\mathbf{I}$  est la matrice identité.

La résolution de cette équation nécessite par conséquent le calcul de la jacobienne de la fonction de projection stéréoscopique. Le capteur est supposé calibré, les paramètres intrinsèques du système ne sont donc plus à évaluer. Les paramètres recherchés dans cet algorithme sont les poses  $\mathbf{V}_1^j$  du système et les coordonnées  $\mathbf{X}_i$  des points 3D exprimées dans le repère monde. Les dérivées partielles à calculer sont par conséquent la dérivée de la fonction de projection par rapport à la pose  $\mathbf{V}_1^j$  du système :

$$\frac{\partial P}{\partial \mathbf{V}_1^j} = \frac{\partial K}{\partial D} \frac{\partial D}{\partial H} \frac{\partial H}{\partial C} \frac{\partial C}{\partial W} \frac{\partial W}{\partial \mathbf{V}_1^j} \quad (66)$$

ainsi que la dérivée de la fonction de projection par rapport aux coordonnées des points 3D :

$$\frac{\partial P}{\partial \mathbf{X}_i} = \frac{\partial K}{\partial D} \frac{\partial D}{\partial H} \frac{\partial H}{\partial C} \frac{\partial C}{\partial W} \frac{\partial W}{\partial \mathbf{X}_i} \quad (67)$$

Le détail de la jacobienne de la fonction de projection est donnée en Annexe D, sa structure peut être représentée par la figure 113.

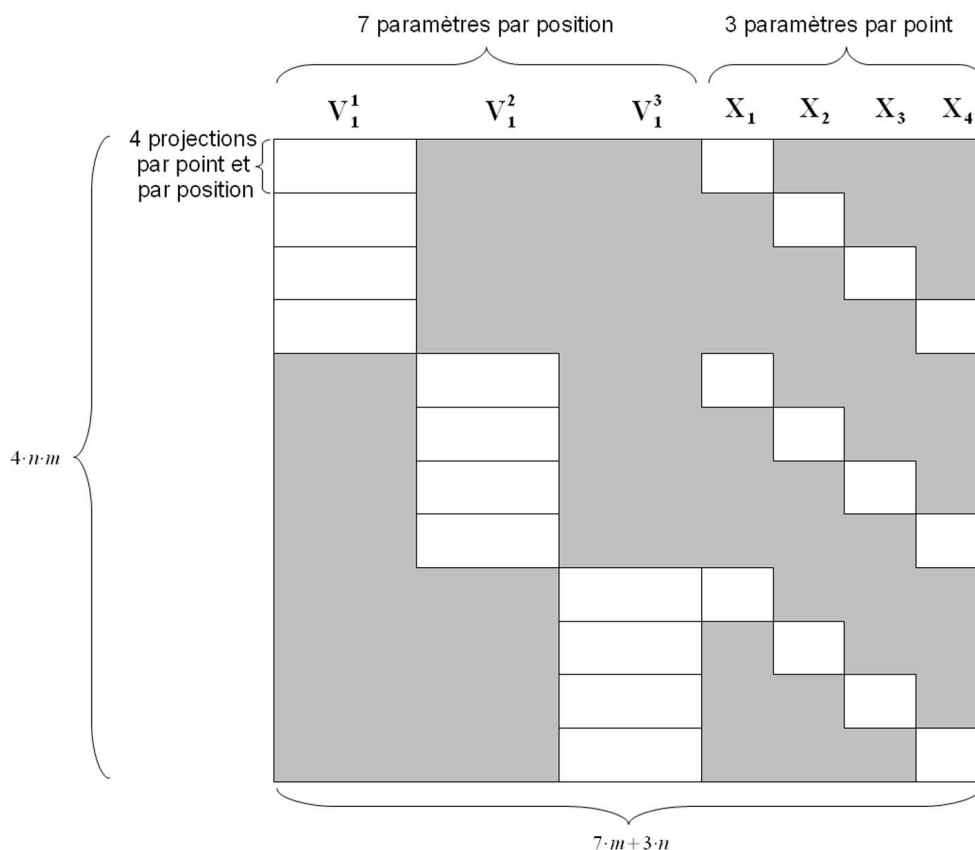


Figure 113: Structure de la jacobienne pour l'ajustement de faisceaux. Les zones grisées correspondent à des valeurs nulles. Cette structure est donnée pour 3 positions et 4 points.

## 4.4. Résultats expérimentaux

### 4.4.1. Phase d'initialisation

#### 4.4.1.1. Mise en correspondance des images

Les résultats de la mise en correspondance des images entre deux instants  $k-1$  et  $k$  sont présentés sur la figure 114 pour les deux méthodes décrites dans la section 4.2.1. Sur cette figure, la mise en correspondance de deux pixels est illustrée par un trait vert reliant la position d'un pixel à l'instant  $k-1$  avec sa position à l'instant  $k$ .

La première méthode (cf figure 114a) correspond à l'appariement de points de Harris en utilisant les corrélations ZNCC et SAD sur l'espace de couleur  $H_1 H_2 H_3$ . La figure 114b est le résultat de la mise en correspondance fondée sur la méthode SIFT.

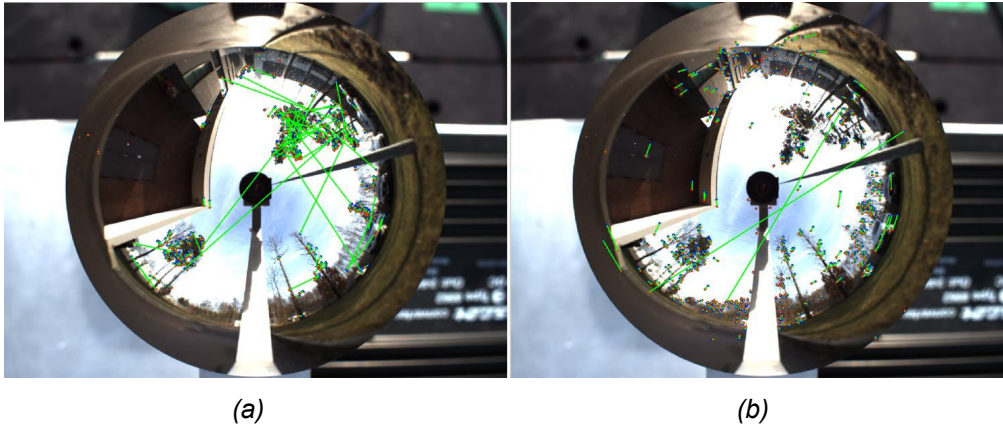


Figure 114: Illustration de la mise en correspondance entre deux instants.  
 (a) Par corrélation sur l'espace couleur  $H_1H_2H_3$  – (b) Par la méthode SIFT

La mise en correspondance a été évaluée en observant la proportion de points appariés et la proportion de points aberrants parmi ces points appariés. Les paramètres du détecteur de Harris ont été réglés de sorte à obtenir un nombre de points proche du nombre de points de SIFT détectés. La proportion de points appariés a ensuite été calculée après le rejet de certains mauvais appariements par corrélation retour pour les points de Harris, et par le calcul de la distance euclidienne des descripteurs pour les points de SIFT. Les résultats résumés dans le tableau 8 illustrent les bonnes performances du SIFT (83 % de points appariés) par rapport aux points de Harris (37% de points appariés). Les points aberrants ont ensuite été identifiés pour calculer l'erreur par rapport au nombre de points appariés. La mise en correspondance fondée sur les points de Harris révèle une proportion de mauvais appariements relativement élevée (12%) comparée à celle utilisant les SIFT.

	<b>Harris</b>	<b>SIFT</b>
Nombre de points détectés	638	641
Proportion de points appariés	37 %	83 %
Proportion de points aberrants	12 %	0,9 %

Tableau 8: Résultats de la mise en correspondance sur la paire d'image de la figure 114.

Cette évaluation permet de mettre en évidence que l'utilisation des SIFT est préférable à la mise en correspondance de points de Harris, même en utilisant des méthodes de rejet des mauvais appariements. Le SIFT présente de plus l'intérêt de mettre en correspondance des points répartis sur toute l'image alors que la corrélation de points de Harris se limite aux zones fortement texturées comme le montre la figure 114.

#### 4.4.1.2. Estimation de la géométrie épipolaire

L'estimation de la géométrie épipolaire, qui consiste à calculer la matrice essentielle, est évaluée en analysant les courbes épipolaires puisque celles-ci sont directement calculées à partir de la matrice essentielle grâce aux équations (54) et (55).

La figure 115a représente les courbes épipolaires théoriques pour un déplacement linéaire suivant l'axe  $y$  (cf figure 60 pour la convention des axes). L'axe reliant les deux épipôles, représenté par une flèche rouge sur la figure suivante, devrait donc être parfaitement vertical sur l'image. La figure 115b a été obtenue en traçant les courbes épipolaires à partir de la matrice essentielle estimée sans l'algorithme du RANSAC. L'estimation de celle-ci par les moindres carrés avec quelques points aberrants fausse le résultat puisque l'axe n'est plus vertical. La figure 115c est le résultat obtenu grâce au RANSAC qui a permis de rejeter les points aberrants et d'obtenir un résultat extrêmement proche du résultat théorique.

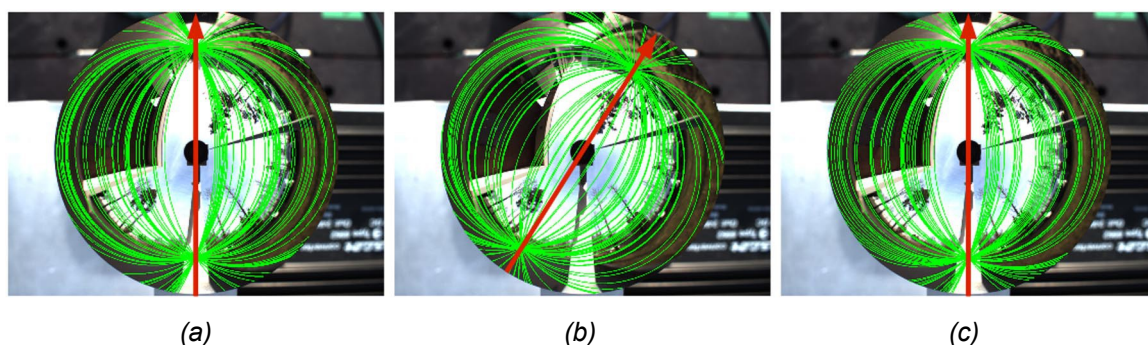


Figure 115: Courbes épipolaires obtenues à partir de la matrice essentielle.  
(a) Courbes théoriques – (b) Courbes sans RANSAC - (c) Courbes avec RANSAC

#### 4.4.1.3. Détermination du déplacement

Le déplacement, obtenu par décomposition de la matrice essentielle, est déterminé dans un premier temps par la matrice de rotation  $\mathbf{R}$  et le vecteur de translation normalisé  $\mathbf{T}$ , puis par la norme de  $\mathbf{T}$  grâce à la méthode présentée dans la section 4.2.3.

L'estimation du déplacement a été évaluée en effectuant un déplacement connu puis en comparant le déplacement estimé au déplacement réalisé. Le capteur a ainsi été déplacé linéairement par pas de 500mm, et le résultat de cette estimation est présenté dans le tableau 9.

Les résultats obtenus pour  $\mathbf{T}$  normalisé sont cohérents puisque les déplacements ont été effectués suivant l'axe  $y$  uniquement. La norme de  $\mathbf{T}$  représente la distance estimée ; l'erreur commise est en moyenne de  $11,66 \% \pm 2,98 \%$ . Cette erreur peut s'expliquer par la présence probable de quelques mauvais appariements lors de la phase de triangulation qui sert à estimer la norme de  $\mathbf{T}$ .

Déplacement n°		1	2	3	4	5	6
<b>T normalisé estimé</b>	<b>Tx</b>	0,006	-0,001	0,057	0,137	-0,013	0,020
	<b>Ty</b>	0,999	0,999	0,997	0,989	0,999	0,999
	<b>Tz</b>	-0,019	-0,049	-0,056	-0,042	-0,028	-0,032
<b>Distance estimée (mm)</b>		458	436	424	435	435	538
<b>Erreur (%)</b>		8,4 %	12,8 %	15,2 %	13 %	13 %	7,6 %

Tableau 9: Déplacements estimés par décomposition de la matrice essentielle.

#### 4.4.2. Ajustement de faisceaux

La phase d'ajustement de faisceaux, qui consiste à optimiser les positions du capteur et les coordonnées 3D des points de façon à minimiser l'erreur de reprojection, a été évaluée sur sa capacité à estimer les déplacements du système. L'environnement utilisé pour cette évaluation, illustré par la figure 116, est complexe puisqu'il contient à la fois des éléments naturels et des éléments très structurés.

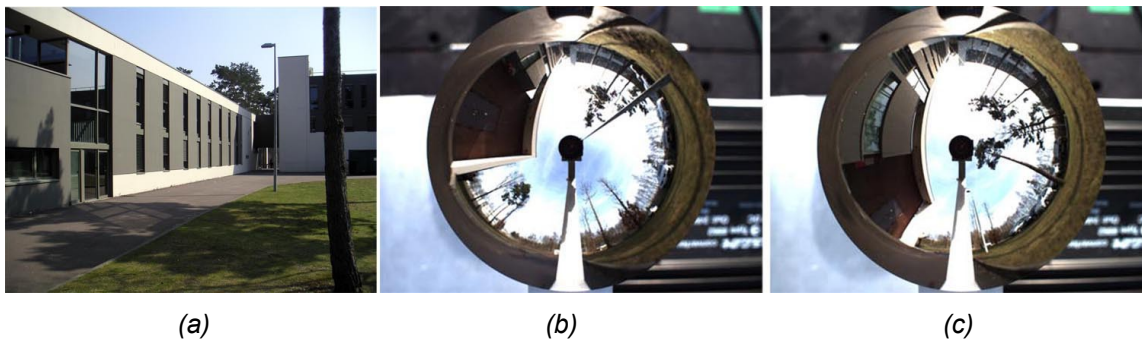


Figure 116: Environnement utilisé pour l'évaluation de l'ajustement de faisceaux.  
(a) Vue perspective – (b) et (c) Extraits de la séquence d'images omnidirectionnelles

Pour chacune des positions du robot, les images omnidirectionnelles des deux capteurs, les données odométriques du robot, ainsi que la carte locale de l'environnement obtenue par le lidar sont sauvegardées. Les images omnidirectionnelles sont utilisées par les algorithmes de reconstruction 3D avec déplacement, alors que les données odométriques et les données issues du lidar servent à évaluer le déplacement du robot. La vérité terrain, c'est-à-dire le déplacement réellement effectué par le robot, est obtenue grâce au lidar car il s'agit du capteur le plus précis.

##### 4.4.2.1. Estimation des déplacements

L'algorithme d'ajustement de faisceaux a été évalué dans un premier temps sur l'estimation des déplacements entre deux instants. Pour chaque déplacement du capteur, la norme du vecteur de translation estimé est calculée et comparée au vecteur de translation obtenu par le lidar. L'erreur commise pour chacun des déplacements de la séquence est présentée sur le graphique de

la figure 117. L'erreur moyenne sur la totalité de cette séquence, qui représente un déplacement de plus de 35 mètres, est de  $9,57\% \pm 5,98\%$ .

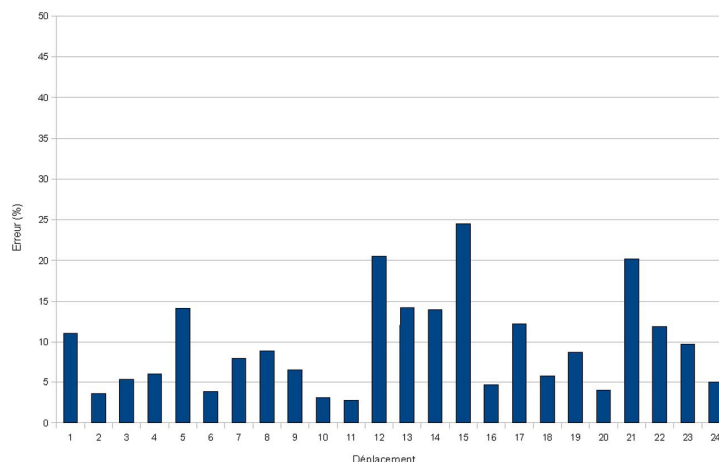


Figure 117: Erreur commise sur l'estimation des déplacements d'une séquence de 25 images.

#### 4.4.2.2. Estimation de la trajectoire

La trajectoire du robot peut être estimée grâce aux données fournies par les différents capteurs embarqués sur le robot. Les figures 118, 119 et 120 représentent la trajectoire du robot (en rouge) au sein de la carte de l'environnement (en vert) obtenue grâce au lidar.

La figure 118 correspond à la trajectoire du robot estimée par odométrie. La trajectoire effectuée lors de cette séquence est une trajectoire linéaire le long de la façade du bâtiment. La trajectoire estimée par odométrie est donc fortement entachée d'erreur puisque la trajectoire estimée est courbe et semble se terminer dans la façade du bâtiment alors que la trajectoire était parallèle à celle-ci. Les informations issues de l'odométrie auraient néanmoins pu être améliorées en appliquant un coefficient de correction sur l'une ou l'autre des roues.

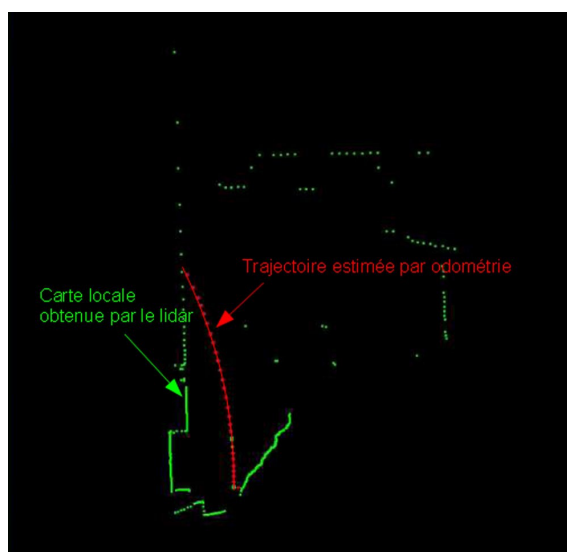


Figure 118: Trajectoire estimée par odométrie.

La trajectoire réelle est estimée à partir des données du lidar, grâce aux cartes locales fournies à chaque instant. Cette trajectoire, représentée sur la figure 119, correspond bien à une trajectoire linéaire.

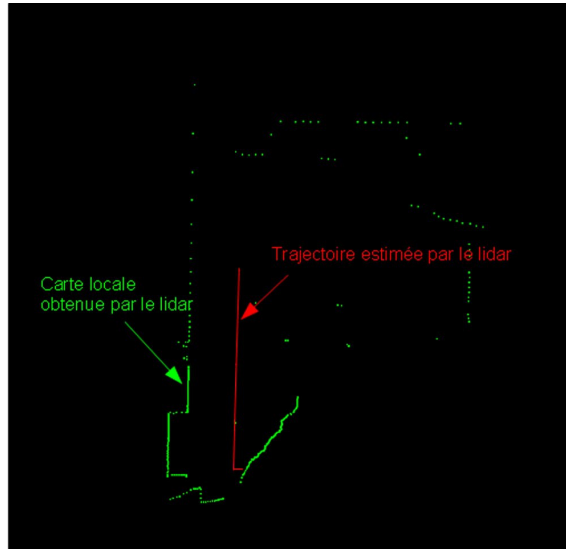


Figure 119: Trajectoire estimée par le lidar.

Enfin, la trajectoire du robot a été estimée grâce à notre méthode d'estimation des déplacements et aboutit à la trajectoire représentée sur la figure 120. Malgré l'erreur commise à chaque déplacement (*cf* section 4.4.2.1), cette trajectoire est très similaire à celle obtenue par le lidar. Notre méthode permet donc d'obtenir une estimation très cohérente en n'utilisant que des données visuelles.

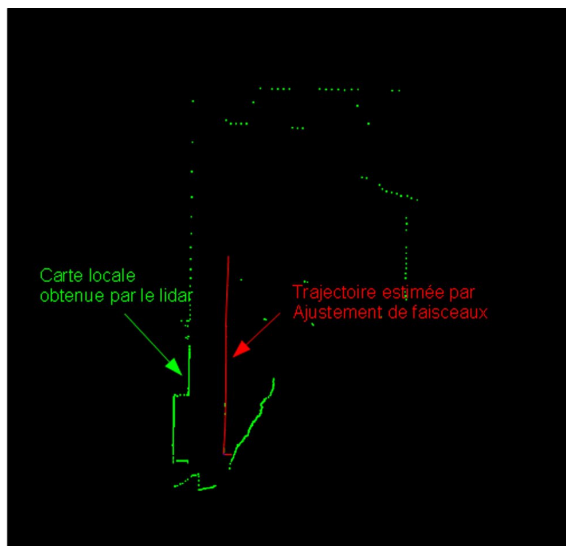


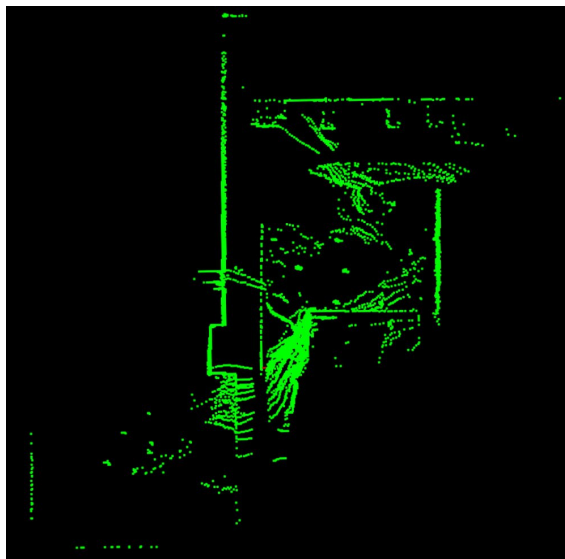
Figure 120: Trajectoire estimée par ajustement de faisceaux.

#### 4.4.2.3. Recalage des cartes locales

L'estimation des déplacements a ensuite été évaluée par recalage des cartes locales

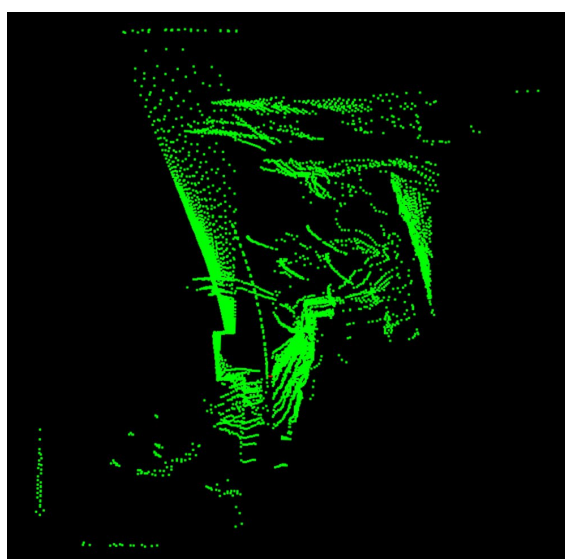
obtenues par le lidar. Connaissant les positions du robot, il est en effet possible de replacer toutes les cartes locales dans un même repère afin d'obtenir la carte globale de l'environnement.

Les déplacements les plus précis sont ceux obtenus par le lidar. Ces bons résultats sont illustrés par la figure 121 : avec ces déplacements, les cartes locales se superposent parfaitement.



*Figure 121: Fusion des cartes locales avec les déplacements estimés par le lidar.*

La même procédure a été répétée avec les déplacements obtenus par odométrie. La carte globale obtenue, présentée sur la figure 122, illustre bien que l'estimation des déplacements par cette méthode n'est pas précise puisque toutes les cartes locales sont décalées. Ces résultats mettent en évidence que les données odométriques ne sont pas suffisamment précises pour être utilisées par les algorithmes de navigation. Elle permet cependant de fournir facilement une première estimation des positions du robot qui peut être utilisée pour initialiser des algorithmes.



*Figure 122: Fusion des cartes locales avec les déplacements estimés par odométrie.*



La fusion des cartes locales en utilisant les déplacements estimés par ajustement de faisceaux donne de bons résultats (*cf* figure 123). Bien qu'une légère dispersion des points soit visible, les cartes locales se superposent correctement et les façades des bâtiments sont facilement identifiables.

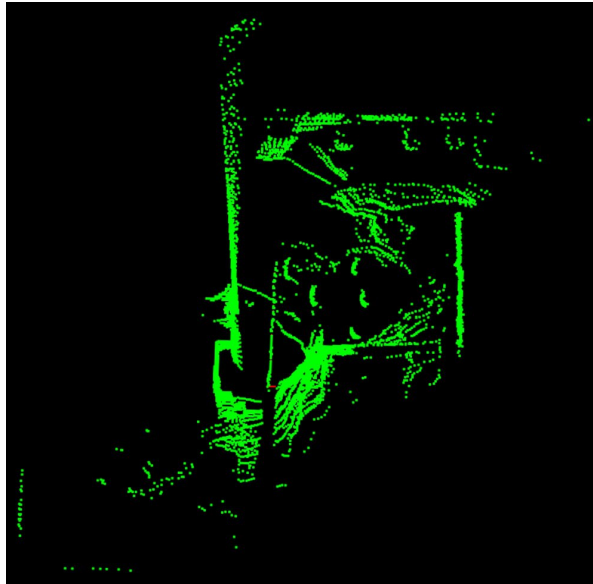


Figure 123: Fusion des cartes locales avec les déplacements estimés par ajustement de faisceaux.

L'algorithme d'ajustement de faisceaux permet donc d'aboutir à une bonne estimation des déplacements en utilisant des données uniquement visuelles. Les résultats obtenus fournissent par conséquent une solution intéressante pour les problèmes de navigation et de reconstruction 3D.

### **4.5. Conclusion et perspectives**

Le chapitre 4 a été consacré à une méthode de reconstruction 3D avec déplacement, c'est-à-dire exploitant le mouvement du système pour retrouver la structure de la scène et les déplacements du capteur. La méthode que nous avons développée avait principalement pour but l'estimation des positions du robot puisqu'elle ne permet la reconstruction 3D que d'un nombre de points limité.

Notre méthode repose sur une phase d'initialisation des déplacements grâce à l'estimation de la géométrie épipolaire du système, suivi d'une optimisation des positions du capteur et des coordonnées des points de la scène par ajustement de faisceaux.

Le principal avantage de notre méthode est sa capacité à fournir les déplacements du système à partir de données visuelles uniquement, c'est-à-dire sans utilisation de données odométriques ou de données issues d'un lidar. Elle permet d'obtenir le facteur d'échelle de la

scène grâce à la structure stéréoscopique du système, là où la majorité des travaux ne le prennent pas en compte ou l'estime grâce à des données supplémentaires.

Les résultats expérimentaux sont prometteurs, mais quelques améliorations peuvent cependant être apportées pour obtenir un système parfaitement exploitable en robotique mobile. La première concerne l'optimisation du temps de calcul pour obtenir un système d'estimation des déplacements fonctionnant en temps réel. Cette optimisation doit se faire en particulier sur la phase qui requiert le plus de temps de calcul : la détection et l'appariement des amers visuels. Une approche intéressante consisterait à utiliser une méthode similaire à la *Scale-Invariant Feature Transform* (SIFT) mais plus rapide, tels que les *Speeded Up Robust Features* (SURF) [Bay08a].

La seconde amélioration porte sur la précision de l'estimation des déplacements. Deux pistes semblent particulièrement intéressantes à explorer : la première consisterait à effectuer un ajustement de faisceaux global, afin de recalibrer toutes les données et éviter le phénomène d'accumulation des erreurs. La seconde consisterait à s'intéresser à la détection de fermeture de boucle, qui a pour but de détecter que le robot est revenu sur une position passée. Cette information est importante puisqu'elle permet d'accroître les performances du système en recalant la totalité des données lorsque le robot repasse par une position déjà explorée.

Le problème de la localisation et de la cartographie simultanées reste néanmoins un problème ouvert. En effet, les solutions proposées actuellement sont souvent limitées soit en terme de taille de l'environnement, soit en terme de nombre de points reconstruits. Seule la résolution de ces deux problèmes conférerait aux robots une réelle autonomie qui leur permettrait de naviguer sur de longues distances et d'explorer des environnements inconnus *a priori*.

# ***Conclusion et perspectives***

## ***Conclusion***

Les travaux de thèse présentés dans ce manuscrit s'inscrivaient dans le cadre d'un marché de la DGA (Délégation Générale pour l'Armement) portant sur l'étude d'un moyen de reconstruction tridimensionnelle de l'environnement, à partir d'informations de vision, pour la préparation d'interventions.

Nos travaux ont été consacrés dans un premier temps à la conception de l'architecture matérielle du capteur. Après une étude des différents capteurs existants, notre choix s'est porté sur un système de stéréovision omnidirectionnelle composé de capteurs catadioptriques. Ce choix est particulièrement pertinent pour la robotique mobile puisqu'il permet d'obtenir des informations sur la totalité de l'environnement sans déplacement du capteur.

Avant de pouvoir modéliser l'environnement en trois dimensions, il est nécessaire de calibrer le capteur. La première phase du calibrage consiste à déterminer la relation mathématique liant un point 3D et sa projection sur le plan image. Nous avons choisi d'utiliser un modèle générique, appelé sphère d'équivalence, et nous avons développé les outils nécessaires à l'estimation des paramètres de ce modèle. La seconde phase du calibrage a eu pour but de déterminer la pose relative des deux capteurs. Pour ce faire, nous avons proposé une méthode fondée sur l'utilisation d'une mire plane qui permet d'obtenir cette pose, en prenant soin d'estimer également le facteur d'échelle. Ce facteur est important car il permet par la suite de pouvoir dimensionner l'environnement.

La structure stéréoscopique du système permet de reconstruire l'environnement du robot sans déplacement. Nous avons donc proposé des algorithmes de reconstruction 3D statique. Nos contributions dans ce domaine concernent le développement d'un algorithme de programmation dynamique pour les capteurs omnidirectionnels, et un algorithme de détection de droites dans les images omnidirectionnelles pour la reconstruction 3D d'environnements peu texturés.

Le capteur étant embarqué sur un robot mobile, nous avons exploité les déplacements pour enrichir les reconstructions 3D. Les déplacements peuvent en effet servir à fusionner les différentes reconstructions statiques, mais il faut alors être capable de les estimer. Notre contribution porte sur un algorithme d'ajustement de faisceaux pour les capteurs stéréoscopiques omnidirectionnels : une première estimation est obtenue en exploitant les propriétés de la géométrie épipolaire puis les résultats sont affinés grâce à l'algorithme de Levenberg-Marquardt.

Notre méthode a l'avantage de ne nécessiter que des données visuelles pour aboutir à la caractérisation complète des déplacements, y compris le facteur d'échelle.

Tous les algorithmes développés ont été évalués expérimentalement soit sur des images réelles, soit sur des images de synthèse.

## ***Perspectives***

La reconstruction 3D de l'environnement d'un robot mobile reste un thème de recherche encore ouvert puisqu'aucune méthode ne permet d'obtenir un modèle 3D réaliste, en temps réel et suffisamment robuste pour répondre à tous types de problèmes.

L'architecture matérielle du capteur pourrait encore être améliorée : il serait par exemple intéressant de miniaturiser le capteur pour que celui-ci soit plus facilement embarquable, par exemple sur un drone. Sur une image omnidirectionnelle, seule la partie contenant le miroir fournit des informations sur l'environnement. Les pixels situés en dehors du miroir pourraient par conséquent être utilisés pour fournir des vues supplémentaires en utilisant des surfaces réfléchissantes supplémentaires.

Le calibrage des systèmes de vision omnidirectionnelle a atteint une bonne maturité, et les résultats obtenus sont généralement très précis. Désormais, les améliorations possibles portent plutôt sur la facilité d'utilisation que sur la précision. Un outil de calibrage totalement automatique permettrait de démocratiser l'usage des capteurs catadioptriques, à l'image des caméras classiques ces dernières années.

Le principal obstacle à la reconstruction 3D statique est la mise en correspondance des entités. Pour la reconstruction dense, il serait particulièrement intéressant de s'intéresser à la forme des fenêtres et aux mesures de corrélation pour les adapter à la géométrie du capteur. Dans des environnements peu texturés, la détection de plans dans les images omnidirectionnelles et leur mise en correspondance permettrait de reconstruire de façon réaliste les façades de bâtiments ou les couloirs.

Le problème de la localisation et de la cartographie simultanées est sans doute le sujet de recherche le plus actif dans le domaine de la robotique mobile ces dernières années car de nombreux problèmes restent à résoudre. Notre algorithme d'estimation des déplacements pourrait être amélioré en traitant les effets de dérive apparaissant dans des séquences longues par une minimisation globale lancée à intervalles réguliers et en prenant en compte les problèmes de fermeture de boucles.

# Bibliographie

- [Ayache89a] Ayache, N., *Vision Stéréoscopique et Perception Multisensorielle : Applications à la Robotique Mobile*, InterEditions, 1989
- [Bailey06a] Bailey, T. and Durrant-Whyte, H., *Simultaneous Localization and Mapping (SLAM): Part II*, Robotics and Automation Magazine, vol. 13, pp. 108-117, 2006
- [Baker81a] Baker, H.H. and Binford, T.O., *Depth from edge and intensity based stereo*, International Joint Conference on Artificial Intelligence (IJCAI), pp. 631-636, 1981
- [Baker99a] Baker, S. and Nayar, S.K., *A Theory of Single-Viewpoint Catadioptric Image Formation*, International Journal of Computer Vision (IJCV), vol. 35, pp. 175-196, 1999
- [Barbosa04a] Gonzalez-Barbosa, J.J., *Vision panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images*, Thèse de doctorat, Université Toulouse III, 2004
- [Barbosa05a] Gonzalez-Barbosa, J.J. and Lacroix, S., *Fast Dense Panoramic Stereovision*, IEEE International Conference on Robotics and Automation (ICRA), pp. 1210-1215, 2005
- [Barreto03b] Barreto, J.P. and Araujo, H., *Direct Least Square Fitting of Paracatadioptric Line Images*, IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), pp. 78-84, 2003
- [Barreto06a] Barreto, J.P., *A Unifying Geometric Representation for Central Projection Systems*, Computer Vision and Image Understanding, vol. 3, pp. 207-217, 2006
- [Barreto06b] Barreto, J.P. and Araujo, H., *Fitting Conics to Paracatadioptric Projections of Lines*, Computer Vision and Image Understanding (CVIU), vol. 101, pp. 151-165, 2006
- [Baumgart74a] Baumgart, B.G., *Geometric modelling for computer vision*, Thèse de doctorat, Stanford University, 1974
- [Bay08a] Bay, H. and Ess, A. and Tuytelaars, T. and Van Gool, L., *SURF: Speeded Up Robust Features*, Computer Vision and Image Understanding (CVIU), vol. 110, pp. 346-359, 2008
- [Benosman01] R. Benosman, S. B. Kang, *Panoramic Vision : Sensors, Theory and Applications*, , 2001
- [Benosman96a] Benosman, R. and Maniere, T. and Devars, J., *Multidirectional stereovision sensor, calibration and scenes reconstruction*, International Conference on Pattern Recognition (ICPR), pp. 161-165, 1996
- [Bigot-Marchand08a] Bigot-Marchand, S., *Outils de traitement d'images adaptés au traitement d'images omnidirectionnelles*, Université de Picardie Jules Verne, 2008
- [Bjorkman00a] Bjorkman, M. and Eklundh, J.O., *A real-time system for epipolar geometry and ego-motion estimation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 506-513, 2000
- [Bobick99a] Bobick, A.F. and Intille, S.S., *Large Occlusion Stereo*, International Journal of Computer Vision (IJCV), vol. 33, pp. 181-200, 1999
- [Bouguet] Bouguet, J.Y., *Camera Calibration Toolbox for Matlab*, [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/), 2009
- [Brand93a] Brand, P. and Mohr, R. and Bodet, P., *Distorsions optiques : correction dans un modèle projectif*, Rapport technique, Institut National de Recherche en Informatique et en Automatique, 1993
- [Braquelair97a] Braquelair, J.P. and Brun, L., *Comparison and Optimization of Methods of*

*Color Image Quantization*, IEEE Transactions on Image Processing, vol. 6, pp. 1048-1052, 1997

- [Bresenham65a] Bresenham, J.E., *Algorithm for computer control of a digital plotter*, IBM Systems Journal, vol. 4, pp. 25-30, 1965
- [Broadhurst01a] Broadhurst, A. and Drummond, T.W. and Cipolla, R., *A Probabilistic Framework for Space Carving*, International Conference on Computer Vision (ICCV), pp. 388-393, 2001
- [Bunschoten03a] Bunschoten, R. and Kröse, B., *Robust Scene Reconstruction from an Omnidirectional Vision System*, IEEE Transactions on Robotics and Automation, vol. 19, pp. 351-357, 2003
- [Caglioti05a] Caglioti, V. and Gasparini, S., *On the localization of straight lines in 3D space from single 2D images*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1129-1134, 2005
- [Carceroni06a] Carceroni, R. and Kumar, A. and Daniilidis, K., *Structure From Motion with Known Camera Positions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 477-484, 2006
- [Chambon03a] Chambon, S. and Cruzil, A., *Dense matching using correlation: new measures that are robust near occlusions*, British Machine Vision Conference (BMVC), pp. 143-152, 2003
- [Chambon04a] S. Chambon and A. Cruzil, *Mesures de corrélation pour des images couleur*, Traitement du Signal, vol. 21, pp. 635-659, 2004
- [Correa05a] Corrêa, F.R. and Guizilini, V.C. and Okamoto, J., *Omnidirectional stereovision system with two-lobe hyperbolic mirror for robot navigation*, International Congress of Mechanical Engineering, pp. 653-660, 2005
- [Courteille07a] Courteille, F. and Durou, J.D. and Gurdjos, P., *Shape From Contour for the Digitization of Curved Documents*, Asian Conference on Computer Vision (ACCV), pp. 196-205, 2007
- [Cox96a] Cox, I.J. and Hingorani, S.L. and Rao, S.B., *A Maximum Likelihood Stereo Algorithm*, Computer Vision and Image Understanding, vol. 63, pp. 542-567, 1996
- [Cozman97a] Cozman, F. and Krotkov, E., *Depth from Scattering*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 801, 1997
- [Delahoche97a] Delahoche, L. and Pégard, C. and Marhic, B. and Vasseur, P., *A Navigation System Based on an Omnidirectional Vision Sensor*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 718-724, 1997
- [Devernay97a] Devernay, F., *Vision stéréoscopique et propriétés différentielles des surfaces*, Thèse de doctorat, Ecole polytechnique, 1997
- [Durou07a] Durou, J.D. and Courteille, F., *Integration of a Normal Field without Boundary Condition*, International Conference on Computer Vision (ICCV), pp. 1-8, 2007
- [Durrant-Whyte06a] Durrant-Whyte, H. and Bailey, T., *Simultaneous Localization and mapping: Part I*, Robotics and Automation Magazine, vol. 13, pp. 99-110, 2006
- [Duvieubourg06a] Duvieubourg, L. and Ambellouis, S. and Cabestaing, F., *Single-Camera stereovision setup*, International Conference on Computer Vision and Graphics (ICCVG), pp. 173-178, 2006
- [Fabrizio08a] Fabrizio, J. and Devars, J., *An Analytical Solution to the Perspective-N-Point Problem for Common Planar Camera and for Catadioptric Sensor*, International Journal of Image and Graphics (IJIG), vol. 8, pp. 135-155, 2008
- [Faugeras01a] O. Faugeras and Q.T. Luong, *The Geometry of multiple images : The laws that govern the formation of multiple images of a scene and some of their applications*, MIT Press, 2001
- [Favaro05a] Favaro, P. and Soatto, S., *A Geometric Approach to Shape from Defocus*, IEEE

- Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 27, pp. 406-417, 2005
- [Fiala02a] Fiala, M. and Basu, A., *Line Segment Extraction in Panoramic Images*, International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), pp. 179-186, 2002
- [Fischler81a] Fischler, M.A. and Bolles, R.C., *Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, pp. 381-395, 1981
- [Forsyth02b] Forsyth, D.A., *Shape From Texture without Boundaries*, European Conference on Computer Vision (ECCV), pp. 225-239, 2002
- [Geyer00a] Geyer, C. and Daniilidis, K., *A Unifying Theory for Central Panoramic Systems and Practical Implications*, European Conference on Computer Vision (ECCV), pp. 445-461, 2000
- [Geyer01a] Geyer, C. and Daniilidis, K., *Catadioptric projective geometry*, International Journal of Computer Vision (IJCV), vol. 43, pp. 223-243, 2001
- [Geyer02a] Geyer, C. and Daniilidis, K., *Paracatadioptric Camera Calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 687-695, 2002
- [Ha09a] Ha, J.E., *Automatic detection of chessboard and its applications*, Optical Engineering, vol. 48, pp. 67205, 2009
- [Harris88a] Harris, C. and Stephens, M., *A Combined Corner and Edge Detector*, Fourth Alvey Vision Conference, pp. 147-151, 1988
- [Hartley04a] Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004
- [Hartley92a] R. Hartley and R. Gupta and T. Chang, *Stereo from uncalibrated cameras*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 761-764, 1992
- [Hartley97b] Hartley, R., *In Defense of the Eight-Point Algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 19, pp. 580-593, 1997
- [Horaud95a] Horaud, R. and Monga, O., *Vision par ordinateur : outils fondamentaux*, Editions Hermès, 1995
- [Horn75a] Horn, B.K.P., *Obtaining Shape From Shading Information*, The Psychology of Computer Vision, vol. 4, pp. 115-155, 1975
- [Hough62a] Hough, P.V.C., *Method and means for recognizing complex patterns*, U.S. Patent 3069654, 1962
- [Huang89a] Huang, T.S. and Faugeras, O., *Some Properties of the E Matrix in Two-View Motion Estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 11, pp. 1310-1312, 1989
- [Ieng05a] Ieng, S. and Benosman, R., *Les surfaces caustiques par la géométrie - Application aux capteurs catadioptriques*, Traitement du Signal, vol. 22, pp. 433-442, 2005
- [Kalman60a] Kalman, R.E., *A New Approach to Linear Filtering and Prediction Problems*, Transactions of the ASME - Journal of Basic Engineering, vol. 82, pp. 35-45, 1960
- [Kutulakos00a] Kutulakos, K.N. and Seitz, S.M., *A Theory of Shape by Space Carving*, International Journal of Computer Vision (IJCV), vol. 38, pp. 199-218, 2000
- [Laurent97a] Laurent, M., *Acquisition 3D au service de l'industrie*, RobAut, vol. 19, pp. 38-32, 1997
- [Laurgeau97a] Laurgeau, C., *Perception géométrique tridimensionnelle en robotique*, Techniques de l'Ingénieur, vol. 3, pp. 1-10, 1997

- [Layerle08a] Layerle, J.F. and Savatier, X. and Ertaud, J.Y. and Mouaddib, E.M., *Catadioptric Sensor for a Simultaneous Tracking of the Driver's Face and the Road Scene*, Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS), pp. , 2008
- [Lee01a] Lee, Y.J. and Chung, M.J., *A Reliable Feature Matching Method in Omnidirectional Views for Autonomous Map Generation of a Mobile Robot*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), pp. 895-900, 2001
- [Levenberg44a] Levenberg, K., *A method for the solution of certain problems in least squares*, Quarterly of Applied Mathematics, vol. 2, pp. 164-168, 1944
- [Lhuillier05a] M. Lhuillier, *Automatic Structure and Motion using a Catadioptric Camera*, 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras (OMNIVIS'05), pp. , 2005
- [Lhuillier08a] M. Lhuillier, *Vers la modélisation 3D flexible avec une caméra catadioptrique*, Proc. of the 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA 2008), pp. , 2008
- [Li06a] Li, H. and Hartley, R., *Five-point motion estimation made easy*, International Conference on Pattern Recognition (ICPR), pp. 630-633, 2006
- [Lin04a] Lin, S. and Gu, J. and Yamazaki, S. and Shum, H.Y., *Radiometric Calibration from a Single Image*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 938-945, 2004
- [Longuet-Higgins81a] Longuet-Higgins, H.C., *A computer algorithm for reconstructing a scene from two projections*, Nature, vol. 293, pp. 133-135, 1981
- [Lowe04a] Lowe, D.G., *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision (IJCV), vol. 60, pp. 91-110, 2004
- [Lowe99a] Lowe, D.G., *Object Recognition from Local Scale-Invariant Features*, International Conference on Computer Vision (ICCV), pp. 1150–1157, 1999
- [Luong93a] Luong, G.T. and Deriche, R. and Faugeras, O. and Papadopoulos, T., *On determining the fundamental matrix: analysis of different methods and experimental results*, Rapport de Recherche, Institut National de Recherche en Informatique et en Automatique, 1993
- [Mariottini05a] Mariottini, G.L. and Prattichizzo, D., *The Epipolar Geometry Toolbox : multiple view geometry and visual servoing for MATLAB*, IEEE Robotics and Automation Magazine, vol. 12, pp. 26-39, 2005
- [Marquardt63a] Marquardt, D.W., *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, Journal of the Society for Industrial and Applied Mathematics, vol. 11, pp. 431-441, 1963
- [Mei07a] Mei, C. and Rives, P., *Single View Point Omnidirectional Camera Calibration from Planar Grids*, IEEE International Conference on Robotics and Automation (ICRA), pp. 3945-3950, 2007
- [Mei07b] C. Mei, *Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping*, Thèse de doctorat, Mines de Paris, 2007
- [Mouaddib05a] Mouaddib, E.M., *La Vision Omnidirectionnelle*, Journées Nationales de la Recherche en Robotique (JNRR'05), pp. , 2005
- [Mouaddib05c] Mouaddib, E.M. and Sagawa, R. and Echigo, T. and Yagi, Y., *Stereo vision with a single camera and multiple mirrors*, IEEE International Conference on Robotics and Automation (ICRA), pp. 812-817, 2005
- [Mouragnon09a] Mouragnon, E. and Lhuillier, M. and Dhome, M. and Dekeyser, F. and Sayd, P., *Generic and Real-Time Structure from Motion using Local Bundle Adjustment*, Image and Vision Computing Journal, vol. 27, pp. 1178-1193, 2009
- [Nalwa96a] Nalwa, V.S., *A True Omnidirectional Viewer*, Rapport Technique, Bell Laboratories, Holmdel, NJ 07733, USA, 1996



- [Nalwa99a] Nalwa, V.S., *Method and system for panoramic viewing*, U.S. Patent 5990934, 1999
- [Nayar94a] Nayar, S.K. and Nakagawa, Y., *Shape From Focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 16, pp. 824-831, 1994
- [Nayar97a] Nayar, S.K. and Baker, S., *Catadioptric Image Formation*, DARPA Image Understanding Workshop (IUW), pp. 1431-1437, 1997
- [Nayar97b] Nayar, S.K., *Omnidirectional Video Camera*, DARPA Image Understanding Workshop (IUW), pp. 235-241, 1997
- [Nayar97c] Nayar, S.K., *Catadioptric Omnidirectional Camera*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 482-488, 1997
- [Negishi04a] Negishi, Y. and Miura, J. and Shirai, Y., *Calibration of Omnidirectional Stereo for Mobile Robots*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2600-2605, 2004
- [Nister01a] Nister, D., *Automatic Dense Reconstruction from Uncalibrated Video Sequence*, Thèse de doctorat, Royal Institute of Technology KTH, Stockholm, Sweden, 2001
- [Nister04a] Nister, D., *An efficient solution to the five-point relative pose problem*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 26, pp. 756-770, 2004
- [Ohta85a] Ohta, Y. and Kanade, T., *Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 7, pp. 139-154, 1985
- [Orghidan05b] Orghidan, R. and Salvi, J. and Mouaddib, E.M., *Accuracy estimation of a new omnidirectional 3D vision sensor*, IEEE International Conference on Image Processing (ICIP), pp. 365-368, 2005
- [Pajdla01a] Pajdla, T. and Svoboda, T. and Hlavac, V., *Panoramic Vision: Sensors, Theory and Applications*, Chapitre Epipolar Geometry of Central Panoramic Catadioptrics Cameras du livre , Ed. Gries, D. and Schneider, F.B., pp. 73-102, 2001
- [Panoscan] Panoscan Inc., *Panoscan Panoramic Camera for High Speed Digital Capture*, <http://www.panoscan.com/>, 2009
- [Pollefeys00a] Pollefeys, M., *Tutorial on 3D Modeling from Images*, Tutorial presented at European Conference on Computer Vision, 2000
- [Pollefeys04a] Pollefeys, M. and Gool, L.V. and Vergauwen, M. and Verbiest, F. and Cornelis, K. and Tops, J., *Visual Modeling with a Hand-Held Camera*, International Journal of Computer Vision (IJCV), vol. 59, pp. 207-232, 2004
- [Ponce96a] Ponce, J. and Genc, Y., *Epipolar geometry and linear subspace methods: A new approach to weak calibration*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 776-781, 1996
- [Prados04a] Prados, E. and Faugeras, O., *A rigorous and realistic Shape From Shading method and some of its applications*, Rapport de Recherche, Institut National de Recherche en Informatique et en Automatique, 2004
- [Prados06a] Prados, E. and Faugeras, O., *Rôle clé de la modélisation en Shape From Shading*, Reconnaissance des Formes et Intelligence Artificielle (RFIA), pp. , 2006
- [Press92a] W.H. Press and S.A. Teukolsky and W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C : The art of scientific computing*, Cambridge University Press, 1992
- [Ragot09a] Ragot, N., *Conception d'un capteur de stéréovision omnidirectionnelle : architecture, étalonnage et applications à la reconstruction 3D*, Thèse de doctorat, Université de Rouen, 2009

- [Ramalingam05b] Ramalingam, S. and Sturm, P. and Lodha, S.K., *Towards Complete Generic Camera Calibration*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1093-1098, 2005
- [Ramalingam06a] Ramalingam, S., *Generic Imaging Models: Calibration and 3D Reconstruction Algorithms*, Thèse de doctorat, Institut National Polytechnique de Grenoble, 2006
- [Rees70a] Rees, D.W., *Panoramic Television Viewing System*, U.S. Patent 3505465, 1970
- [Rodriguez03a] Rodriguez, T. and Sturm, P. and Wilczkowiak, M. and Bartoli, A. and Personnaz, M. and Guilbert, N. and Kahl, F. and Johansson, M. and Heyden, A. and Menendez, J.M. and Ronda, J.I. and Jaureguizar, F., *VISIRE. Photorealistic 3D reconstruction from video sequences*, IEEE International Conference on Image Processing (ICIP), pp. 705-708, 2003
- [Rossi09a] Rossi, R. and Savatier, X. and Ertaud, J.Y. and Mazari, B., *Real-Time 3D Reconstruction for Mobile Robot Using Catadioptric Cameras*, IEEE International Workshop on RObotic and Sensors Environments (ROSE), pp. , 2009
- [Scaramuzza] Scaramuzza, D., *OcamCalib: Omnidirectional Camera Calibration Toolbox for Matlab*, [http://asl.epfl.ch/~scaramuz/research/Davide\\_Scaramuzza\\_files/Research/OcamCalib\\_Tutorial.htm](http://asl.epfl.ch/~scaramuz/research/Davide_Scaramuzza_files/Research/OcamCalib_Tutorial.htm), 2009
- [Scaramuzza06a] Scaramuzza, D. and Martinelli, A. and Siegwart, R., *A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion*, IEEE International Conference on Computer Vision Systems (ICVS), pp. , 2006
- [Scaramuzza06b] Scaramuzza, D. and Martinelli, A. and Siegwart, R., *A Toolbox for Easily Calibrating Omnidirectional Cameras*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5695-5701, 2006
- [Seitz97a] Seitz, S.M. and Dyer, C.R., *Photorealistic Scene Reconstruction by Voxel Coloring*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1067-1073, 1997
- [Sturm04a] Sturm, P. and Ramalingam, S., *A Generic Concept for Camera Calibration*, European Conference on Computer Vision (ECCV 2004), pp. 1-13, 2004
- [Svoboda98a] Svoboda, T. and Pajdla, T. and Hlavac, V., *Epipolar Geometry for Panoramic Cameras*, European Conference on Computer Vision (ECCV), pp. 218-232, 1998
- [Svoboda99a] Svoboda, T., *Central Panoramic Cameras, Design, Geometry, Egomotion*, Thèse de doctorat, Czech Technical University, 1999
- [Tarel95a] Tarel, J.P., *Calibration radiométrique de caméra*, Rapport de Recherche, Institut National de Recherche en Informatique et en Automatique, 1995
- [Thrun00a] Thrun, S. and Fox, D. and Burgard, W. and Dellaert, F., *Robust Monte Carlo Localization for Mobile Robots*, Artificial Intelligence, vol. 128, pp. 99-141, 2000
- [Thrun02a] Thrun, S., *Exploring Artificial Intelligence in the New Millenium*, Chapitre Robotic mapping: A survey du livre , Ed. Lakemeyer, G. and Nebel, B., pp. 1-29, 2002
- [Triggs00b] Triggs, B. and McLauchlan, P. and Hartley, R. and Fitzgibbon, A., *Bundle Adjustment - A Modern Synthesis*, Vision Algorithms: Theory and Practice, vol. 1883, pp. 298-372, 2000
- [Vandeportaele06a] Vandeportaele, B. and Cattoen, M. and Marthon, P., *A Fast Detector of Line Images Acquired by an Uncalibrated Paracatadioptric Camera*, International Conference on Pattern Recognition (ICPR), pp. 1042-1045, 2006
- [Vasseur04a] Vasseur, P. and Mouaddib, E.M., *Central catadioptric line extraction*, British Machine Vision Conference (BMVC), pp. 57-66, 2004
- [Wang00a] Wang, W. and Tsui, H.T., *A SVD decomposition of essential matrix with eight solutions for the relative positions of two perspective cameras*, International

- Conference on Pattern Recognition (ICPR), pp. 362-365, 2000
- [Wang92a] Wang, H. and Brady, M., *A Structure-From-Motion algorithm for robot vehicle guidance*, Intelligent Vehicles (IV), pp. 30-35, 1992
- [Wei00a] Wei, S.K., *Stereo Matching of Catadioptric Panoramic Images*, Rapport Technique, Czech Technical University, 2000
- [Werth00a] Werth, P. and Scherer, S., *A novel bidirectional framework for control and refinement of areabased correlation techniques*, International Conference on Pattern Recognition (ICPR), pp. 730-733, 2000
- [Woodham80a] Woodham, R.J., *Photometric Method for Determining Surface Orientation from Multiple Images*, Optical Engineering, vol. 19, pp. 139-144, 1980
- [Yachida98a] Yachida, M., *Omnidirectional sensing and combined multiple sensing*, IEEE/ATR Workshop on Computer Vision for Virtual Reality Based Human Communications (CVVRHC), pp. 20-27, 1998
- [Yagi90a] Yagi, Y. and Kawato, S., *Panorama scene analysis with conic projection*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 181-187, 1990
- [Yagi91a] Yagi, Y. and Yachida, M., *Real-time generation of environmental map and obstacle avoidance using omnidirectional image sensor with conic mirror*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 160-165, 1991
- [Yagi99a] Yagi, Y., *Omnidirectional Sensing and Its Applications*, IEICE Transactions on Information and Systems, vol. E82-D, pp. 568-579, 1999
- [Yamazawa00a] Yamazawa, K. and Yagi, Y. and Yachida, M., *3D line segment reconstruction by using HyperOmni vision and omnidirectional Hough transforming*, International Conference on Pattern Recognition (ICPR), pp. 483-486, 2000
- [Yamazawa93a] Yamazawa, K. and Yagi, Y. and Yachida, M., *Omnidirectional Imaging with Hyperboloidal Projection*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1029-1034, 1993
- [Yuen90a] Yuen, H.K. and Princen, J. and Illingworth, J. and Kittler, J., *Comparative study of Hough transform methods for circle finding*, Image and Vision Computing Journal, vol. 8, pp. 71-77, 1990
- [Zhang00a] Zhang, Z., *A Flexible New Technique for Camera Calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 1330-1334, 2000
- [Zhang93a] Zhang, Z., *Le problème de la mise en correspondance : L'état de l'art*, Institut National de Recherche en Informatique et en Automatique, 1993
- [Zhang96a] Zhang, Z., *Motion of a Stereo Rig: Strong, Weak and Self Calibration*, Lecture Notes in Computer Science, vol. 1035, pp. 241-253, 1996
- [Zhang97a] Zhang, Z., *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting*, Image and Vision Computing Journal, vol. 15, pp. 59-76, 1997
- [Zhang98a] Zhang, Z., *A Flexible New Technique for Camera Calibration*, Rapport de Recherche, Microsoft Research, 1998
- [Zhang98b] Zhang, Z., *Determining the Epipolar Geometry and its Uncertainty: A Review*, International Journal of Computer Vision (IJCV), vol. 27, pp. 161-198, 1998
- [Zhang99a] Zhang, Z., *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*, International Conference on Computer Vision (ICCV), pp. 666-673, 1999

# Annexes

## Sommaire

---

Représentation des rotations par les quaternions.....	171
Algorithme de Levenberg-Marquardt.....	175
RANSAC.....	177
Jacobienne de la fonction de projection.....	179
Missions opérationnelles.....	183

## Annexe A

# Représentation des rotations par les quaternions

### Introduction

Les quaternions sont une extension des nombres complexes. Ils ont été proposés par Hamilton en 1843. Un quaternion est composé d'une partie réelle et de trois parties imaginaires :

$$\mathbf{q} = a + b \cdot i + c \cdot j + d \cdot k \text{ avec } i \cdot i = -1, j \cdot j = -1, k \cdot k = -1$$

La multiplication de deux de ces nombres se comporte comme le produit vectoriel de vecteurs unitaires orthogonaux :

$$\begin{aligned} i \cdot j &= -j \cdot i = k \\ j \cdot k &= -k \cdot j = i \\ k \cdot i &= -i \cdot k = j \end{aligned}$$

Les quaternions unitaires sont une façon de représenter une rotation en évitant les singularités et les discontinuités.

### Notation

Les quaternions peuvent être représentés de différentes façons :

- comme une combinaison linéaire de  $1, i, j, k$  :  $\mathbf{q} = q_w + q_x \cdot i + q_y \cdot j + q_z \cdot k$  avec  $q_w, q_x, q_y$  et  $q_z \in \mathbb{R}$ ,
- comme un vecteur de 4 coefficients :  $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$  avec  $q_w, q_x, q_y$  et  $q_z \in \mathbb{R}$ ,
- comme un scalaire pour le coefficient de la partie réelle, et comme un vecteur pour les coefficients de la partie imaginaire :  $\mathbf{q} = (q_w, \mathbf{v})$  avec  $q_w \in \mathbb{R}$  et  $\mathbf{v} = [q_x \ q_y \ q_z]^T$ .

### Propriétés des quaternions

La **norme** d'un quaternion est définie par :  $\|\mathbf{q}\| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}$ .

Le **conjugué** d'un quaternion  $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$ , noté  $\mathbf{q}'$  est :  $\mathbf{q}' = [q_w \ -q_x \ -q_y \ -q_z]^T$ .

### Algèbre des quaternions

Soient  $\mathbf{q}_1 = [q_{w1} \ q_{x1} \ q_{y1} \ q_{z1}]^T$  et  $\mathbf{q}_2 = [q_{w2} \ q_{x2} \ q_{y2} \ q_{z2}]^T$  deux quaternions unitaires,

**l'addition** et la **soustraction** de ces deux quaternions sont données par :

$$\mathbf{q}_1 + \mathbf{q}_2 = (q_{w1} + q_{w2}) + (q_{x1} + q_{x2}) \cdot \mathbf{i} + (q_{y1} + q_{y2}) \cdot \mathbf{j} + (q_{z1} + q_{z2}) \cdot \mathbf{k}$$

$$\mathbf{q}_1 - \mathbf{q}_2 = (q_{w1} - q_{w2}) + (q_{x1} - q_{x2}) \cdot \mathbf{i} + (q_{y1} - q_{y2}) \cdot \mathbf{j} + (q_{z1} - q_{z2}) \cdot \mathbf{k}$$

Soient  $\mathbf{q}_1 = [q_{w1} \ q_{x1} \ q_{y1} \ q_{z1}]^T$  et  $\mathbf{q}_2 = [q_{w2} \ q_{x2} \ q_{y2} \ q_{z2}]^T$  deux quaternions unitaires, la

**multiplication** de  $\mathbf{q}_1$  par  $\mathbf{q}_2$  est donnée par :

$$\begin{aligned} \mathbf{q}_1 \cdot \mathbf{q}_2 &= (q_{w1} + q_{x1} \cdot \mathbf{i} + q_{y1} \cdot \mathbf{j} + q_{z1} \cdot \mathbf{k}) \cdot (q_{w2} + q_{x2} \cdot \mathbf{i} + q_{y2} \cdot \mathbf{j} + q_{z2} \cdot \mathbf{k}) \\ &= (q_{w1} \cdot q_{w2} - q_{x1} \cdot q_{x2} - q_{y1} \cdot q_{y2} - q_{z1} \cdot q_{z2}) + \dots \\ &\quad (q_{w1} \cdot q_{x2} + q_{x1} \cdot q_{w2} + q_{y1} \cdot q_{z2} - q_{z1} \cdot q_{y2}) \cdot \mathbf{i} + \dots \\ &\quad (q_{w1} \cdot q_{y2} - q_{x1} \cdot q_{z2} + q_{y1} \cdot q_{w2} + q_{z1} \cdot q_{x2}) \cdot \mathbf{j} + \dots \\ &\quad (q_{w1} \cdot q_{z2} + q_{x1} \cdot q_{y2} - q_{y1} \cdot q_{x2} + q_{z1} \cdot q_{w2}) \cdot \mathbf{k} \end{aligned}$$

Attention : La multiplication de quaternions n'est pas commutative.

### Représentation des rotations avec des quaternions

La rotation d'un angle  $\theta$  autour d'un vecteur unitaire  $\mathbf{u} = [x \ y \ z]^T$  est donnée par le quaternion unitaire  $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$ , tel que :

$$\mathbf{q} = \left[ \cos \frac{\theta}{2} \quad x \cdot \sin \frac{\theta}{2} \quad y \cdot \sin \frac{\theta}{2} \quad z \cdot \sin \frac{\theta}{2} \right]^T$$

### Passage d'un quaternion à une matrice de rotation

Soit une rotation caractérisée par un quaternion unitaire  $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$ . La matrice de rotation  $\mathbf{R}$  correspondante est donnée par :

$$\mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 + q_y^2 - q_x^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 + q_z^2 - q_x^2 - q_y^2 \end{bmatrix}$$

### Passage d'un quaternion à un vecteur et un angle

Soit un quaternion unitaire  $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$ . L'angle de rotation  $\theta$  autour de l'axe  $\mathbf{u}$  est obtenu par :

$$\theta = 2 \cdot \arccos(q_w)$$

L'axe  $\mathbf{u} = [u_x \ u_y \ u_z]^T$  autour duquel la rotation est effectuée est quant à lui obtenu par :

$$u_x = x / \sin \frac{\theta}{2}, \quad u_y = y / \sin \frac{\theta}{2}, \quad u_z = z / \sin \frac{\theta}{2}$$

à condition que l'angle  $\theta$  ne soit pas nul. Dans le cas contraire, l'axe importe peu puisqu'il n'y a pas de rotation.

### Combinaison de rotations

La nécessité de combiner des rotations est très fréquente, par exemple pour déterminer l'orientation finale du capteur à partir des orientations aux instants précédents.

Les quaternions permettent de combiner les rotations de façon beaucoup moins coûteuse que par la méthode matricielle. En effet, soient  $\mathbf{q}_1$  et  $\mathbf{q}_2$  deux quaternions unitaires représentant

deux rotations. Si l'on veut appliquer la rotation  $\mathbf{q}_1$  en premier puis la rotation  $\mathbf{q}_2$ , la rotation résultante  $\mathbf{q}_3$  est obtenue en multipliant ces deux quaternions :

$$\mathbf{q}_3 = \mathbf{q}_2 \cdot \mathbf{q}_1$$

### Rotation inverse

Soit  $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$  un quaternion caractérisant une rotation d'angle  $\theta$  autour du vecteur  $\mathbf{u} = [x \ y \ z]^T$ . La rotation inverse est obtenue en calculant le conjugué du quaternion  $\mathbf{q}$  noté  $\mathbf{q}'$  :

$$\mathbf{q}' = [q_w \ -q_x \ -q_y \ -q_z]^T$$

Ce résultat peut être justifié de deux façons :

- en gardant le même angle, la conjugaison du quaternion implique l'inversion du vecteur de rotation (il pointe dans la direction opposée),
- en utilisant une rotation d'un angle opposé, le cosinus reste le même alors que le sinus devient négatif.

## Annexe B

### Algorithme de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt est une technique itérative de minimisation d'une fonction non-linéaire et dépendant de plusieurs variables [Levenberg44a] [Marquardt63a]. Cet algorithme, devenu un standard en vision par ordinateur, peut être interprété comme une combinaison des algorithmes de Gauss-Newton et de Descente du Gradient [Press92a]. Lorsque la solution est loin de l'estimée, l'algorithme se comporte comme la méthode de Descente du Gradient et permet de se rapprocher rapidement de la solution. Au voisinage du minimum, il se comporte alors comme la méthode de Gauss-Newton qui a l'avantage de converger plus rapidement que la descente du gradient.

Soit  $f$  la fonction liant le vecteur de paramètres  $\mathbf{P} \in \mathbb{R}^n$  au vecteur de mesures  $\mathbf{X} \in \mathbb{R}^m$  :

$$\mathbf{X} = f(\mathbf{P}) \quad (68)$$

Le but de la minimisation est de trouver le vecteur optimal, noté  $\hat{\mathbf{P}}$ , qui satisfait au mieux la relation (68). Ce problème revient donc à calculer le vecteur  $\hat{\mathbf{P}}$  qui satisfait  $\mathbf{X} = f(\hat{\mathbf{P}}) - \mathbf{e}$  pour lequel  $\|\mathbf{e}\|$  est minimale.

Pour résoudre le cas où  $f$  n'est pas linéaire, il faut partir d'une solution initiale  $\mathbf{P}_0$  et affiner l'estimée en faisant l'hypothèse que  $f$  est localement linéaire. Dans ce cas, pour un petit  $\Delta$ ,  $f(\mathbf{P}_0 + \Delta)$  peut être approximée par une série de Taylor :

$$f(\mathbf{P}_0 + \Delta) \approx f(\mathbf{P}_0) + \mathbf{J}\Delta \quad (69)$$

où  $\mathbf{J}$  est la matrice jacobienne  $\frac{\partial f(\mathbf{P})}{\partial \mathbf{P}}$ .

Les algorithmes d'optimisation non linéaire sont itératifs : à chaque itération  $i$ , le vecteur de paramètres  $\mathbf{P}_{i+1} = \mathbf{P}_i + \Delta_i$  est calculé de sorte à minimiser l'erreur entre les mesures et les prédictions :

$$\|\mathbf{X} - f(\mathbf{P}_i + \Delta_i)\| \approx \|\mathbf{X} - f(\mathbf{P}_i) - \mathbf{J}\Delta_i\| = \|\mathbf{e} + \mathbf{J}\Delta\| \quad (70)$$

Le vecteur de correction  $\Delta$  est obtenu, dans le cas de l'algorithme de Gauss-Newton, par résolution de l'équation (71), appelée équation normale :

$$\mathbf{J}^T \mathbf{J} \Delta = -\mathbf{J}^T \mathbf{e} \quad (71)$$

L'algorithme de Levenberg-Marquardt consiste à résoudre une variante de l'équation (71),



appelée équation normale augmentée :

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \Delta = -\mathbf{J}^T \mathbf{e} \quad (72)$$

où  $\lambda$  est un réel, appelé *damping term*, qui varie d'itération en itération et  $\mathbf{I}$  est la matrice identité.

Le *damping term*  $\lambda$  est initialisé empiriquement à 0,001 et varie en fonction de l'évolution de l'erreur :

- si le vecteur  $\mathbf{P}_{i+1}$  calculé grâce à l'équation (72) conduit à une réduction de l'erreur  $\|\mathbf{e}\|$ , alors la mise à jour est validée et l'itération suivante utilisera un *damping term* divisé par 10,
- si le vecteur  $\mathbf{P}_{i+1}$  calculé grâce à l'équation (72) conduit à une augmentation de l'erreur  $\|\mathbf{e}\|$ , le vecteur de paramètres n'est pas modifié et le *damping term* est multiplié par 10 jusqu'à trouver un vecteur de paramètres qui réduise cette erreur.

L'algorithme de Levenberg-Marquardt peut être expliqué en étudiant le comportement de l'équation normale augmentée (72) en fonction de la valeur du *damping term*  $\lambda$  :

- si  $\lambda$  est faible, le terme  $\lambda \mathbf{I}$  devient négligeable et l'équation normale augmentée peut être approximée par  $\mathbf{J}^T \mathbf{J} \Delta = -\mathbf{J}^T \mathbf{e}$ , ce qui correspond à l'équation normale de Gauss-Newton,
- si  $\lambda$  est grand, l'équation normale augmentée peut être approximée par  $\lambda \Delta = -\mathbf{J}^T \mathbf{e}$  qui correspond à une itération de type descente du gradient.

L'algorithme de Levenberg-Marquardt, présenté dans cette annexe, peut être résumé par le pseudo-code suivant :

```

i = 0
λ=0,001
Calculer ‖e(P₀)‖
Tant que ( i < MAX_ITERATIONS et ‖e(Pᵢ)‖ > seuil )
    Résoudre l'équation normale augmentée :
        (JᵀJ + λI)Δ = -Jᵀe pour Δ évalué à Pᵢ
    Évaluer le nouveau vecteur de paramètres Pᵢ₊₁ = Pᵢ + Δ :
        Si ‖e(Pᵢ₊₁)‖ ≥ ‖e(Pᵢ)‖ alors λ = 10 λ
        Si ‖e(Pᵢ₊₁)‖ < ‖e(Pᵢ)‖ alors Pᵢ₊₁ = Pᵢ + Δ , λ = λ/10
    i = i + 1

```

Figure 124: Pseudo-code de l'algorithme de Levenberg-Marquardt.

## Annexe C

### RANSAC

Le RANSAC (RANdom SAmple Consensus) est un algorithme d'estimation robuste publié pour la première fois en 1981 par Fischler et Bolles [Fischler81a]. Cet algorithme itératif permet d'estimer les paramètres d'un modèle à partir d'un jeu de données contenant des données erronées (les *outliers*).

Le principe du RANSAC peut être illustré par un cas simple : l'estimation de l'équation d'une droite à partir d'un ensemble de points. La figure 125 représente le jeu de points pour lequel l'équation de la droite doit être estimée. Ce jeu de points contient quelques *outliers*, qui sont représentés en rouge sur la figure 125.

Si l'équation de la droite est estimée au sens des moindres carrés avec la totalité du jeu de points, celle-ci ne correspondra pas aux données car les *outliers* ont une forte influence par cette méthode. La droite obtenue par cette méthode est représentée sur la figure 125a.

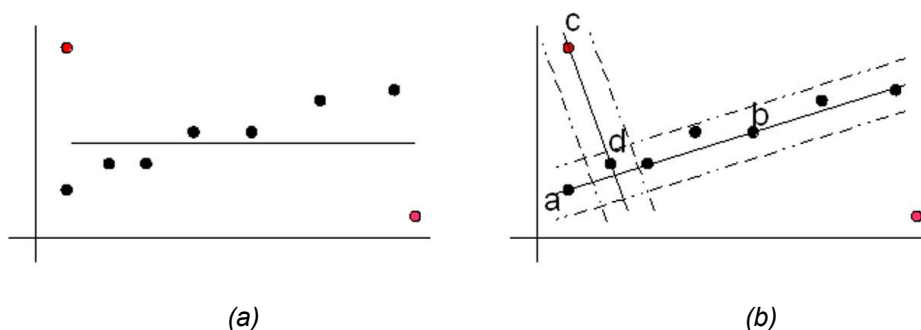


Figure 125: Estimation de l'équation d'une droite par le RANSAC.

(a) Équation estimée par les moindres carrés – (b) Équations estimées par le RANSAC avec et sans outlier

Le principe du RANSAC, pour l'estimation de l'équation d'une droite, est le suivant :

- deux points sont tirés aléatoirement et sont considérés comme des *inliers* hypothétiques,
- l'équation de la droite est estimée à partir de ces points,
- les autres points sont testés pour voir s'ils appartiennent à la droite. Si la distance minimale entre le point et la droite est faible, le point est alors considéré comme bon et est ajouté aux *inliers*,
- si le nombre d'*inliers* est grand, c'est que l'équation de la droite correspond bien aux

données et cette équation est alors recalculée avec tous les *inliers*.

Sur l'exemple de la figure 125, si les points **c** et **d** sont tirés, l'équation de la droite sera fautive car le point **c** est un *outlier*. Les autres points seront très éloignés de cette droite et le nombre d'*inliers* sera très réduit (2 dans cet exemple). A contrario, les points **a** et **b** sont tirés, la grande majorité des points sera proche de la droite estimée et les *outliers* (les deux points rouges) seront rejetés.

Le principe du RANSAC, dont le pseudo-code est présenté sur la figure 126, peut être transposé à de nombreux problèmes consistant à estimer un modèle à partir d'un jeu de données. Le nombre de points à tirer aléatoirement doit alors être le nombre de points minimal nécessaire pour pouvoir calculer le modèle.

```
k = 0
Tant que k < MAX_ITERATIONS
    maybe_inliers = n points sélectionnés aléatoirement
    maybe_model = modèle estimé avec ces n points
    consensus_set = maybe_inliers
    Pour chaque point pas dans maybe_inliers
        Si le point respecte le modèle avec une erreur plus
            petite que MAX_ERROR alors ajouter le point à
                consensus_set
    Si nb_points(consensus_set) > nb_points(meilleur_consensus)
        alors best_model = maybe_model, best_consensus =
            consensus_set.
    k++
Ré-estimation du modèle avec tous les points de best_consensus.
```

Figure 126: Pseudo-code pour l'estimation robuste de la matrice essentielle.

## Annexe D

### Jacobienne de la fonction de projection

Rappel de la fonction de projection :

$$P(\mathbf{V}, \mathbf{X}) = K \circ D \circ H \circ W(\mathbf{V}, \mathbf{X})$$

Changement de repère :

Transformation :

$$W(\mathbf{X}, \mathbf{V}_1) = \mathbf{R}(\mathbf{Q}') \mathbf{X} + \mathbf{T}$$

$$\mathbf{Q}' \begin{pmatrix} q_w \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{bmatrix} q_w' \\ q_x' \\ q_y' \\ q_z' \end{bmatrix} = \begin{bmatrix} \frac{q_w}{\sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}} \\ \frac{q_x}{\sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}} \\ \frac{q_y}{\sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}} \\ \frac{q_z}{\sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}} \end{bmatrix}$$

$$\mathbf{R}(\mathbf{Q}') = \begin{bmatrix} q_w'^2 + q_x'^2 - q_y'^2 - q_z'^2 & 2(q_x'q_y' - q_w'q_z') & 2(q_x'q_z' + q_w'q_y') \\ 2(q_x'q_y' + q_w'q_z') & q_w'^2 + q_y'^2 - q_x'^2 - q_z'^2 & 2(q_y'q_z' - q_w'q_x') \\ 2(q_x'q_z' - q_w'q_y') & 2(q_y'q_z' + q_w'q_x') & q_w'^2 + q_z'^2 - q_x'^2 - q_y'^2 \end{bmatrix}$$

Jacobienne :

$$\frac{\partial W}{\partial \mathbf{Q}'} = \frac{\partial W}{\partial \mathbf{Q}'} \frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}}$$

$$\frac{\partial W}{\partial \mathbf{Q}'} = \begin{bmatrix} \frac{\partial W}{\partial q_w'} & \frac{\partial W}{\partial q_x'} & \frac{\partial W}{\partial q_y'} & \frac{\partial W}{\partial q_z'} \end{bmatrix}$$

$$\frac{\partial W}{\partial q_w'} = 2 \begin{bmatrix} q_w'x - q_z'y + q_y'z \\ q_z'x + q_w'y - q_x'z \\ -q_y'x + q_x'y + q_w'z \end{bmatrix}$$

$$\frac{\partial W}{\partial q_x'} = 2 \begin{bmatrix} q_x'x + q_y'y + q_z'z \\ q_y'x - q_x'y - q_w'z \\ q_z'x + q_w'y - q_x'z \end{bmatrix}$$

$$\frac{\partial W}{\partial q_y'} = 2 \begin{bmatrix} -q_y'x + q_x'y + q_w'z \\ q_x'x + q_y'y + q_z'z \\ -q_w'x + q_z'y - q_y'z \end{bmatrix}$$

$$\frac{\partial W}{\partial q_z'} = 2 \begin{bmatrix} -q_z'x - q_w'y + q_x'z \\ q_w'x - q_z'y + q_y'z \\ q_x'x + q_y'y + q_z'z \end{bmatrix}$$

$$\frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}} = \frac{1}{\sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}} \begin{bmatrix} q_x^2 + q_y^2 + q_z^2 & -q_w q_x & -q_w q_y & -q_w q_z \\ -q_w q_x & q_w^2 + q_y^2 + q_z^2 & -q_x q_y & -q_x q_z \\ -q_w q_y & -q_x q_y & q_w^2 + q_x^2 + q_z^2 & -q_y q_z \\ -q_w q_z & -q_x q_z & -q_y q_z & q_w^2 + q_x^2 + q_y^2 \end{bmatrix}$$

$$\frac{\partial W}{\partial \mathbf{T}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial W}{\partial \mathbf{V}_1} = \begin{bmatrix} \frac{\partial W}{\partial \mathbf{Q}} & \frac{\partial W}{\partial \mathbf{T}} \end{bmatrix}$$

$$\frac{\partial W}{\partial \mathbf{X}} = \begin{bmatrix} q_w'^2 + q_x'^2 - q_y'^2 - q_z'^2 & 2(q_x'q_y' - q_w'q_z') & 2(q_x'q_z' + q_w'q_y') \\ 2(q_x'q_y' + q_w'q_z') & q_w'^2 + q_y'^2 - q_x'^2 - q_z'^2 & 2(q_y'q_z' - q_w'q_x') \\ 2(q_x'q_z' - q_w'q_y') & 2(q_y'q_z' + q_w'q_x') & q_w'^2 + q_z'^2 - q_x'^2 - q_y'^2 \end{bmatrix}$$

### Transformation du miroir :

Transformation :

$$H = h \circ S$$

$$S \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \begin{bmatrix} \frac{X}{\sqrt{X^2 + Y^2 + Z^2}} \\ \frac{Y}{\sqrt{X^2 + Y^2 + Z^2}} \\ \frac{Z}{\sqrt{X^2 + Y^2 + Z^2}} \end{bmatrix}$$

$$h \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{X_s}{Z_s + \xi} \\ \frac{Y_s}{Z_s + \xi} \\ 1 \end{bmatrix}$$

Jacobienne :

$$\frac{\partial H}{\partial W} = \frac{\partial h \circ S}{\partial W} = \frac{\partial h}{\partial S} \frac{\partial S}{\partial W}$$

$$\frac{\partial S}{\partial W} = \frac{1}{r^2} \begin{bmatrix} r - \frac{X^2}{r} & -\frac{XY}{r} & -\frac{XZ}{r} \\ -\frac{XY}{r} & r - \frac{Y^2}{r} & -\frac{YZ}{r} \\ -\frac{XZ}{r} & -\frac{YZ}{r} & r - \frac{Z^2}{r} \end{bmatrix}, \text{ avec } r = \sqrt{X^2 + Y^2 + Z^2}$$

$$\frac{\partial h}{\partial S} = \frac{1}{(Z_s + \xi)^2} \begin{bmatrix} Z_s + \xi & 0 & -X_s \\ 0 & Z_s + \xi & -Y_s \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial H}{\partial W} = \frac{1}{r(Z + \xi r)^2} \begin{bmatrix} rZ + \xi(Y^2 + Z^2) & -\xi XY & X(-\xi Z - r) \\ -\xi XY & rZ + \xi(X^2 + Z^2) & Y(-\xi Z - r) \end{bmatrix}$$

### Distorsions :

Transformation :

$$D \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{bmatrix} x(1 + k_1 \rho^2 + k_2 \rho^4 + k_3 \rho^6) + 2k_3 x y + k_4(\rho^2 + 2x^2) \\ y(1 + k_1 \rho^2 + k_2 \rho^4 + k_3 \rho^6) + 2k_4 x y + k_3(\rho^2 + 2y^2) \end{bmatrix}, \text{ avec } \rho = \sqrt{x^2 + y^2}$$

Jacobienne :

$$\frac{\partial D}{\partial H} = \begin{bmatrix} 1 + k_1(\rho^2 + 2x^2) + k_2 \rho^2(\rho^2 + 4x^2) + k_3 \rho^4(\rho^2 + 6x^2) + 2k_3 y + 6k_4 x & \dots \\ 2k_1 x y + 4k_2 \rho^2 x y + 6k_3 \rho^4 x y + 2k_3 x + 2k_4 y & \dots \\ 2k_1 x y + 4k_2 \rho^2 x y + 6k_3 \rho^4 x y + 2k_3 x + 2k_4 y & \dots \\ 1 + k_1(\rho^2 + 2y^2) + k_2 \rho^2(\rho^2 + 4y^2) + k_3 \rho^4(\rho^2 + 6y^2) + 6k_3 y + 2k_4 x & \dots \end{bmatrix}$$

### Projection perspective :

Transformation :

$$\mathbf{p} = \mathbf{K} \mathbf{m}_d = \begin{bmatrix} \gamma_u & \gamma_u \alpha & u_0 \\ 0 & \gamma_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

Jacobienne :

$$\frac{\partial K}{\partial D} = \begin{bmatrix} \gamma_u & \gamma_u \alpha \\ 0 & \gamma_v \end{bmatrix}$$

## **Annexe E**

### **Missions opérationnelles**

#### **Mission 1 : Préparation de la progression en milieu urbain**

##### **Objectif de la mission**

Préparation de la progression en milieu urbain : relevé d'architecture le long de la façade d'un bâtiment. La plate forme mobile progressant dans une rue le long d'une façade devra être capable de repérer les ouvertures et de dimensionner ces dernières (portes, fenêtres, garages, soupiraux).

Le système de vision devra pouvoir dimensionner l'environnement (exemple : une porte ou une fenêtre). Les dimensions mesurées seront alors comparées avec les dimensions réelles. L'écart entre la mesure et la réalité sera inférieure à 5% pour les objets d'une taille supérieure à 1 mètre situés à une distance inférieure à 10 mètres. La distance d'un objet situé entre 10 et 50 mètres devra être estimée avec une précision meilleure que 20%.

Le système devra en outre être capable de dimensionner et de positionner relativement au porteur un obstacle fixe du volume d'un cube opaque de 50 cm de côté situé à 3 mètres de distance.

##### **Lieu**

L'acquisition a eu lieu devant la façade de l'IRSEEM. Cet environnement est très riche car il contient à la fois des éléments très structurés (les bâtiments) et des éléments naturels (arbres, pelouse). Il contient également des surfaces transparentes et réfléchissantes (fenêtres, voitures, etc) qui posent des problèmes aux systèmes de vision, ce qui nous permettra d'évaluer le comportement des algorithmes face à ces difficultés. Une séquence d'images a été enregistrée lors du déplacement du robot dans cet environnement.



Figure 127: Environnement de la mission 1.

### Images omnidirectionnelles

La figure ci-dessous présente des images extraites de la séquence qui est utilisée pour cette mission :

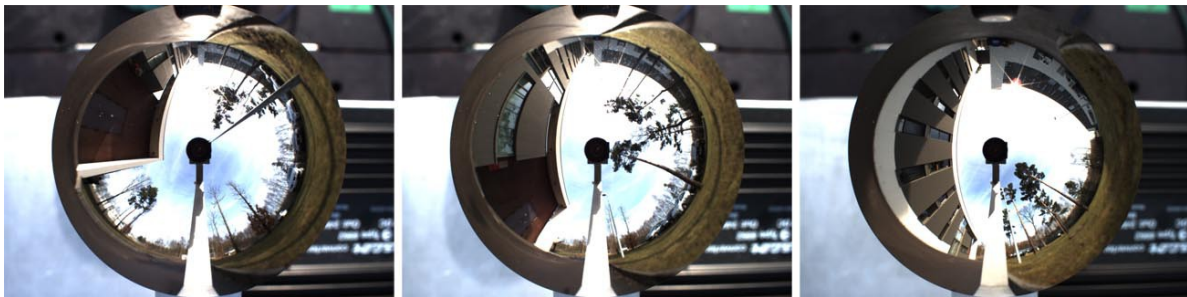


Figure 128: Images omnidirectionnelles de la séquence utilisée pour la mission 1.

### Algorithmes utilisés et résultats

Les algorithmes utilisés pour répondre à cette mission sont :

- l'ajustement de faisceaux pour estimer la trajectoire du robot (les résultats sont présentés dans le chapitre 4 de ce mémoire),
- la détection et la triangulation de droites pour identifier les ouvertures (*cf* chapitre 3),
- la triangulation de points pour vérifier les dimensions (*cf* chapitre 3).

Les résultats sur la précision du capteur sont présentés dans le corps du mémoire (*cf* section 3.2.3.2). Le dimensionnement et le positionnement du cube par rapport au porteur a été validé lors de la mission 2.

La séquence de la mission 1 a également été utilisée pour valider « qualitativement » les données issues du calibrage, en réalisant une reconstruction 3D sous forme de plans. En effet, si les données du calibrage avaient été erronées, le placage de la texture n'aurait pas collé au modèle et il y aurait eu des aberrations sur le modèle 3D présenté sur la figure 129.



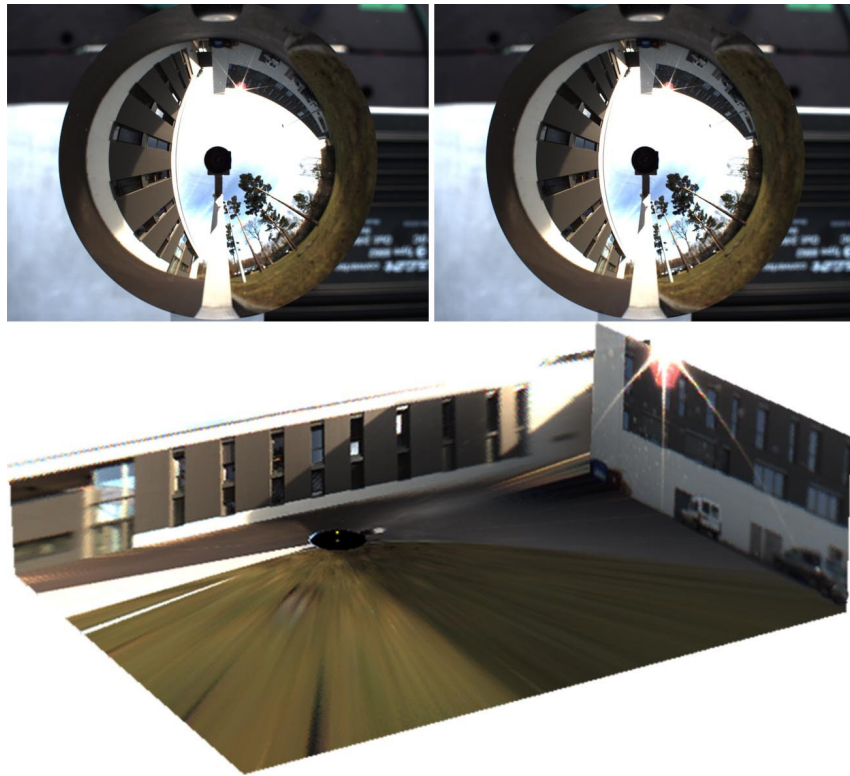


Figure 129: Validation qualitative des données de calibration.

## **Mission 2 : Préparation de la progression en milieu fermé**

### **Objectif de la mission**

Préparation de la progression en milieu fermé (bâtiment) : relever la configuration (portes, fenêtres) de l'intérieur d'un bâtiment, le long d'un couloir.

Le système de vision devra pouvoir dimensionner l'environnement (exemple : une porte ou une fenêtre). Les dimensions mesurées seront alors comparées avec les dimensions réelles. L'écart entre la mesure et la réalité sera inférieure à 5% pour les objets d'une taille supérieure à 1 mètre situés à une distance inférieure à 10 mètres. La distance d'un objet situé entre 10 et 50 mètres devra être estimée avec une précision meilleure que 20 %.

Le système devra en outre être capable de dimensionner et de positionner relativement au porteur un obstacle fixe du volume d'un cube opaque de 50 cm de côté situé à 3 mètres de distance.

### **Lieu**

Cette mission se déroule dans un couloir de l'IRSEEM qui est assez représentatif de la majorité des couloirs. La principale difficulté dans ce type d'environnement est l'absence de texture au niveau des murs. Une séquence d'images a été enregistrée lors du déplacement du robot dans cet environnement.



Figure 130: Environnement de la mission 2.

### Images omnidirectionnelles

La figure ci-dessous présente des images extraites de la séquence qui est utilisée pour cette mission :

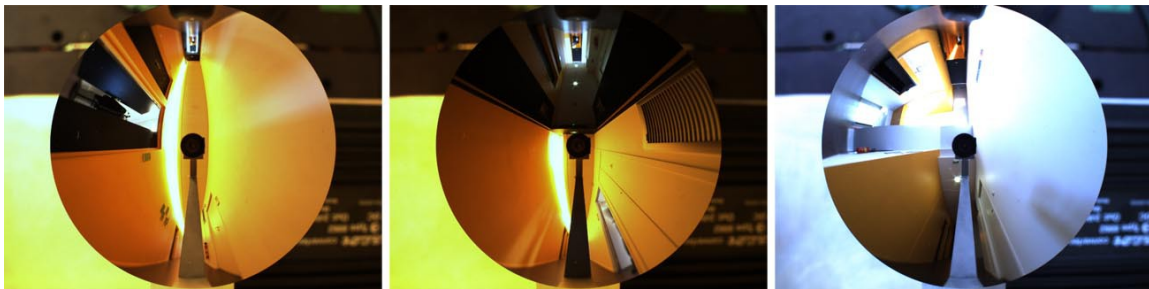


Figure 131: Images omnidirectionnelles de la séquence utilisée pour la mission 2.

### Algorithmes utilisés et résultats

Les algorithmes utilisés pour répondre à cette mission sont :

- la détection et la triangulation de droites pour identifier et positionner les murs par rapport au porteur (*cf* chapitre 3),
- la triangulation de points pour estimer les dimensions (*cf* chapitre 3).

De plus, cette mission a été utilisée pour valider le dimensionnement et le positionnement du cube. Un carton d'environ 50 cm de côté a été placé à une distance de 3 mètres du capteur et une paire d'images stéréoscopiques a été acquise (voir figure 132).

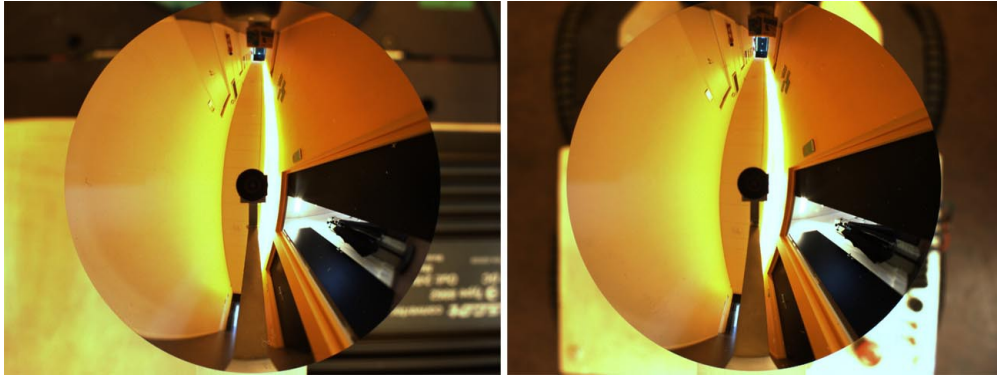


Figure 132: Images omnidirectionnelles utilisées pour la reconstruction 3D du cube.

Les coins de ce cube ont ensuite été sélectionnés dans les deux images omnidirectionnelles :

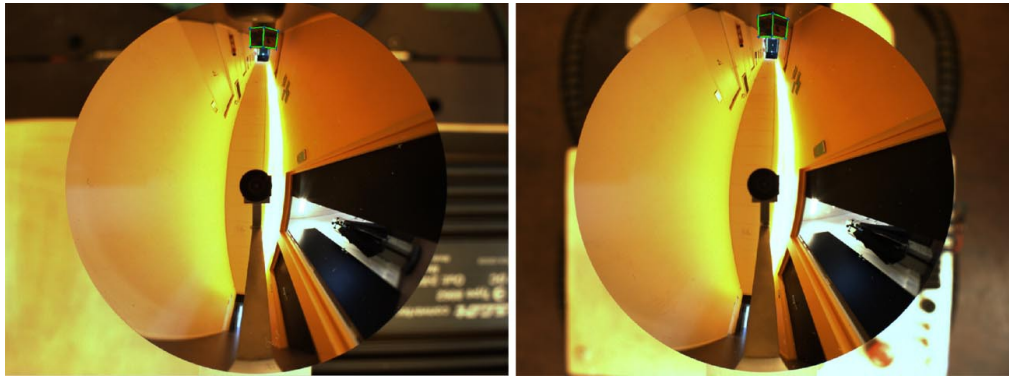


Figure 133: Sélection du cube dans les deux images omnidirectionnelles.

Les points sélectionnés ont été triangulés pour estimer leurs positions. Des plans passant par ces points ont été définis pour représenter les faces du cube comme le montre la figure suivante :

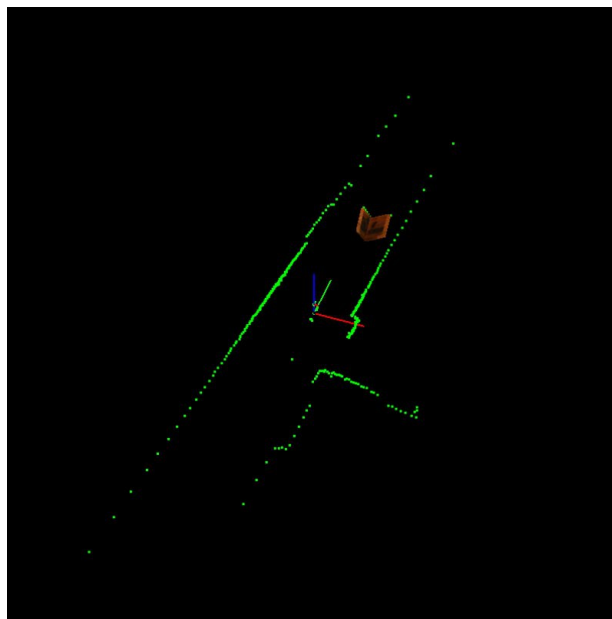


Figure 134: Reconstruction 3D du cube comparé à la vérité terrain (en vert).

La distance estimée par triangulation a ensuite été comparée à la vérité terrain obtenue par

le lidar (cf tableau 10).

Distance estimée (mm)	3149
Vérité terrain (mm)	3226
Erreur (%)	2,38%

Tableau 10: Distance estimée du cube par rapport à la vérité terrain.

Les dimensions du cube ont également été vérifiées. Pour cela, les longueurs de chaque arête visible ont été comparées aux dimensions réelles du cube :

Erreur moyenne (mm)	14,3
Erreur moyenne (%)	2,86

Tableau 11: Erreur sur le dimensionnement du cube.

L'erreur de positionnement est de 2,38 % et l'erreur de dimensionnement est de 2,86 % pour un objet situé à 3 mètres dont les dimensions sont de 50 cm de côté, ce qui répond bien aux exigences de la spécification technique de besoin.

### **Mission 3 : Préparation de la progression en milieu naturel**

#### **Objectif de la mission**

Préparation de la progression en milieu naturel : le porteur suivant un axe (route), le système de vision permet de localiser les obstacles locaux (arbres, mouvements de terrain, construction). Les arbres ou bâtiments visibles présents dans un rayon de 10 mètres du porteur lors du déplacement devront être détectés dans une proportion supérieure à 80 %.

#### **Lieu**

La séquence a eu lieu dans un environnement naturel de type forêt. Le porteur a été déplacé le long d'un chemin et devra être capable de localiser les arbres.



Figure 135: Environnement de la mission 3.



## Images omnidirectionnelles

La figure ci-dessous présente des images extraites de la séquence qui est utilisée pour cette mission :



Figure 136: Images omnidirectionnelles de la séquence utilisée pour la mission 3.

## Algorithmes utilisés et résultats

L'algorithme de programmation dynamique a été utilisé pour répondre à cette mission. Les cartes des profondeurs permettent de trianguler les points, et de vérifier s'il y a des obstacles dans un rayon de 10m. La même méthode a été appliquée pour la mission 5, les résultats seront donc présentés pour cette mission.

## Mission 4 : Préparation d'intervention d'aide à la mobilité

### Objectif de la mission

Préparation d'intervention d'aide à la mobilité : estimation du volume d'obstacles placés sur la voie (gravats, véhicules). Le porteur pourra soit être fixe, soit accomplir un déplacement dont la caractéristique pourra être fournie au système. Le système devra estimer le volume d'un obstacle avec une précision de 20%.

### Lieu

Pour cette mission, nous avons placé un véhicule sur la voie. Le porteur a effectué le tour du véhicule afin d'en estimer le volume.

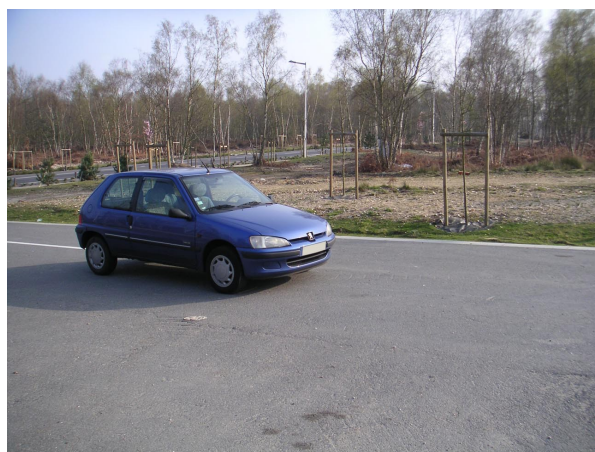


Figure 137: Environnement de la mission 4.

## Images omnidirectionnelles

La figure ci-dessous présente des images extraites de la séquence qui est utilisée pour cette mission :

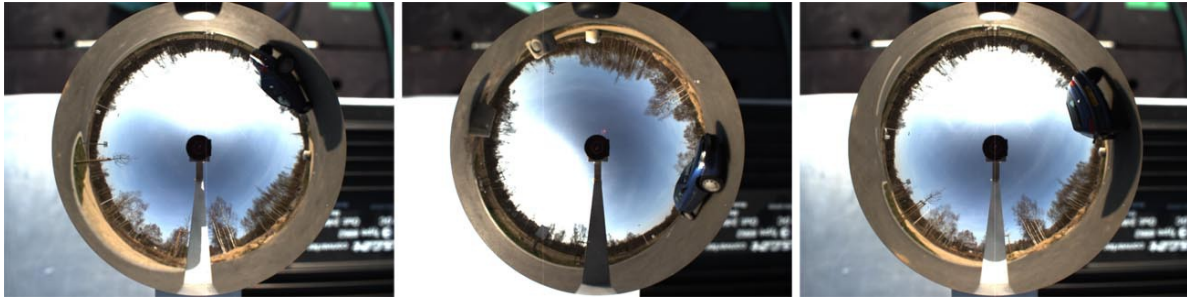


Figure 138: Images omnidirectionnelles de la séquence utilisée pour la mission 4.

## Algorithmes utilisés et résultats

L'évaluation du volume n'a pu être réalisée car la segmentation des objets n'a pas été abordée dans notre étude. Les algorithmes qui permettraient d'accomplir cette mission sont :

- l'algorithme d'ajustement de faisceaux pour estimer les positions du robot,
- la segmentation de l'objet dans les différentes images,
- un algorithme de *Shape-From-Silhouette* (cf section 1.2.2.7) pour obtenir l'enveloppe visuelle de l'objet, et donc son volume.

## Mission 5 : Préparation d'aide au déploiement

### Objectif de la mission

Préparation d'aide au déploiement : identification de zone d'héliport (évaluation de volume libre). Le système de reconstruction 3D, en position fixe ou en mouvement, sera placé dans un environnement cloisonné en partie par des bâtiments ou par des arbres (clairière). Le système devra statuer pour savoir si un cylindre de 10 mètres de diamètre est inscriptible dans le site.



Figure 139: Environnement de la mission 5.

## Lieu

Le porteur a effectué un déplacement le long d'un chemin en partant d'une forêt et en aboutissant sur une clairière. Le système devra être capable de déterminer à partir de quel moment il y a suffisamment de place pour faire atterrir un hélicoptère.

## Images omnidirectionnelles

La figure ci-dessous présente des images extraites de la séquence qui est utilisée pour cette mission :

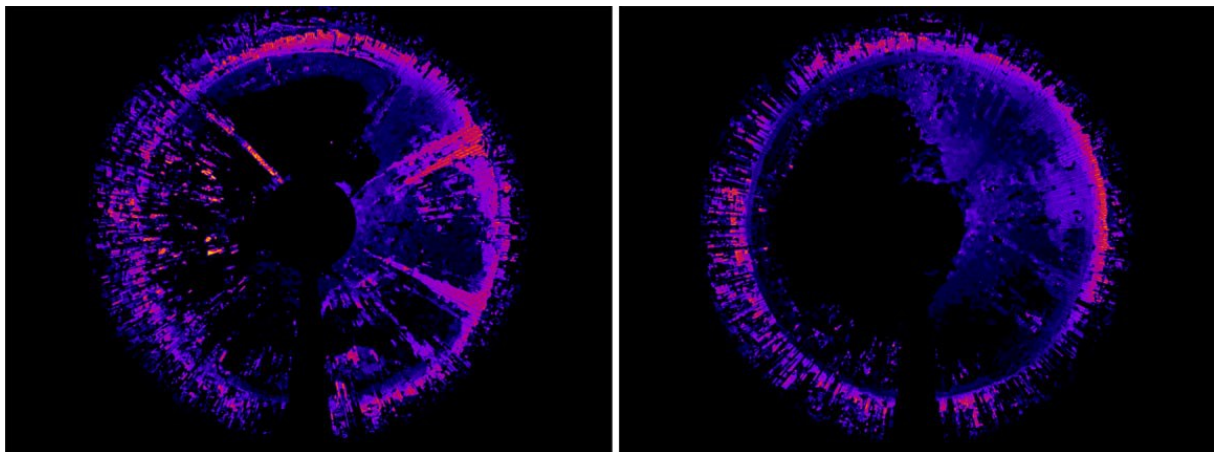


Figure 140: Images omnidirectionnelles de la séquence utilisée pour la mission 5.

## Algorithmes utilisés et résultats

L'algorithme utilisé pour répondre est l'algorithme de programmation dynamique (cf chapitre 3). Cet algorithme permet de faire la mise en correspondance des pixels, et donc de trianguler les points pour avoir leurs coordonnées 3D. Il est alors facile de vérifier si les points appartiennent ou non à un cylindre de 10 mètres de diamètre. Les résultats sont donc :

- la carte des profondeurs (cf figure 141),
- la décision sur la possibilité ou non de poser un hélicoptère (cf figure 142).



(a)

(b)

Figure 141: Cartes des profondeurs pour deux positions de la mission 5.  
(a) Position 7, atterrissage impossible – (b) Position 14, atterrissage possible



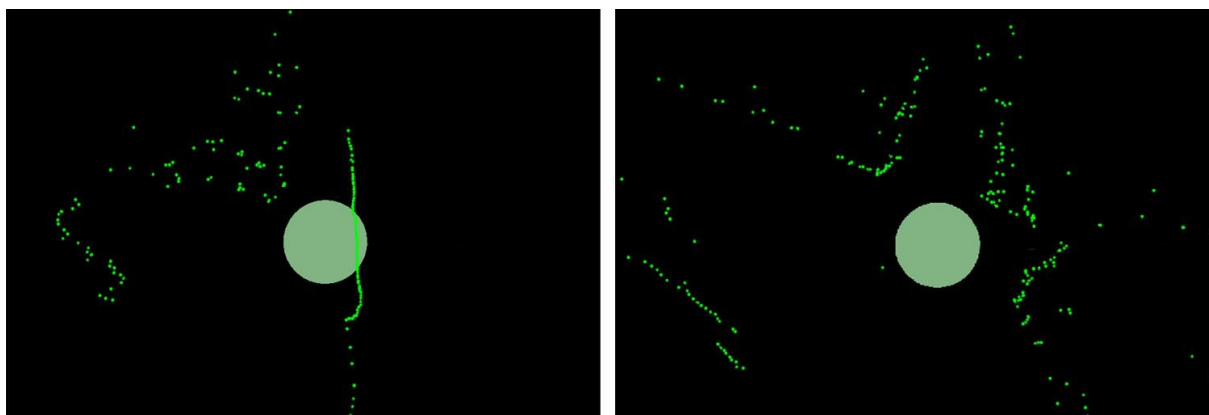
```

D:\These\Development\CMission5\bin\Debug\Mission5.exe
Image 0 : Atterissage impossible
Image 1 : Atterissage impossible
Image 2 : Atterissage impossible
Image 3 : Atterissage impossible
Image 4 : Atterissage impossible
Image 5 : Atterissage impossible
Image 6 : Atterissage impossible
Image 7 : Atterissage impossible
Image 8 : Atterissage impossible
Image 9 : Atterissage impossible
Image 10 : Atterissage impossible
Image 11 : Atterissage impossible
Image 12 : Atterissage possible
Image 13 : Atterissage possible
Image 14 : Atterissage possible
Image 15 : Atterissage possible

```

Figure 142: Décision sur la possibilité de poser un hélicoptère

La transition entre l'impossibilité d'atterrir et la possibilité de le faire correspond à la transition entre la position 11 et la position 12 (cf figure 142). Les résultats ont donc été validés pour ces deux positions grâce aux cartes fournies par le lidar. Sur ces cartes, un cercle de 5 mètres de rayon autour du porteur a été tracé pour vérifier s'il y avait ou non des obstacles dans ce rayon. Le résultat est donné sur la figure 143.



(a) (b)  
 Figure 143: Vérification des décisions grâce aux cartes du lidar.  
 (a) Position 11 – (b) Position 12.

La figure 143 valide les décisions qui ont été prises. En effet, pour la position 11, des points 3D sont présents dans le cylindre de 5 mètres de rayon ; l'atterrissage est donc impossible. Pour la position 12, les points 3D sont situés à l'extérieur de ce cercle, l'atterrissage est donc possible.

## Mission 6 : Surveillance du champ de bataille

### Objectif de la mission

Surveillance du champ de bataille : discrimination des obstacles fixes et mobiles (lors d'un déplacement). Estimation des distances et vitesses de déplacement des mobiles.

Porteur fixe : une autre plate-forme robotisable se déplacera dans le champ de vue du



système et devra être détectée.

Porteur mobile : le porteur en mouvement détectera une autre plate-forme mobile se déplaçant à proximité dans un rayon de 10 mètres.

### **Remarque**

La mission 6 a été écartée car l'approche est fondamentalement différente des autres missions. Cette mission avait en effet lieu dans un environnement dynamique, ce qui n'est pas pris en compte dans notre étude.

# Résumé

Les travaux présentés dans cette thèse concernent la reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions. Nous nous intéressons dans un premier temps à la conception d'une architecture matérielle adaptée aux problématiques de la reconstruction 3D et de la navigation autonome.

Le calibrage d'un système de vision est une étape indispensable dès lors que celui-ci est destiné à effectuer des mesures sur son environnement. Cette phase consiste à modéliser le système pour établir la relation mathématique liant les points 3D et leurs projections dans les images. Après une discussion sur le choix du modèle, nous présentons une méthodologie pour estimer les paramètres du modèle retenu.

La structure stéréoscopique du capteur que nous avons développé rend possible la reconstruction tridimensionnelle de l'environnement sans déplacement. Nous proposons donc des algorithmes permettant la reconstruction dense de l'environnement, ainsi que des algorithmes de détection de primitives dans les images omnidirectionnelles.

Lorsque le capteur est en mouvement, nous exploitons ses déplacements pour enrichir le modèle 3D. Notre principale contribution porte sur le développement d'un algorithme d'ajustement de faisceaux pour les capteurs stéréoscopiques omnidirectionnels qui permet d'obtenir une estimation des déplacements en ne nécessitant que des données visuelles.

**Mots clés :** vision par ordinateur, vision omnidirectionnelle, capteur catadioptrique, étalonnage, stéréovision, reconstruction 3D, estimation de déplacements.

# Abstract

This thesis deals with three-dimensional reconstruction of a mobile robot's environment. Based on omnidirectional vision data, it aims at preparing military operations. We first developed a well-adapted architecture for 3D reconstruction and autonomous navigation.

Calibration is a key step for a vision system in order to measure its environment. This step consists in modelling the system to establish the mathematical relationship between 3D points and their projections onto the image. After discussing a model choice, we will present a method to estimate its parameters.

The stereoscopic structure of the sensor allows three-dimensional reconstruction from a static viewpoint. We thus propose algorithms allowing a dense reconstruction as well as detection of the primitives in omnidirectional images.

When the sensor is moving, we use its motion to enhance the 3D model. Our major contribution consists in developing a bundle adjustment algorithm for omnidirectional stereoscopic sensors, which provides a motion estimation using only visual data.

**Key words :** computer vision, omnidirectional vision, catadioptric sensor, calibration, stereovision, 3D reconstruction, motion estimation.