



HAL
open science

Analyse adaptative du mouvement dans des séquences monoculaires non calibrées

Diane Lingrand

► **To cite this version:**

Diane Lingrand. Analyse adaptative du mouvement dans des séquences monoculaires non calibrées. Interface homme-machine [cs.HC]. University of Nice - Sophia Antipolis, France, 1999. Français. NNT: . tel-00459424

HAL Id: tel-00459424

<https://theses.hal.science/tel-00459424>

Submitted on 23 Feb 2010

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

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

THÈSE

préparée à

L'INRIA Sophia Antipolis

et présentée à

L'UNIVERSITÉ de NICE - SOPHIA ANTIPOLIS

pour obtenir le grade de

DOCTEUR

Spécialité

Sciences pour l'Ingénieur

par

Diane LINGRAND

Sujet de la thèse :

ANALYSE ADAPTATIVE DU MOUVEMENT DANS DES SEQUENCES MONOCULAIRES NON CALIBREES

Soutenue le 22 Juillet 1999 devant le jury composé de :

M.	Pierre	BERNHARD	Président
MM.	Roger	MOHR	Rapporteurs
	Phil	TORR	
MM.	Rachid	DERICHE	Examineurs
	Michel	DHOME	
	Andrew	ZISSERMAN	
M.	Thierry	VIÉVILLE	Directeur

Remerciements

Je voudrais remercier Pierre Bernhard d'avoir accepté, malgré sa récente charge de grand-père, de bien vouloir présider mon jury de thèse, Roger Mohr, bien que camouflé au fin fond de l'Australie ainsi que Phil Torr d'effectuer le lourd travail de rapporteur (thanks Phil for your big efforts in reading french), Rachid Deriche, Michel Dhome et Andrew Zisserman de venir examiner ma thèse.

Je tiens également à remercier Karl Tombre et Brigitte Wrobel-Dautcourt de m'avoir initié à la Vision par Ordinateur, l'ESSI et son DEA_VAR d'avoir continué à m'intéresser à ce domaine et, bien-sûr, le projet RobotVis pour m'avoir permis de jouer pendant trois ans avec mes équations et caméras et de m'avoir encouragé dans cette voie, Olivier, le chef et l'Academicien, toi qui supporta dignement plusieurs 1er Avril consecutifs et autres canulars, Thierry, qui me fit decouvrir de façon ludique le petit monde de la recherche (à quand ton nouveau bouquin?) et su m'encourager tout au long de ma thèse mais surtout pour la période bibliographique ainsi que tous les RobotVisiens et autres personnages de l'INRIA pour ces années sympathiques (voir figure ci-dessous).



Il me faut aussi remercier mes camarades de musique: Sue, Jean-Daniel, Mihai, Hervé et Marc, mes camarades d'aventures souterraines et plus généralement de montagne: le Barbare et ses "col-lègues" du SIS, le Yéti, les Ecureuils, ... (tout un bestiaire), Michoko pour les soirées chocolat et Ryad pour ses nombreux encouragements tout au long de la thèse.

Et Johan, bien entendu, pour le bonheur du *two bodies solution*.

ANALYSE ADAPTATIVE DU MOUVEMENT DANS DES SÉQUENCES MONOCULAIRES NON CALIBRÉES.

Diane LINGRAND

INRIA - projet RobotVis, B.P. 93, F-06902 Sophia Antipolis Cédex, France.

Dans le cadre de la perception visuelle liée à des problèmes de robotique, cette thèse s'intéresse plus précisément à l'analyse de séquences d'images monoculaires non calibrées, en prenant compte les cas particuliers physiques (modèle de caméras, évolution des paramètres internes de la caméra, déplacement des objets dans la scène, de la caméra, structure de la scène) conduisant à des équations spécifiques. Les singularités permettent de retrouver, selon les cas, plus ou moins d'éléments que dans le cas général sur le mouvement ou la structure, mais toujours avec plus de précision car moins de paramètres sont en jeu. Il est donc important de pouvoir les détecter et gérer au mieux les propriétés cinématiques et géométriques qui en découlent. Un examen exhaustif de tous les cas particuliers sur deux images, puis sur une séquences d'images montre que la complexité du problème nécessite un traitement adapté.

Le module développé, opérationnel sur le système robotique Argès, permet, à partir de 2 images, de déterminer quel mouvement spécifique a été effectué et fournit ensuite les paramètres idoines du modèle.

Mots clés : *vision par ordinateur, mouvement 3D, auto-calibration, singularités, séquences d'images monoculaires, mise en correspondance.*

ADAPTATIVE MOTION ANALYSIS IN MONOCULAR IMAGE SEQUENCES WITHOUT CALIBRATION.

Part of the robotic vision framework, this thesis focuses on uncalibrated monocular video sequence analysis, taking singular physical cases into account (camera models, internal camera parameter evolution, object displacements, scene structure) that leads to specific equations. Singularities may allow us to retrieve more movement or structure elements than general equations. Moreover, numerical precision is improved as the number of parameters decreases. Thus, the detection and the proper management of geometric and kinematic properties of those singular cases are fundamental. The complete study of all singular cases in a pair of images, or a video sequence, is computationally intractable and requires an appropriately adapted algorithm.

The implementation of the theoretical study is based on the Argès robotic system. The system is able to determine the specific movement from a pair of images and computes the related parameters.

Keywords: *computer vision, motion, auto-calibration, no-calibration, specific cases, monocular image sequences.*

Notations

les vecteurs sont représentés par des minuscules et notés en **gras**

les matrices sont représentées par des majuscules et notés en **gras**

$\mathbf{x} \wedge \mathbf{y}$ représente le produit vectoriel des vecteurs \mathbf{x} et \mathbf{y}

$\tilde{\mathbf{x}}$ représente la matrice anti-symétrique de l'opérateur produit vectoriel du vecteur \mathbf{x}

$\tilde{\mathbf{x}} \mathbf{y}$ représente alors le produit vectoriel des vecteurs \mathbf{x} et \mathbf{y}

$\mathbf{x} \cdot \mathbf{y}$ représente le produit scalaire des vecteurs \mathbf{x} et \mathbf{y}

$\mathbf{x}^T \mathbf{y}$ représente le produit scalaire des vecteurs \mathbf{x} et \mathbf{y}

\otimes représente le connecteur logique "ET"

\oplus représente le connecteur logique "OU"

Table des matières

Introduction	1
1 Qu'est-ce que la Vision par Ordinateur?	3
2 Cadre de l'étude	5
3 But du travail	5
4 Cheminement	6
1 Modélisation	7
1.1 Les capteurs visuels.	9
1.1.1 Qu'est-ce qu'un capteur visuel?	9
1.1.2 La caméra sténopée	10
1.1.3 Les composants optiques d'une caméra	12
1.1.4 Modélisation d'une caméra	17
1.2 La géométrie projective : généralités et applications	20
1.2.1 Définitions - Principes	20
1.2.2 Application à la vision par ordinateur	21
1.3 Les projections	21
1.3.1 La projection perspective	22
1.3.2 La projection para-perspective	23
1.3.3 La projection ortho-perspective	26
1.3.4 La projection orthographique ou affine	27
1.3.5 Différents modèles en une équation	29
1.4 Raffinements possibles	31
1.4.1 Le modèle des lentilles épaisses	31
1.4.2 Prise en compte des distorsions	32
1.4.3 Différents types de caméras	34
1.5 La paramétrisation du mouvement	36
1.5.1 Mouvement rigide	36
1.5.2 Paramétrisation de la rotation	37
1.6 Paramétrisation de 2 vues : la géométrie épipolaire	41
1.6.1 Géométrie épipolaire: généralités	41

1.6.2	La disparité vectorielle	42
1.6.3	Retour au cas de la projection perspective	45
1.6.4	La matrice fondamentale	46
1.6.5	Les homographies	50
1.7	Résumé du chapitre.	56
2	Les Cas Particuliers	59
2.1	Introduction.	61
2.2	Étude Bibliographique.	61
2.2.1	Modèle de projection.	62
2.2.2	Paramètres internes de la caméra.	62
2.2.3	Mouvement.	63
2.2.4	Géométrie de la scène, de l'objet.	64
2.2.5	L'ensemble des cas particuliers.	65
2.3	Les cas particuliers concernant la modélisation de la caméra.	65
2.3.1	Modèle de projection.	65
2.3.2	Les paramètres intrinsèques.	66
2.4	Les cas particuliers de mouvement.	68
2.4.1	Mouvement discret - mouvement continu.	68
2.4.2	Relations entre rotation et translation.	69
2.4.3	Expression des contraintes sur les paramètres de mouvement	69
2.5	Les cas particuliers géométriques.	72
2.6	Le cas de plus de 2 vues	75
2.6.1	Introduction: qu'en pensent les chercheurs?	75
2.6.2	Variations du mouvement et du zoom dans le cas de N vues	79
2.6.3	Relations entre rotation et translation dans le cas de N vues	80
2.7	Gestion de l'inventaire des cas particuliers.	81
2.7.1	Contraintes atomiques - contraintes moléculaires	81
2.7.2	Conditions d'existence.	82
2.7.3	Consistance	84
2.8	Les cas particuliers dans le cas de 2 vues.	84
2.8.1	Les formes particulières de matrices fondamentales.	85
2.8.2	Les formes particulières de matrices d'homographie.	92
2.9	Conclusion	104
3	Estimation	105
3.1	Introduction	107
3.2	Le bruit	107
3.2.1	Origine des incertitudes.	107
3.2.2	Modélisation du bruit	108

3.2.3	Elimination du bruit: les techniques robustes	110
3.3	Les techniques de minimisation.	110
3.3.1	Une méthode robuste historique: la transformée de Hough	111
3.3.2	Méthodes de régression (<i>Case Deletion Diagnostic</i>).	111
3.3.3	Moindres carrés linéaires.	111
3.3.4	Moindres carrés non-linéaires: Levenberg-Marquardt	112
3.3.5	Méthode de la moindre médiane des carrés (<i>LMedS: Least-Median Square</i>)	113
3.3.6	Méthode de la ν -iane (<i>ν-trimmed Square</i>).	114
3.3.7	RANSAC (<i>RANdom SAmple Consensus</i>)	114
3.3.8	M-Estimeurs	115
3.3.9	Comparaison des méthodes	116
3.4	Evaluation du résultat	116
3.4.1	Evaluation d'un modèle	116
3.4.2	Comparaison de différents modèles.	118
3.5	Une autre alternative: Les algorithmes génétiques.	122
3.5.1	Présentation du problème: résumé	122
3.5.2	Présentation des algorithmes génétiques	123
3.5.3	Adaptation à notre problème	124
3.6	Approche bayésienne	125
3.7	Calcul de l'erreur dans le cadre de notre problème de vision	125
3.7.1	Erreur algébrique et erreur géométrique	125
3.7.2	Calcul d'un résidu du problème	126
3.7.3	Erreurs épipolaires pondérées par le gradient	127
3.7.4	Distances des points aux retroprojections	127
3.7.5	Comparaison des différents critères	127
3.7.6	Autres cas	127
3.8	Conclusion de cette étude: notre choix	128
4	Expérimentations et applications	129
4.1	Présentation de la plate-forme expérimentale Argès.	131
4.1.1	Présentation du système robotique Argès.	131
4.1.2	Présentation de l'environnement logiciel Argès	134
4.2	Prototype d'expérimentations d'un échantillon de cas particuliers concernant deux images	135
4.2.1	Présentation de l'organisation logicielle: implémentation multi-langages	135
4.2.2	Expérimentations	137
4.3	Expérimentations sur les différentes formes d'homographies à l'infini.	142
4.4	Une application de vision active sur un cas particulier	150
4.5	Application: reconstruction hiérarchique.	154
4.6	Application: segmentation d'une séquence vidéo.	154

1	Conclusion	157
2	Perspectives	157
	Conclusion	155
	Annexes	159
A	Maple : un outil adapté aux calculs lourds, longs et fastidieux	161
A.1	Code Maple :coefficients de la disparité vectorielle dans le cas de la projection perspective	162
A.2	Code Maple :coefficients de la disparité vectorielle dans le cas de la projection paraperspective	163
A.3	Code Maple : Expressions de la matrice fondamentale \mathbf{F}	164
A.4	Condition d'existence d'une relation homographique dans le cas de la projection paraperspective	166
A.5	Génération des contraintes moléculaires dans le cas de 2 vues	167
B	Analyse détaillée de quelques cas particuliers.	181
B.1	Considering screw rotations with fixed camera parameters.	182
B.2	Considering fixed axis rotations with fixed camera parameters.	183
B.3	Considering pure translations with fixed camera parameters.	185
B.4	Considering a zoom of the camera.	186
B.5	Considering retinal displacements.	188
C	Ebauche de reflexion à propos de la préhension du mouvement chez l'être humain.	193
C.1	Introduction	194
C.2	Présentation du sujet	194
D	Le pot de thèse	197
D.1	Le cake aux olives.	198
D.2	La tarte au fromage de Michoko.	198
D.3	Le cake au citron.	199
D.4	Le nougat glacé.	199
D.5	Le tiramisu.	200
D.6	Gateau au chocolat, amandes et gingembre.	200
	Bibliographie	201
	Index	211

Table des figures

1	Les quatre problèmes élémentaires en Vision par Ordinateur.	4
1.1	Principe de la caméra sténopée	10
1.2	L’oeil du Nautilus fonctionne selon le principe sténopé	10
1.3	Schéma de caméra sténopée (dans <i>De Radio Astronomica et Geometrica</i> , par Gemma Frisius, 1545), utilisée pour l’observation de l’éclipse de soleil de 1544.	11
1.4	Réflexion et réfraction entre deux milieux.	12
1.5	Schéma de prisme réflecteur	13
1.6	(a): bi-convexe, (b): plan-convexe, (c): ménisque positif, (d): bi-concave, (e): plan-concave, (f): ménisque négatif	13
1.7	Schéma de lentille convergente	14
1.8	Schéma de lentille divergente	14
1.9	Schéma de miroir concave	15
1.10	Schéma de miroir convexe	16
1.11	Schéma de prisme	16
1.12	caméra Tessar, par Zeiss, 1902	17
1.13	Le schéma interne de l’objectif “SUMMILUX-R f/1.4/50mm” de chez Leica	18
1.14	Le modèle sténopé	19
1.15	Principe de la projection projective	22
1.16	Principe de la projection para-perspective	23
1.17	Principe de la projection ortho-perspective	26
1.18	Principe de la projection orthographique	27
1.19	La projection perspective et ses approximations.	29
1.20	Modèle de lentille épaisse, d’après (Pahlavan et al., 1993)	31
1.21	Les types de distorsions les plus couramment observés: barillet et coussinet	32
1.22	Miroir conique: notations et calcul des angles	35
1.23	de gauche à droite et de haut en bas: (a) Lentille sphérique; (b) Miroir conique; (c) Miroir à double courbure; (d) Barettes CCD en rotation;	36
1.24	Les angles d’Euler	37
1.25	Géométrie épipolaire: Notations	41

1.26	Principe de la géométrie épipolaire	42
1.27	Violation de la contrainte d'unicité: le point \mathbf{m} de la première image admet deux points \mathbf{m}'_1 et \mathbf{m}'_2 pour correspondants dans la deuxième image, ceci par transparence.	42
1.28	Violation de la contrainte de continuité: il y a une discontinuité de profondeur en \mathbf{M}_1 et \mathbf{M}_2 car il y a occlusion d'un objet par un autre.	43
1.29	Violation de la contrainte d'ordre: sur la première image, \mathbf{m} est entre \mathbf{n} et \mathbf{p} , sur la seconde, \mathbf{n}' est entre \mathbf{m}' et \mathbf{p}' . La différence de profondeur des points \mathbf{M} , \mathbf{N} et \mathbf{P} est importante par rapport à l'écart des centres optiques des caméras.	43
2.1	Cas de trois vues	75
2.2	Séquence comportant N images	76
2.3	Représentation en \mathbf{Qs} du mouvement rétinien dans une séquence d'images	78
2.4	Principe de la factorisation de Sturm (dans le cas affine). La matrice des mesures \mathbf{W} est de rang 3 (possède alors 3 valeurs singulières : s_1 , s_2 et s_3). Les parties grisées sont donc inutiles.	79
2.5	Exemples de cas particuliers physiquement réalisables en utilisant des mécanismes robotiques générant des translations pures, des rotations pures ou translations dans le plan orthogonal à l'axe de rotation. <i>Si, pour le bras articulé, (a) et (b) ont des valeurs opposées, c'est une translation pure. Effectuant la même commande (c) sur les roues du robot mobile, on obtient une translation. Un déplacement de (a) seul, ou (b) seul, sur le bras robotique fait que la translation est dans le plan orthogonal à l'axe de rotation. Une commande (c) appliquée de façon opposée aux roues du robot mobile provoque le même déplacement. Les tourelles peuvent effectuer des rotations pures en pan (e) ou tilt (f) autour d'un centre optique approximatif.</i>	82
3.1	Détermination de la médiane et ν -iane	114
3.2	Notations pour la justification du critère d'Akaike	119
3.3	Opérateurs de croisement et de mutation	124
3.4	Description d'une chaîne ou chromosome correspondant à notre problème.	124
4.1	Photo du système robotique Argès	131
4.2	Caractéristiques techniques de la caméra utilisée dans le système robotique Argès (d'après le constructeur, "I magic").	132
4.3	schéma de câblage des éléments constituant le système robotique Argès	133
4.4	Schéma synthétique du système robotique Argès	134
4.5	Interface du système robotique Argès	134
4.6	Organisation logicielle	137
4.7	Interface du module de test de cas particuliers.	138
4.8	images prises par Argès lors d'un mouvement de translation pure	139
4.9	154 points ont été détectés et appariés entre les 2 images	139

4.10	disparité vectorielle entre les deux images de la figure 4.8.	139
4.11	images prises par Argès lors d'un zoom et translation en x	140
4.12	196 points ont été détectés et appariés entre les 2 images	140
4.13	disparité vectorielle entre les deux images de la figure 4.11.	141
4.14	Rotation de la caméra d'Argès autour de son axe optique.	149
4.15	Mouvement approximatif de rotation autour de l'axe optique et translation	150
4.16	Evolution de l'erreur en mouvement en fonction de l'orientation en roulis de la caméra, modèle théorique et expérimentation (scène de la mire).	151
4.17	Scène réelle vue par Argès.	152
4.18	Vues de face et de dessus de la reconstruction.	153
B.1	Notations for a fixed axis rotation	183

Introduction

Pourquoi encore une thèse en Vision par Ordinateur?

« Un chercheur qui trouve est un chercheur qui casse le métier. »

Dominique Méry

Introduction (english abstract)

In Computer Vision, many methods and algorithms provide very good results on generic sets of data but fail in other cases which yield singularities in the equations used. In such situations, the adaptation of basic algorithms may still provide good results, by introducing some human interaction when automatic processing fails. The approach is to assist the computer in choosing a "specific" model to get rid out of such singular cases.

Indeed, we observe that if we model particular cases in the equations:

- we obtain simpler equations with less parameters,*
- we can estimate the parameters with a better numerical stability,*
- we retrieve some parameters that we cannot retrieve with the general equations, since additional equations have been introduced.*

thus, the key point is that, since general methods fail in cases with singularities or near singularities, it is possible to take into account these singularities, by explicitly managing the underlying specific equation sets. In a further step, we may attempt to generate them actively on a robot to retrieve some information that we cannot retrieve in the general case, thus reconstructing iteratively a scene, using a hierarchy of such singular cases.

Therefore, in this thesis, we study the modelisation of the problem, including an exhaustive list of effective singular cases. We then build an hierarchy of these models and study how to navigate in this hierarchy during the estimation process. Lastly, we present experimental tests of this method using a robotic system and video sequences.

1 Qu'est-ce que la Vision par Ordinateur ?

La Vision par Ordinateur est une discipline très vaste au croisement de différents domaines des Sciences de l'Ingénieur tels que les mathématiques, l'intelligence artificielle, le traitement de signal, l'automatique et l'informatique afin de réaliser différentes tâches perceptives. Au niveau théorique, elle a pour but de fournir des modèles de la perception visuelle des formes, du mouvement et des objets. Au niveau applicatif, elle a pour but d'automatiser différentes opérations telles que la détection d'obstacles pour la navigation d'un robot, la reconstruction tridimensionnelle de scène, la reconstruction de volumes dans le cas d'images géologiques ou médicales, la détection d'objets en mouvement, la détermination du mouvement propre d'un robot, le suivi d'éléments d'intérêts, la reconnaissance de formes ou d'objets, la fusion d'images provenant de plusieurs modes d'acquisition (en imagerie médicale, images IRM, scanner, rayons X, ...), l'indexation d'images ou séquences vidéos, ...

Dans cette thèse, on s'intéresse à la partie purement géométrique de la vision par ordinateur et qui concerne la perception du mouvement et de la structure à partir de deux ou plusieurs images acquises par des systèmes vidéos.

On peut facilement représenter cette partie du problème de la vision sous la forme de quatre problèmes élémentaires: (i) la détermination des paramètres de projection, (ii) la mise en correspondance, (iii) la détermination du mouvement rigide et (iv) la détermination de la structure tridimensionnelle. On exprime alors différents problèmes rencontrés en vision comme étant la détermination de certaines variables à partir d'autres de ces composantes supposées connues, comme illustré par la figure 1 et le tableau suivant dans lequel, pour chaque problème classique de la vision, on donne son équivalent en terme de détermination de composantes en fonction d'autres composantes connues :

problème	composantes équivalentes			
	projection	correspondances	mouvement rigide	structure 3D
structure par mouvement	connue	connues	connu	inconnue
appariement et reconstruction	connue	inconnues	connu	inconnue
calcul du mouvement	connue	connues	inconnu	inconnue
suivi de motif	connue	inconnues	inconnu	inconnue
calibration	inconnue	connues	inconnu	non nécessaire

La géométrie projective est l'outil adapté à la modélisation des projections effectuées par les caméras courantes. Les projections et le mouvement d'objets ou de caméra sont donc représentés algébriquement. On en déduit alors des relations entre les projections dans les images comme la géométrie épipolaire dans le cas de deux vues.

Cependant, les données réelles en entrée de ces traitements sont bruitées. Il faut alors en tenir compte à chaque étape : acquisition et numérisation des images, modélisation des caméras, traitement des images pour l'extraction de primitives géométriques (contours, régions, points, droites, courbes, ensemble de primitives), mise en correspondance des primitives, estimation des paramètres des modèles, minimisation sur un grand nombre de données, évaluation statique des résultats, ...

Spetsakis et Aloimonos (Spetsakis et Aloimonos, 1991) distinguent à ce propos trois époques de

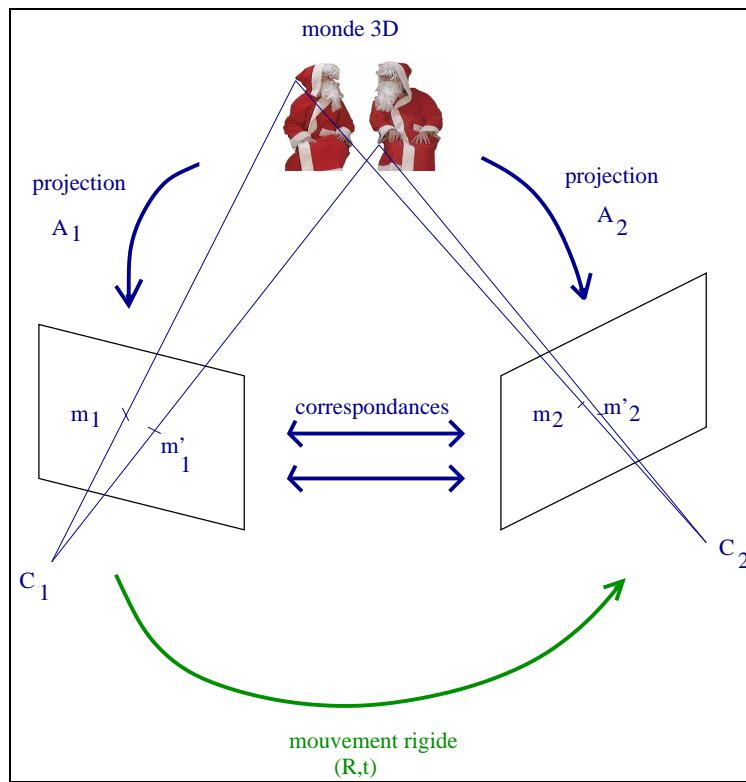


FIG. 1 – *Les quatre problèmes élémentaires en Vision par Ordinateur.*

la recherche en vision par :

1. Le problème générale de la vision admet-il une solution?
2. Cette solution est-elle unique?
3. Comment la rendre robuste au bruit?

Chaque étape possède de nombreux algorithmes dont la complexité en taille mémoire et en temps de calcul ainsi que la précision et la robustesse au bruit varient. Le choix se portera sur les algorithmes les plus adaptés aux champs d'application liés au problème considéré :

robotique temps-réel où la rapidité des algorithmes est un élément prépondérant. En effet, la navigation autonome d'un robot dans un environnement inconnu ne nécessite qu'une reconstruction grossière de l'environnement mais demande d'agir en temps réel afin de ne pas heurter un obstacle avant d'avoir analysé l'image montrant l'obstacle.

reconstruction réaliste où la qualité des reconstructions tridimensionnelles, quitte à faire intervenir l'utilisateur pour corriger certains points ou pour introduire des informations supplémentaire, est cruciale. Ces reconstructions peuvent servir de base à des applications de réalité augmentée dans des domaines divers.

Sur ces bases, en vision robotique, de nombreux problèmes simples sont résolus avec efficacité, preuve en sont les applications industrielles actuelles. Cependant, il est encore difficile en pratique pour un robot d'évoluer de façon autonome dans un environnement inconnu et d'y effectuer des

tâches perceptives avec précision. Il est alors nécessaire qu'un utilisateur rentre manuellement des informations ou interagisse avec le système, ce qui n'est pas en soit une méthode à rejeter.

Cette thèse pose la question de savoir si une prise en compte de cas spécifiques de mouvement ne serait pas une solution permettant à la fois (i) d'effectuer une reconstruction en commençant par des mouvements simples et des éléments simples de structures pour lesquels on sait relativement bien résoudre le problème pour ensuite se servir de ces éléments pour des mouvements et éléments de structure plus complexes, (ii) d'échapper aux singularités théoriques et numériques des équations générales pour lesquelles il faut adapter les équations et les paramètres, (iii) d'obtenir une vraie reconstruction adaptée à la tâche perceptive recherchée.

2 Cadre de l'étude

On se place dans le cadre de l'observation de scènes tridimensionnelles dans lesquelles un ou plusieurs objets rigides sont en mouvement. Notre point de vue est celui d'une caméra unique pouvant se déplacer dans cette scène. Généralement, cette caméra ne sera pas calibrée, mais pourra faire varier sa mise au point, sa focale, ceci faisant également varier d'autres paramètres du système de prise de vue.

On s'intéresse alors à l'étude du mouvement de telles caméras ou celui des objets observés. Cela implique aussi l'analyse de la structure de scènes tridimensionnelles et de la modélisation de ces caméras car leurs équations sont liées. Plus précisément, on s'intéresse aux singularités de mouvement, de structure ou de caméras, c'est-à-dire aux cas particuliers des équations qui les modélisent.

Cette étude peut être généralisée à plusieurs caméras : il suffit de considérer que le changement de caméra correspond à un déplacement spatial du capteur. Il faudrait cependant faire une étude plus poussée afin d'utiliser les particularités si des caméras filment simultanément une même scène.

3 But du travail

Il existe des cas particuliers pour lesquels les méthodes générales d'estimation du mouvement ne fonctionnent pas. Par exemple, dans le cas d'un objet en rotation pure, la matrice fondamentale n'existe pas et les méthodes basées sur son estimation ne peuvent donc pas être employées. On s'intéresse dans ce cas à une autre quantité : l'homographie. D'autres cas semblables existent, et notre premier objectif est d'énumérer tous ces cas.

D'un autre côté, de tels cas particuliers peuvent fournir des équations simplifiées permettant d'obtenir des informations inobservables dans le cas général : on peut alors, d'une part, déterminer des paramètres qui ne sont généralement pas déterminables, et, d'autre part, obtenir des équations plus simples, permettant une estimation numérique plus robuste. En pratique, on est quasiment toujours en présence ou proche d'un cas particulier sur un système dont les mouvements sont contrôlés (véhicule, robot, caméra déplacée sur un support).

Notre but est donc ici multiple. On veut montrer que

- même pour un cas particulier non critique, le résultat est meilleur avec un modèle simplifié, comportant moins de paramètres qu’avec un modèle plus général (Viéville et Lingrand, 1999)
- on peut déterminer automatiquement dans quel cas particulier on se trouve
- le nombre de cas particuliers intéressant n’est pas indéfini, il est même relativement restreint si l’on considère uniquement les cas ayant une signification physique. Pour cela, on effectuera une énumération de tous les cas, à la fois algébriquement et comparativement à la littérature.
- on peut facilement appliquer ces modèles à des mécanismes de reconstruction hiérarchique.

4 Cheminement

Les premiers chapitres de cette thèse ont pour but de définir le cadre de notre étude de façon précise. Nous allons nous intéresser à la modélisation (chapitre 1) de la caméra, à la mise en équation des déplacements et aux propriétés géométriques de la scène. Ceci posé, nous pourrons aborder le problème des cas particuliers (chapitre 2), par un examen systématique et modulaire d’une part, par une analyse plus synthétique d’autre part.

Afin d’exploiter numériquement ces cas particuliers, nous allons tout d’abord examiner les méthodes d’estimation (chapitre 3) numériques et les choix qui ont été fait. L’implémentation et les expérimentations feront l’objet du chapitre 4. Une fois nos méthodes vérifiées expérimentalement, nous aborderons les différentes applications de la méthode proposée.

D’autre part, des perspectives de cette étude dans le domaine de l’étude de la perception biologique du mouvement sont présentées en conclusion.

Dans la mesure où ces points couvrent plusieurs aspects fondamentaux et pragmatiques de la vision par ordinateur, des parties de ce document pourront faire l’objet de support de cours.

Chapitre 1

Modélisation

Comment modéliser? Connaissant les composants d'une caméra et les lois qui les régissent, on peut déterminer les équations de la caméra. Malheureusement, ce modèle de connaissance est trop complexe, incomplet et il est rarement possible d'avoir accès aux données nécessaires à la constitution d'un jeu complet de paramètres pour un tel modèle. Ainsi, avons nous recherché d'autres mo-

dèles plus fonctionnels qui reflètent mieux le comportement effectif d'une caméra. Pour toutes ces alternatives, l'expression d'un système projectif linéaire est celui qui correspond le mieux à nos caméras usuelles. Cependant, moyennant quelques raffinements, cette formalisation correspond encore à d'autres caméras plus "exotiques".

« Si on parle philosophie, on ne va pas intégrer des laplaciens dans des ouverts simplement connexes, quand même hein? »

Olivier Faugeras

How to model camera projections? (english abstract)

How to model camera projections?

The camera being composed of several optical components, a first approach is to combine their optical laws to build a realistic model of the camera projection. However it is often not possible to obtain explicit information on each component and on their relative positions and orientations. Furthermore, the equations obtained would be intractable. We thus propose to use a simpler model.

The perspective model is the best model for most cameras in the field. This model's accuracy may be improved by additional elements in some cases, mostly concerning optical distortions. Here, we have decided to solve this problem by correcting distortions first.

Furthermore, in section 1.3, we have studied perspective projection approximation, para-perspective projection, ortho-perspective, and scaled orthographic, which may be considered both as geometric and numerical (limited-order expansions) restrictions, as demonstrated (pages 23 to 28). We analyze the relations between these approximations in figure 1.19 and propose a comprehensive parametrization.

Analyzing the simplifications obtained using these projections, in the paraperspective case, we show that the equations are linear but require 2 additional parameters, while in the orthographic case, corresponding to a particular case of the para-perspective case, the equations are linear without additional parameters.

These restrictive models are especially useful when considering the focus of attention since they correctly represent visual data in a small area of the retina, e.g. the fovea, while they yield robust numerical analysis of structure and scene motion, even when using a small amount of data.

How to model camera or object displacements?

In section 1.5, we revisit the parametrization of a rotation representation based on Lie group theory which allows us to model all rotations (including symmetries) with only one set of 3 parameters and who is still valid for singular rotations (symmetries). This model is equivalent to the quaternion representation but we have preferred the matrix formalism, for convenience.

How to model the relations between projected points in two frames?

We first verify, in section 1.6.2, that the flow disparity between two frames is affine for the para-perspective and orthographic approximations and determine its relations to the camera parameters. We note that the flow disparity is still affine in the perspective case, in case of retinal displacement.

After a generalization of the fundamental matrix to these projection models (see equation 1.17), we have analyzed in which conditions image points could be related a collineation, both as far as geometry and kinematic constraints are concerned. The results are collected in the table on page 57.

1.1 Les capteurs visuels.

1.1.1 Qu'est-ce qu'un capteur visuel?

Un capteur visuel est un système qui effectue une projection d'un espace tridimensionnel vers un espace mono-, bi-, ou tri-dimensionnel, et mesure l'amplitude d'énergie lumineuse reçue. Il discrétise l'espace en éléments d'image (pixel dans le cas bidimensionnel, voxel dans le cas tridimensionnel). Les capteurs que nous allons considérer par la suite sont soit des capteurs bidimensionnels, soit des ensembles de capteurs monodimensionnels afin d'obtenir dans tous les cas des images bidimensionnelles. On peut classer les capteurs en plusieurs catégories: les capteurs photo-chimiques (systèmes biologiques, films photographiques, ...), les capteurs thermiques (thermopile, ...) et les capteurs photoélectriques (photo-diode, détecteurs à transfert de charge ou DTC, ..)

Nous avons utilisé dans le cadre de cette thèse des caméras à base de barrettes à transfert de charge dite CCD (*Charged Coupled Device*). Ces barrettes sont des alignements de cellules photo-conductrices au silicium correspondant chacune à un élément d'image appelé "pixel" (*picture element*). En présence de lumière, l'impact des photons incidents fait apparaître localement des charges électriques qui sont piégées dans un puits de potentiel où elles s'accumulent pendant le temps qui sépare deux scrutations. A des instants donnés, les charges stockées (dépendant du nombre d'impacts de photons reçus) sont transférées simultanément sur les entrées parallèles d'un registre analogique à la sortie duquel l'image de la ligne analysée sera disponible. Après numérisation, chaque pixel possède une valeur de niveau de gris entre 0 et 255.

Le cas des caméras couleurs est un peu plus complexe. On distinguera quatre types de caméras couleurs:

- **filtres optiques colorés** : Kodak et Wratten utilisèrent des filtres colorés sur leurs caméras à niveau de gris. Ces manipulations étaient lourdes et le résultat nécessite des corrections à cause de la densité variable des filtres.
- **caméras couleur mono-CCD** : consistant en une superposition d'un masque de filtres sur la matrice CCD. De tels capteurs n'ont pas beaucoup de dynamique.
- **couleur tri-CCD** : il s'agit de séparer la lumière en trois faisceaux de longueurs d'onde différentes (rouge, vert, bleu) en utilisant un prisme puis d'envoyer chacun des faisceaux primaires sur un détecteur CCD sensible aux fréquences correspondantes. Une image est ainsi séparée en 3 images: une rougeâtre, une verdâtre et une bleuâtre. Plus cher, ce capteur demande une grande précision pour faire coïncider les 3 images (voir pour exemple la caméra Canon XL1, <http://www.canondv.com/xl1/prodinfo/ccd.html>)
- **couleur quadri-CCD** : à chaque pixel sont associées quatre cellules CCD différemment sensibles aux longueurs d'onde: une dans les fréquences proches du rouge, une pour le vert, une pour le bleu et une dernière afin d'améliorer la précision dans les fréquences proches du vert (le vert détient environ 60% de l'information dans une image).

Tous ces capteurs nécessitent une **gamma-correction** car l'intensité mesurée par les caméras est une fonction concave logarithmique de l'intensité réelle. Les couleurs apparaissent moins saturées

et les coordonnées chromatiques sont inexactes. La fonction correctrice inverse est appelée gamma-correction.

1.1.2 La caméra sténopée

La caméra sténopée est la caméra la plus simple qui existe. Le principe en est trivial: si la lumière passe par un petit trou (d'où le nom anglophone de "pinhole camera"), il se forme alors une image inversée (voir figure 1.1).

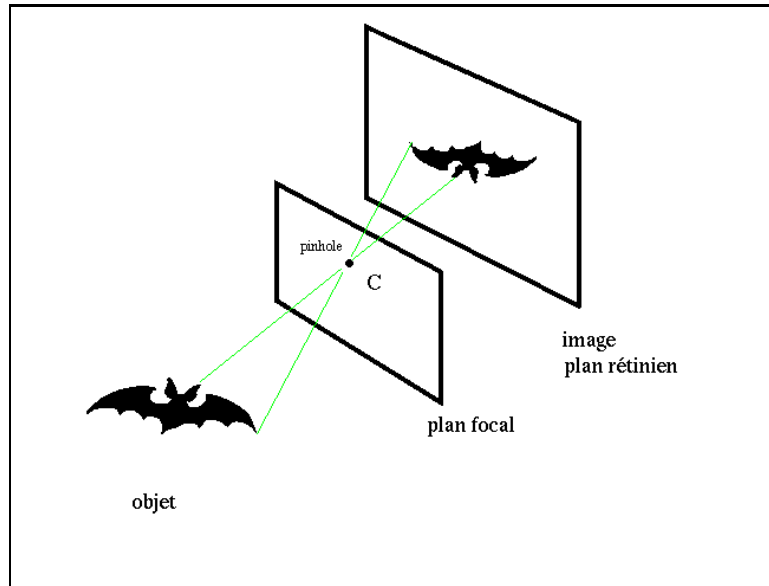


FIG. 1.1 – *Principe de la caméra sténopée*

Il n'a été trouvé qu'une seule espèce biologique utilisant ce principe, c'est le mollusque "Nautilus" (voir figure 1.1.2) dont l'oeil est un trou à ouverture réglable.

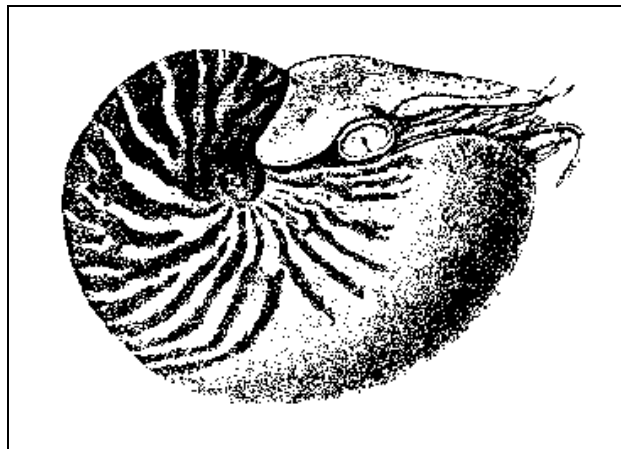


FIG. 1.2 – *L'oeil du Nautilus fonctionne selon le principe sténopé*

D'après des écrits retrouvés en Chine et datant du Vème siècle avant Jésus-Christ, on pense que

le phénomène de formation des images sténopées a été observé dès cette époque. A la Renaissance (voir figure 1.3) on retrouve le début d'une utilisation de caméras sténopées aussi bien au niveau artistique que scientifique (en astronomie, pour l'observation des éclipses). Ces caméras sont encore

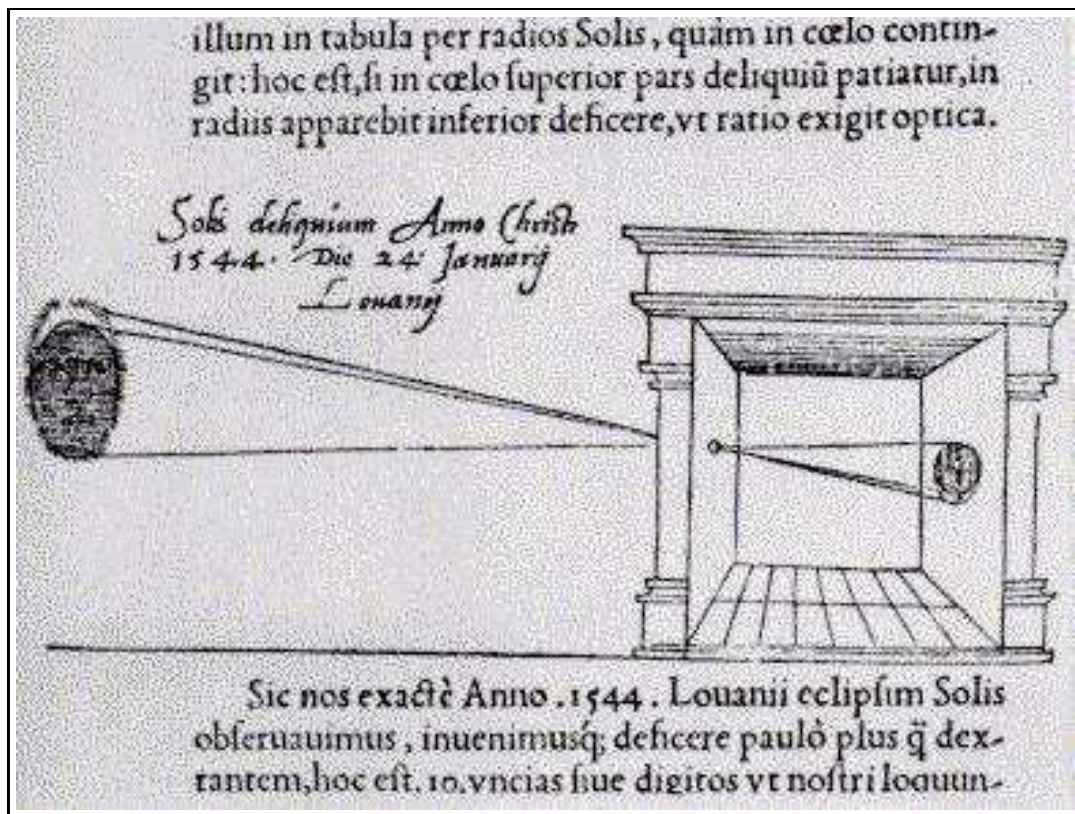


FIG. 1.3 – Schéma de caméra sténopée (dans *De Radio Astronomica et Geometrica*, par Gemma Frisius, 1545), utilisée pour l'observation de l'éclipse de soleil de 1544.

utilisées au XXème siècle pour des applications bien spécifiques en physique nucléaire telles que l'observation de rayons X et gamma en provenance du soleil ou des rayons de hautes énergies dans les plasmas laser.

En ce qui concerne les images dans le spectre visible, ces caméras, bien qu'elles aient l'avantage d'être très simples et de minimiser les distorsions sur des grands angles, fournissent des images dont la netteté n'est pas très bonne, surtout lorsqu'on s'écarte du centre de l'image. Elles possèdent des aberrations chromatiques (l'indice d'un milieu dépend de la longueur d'onde: la vitesse de la lumière dépend de la longueur d'onde). De plus, le temps d'exposition est très long (de une demi seconde à plusieurs heures), ce qui est incompatible avec l'observation d'objets en mouvement.

Par la suite, vers les années 1620, on commença à ajouter une lentille afin d'améliorer la répartition de la lumière dans l'image ainsi que la netteté: ce fût la naissance de la "camera obscura". Puis, progressivement, différents éléments optiques permettant d'améliorer encore la qualité d'image ainsi que d'étendre les possibilités de variation de focale ou de variation de lumière furent ajoutés.

1.1.3 Les composants optiques d'une caméra

Ce paragraphe a pour but de rappeler les éléments nécessaires à la modélisation physique d'une caméra. D'un intérêt culturel, il vise à montrer qu'il est préférable de considérer un modèle de fonctionnement du système optique considéré. Il devient indispensable dans le cas de caméras alternatives abordées au paragraphe 1.4.3. A ce titre, nous allons nous intéresser aux divers composants d'une caméra ainsi qu'aux lois de l'optique auxquelles ils sont soumis.

Les éléments optiques présents dans une caméra sont des lentilles, des miroirs, des prismes, des filtres, des polarisateurs et des mécanismes d'ouverture.

« **dioptrique:** partie de la physique qui traite de la lumière réfractée et des phénomènes qu'elle produit en traversant des milieux de densité différente. »

Le Petit LITTRÉ

1.1.3.1 Les lois de l'optique

La lumière se propage linéairement dans des milieux homogènes d'indice de réfraction $n = \frac{\text{vitesse dans le vide}}{\text{vitesse dans le milieu}} = \frac{c}{v}$. A la frontière entre deux milieux, une partie de la lumière peut être réfractée selon la loi de Snell-Descartes (voir figure 1.4) :

$$n_1 \sin(i_1) = n_2 \sin(i_2)$$

tandis que l'autre partie de la lumière est réfléchiée avec un angle de réflexion identique à l'angle d'incidence.

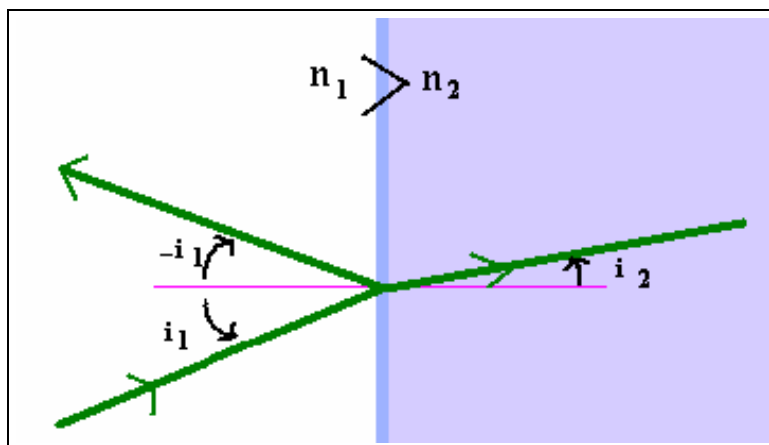


FIG. 1.4 – Réflexion et réfraction entre deux milieux.

Dans le cas où $n_2 < n_1$, on définit un angle critique i_C tel que $\sin(i_C) = \frac{n_2}{n_1}$. Si l'angle d'incidence est supérieur à l'angle critique, il y a réflexion totale, sinon, une partie de la lumière est réfléchiée tandis que l'autre partie est réfractée. La propriété de réflexion totale est utilisée pour une fonction de miroir afin de minimiser les pertes d'énergie (voir figure 1.5 pour un prisme utilisant la réflexion totale).

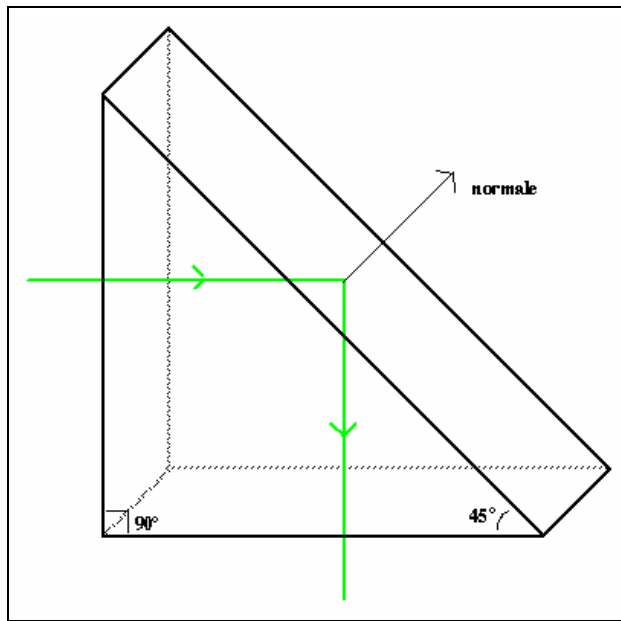


FIG. 1.5 – Schéma de prisme réflecteur

La loi de Snell-Descartes peut être simplifiée lorsqu'on se place dans les conditions de l'approximation de Gauss : le système est centré, les centres des dioptries sont alignés, les milieux sont homogènes et les rayons paraxiaux. On obtient alors la loi de Képler : $n_1 i_1 = n_2 i_2$.

Cette équation permet de déterminer le chemin de la lumière à travers les différents dioptries qui constituent une caméra.

1.1.3.2 Les lentilles

Il existe différentes sortes de lentilles (figure 1.6) : Les lentilles convergentes, (a), (b) et (c),

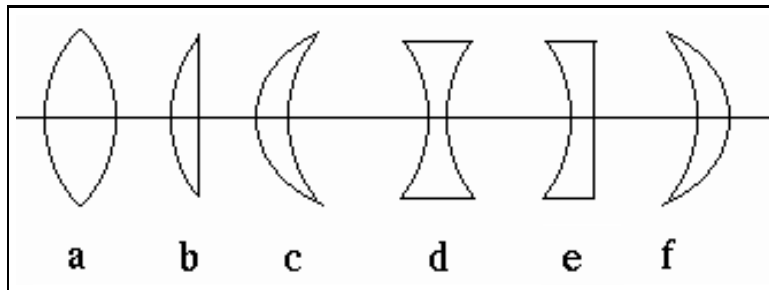


FIG. 1.6 – (a): bi-convexe, (b): plan-convexe, (c): ménisque positif, (d): bi-concave, (e): plan-concave, (f): ménisque négatif

fournissent des images réelles (sur un écran, sur une pellicule photo, sur un plan CCD, ...) tandis que les lentilles divergentes, (d), (e) et (f), fournissent des images virtuelles (que l'on voit, mais que l'on ne peut pas projeter).

On considère qu'une lentille est mince si son épaisseur est négligeable devant les autres grandeurs qui la caractérisent (rayon de courbure, distance focale).

Dans le cas d'une lentille mince, la construction de l'image d'un objet est déterminée par (voir figures 1.7 et 1.8) :

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f}$$

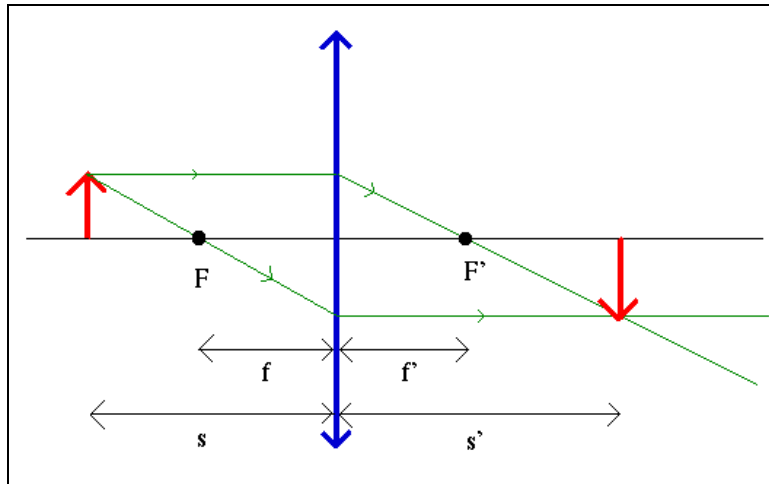


FIG. 1.7 – Schéma de lentille convergente

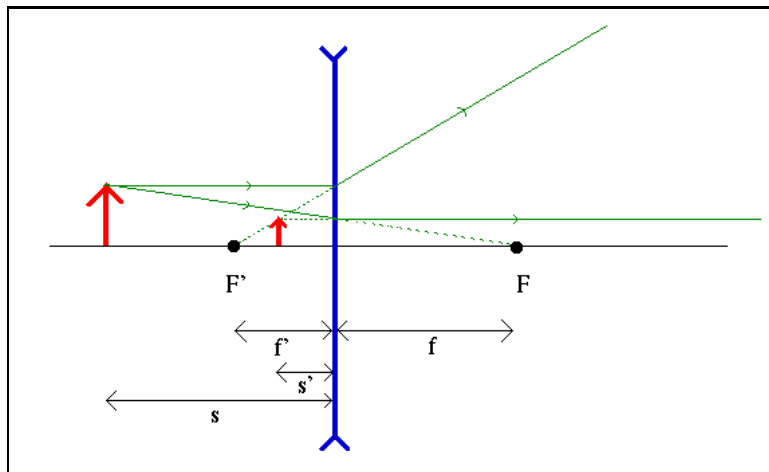


FIG. 1.8 – Schéma de lentille divergente

Dans le cas des lentilles épaisses, il faut considérer les deux dioptries séparément, l'un à la suite de l'autre et combiner les équations de chacun d'eux.

Il faut noter que toutes ces constructions sont des approximations, sachant qu'on a considéré que le terme d'ordre 1 du sinus de la loi de Snell-Descartes. On observe effectivement des aberrations :

aberration sphérique : concerne les objets proches de l'axe optique. L'image d'un point situé sur l'axe optique est une tache sphérique due à la diffraction de la lumière. Sachant que les

aberrations sphériques d'une lentille convergente et d'une lentille divergente se compensent, on pourra atténuer ce type d'aberration en ajoutant à une lentille convergente une lentille divergente ou bien en utilisant plutôt une lentille plan-convexe à la place d'une lentille bi-convexe.

aberration de coma (ou effet de comète) : concerne les objets paraxiaux. Comme pour l'aberration sphérique, l'image d'un point est une tache.

astigmatisme : la surface des lentille ne peut être considérée comme étant exactement sphérique; il en résulte une mauvaise convergence (objets légèrement flous).

distorsions : l'image d'une droite est une courbe (voir 1.4.2 pour plus de détails)

qui correspondent, d'une part, aux différents termes de la série du sinus (voir (Jenkins et White, 1976) pour plus de détails), d'autre part à la géométrie du dioptré de la lentille.

$$\sin(\theta) = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots$$

1.1.3.3 Les miroirs

Les miroirs ont pour propriété de réfléchir toute la lumière qu'ils reçoivent. La construction des images par un miroir mince est montrée sur les figures 1.9 et 1.10. Pour les miroirs épais, il suffit de considérer une lentille épaisse dont une des surfaces est totalement réfléchissante.

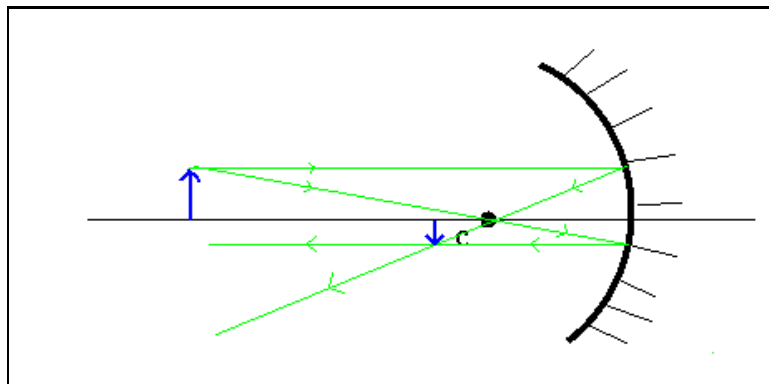


FIG. 1.9 – Schéma de miroir concave

Les prismes On trouve également des prismes dans les caméras dont le principe est montré sur la figure 1.11. La déviation minimale δ_m d'un rayon lumineux par un prisme est donnée par l'équation suivante, obtenue en application la loi de Snell-Descartes sur les surfaces aval et amont du prisme (voir figure 1.11), où n et n' représentent les indices des milieux, α l'angle principal du prisme :

$$\frac{n'}{n} = \frac{\sin(\frac{\alpha + \delta_m}{2})}{\sin(\frac{\alpha}{2})}$$

Si le prisme peut être considéré comme mince, cette équation devient:

$$\frac{n'}{n} = \frac{\delta_m + \alpha}{\alpha}$$

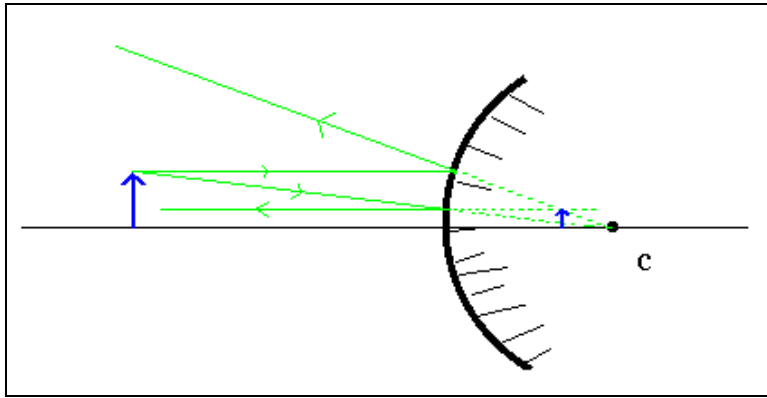


FIG. 1.10 – Schéma de miroir convexe

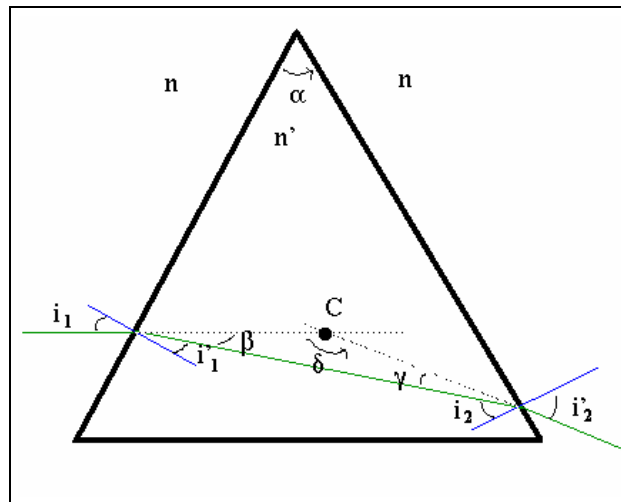


FIG. 1.11 – Schéma de prisme

1.1.3.4 Assemblage des éléments optiques

Ajouter une lentille à une caméra sténopée apporte des améliorations mais les approximations sur les lois de Snell-Descartes ont pour conséquence des aberrations. Or, lorsque l'on combine plusieurs dioptries, les premiers ordres d'aberration disparaissent ce qui a conduit à des caméras un peu plus complexes comme par exemple la caméra Tessar de Zeiss (1902) illustrée par la figure 1.12 sur laquelle on notera l'apparition d'un diaphragme. L'ajout d'autres éléments optiques permet aussi de réaliser un compromis entre le temps d'exposition et un plus grand angle d'observation, de pouvoir faire varier la focale avec plus de liberté, ...

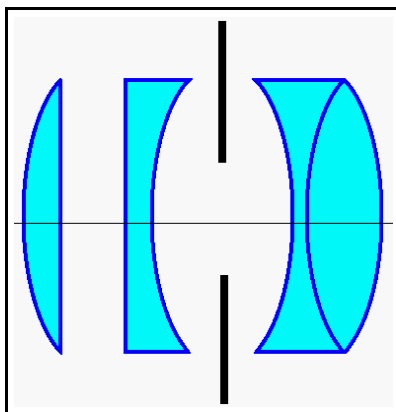


FIG. 1.12 – caméra Tessar, par Zeiss, 1902

Connaissant les équations générales de chaque composant optique de la caméra, on peut déterminer les équations de la caméra. Bien qu'on obtienne alors un modèle en fonction des éléments présents dans la caméra, on sait déjà que ce sont, d'une part, des équations approchées et d'autre part, qu'elles nécessitent une connaissance très précise des positions relatives de ces divers éléments.

Mais bien plus important, on obtient alors une modélisation de la caméra qui comprend un très grand nombre de paramètres et qui est une expression trop complexe pour être utilisée numériquement. Un tel modèle est donc pratiquement inutilisable. D'autre part, il faut également noter qu'il est rare d'avoir accès au schéma interne d'une caméra (confidentialité commerciale, ...) ainsi qu'aux mécanismes mis en jeu lors de variations de zoom par exemple (il y a des mouvements complexes des dioptries les uns par rapport aux autres) ou de mise au point (Bobet, 1994).

A titre d'exemple, le schéma interne de l'objectif SUMMILUX-R f/1.4/50mm de chez Leica (voir http://www.leica-camera.com/r_system/pdf/50summ_e.pdf) est reproduite sur la figure 1.13.

Ces multiples raisons nous conduisent à rechercher un modèle simplifié et utilisable d'une caméra qui représente son fonctionnement *vu de l'extérieur*. Cela est l'objet du paragraphe suivant.

1.1.4 Modélisation d'une caméra

Nous nous intéressons maintenant au modèle le plus simple qui rende compte du comportement d'une caméra. En particulier, on s'attachera à trouver un modèle linéaire, ce qui nous permettra de

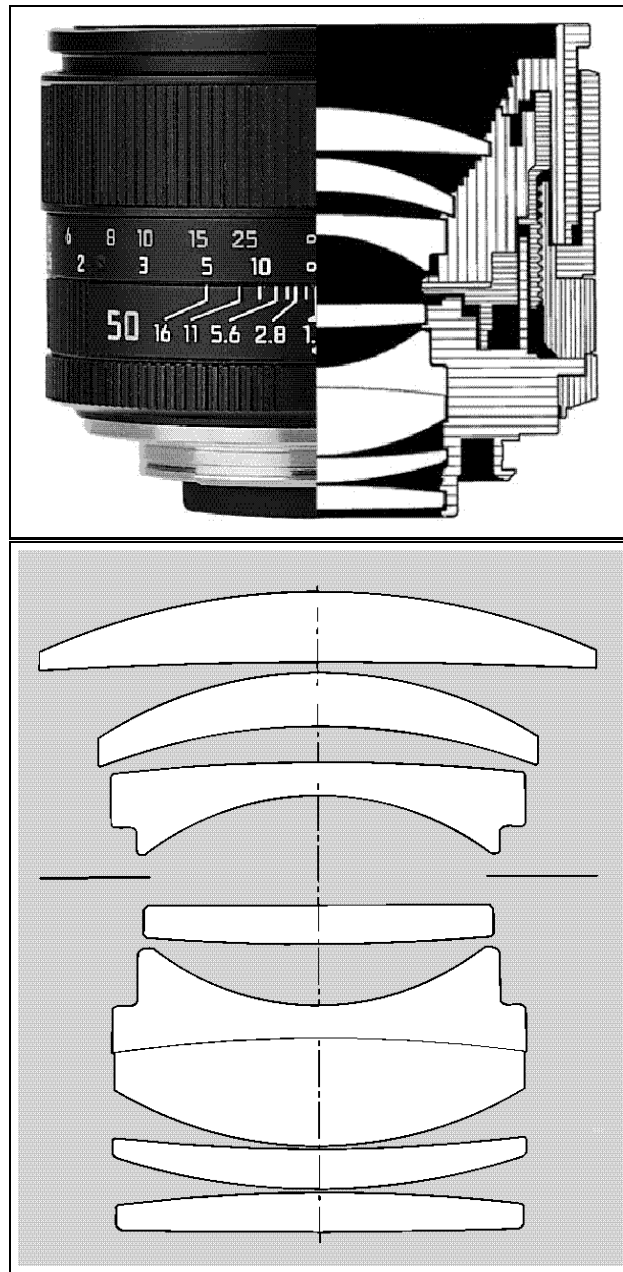


FIG. 1.13 – Le schéma interne de l'objectif "SUMMILUX-R f/1.4/50mm" de chez Leica

le manipuler facilement à l'aide des outils d'algèbre linéaire et de dériver des algorithmes facilement implémentables. Une caméra étant une transformation d'un monde tridimensionnel vers un plan, on pense naturellement à une projection. Il reste alors à déterminer quel type de projection et avec quels paramètres.

1.1.4.1 Modèle projectif linéaire

Ce modèle, appelé également modèle sténopé, consiste à considérer que la caméra effectue une projection perspective de centre C sur un plan rétinien Π . Afin d'exprimer cette projection dans les coordonnées de l'espace et de l'image, il est nécessaire de connaître le changement de repère entre la scène observée et le repère attaché à la caméra. Un point M de l'espace a pour coordonnées (X, Y, Z) dans le repère \mathcal{R} de l'espace. Un point m de l'image a pour coordonnées (u, v) dans le repère image \mathcal{R}_i , déterminé par les axes horizontal et vertical de l'image ainsi que du coin supérieur gauche. Ce repère image induit un repère espace \mathcal{R}' dont le troisième axe est donné par l'axe optique et le centre par le centre optique C . Dans le repère image \mathcal{R}_i , le point C a pour coordonnées $(u_0, v_0, -f)$, où f est la distance focale (distance du plan rétinien au centre optique). Le point M de l'espace a pour coordonnées (X', Y', Z') dans le repère \mathcal{R}' . La transformation entre les repères \mathcal{R} et \mathcal{R}' est la composée d'une rotation \mathbf{R} et d'une translation \mathbf{t} .

La figure 1.14 clarifie ces notations. Les équations de la projection perspective s'écrivent :

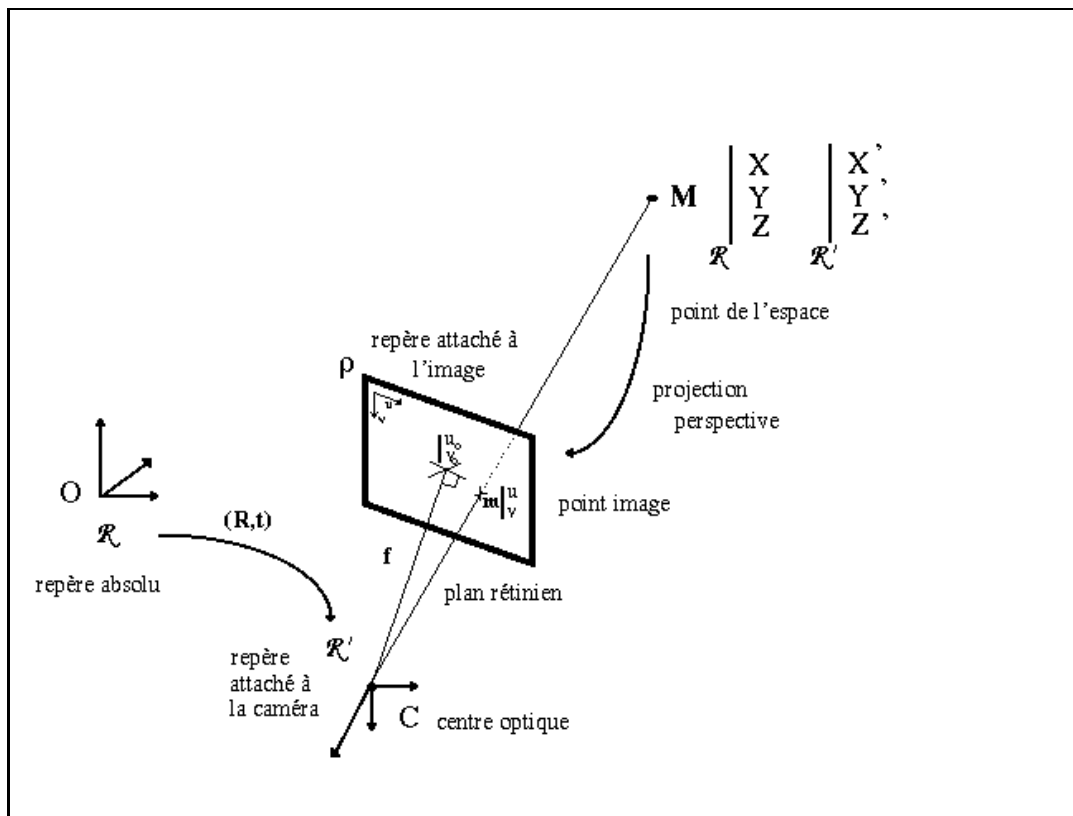


FIG. 1.14 – Le modèle sténopé

$$\begin{cases} u &= u_0 + \alpha_u \frac{X'}{Z'} + \gamma \frac{Y'}{Z'} \\ v &= v_0 + \alpha_v \frac{Y'}{Z'} \end{cases} \quad (1.1)$$

avec

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t}$$

Les paramètres α_u , α_v , γ , u_0 et v_0 seront détaillés au paragraphe 1.3.1.

1.2 La géométrie projective : généralités et applications

La géométrie projective date du début du XX^{ème} siècle et on doit son utilisation massive en vision par ordinateur à Olivier Faugeras et Richard Hartley. Pour une bonne introduction, on conseillera la lecture de (Mohr, 1993) ou du chapitre 2 de (Faugeras, 1993).

Cette géométrie permet l'unification des géométries Euclidienne et Affine qui en deviennent des cas particuliers.

1.2.1 Définitions - Principes

Un espace projectif \mathbb{P}^n peut être défini comme l'ensemble des droites de \mathbb{R}^{n+1} passant par 0. On note $\mathbb{P}^n = \mathcal{P}(\mathbb{R}^{n+1}) = \mathbb{R}^{n+1} \setminus \{0\}$.

Un point projectif x de \mathbb{P}^n est un vecteur de $(n+1)$ éléments dont au moins une des composantes est non nulle :

$$\mathbf{x} = [x_1, x_2, \dots, x_{n+1}]^T \in \mathbb{P}^n \iff \exists i / x_i \neq 0$$

L'égalité entre deux vecteurs de \mathbb{P}^n est définie comme suit :

$$\mathbf{x}' = \mathbf{x} \iff \exists \kappa / \mathbf{x}' = \kappa \mathbf{x}$$

La base canonique de l'espace projectif \mathbb{P}^n est composée de $(n+2)$ vecteurs (la base comporte un élément supplémentaire par rapport à \mathbb{R}^{n+1} à cause de la notion d'égalité, comme justifié dans (Robert, 1993)) :

$$\begin{aligned} & [1, 1, \dots, 1]^T \\ & [1, 0, 0, \dots, 0]^T \\ & \vdots \\ & [0, \dots, 0, \overset{i}{1}, 0, \dots, 0]^T \\ & \vdots \\ & [0, \dots, 0, 1]^T \end{aligned}$$

On définit une collinéation ou homographie \mathbf{c} comme une application linéaire inversible de \mathbb{P}^n vers \mathbb{P}^n . L'ensemble des collinéations définit le groupe projectif.

1.2.2 Application à la vision par ordinateur

En coordonnées homogènes, les points de l'espace à distance finie sont représentés par des vecteurs à 4 coordonnées $\mathbf{M} = [X, Y, Z, 1]^T$ tandis que les points à l'infini sont représentés par $\mathbf{M} = [X, Y, Z, 0]^T$. Les points de l'image à distance finie sont représentés par $\mathbf{m} = [x, y, 1]^T$ et ceux à l'infini par $\mathbf{m} = [x, y, 0]^T$. On obtient ainsi une représentation uniforme des points à l'infini (points de fuite), c'est-à-dire des directions de droite.

Le principe de dualité dans le plan projectif concerne les droites et points qui sont représentés et se manipulent de la même façon. Un point \mathbf{p} est défini comme l'intersection de deux droites \mathbf{d}_1 et \mathbf{d}_2 ; deux points \mathbf{p}_1 et \mathbf{p}_2 définissent une droite \mathbf{d} :

$$\begin{aligned}\mathbf{p} &= \mathbf{d}_1 \wedge \mathbf{d}_2 \\ \mathbf{d} &= \mathbf{p}_1 \wedge \mathbf{p}_2\end{aligned}$$

Ainsi, toutes les opérations effectuées sur les points sont également applicables aux droites.

Une collinéation \mathbf{H} du plan vers le plan (homographie) est représentée par une matrice (3x3) telle que $\kappa \mathbf{m}' = \mathbf{H} \mathbf{m}$, où κ est le coefficient d'égalité projective. Par exemple, un changement de repère image représenté par la composée d'une rotation de matrice \mathbf{R} et translation de vecteur \mathbf{t} s'écrit $\mathbf{m}' = [\mathbf{R}|\mathbf{t}] \mathbf{m}$; les changements de base sont linéaires plutôt qu'affines grâce à l'introduction de la coordonnée supplémentaire.

Un autre intérêt de la géométrie projective est la linéarisation du modèle sténopé. En effet, l'équation (1.1) s'écrit maintenant :

$$Z' \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_u & \gamma & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix}$$

et la relation $\mathbf{M}' = \mathbf{R} \mathbf{M} + \mathbf{t}$ devient $\kappa \mathbf{M}' = [\mathbf{R}|\mathbf{t}] \mathbf{M}$ d'où :

$$\kappa \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{P}} (\mathbf{R}|\mathbf{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

On note couramment \mathbf{P} la composée de la projection et du changement de repère.

1.3 Les projections

Le modèle projectif est le plus employé mais d'autres variantes ont été utilisées par des auteurs tels que Aloimonos, (Aloimonos, 1990), Dementhon et Davis, (Dementhon et Davis, 1989) et reprises par Horaud, (Horaud et al., 1994). Il s'agit des :

- projection para-perspective

- projection ortho-perspective
- projection affine, appelée aussi orthographique, perspective faible (“*weak perspective*”) ou orthographique graduée (“*scaled orthographique*”) selon son paramétrage.

Ces types de projection diffèrent par le nombre de paramètres qu’ils comportent et par leur plus ou moins bonne adéquation aux données observés. Leur analyse est l’objet des paragraphes suivants.

Dans la suite, on se placera dans le repère \mathcal{R}' attaché à la caméra et on notera $\mathbf{M} = [X, Y, Z, \{0, 1\}]$ un point de l’espace dans ce repère.

1.3.1 La projection perspective

La projection perspective (voir figure 1.15), qui est la projection la plus généralement validée, est représentée par l’équation suivante, dans le repère attaché à la caméra :

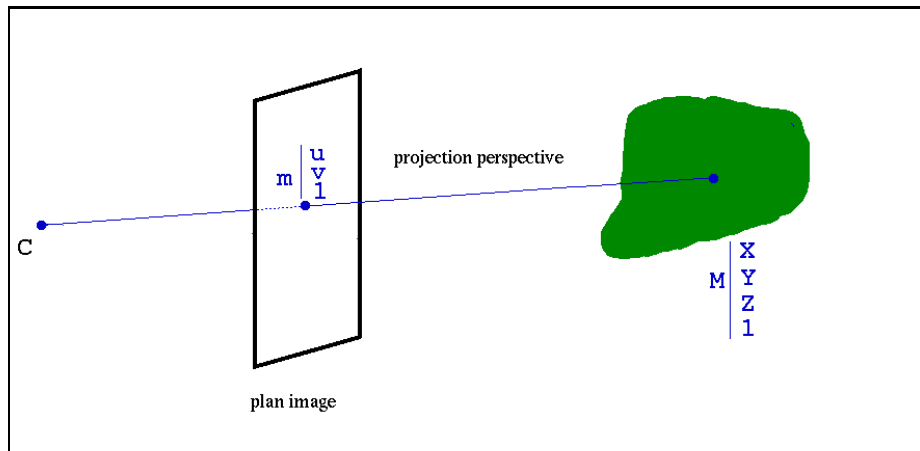


FIG. 1.15 – *Principe de la projection projective*

$$Z \mathbf{m} = \begin{pmatrix} \alpha_u & \gamma & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{M} \quad (1.2)$$

Dans cette projection,

α_u et α_v représentent les tailles horizontale et verticale des pixels. On décompose parfois $\alpha_u = f k_u$ et $\alpha_v = f k_v$ afin d’exhiber que ces variables contiennent la focale. Lorsque α_u et α_v diffèrent, les pixels ne sont pas carrés mais rectangulaires.

u_0 et v_0 représentent la projection du centre optique sur la rétine, le centre de la zone fovéale.

γ mesure le défaut d’orthogonalité du plan des cellules CCD par rapport à l’axe optique. Si γ n’est pas nul, les pixels, au lieu d’être rectangulaires, sont de la forme de parallélogrammes.

1.3.2 La projection para-perspective

La projection para-perspective a été utilisée pour la première fois par Ohta (Ohta et al., 1981) afin de résoudre un problème de reconstruction 3D à partir de textures (*shape from texture*). Elle fut ensuite réutilisée par divers auteurs, notamment (Christy et Horaud, 1994). Le but de ce paragraphe est de montrer, en s'appuyant sur les travaux cités précédemment, que cette approximation est à la fois une approximation géométrique et une approximation au premier ordre des équations de projection perspective.

Ce modèle est bien adapté à la focalisation d'attention autour d'un angle de vue particulier.

1.3.2.1 Approximation géométrique

Cette approximation peut être vue comme une approximation géométrique, c'est-à-dire en supposant que l'objet est loin de la caméra ou proche de l'axe optique. La projection est alors correctement approximée par la composée d'une projection parallèle sur un plan orthogonal à l'axe optique et d'une projection perspective. La direction de la projection parallèle dépend de la scène observée (ou de l'objet observé) et est donnée par la direction du rayon partant du centre optique de la caméra vers le centroïde de la scène, i.e. le point de profondeur moyenne. Le plan image fictif sur lequel sont projetés les points issus de la projection parallèle est parallèle au plan rétinien et passe par le centroïde. La figure 1.16 illustre cette situation qui consiste à regarder "en coin".

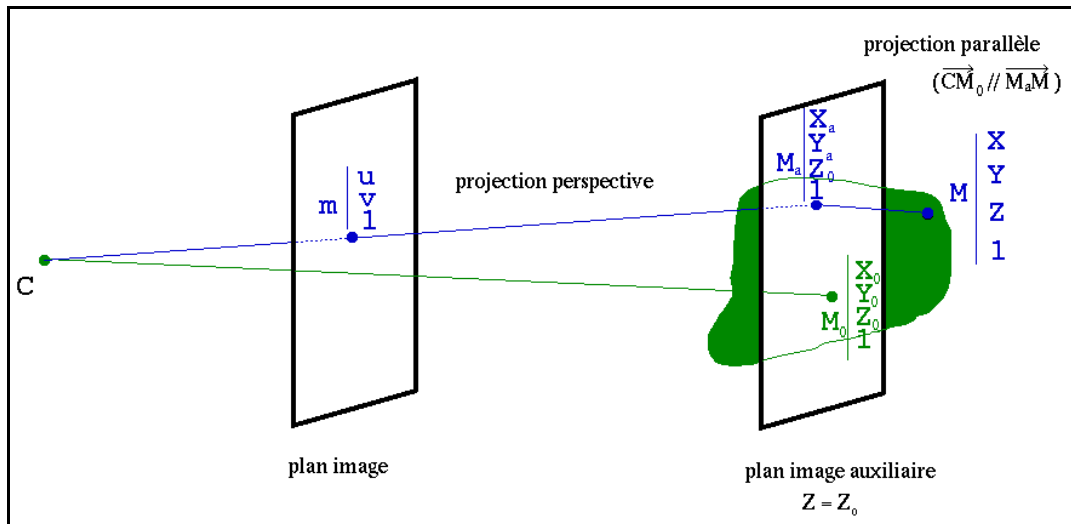


FIG. 1.16 – Principe de la projection para-perspective

Notons $\mathbf{M}_0 = [X_0 \ Y_0 \ Z_0 \ 1]^T$ le point de coordonnées moyennes (et donc de profondeur moyenne). Le plan image auxiliaire a pour équation $Z = Z_0$. Pour un point $\mathbf{M} = [X \ Y \ Z \ 1]^T$ de l'espace, notons $\mathbf{M}_a = [X_a \ Y_a \ Z_0 \ 1]^T$ le point image auxiliaire correspondant.

Ecrivons la relation de parallélisme pour la projection parallèle :

$$\begin{cases} \mathbf{C} \mathbf{M}_0 \parallel \mathbf{M}_a \mathbf{M} \\ \mathbf{M}_a \in \mathcal{P}(Z = Z_0) \end{cases} \Leftrightarrow \begin{cases} X_a = X - \frac{X_0}{Z_0} Z + X_0 \\ Y_a = Y - \frac{Y_0}{Z_0} Z + Y_0 \end{cases}$$

Ecrivons maintenant les équations de projection perspective :

$$\begin{cases} u &= \alpha_u \frac{X_a}{Z_0} + \gamma \frac{Y_a}{Z_0} + u_0 \\ v &= \alpha_v \frac{Y_a}{Z_0} + v_0 \end{cases}$$

Soit, en remplaçant X_a et Y_a par leurs expressions :

$$\begin{cases} u &= \frac{\alpha_u}{Z_0} X + \frac{\gamma}{Z_0} Y - \frac{1}{Z_0} \frac{\alpha_u X_0 + \gamma Y_0}{Z_0} Z + \alpha_u \frac{X_0}{Z_0} + \gamma \frac{Y_0}{Z_0} + u_0 \\ v &= \frac{\alpha_v}{Z_0} Y - \frac{\alpha_v}{Z_0} \frac{Y_0}{Z_0} Z + \alpha_v \frac{Y_0}{Z_0} + v_0 \end{cases}$$

On remarque qu'on obtient alors un modèle linéaire en (X, Y, Z) (la profondeur Z n'apparaît plus au dénominateur) mais, en contre-partie, ce modèle comporte plus de paramètres que le modèle perspectif originel, et ces paramètres dépendent de la scène observée. Sachant qu'il s'agit d'une approximation, on considérera que la variation des paramètres X_0 , Y_0 et Z_0 est faible par rapport aux variations des paramètres internes physiques de la caméra lorsque le mouvement est faible (voir chapitre 2).

En posant :

$$\begin{cases} \alpha_{u_{para}} &= \frac{\alpha_u}{Z_0} \\ \alpha_{v_{para}} &= \frac{\alpha_v}{Z_0} \\ \gamma_{para} &= \frac{\gamma}{Z_0} \\ \beta_{u_{para}} &= -\frac{1}{Z_0} \frac{\alpha_u X_0 + \gamma Y_0}{Z_0} \\ \beta_{v_{para}} &= -\frac{1}{Z_0} \frac{\alpha_v Y_0}{Z_0} \\ u_{0_{para}} &= u_0 + \alpha_u \frac{X_0}{Z_0} + \gamma \frac{Y_0}{Z_0} \\ v_{0_{para}} &= v_0 + \alpha_v \frac{Y_0}{Z_0} \end{cases} \Leftrightarrow \begin{cases} \alpha_u &= \alpha_{u_{para}} Z_0 \\ \alpha_v &= \alpha_{v_{para}} Z_0 \\ \gamma &= \gamma_{para} Z_0 \\ u_0 &= u_{0_{para}} - \beta_{u_{para}} Z_0 \\ v_0 &= v_{0_{para}} - \beta_{v_{para}} Z_0 \\ \frac{X_0}{Z_0} &= \frac{\gamma_{para}}{\alpha_{u_{para}}} \frac{\beta_{v_{para}}}{\alpha_{v_{para}}} - \frac{\beta_{u_{para}}}{\alpha_{u_{para}}} \\ \frac{Y_0}{Z_0} &= -\frac{\beta_{v_{para}}}{\alpha_{v_{para}}} \end{cases} \quad (1.3)$$

En sus des paramètres intrinsèques de la caméra interviennent $\frac{X_0}{Z_0}$ et $\frac{Y_0}{Z_0}$, paramètres représentant la direction $g = [X_0 \ Y_0 \ Z_0]^T$ du regard sur la scène, ainsi que Z_0 au titre de facteur d'échelle.

Les équations de la projection para-perspective s'écrivent alors sous la forme :

$$\begin{cases} u &= \alpha_{u_{para}} X + \gamma_{para} Y + \beta_{u_{para}} Z + u_{0_{para}} \\ v &= \alpha_{v_{para}} Y + \beta_{v_{para}} Z + v_{0_{para}} \end{cases}$$

On remarque que $(u_{0_{para}}, v_{0_{para}})$ représente le projeté dans le plan rétinien du point \mathbf{M}_0 de l'espace, selon une projection perspective et on note également les relations suivantes :

$$\begin{cases} \beta_{u_{para}} &= -\alpha_{u_{para}} \frac{X_0}{Z_0} - \gamma_{para} \frac{Y_0}{Z_0} \\ \beta_{v_{para}} &= -\alpha_{v_{para}} \frac{Y_0}{Z_0} \\ u_{0_{para}} &= u_0 - \beta_{u_{para}} \\ v_{0_{para}} &= v_0 - \beta_{v_{para}} \end{cases} \quad (1.4)$$

On obtient ainsi le modèle para-perspectif qui s'écrit matriciellement de la façon suivante :

$$\mathbf{m} = \begin{pmatrix} \alpha_{u_{para}} & \gamma_{para} & \beta_{u_{para}} & u_{0_{para}} \\ 0 & \alpha_{v_{para}} & \beta_{v_{para}} & v_{0_{para}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{M} \quad (1.5)$$

et qui comporte 2 paramètres supplémentaires par rapport au modèle perspectif, β_u et β_v . Cependant, si l'on se réfère aux relations 1.4, ces paramètres s'expriment en fonction des autres paramètres et de la direction du regard, $(\frac{X_0}{Z_0}, \frac{Y_0}{Z_0})$.

1.3.2.2 Approximation au premier ordre

D'un autre point de vue, on peut considérer que l'objet est loin de la scène et écrire la coordonnée de profondeur Z sous la forme :

$$Z = Z_0 (1 + \varepsilon_Z + o(\varepsilon_Z))$$

ainsi que les autres coordonnées :

$$X = X_0 (1 + \varepsilon_X + o(\varepsilon_X))$$

$$Y = Y_0 (1 + \varepsilon_Y + o(\varepsilon_Y))$$

En écrivant les équations de projection projective, on obtient :

$$\begin{cases} u = \alpha_u \frac{X_0}{Z_0} (1 + \varepsilon_X + o(\varepsilon_X)) \frac{1}{1 + \varepsilon_Z + o(\varepsilon_Z)} + \gamma \frac{Y_0}{Z_0} (1 + \varepsilon_Y + o(\varepsilon_Y)) \frac{1}{1 + \varepsilon_Z + o(\varepsilon_Z)} + u_0 \\ v = \alpha_v \frac{Y_0}{Z_0} (1 + \varepsilon_Y + o(\varepsilon_Y)) \frac{1}{1 + \varepsilon_Z + o(\varepsilon_Z)} + v_0 \end{cases}$$

Or, en utilisant le développement limité suivant, au premier ordre au voisinage de 0 :

$$\frac{1}{1 + \varepsilon} = 1 - \varepsilon + o(\varepsilon) \text{ où } \frac{o(\varepsilon)}{\varepsilon} \xrightarrow{\varepsilon \rightarrow 0} 0$$

et en l'appliquant à $\varepsilon_Z + o(\varepsilon_Z)$, on fait disparaître les termes en Z du dénominateur, soit :

$$\begin{cases} u = \alpha_u \frac{X_0}{Z_0} (1 + \varepsilon_X + o(\varepsilon_X)) (1 - \varepsilon_Z + o(\varepsilon_Z)) + \gamma \frac{Y_0}{Z_0} (1 + \varepsilon_Y + o(\varepsilon_Y)) (1 - \varepsilon_Z + o(\varepsilon_Z)) + u_0 \\ v = \alpha_v \frac{Y_0}{Z_0} (1 + \varepsilon_Y + o(\varepsilon_Y)) (1 - \varepsilon_Z + o(\varepsilon_Z)) + v_0 \end{cases}$$

Au premier ordre, il vient :

$$\begin{cases} u = \alpha_u \frac{X_0}{Z_0} (1 + \varepsilon_X - \varepsilon_Z) + \gamma \frac{Y_0}{Z_0} (1 + \varepsilon_Y - \varepsilon_Z) + u_0 \\ v = \alpha_v \frac{Y_0}{Z_0} (1 + \varepsilon_Y - \varepsilon_Z) + v_0 \end{cases}$$

Or, toujours au premier ordre :

$$\begin{cases} \varepsilon_X = \frac{X}{X_0} - 1 \\ \varepsilon_Y = \frac{Y}{Y_0} - 1 \\ \varepsilon_Z = \frac{Z}{Z_0} - 1 \end{cases}$$

D'où les équations suivantes :

$$\begin{cases} u = \frac{\alpha_u}{Z_0} X + \frac{\gamma}{Z_0} Y - \frac{1}{Z_0} \frac{\alpha_u X_0 + \gamma Y_0}{Z_0} Z + u_0 + \frac{\alpha_u X_0 + \gamma Y_0}{Z_0} \\ v = \frac{\alpha_v}{Z_0} Y - \frac{1}{Z_0} \frac{\alpha_v Y_0}{Z_0} Z + v_0 + \frac{\alpha_v Y_0}{Z_0} \end{cases}$$

On retrouve ainsi les expressions linéaires en les coordonnées du point \mathbf{M} de l'espace mais dont les coefficients dépendent d'un point "milieu" de la scène \mathbf{M}_0 .

1.3.3 La projection ortho-perspective

Cette approximation de la projection perspective, inventée par Dementhon et Davis (Dementhon et Davis, 1989), consiste en une projection orthogonale sur un plan image auxiliaire dont la normale est donnée par l'angle sous lequel le point de profondeur moyenne \mathbf{M}_0 est vu depuis le centre optique de la caméra. La figure 1.17 illustre ce principe et les notations associées.

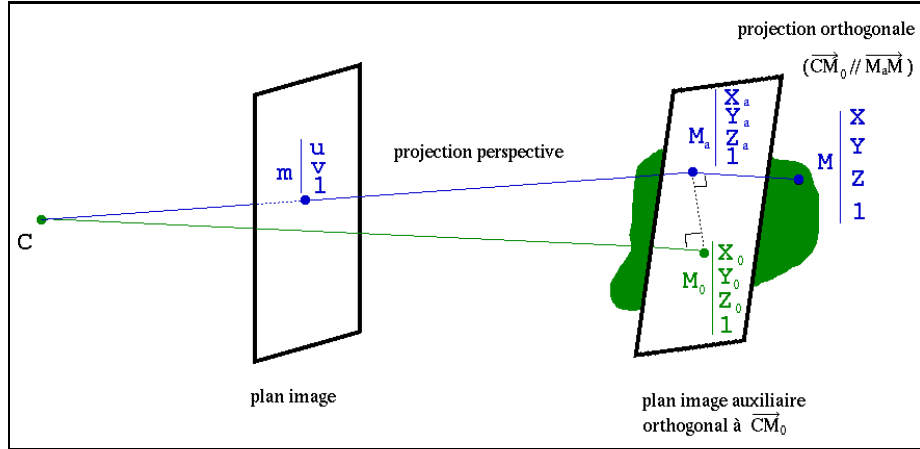


FIG. 1.17 – Principe de la projection ortho-perspective

Les propriétés de cette projection sont le parallélisme des vecteurs $\mathbf{C M}_0$ et $\mathbf{M}_a \mathbf{M}$ et l'orthogonalité des vecteurs $\mathbf{M}_a \mathbf{M}_0$ et $\mathbf{M}_a \mathbf{M}$, ce qui se traduit par :

$$\begin{cases} \mathbf{C M}_0 \parallel \mathbf{M}_a \mathbf{M} \\ \mathbf{M}_a \mathbf{M}_0 \perp \mathbf{M}_a \mathbf{M} \end{cases} \Leftrightarrow \begin{cases} X_0 (X_a - X_0) + Y_0 (Y_a - Y_0) + Z_0 (Z_a - Z_0) = 0 \\ \frac{X - X_a}{X_0} = \frac{Y - Y_a}{Y_0} = \frac{Z - Z_a}{Z_0} \end{cases}$$

Avec un peu de calcul, on en déduit les coordonnées du point auxiliaire \mathbf{M}_a :

$$\begin{cases} X_a = X_0 + X \frac{Y_0 + Z_0}{X_0^2 + Y_0^2 + Z_0^2} - Z \frac{X_0 Z_0}{X_0^2 + Y_0^2 + Z_0^2} - Y \frac{X_0 Y_0}{X_0^2 + Y_0^2 + Z_0^2} \\ Y_a = Y_0 + Y \frac{X_0 + Z_0}{X_0^2 + Y_0^2 + Z_0^2} - X \frac{Y_0 X_0}{X_0^2 + Y_0^2 + Z_0^2} - Z \frac{Y_0 Z_0}{X_0^2 + Y_0^2 + Z_0^2} \\ Z_a = Z_0 + Z \frac{X_0 + Y_0}{X_0^2 + Y_0^2 + Z_0^2} - X \frac{X_0 Z_0}{X_0^2 + Y_0^2 + Z_0^2} - Y \frac{Y_0 Z_0}{X_0^2 + Y_0^2 + Z_0^2} \end{cases}$$

Ainsi, en reportant dans les équations de la projection perspective les expressions des coordonnées de \mathbf{M}_a , on obtient :

$$\begin{cases} u = u_0 + \frac{(-\alpha_u (Y_0^2 + Z_0^2) + \gamma X_0 Y_0) \mathbf{X} + (\alpha_u X_0 Y_0 - \gamma (X_0^2 + Z_0^2)) \mathbf{Y} + (\alpha_u X_0 Z_0 + \gamma Y_0 Z_0) \mathbf{Z} - (X_0^2 + Y_0^2 + Z_0^2) (\alpha_u X_0 + \gamma Y_0)}{X_0 Z_0 \mathbf{X} + Y_0 Z_0 \mathbf{Y} - (X_0^2 + Y_0^2) \mathbf{Z} - Z_0 (X_0^2 + Y_0^2) - Z_0^3} \\ v = v_0 + \alpha_v \frac{X_0 Y_0 \mathbf{X} - (X_0^2 + Z_0^2) \mathbf{Y} - (X_0^2 + Y_0^2 + Z_0^2) Y_0}{X_0 Z_0 \mathbf{X} + Y_0 Z_0 \mathbf{Y} - (X_0^2 + Y_0^2) \mathbf{Z} - Z_0 (X_0^2 + Y_0^2) - Z_0^3} \end{cases}$$

Cette approximation, bien qu'intéressante d'un point de vue géométrique n'apporte aucune simplification dans les équations, bien au contraire : elle nécessite des coefficients supplémentaires, le dénominateur des équations est un polynôme en les coordonnées tridimensionnelles X , Y et Z .

Au vue de cette complexité et sachant que les conditions de validité de cette projection sont similaires à celles de la projection para-perspective, on préférera cette dernière, à moins, comme

dans (Aloimonos, 1990), de considérer une approximation de cette projection ortho-perspective : la projection orthographique qui apporte quant à elle de nombreux avantages et constitue l'objet du paragraphe suivant.

1.3.4 La projection orthographique ou affine

On rencontre également cette approximation sous les noms de perspective faible (*“weak perspective”*) ou plus exactement orthographique graduée (*“scaled orthographique”*), la projection orthographique étant un cas particulier de cette dernière pour lequel on ne tient absolument pas compte de la profondeur. Nous utiliserons par la suite le terme de projection orthographique désignant l'ensemble des projections orthographiques graduées ou non.

Cette approximation de la projection perspective a été utilisée par de nombreux auteurs comme (Aloimonos, 1990), (Christy et Horaud, 1994), et bien d'autres encore (voir le paragraphe 2.2.1 pour une bibliographie plus détaillée).

Ce modèle est bien adapté à la vision fovéale.

1.3.4.1 Approximation géométrique

Il s'agit d'un cas particulier à la fois de l'approximation para-perspective et de l'approximation ortho-perspective dans le sens que la projection perspective est approximée par la composée d'une projection orthogonale sur un plan auxiliaire qui est perpendiculaire à l'axe optique; c'est également une projection para-perspective dont la direction est orthogonale au plan image auxiliaire et une projection ortho-perspective dont le plan est orthogonale à l'axe optique de la caméra.

Plus concrètement, il s'agit d'une projection para-perspective pour laquelle la direction du regard et la même que celle de l'axe optique de la caméra.

La figure 1.18 illustre cette composée et précise les notations.

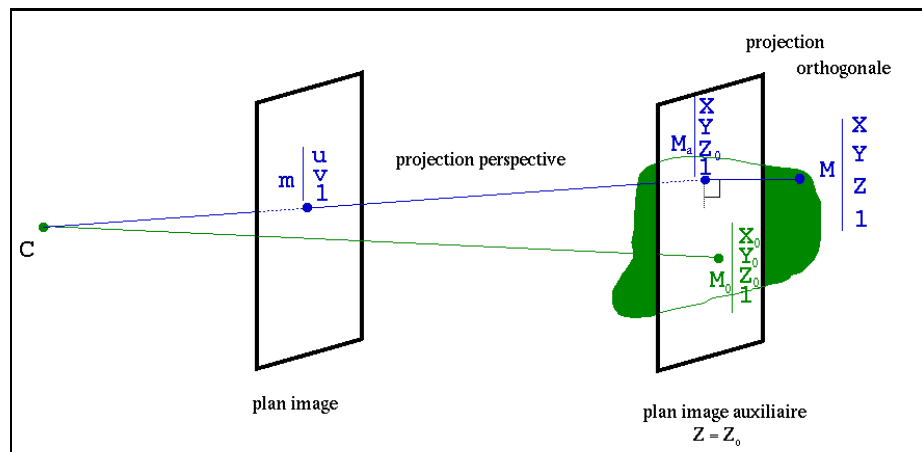


FIG. 1.18 – Principe de la projection orthographique

Les relations géométriques sont les suivantes :

$$\begin{cases} \mathbf{C} \mathbf{M}_0 \parallel \mathbf{M}_a \mathbf{M} \\ \mathbf{M}_a \mathbf{M}_0 - \mathbf{M}_a \mathbf{M} \\ \mathbf{M}_a \in \mathcal{P}(Z = Z_0) \end{cases} \Leftrightarrow \begin{cases} X_0 (X_a - X_0) + Y_0 (Y_a - Y_0) + Z_0 (Z_a - Z_0) = 0 \\ \frac{X - X_a}{X_0} = \frac{Y - Y_a}{Y_0} = \frac{Z - Z_a}{Z_0} \\ Z_a = Z_0 \end{cases}$$

Soit :

$$\begin{cases} X_a = X \\ Y_a = Y \\ Z_a = Z_0 \end{cases}$$

Après projection, il vient :

$$\begin{cases} u = \frac{\alpha_u}{Z_0} X + \frac{\gamma}{Z_0} Y + u_0 \\ v = \frac{\alpha_v}{Z_0} Y + v_0 \end{cases}$$

On remarque alors qu'on obtient des équations linéaires en les coordonnées X et Y mais, contrairement à l'approximation para-perspective, il n'a pas été nécessaire d'introduire des paramètres supplémentaires. Cependant, les paramètres intrinsèques dépendent, tout comme pour la projection para-perspective, de la profondeur moyenne de la scène.

Ce modèle est équivalent au modèle para-perspectif dna le cas où on regarde droit devant soi (vision fovéale), c'est-à-dire le cas où $X_0 = 0$ et $Y_0 = 0$.

1.3.4.2 Approximation à l'ordre 0

D'un point de vue équationnel, il s'agit d'un développement à l'ordre 0 en la profondeur Z du modèle perspectif, $Z = Z_0 + o(1)$:

$$\begin{cases} u = \alpha_u \frac{X}{Z_0} + \gamma \frac{Y}{Z_0} + u_0 \\ v = \alpha_v \frac{Y}{Z_0} + v_0 \end{cases}$$

Soit, en notant :

$$\begin{cases} \alpha_{u_{ortho}} = \frac{\alpha_u}{Z_0} \\ \alpha_{v_{ortho}} = \frac{\alpha_v}{Z_0} \\ \gamma_{ortho} = \frac{\gamma}{Z_0} \end{cases}$$

sous forme matricielle :

$$\mathbf{m} = \begin{pmatrix} \alpha_{u_{ortho}} & \gamma_{ortho} & 0 & u_0 \\ 0 & \alpha_{v_{ortho}} & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{M} \quad (1.6)$$

On retrouve bien le modèle orthographique, ce qui nous montre l'équivalence entre l'approximation géométrique et l'approximation à l'ordre 0.

L'avantage de cette projection est de nous procurer des équations affines ce qui simplifie énormément les calculs par la suite. Bien plus restreint que le modèle para-perspectif, en général, ce modèle est cependant bien réaliste dans des cas particuliers précis que nous allons étudier maintenant.

1.3.5 Différents modèles en une équation

Nous résumons le principe de la projection perspective et de ses approximations sur la figure 1.19.

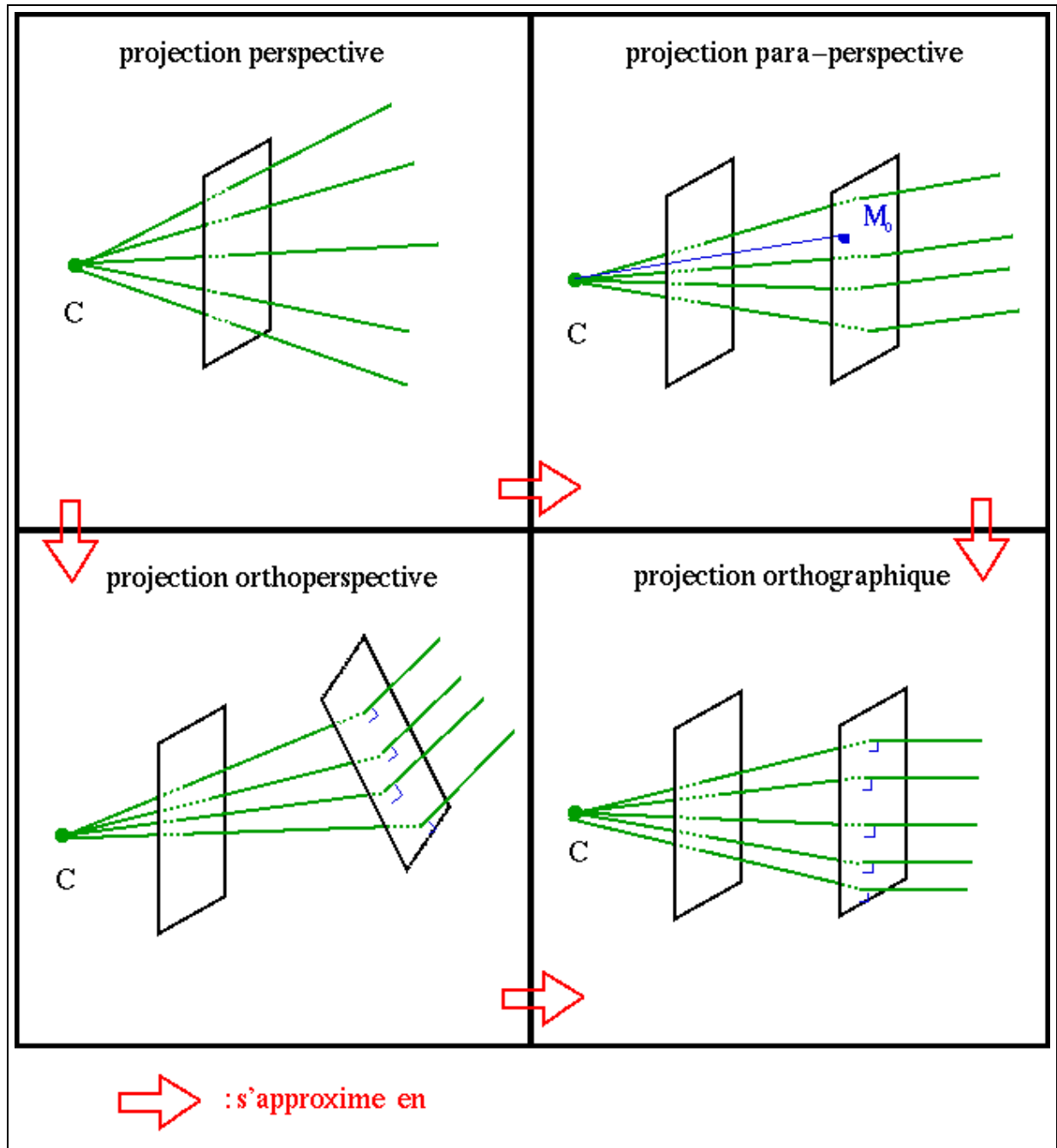


FIG. 1.19 – La projection perspective et ses approximations.

Le modèle perspectif est le plus général. Il peut être approximé, lors de conditions expérimentales similaires en une projection para-perspective ou ortho-perspective. Ces deux approximations peuvent être à leur tour approximées par la projection orthographique.

Compte-tenu de la pauvreté en simplification algébrique de la projection perspective, celle-ci n'est utilisée dans la littérature, voir par exemple (Aloimonos, 1990), que dans le cas de son approximation orthographique. D'autre part, des auteurs ayant développés des algorithmes dans le cas spécifique de la projection orthographique, (Tomasi et Kanade, 1991), (Poelman et Kanade, 1993) et (Koenderinck, 84), ont réussi à généraliser leurs méthodes au cas para-perspectif, ce qui n'était pas faisable pour le cas ortho-perspectif. Enfin, la projection ortho-perspective ayant le même domaine de validité que la projection para-perspective en étant plus compliquée que celle-ci, nous ne considérons, dans la suite que les approximations para-perspective ou orthographique.

Le modèle para-perspectif a l'avantage de linéariser les équations de projection mais possède le désavantage d'introduire deux paramètres supplémentaires. Le modèle orthographique réunit les avantages en linéarisant les équations sans faire intervenir de paramètre supplémentaire.

Ces deux modèles font intervenir la scène par l'intermédiaire d'un point de profondeur moyenne mais conservent tous deux le facteur d'échelle (les objets proches paraissent plus grands que les objets lointains).

Cependant, le modèle para-perspectif reste tout de même plus précis que le modèle orthographique, car il tient compte des positions des objets en périphérie (vus sous un angle différent).

Nous proposons ici un modèle général de projection pouvant englober les modèles vus précédemment (1.2, 1.5, 1.6), en notant \hat{x} la variable générique x :

$$\kappa \mathbf{m} = \begin{pmatrix} \hat{\alpha}_u & \gamma & \lambda \hat{\beta}_u + \mu u_0 & (1 - \mu) \hat{u}_0 \\ 0 & \hat{\alpha}_v & \lambda \hat{\beta}_v + \mu v_0 & (1 - \mu) \hat{v}_0 \\ 0 & 0 & \mu & (1 - \mu) \end{pmatrix} \mathbf{M} \quad (1.7)$$

Dans ce modèle, on distingue deux types de paramètres :

- **Paramètres modaux** : λ et μ
- **Paramètres intrinsèques** : $\alpha_u, \alpha_v, \beta_u, \beta_v, u_0, v_0$ et γ

Les paramètres modaux $\lambda \in \{0,1\}$ et $\mu \in \{0,1\}$ déterminent le modèle de projection :

	λ	μ
Projection perspective pure	1	1
Projection orthographique	0	0
Projection para-perspective	1	0

Et nous récapitulons dans le tableau suivant les valeurs des paramètres intrinsèques pour les approximations de la projection perspective :

	projection perspective	projection para-perspective	projection orthographique
$\hat{\alpha}_u$	α_u	$\frac{\alpha_u}{Z_0}$	$\frac{\alpha_u}{Z_0}$
$\hat{\alpha}_v$	α_v	$\frac{\alpha_v}{Z_0}$	$\frac{\alpha_v}{Z_0}$
$\hat{\beta}_u$	0	$-\frac{1}{Z_0} \frac{\alpha_u X_0 + \gamma Y_0}{Z_0}$	0
$\hat{\beta}_v$	0	$-\frac{1}{Z_0} \frac{\alpha_v Y_0}{Z_0}$	0
$\hat{\gamma}$	γ	$\frac{\gamma}{Z_0}$	$\frac{\gamma}{Z_0}$
\hat{u}_0	u_0	$u_0 + \alpha_u \frac{X_0}{Z_0} + \gamma \frac{Y_0}{Z_0}$	u_0
\hat{v}_0	v_0	$v_0 + \alpha_v \frac{Y_0}{Z_0}$	v_0

1.4 Raffinements possibles

1.4.1 Le modèle des lentilles épaisses

Principalement utilisé dans le cas de la modélisation du zoom, le modèle de lentille épaisse (figure 1.20) nécessite de déterminer des paramètres (position des plans principaux, points focaux,

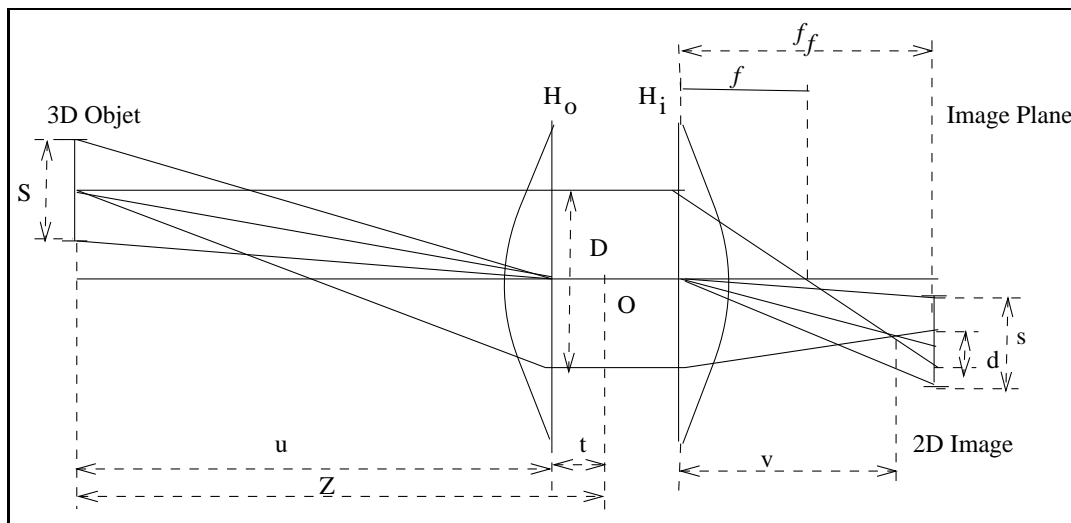


FIG. 1.20 – *Modèle de lentille épaisse, d'après (Pahlavan et al., 1993)*

positions d'entrée et de sortie des lentilles et position du centre de la lentille) avec un dispositif particulier et des instruments très précis, ce qu'il est impossible de faire pour les caméras du commerce comme discuté précédemment.

Heureusement, le modèle de lentille épaisse peut être ré-exprimé en un modèle sténopé (Lavest, 1992) mais Lavest a mis en évidence que le modèle sténopé est erroné sauf si la distance de l'objet à l'image est considérée comme variable et que l'image de l'objet reste dans un voisinage restreint de la rétine.

Nous ne considérerons donc pas le modèle de lentille épaisse dans la suite mais on considérera les paramètres intrinsèques comme étant des fonctions de la scène observée ce qui est déjà le cas

pour les restrictions du modèle projectif comme vu au paragraphe 1.3. Nous tenterons par ailleurs de rester le plus possible dans la zone fovéale de l'image pour une meilleure adéquation de notre modèle.

1.4.2 Prise en compte des distorsions

Les distorsions linéaires ont déjà été prises en compte dans le modèle de projection perspective. Nous nous intéresserons donc ici aux distorsions non-linéaires dont les causes proviennent de défauts de géométrie du système optique : mauvais alignement des centres optiques des dioptries, courbure non parfaite des lentilles,

Nous avons déjà présenté quelques types de distorsions au paragraphe 1.1.3.2: distorsions géométriques, astigmatisme, courbure de champ et aberration du coma. En pratique, c'est la distorsion géométrique radiale symétrique qui a l'influence la plus importante.

Les distorsions géométriques radiales les plus couramment observées sont de deux formes, comme illustré par la figure 1.21 :

- distorsions en forme de barillet
- distorsions en forme de coussinet

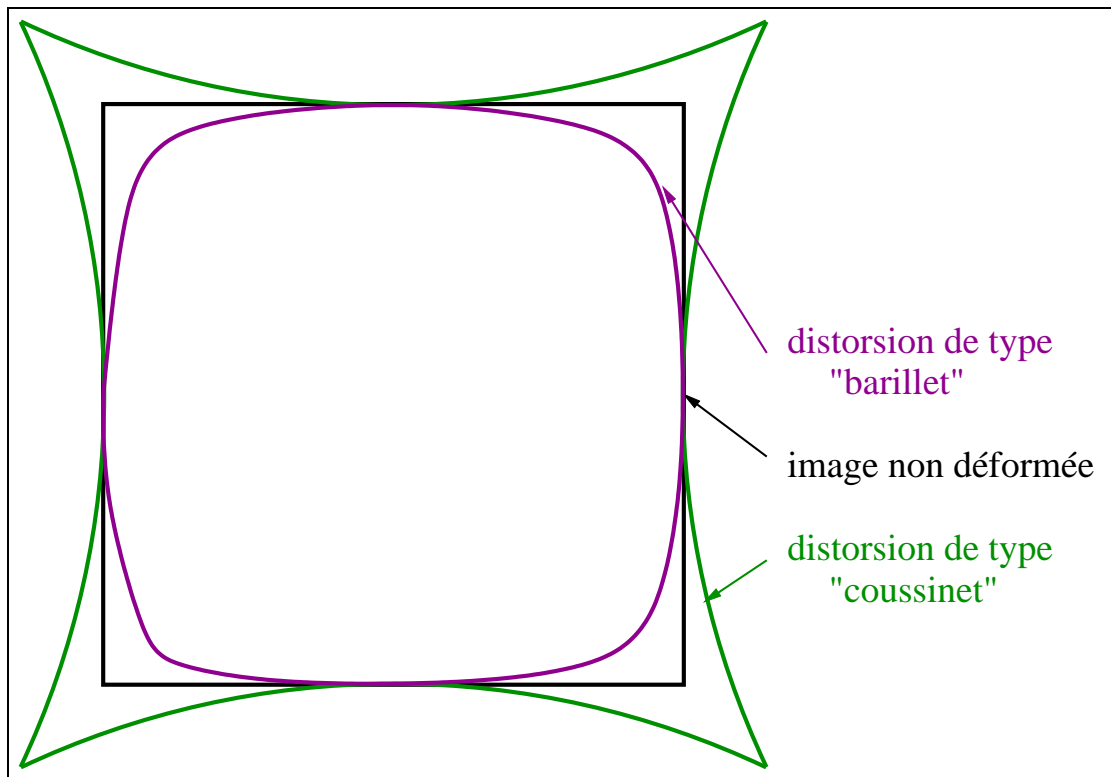


FIG. 1.21 – Les types de distorsions les plus couramment observés : barillet et coussinet

On découple généralement la distorsion en deux composantes radiale et tangentielle. Si on note ρ et θ les coordonnées polaires d'un point (x,y) de l'image, $\delta_\rho(\rho,\theta)$ et $\delta_\theta(\rho,\theta)$ les composantes radiale

et tangentielle de la distorsion, $\delta_x(x,y)$ et $\delta_y(x,y)$, les composantes selon x et y de la distorsion, s'expriment au premier ordre selon :

$$\begin{cases} \delta_x(x,y) &= \cos(\theta) \delta_\rho(\rho,\theta) - \sin(\theta) \delta_\theta(\rho,\theta) \\ \delta_y(x,y) &= \sin(\theta) \delta_\rho(\rho,\theta) + \cos(\theta) \delta_\theta(\rho,\theta) \end{cases}$$

Les principales causes de distorsions sont généralement au nombre de trois. La première concerne les défauts de courbure des lentilles constituant la caméra qui induisent des distorsions purement radiales. Cette déformation de la courbure est cependant utilisée dans le but d'avoir une luminosité constante sur toute la surface de l'image, afin d'éliminer l'effet de vignetage (luminosité forte au centre, plus faible en périphérie). Une approximation suffisante en pratique pour modéliser est donnée par :

$$\begin{cases} \delta_x(x,y) &= k_1 x (x^2 + y^2) \\ \delta_y(x,y) &= k_1 y (x^2 + y^2) \end{cases}$$

k_1 étant une constante dont le signe détermine le sens de distorsion ($k_1 > 0$ correspond aux distorsions en forme de coussinet; $k_1 < 0$ aux distorsions en forme de barillet)

Une deuxième cause de distorsion concerne le mauvais alignement des centres optiques des lentilles à l'intérieur de la caméra qui induit une distorsion de décentrage. Une approximation est donnée par :

$$\begin{cases} \delta_x(x,y) &= p_1 (3x^2 + y^2) + 2p_2 xy \\ \delta_y(x,y) &= 2p_1 xy + p_2 (x^2 + 3y^2) \end{cases}$$

Et enfin, une dernière cause est l'inclinaison des lentilles les unes par rapport aux autres, phénomène équivalent à l'ajout d'un prisme mince, donnant lieu à des distorsions dites prismatiques. Une approximation est donnée par :

$$\begin{cases} \delta_x(x,y) &= s_1 (x^2 + y^2) \\ \delta_y(x,y) &= s_2 (x^2 + y^2) \end{cases}$$

La distorsion totale s'exprime alors au premier ordre comme la somme de ces distorsions, soit :

$$\begin{cases} \delta_x(x,y) &= k_1 x (x^2 + y^2) + p_1 (3x^2 + y^2) + 2p_2 xy + s_1 (x^2 + y^2) \\ \delta_y(x,y) &= k_1 y (x^2 + y^2) + 2p_1 xy + p_2 (x^2 + 3y^2) + s_2 (x^2 + y^2) \end{cases}$$

Ce modèle de distorsion comporte 5 paramètres. Si on envisage le problème dans sa globalité, il faut tenir compte de ces 5 paramètres supplémentaires. Dans notre cas, on préfère découpler les problèmes, sachant que ceux-ci sont décorrélés, et effectuer dans un premier temps la correction des distorsions afin d'appliquer ensuite un modèle de projection linéaire.

D'autre part, l'effet des distorsions étant minime dans la zone fovéale de l'image, on pourra la négliger si on se concentre sur cette zone.

Pour plus de détails, on pourra se référer à (Slama, 1980), (Brand et al., 1993) ou (Devernay, 97).

1.4.3 Différents types de caméras

Les problèmes que l'on cherche à résoudre en Vision par Ordinateur sont de nature assez diverses et d'autres alternatives sont proposées, permettant d'adapter les capteurs à la tâche de vision considérée.

Prenons l'exemple de la navigation d'un robot dans un monde comportant des obstacles. Une reconstruction 3D très précise de l'environnement n'est pas nécessaire car seule une mesure de distance des objets les plus proches est nécessaire. Par contre, une rapidité de calcul est impérative. On peut alors choisir un algorithme simple et efficace donnant une reconstruction même grossière de l'environnement (Viéville et al., 1993).

Ces problèmes ont cependant été abordés par certains auteurs selon un axe tout à fait différent : ils ont cherché à construire une caméra adaptée à un problème donné en fournissant des données appropriées au problème considéré (voir figure 1.23).

les lentille demi-sphérique ont été introduites par (Cao et al., 1986), ces lentilles offrent une vue panoramique pour une surveillance sur un grand angle (exemple au fond d'une piscine). Par contre, elles ne permettent pas la détection d'obstacle car justement, ceux-ci apparaissent en périphérie des images et la résolution est alors mauvaise.

le miroir conique a été introduit par (Jarvis et Byrne, 1988) puis implémenté par (Yagi et al., 1994).

En exprimant les coordonnées des points 3D dans un repère cylindrique $\mathbf{M}(r, \theta, Z)$ et celles des points 2D dans un repère polaire $\mathbf{m}(\rho, \theta)$ (voir figure 1.22), on obtient les équations suivantes :

$$\begin{aligned}\rho &= Z \tan(\alpha) + Z \tan\left(\frac{\pi}{2} - 2\alpha\right) \\ &= \frac{Z}{\sin(2\alpha)}\end{aligned}$$

Pour obtenir un champ de vue le plus large possible, on prend $\alpha = \frac{\pi}{4}$ et obtient alors la relation suivante : $\rho = Z$. Par contre, la distance d'un objet à l'axe du cône n'intervient pas dans la projection.

Ce système est donc dédié à l'évitement d'obstacle verticaux posés sur le sol mais ne permet pas d'évitement d'obstacles généraux et ne se prête absolument pas à la reconstruction tridimensionnelle.

le miroir à double courbure a été introduit récemment par (Southwell et al., 1996), ce miroir à pour but la reconstruction tridimensionnelle de scène et d'éviter de projeter les objets sur de trop petites portions d'image. Pour le moment, la qualité des images est faible et les distorsions grandes.

le miroir parabolique a pour but la vision panoramique ainsi que l'évitement d'obstacle ou la reconstruction tridimensionnelle. Cette caméra est disponible dans le commerce (voir par exemple : <http://www.cyclovision.com/>) et fournit des images de bonne qualité.

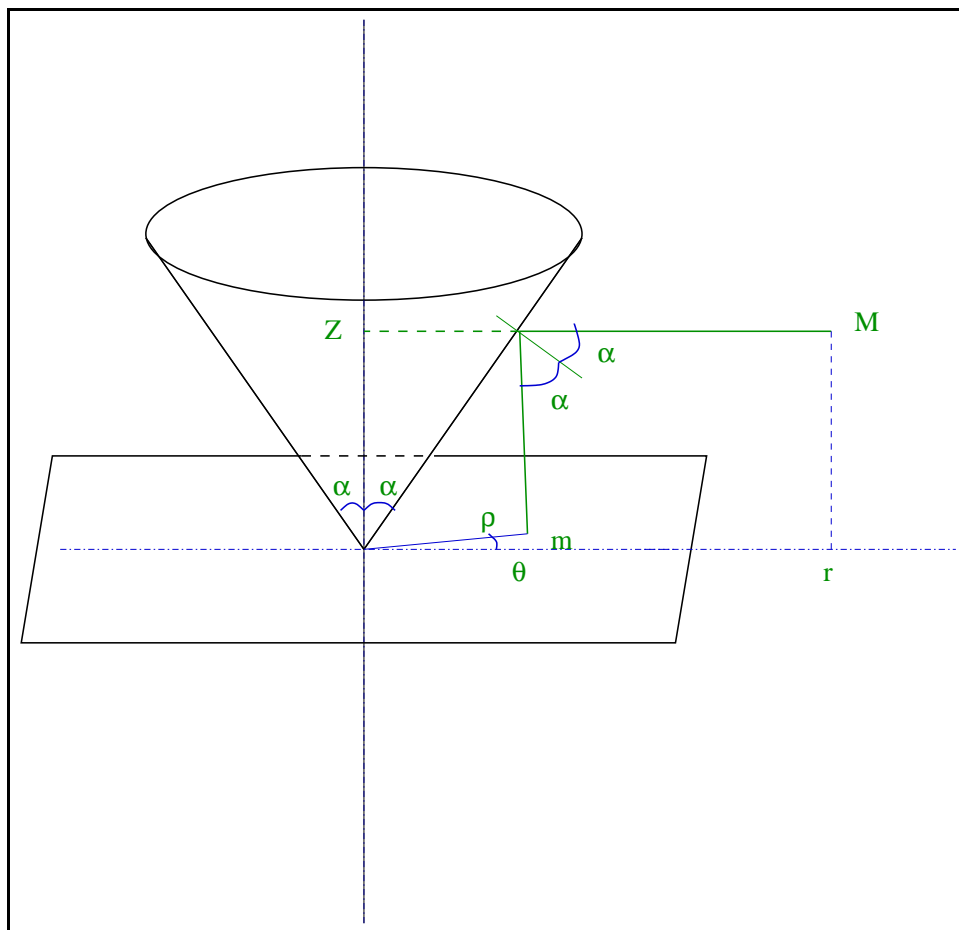


FIG. 1.22 – Miroir conique: notations et calcul des angles

des barettes CCD en rotation autour d'un axe commun : ce système (Manière, 1997; Benosman, 1997) a été conçu récemment dans un but de reconstruction intérieure de scènes tridimensionnelles. Outre la qualité des images, ce système possède l'avantage de faire de la stéréo d'une dimension de moins que dans le cas des caméras classiques puisque les barettes CCD sont suffisamment précisément alignées pour réduire l'espace de recherche à une colonne de l'image. Le cas de la géométrie épipolaire en dimension 1 a été traité en détail dans la thèse (Benosman, 1997) ainsi que par d'autres auteurs comme (Faugeras et al., 98).

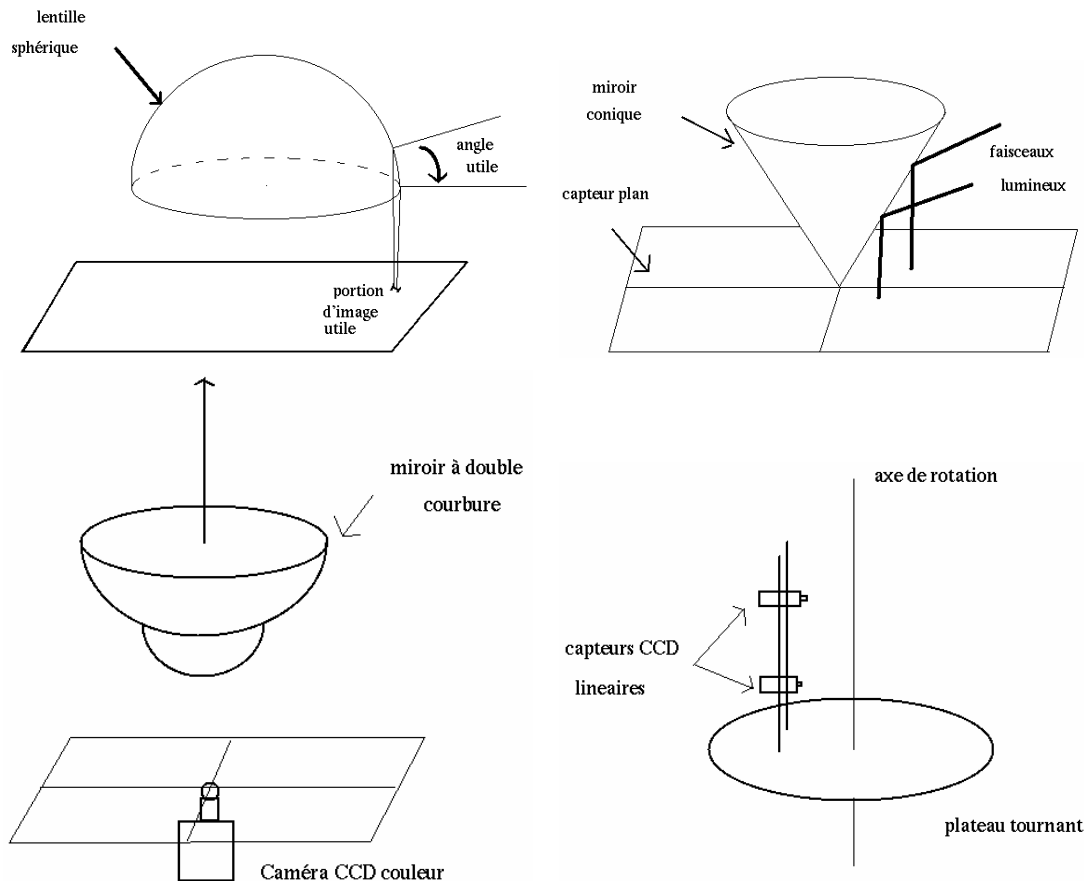


FIG. 1.23 – de gauche à droite et de haut en bas : (a) Lentille sphérique; (b) Miroir conique; (c) Miroir à double courbure; (d) Barettes CCD en rotation;

Dans la suite, nous ne considérerons que les caméras ou appareils photos standards que l'on trouve dans le commerce à base d'éléments à transfert de charge (CCD).

1.5 La paramétrisation du mouvement

1.5.1 Mouvement rigide

On suppose que notre scène est rigide ou bien constituée d'objets rigides. On verra au chapitre 4 que l'on peut éliminer de nos données les amers visuels correspondants à des objets non-rigides.

La contrainte de rigidité impose que les déplacements d'une position \mathbf{M} vers une position \mathbf{M}' sont des translations, exprimées par un vecteur $\mathbf{t} = [t_0 \ t_1 \ t_2]^T$ et des rotations, exprimées par une matrice orthogonale positive \mathbf{R} , composées selon :

$$\mathbf{M}' = \mathbf{R} \mathbf{M} + \mathbf{t}$$

1.5.2 Paramétrisation de la rotation

Une matrice de rotation \mathbf{R} dans l'espace est une matrice carrée (3x3) qui a pour propriété :

$$\mathbf{R}^{-1} = \mathbf{R}^T$$

ce qui fournit 6 contraintes indépendantes sur les coefficients de la matrice de rotation. Ainsi, \mathbf{R} ne dépend plus que de 3 paramètres. Il existe plusieurs façons pour choisir ces paramètres, ce qui est l'objet des paragraphes suivants.

1.5.2.1 Angles d'Euler

Cette paramétrisation est la plus naturelle car chacun des paramètres correspond à la rotation autour d'un des trois axes du repère, comme définis sur la figure 1.24.

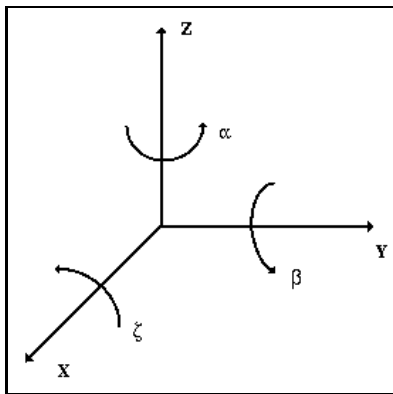


FIG. 1.24 – Les angles d'Euler

La matrice de rotation s'exprime alors comme la composée de ces trois rotations dites canoniques :

$$\mathbf{R} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \zeta & -\sin \zeta \\ 0 & \sin \zeta & \cos \zeta \end{bmatrix}$$

L'apparente simplicité de cette représentation cache la complexité de la relation qui permet de déterminer les paramètres α (lacet), β (tangage) et γ (roulis) à partir d'une matrice de rotation \mathbf{R} . Par exemple, l'ordre des matrices intervient d'où différentes définitions non compatibles (coordonnées sphériques, ...). Mais plus important encore, comme montré en détails dans (Ayache, 1989), il faut

un atlas de 4 cartes avec les paramètres d'Euler pour paramétriser l'ensemble des rotations exceptées celle pour lesquelles $\beta = \pm \frac{\pi}{2}$ et que l'on ne peut pas paramétriser ainsi.

Ainsi, un nombre important de rotations, que l'on ne veut pas s'interdire dans des applications de vision-robotique, ne sont pas représentables par les angles d'Euler.

Nous allons alors nous intéresser à d'autres représentations.

1.5.2.2 Exponentielle de vecteur

On note $\boldsymbol{\omega} = [\omega_0 \ \omega_1 \ \omega_2]^T$ le vecteur rotation dont la norme correspond à l'angle de rotation θ et s'exprime par $\theta = \|\boldsymbol{\omega}\|$, tandis que l'axe de rotation a pour vecteur directeur unitaire : $\mathbf{u} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$. L'opérateur produit vectoriel avec $\boldsymbol{\omega}$ s'écrit :

$$\tilde{\boldsymbol{\omega}} = [\boldsymbol{\omega}]_{\times} = \boldsymbol{\omega} \wedge \cdot = \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix} \wedge \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{bmatrix}$$

La matrice de rotation \mathbf{R} s'exprime alors par (voir (Hochschild, 1968)) :

$$\mathbf{R} = e^{\tilde{\boldsymbol{\omega}}} = e^{[\boldsymbol{\omega}]_{\times}}$$

On vérifie aisément que cette expression est celle d'une rotation :

$$\mathbf{R} \mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1$$

Cependant, nous devons ajouter 3 cartes de paramètres (Ayache, 1989) pour représenter l'ensemble des rotations, y compris celles qui sont des retournements ($\theta = \pi$).

Si on développe cette expression en utilisant le développement en série de l'exponentielle, les puissances de $\tilde{\boldsymbol{\omega}}$ supérieures à 2 s'expriment en fonction de $(I, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\omega}}^2)$ (car on peut facilement montrer que $\tilde{\boldsymbol{\omega}}^3 = -\theta^2 \tilde{\boldsymbol{\omega}}$) et on obtient une représentation sous forme de trois termes, connue sous le nom de formule de Rodrigues (Rodrigues, 1840), et dont les coefficients sont eux-mêmes des séries entières trigonométriques :

$$\mathbf{R} = \mathbf{I} + \frac{\sin \theta}{\theta} \tilde{\boldsymbol{\omega}} + \frac{(1 - \cos \theta)}{\theta^2} \tilde{\boldsymbol{\omega}}^2$$

soit :

$$\mathbf{R} = \mathbf{I} + \sin \theta \tilde{\mathbf{u}} + (1 - \cos \theta) \tilde{\mathbf{u}}^2$$

Compte-tenu de la nécessité de plusieurs cartes de paramètres pour cette représentation, on propose alors de choisir le vecteur rotation de la façon suivante :

$$\mathbf{r} = 2 \tan\left(\frac{\theta}{2}\right) \mathbf{u} \Leftrightarrow \theta = 2 \arctan\left(\frac{\|\mathbf{r}\|}{2}\right) \quad (1.8)$$

car on obtient alors une formule de même type que la formule de Rodrigues, mais qui, cette fois, est rationnelle :

$$\mathbf{R} = \mathbf{I} + \left[\frac{\tilde{\mathbf{r}} + \frac{1}{2}\tilde{\mathbf{r}}^2}{1 + \frac{\mathbf{r}^T \cdot \mathbf{r}}{4}} \right] \quad (1.9)$$

Les développements limités pour de faible rotation ($\theta \rightarrow 0$) ou des rotations proches des retournements ($\theta \rightarrow \pi$) s'écrivent de façon simple :

$$\begin{aligned} \mathbf{R} &= \mathbf{I} + \tilde{\mathbf{r}} + \frac{1}{2}\tilde{\mathbf{r}}^2 + o(\|\mathbf{r}\|^2) \\ \mathbf{R} &= \mathbf{I} + \frac{2\tilde{\mathbf{r}}^2}{\|\mathbf{r}\|^2} + o\left(\frac{1}{\|\mathbf{r}\|^2}\right) \end{aligned}$$

Le développement limité en $(\frac{1}{\|\mathbf{r}\|})^2$ permet de modéliser le comportement de la modélisation de la rotation lorsque son angle se rapproche de π et donc que la tangente du demi-angle tend vers l'infini. Dans ce cas, on obtient effectivement l'expression d'un retournement d'axe \mathbf{u} :

$$\mathbf{R} = \mathbf{I} + 2\tilde{\mathbf{u}}^2$$

d'où :

$$\begin{aligned} \mathbf{R} \mathbf{x} &= \mathbf{x} + 2\mathbf{u} \wedge (\mathbf{u} \wedge \mathbf{x}) \\ &= 2(\mathbf{u} \cdot \mathbf{x})\mathbf{u} - \mathbf{x} \end{aligned}$$

Cette représentation nous permet donc de paramétrer toutes les rotations de l'espace y compris celles qui sont des retournements. Il est donc naturel, si l'on doit tenir compte de ces retournements, de réécrire \mathbf{r} en coordonnées homogènes sous la forme de $\epsilon \mathbf{q}$:

$$\mathbf{q} = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{u} \right) = (s, \mathbf{v})$$

L'équation 1.9 devient alors rationnelle :

$$\mathbf{R} = \mathbf{I} + 2s\tilde{\mathbf{v}} + 2\tilde{\mathbf{v}}^2$$

1.5.2.3 Représentation par quaternions

C'est à Hamilton que nous devons l'invention des quaternions au siècle dernier, précurseurs des vecteurs. Leur utilisation première fut l'expression de la rotation ce qui ne fut pas tâche aisée (Altmann, 1989).

Un quaternion \mathbf{q} peut s'exprimer de deux façons différentes par :

- $\mathbf{q} = [\lambda_0 \ \lambda_1 \ \lambda_2 \ \lambda_3]^T$ où $\lambda_0, \lambda_1, \lambda_2$ et λ_3 , sont des scalaires

- $\mathbf{q} = (s, \mathbf{v})$ où s est un scalaire, \mathbf{v} un vecteur : $\mathbf{v} = [v_0 \ v_1 \ v_2]^T$

Le produit \times sur les quaternions est défini par :

$$\mathbf{q} \times \mathbf{q}' = (s s' - \mathbf{v} \cdot \mathbf{v}', s \mathbf{v}' + s' \mathbf{v} + \mathbf{v} \wedge \mathbf{v}')$$

le conjugué par :

$$\bar{\mathbf{q}} = (-s, \mathbf{v})$$

et la norme par :

$$\|\mathbf{q}\|^2 = \bar{\mathbf{q}}\mathbf{q} = \mathbf{q}\bar{\mathbf{q}} = (s^2 + \|\mathbf{v}\|^2, 0) = (\|\mathbf{q}\|^2, 0)$$

Dans le cas de la rotation, $\|\mathbf{q}\| = 1$.

La rotation d'un vecteur χ par le quaternion \mathbf{q} s'exprime par :

$$\mathbf{R}\chi = \mathbf{q} \times \chi \times \bar{\mathbf{q}}$$

En notant θ l'angle de rotation et \mathbf{u} le vecteur directeur unitaire de l'axe de rotation, comme précédemment :

$$\begin{cases} s &= \cos\left(\frac{\theta}{2}\right) \\ \mathbf{v} &= \sin\left(\frac{\theta}{2}\right) \mathbf{u} \end{cases}$$

On peut remarquer ici que cela reste vrai pour $\theta = 0$ ou $\theta = \pi$, c'est-à-dire pour toutes les rotations.

Cette représentation est équivalente à la représentation précédente car la matrice de rotation s'exprime par :

$$\begin{bmatrix} \lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2 & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & \lambda_0^2 - \lambda_1^2 + \lambda_2^2 - \lambda_3^2 & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & \lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2 \end{bmatrix}$$

sous la contrainte $\lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2 = 1$, expression qui se simplifie en :

$$\mathbf{R} = \mathbf{I} + 2s\tilde{\mathbf{v}} + 2\tilde{\mathbf{v}}^2$$

qui correspond à une des formes de la formule de Rodrigues.

1.5.2.4 Analyse des diverses représentations

Nous avons exclu le choix de la représentation par les angles d'Euler pour les raisons évoquées au paragraphe 1.5.2.1.

Notre choix s'est porté sur l'expression de la matrice de rotation \mathbf{R} en fonction du vecteur rotation \mathbf{r} défini par l'équation (1.8), cette expression ayant l'avantage d'être en fraction de polynômes contrairement aux autres représentations proposées.

Il est à noter que notre choix a été motivé par des raisons de commodité, en particulier lors des développements algébriques, mais il a été démontré dans (Lorusso et al., 1995) que la représentation de la rotation n'a pas d'influence numérique, sauf lorsque l'on s'approche du voisinage d'un point critique.

Ainsi, r_0, r_1, r_2, t_0, t_1 et t_2 , constituent les paramètres extrinsèques. Il faut cependant noter qu'ils ne constituent pas forcément un ensemble de paramètres indépendants (voir 1.6.4.1 par exemple)

1.6 Paramétrisation de 2 vues : la géométrie épipolaire

Afin de simplifier les notations, on va considérer par la suite que le repère de l'espace et celui de la caméra pour la première image sont confondus. On sait alors se ramener à n'importe quel repère de l'espace par une transformation rigide (rotation et translation), cas particulier de collinéation. On notera, dans le cas d'un mouvement de la scène \mathbf{R} et \mathbf{t} la rotation et la translation effectuée par la scène ou, dans le cas d'un mouvement propre de la caméra, \mathbf{R}^T et $-\mathbf{t}$ la rotation et la translation effectuées par la caméra entre les deux images. Si plusieurs objets de la scène sont en mouvement, on ne considèrera dans un premier temps qu'un seul objet à la fois.

1.6.1 Géométrie épipolaire: généralités

On note (voir figure 1.25 pour plus de clarté) \mathbf{C}_1 et \mathbf{C}_2 les centres optiques des caméras 1 et 2, \mathbf{m}_1 et \mathbf{m}_2 les images de \mathbf{M} . L'image de la droite $(\mathbf{C}_1\mathbf{M})$ (resp. $(\mathbf{C}_2\mathbf{M})$) par la caméra 2 (resp. 1) est la droite épipolaire $(\mathbf{e}_2\mathbf{m}_2)$ (resp. $(\mathbf{e}_1\mathbf{m}_1)$) dont \mathbf{e}_2 (resp. \mathbf{e}_1), appelé épipôle, est le point d'intersection avec l'axe $(\mathbf{C}_1\mathbf{C}_2)$. Si on fait varier le point \mathbf{M} , les droites épipolaires correspondantes vont varier, mais les épipôles \mathbf{e}_1 et \mathbf{e}_2 restent fixes (voir figure 1.26).

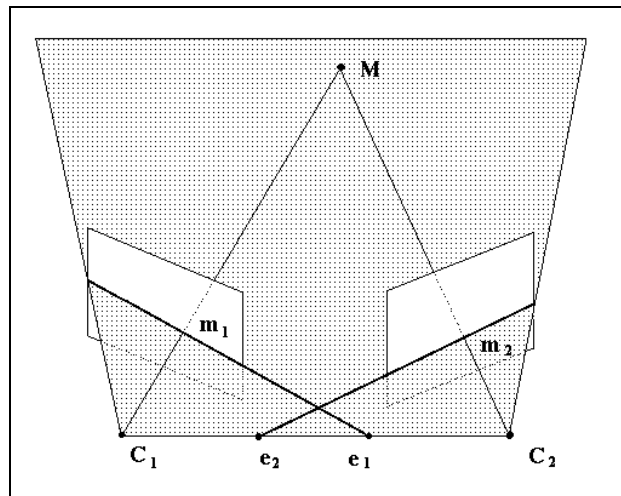


FIG. 1.25 – Géométrie épipolaire : Notations

La géométrie épipolaire implique naturellement certaines contraintes :

contrainte d'unicité : chaque point de l'image admet au plus un correspondant dans la seconde image (et vice-versa), voir figure 1.27. Cette contrainte est respectée lorsque la scène est opaque (il n'y a pas de transparence).

contrainte de continuité : on suppose la profondeur de la scène continue (figure 1.28). En pratique, il s'agira d'une continuité par morceaux car des ruptures ont lieu lors de l'occlusion d'un objet par un autre.

contrainte d'ordre : l'ordre des points image est conservé d'une image à l'autre (figure 1.29). Cette contrainte est respectée uniquement lors de faibles variations de profondeur par rapport

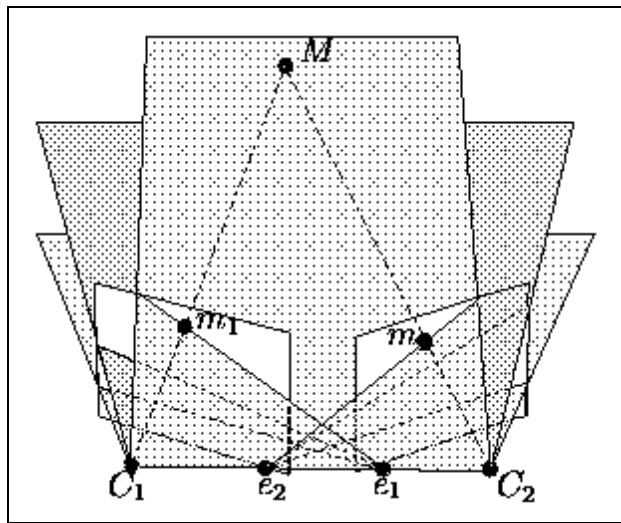


FIG. 1.26 – *Principe de la géométrie épipolaire*

à l'écart entre les caméras.

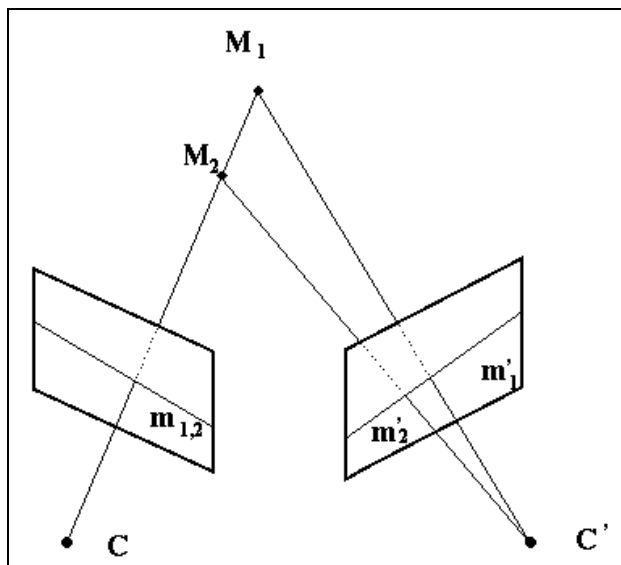


FIG. 1.27 – *Violation de la contrainte d'unicité: le point \mathbf{m} de la première image admet deux points \mathbf{m}'_1 et \mathbf{m}'_2 pour correspondants dans la deuxième image, ceci par transparence.*

1.6.2 La disparité vectorielle

Nous allons maintenant nous intéresser à la disparité vectorielle qui représente la mesure du déplacement rétinien. C'est, de fait, la première quantité liée au mouvement et mesurable entre

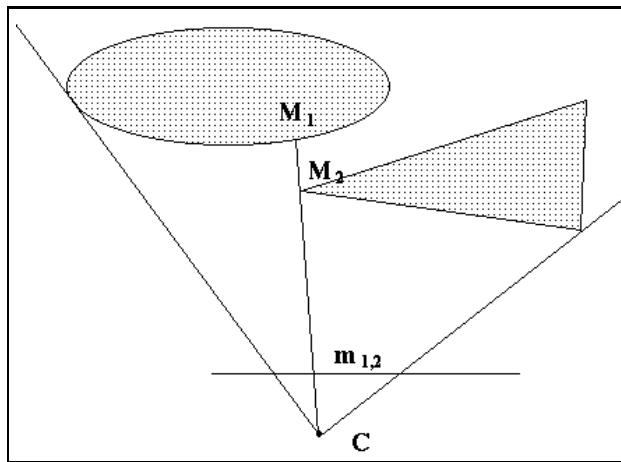


FIG. 1.28 – Violation de la contrainte de continuité: il y a une discontinuité de profondeur en M_1 et M_2 car il y a occlusion d'un objet par un autre.

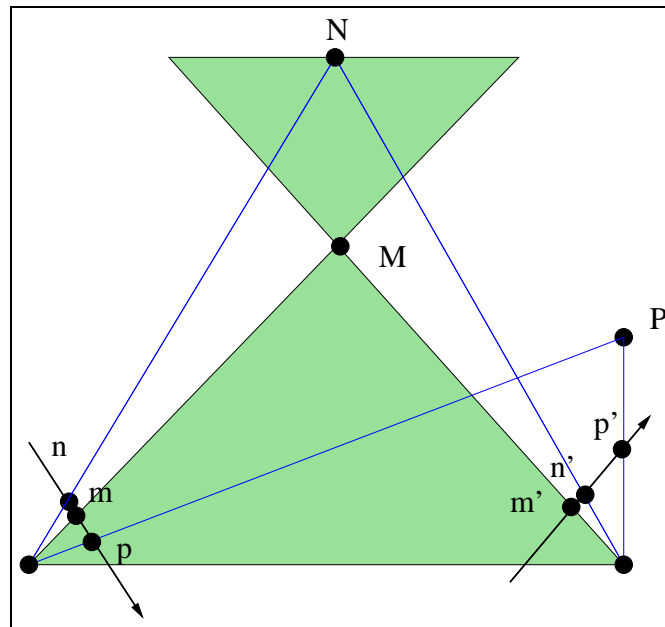


FIG. 1.29 – Violation de la contrainte d'ordre: sur la première image, m est entre n et p , sur la seconde, n' est entre m' et p' . La différence de profondeur des points M , N et P est importante par rapport à l'écart des centres optiques des caméras.

deux images, dès que les appariements ont été établis. Elle est définie de la façon suivante :

$$\begin{cases} du &= u_2 - u_1 \\ dv &= v_2 - v_1 \end{cases}$$

où $\mathbf{m}_1 = [u_1 \ v_1 \ 1]^T$ et $\mathbf{m}_2 = [u_2 \ v_2 \ 1]^T$ sont les projetés dans les images 1 et 2 respectivement.

Dans le cas de la projection perspective, la disparité vectorielle entre deux images s'exprime par :

$$\begin{cases} u &= u_0 + \alpha_u \frac{X}{Z} + \gamma \frac{Y}{Z} \\ v &= v_0 + \alpha_v \frac{Y}{Z} \end{cases}$$

Sachant que les équations du mouvement s'écrivent :

$$\begin{cases} X_2 &= r_{11} X_1 + r_{12} Y_1 + r_{13} Z_1 + t_0 \\ Y_2 &= r_{21} X_1 + r_{22} Y_1 + r_{23} Z_1 + t_1 \\ Z_2 &= r_{31} X_1 + r_{32} Y_1 + r_{33} Z_1 + t_2 \end{cases}$$

et, en approximant localement une surface par un morceau de plan de normale $\mathbf{N} = [N_x \ N_y \ -1]^T$ (ne contenant pas l'axe optique) et de distance à l'origine Z_N :

$$Z_1 = N_X X_1 + N_Y Y_1 + Z_N$$

on obtient ainsi une expression de la disparité vectorielle sous la forme :

$$\begin{cases} du &= [C_u + a u + b v + u (e u + g v)] / [C_d - e u - g v] \\ dv &= [C_v + c u + d v + v (e u + g v)] / [C_d - e u - g v] \end{cases}$$

Les valeurs des coefficients sont données en annexe (voir A.1).

1.6.2.1 Disparité vectorielle dans le cas de la projection para-perspective

On effectue le même raisonnement que précédemment mais en utilisant les équations de la projection para-perspective :

$$\begin{cases} u &= u_0 + \alpha_u X + \gamma Y + \beta_u Z \\ v &= v_0 + \alpha_v Y + \beta_v Z \end{cases}$$

et on obtient alors une expression affine en (u, v) de la disparité vectorielle sous la forme :

$$\begin{cases} du &= C_u + a u + b v \\ dv &= C_v + c u + d v \end{cases}$$

Les valeurs des coefficients sont données en annexe (voir A.2).

1.6.2.2 Disparité vectorielle dans le cas de la projection orthographique

De même que précédemment et en utilisant les équations de la projection orthographique :

$$\begin{cases} u &= u_0 + \alpha_u X + \gamma Y \\ v &= v_0 + \alpha_v Y \end{cases}$$

on obtient encore une expression affine en (u, v) de la disparité vectorielle mais avec des coefficients plus simples et que l'on va pouvoir exploiter :

$$\begin{cases} du &= C_u + a u + b v \\ dv &= C_v + c u + d v \end{cases}$$

avec :

$$\begin{cases} C_u &= u'_0 + \alpha'_u (r_{13} Z_N + t_0) + \gamma' (r_{23} Z_N + t_1) - \frac{u_0}{\alpha_u} (\alpha'_u (r_{11} + N_X r_{13}) + \gamma' (r_{21} + N_X r_{23})) \\ &\quad - \frac{v_0}{\alpha_v} (\alpha'_u (r_{12} + N_Y r_{13}) + \gamma' (r_{22} + N_Y r_{23}) - \frac{\gamma}{\alpha_u} (\alpha'_u (r_{11} + N_X r_{13}) + \gamma' (r_{21} + N_X r_{23}))) \\ C_v &= v'_0 + \alpha'_v (r_{23} Z_N + t_1) - \frac{u_0 \alpha'_v}{\alpha_u} (r_{21} + N_X r_{23}) - \frac{v_0 \alpha'_v}{\alpha_v} ((r_{22} + N_Y r_{23}) + \frac{\gamma \alpha'_v}{\alpha_u \alpha_v} (r_{21} + N_X r_{23})) \\ a &= \frac{1}{\alpha_u} [\alpha'_u (r_{11} + N_X r_{13}) + \gamma' (r_{21} + N_X r_{23})] - 1 \\ b &= \frac{1}{\alpha_v} [\alpha'_v (r_{12} + N_Y r_{13}) + \gamma' (r_{22} + N_Y r_{23}) - \frac{\gamma}{\alpha_u} (\alpha'_u (r_{11} + N_X r_{13}) + \gamma' (r_{21} + N_X r_{23}))] \\ c &= \frac{\alpha_u}{\alpha_v} (r_{21} + N_X r_{23}) \\ d &= \frac{\alpha_u}{\alpha_v} ((r_{22} + N_Y r_{23}) - \frac{\gamma}{\alpha_u} (r_{21} + N_X r_{23})) - 1 \end{cases}$$

Nous avons déjà étudié dans (Lingrand et Viéville, 1996) le cas où le mouvement est suffisamment faible pour être approximé au premier ordre : $\mathbf{R} = \mathbf{I} + \tilde{\mathbf{r}}$. En supposant de plus que le facteur γ est négligeable, et que les paramètres intrinsèques sont fixes, on obtient alors des relations bien plus simples entre les paramètres permettant de retrouver la normale $[N_X \ N_Y \ -1]^T$ (voir 4.4 pour plus de détails).

1.6.3 Retour au cas de la projection perspective

Nous avons vu que dans ce cas, l'expression n'est pas affine :

$$\begin{cases} du &= [C_u + a u + b v + u(e u + g v)]/[C_d - e u - g v] \\ dv &= [C_v + c u + d v + v(e u + g v)]/[C_d - e u - g v] \end{cases}$$

sauf si on annule les coefficients suivants :

$$\begin{cases} e &= t_2 (r_0^2 + r_1^2 + r_2^2 + 4) N_X \alpha_v - 2 Z_N (r_2 r_0 - 2 r_1) \alpha_v \\ g &= -t_2 (r_0^2 + r_1^2 + r_2^2 + 4) \gamma N_X + t_2 (r_0^2 + r_1^2 + r_2^2 + 4) \alpha_u N_Y - 2 \alpha_u Z_N (r_2 r_1 + 2 r_0) \\ &\quad + 2 Z_N (r_2 r_0 - 2 r_1) \gamma \end{cases}$$

Dans le cas d'une configuration générale de points 3D, pour que ces coefficients soient toujours nuls, il faut qu'ils soient indépendants du point (donc de la normale). Ainsi, $t_2 = 0$ et $r_0 = r_1 = 0$ ce qui correspondant à un mouvement rétinien (pour lequel le plan rétinien est invariant).

Dans des configurations particulières de points 3D telles que points coplanaires (N_X , N_Y et Z_N constants), ou plus généralement, points tels que $\frac{N_X}{Z_N}$ et $\frac{N_Y}{Z_N}$ soient constants, on peut établir des conditions sur t_2 et r_2 par exemple (à moduler en fonction des annulations de dénominateurs) :

$$\begin{cases} t_2 &= 2 \frac{Z_N}{N_X} \frac{r_2 r_0 - 2 r_1}{r_0^2 + r_1^2 + r_2^2 + 4} \\ r_2 &= 2 \frac{N_Y r_0 - N_X r_1}{N_X r_0 + N_Y r_1} \end{cases} \quad (1.10)$$

Une application à la robotique du cas de mouvement rétinien est présentée dans (Lingrand et Viéville, 1996).

On a ainsi montré que dans le cas du mouvement rétinien ou de mouvement défini par l'équation (1.10) dans des configurations particulières de points, au niveau de la disparité vectorielle, les modèles para-perspectifs ou orthographiques ont des équations similaires à celle de la projection perspective. Les différences entre les deux approximations se situent au niveau des expressions des coefficients.

1.6.4 La matrice fondamentale

Après avoir étudié la disparité vectorielle, nous allons maintenant nous intéresser à la matrice fondamentale qui établit une relation entre les points des images sans dépendre de la profondeur comme c'était le cas dans le chapitre précédent.

Nous allons tout d'abord rappeler l'expression de la matrice fondamentale (Faugeras, 1993) dans le cas de la projection perspective avant de généraliser son expression et sa validité dans les cas para-perspectif et orthographique. Nous disposerons alors pour la suite d'une expression de cette quantité afin de pouvoir la déterminer dans tous les cas de projections. Nous étudierons alors les formes particulières de la matrice fondamentale dans le chapitre 2.

1.6.4.1 La matrice fondamentale dans le cas de la projection perspective

Dans ce paragraphe, on notera par $\mathbf{M}_1 = [X_1 \ Y_1 \ Z_1]^T$ un point de l'espace qui se projette en $\mathbf{m}_1 = [u_1 \ v_1 \ 1]^T$ (on prend la partie (3x3) non nulle de la matrice de projection) selon :

$$\kappa_1 \mathbf{m}_1 = \mathbf{A}_1 \mathbf{M}_1 \quad (1.11)$$

Après une rotation \mathbf{R} et une translation \mathbf{t} , il atteint une position $\mathbf{M}_2 = [X_2 \ Y_2 \ Z_2]^T$ et se projette en $\mathbf{m}_2 = [u_2 \ v_2 \ 1]^T$:

$$\mathbf{M}_2 = \mathbf{R} \mathbf{M}_1 + \mathbf{t} \quad (1.12)$$

et

$$\kappa_2 \mathbf{m}_2 = \mathbf{A}_2 \mathbf{M}_2 \quad (1.13)$$

En combinant (1.12) et (1.13), on obtient :

$$\kappa_2 \mathbf{m}_2 = \mathbf{A}_2 \mathbf{R} \mathbf{M}_1 + \mathbf{A}_2 \mathbf{t}$$

\mathbf{M}_1 est remplacé par sa valeur en (1.11) ce qui fournit :

$$\kappa_2 \mathbf{m}_2 = \kappa_1 \underbrace{\mathbf{A}_2 \mathbf{R} \mathbf{A}_1^{-1}}_{\mathbf{H}_\infty} \mathbf{m}_1 + \underbrace{\mathbf{A}_2 \mathbf{t}}_{\mathbf{e}} \quad (1.14)$$

On remarque l'apparition dans cette équation de l'homographie du plan à l'infini (voir paragraphe 1.6.5) \mathbf{H}_∞ ainsi que de l'épipôle \mathbf{e} défini précédemment.

Il suffit alors de prendre le produit vectoriel de $\mathbf{A}_2 \mathbf{t}$ par l'équation 1.14 et d'en prendre ensuite la produit scalaire par \mathbf{m}_2 pour obtenir :

$$0 = \mathbf{m}_2^T (\mathbf{A}_2 \tilde{\mathbf{t}}) \mathbf{A}_2 \mathbf{R} \mathbf{A}_1^{-1} \mathbf{m}_1$$

La matrice fondamentale, dans le cas de la projection perspective, s'exprime alors par :

$$\mathbf{F} = (\mathbf{A}_2 \tilde{\mathbf{t}}) \mathbf{A}_2 \mathbf{R} \mathbf{A}_1^{-1}$$

ce qui est une expression qui ne dépend pas des coordonnées des points de l'image, contrairement aux coefficients de la disparité vectorielle.

On peut remarquer dans ce cas que la matrice fondamentale ne permettra jamais de retrouver la norme de la translation; celle-ci ne dépend donc que de deux paramètres.

1.6.4.2 La matrice fondamentale dans le cas des approximations de la projection perspective

La matrice fondamentale peut en fait être définie quelque soit la forme des matrices de projections $\mathbf{A}_1 \in \mathbb{R}^3 \times \mathbb{R}^4$ et $\mathbf{A}_2 \in \mathbb{R}^3 \times \mathbb{R}^4$ des caméras.

En effet, considérant deux images consécutives (les coordonnées des points sont à nouveau des coordonnées projectives) :

$$\exists(\lambda_1, \lambda_2, \mathbf{M}_1, \mathbf{M}_2, \mathbf{A}_1, \mathbf{A}_2) / \begin{cases} \lambda_1 \mathbf{m}_1 = \mathbf{A}_1 \mathbf{M}_1 \\ \lambda_2 \mathbf{m}_2 = \mathbf{A}_2 \mathbf{M}_2 \end{cases} \quad (1.15)$$

La position \mathbf{M}_2 du point \mathbf{M} est obtenue après une rotation \mathbf{R} et une translation \mathbf{t} depuis la position \mathbf{M}_1 :

$$\mathbf{M}_2 = [\mathbf{R}|\mathbf{t}] \mathbf{M}_1$$

Les équations précédentes s'écrivent :

$$\exists(\lambda_1, \lambda_2, X_1, Y_1, Z_1, \mathbf{A}_1, \mathbf{A}_2) / \begin{cases} \lambda_1 \mathbf{m}_1 = \mathbf{A}_1 \mathbf{M}_1 \\ \lambda_2 \mathbf{m}_2 = \mathbf{A}_2 [\mathbf{R}|\mathbf{t}] \mathbf{M}_1 \end{cases}$$

Ce qui peut également se mettre sous la forme :

$$\exists(\mathbf{A}_1, \mathbf{A}_2) / \left| \begin{array}{ccc} \mathbf{m}_1 & \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \mathbf{A}_1 \\ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \mathbf{m}_2 & \mathbf{A}_2 [\mathbf{R}|\mathbf{t}] \end{array} \right| = 0 \quad (1.16)$$

On peut remarquer qu'il s'agit d'une forme bilinéaire en \mathbf{m}_1 et \mathbf{m}_2 , ce qui montre que ce résultat peut se mettre sous une forme plus habituelle, utilisant une généralisation de la matrice fondamentale :

$$\boxed{\mathbf{m}_2^T \mathbf{F}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{R}, \mathbf{t}) \mathbf{m}_1 = 0} \quad (1.17)$$

Notons par \mathbf{a}_i les lignes des matrices de projection :

$$\begin{aligned} \mathbf{A}_1 &= [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3]^T \\ \mathbf{A}_2 [\mathbf{R}|\mathbf{t}] &= [\mathbf{a}_4 \ \mathbf{a}_5 \ \mathbf{a}_6]^T \end{aligned}$$

où, pour chaque i : $\mathbf{a}_i = [a_{i1} \ a_{i2} \ a_{i3} \ a_{i4}]$.

Réécrivons l'équation (1.16) en utilisant cette notation :

$$\left| \begin{array}{ccc} u_1 & 0 & \mathbf{a}_1 \\ v_1 & 0 & \mathbf{a}_2 \\ 1 & 0 & \mathbf{a}_3 \\ 0 & u_2 & \mathbf{a}_4 \\ 0 & v_2 & \mathbf{a}_5 \\ 0 & 1 & \mathbf{a}_6 \end{array} \right| = 0 \quad (1.18)$$

et notons $d_{\alpha\beta\gamma\zeta}$ le déterminant suivant :

$$d_{\alpha\beta\gamma\zeta} = \left| \begin{array}{c} \mathbf{a}_\alpha \\ \mathbf{a}_\beta \\ \mathbf{a}_\gamma \\ \mathbf{a}_\delta \end{array} \right| = \left| \begin{array}{cccc} a_{\alpha 1} & a_{\alpha 2} & a_{\alpha 3} & a_{\alpha 4} \\ a_{\beta 1} & a_{\beta 2} & a_{\beta 3} & a_{\beta 4} \\ a_{\gamma 1} & a_{\gamma 2} & a_{\gamma 3} & a_{\gamma 4} \\ a_{\delta 1} & a_{\delta 2} & a_{\delta 3} & a_{\delta 4} \end{array} \right|$$

En développant le déterminant de l'équation (1.18) par la première colonne puis par la seconde colonne, on obtient :

$$\mathbf{m}_2^T \underbrace{\begin{pmatrix} d_{2356} & -d_{1356} & d_{1256} \\ -d_{p2346} & d_{1346} & -d_{1246} \\ d_{2345} & -d_{1345} & d_{1245} \end{pmatrix}}_{\mathbf{F}} \mathbf{m}_1 = 0$$

Ce qui nous fournit une expression de \mathbf{F} et permet de vérifier, par un simple calcul, que son déterminant est nul. Son expression en fonction des paramètres intrinsèques et extrinsèques ne fournit pas plus de clarté (voir annexe A.3) sauf dans le cas de la projection perspective, comme présenté au paragraphe 1.6.4.1.

Cette matrice étant par ailleurs définie à un facteur d'échelle près, comme on peut s'en convaincre par l'équation (1.17), elle possède ainsi au plus 7 degrés de liberté.

Si on fait le décompte des degrés de liberté :

- \mathbf{R} dépend de 3 paramètres
- \mathbf{t} dépend de 3 paramètres
- \mathbf{A}_1 et \mathbf{A}_2 dépendent généralement de 3 à 7 paramètres

ce qui fait généralement plus que 7 paramètres: la matrice fondamentale ne permet ainsi pas de retrouver les paramètres dans le cas général. Les cas particuliers vont être abordés dans le chapitre 2 mais on peut déjà voir que si une partie de ces paramètres est connue, il se peut que l'on puisse déterminer les autres.

D'autre part, dans le cas des approximations para-perspective et orthographique, et plus généralement, dans le cas des projections affines, on remarque que la dernière ligne de la matrice de projection est $[0 \ 0 \ 0 \ 1]$ ce qui reste vrai après mouvement (\mathbf{R}, \mathbf{t}) donc \mathbf{a}_3 et \mathbf{a}_6 sont identiques et tout déterminant $d_{\alpha\beta\gamma\zeta}$ pour lequel $\{3,6\} \subset \{\alpha,\beta,\gamma,\zeta\}$ est nul. Ainsi, la matrice fondamentale aura pour forme :

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & d_{1256} \\ 0 & 0 & -d_{1246} \\ d_{2345} & -d_{1345} & d_{1245} \end{pmatrix}$$

ce qui n'apporte que peu d'information même si elle est alors plus facile à estimer.

En particulier, dans le cas orthographique, et en supposant γ nul et que la rotation n'est pas un retournement, l'expression de la matrice fondamentale est :

$$\begin{pmatrix} 0 & 0 & -\frac{1}{2} \alpha_u \alpha_v \alpha'_v \cos^2(\frac{\theta}{2}) (-r_2 r_1 + 2r_0) \\ 0 & 0 & -\frac{1}{2} \alpha_u \alpha_v \alpha'_u \cos^2(\frac{\theta}{2}) (r_2 r_0 + 2r_1) \\ \frac{1}{2} \alpha_v \alpha'_u \alpha'_v \cos^2(\frac{\theta}{2}) (r_2 r_1 + 2r_0) & \frac{1}{2} \alpha_u \alpha'_u \alpha'_v \cos^2(\frac{\theta}{2}) (2r_1 - r_2 r_0) & f_{33} \end{pmatrix}$$

où :

$$\begin{aligned} f_{33} = & \frac{1}{2} \cos^2(\frac{\theta}{2}) [\\ & -u_0 \alpha_v \alpha'_u \alpha'_v (2r_0 + r_1 r_2) - v_0 \alpha_u \alpha'_u \alpha'_v (2r_1 - r_0 r_2) \\ & u'_0 \alpha_u \alpha_v \alpha'_v (2r_0 - r_1 r_2) + v'_0 \alpha_u \alpha_v \alpha'_u (2r_1 + r_0 r_2) \\ & t_0 \alpha_u \alpha_v \alpha'_u \alpha'_v (2r_0 - r_1 r_2) + t_1 \alpha_u \alpha_v \alpha'_u \alpha'_v (2r_1 + r_0 r_2)] \end{aligned}$$

L'expression de la matrice fondamentale dans ce cas est encore plus simple, si, tout comme dans (Shapiro et al., 1995), on utilise la décomposition en angles d'Euler de Koenderink et van Doorn (Koenderink et van Doorn, 1991) :

$$\mathbf{R} = \begin{pmatrix} (1 - \cos(\rho)) \cos(\varphi) \cos(\varphi - \theta) + \cos(\rho) \cos(\theta) & (1 - \cos(\rho)) \cos(\varphi) \sin(\varphi - \theta) - \cos(\rho) \sin(\theta) & \sin(\varphi) \sin(\rho) \\ (1 - \cos(\rho)) \sin(\varphi) \cos(\varphi - \theta) + \cos(\rho) \sin(\theta) & (1 - \cos(\rho)) \sin(\varphi) \sin(\varphi - \theta) + \cos(\rho) \cos(\theta) & -\cos(\varphi) \sin(\rho) \\ -\sin(\rho) \sin(\varphi - \theta) & \sin(\rho) \cos(\varphi - \theta) & \cos(\rho) \end{pmatrix}$$

On obtient alors :

$$\mathbf{F} = \sin(\rho) \begin{pmatrix} 0 & 0 & -\alpha_u \alpha_v \alpha'_v \cos(\varphi) \\ 0 & 0 & -\alpha_u \alpha_v \alpha'_u \sin(\varphi) \\ \alpha_v \alpha'_u \alpha'_v \cos(\theta - \varphi) & -\alpha_u \alpha'_u \alpha'_v \sin(\theta - \varphi) & f_{33} \end{pmatrix}$$

avec :

$$\begin{aligned} f_{33} &= -u_0 \alpha_v \alpha'_u \alpha'_v \cos(\theta - \varphi) + v_0 \alpha_u \alpha'_u \alpha'_v \sin(\theta - \varphi) \\ &\quad + u'_0 \alpha_u \alpha_v \alpha'_v \cos(\varphi) + v'_0 \alpha_u \alpha_v \alpha'_u \sin(\varphi) \\ &\quad + t_0 \alpha_u \alpha_v \alpha'_u \alpha'_v \cos(\varphi) + t_1 \alpha_u \alpha_v \alpha'_u \alpha'_v \sin(\varphi) \end{aligned}$$

Au niveau de la rotation, l'expression de la matrice fondamentale ne dépend que de deux angles : φ et $\theta - \varphi$. Cependant, pour des raisons de généralité, nous conserverons l'expression précédente.

On a démontré dans (Viéville et Lingrand, 1995), dans le cas de la projection perspective, l'équivalence entre une matrice fondamentale de la forme :

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & f_{13} \\ 0 & 0 & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix}$$

et un mouvement rétinien (translation parallèle au plan rétinien et rotation d'axe parallèle à l'axe optique). Ceci nous montre donc que, au niveau de la matrice fondamentale, les cas para-perspectif et orthographique sont équivalents au cas de la projection perspective avec mouvement rétinien.

1.6.5 Les homographies

Dans certains cas de mouvement ou de configuration des points de la scène, la matrice fondamentale n'existe pas. Ces cas sont bien connus pour la projection perspective et on sait qu'il existe alors une relation homographique entre les points des images. Dans ce cas, on s'intéresse donc à la matrice d'homographie.

Après avoir rappelé les conditions d'existence d'une telle relation dans le cas de la projection perspective, nous allons déterminer les conditions d'existence d'une même relation pour les approximations de la projection perspective. Ces résultats sont importants pour la suite car nous permettent de savoir, pour un cas donné, si la matrice fondamentale existe où s'il faut alors plutôt considérer la matrice d'homographie.

1.6.5.1 Homographie dans le cas de la projection perspective

Dans le cas où la projection est perspective et la translation \mathbf{t} nulle, l'expression de la matrice fondamentale vue précédemment s'annule et on ne peut plus parler alors de matrice fondamentale. Par contre, si on prend les parties (3x3) non nulles de \mathbf{A}_1 et \mathbf{A}_2 , alors, les équations :

$$\begin{cases} Z_1 \mathbf{m}_1 &= \mathbf{A}_1 \mathbf{M}_1 \\ Z_2 \mathbf{m}_2 &= \mathbf{A}_2 \mathbf{M}_2 \\ \mathbf{M}_2 &= \mathbf{R} \mathbf{M}_1 \end{cases}$$

fournissent la relation suivante :

$$\mathbf{m}_2 = \frac{Z_1}{Z_2} \mathbf{A}_2 \mathbf{R} \mathbf{A}_1^{-1} \mathbf{m}_1$$

ce qui constitue une homographie entre les deux images: $\kappa \mathbf{m}_1 = \mathbf{H}_\infty \mathbf{m}_2$ où $\mathbf{H}_\infty = \mathbf{A}_2 \mathbf{R} \mathbf{A}_1^{-1}$.

D'autre part, si on considère une scène plane où le plan a pour normale \mathbf{n} et distance par rapport à l'origine d , on obtient :

$$[\mathbf{n}^T \mathbf{M}_1 = d] \cap [\mathbf{M}_2 = \mathbf{R} \mathbf{M}_1 + \mathbf{t}] \implies \mathbf{M}_2 = \left(\mathbf{R} + \frac{\mathbf{t} \mathbf{n}^T}{d} \right) \mathbf{M}_1$$

Le cas de la rotation pure correspond à ce cas en faisant tendre $\frac{1}{d}$ vers 0. Inversément, ce cas correspond à celui de la rotation pure en prenant $\mathbf{R}' = \mathbf{R} + \frac{\mathbf{t} \mathbf{n}^T}{d}$. Ainsi, il y a équivalence entre mouvement sans translation et mouvement d'un plan. Si on se place dans le cas d'une projection perspective, il existe une relation homographique entre les deux images :

$$\lambda \mathbf{m}_2 = \mathbf{H} \mathbf{m}_1$$

avec :

$$\mathbf{H} = \mathbf{A}_2 \left(\mathbf{R} + \frac{\mathbf{t} \mathbf{n}^T}{d} \right) \mathbf{A}_1^{-1}$$

Lorsque d tend vers l'infini, ce qui correspond à des points situés sur un plan à l'infini, le terme $\frac{\mathbf{t} \mathbf{n}^T}{d}$ tend vers 0 et l'expression de l'homographie \mathbf{H} tend vers l'expression correspondant au cas de la rotation pure :

$$\mathbf{H} \xrightarrow{d \rightarrow \infty} \mathbf{A}_2 \mathbf{R} \mathbf{A}_1^{-1} = \mathbf{H}_\infty$$

\mathbf{H}_∞ représente ainsi l'homographie du plan à l'infini.

1.6.5.2 Homographie dans le cas de la projection para-perspective

Afin de simplifier les notations, nous noterons les paramètres intrinsèques de la caméra $\alpha_u, \alpha_v, \beta_u, \beta_v, \gamma, u_0$ et v_0 sans utiliser les indices relatifs au type de projection. De plus, pour une matrice \mathbf{A} , on notera par $(\mathbf{A})_3$ son troisième vecteur colonne et par $(\mathbf{A})_{-3}$ la matrice composée des colonnes des \mathbf{A} sauf la troisième colonne. Nous noterons également $\underline{\mathbf{M}} = [X \ Y \ 1]^T$ le vecteur réduit d'un point de l'espace.

Nous avons vu au paragraphe 1.3.1 que la projection para-perspective s'exprime de la façon suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & \gamma & \beta_u & u_0 \\ 0 & \alpha_v & \beta_v & v_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

ce qui peut s'écrire également :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{(\mathbf{A})_{-3}} \underbrace{\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}}_{\underline{\mathbf{M}}} + Z \underbrace{\begin{pmatrix} \beta_u \\ \beta_v \\ 1 \end{pmatrix}}_{(\mathbf{A})_3}$$

où $(\mathbf{A})_{-3}$ est une matrice carrée inversible :

$$\det((\mathbf{A})_{-3}) = \alpha_u \alpha_v \neq 0$$

Pour deux vues, les équations de projection s'écrivent alors :

$$\begin{cases} \mathbf{m}_1 &= (\mathbf{A}_1)_{-3} \underline{\mathbf{M}}_1 + Z_1 (\mathbf{A}_1)_3 \Rightarrow \underline{\mathbf{M}}_1 = ((\mathbf{A}_1)_{-3})^{-1} \mathbf{m}_1 - Z_1 ((\mathbf{A}_1)_{-3})^{-1} (\mathbf{A}_1)_3 \\ \mathbf{m}_2 &= \mathbf{A}_2 \mathbf{M}_2 \end{cases} \quad (1.19)$$

L'équation du mouvement se met sous la forme : $\mathbf{M}_2 = [\mathbf{R}|\mathbf{t}] \mathbf{M}_1$

Ainsi, en combinant ceci avec les équations (1.19), on obtient :

$$\begin{aligned} \mathbf{m}_2 &= \mathbf{A}_2 [\mathbf{R}|\mathbf{t}] \mathbf{M}_1 \\ &= (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} \underline{\mathbf{M}}_1 + Z_1 (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3 \\ &= \underbrace{(\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1}}_{\mathbf{H}_{\infty para}} \mathbf{m}_1 - Z_1 (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1} (\mathbf{A}_1)_3 + Z_1 (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3 \end{aligned}$$

Soit :

$$\mathbf{m}_2 = \mathbf{H}_{\infty para} \mathbf{m}_1 + Z_1 \mathbf{K} \quad (1.20)$$

où $\mathbf{K} = (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3 - (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1} (\mathbf{A}_1)_3$

L'équation 1.20 représente une homographie si et seulement si le vecteur $Z_1 \mathbf{K}$ est lui-même issu de \mathbf{m}_1 par une homographie ce qui est équivalent aux deux cas suivants :

- soit Z_1 est une fonction affine de X_1 et Y_1 .
- soit les coordonnées de \mathbf{K} sont nulles,

Nous allons maintenant examiner ces deux cas, le premier correspondant à une contrainte de nature géométrique, le second à une contrainte sur les paramètres de mouvement.

Condition de géométrie : La première condition est une condition de géométrie des points de la scène et impose que ceux-ci soient sur un plan d'équation $Z = aX + bY + c$, c'est-à-dire un plan ne passant pas par l'axe optique. Le centroïde \mathbf{M}_0 appartenant à ce plan, l'équation sera de la forme :

$$Z = a(X - X_0) + b(Y - Y_0) + Z_0 = \mathbf{n} \underline{\mathbf{M}}$$

avec $\mathbf{n} = [a \ b \ -aX_0 - bY_0 + Z_0]^T$. Ainsi, l'équation 1.20 s'écrit :

$$\mathbf{m}_2 = \mathbf{H}_{para} \mathbf{m}_1 + \mathbf{K} \mathbf{n} \underline{\mathbf{M}}_1$$

et l'équation de projection dans la première image issue de 1.19 :

$$\begin{aligned}\mathbf{m}_1 &= (\mathbf{A}_1)_{-3} \underline{\mathbf{M}}_1 + (\mathbf{A}_1)_3 \mathbf{n} \underline{\mathbf{M}}_1 \\ &= \underbrace{((\mathbf{A}_1)_{-3} + (\mathbf{A}_1)_3 \mathbf{n})}_{\mathbf{B}} \underline{\mathbf{M}}_1\end{aligned}$$

Il nous reste à vérifier que la matrice \mathbf{B} est inversible :

$$\begin{aligned}\det(\mathbf{B}) &= \begin{vmatrix} \alpha_u + a \beta_u & \gamma + b \beta_u & u_0 + (-a X_0 - b Y_0 + Z_0) \beta_u \\ a \beta_v & \alpha_v + b \beta_v & v_0 + (-a X_0 - b Y_0 + Z_0) \beta_v \\ 0 & 0 & 1 \end{vmatrix} \\ &= (\alpha_u + a \beta_u) (\alpha_v + b \beta_v) - (\gamma + b \beta_u) a \beta_v \\ &= \alpha_u \alpha_v + a \beta_u \alpha_v + b \beta_v \alpha_u - \gamma a \beta_v\end{aligned}$$

Dans le cas où le paramètre γ est nul et en utilisant les relations 1.3, cela revient à imposer :

$$Z_0 \neq a X_0 + b Y_0$$

qui signifie que le plan ne doit pas contenir le centre optique. On veut donc des points éloignés et sur un même plan ne contenant ni l'axe optique ni le centre optique, on veillera donc à ce que ce plan soit un plan approximativement fronto-parallèle.

Dans ce cas, l'homographie s'exprime par :

$$\mathbf{H}_{para} = (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1} + ((\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3 - (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1} (\mathbf{A}_1)_3 \mathbf{n} ((\mathbf{A}_1)_{-3} + (\mathbf{A}_1)_3 \mathbf{n})^{-1}$$

Condition sur le mouvement : La seconde condition revient à imposer une contrainte sur les paramètres du mouvement. L'expression de l'homographie est simplifiée :

$$\mathbf{H}_{\infty para} = (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1}$$

par la nullité du vecteur :

$$\mathbf{K} = (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3 - (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1} (\mathbf{A}_1)_3 = 0$$

Soit, en développant :

$$\begin{pmatrix} \alpha'_u & \gamma' & \beta'_u & u'_0 \\ 0 & \alpha'_v & \beta'_v & v'_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{13} \\ r_{23} \\ r_{33} \\ 0 \end{pmatrix} - \begin{pmatrix} \alpha'_u & \gamma' & \beta'_u & u'_0 \\ 0 & \alpha'_v & \beta'_v & v'_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_0 \\ r_{21} & r_{22} & t_1 \\ r_{31} & r_{32} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha_u} & \frac{-\gamma}{\alpha_u \alpha_v} & \frac{\gamma v_0 - \alpha_v u_0}{\alpha_u \alpha_v} \\ 0 & \frac{1}{\alpha_v} & \frac{-v_0}{\alpha_u} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_u \\ \beta_v \\ 0 \end{pmatrix}$$

Dans le produit $\mathbf{A}_2 [\mathbf{R}|\mathbf{t}]$, c'est la dernière colonne qui comprend les paramètres de translation et donc, le vecteur colonne $(\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3$ ne dépend pas de la translation. Ainsi, dans le produit $(\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1}$, c'est toujours dans la dernière colonne qu'on retrouve les paramètres de translation. Lors du produit de ceci par $(\mathbf{A}_1)_3$ dont la troisième coordonnée est nulle, les termes

concernant la translation disparaissent. Ainsi, les contraintes sur le mouvement ne concernent que la rotation. De plus, la dernière coordonnée de \mathbf{K} est nulle. Nous aurons ainsi deux contraintes que l'on peut montrer indépendantes sur les paramètres de rotation: l'axe de rotation est donc fixé.

Afin d'exhiber plus clairement ces contraintes, nous allons tout d'abord considérer que le paramètre γ de la caméra est nul pour les deux images, ce qui constitue une hypothèse tout à fait plausible vue que nous sommes déjà en présence d'une approximation de la projection perspective. Prenant aussi en compte le fait que $\beta_u = -\alpha_u \frac{X_0}{Z_0}$ et $\beta_v = -\alpha_v \frac{Y_0}{Z_0}$, on obtient alors les contraintes suivantes, dans le cas $\theta \neq \pi$:

$$\begin{cases} 0 &= -X_0 Z_0 r_0^2 + X_0 Z_0 r_2^2 - 2 X_0^2 r_1 + X_0^2 r_2 r_0 + 2 Y_0 Z_0 r_2 - Y_0 Z_0 r_1 r_0 + 2 Y_0 X_0 r_0 \\ &+ Y_0 X_0 r_2 r_1 - 2 Z_0^2 r_1 - Z_0^2 r_2 r_0 \\ 0 &= -2 X_0 Z_0 r_2 - X_0 Z_0 r_1 r_0 - 2 X_0 Y_0 r_1 + X_0 Y_0 r_2 r_0 - Y_0 Z_0 r_1^2 + Y_0 Z_0 r_2^2 + 2 Y_0^2 r_0 \\ &+ Y_0^2 r_2 r_1 + 2 Z_0^2 r_0 - Z_0^2 r_2 r_1 \end{cases}$$

Ce qui se met sous la forme :

$$\begin{cases} 0 &= (-r_2 X_0 + r_0 Z_0) \left((r_0 Y_0 + 2 Z_0)^2 + (X_0 r_0 + Z_0 r_2)^2 + (Y_0 r_2 - 2 X_0)^2 \right) \\ r_1 &= -\frac{X_0 Z_0 r_0^2 - X_0 Z_0 r_2^2 - 2 Y_0 X_0 r_0 - X_0^2 r_2 r_0 - 2 Y_0 Z_0 r_2 + Z_0^2 r_2 r_0}{2 X_0^2 - Y_0 X_0 r_2 + 2 Z_0^2 + Y_0 Z_0 r_0} \end{cases}$$

La première équation offre deux possibilités :

- soit $r_0 = r_2 \frac{X_0}{Z_0}$ et, par substitution dans la seconde équation, $r_1 = r_2 \frac{Y_0}{Z_0}$
- soit $r_2 = 2 \frac{X_0}{Y_0}$ et $r_0 = -2 \frac{Z_0}{Y_0}$ mais ces valeurs annulent le dénominateur de r_1 . Il n'y a pas de solution dans ce cas.

On obtient donc une solution sous la forme : $\mathbf{r} = \theta \mathbf{M}_0$ L'angle de la rotation est libre mais son axe correspond à la direction du regard.

Dans le cas des retournements ($\theta = \pi$), on exprime la rotation sous la forme $\mathbf{R} = \mathbf{I} + 2 \tilde{\mathbf{u}}$, ce qui fournit :

$$\begin{cases} 0 &= \alpha'_u (-u_0 Z_0 + u_2 X_0) (u_2 Z_0 + u_0 X_0 + u_1 Y_0) \\ 0 &= \alpha'_v (-u_1 Z_0 + u_2 Y_0) (u_2 Z_0 + u_0 X_0 + u_1 Y_0) \end{cases}$$

On obtient alors deux solutions: soit $\mathbf{u} \parallel \mathbf{M}_0$, ce qui correspond au cas général de rotation, soit $\mathbf{u} \perp \mathbf{M}_0$, ce qui constitue une solution supplémentaire mais peu utilisable pour un angle de vue inférieur à 180 degrés.

Un raisonnement similaire peut être fait dans le cas où on prend en compte la non nullité du paramètre γ et est présenté au paragraphe A.4. L'influence de γ se retrouve sur l'axe de rotation dont la direction est légèrement modifiée.

En résumé, dans le cas de la projection para-perspective, on est en présence d'une relation homographique si et seulement si on respecte une des deux conditions suivantes :

condition sur le mouvement : l'axe de rotation correspond à la direction du regard (direction par laquelle la caméra voit le centroïde de la scène), ou, dans le cas des retournements, l'axe est orthogonal à la direction du regard.

condition géométrique : les points de la scène sont sur un plan ne contenant pas l'axe optique

1.6.5.3 Homographie dans le cas de la projection orthographique

On emploie les mêmes notations que lors du paragraphe précédent.

Ainsi, nous avons vu précédemment que la projection orthographique s'exprime de la façon suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & \gamma & 0 & u_0 \\ 0 & \alpha_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

ce qui peut s'écrire également :

$$\mathbf{m} = (\mathbf{A})_{-3} \underline{\mathbf{M}}$$

où $(\mathbf{A})_{-3}$ est une matrice carrée inversible : $\det((\mathbf{A})_{-3}) = \alpha_u \alpha_v \neq 0$

Pour deux vues, les équations de projection s'écrivent alors :

$$\begin{cases} \mathbf{m}_1 = (\mathbf{A}_1)_{-3} \underline{\mathbf{M}}_1 \Rightarrow \underline{\mathbf{M}}_1 = ((\mathbf{A}_1)_{-3})^{-1} \mathbf{m}_1 \\ \mathbf{m}_2 = \mathbf{A}_2 \mathbf{M}_2 \end{cases} \quad (1.21)$$

toujours par :

$$\mathbf{M}_2 = [\mathbf{R}|\mathbf{t}] \mathbf{M}_1$$

ce qui donne :

$$\begin{aligned} \mathbf{m}_2 &= \mathbf{A}_2 [\mathbf{R}|\mathbf{t}] \mathbf{M}_1 \\ &= (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} \underline{\mathbf{M}}_1 + Z_1 (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3 \end{aligned}$$

En combinant ces équations, on obtient :

$$\mathbf{m}_2 = \underbrace{(\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_{-3} ((\mathbf{A}_1)_{-3})^{-1}}_{\mathbf{H}_{\infty ortho}} \mathbf{m}_1 + Z_1 \underbrace{(\mathbf{A}_2 [\mathbf{R}|\mathbf{t}])_3}_{\mathbf{K}'}$$

On est donc en présence d'une relation homographique si et seulement si \mathbf{K}' est nul ou Z_1 est un polynôme du premier degré en X_1 et Y_1 .

Condition sur le mouvement : Tout comme dans le cas de la projection para-perspective, on obtient deux conditions indépendantes pour l'existence de l'homographie. On retrouve la condition sur la géométrie de la scène : les points sont situés sur un même plan ne contenant pas l'axe optique. La nullité du vecteur \mathbf{K}' impose des conditions sur les paramètres du mouvement :

$$\mathbf{K}' = \mathbf{0} \Leftrightarrow r_{13} = r_{23} = 0 \Leftrightarrow \begin{cases} u_0 = u_1 = 0 \text{ ou bien} \\ u_2 = 0 \text{ dans le cas } \theta = \pi \end{cases}$$

ce qui revient à imposer à l'axe de rotation d'être parallèle à l'axe optique qui, dans le cas de la projection orthographique correspond à la direction du regard, ou, dans le cas des retournements, à l'axe d'être orthogonal à l'axe optique. La translation n'est pas contrainte.

L'homographie s'exprime par :

$$\mathbf{H}_{\infty_{ortho}} = (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}]_{-3} ((\mathbf{A}_1)_{-3})^{-1}$$

Condition de géométrie : La seconde condition impose aux points de la scène d'appartenir à un plan ne contenant pas l'axe optique et d'équation :

$$Z = [a \ b \ Z_0] [X \ Y \ 1]^T = \mathbf{n}^T \underline{\mathbf{M}}$$

d'où :

$$\mathbf{m}_2 = ((\mathbf{A}_2 [\mathbf{R}|\mathbf{t}]_{-3} ((\mathbf{A}_1)_{-3})^{-1} + (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}]_3 \mathbf{n}^T ((\mathbf{A}_1)_{-3})^{-1}) \mathbf{m}_1$$

L'homographie s'exprime donc par :

$$\mathbf{H}_{ortho} = ((\mathbf{A}_2 [\mathbf{R}|\mathbf{t}]_{-3} ((\mathbf{A}_1)_{-3})^{-1} + (\mathbf{A}_2 [\mathbf{R}|\mathbf{t}]_3 \mathbf{n}^T ((\mathbf{A}_1)_{-3})^{-1})$$

Il n'est pas étonnant que le cas orthographique s'insère dans le cas para-perspectif car il est un cas particulier de ce dernier pour lequel la direction du regard est celle de l'axe optique.

On peut également remarquer que, étant donné les conditions de validité de la projection orthographique vue au paragraphe 1.3.4, i.e. l'objet est loin de la caméra et le mouvement laisse invariant le plan rétinien, on aura ainsi toujours une relation homographique entre les images.

1.7 Résumé du chapitre.

Nous sommes parvenus, dans ce chapitre, à modéliser la projection réalisée par une caméra selon différents modèles. La projection perspective est la plus couramment employée mais nécessite des améliorations dans certains cas que nous effectuerons en prétraitement si besoin. Des approximations de la projection perspective: la projection para-perspective et la projection orthographique, pour lesquelles nous avons clarifié les approximations effectuées et les paramètres mis en jeu, seront également utilisées dans la suite de ce travail. D'autres modèles de projection correspondent à des cas de caméras particulières vues au paragraphe 1.4.3 mais ne seront pas utilisés dans la suite. Cependant, nous prévoyons l'introduction de ces modèles dans le futur, en construisant, dans le chapitre suivant, un modèle d'étude des particularités qui pourra être utilisé avec ces caméras.

Nous avons également modélisé le mouvement rigide par deux vecteurs, un vecteur de translation classique et un vecteur de rotation en apportant une amélioration à l'expression classique de la rotation en exponentielle de vecteur dans le sens où, dans notre cas, seule une carte de paramètres est nécessaire.

Enfin, nous nous sommes intéressés aux relations existant entre les points projetés dans deux images: la relation fondamentale et la relation homographique. Après avoir rappelé les cas d'existence

de ces relations dans le cas de la projection perspective, nous avons établi des conditions similaires dans les cas des projections para-perspective et orthographique que nous résumons dans le tableau suivant :

projection	condition sur le mouvement	condition géométrique
perspective	pas de translation	plan quelconque
para-perspective	axe de rotation \parallel direction du regard ou angle π et axe \perp direction du regard	plan* $Z = a(X - X_0) + b(Y - Y_0) + Z_0$
orthographique	axe de rotation \parallel axe optique ou angle π et axe \perp axe optique	plan $Z = aX + bY + Z_0$

*: ce plan ne doit pas contenir le centre optique i.e. $Z_0 \neq aX_0 + bY_0$

Chapitre 2

Les Cas Particuliers

Après une étude bibliographique détaillée des utilisations de cas particuliers de projection, mouvement et structure de scènes pour des couples d'images puis des séquences monoculaires, ce cha-

pitre fait l'inventaire exhaustif de toutes les combinaisons de cas particuliers et propose une méthode d'analyse de cet inventaire.

« Mon premier est la première syllabe de biberon

On dort dans mon second

Le bas est le contraire de mon troisième

On finit mon tout avant l'an 2000 »

Thierry Viéville

Singular cases (english abstract)

Several authors have studied some singular cases of projection (simplified camera models), rigid displacements (more specifically: pure rotations, pure translations, simple focal length variation, planar displacements, fixed axis rotations), or specific geometric configuration (e.g., planar case), in the uncalibrated case.

These studies are motivated by the fact that, in the general case, it is not possible to self-calibrate the camera when zooming or modifying the intrinsic calibration parameters because we have more unknowns than equations, while some of these equations are subject to unrecoverable singularities. Moreover, an adequate model, with a minimal set of parameters, will increase the statistical inference.

The goal of this chapter is to establish an exhaustive list of "atomic" singular cases of projection, intrinsic parameters, extrinsic parameters, and scene geometry. Then, "molecular" cases combining these atomic cases are built using some rules to ensure that these cases exists in the reality and that they are not redundant.

Among these cases, those for which a more effective evaluation of Euclidean parameters (intrinsic and extrinsic) and affine parameters are identified will be studied in chapter 4.

2.1 Introduction.

De nombreuses séquences vidéos présentent des cas particuliers : un piéton ou une voiture se déplacent en restant en contact avec le sol (mouvement sur une surface ou mouvement planaire), une caméra de surveillance est généralement fixe, un robot n'évolue que selon certains degrés de liberté fixés.

De nombreux auteurs ont essayé d'analyser plus simplement le problème de la vision en se plaçant dans des cas de mouvements particuliers, car les équations générales ne sont pas toujours solubles dans ces cas spécifiques. De plus, on peut espérer simplifier le problème car les contraintes ajoutées permettent d'éliminer des inconnues du problème. Par ailleurs, certains mouvements conduisent à des singularités des équations. Elles peuvent aussi conduire à des simplifications nous permettant de déterminer des paramètres qu'il n'est pas possible de déterminer dans le cas général.

Après avoir passé en revue les différents cas traités dans la littérature, nous ferons l'inventaire des cas particuliers que nous allons traiter dans cette thèse.

2.2 Étude Bibliographique.

La plupart des articles publiés dans le domaine s'intéresse à un seul cas particulier concernant le modèle de projection, l'approximation des paramètres ou encore, le plus souvent, le mouvement. De plus, il s'agit la plupart du temps d'étudier seulement deux images, bien que certains auteurs traitent également le cas des séquences vidéo, soit en tenant compte des vues 2 par 2, soit 3 par 3, voir même dans leur ensemble de N vues.

Plus récemment, un certain nombre d'auteurs se sont intéressés à l'étude de plusieurs modèles ou cas particuliers différents et aux méthodes de sélection du modèle le plus en adéquation avec les données. Nous allons donc détailler les différentes approches concernant 2 vues dans ce paragraphe tandis que celles comportant plus de deux vues seront étudiées dans le paragraphe 2.6.

L'étude des cas particuliers est motivée par deux facteurs : le premier concerne la simplification des équations et la réduction des paramètres à estimer, le second, le fait que dans un cas singulier, des données aberrantes peuvent leurrer complètement la situation. Ainsi, comme l'expliquent (Torr et Murray, 1997) en partant du cas de l'estimation d'une droite à partir de points effectivement groupés et en montrant qu'un seul point peut illusoirement suffire à déterminer une droite modélisant l'ensemble des points. D'autre part, certains auteurs comme par exemple (Triggs, 1998b) ont remarqué que beaucoup de caméras et de géométries de scènes correspondent à des cas dégénérés dont il est nécessaire d'étudier et d'exploiter les propriétés afin d'en accroître la stabilité.

Des définitions de tels "cas dégénérés" ou de "données dégénérées" ont été données par différents auteurs. Torr (Torr et al., 1995) considère dégénérées des données qui ne permettent pas de déterminer de façon unique la solution du problème du mouvement. Sturm (Sturm, 1997b) quant à lui considère qu'une séquence d'images est critique lorsqu'il existe de fausses quadriques absolues, c'est-à-dire plusieurs solutions au problème de la calibration Euclidienne (dans ce cas, il montre que l'hyperplan à l'infini ne peut pas être identifié de façon unique et que la reconstruction affine ne

peut pas être réalisée).

Nous allons maintenant faire un état de l'art des cas particuliers déjà étudiés. On distinguera pour cela les cas particuliers concernant le modèle de projection, le mouvement et les propriétés géométriques de la scène.

2.2.1 Modèle de projection.

Généralement, le modèle le plus souvent utilisé dans la littérature est le modèle de projection perspective. Cependant, différentes approximations (voir chapitre 1) ont été utilisées notamment par Aloimonos (Aloimonos, 1990), Dementhon (Dementhon et Davis, 1989), Horaud et Christy (Horaud et al., 1994; Horaud et al., 97) et Soatto et Perona (Soatto et Perona, 1995).

Poelman et Kanade (Poelman et Kanade, 1994) ont utilisé le modèle para-perspectif sur des séquences d'images et trouvent des résultats meilleurs que pour le modèle orthographique lorsque le mouvement de translation le long de l'axe optique ou orthogonal à l'axe optique est important. Cependant, le modèle para-perspectif nécessite de connaître le centre optique ainsi que la focale. Si ces paramètres ne sont pas connus, alors, le modèle orthographique lui est équivalent.

Boufama et Weinshall (Boufama et al., 1994) ont effectué une comparaison des modèles de projection orthographique et perspectif avec des algorithmes utilisant des invariants sur trois séquences d'images aux différentes tailles de champ de vue. Le modèle orthographique s'avérant plus performant pour un petit champ de vue et le modèle perspectif plus performant pour un large champ de vue, il vient naturellement à l'esprit des auteurs d'utiliser un algorithme combinant ces deux approches. Ce type d'approche par stratification a été utilisé par d'autres auteurs pour le mouvement (voir exemples au paragraphe 2.2.3). Lingrand et Viéville, (Lingrand et Viéville, 1995), ont utilisé le modèle orthographique dans le cas où le mouvement de la caméra conserve son plan rétinien. Quan, (Quan, 1996), a utilisé ce même modèle mais pour des séquences d'images.

2.2.2 Paramètres internes de la caméra.

Les modèles de projection sont paramétrés par les paramètres intrinsèques que la plupart des méthodes d'auto-calibration suppose constants. Or, dans des applications de vision active il n'est pas raisonnable de les supposer ainsi (Viéville et al., 1994). Cependant, différentes remarques peuvent être formulées.

Enciso (Enciso, 1995) a montré expérimentalement que le rapport $\frac{\alpha_u}{\alpha_v}$ est constant pour des caméras ordinaires (appareils photographiques, caméscopes, ...) et que le paramètre d'orthogonalité γ peut raisonnablement être considéré comme négligeable. Willson (Willson, 1994) a montré cette propriété pour les capteurs visuels de haut niveau. Luong et Viéville, (Luong et Viéville, 1996), Tomasi et Kanade, (Tomasi et Kanade, 1992) ainsi que Zisserman et Liebowitz (Zisserman et al., 1998), ont alors éliminé le paramètre d'orthogonalité de leur travaux. Heyden et Åström (Heyden et Åström, 1997) ont utilisé ces propriétés et montrent ainsi qu'en connaissant le foyer d'expansion et le paramètre d'orthogonalité, on peut effectuer une reconstruction à des similarités près. Pollefeys et Van Koch (Pollefeys et al., 1997) ont généralisé ces résultats en prouvant que l'autocalibration

est possible dans le cas où les paramètres intrinsèques varient dès que l'on sait que $\gamma = 0$. De plus, pour Pollefeys et Van Koch, considérer que tous les paramètres intrinsèques sont constants n'est pas une hypothèse réaliste tandis que considérer qu'ils sont tous variables est bien trop général. Ainsi, toujours dans (Pollefeys et al., 1997), ils utilisent des hypothèses de connaissances de certains paramètres intrinsèques ou de relations linéaires entre eux, à partir des équations de Kruppa (Faugeras, 1993), obtiennent une relation sur les valeurs singulières des contraintes : celles-ci doivent être non nulles afin d'éviter toute dégénérescence.

Des études sur la variation de focale et le zoom (Lavest, 1992) suivi de (Bobet, 1994), (Enciso, 1995) et (Gaspard et al., 1997) ont mis en évidence que les coordonnées du point principal pouvaient être considérées comme constantes lors de faibles variations de focale mais qu'il était difficile de modéliser le déplacement du point principal lors de grande variation de focale. Lavest (Lavest, 1992) ainsi qu'Enciso (Enciso, 1995) et (Pollefeys et al., 95) ont utilisé le zoom mais aussi d'autres paramètres tels que le changement de focus pour la reconstruction, ou la vergence, tout comme Bobet, (Bobet, 1994), dans le cadre de l'utilisation d'une tête stéréoscopique. Brooks et de Agapito (Brooks et al., 1996) ont également étudié l'autocalibration d'une tête stéréo ainsi que les dégénérescences associées. Gaspard, quant à lui, a étudié le zoom dans le cas de séquences monoculaires. Heyden (Heyden et Åström, 1997), a poursuivi ces études dans le but d'une reconstruction euclidienne par variation de focale sur une séquence d'images.

2.2.3 Mouvement.

Les cas particuliers de mouvements ont été les premiers à être étudiés et sont également les plus largement utilisés. Cela provient de plusieurs raisons. La première est que beaucoup de mouvements d'objets ou de caméras sont particuliers: une voiture sur une route, des mouvements effectués par des systèmes robotiques dont les degrés de liberté sont restreints ou contraints, ... La seconde raison est plus mathématique : la matrice fondamentale correspondant à la projection perspective devient nulle dès que la rotation est pure (voir autres cas pour les autres modèles de projection dans le tableau en fin du paragraphe 1.7, chapitre 1); la nullité de certaines composantes du mouvement peut simplifier également l'expression de la matrice fondamentale en diminuant le nombre de paramètres dont celle-ci dépend.

L'étude de la rotation pure a été effectuée par de nombreux auteurs. Certains supposent que les paramètres intrinsèques sont constants, d'autres qu'ils peuvent varier, certains considèrent que la rotation est inconnue, d'autres que son angle ou bien son axe est connu. Hartley, (Hartley, 1994b), considère une séquence d'images acquises par une caméra en rotation pure d'axe passant par son centre optique. Considérant de plus que les paramètres intrinsèques de la caméra sont constants, Hartley montre avec un raisonnement algébrique qu'il est alors aisé de retrouver les paramètres intrinsèques de la caméra. Par la suite, De Agapito et Hayman, (de Agapito et al., 1998), ont étudié le cas de la rotation pure avec paramètres intrinsèques variables. Viéville (Viéville, 1994), dans le cas où l'angle de rotation est connu, calcule les paramètres intrinsèques, le centre optique, l'axe de rotation et les coordonnées 3D de points à un facteur d'échelle près en effectuant au moins quatre

rotations de même axe, pour deux axes non confondus et en suivant au moins deux points.

Plus récemment, différents auteurs ont étudiés un certain nombre de cas particuliers. Sturm (Sturm, 1997a) a étudié les conditions de dégénérescence des méthodes d'auto-calibration générales et explicité l'ensemble des mouvements critiques pour lesquels les méthodes d'auto-calibration ne peuvent être appliquées. Horaud et Christy (Horaud et al., 1997) ont effectué des reconstructions euclidiennes et affines en utilisant différents mouvements contrôlés par un robot (mouvement connu de paramètres connus).

Le cas de la translation pure a été moins largement étudiée que la rotation. Cependant, Armstrong et Zisserman (Armstrong et al., 1994) ont étudié la translation ainsi que Pollefeys et Van Gool (Pollefeys et al., 1996) en combinant translation et zoom.

Sturm (Sturm, 1997a), Wiles et Brady (Wiles et Brady, 1996) et Armstrong et Zisserman (Armstrong et al., 1996) se sont intéressés aux reconstructions de scènes par mouvement planaire. Sturm montre qu'on ne peut lever d'ambiguïté sur les solutions sans information supplémentaire tandis que Wiles, Brady, Armstrong et Zisserman, ont étudié différents modèles de caméra pour le mouvement planaire ainsi que les simplifications des matrices fondamentales et homographiques qui s'en suivent. Sturm s'est ensuite également intéressé aux mouvements linéaires.

Plus généralement, un certain nombre d'auteurs ont étudiés des cas particuliers et explicités les simplifications apportées par ceux-ci dans les expressions de la matrice fondamentale ou de l'homographie afin de pouvoir d'une part éviter d'avoir des équations critiques, d'autre part de profiter d'un nombre plus faible de paramètres. Viéville et Lingrand (Viéville et Lingrand, 1995; Viéville et Lingrand, 1999) ont examiné ces cas particuliers de translations et rotations et mis en oeuvre des critères de sélection automatique du cas correspondant au mieux aux données. Torr et Zisserman (Torr et al., 1995) ont suivi cette voie afin d'améliorer leur méthode d'estimation de paramètres à base de l'algorithme robuste de RANSAC: PLUNDER (*Pick Least UNDEgenerate Randomly*). Torr s'est intéressé de près aux méthodes de sélection de modèle. Nous en reparlerons dans le chapitre 3. Clarke (Clarke, 1997) a étudié dans sa thèse un autre échantillon de cas particuliers de mouvement mais avec des paramètres intrinsèques constants et surtout sans expliciter de processus de sélection de cas.

2.2.4 Géométrie de la scène, de l'objet.

Le cas d'objets ou de scènes planaires a été largement étudié car il correspond dans le cas de modèle perspectif de caméra à l'étude des homographies, tout comme dans le cas de rotation pure. Gaspard et Viéville (Gaspard et Viéville, 1996) ont utilisé les homographies pour extraire différentes structures planaires dans des scènes monoculaires non calibrées. Triggs (Triggs, 1998a) a étudié l'auto-calibration dans le cas de scènes planaires mais avec des paramètres intrinsèques constants.

Torr et Fitzgibbon (Torr et al., 1998) ont étudié différentes hypothèses concernant les relations entre correspondances de points pour trois images consécutives en les considérant deux par deux : homographie puis homographie (H-H), homographie puis matrice fondamentale (H-F), matrice fon-

damentale puis homographie (F-H) ou encore matrice fondamentale puis matrice fondamentale (F-F). Ces auteurs proposent un mécanisme d'identification automatique d'un de ces cas, puis, en présence d'une homographie, à l'aide du critère GRIC (Torr, 1998), déterminent s'il s'agit d'une structure plane ou bien d'une rotation pure. Cet algorithme permet donc de s'affranchir du problème de suivi de points sur une séquence d'images et d'identifier des images posant problème pour retrouver la structure de façon sûre.

2.2.5 L'ensemble des cas particuliers.

Les cas particuliers que nous venons de présenter sont pour la plupart d'entre eux des combinaisons de cas particuliers concernant le modèle de projection, le mouvement des composantes de la scène ou bien la structure de la scène. Ceci est dû au fait que tous les paramètres sont ensuite regroupés dans une même équation (fondamentale, homographique, ...)

Sturm (Sturm, 1997a) a effectué une étude complète des cas critiques mais aucune étude exhaustive concernant l'ensemble des cas particuliers n'a été envisagée. C'est justement l'objet de ce chapitre.

Nous allons, dans un premier temps, examiner tous les cas particuliers de projection, mouvement et certains de structure afin, dans un deuxième temps, de déterminer l'ensemble des combinaisons de cas. Nous proposerons un module générique de traitement des cas particuliers afin de pouvoir ajouter d'autres cas particuliers dans l'avenir (par exemple, d'autres modèles de projection).

Enfin, nous utiliserons ces résultats pour déterminer les formes particulières de matrices fondamentales et homographiques

2.3 Les cas particuliers concernant la modélisation de la caméra.

Nous allons maintenant faire l'inventaire des cas particuliers en commençant par ceux concernant le modèle de projection de la caméra ainsi que les paramètres du modèle choisi.

2.3.1 Modèle de projection.

Nous nous intéressons ici aux paramètres modaux et intrinsèques. Nous avons déjà vu les valeurs possibles pour les paramètres modaux, ce qui se résume à trois cas : projection perspective, projection para-perspective et projection orthographique. Il faut cependant noter que les paramètres β_u et β_v n'existent que dans le cas de la projection para-perspective, on les fixera donc à la valeur 0 dans les autres cas de projection.

Nous noterons ces cas particuliers de la façon suivante, λ et μ ayant été définis page 30 :

p1	$\lambda = 0$ et $\mu = 0$	projection orthographique
p2	$\lambda = 1$ et $\mu = 0$	projection para-perspective
p3	$\lambda = 1$ et $\mu = 1$	projection perspective

sachant que nous avons déjà établi que **p1** est un cas particulier de **p2** lui-même cas particulier de **p2**, cas général.

2.3.2 Les paramètres intrinsèques.

Concernant les paramètres intrinsèques, au nombre de 5 ou 7, on peut faire plusieurs hypothèses. L'hypothèse de calibration revient à considérer que les paramètres sont connus et constants. Une hypothèse un peu plus générale et qui est souvent utilisée dans les cas d'auto-calibration consiste à supposer que les paramètres sont constants mais pas forcément connus. D'autres hypothèses concernent plus particulièrement certains paramètres.

2.3.2.1 Le point principal

La position du point principal, de coordonnées (u_0, v_0) est variable dans le cas général mais peut être considérée comme fixe dans certains cas. Si, de plus, on connaît sa position (par exemple au centre de l'image), on effectue alors un changement de repère image en prenant ce point pour origine afin d'annuler ces coordonnées.

2.3.2.2 Le paramètre γ .

Le paramètre γ est couramment négligé, ou au moins considéré comme constant, et son apport de précision est encore moindre dans les cas d'approximations de la projection perspective.

2.3.2.3 Les paramètres α_u et α_v .

Enciso (Enciso, 1995) a expérimentalement prouvé que pour un grand nombre de caméra, le rapport $\frac{\alpha_u}{\alpha_v}$ peut être considéré comme constant même lorsque d'autres paramètres varient. Ainsi, on pourra considérer une variation de focale à rapport $\frac{\alpha_u}{\alpha_v}$ constant. On exprimera la constance de ce rapport par l'égalité de ces valeurs: $f = \alpha_u = \alpha_v$, et on se ramènera au cas général par la transformation suivante :

$$\begin{pmatrix} \alpha_u & \gamma & \lambda \beta_u + \mu u_0 & (1 - \mu) u_0 \\ 0 & \alpha_v & \lambda \beta_v + \mu v_0 & (1 - \mu) v_0 \\ 0 & 0 & \mu & (1 - \mu) \end{pmatrix} = \begin{pmatrix} f & \gamma & \lambda \beta_u + \mu u_0 & (1 - \mu) u_0 \\ 0 & f & \lambda \beta_v + \mu v_0 & (1 - \mu) v_0 \\ 0 & 0 & \mu & (1 - \mu) \end{pmatrix} \cdot \begin{pmatrix} \frac{\alpha_u}{\alpha_v} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.3.2.4 Les paramètres β_u et β_v .

Dans tous les cas de projection, excepté celui de la projection para-perspective, les valeurs de β_u et β_v sont nulles.

Dans le cas de la projection para-perspective, nous rappelons leurs expressions :

$$\begin{cases} \beta_u & = & \alpha_u \frac{X_0}{Z_0} + \gamma \frac{Y_0}{Z_0} \\ \beta_v & = & \alpha_v \frac{Y_0}{Z_0} \end{cases}$$

Le rapport de ces paramètres vaut donc :

$$\frac{\beta_u}{\beta_v} = \frac{\alpha_u X_0 + \gamma Y_0}{\alpha_v Y_0}$$

On remarque alors que si on néglige γ devant $\alpha_u \frac{X_0}{Y_0}$, il vient :

$$\frac{\beta_u}{\beta_v} = \frac{\alpha_u}{\alpha_v} \frac{X_0}{Y_0}$$

qui est aussi un rapport constant, connu si on connaît $\frac{X_0}{Y_0}$.

Nous récapitulons dans le tableau 2.1, pour chacun des paramètres, les cas particuliers qui lui correspondent en notant avec un indice nul une valeur constante (par exemple θ_0) et préfixée par $_$ une valeur constante et connue (par exemple $_ \theta_0$). Le paramètre τ correspond à la variable temporelle ou numéro d'image dans la séquence vidéo.

g1	$\gamma = 0$	γ est constant et nul
g2	$\gamma = _ \gamma_0$	γ est constant et connu
g3	$\gamma = \gamma_0$	γ est constant
g4	$\gamma = \gamma(\tau)$	γ est libre
f1	$\alpha_v = 1$	α_v est constant et connu
f2	$\alpha_v = f_0$	α_v est constant
f3	$\alpha_v = \alpha_v(\tau)$	α_u est quelconque
s1	$\alpha_u = \alpha_v(\tau)$	le rapport $\frac{\alpha_u}{\alpha_v}$ est constant et connu
s2	$\alpha_u = k_\alpha \alpha_v(\tau)$	le rapport $\frac{\alpha_u}{\alpha_v}$ est constant
s3	$\alpha_u = \alpha_u(\tau)$	α_u est quelconque
b1	$\beta_v = 0$	β_v est constant et nul
b2	$\beta_v = _ \beta_0$	β_v est constant et connu
b3	$\beta_v = \beta_0$	β_v est constant
b4	$\beta_v = \beta_v(\tau)$	β_v est libre
B1	$\beta_u = \beta_v(\tau)$	β_u et β_v sont égaux
B2	$\beta_u = _ k_\beta \beta_v(\tau)$	le rapport $\frac{\beta_u}{\beta_v}$ est constant et connu
B3	$\beta_u = k_\beta \beta_v(\tau)$	le rapport $\frac{\beta_u}{\beta_v}$ est constant
B4	$\beta_u = \beta_u(\tau)$	le rapport $\frac{\beta_u}{\beta_v}$ est quelconque
c1	$u_0 = v_0 = 0$	u_0 et v_0 sont constants et connus (donc nuls)
c2	$u_0 = u_{0_0}$ et $v_0 = v_{0_0}$	u_0 et v_0 sont constants mais inconnus
c3	$u_0 = u_0(\tau)$ et $v_0 = v_0(\tau)$	u_0 et v_0 sont libres

TAB. 2.1 – Tableau des cas particuliers de paramètres pour 2 vues

Pour chacun des cas énumérés dans ce tableau, **Ai** est un cas particulier de **A(i+1)**, où **A** est une lettre parmi {g,f,s,b,B,c}.

2.4 Les cas particuliers de mouvement.

Le mouvement est paramétré par les paramètres extrinsèques de rotation et de translation définis au paragraphe 1.5.2.4. Nous allons tout d'abord énumérer les différents cas particuliers avant de proposer une paramétrisation adéquate.

2.4.1 Mouvement discret - mouvement continu.

Dans une séquence d'images, si le mouvement est très faible par rapport à la cadence des images, on peut approximer le mouvement à son premier ordre, ce qui concerne uniquement la rotation :

$$\mathbf{R} = e^{\tilde{\mathbf{r}}} = \mathbf{I} + \tilde{\mathbf{r}} + o(\tilde{\mathbf{r}})$$

Sous forme matricielle, cela donne :

$$\mathbf{R} = \begin{pmatrix} 1 & -r_2 & r_1 \\ r_2 & 1 & -r_0 \\ -r_1 & r_0 & 1 \end{pmatrix}$$

D'autre part, si le mouvement est faible mais pas assez pour être exprimé au premier ordre, on peut considérer alors son second ordre :

$$\mathbf{R} = \mathbf{I} + \tilde{\mathbf{r}} + \frac{\tilde{\mathbf{r}}^2}{2} + o(\tilde{\mathbf{r}}^2)$$

soit :

$$\mathbf{R} = \begin{pmatrix} 1 - (r_1^2 + r_2^2) & r_1 r_0 - r_2 & r_2 r_0 + r_1 \\ r_1 r_0 + r_2 & 1 - (r_0^2 + r_2^2) & r_2 r_1 - r_0 \\ r_2 r_0 - r_1 & r_2 r_1 + r_0 & 1 - (r_0^2 + r_1^2) \end{pmatrix}$$

2.4.1.1 A propos des paramètres extrinsèques

Les paramètres de rotation concernent l'axe de rotation et l'angle associé :

$$\mathbf{r} = 2 \tan \frac{\theta}{2} \mathbf{u}$$

où \mathbf{u} est un vecteur unitaire dépendant donc de deux paramètres.

Parmi les différents cas particuliers de la rotation, on trouve ceux concernant l'axe de rotation dont la direction ou seulement certaines composantes peuvent être connues et éventuellement nulles. Afin d'exprimer l'ensemble des transformations rigides, l'angle de rotation peut être amené à prendre des valeurs particulières correspondant par exemple à une absence de rotation (0) ce qui revient au cas de la rotation nulle ou un demi-tour (π) pour lequel l'expression de la rotation devient :

$$\mathbf{R} = \mathbf{I} + 2 \tilde{\mathbf{u}}^2$$

ou encore, un mouvement de vissage, correspondant au cas où l'axe de rotation est parallèle à la direction de translation, pour lequel l'angle de rotation vaut $\frac{\pi}{2}$.

Certains systèmes robotiques fournissent des valeurs très précises de l'angle de rotation, de son axe et du vecteur translation. On peut ainsi être dans un cas où l'angle de rotation est connu et vaut θ_0 . D'autre part, on peut connaître la direction de l'axe de rotation ou bien ne posséder qu'une information partielle d'orthogonalité avec une direction connue.

Les cas particuliers de la translation concernent ses degrés de liberté : translation dans l'espace, translation dans un plan ou bien translation selon une direction fixe. Ainsi, on peut considérer comme nulle une ou plusieurs composantes du vecteur selon un ou plusieurs axes du repère mais aussi par rapport à une direction \mathbf{g} connue.

2.4.2 Relations entre rotation et translation.

Des mouvements particuliers concernent les relations entre rotation et translation :

– l'axe de rotation est perpendiculaire au plan de la translation planaire :

$$\mathbf{r} \perp \mathbf{t} \Leftrightarrow \mathbf{r} \cdot \mathbf{t} = 0 \Leftrightarrow r_0 t_0 + r_1 t_1 + r_2 t_2 = 0$$

C'est le cas d'un mouvement planaire, par exemple.

– la translation est le long de l'axe de rotation (mouvement de vissage) :

$$\mathbf{r} \parallel \mathbf{t} \Leftrightarrow \exists \kappa / \mathbf{r} = \kappa \mathbf{t} \Leftrightarrow \exists \kappa / \begin{cases} r_0 = \kappa t_0 \\ r_1 = \kappa t_1 \\ r_2 = \kappa t_2 \end{cases}$$

Il faut préciser que, dans le cas général, un mouvement rigide peut toujours s'exprimer en terme de vissage. Ceci n'est pas vrai dans notre cas car on a imposé à l'axe de rotation de passer par l'origine du repère, à savoir le centre optique de la première caméra.

2.4.3 Expression des contraintes sur les paramètres de mouvement

L'objet de ce paragraphe est de mettre sous forme d'équations les différentes contraintes ou cas particuliers que nous venons d'énumérer.

La translation étant définie à sa norme près (donc ne dépendant que de deux paramètres), on divisera, par simplicité, chacune de ses composantes par une composante non nulle. De même, la norme du vecteur directeur de l'axe de rotation étant fixée, on cherchera un vecteur dépendant de deux paramètres et de norme quelconque; il suffira de normaliser celui-ci par la suite. En adoptant le même formalisme pour tous les vecteurs dépendant de deux paramètres uniquement, on obtient une façon d'exprimer les relations de parallélisme ou d'orthogonalité avec d'autres vecteurs de façon très simple puisque, dans le cas d'une translation non nulle, au moins une composante vaut 1 et le produit scalaire induit alors une relation linéaire. Par exemple, si $t_2 = 1$, l'orthogonalité de \mathbf{t} et \mathbf{r} s'exprime par : $t_0 u_0 + t_1 u_1 + u_2 = 0 \Rightarrow u_2 = -t_0 u_0 - t_1 u_1$.

Nous répertorions alors les différents cas particuliers concernant le mouvement dans le tableau suivant :

Tableau des cas particuliers de mouvement

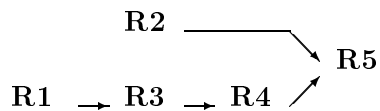
T1	$\mathbf{t} = \mathbf{0}$	pas de translation
T2	$\mathbf{t} = [t_0 \ t_1 \ t_2]^T$	translation
t1	$t_1 = t_2 = 0$ et $t_0 = 1$	translation selon l'axe x
t2	$t_1 = 0$ et $t_0 = 1$	translation orthogonale à l'axe y
t3	$t_2 = 0$ et $t_0 = 1$	translation orthogonale à l'axe optique
t4	$t_0 = 1$, t_1 et t_2 quelconques	translation générale
t5	$t_0 = t_2 = 0$ et $t_1 = 1$	translation selon l'axe y
t6	$t_0 = 0$ et $t_1 = 1$	translation orthogonale à l'axe x
t7	$t_2 = 0$ et $t_1 = 1$	translation orthogonale à l'axe optique
t8	$t_1 = 1$, t_0 et t_2 quelconques	translation générale
t9	$t_0 = t_1 = 0$ et $t_2 = 1$	translation selon l'axe optique
t10	$t_0 = 0$ et $t_2 = 1$	translation orthogonale à l'axe x
t11	$t_1 = 0$ et $t_2 = 1$	translation orthogonale à l'axe y
t12	$t_2 = 1$, t_0 et t_1 quelconques	translation générale
D1	$\mathbf{t} \cdot \mathbf{t} = 0$	translation orthogonale à une direction connue
D2	$\mathbf{t} \wedge \mathbf{t} = 0$	translation de direction connue
D3		pas de relation (cas général)
R1	$\mathbf{R} = \mathbf{I}$	pas de rotation
R2	$\mathbf{R} = \mathbf{I} + 2 \tilde{\mathbf{u}}^2$	retournement
R3	$\mathbf{R} = \mathbf{I} + \tilde{\mathbf{r}}$	au premier ordre
R4	$\mathbf{R} = \mathbf{I} + \tilde{\mathbf{r}} + \frac{1}{2} \tilde{\mathbf{r}}^2$	au deuxième ordre
R5	$\mathbf{R} = \mathbf{I} + \frac{\tilde{\mathbf{r}} + \frac{1}{2} \tilde{\mathbf{r}}^2}{1 + \frac{\mathbf{r}^T \mathbf{r}}{4}}$	cas général
r1	$\mathbf{r} = 2 \tan\left(\frac{\theta}{2}\right) \frac{\mathbf{u}}{\ \mathbf{u}\ }$	cas général
a0	$\theta = \frac{\pi}{2}$	quart de tour
a1	$\theta = _ \theta_0$	angle connu
a2	θ	angle quelconque
u1	$u_0 = u_2 = 0$ et $u_1 = 1$	axe parallèle à l'axe des y
u2	$u_0 = 0$ et $u_1 = 1$	axe orthogonal à l'axe des x
u3	$u_2 = 0$ et $u_1 = 1$	axe orthogonal à l'axe optique
u4	$u_1 = 1$	cas général
u5	$u_0 = u_2 = 0$ et $u_1 = -1$	axe parallèle à l'axe des y
u6	$u_0 = 0$ et $u_1 = -1$	axe orthogonal à l'axe des x
u7	$u_2 = 0$ et $u_1 = -1$	axe orthogonal à l'axe optique
u8	$u_1 = -1$	cas général

Tableau des cas particuliers de mouvement

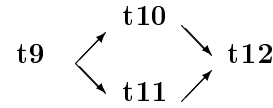
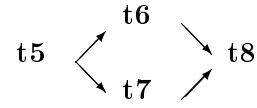
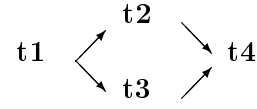
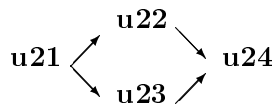
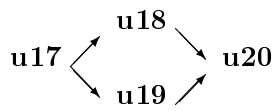
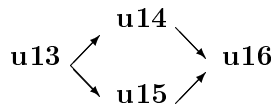
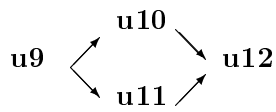
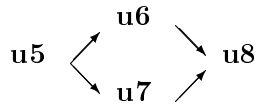
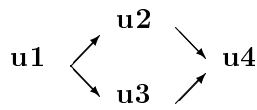
u9	$u_0 = u_1 = 0$ et $u_2 = 1$	axe parallèle à l'axe optique
u10	$u_0 = 0$ et $u_2 = 1$	axe orthogonal à l'axe des x
u11	$u_1 = 0$ et $u_2 = 1$	axe orthogonal à l'axe des y
u12	$u_2 = 1$	cas général
u13	$u_0 = u_1 = 0$ et $u_2 = -1$	axe parallèle à l'axe optique
u14	$u_0 = 0$ et $u_2 = -1$	axe orthogonal à l'axe des x
u15	$u_1 = 0$ et $u_2 = -1$	axe orthogonal à l'axe des y
u16	$u_2 = -1$	cas général
u17	$u_1 = u_2 = 0$ et $u_0 = 1$	axe parallèle à l'axe des x
u18	$u_1 = 0$ et $u_0 = 1$	axe orthogonal à l'axe des y
u19	$u_2 = 0$ et $u_0 = 1$	axe orthogonal à l'axe optique
u20	$u_0 = 1$	cas général
u21	$u_1 = u_2 = 0$ et $u_0 = -1$	axe parallèle à l'axe des x
u22	$u_1 = 0$ et $u_0 = -1$	axe orthogonal à l'axe des y
u23	$u_2 = 0$ et $u_0 = -1$	axe orthogonal à l'axe optique
u24	$u_0 = -1$	cas général
W1	$\mathbf{u} \cdot \mathbf{u} = 0$	axe de rotation orthogonal à une direction connue
W2	$\mathbf{u} \wedge \mathbf{u} = 0$	axe de rotation connu
W3		pas de relation (cas général)
Z1	$\mathbf{t} \cdot \mathbf{u} = 0$	translation orthogonale à l'axe de rotation
Z2	$\mathbf{t} \wedge \mathbf{u} = 0$	translation parallèle à l'axe de rotation
Z3		pas de relation (cas général)

Parmi les cas présentés dans ce tableau, une hiérarchie un peu plus complexe existe entre les différents cas d'une même famille (i.e. portant la même lettre) et est présentée dans les figures suivantes.

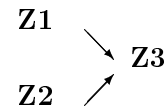
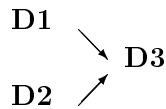
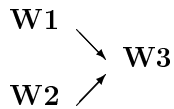
Par exemple, pour la construction de la matrice de rotation, les développements limités sont des cas particuliers de l'expression, tandis que les retournements (**R2**) sont à distinguer (on note par une flèche \rightarrow la relation "est cas particulier de") :



Pour les axes de rotation ou de translation ne dépendant que de 2 paramètres, on obtient les relations suivantes :



et enfin, on matérialise le fait que les relations d'orthogonalité et de parallélisme entre les vecteurs de rotation, de translation et direction connue sont indépendants, de la façon suivante :



2.5 Les cas particuliers géométriques.

Dans l'ensemble des équations qui nous ont permis de modéliser deux ou plusieurs vues, nous avons supposé que la scène était fixe ou bien que les objets étaient rigides. Cette hypothèse de rigidité se représente par :

- le mouvement euclidien (\mathbf{R}, \mathbf{t})
- 4 points non-coplanaires
- l'ensemble des points de la structure

Nous avons vu au chapitre 1 que la structure de l'ensemble des points de la scène tridimensionnelle peut avoir une influence sur la nature des équations. En particulier, si les points sont coplanaires et que, généralement, le plan défini par ces points ne passe pas par l'axe optique, la

matrice fondamentale n'existe plus et on a alors une relation homographique entre les deux images (voir 1.6.5) :

$$\forall(\mathbf{m}, \mathbf{m}'), \mathbf{m}' = \mathbf{H} \mathbf{m}$$

Pour représenter une structure plane, plusieurs modélisations existent :

- l'équation du plan $\mathbf{n}^T \mathbf{M} - d = 0$
- 3 points non alignés
- l'ensemble des points du plan

Une structure encore un peu plus particulière est la structure linéaire que l'on représente par, au choix :

- l'équation de la droite

$$\begin{cases} \mathbf{n}_1^T \mathbf{M} - d_1 = 0 \\ \mathbf{n}_2^T \mathbf{M} - d_2 = 0 \end{cases}$$

- 2 points distincts
- l'ensemble des points de la droite

On s'intéressera dans ce cas à la transformation entre les deux droites.

D'autres structures plus particulières de points existent mais nous nous focaliserons, dans le cadre de cette thèse uniquement à la structure plane car c'est un cas où l'équation fondamentale n'existe pas et la relation entre les points images est une homographie, comme nous l'avons déjà vu dans le tableau de la page 57.

Les plans sont paramétrisés par la normale au plan \mathbf{n} et la distance à l'origine d ce qui fait 3 paramètres. Des cas particuliers de cette situation concernent l'orientation du plan (plan fronto-parallèle, contenant l'axe optique, ...) ainsi que la distance d du plan à l'origine (centre optique). Ces paramètres correspondent à la première vue.

Nous récapitulons les différents cas de géométrie que nous considérerons dans la suite dans le tableau 2.3.

Nous verrons au paragraphe 2.7 comment générer les combinaisons de ces contraintes afin d'étudier l'ensemble des cas particuliers.

d1	$d = 0$	distance nulle
d2	$d = _d_0$	distance connue
d3	d quelconque	distance inconnue
n1	$\mathbf{n} = [1\ 0\ 0]^T$	plan orthogonal à l'axe x
n2	$\mathbf{n} = [0\ 1\ 0]^T$	plan orthogonal à l'axe y
n3	$\mathbf{n} = [0\ 0\ 1]^T$	plan fronto-parallèle
n4	$n_0 = 0$ et $n_1 = 1$	plan contenant l'axe x
n5	$n_0 = 0$ et $n_2 = 1$	plan contenant l'axe x
n6	$n_1 = 0$ et $n_0 = 1$	plan contenant l'axe y
n7	$n_1 = 0$ et $n_2 = 1$	plan contenant l'axe y
n8	$n_2 = 0$ et $n_0 = 1$	plan contenant l'axe optique
n9	$n_2 = 0$ et $n_1 = 1$	plan contenant l'axe optique
n10	$n_0 = 1$	cas général
n11	$n_1 = 1$	cas général
n12	$n_2 = 1$	cas général
n13		pas de normale (pas de plan) : cas général
l1	$\mathbf{n} \cdot \mathbf{t} = 0$	translation et normale parallèles
l2	$\mathbf{n} \wedge \mathbf{t} = 0$	translation parallèle au plan
l3		pas de relation (cas général)
m1	$\mathbf{n} \cdot \mathbf{u} = 0$	axe de rotation et normale parallèles
m2	$\mathbf{n} \wedge \mathbf{u} = 0$	axe de rotation parallèle au plan
m3		pas de relation (cas général)

TAB. 2.3 – Tableau des cas particuliers de paramétrisation d'une structure planaire.

2.6 Le cas de plus de 2 vues

2.6.1 Introduction: qu'en pensent les chercheurs?

2.6.1.1 Le cas de 3 vues

Nous avons vu précédemment que les relations entre correspondances de points de deux images s'expriment sous forme de relation fondamentale ou homographique. Très rapidement, des auteurs (Shashua, 1994; Hartley, 1994a; Faugeras et Robert, 1993) ont eu l'idée de rajouter une troisième vue afin d'obtenir des contraintes supplémentaires. En effet, comme illustré par la figure 2.1, si l'on dispose des matrices fondamentales entre les vues i et k ainsi que j et k , et si l'on connaît les projetés d'un point \mathbf{M} de l'espace dans les images i et j , alors le projeté dans l'image k est déterminé de façon unique par l'intersection des droites épipolaires :

$$\left. \begin{aligned} \mathbf{m}_k^T \mathbf{F}_{ik} \mathbf{m}_i &= 0 \\ \mathbf{m}_k^T \mathbf{F}_{jk} \mathbf{m}_j &= 0 \end{aligned} \right\} \Rightarrow \mathbf{m}_k = \mathbf{F}_{ik} \mathbf{m}_i \wedge \mathbf{F}_{jk} \mathbf{m}_j$$

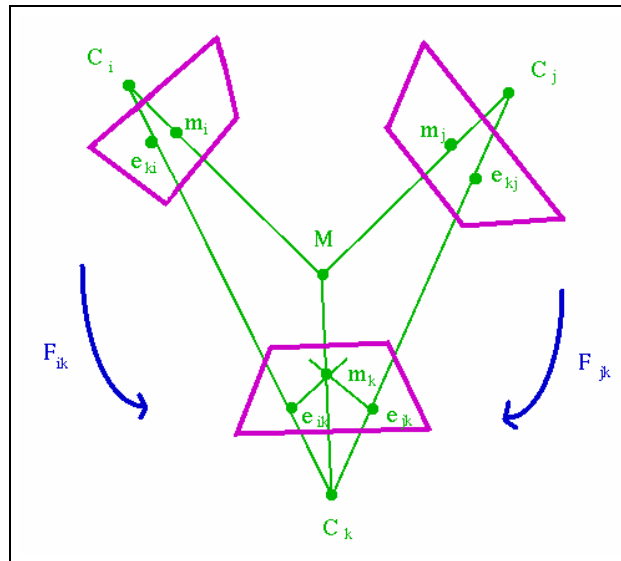


FIG. 2.1 – Cas de trois vues

De plus, on peut déterminer la tangente τ et la courbure en ce point en fonction des tangentes et courbures dans les autres images :

$$\tau_k = (\mathbf{F}_{ik} \mathbf{m}_i \wedge \mathbf{F}_{jk} \mathbf{m}_j) \wedge (\mathbf{F}_{ik} \mathbf{t}_i \wedge \mathbf{F}_{jk} (\mathbf{F}_{ij} \mathbf{t}_i \wedge (\mathbf{m}_j \wedge \mathbf{t}_j)))$$

Le lecteur intéressé consultera (Robert, 1993) pour une démonstration ainsi que pour l'expression de la courbure (plus complexe).

De nombreux auteurs ont alors utilisé ce formalisme : Torr (Torr, 1995) a étendu son algorithme PLUNDER aux trilinearités, Zeller (Zeller, 1996), Laveau (Laveau, 96), Armstrong et Zisserman (Armstrong et al., 1996) et encore bien d'autres.

2.6.1.2 Le cas de plus de 3 vues ou cas des N vues

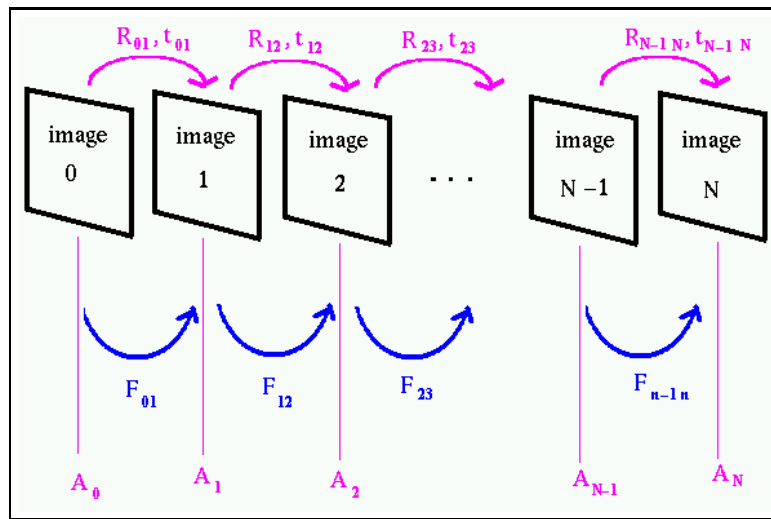


FIG. 2.2 – Séquence comportant N images

On se pose alors naturellement la question d'ajouter une quatrième vue puis N vues et d'étudier les multilinéarités.

Heyden (Heyden, 1995) a généralisé les contraintes épipolaires à une séquence d'images par une formalisation des contraintes multilinéaires pour simplifier l'analyse. Il utilise en premier temps une réduction projective et étudie les cas discrets et continus (développement de Taylor des équations précédentes).

Malheureusement, ajouter une quatrième vue n'apporte pas de contrainte supplémentaire. Triggs (Triggs, 1995), Faugeras et Mourrain (Faugeras et Mourrain, 1995b) ont expliqué que les quadrilinéarités sont dépendantes des bi- et trilinearités. Ainsi, pour N vues, on ne considérera que les bi- et tri-linéarités. D'autre part, Heyden (Heyden et Aström, 1996) a montré que pour $N \geq 4$, seule C_2^N bilinéarités sont à considérer.

Il est cependant clair que si plus de 3 images n'apporte pas de contraintes au niveau algébrique, l'apport de nouvelles vues comporte des avantages certains :

- les parties occultées de la scène seront évidemment d'autant plus visibles que l'on accumulera les points de vues différents
- les parties ne vérifiant pas les contraintes de la géométrie épipolaire (voir 1.6.1) par suite de configuration singulière seront vues correctement si l'on dispose de plusieurs vues
- la détermination de paramètres du problème lorsqu'ils sont trop nombreux pour être déterminés par deux ou trois vues (par exemple en effectuant différents mouvements de rotation (Viéville, 1994))
- le raffinement numérique de l'estimation de la structure tridimensionnelle

De plus, dans un but d'application de réalité virtuelle, deux ou trois vues ne sont pas suffisantes si on désire effectuer une reconstruction sous tous les angles et s'affranchir un maximum du problème

des occlusions. Kanade et son équipe (Kanade et al., 1998) ont construit un dôme composé de 52 caméras afin d'effectuer une reconstruction permettant de mélanger événements réels et de synthèse et de choisir n'importe quel point de vue. Les caméras sont calibrées et leur position est connue. Une nouvelle séquence d'images prise d'un point de vue virtuel est réalisée à partir de 52 séquences d'images synchronisés. Olague et Mohr (Olague et Mohr, 1998) ont étudié comment placer de façon optimale un ensemble de N caméras.

Nous considérerons dans la suite que les n vues sont issues d'une séquence d'images monoculaires. Différentes approches ou modélisations du problème existent. Nous allons décrire certaines d'entre elles, en particulier celles qui concernent des points d'intérêt mis en correspondance.

Mise en correspondance de points sur une séquence d'images : Pour cela, plusieurs approches coexistent. On peut considérer simplement le problème de mise en correspondance pour des images prises deux à deux. Sachant par ailleurs que les trilinearités apportent des contraintes supplémentaires, on peut alors songer à considérer les images trois par trois. Lorsqu'il s'agit d'une séquence d'images avec forte disparité d'une image à l'autre, on propagera ainsi les correspondances. Par contre, à moins d'un mouvement de va-et-vient, chaque point n'aura des correspondants que dans quelques images. C'est un cas qui nous permettra d'avoir une vue bien plus large mais qui ne permettra pas d'obtenir plus de précision sur les paramètres de mouvement, de caméra ou de structure. Diverses techniques de fusion des données ainsi recueillies existent mais ne font pas partie du sujet abordé ici.

Si la séquence vidéo ne présente pas de forte disparité entre deux images, on va pouvoir suivre les correspondances sur un plus grand nombre de vues et des algorithmes plus simples que ceux évoqués précédemment peuvent être mis en oeuvre. Les algorithmes travaillant "au vol" sur les séquences d'images, c'est-à-dire traitant les images au fur et à mesure de leur acquisition, calculent une estimée de la position future puis, lorsque l'image suivante est disponible, réajustent au besoin leur mécanisme de précision. C'est le principe du filtre de Kalman que nombre d'auteurs ont utilisé (Viéville et al., 1994; Tomasi et Kanade, 1992; Cui et al., 1990; Azarbayejani et P., 1995; MacLauchlan et Murray, 1995).

Le problème des correspondances au long d'une séquence a été traité par différents auteurs dont Laveau (Laveau, 96), Faugeras et Mourrain (Faugeras et Mourrain, 1995a) ainsi que plus récemment, Clarke (Clarke, 1997).

Nous allons maintenant nous intéresser, dans les deux paragraphes suivants, à deux approches distinctes : l'une concerne les vues comme étant distinctes, l'autre considère les différentes images comme formant un volume continu.

Cas discret des N vues Une première modélisation est due à Viéville, Luong et Faugeras (Viéville et al., 1996a) qui introduisent une représentation dite Q_s des relations discrètes entre images

d'une séquence monoculaire :

$$Z_i \mathbf{m}_i = Z_j \underbrace{\mathbf{A}_i \mathbf{R}_{ij} \mathbf{A}_j^{-1}}_{\mathbf{Q}_{ij}} \mathbf{m}_j + \underbrace{\mathbf{A}_i \mathbf{t}_{ij}}_{\mathbf{s}_{ij}}$$

\mathbf{Q}_{ij} représente la partie rotationnelle non calibrée du mouvement tandis que \mathbf{s}_{ij} représente la partie translationnelle non calibrée. A l'époque, ce formalisme constituait en fait la généralisation la plus

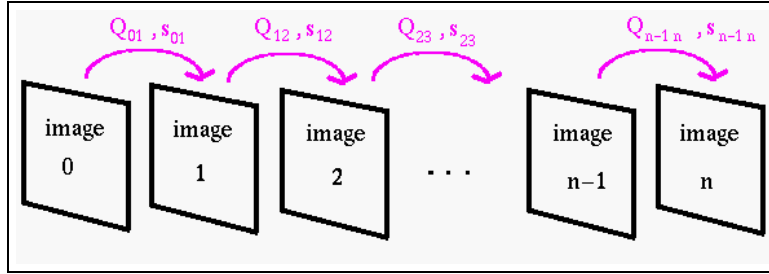


FIG. 2.3 – Représentation en \mathbf{Qs} du mouvement rétinien dans une séquence d'images

simple de ce qui était connu dans le cas calibré. Cette représentation permet, pour le cas d'une séquence de N images de déterminer ses $11N - 4$ paramètres reliés au mouvement rétinien et aux proximités des points, correspondant aux profondeurs rétiniennes.

Christy et Horaud (Christy et Horaud, 1994) proposent une méthode de reconstruction incrémentale (modèle de projection para-perspectif ou affine).

Sturm, (Sturm, 1997b) chapitre 5, effectue une factorisation en sous-séquences suivant la méthode de Tomasi et Kanade (Tomasi et Kanade, 1992), avec un modèle de projection orthographique en initialisation puis revient au modèle projectif ensuite. Sturm effectue une décomposition en valeurs singulières de la matrice \mathbf{W} des mesures en utilisant le rang de celle-ci (voir 2.4). En notant \mathbf{m}_{ij} le i ème point 2D de la j ème image, Sturm définit la matrice des mesures \mathbf{W} par, m étant le nombre de points par image :

$$\mathbf{W} \equiv \begin{pmatrix} \mathbf{m}_{11} & \mathbf{m}_{12} & \dots & \mathbf{m}_{1m} \\ \mathbf{m}_{21} & \mathbf{m}_{22} & \dots & \mathbf{m}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{m}_{N1} & \mathbf{m}_{N2} & \dots & \mathbf{m}_{Nm} \end{pmatrix}_{2N \times m} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \dots \\ \mathbf{P}_N \end{pmatrix}_{2N \times 3} \begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 & \dots & \mathbf{M}_m \end{pmatrix}_{3 \times m}$$

Triggs (Triggs, 1998a) détermine une fonction de coût sur l'ensemble des m images et cherche à minimiser celle-ci par rapport à deux directions de base et tout sous-ensemble des paramètres intrinsèques. Trois images suffisent si seule la focale varie, 5 si tous les paramètres varient. Cet algorithme fonctionne relativement bien dans les cas dégénérés mais ne permet pas de détecter ceux-ci.

Cas continu des N vues Lorsque les N vues sont issues d'une séquence d'images, il est envisageable de considérer les N vues comme un volume continu. Certains auteurs comme Viéville et

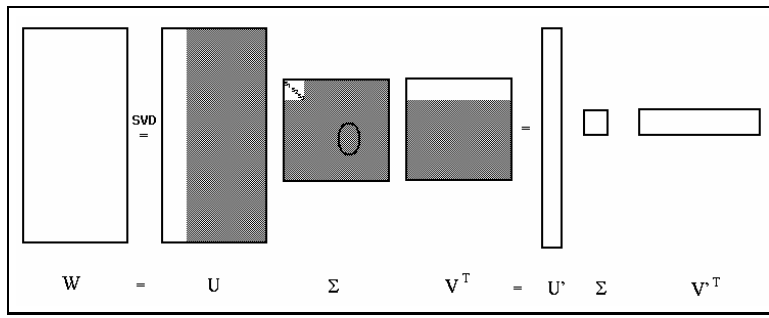


FIG. 2.4 – *Principe de la factorisation de Sturm (dans le cas affine). La matrice des mesures \mathbf{W} est de rang 3 (possède alors 3 valeurs singulières : s_1 , s_2 et s_3). Les parties grisées sont donc inutiles.*

Faugeras (Viéville et Faugeras, 1996) ont utilisé l'approximation au premier ordre des équations dans le cas non calibré afin de stabiliser les images en rotation. Ces auteurs ont montré que les résultats sont similaires au cas discret, excepté le fait que le foyer d'expansion est mis en évidence dans le cas continu.

Autre approche continue, celle de Åström et Heyden (Åström et Heyden, 1998) qui ont étudiés les développements de Taylor du mouvement 3D (camera ou scène) et du mouvement des points dans les images (mouvement 3D). Ces auteurs ont été amenés à présenter deux cas particuliers ou simplifications, réduction affine ou réduction projective par rapport à l'image d'une configuration planaire, ceci afin de déterminer le mouvement 3D.

2.6.2 Variations du mouvement et du zoom dans le cas de N vues

Dans une séquence d'images, on considère qu'une unité de temps τ sépare deux images consécutives.

Lors d'une séquence d'images, en plus des cas particuliers concernant les images deux à deux déjà étudiés aux paragraphes 2.3, 2.5 et 2.4, nous avons des cas particuliers concernant les variations sur plusieurs images que nous allons maintenant étudier.

Concernant les paramètres intrinsèques, dans le cas où $\frac{\alpha_u}{\alpha_v}$ est considéré comme constant, ce qui est couramment vérifié, nous allons remplacer α_u et α_v par $f \alpha_u$ et $f \alpha_v$ pour mettre en évidence le facteur de zoom des paramètres de projection. On pourra ainsi considérer un mouvement de zoom à α_u et α_v constants. Par contre, de nombreuses études expérimentales ont été faites pour modéliser les variations du point principal mais ont toutes conduit à des échecs car les mécanismes internes de zoom sont trop complexes pour être modélisés effectivement.

Les cas particuliers à ajouter pour l'étude de plusieurs vues concernent donc le mouvement et le zoom :

- le mouvement est nul (séquence fixe)
- le mouvement est constant au cours de la séquence
- le mouvement est à accélération constante (le mouvement est une fonction affine du temps)
- le mouvement varie de façon continue

- le mouvement est quelconque
- le zoom est fixe
- le zoom varie à vitesse constante
- le zoom varie à accélération constante
- le zoom varie de façon continue
- le zoom varie de façon quelconque

On va en réalité appliquer ces cas aux composantes du mouvement, les vecteurs de rotation et translation ainsi qu'au facteur de zoom (τ est la variable temporelle, a_i , b_i et c_i sont des constantes) :

- $r, t_{0,1,2}, f = 0$
- $r, t_{0,1,2}, f = a_{(r,t,f),(0,1,2)}$
- $r, t_{0,1,2}, f = b_{(r,t,f),(0,1,2)} \tau + a_{(r,t,f),(0,1,2)}$
- $r, t_{0,1,2}, f = c_{(r,t,f),(0,1,2)} \tau^2 + b_{(r,t,f),(0,1,2)} \tau + a_{(r,t,f),(0,1,2)}$
- $r, t_{0,1,2}, f = \varphi_{(r,t,f),(0,1,2)}(\tau)$ où φ est continue
- $r, t_{0,1,2}, f = \psi_{(r,t,f),(0,1,2)}(\tau)$ où ψ est quelconque

D'autres cas sont plus difficiles à mettre en oeuvre et concernent la variation de l'axe de rotation et de l'angle de rotation qui peut se faire à vitesse constante, à accélération constante ou selon une loi connue (modélisée par une fonction φ , ψ , ...).

- variations particulières de l'axe de rotation (\mathbf{u}_0 , \mathbf{u}_1 et \mathbf{u}_2 sont des vecteurs constants)
 - $\frac{\mathbf{r}}{\|\mathbf{r}\|} = \mathbf{u}_0$
 - $\frac{\mathbf{r}}{\|\mathbf{r}\|} = \mathbf{u}_0 + \mathbf{u}_1 \tau$
 - $\frac{\mathbf{r}}{\|\mathbf{r}\|} = \mathbf{u}_0 + \mathbf{u}_1 \tau + \mathbf{u}_2 \tau^2$
 - $\frac{\mathbf{r}}{\|\mathbf{r}\|} = \varphi_{\mathbf{r}}(\tau)$
- variations particulières de l'angle de rotation θ (θ_0 , θ_1 et θ_2 sont des constantes)
 - $\theta = \theta_0$
 - $\theta = \theta_0 + \theta_1 \tau$
 - $\theta = \theta_0 + \theta_1 \tau + \theta_2 \tau^2$
 - $\theta = \varphi_{\theta}(\tau)$

2.6.3 Relations entre rotation et translation dans le cas de N vues

Outre les relations examinées au paragraphe 2.4.2, on peut considérer des cas supplémentaires (ρ représente une direction constante, ρ_0 une direction constante et connue, α un angle constant et α_0 un angle constant et connu) :

- $\mathbf{r} \parallel \rho$: l'axe de rotation est fixe
- $\mathbf{r} \parallel \rho_0$: l'axe de rotation est fixe et connu
- $\mathbf{r} \perp \rho$: l'axe de rotation est orthogonal à une direction fixe
- $\mathbf{r} \perp \rho_0$: l'axe de rotation est orthogonal à une direction fixe et connue

- $\theta = \alpha$: l'angle de rotation est constant
- $\theta = \alpha_0$: l'angle de rotation est constant et connu
- $\mathbf{t} - \rho$: le plan de translation est fixe
- $\mathbf{t} - \rho_0$: le plan de translation est fixe et connu
- $\mathbf{t} \parallel \rho$: l'axe de translation est fixe
- $\mathbf{t} \parallel \rho_0$: l'axe de translation est fixe et connu
- $\mathbf{r} \parallel \mathbf{t}$: axe de rotation et translation varient mais sont toujours parallèles
- $\mathbf{r} - \mathbf{t}$: axe de rotation et translation varient mais sont toujours orthogonaux

Un autre point concerne les systèmes non robotiques ou tout du moins les systèmes ne fournissant pas une mesure précise du déplacement pour lesquels on aura une estimation grossière du mouvement ou de variation de la focale. On va vouloir alors prendre en compte ces valeurs approchées et éventuellement considérer des développements limités autour de ces valeurs. Un exemple est un mouvement d'un piéton tenant une caméra et mesurant au sol la distance parcourue (exemple:30m à la taille des pieds près et vitesse constante au temps d'un pas près) ou un voiture dans laquelle est posée une caméra et qui parcourt une certaine distance dans une direction fixe et sur un sol quasiment plat.

2.7 Gestion de l'inventaire des cas particuliers.

Nous venons de faire un inventaire exhaustif de cas particuliers isolés. Nous allons maintenant nous intéresser aux combinaisons de ces cas particuliers.

2.7.1 Contraintes atomiques - contraintes moléculaires

Nous avons, dans le cas général, pour 2 images, 2 paramètres modaux, 2*7 paramètres intrinsèques et 5 paramètres extrinsèques. En supposant le repère attaché à la première image, cela nous fait 19 paramètres. Il est clair que ce n'est pas la matrice fondamentale F qui nous permettra d'obtenir ces paramètres. Cependant, considérant suffisamment de points, les équations (1.15) nous permettent de déterminer \mathbf{P}_1 (donc les paramètres modaux et les paramètres intrinsèques relatifs à la première image) et \mathbf{P}_2 (permettant de retrouver les paramètres intrinsèques relatifs à la seconde image et les paramètres extrinsèques).

L'ensemble des contraintes est formé des contraintes dites atomiques car élémentaires (vues précédemment) et des contraintes dites moléculaires constituées par des "ET" logiques (\otimes) entre contraintes atomiques ceci selon certaines règles : on élimine les combinaisons contradictoires (voir paragraphe 2.7.2.2) et les combinaisons redondantes (voir paragraphe 2.7.2.3) en ne considérant que le résultat de la combinaison et on impose des conditions d'adéquation à une réalité physique :choix obligatoire d'un modèle de caméra, d'un mode de rotation et non nullité de la translation lors de l'estimation de \mathbf{F} dans le cas perspectif, ...

Les contraintes atomiques sont de deux types :

substitution : $a: i \rightarrow$. (exemple: paramètre constant)

contrainte : $a(i) \cdot x = 0$ (exemple: orthogonalité de deux vecteurs)

et l'expression des vecteurs montrée dans les paragraphes précédents (en fixant une composante à 1) permet toujours de résoudre les contraintes.

2.7.2 Conditions d'existence.

Après avoir combiné les cas particuliers atomiques pour obtenir un ensemble de cas particuliers moléculaires, nous allons nous maintenant à la signification de ces combinaisons. On va se demander en premier lieu si elles correspondent effectivement à une réalité physique c'est-à-dire s'il s'agit effectivement d'un cas qu'on peut rencontrer. On s'intéressera ensuite aux différentes combinaisons produisant un cas moléculaire identique et on essaiera d'établir des liens hiérarchiques entre les combinaisons.

2.7.2.1 conditions physiques

La figure 2.5 propose un exemple de mouvements particuliers effectués par divers systèmes robotiques. Un autre système robotique sera présenté en 4.1.1.

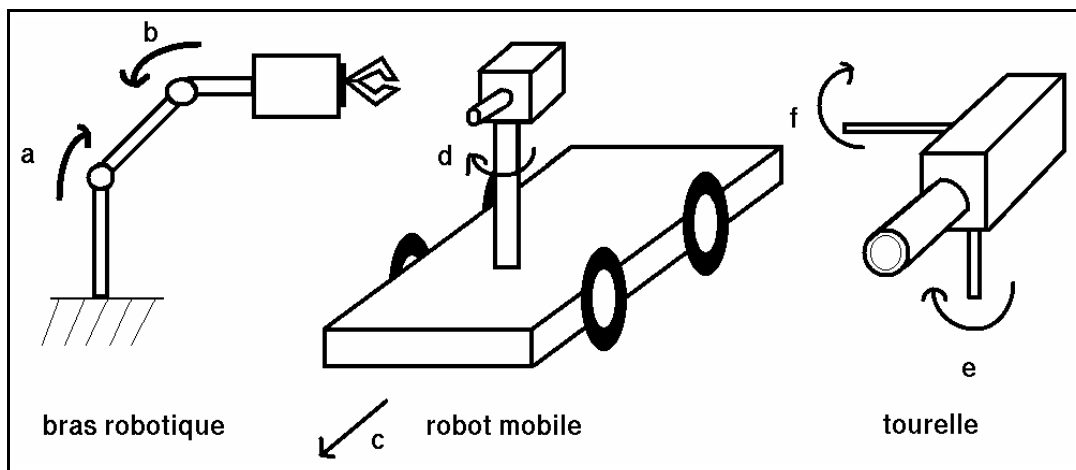


FIG. 2.5 – Exemples de cas particuliers physiquement réalisables en utilisant des mécanismes robotiques générant des translations pures, des rotations pures ou translations dans le plan orthogonal à l'axe de rotation. Si, pour le bras articulé, (a) et (b) ont des valeurs opposées, c'est une translation pure. Effectuant la même commande (c) sur les roues du robot mobile, on obtient une translation. Un déplacement de (a) seul, ou (b) seul, sur le bras robotique fait que la translation est dans le plan orthogonal à l'axe de rotation. Une commande (c) appliquée de façon opposée aux roues du robot mobile provoque le même déplacement. Les tourelles peuvent effectuer des rotations pures en pan (e) ou tilt (f) autour d'un centre optique approximatif.

La réalité physique d'une combinaison de cas particuliers est obtenue en contraignant certains choix de cas particuliers atomiques. En particulier, on choisira :

- un et un seul modèle de caméra (perspectif, para-perspectif, orthographique)
- un et un seul mode de rotation (nulle, ordre 0, 1, 2 ou ∞)
- un et un seul type de structure (quelconque, plane, linéaire)
- une et une seule quantité \mathbf{F} ou \mathbf{H} soit :
 - \mathbf{H} dans les cas suivants :
 - structure plane ne comprenant pas l'axe optique
 - mouvement de rotation pure dans le cas de la projection perspective
 - mouvement de rotation autour de l'axe du regard (translation quelconque) dans le cas de la projection para-perspective
 - mouvement de rotation autour de l'axe optique (translation quelconque) dans le cas de la projection orthographique
 - \mathbf{F} dans les autres cas
- un et un seul type de variation par composante dans le cas de plus de 2 vues

2.7.2.2 condition de compatibilité

Des combinaisons des contraintes examinées dans ce chapitre peuvent être incohérentes. Considérons l'exemple suivant :

$$(\mathbf{r} - \mathbf{t}) \otimes (\mathbf{r} \parallel \mathbf{t}) \otimes (\mathbf{r} \neq 0) \otimes (\mathbf{t} \neq 0)$$

Les deux premières contraintes imposent

$$(\mathbf{r} \neq 0) \oplus (\mathbf{t} \neq 0)$$

ce qui est contradictoire avec les deux dernières contraintes.

On éliminera ainsi toute contrainte moléculaire aboutissant à des impossibilités, ce qui se fait simplement en formalisant ces contraintes sous Maple.

2.7.2.3 notion de redondance

Si on compare :

$$(r_0 = 0) \otimes (\mathbf{t} - \mathbf{r})$$

et

$$(t_1 = 0) \otimes (t_2 = 0) \otimes (\mathbf{t} - \mathbf{r})$$

on s'aperçoit qu'on obtient la même contrainte dans les deux cas, ce qui montre une redondance des contraintes moléculaires, ne permettant pas ainsi de nommer les contraintes moléculaires en fonction des contraintes atomiques les constituant.

2.7.3 Consistance

D'autres part, les contraintes ne sont pas forcément suffisantes pour résoudre le problème. Dans ce cas, on peut, soit se limiter aux éléments déterminables, soit ajouter des contraintes "par défaut" afin d'effectuer la résolution complète.

Une application importante est la reconstruction hiérarchique. En effet, on va modéliser des cas particuliers sous forme d'opérateurs nécessitant des paramètres connus en entrée et fournissant en sortie des paramètres supplémentaires. Il suffit ensuite d'emboîter les boîtes en cascade les unes derrière les autres.

2.8 Les cas particuliers dans le cas de 2 vues.

Nous présentons ici quelques formes particulières de matrice fondamentale courantes dans le tableau 2.4.

nomenclature	paramétrisation	nombre de paramètres indépendants
p3f1s1b1B1c2T2t[2;6;8;13]R1	$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	0
p3f1s1b1B1c2T2t[9-10]R1	$\begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ -1 & -b & 0 \end{pmatrix}$	1
p3f1s1b1B1c2T2t[2;6;8;13]R1	$\begin{pmatrix} 0 & c & a \\ -c & 0 & b \\ -a & -b & 0 \end{pmatrix}$	2
p3f[1-2]s1b1B1c2T2t[9-10]R5u[9-10]a3	$\begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{pmatrix}$	4

TAB. 2.4 – *Formes particulières de matrice fondamentale.*

Nous présentons également quelques formes particulières connues de matrice d'homographie dans le tableau 2.5.

Le lecteur intéressé trouvera en annexe B une analyse plus détaillée de certains cas particuliers courants.

Nous cherchons maintenant toutes les formes particulières de matrice fondamentale. Pour cela, nous générons l'ensemble des contraintes moléculaires qui satisfont les contraintes d'existence et qui ne sont pas formées de contraintes incompatibles. Le lecteur aventureux trouvera le code Maple correspondant en annexe A.5. Nous considérons par formes particulières, les matrices dont certains

nomenclature	paramétrisation	nombre de paramètres indépendants
f[1-2]s1b1B1c3T1R1	\mathbf{I}	0
p3f1s1b1B1c2T2t[9-10]R1	$\begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ 0 & 0 & 1 \end{pmatrix}$	2
p3f[1-2]s1b1B1c3T1R1	$\begin{pmatrix} c & 0 & a \\ 0 & c & b \\ 0 & 0 & 1 \end{pmatrix}$	4
p3f1s1b1B1c2T2t[9-10]R5u[9-10]a3	$\begin{pmatrix} c & d & a \\ -d & c & b \\ 0 & 0 & 1 \end{pmatrix}$	4
p3f[1-2]s1b1B1c3T2t[9-10]R5u[9-10]a3	$\begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix}$	6

TAB. 2.5 – *Formes particulières de matrice d'homographie.*

coefficients sont nuls ou égaux ou opposés à d'autres coefficients, réduisant ainsi leur nombre de degrés de liberté.

Peut-être existe-t-il d'autres cas où des changements de variables exhibent des degrés de liberté plus faibles. Cependant, il n'existe pas d'algorithme mathématique pour essayer tous les changements de variables possibles, mais surtout, il n'est pas évident que ces cas apportent beaucoup d'information.

Prenons les expressions des contraintes formulées aux paragraphes 2.3, 2.4 et 2.5. La contrainte d'existence est respectée en prenant une contrainte dans chaque famille. Les cas moléculaires sont alors générés. Globalement, cela fait 17.000.000.000 cas, parmi lesquels on a un certain nombre de cas redondants.

2.8.1 Les formes particulières de matrices fondamentales.

On va, dans un premier temps, s'intéresser aux matrices fondamentales, donc éliminer les cas pour lesquels l'expression de cette dernière est nulle. De plus, sachant qu'on ne s'intéresse qu'aux formes de ces matrices, on ne tiendra pas compte des cas où une constante est connue ou non si ces cas ne diffèrent pas algébriquement. Dans le cas de la projection perspective, cela revient à s'intéresser aux cas de la forme :

p3g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[1-5]u[1-4;9-12;17-20]Z[1-3]W3D3a2n913m3

ce qui fait un total de 77760 cas.

Dans le cas de la projection para-perspective, on conserve les cas de la forme :

p2g[1-2;4]f[1;3]s[1;3]b[1;3-4]B[1;3-4]c[1-3]T2t[1-12]r1U1R[1-5]u[1-4;9-12;17-20]Z[1-3] \

W3D3a2n9l3m3

soit 699840 cas.

Finalement, dans le cas orthographique, on s'intéresse aux cas de la forme :

$p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n9l3m3$

et

$p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n9l3m3$

soit 44928 cas.

Dans le cas perspectif, et en ne considérant, pour des raisons de temps de calcul, que les cas $p3g1f1s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[1-5]u[1-4;9-12;17-20]Z[1-3]W3D3a2n9l3m3$

on obtient, parmi ces 12960 cas, 7089 cas pour lesquels des simplifications ont été apportés.

On présente la répartition des formes simplifiées dans le tableau suivant et on précise, pour chaque forme, un exemple de cas particulier correspondant à cette forme et le nombre de paramètres dont dépend la matrice fondamentale, sans tenir compte du fait que le déterminant de cette dernière est nul ni qu'elle est définie à un facteur d'échelle près. D'autre part, la contrainte sur le déterminant ne fait pas diminuer le nombre de degrés de liberté dans tous les cas. Ainsi, si on note par d le nombre de paramètres indiqué dans le tableau suivant, le nombre de degrés de liberté effectif est soit $d - 1$, soit $d - 2$. On verra au chapitre suivant que ce nombre de degrés de liberté est important pour la comparaison des modèles. Dans un premier temps, nous avons échappé à ce problème en ne tenant pas compte de la contrainte sur le déterminant, sauf dans le cas général.

nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
1	$[0\ 0\ 0\ 0\ 0\ x6\ 0\ -x6\ 0\]$	$p3g1f1s1b1B1c1T2t1r1U1R1u10Z2W3D3a2n9l3m3$	177
1	$[0\ 0\ x3\ 0\ 0\ 0\ -x3\ 0\ 0\]$	$p3g1f1s1b1B1c1T2t5r1U1R1u10Z2W3D3a2n9l3m3$	88
1	$[0\ x2\ 0\ -x2\ 0\ 0\ 0\ 0\ 0\]$	$p3g1f1s1b1B1c1T2t10r1U1R1u11Z1W3D3a2n9l3m3$	44
		TOTAL de cas pour 1 paramètre	309
2	$[0\ 0\ 0\ 0\ 0\ x6\ 0\ -x6\ x9\]$	$p3g1f1s1b1B1c3T2t1r1U1R1u10Z2W3D3a2n9l3m3$	88
2	$[0\ 0\ 0\ 0\ 0\ x6\ 0\ x8\ 0\]$	$p3g1f1s1b1B1c1T2t1r1U1R2u1Z2W3D3a2n9l3m3$	8
2	$[0\ 0\ 0\ 0\ 0\ x6\ x7\ -x6\ 0\]$	$p3g1f1s1b1B1c1T2t1r1U1R3u9Z2W3D3a2n9l3m3$	4
2	$[0\ 0\ 0\ 0\ x5\ x6\ 0\ -x6\ x5\]$	$p3g1f1s1b1B1c1T2t1r1U1R3u17Z1W3D3a2n9l3m3$	66
2	$[0\ 0\ 0\ x4\ 0\ x6\ 0\ -x6\ 0\]$	$p3g1f1s1b1B1c1T2t1r1U1R3u1Z2W3D3a2n9l3m3$	4
2	$[0\ 0\ x3\ 0\ 0\ 0\ -x3\ 0\ x9\]$	$p3g1f1s1b1B1c3T2t5r1U1R1u10Z2W3D3a2n9l3m3$	44
2	$[0\ 0\ x3\ 0\ 0\ 0\ -x3\ x8\ 0\]$	$p3g1f1s1b1B1c1T2t5r1U1R3u9Z2W3D3a2n9l3m3$	2
2	$[0\ 0\ x3\ 0\ 0\ 0\ x7\ 0\ 0\]$	$p3g1f1s1b1B1c1T2t5r1U1R2u17Z2W3D3a2n9l3m3$	52
2	$[0\ 0\ x3\ 0\ 0\ x6\ -x3\ -x6\ 0\]$	$p3g1f1s1b1B1c1T2t3r1U1R1u10Z2W3D3a2n9l3m3$	144
2	$[0\ x2\ 0\ -x2\ 0\ x6\ 0\ -x6\ 0\]$	$p3g1f1s1b1B1c1T2t11r1U1R1u10Z2W3D3a2n9l3m3$	72
2	$[0\ x2\ 0\ -x2\ 0\ x6\ 0\ 0\ 0\]$	$p3g1f1s1b1B1c1T2t9r1U1R3u1Z2W3D3a2n9l3m3$	2
2	$[0\ x2\ 0\ x4\ 0\ 0\ 0\ 0\ 0\]$	$p3g1f1s1b1B1c1T2t9r1U1R2u17Z2W3D3a2n9l3m3$	52

suite page suivante

nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
2	$[0 \ x2 \ x3 \ -x2 \ 0 \ 0 \ -x3 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t10r1U1R1u10Z1W3D3a2n9l3m3	72
2	$[0 \ x2 \ x3 \ -x2 \ 0 \ 0 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R3u17Z2W3D3a2n9l3m3	2
2	$[0 \ x2 \ x3 \ 0 \ 0 \ 0 \ -x3 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t5r1U1R3u17Z2W3D3a2n9l3m3	2
2	$[x1 \ 0 \ x3 \ 0 \ 0 \ 0 \ -x3 \ 0 \ x1 \]$	p3g1f1s1b1B1c1T2t5r1U1R3u1Z1W3D3a2n9l3m3	33
2	$[x1 \ x2 \ 0 \ -x2 \ x1 \ 0 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t10r1U1R3u11Z1W3D3a2n9l3m3	33
		TOTAL de cas pour 2 paramètres	680
3	$[0 \ 0 \ 0 \ 0 \ 0 \ x6 \ 0 \ x8 \ x9 \]$	p3g1f1s1b1B1c2T2t1r1U1R2u1Z2W3D3a2n9l3m3	16
3	$[0 \ 0 \ 0 \ 0 \ 0 \ x6 \ x7 \ -x6 \ x9 \]$	p3g1f1s1b1B1c2T2t1r1U1R3u9Z2W3D3a2n9l3m3	8
3	$[0 \ 0 \ 0 \ 0 \ 0 \ x6 \ x7 \ x8 \ 0 \]$	p3g1f1s1b1B1c1T2t1r1U1R2u19Z2W3D3a2n9l3m3	16
3	$[0 \ 0 \ 0 \ 0 \ x5 \ x6 \ 0 \ x8 \ -x5 \]$	p3g1f1s1b1B1c1T2t1r1U1R2u10Z2W3D3a2n9l3m3	8
3	$[0 \ 0 \ 0 \ 0 \ x5 \ x6 \ x7 \ -x6 \ x5 \]$	p3g1f1s1b1B1c1T2t1r1U1R3u11Z2W3D3a2n9l3m3	8
3	$[0 \ 0 \ 0 \ x4 \ 0 \ x6 \ 0 \ x8 \ 0 \]$	p3g1f1s1b1B1c1T2t1r1U1R2u11Z2W3D3a2n9l3m3	16
3	$[0 \ 0 \ 0 \ x4 \ 0 \ x6 \ x7 \ -x6 \ 0 \]$	p3g1f1s1b1B1c1T2t1r1U1R3u10Z2W3D3a2n9l3m3	8
3	$[0 \ 0 \ 0 \ x4 \ x5 \ x6 \ 0 \ -x6 \ x5 \]$	p3g1f1s1b1B1c1T2t1r1U1R3u19Z2W3D3a2n9l3m3	8
3	$[0 \ 0 \ x3 \ 0 \ 0 \ 0 \ -x3 \ x8 \ x9 \]$	p3g1f1s1b1B1c2T2t5r1U1R3u9Z2W3D3a2n9l3m3	4
3	$[0 \ 0 \ x3 \ 0 \ 0 \ 0 \ x7 \ 0 \ x9 \]$	p3g1f1s1b1B1c2T2t5r1U1R2u17Z2W3D3a2n9l3m3	104
3	$[0 \ 0 \ x3 \ 0 \ 0 \ 0 \ x7 \ x8 \ 0 \]$	p3g1f1s1b1B1c1T2t5r1U1R2u19Z2W3D3a2n9l3m3	18
3	$[0 \ 0 \ x3 \ 0 \ 0 \ x6 \ -x3 \ -x6 \ x9 \]$	p3g1f1s1b1B1c3T2t3r1U1R1u10Z2W3D3a2n9l3m3	72
3	$[0 \ 0 \ x3 \ 0 \ 0 \ x6 \ -x3 \ x8 \ 0 \]$	p3g1f1s1b1B1c1T2t3r1U1R2u1Z2W3D3a2n9l3m3	4
3	$[0 \ 0 \ x3 \ 0 \ 0 \ x6 \ x7 \ -x6 \ 0 \]$	p3g1f1s1b1B1c1T2t3r1U1R2u17Z2W3D3a2n9l3m3	80
3	$[0 \ x2 \ 0 \ -x2 \ 0 \ x6 \ 0 \ x8 \ 0 \]$	p3g1f1s1b1B1c1T2t11r1U1R2u9Z2W3D3a2n9l3m3	4
3	$[0 \ x2 \ 0 \ x4 \ 0 \ x6 \ 0 \ -x6 \ 0 \]$	p3g1f1s1b1B1c1T2t11r1U1R2u17Z2W3D3a2n9l3m3	80
3	$[0 \ x2 \ 0 \ x4 \ 0 \ x6 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R2u11Z2W3D3a2n9l3m3	18
3	$[0 \ x2 \ x3 \ -x2 \ 0 \ 0 \ x7 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t10r1U1R2u9Z2W3D3a2n9l3m3	4
3	$[0 \ x2 \ x3 \ -x2 \ 0 \ x6 \ -x3 \ -x6 \ 0 \]$	p3g1f1s1b1B1c1T2t12r1U1R1u10Z2W3D3a2n9l3m3	344
3	$[0 \ x2 \ x3 \ -x2 \ 0 \ x6 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R3u19Z2W3D3a2n9l3m3	4
3	$[0 \ x2 \ x3 \ 0 \ 0 \ 0 \ -x3 \ x8 \ 0 \]$	p3g1f1s1b1B1c1T2t5r1U1R3u11Z2W3D3a2n9l3m3	4
3	$[0 \ x2 \ x3 \ 0 \ 0 \ 0 \ x7 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t5r1U1R2u10Z2W3D3a2n9l3m3	18
3	$[0 \ x2 \ x3 \ x4 \ 0 \ 0 \ -x3 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t10r1U1R2u1Z2W3D3a2n9l3m3	4
3	$[0 \ x2 \ x3 \ x4 \ 0 \ 0 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R2u10Z2W3D3a2n9l3m3	18
3	$[x1 \ 0 \ x3 \ 0 \ 0 \ 0 \ -x3 \ x8 \ x1 \]$	p3g1f1s1b1B1c1T2t5r1U1R3u10Z2W3D3a2n9l3m3	4
3	$[x1 \ 0 \ x3 \ 0 \ 0 \ 0 \ x7 \ 0 \ -x1 \]$	p3g1f1s1b1B1c1T2t5r1U1R2u11Z2W3D3a2n9l3m3	4
3	$[x1 \ x2 \ 0 \ -x2 \ x1 \ x6 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R3u10Z2W3D3a2n9l3m3	4
3	$[x1 \ x2 \ 0 \ x4 \ -x1 \ 0 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R2u19Z2W3D3a2n9l3m3	4
3	$[x1 \ x2 \ x3 \ -x2 \ x1 \ 0 \ 0 \ 0 \ 0 \]$	p3g1f1s1b1B1c1T2t9r1U1R3u11Z2W3D3a2n9l3m3	4

nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
3	$[x_1 \ x_2 \ x_3 \ 0 \ 0 \ 0 \ -x_3 \ 0 \ x_1]$	p3g1f1s1b1B1c1T2t5r1U1R3u19Z2W3D3a2n9l3m3	4
		TOTAL de cas pour 3 paramètres	892
4	$[0 \ 0 \ 0 \ 0 \ 0 \ x_6 \ x_7 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t1r1U1R2u19Z2W3D3a2n9l3m3	32
4	$[0 \ 0 \ 0 \ 0 \ x_5 \ x_6 \ 0 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t1r1U1R2u10Z2W3D3a2n9l3m3	148
4	$[0 \ 0 \ 0 \ x_4 \ x_5 \ x_6 \ x_7 \ -x_6 \ x_5]$	p3g1f1s1b1B1c1T2t1r1U1R3u12Z2W3D3a2n9l3m3	12
4	$[0 \ 0 \ x_3 \ 0 \ 0 \ 0 \ x_7 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t5r1U1R2u19Z2W3D3a2n9l3m3	36
4	$[0 \ 0 \ x_3 \ 0 \ 0 \ x_6 \ -x_3 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t3r1U1R2u1Z2W3D3a2n9l3m3	8
4	$[0 \ 0 \ x_3 \ 0 \ 0 \ x_6 \ x_7 \ -x_6 \ x_9]$	p3g1f1s1b1B1c2T2t3r1U1R2u17Z2W3D3a2n9l3m3	160
4	$[0 \ 0 \ x_3 \ 0 \ 0 \ x_6 \ x_7 \ x_8 \ 0]$	p3g1f1s1b1B1c1T2t3r1U1R2u19Z2W3D3a2n9l3m3	52
4	$[0 \ x_2 \ 0 \ x_4 \ 0 \ x_6 \ 0 \ x_8 \ 0]$	p3g1f1s1b1B1c1T2t11r1U1R2u1Z2W3D3a2n9l3m3	52
4	$[0 \ x_2 \ x_3 \ -x_2 \ x_5 \ x_6 \ 0 \ -x_6 \ x_5]$	p3g1f1s1b1B1c1T2t11r1U1R3u17Z2W3D3a2n9l3m3	4
4	$[0 \ x_2 \ x_3 \ 0 \ 0 \ 0 \ x_7 \ x_8 \ 0]$	p3g1f1s3b1B1c1T2t5r1U1R3u1Z2W3D3a2n9l3m3	4
4	$[0 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ -x_3 \ -x_6 \ x_5]$	p3g1f1s1b1B1c1T2t3r1U1R3u17Z2W3D3a2n9l3m3	4
4	$[0 \ x_2 \ x_3 \ x_4 \ 0 \ 0 \ x_7 \ 0 \ 0]$	p3g1f1s1b1B1c1T2t10r1U1R2u10Z2W3D3a2n9l3m3	128
4	$[0 \ x_2 \ x_3 \ x_4 \ 0 \ x_6 \ 0 \ 0 \ 0]$	p3g1f1s3b1B1c1T2t9r1U1R3u19Z2W3D3a2n9l3m3	4
4	$[x_1 \ 0 \ x_3 \ 0 \ 0 \ 0 \ x_7 \ 0 \ x_9]$	p3g1f1s1b1B1c2T2t5r1U1R2u1Z2W3D3a2n9l3m3	185
4	$[x_1 \ 0 \ x_3 \ x_4 \ 0 \ x_6 \ -x_3 \ -x_6 \ x_1]$	p3g1f1s1b1B1c1T2t3r1U1R3u1Z2W3D3a2n9l3m3	4
4	$[x_1 \ x_2 \ 0 \ -x_2 \ x_1 \ x_6 \ x_7 \ -x_6 \ 0]$	p3g1f1s1b1B1c1T2t11r1U1R3u9Z2W3D3a2n9l3m3	4
4	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ 0]$	p3g1f1s3b1B1c1T2t10r1U1R3u1Z1W3D3a2n9l3m3	37
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ 0 \ x_6 \ -x_3 \ 0 \ x_1]$	p3g1f1s1b1B1c1T2t10r1U1R3u1Z2W3D3a2n9l3m3	4
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ 0 \ -x_3 \ x_8 \ 0]$	p3g1f1s1b1B1c1T2t10r1U1R3u9Z2W3D3a2n9l3m3	4
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ 0]$	p3g1f1s1b1B1c1T2t9r1U1R3u12Z2W3D3a2n9l3m3	6
4	$[x_1 \ x_2 \ x_3 \ 0 \ 0 \ 0 \ -x_3 \ x_8 \ x_1]$	p3g1f1s1b1B1c1T2t5r1U1R3u12Z2W3D3a2n9l3m3	6
		TOTAL de cas pour 4 paramètres	894
5	$[0 \ 0 \ 0 \ 0 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t1r1U1R3u1Z2W3D3a2n9l3m3	16
5	$[0 \ 0 \ 0 \ x_4 \ 0 \ x_6 \ x_7 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t1r1U1R2u1Z2W3D3a2n9l3m3	56
5	$[0 \ 0 \ 0 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ -x_5]$	p3g1f1s1b1B1c1T2t1r1U1R2u12Z2W3D3a2n9l3m3	28
5	$[0 \ 0 \ 0 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_5]$	p3g1f1s1b1B1c1T2t1r1U1R4u1Z2W3D3a2n9l3m3	32
5	$[0 \ 0 \ x_3 \ 0 \ 0 \ x_6 \ x_7 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t3r1U1R2u19Z2W3D3a2n9l3m3	104
5	$[0 \ x_2 \ x_3 \ -x_2 \ 0 \ x_6 \ -x_3 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t11r1U1R2u9Z2W3D3a2n9l3m3	6
5	$[0 \ x_2 \ x_3 \ -x_2 \ 0 \ x_6 \ x_7 \ -x_6 \ x_9]$	p3g1f1s1b1B1c2T2t10r1U1R2u9Z2W3D3a2n9l3m3	6
5	$[0 \ x_2 \ x_3 \ -x_2 \ 0 \ x_6 \ x_7 \ x_8 \ 0]$	p3g1f1s1b1B1c1T2t12r1U1R2u9Z2W3D3a2n9l3m3	6
5	$[0 \ x_2 \ x_3 \ 0 \ 0 \ 0 \ x_7 \ x_8 \ x_9]$	p3g1f1s1b1B1c2T2t5r1U1R2u10Z2W3D3a2n9l3m3	56
5	$[0 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ x_7 \ -x_6 \ x_5]$	p3g1f1s1b1B1c1T2t3r1U1R4u17Z2W3D3a2n9l3m3	20
5	$[0 \ x_2 \ x_3 \ x_4 \ 0 \ x_6 \ -x_3 \ x_8 \ 0]$	p3g1f1s1b1B1c1T2t12r1U1R2u1Z2W3D3a2n9l3m3	6
<i>suite page suivante</i>			

nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
5	[0 x2 x3 x4 0 x6 x7 -x6 0]	p3g1f1s1b1B1c1T2t12r1U1R2u17Z2W3D3a2n9l3m3	90
5	[0 x2 x3 x4 x5 x6 0 -x6 x5]	p3g1f1s1b1B1c1T2t11r1U1R4u17Z2W3D3a2n9l3m3	20
5	[x1 0 x3 0 0 0 x7 x8 x9]	p3g1f1s1b1B1c2T2t5r1U1R3u10Z2W3D3a2n9l3m3	20
5	[x1 0 x3 x4 0 x6 -x3 x8 x1]	p3g1f1s1b1B1c1T2t3r1U1R4u1Z2W3D3a2n9l3m3	8
5	[x1 x2 0 -x2 x1 x6 x7 x8 0]	p3g1f1s1b1B1c1T2t11r1U1R4u9Z2W3D3a2n9l3m3	8
5	[x1 x2 0 x4 x5 x6 0 0 0]	p3g1f1s3b1B1c1T2t9r1U1R3u10Z2W3D3a2n9l3m3	4
5	[x1 x2 x3 -x2 x1 0 x7 x8 0]	p3g1f1s1b1B1c1T2t10r1U1R4u9Z2W3D3a2n9l3m3	8
5	[x1 x2 x3 0 0 0 x7 0 x9]	p3g1f1s3b1B1c1T2t5r1U1R3u19Z2W3D3a2n9l3m3	4
5	[x1 x2 x3 0 0 0 x7 x8 -x1]	p3g1f1s1b1B1c1T2t5r1U1R2u12Z2W3D3a2n9l3m3	14
5	[x1 x2 x3 0 0 0 x7 x8 x1]	p3g1f1s1b1B1c1T2t5r1U1R4u10Z2W3D3a2n9l3m3	16
5	[x1 x2 x3 x4 -x1 x6 0 0 0]	p3g1f1s1b1B1c1T2t9r1U1R2u12Z2W3D3a2n9l3m3	14
5	[x1 x2 x3 x4 0 x6 -x3 0 x1]	p3g1f1s1b1B1c1T2t10r1U1R4u1Z2W3D3a2n9l3m3	8
5	[x1 x2 x3 x4 x1 x6 0 0 0]	p3g1f1s1b1B1c1T2t9r1U1R4u10Z2W3D3a2n9l3m3	16
5	[x1 x2 x3 x4 x5 0 0 0 0]	p3g1f1s3b1B1c1T2t9r1U1R3u11Z2W3D3a2n9l3m3	4
		TOTAL de cas pour 5 paramètres	570
6	[0 0 0 x4 x5 x6 x7 x8 x9]	p3g1f1s1b1B1c1T2t1r1U1R4u12Z2W3D3a2n9l3m3	232
6	[0 x2 x3 -x2 0 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t12r1U1R2u9Z2W3D3a2n9l3m3	298
6	[0 x2 x3 0 x5 x6 x7 x8 -x5]	p3g1f1s1b1B1c1T2t3r1U1R2u10Z2W3D3a2n9l3m3	16
6	[0 x2 x3 0 x5 x6 x7 x8 x5]	p3g1f1s1b1B1c1T2t3r1U1R3u11Z2W3D3a2n9l3m3	16
6	[0 x2 x3 x4 0 x6 x7 x8 0]	p3g1f1s3b1B1c1T2t12r1U1R2u1Z2W3D3a2n9l3m3	12
6	[0 x2 x3 x4 x5 x6 0 x8 -x5]	p3g1f1s1b1B1c1T2t11r1U1R2u10Z2W3D3a2n9l3m3	16
6	[0 x2 x3 x4 x5 x6 0 x8 x5]	p3g1f1s1b1B1c1T2t11r1U1R3u9Z2W3D3a2n9l3m3	16
6	[0 x2 x3 x4 x5 x6 x7 -x6 x5]	p3g1f1s1b1B1c1T2t12r1U1R3u17Z2W3D3a2n9l3m3	36
6	[x1 0 x3 x4 0 x6 x7 -x6 x9]	p3g1f1s3b1B1c1T2t3r1U1R3u1Z2W3D3a2n9l3m3	4
6	[x1 0 x3 x4 0 x6 x7 x8 -x1]	p3g1f1s1b1B1c1T2t3r1U1R2u11Z2W3D3a2n9l3m3	8
6	[x1 0 x3 x4 0 x6 x7 x8 x1]	p3g1f1s1b1B1c1T2t3r1U1R3u10Z2W3D3a2n9l3m3	8
6	[x1 x2 0 x4 -x1 x6 x7 x8 0]	p3g1f1s1b1B1c1T2t11r1U1R2u19Z2W3D3a2n9l3m3	8
6	[x1 x2 0 x4 x1 x6 x7 x8 0]	p3g1f1s1b1B1c1T2t11r1U1R3u10Z2W3D3a2n9l3m3	8
6	[x1 x2 0 x4 x5 x6 x7 -x6 0]	p3g1f1s3b1B1c1T2t11r1U1R3u9Z2W3D3a2n9l3m3	4
6	[x1 x2 x3 -x2 x1 x6 x7 x8 0]	p3g1f1s1b1B1c1T2t12r1U1R3u9Z2W3D3a2n9l3m3	18
6	[x1 x2 x3 -x2 x5 x6 -x3 x6 x9]	p3g1f1s1b1B1c1T2t10r1U1R3u10Z1W3D3a2n9l3m3	36
6	[x1 x2 x3 -x2 x5 x6 x3 -x6 x9]	p3g1f1s1b1B1c1T2t11r1U1R3u11Z1W3D3a2n9l3m3	36
6	[x1 x2 x3 0 0 0 x7 x8 x9]	p3g1f1s1b1B1c1T2t5r1U1R4u12Z2W3D3a2n9l3m3	268
6	[x1 x2 x3 x2 x5 x6 -x3 -x6 x9]	p3g1f1s1b1B1c1T2t3r1U1R3u19Z1W3D3a2n9l3m3	36
6	[x1 x2 x3 x4 -x1 0 x7 x8 0]	p3g1f1s1b1B1c1T2t10r1U1R2u19Z2W3D3a2n9l3m3	8

<i>suite de la page précédente</i>			
nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
6	[x1 x2 x3 x4 0 x6 -x3 x8 x1]	p3g1f1s1b1B1c1T2t12r1U1R3u1Z2W3D3a2n9l3m3	18
6	[x1 x2 x3 x4 0 x6 x7 0 -x1]	p3g1f1s1b1B1c1T2t10r1U1R2u11Z2W3D3a2n9l3m3	8
6	[x1 x2 x3 x4 0 x6 x7 0 x1]	p3g1f1s1b1B1c1T2t10r1U1R3u19Z2W3D3a2n9l3m3	8
6	[x1 x2 x3 x4 x1 0 x7 x8 0]	p3g1f1s1b1B1c1T2t10r1U1R3u11Z2W3D3a2n9l3m3	8
6	[x1 x2 x3 x4 x5 x6 0 0 0]	p3g1f1s1b1B1c1T2t9r1U1R4u12Z2W3D3a2n9l3m3	60
		TOTAL de cas pour 6 paramètres	1186
7	[0 x2 x3 -x2 x5 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t11r1U1R3u17Z2W3D3a2n9l3m3	8
7	[0 x2 x3 0 x5 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t3r1U1R2u10Z2W3D3a2n9l3m3	112
7	[0 x2 x3 x4 0 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R2u10Z2W3D3a2n9l3m3	928
7	[0 x2 x3 x4 x5 x6 x7 x8 -x5]	p3g1f1s1b1B1c1T2t12r1U1R2u10Z2W3D3a2n9l3m3	24
7	[x1 0 x3 x4 0 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t3r1U1R2u11Z2W3D3a2n9l3m3	136
7	[x1 x2 0 x4 x5 x6 x7 x8 0]	p3g1f1s3b1B1c1T2t11r1U1R2u19Z2W3D3a2n9l3m3	24
7	[x1 x2 x3 -x2 0 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R3u1Z2W3D3a2n9l3m3	8
7	[x1 x2 x3 -x2 x1 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R3u11Z1W3D3a2n9l3m3	178
7	[x1 x2 x3 -x2 x5 x6 -x3 x8 x9]	p3g1f1s1b1B1c1T2t10r1U1R3u10Z2W3D3a2n9l3m3	8
7	[x1 x2 x3 -x2 x5 x6 x7 -x6 x9]	p3g1f1s1b1B1c1T2t11r1U1R3u11Z2W3D3a2n9l3m3	8
7	[x1 x2 x3 x4 -x1 x6 x7 x8 0]	p3g1f1s1b1B1c1T2t12r1U1R2u19Z2W3D3a2n9l3m3	12
7	[x1 x2 x3 x4 0 x6 x7 0 x9]	p3g1f1s3b1B1c1T2t10r1U1R2u11Z2W3D3a2n9l3m3	28
7	[x1 x2 x3 x4 0 x6 x7 x8 -x1]	p3g1f1s1b1B1c1T2t12r1U1R2u11Z2W3D3a2n9l3m3	12
7	[x1 x2 x3 x4 x5 0 x7 x8 0]	p3g1f1s3b1B1c1T2t10r1U1R2u19Z2W3D3a2n9l3m3	28
7	[x1 x2 x3 x4 x5 x6 -x3 -x6 x9]	p3g1f1s1b1B1c1T2t3r1U1R3u19Z2W3D3a2n9l3m3	8
		TOTAL de cas pour 7 paramètres	1522
8	[0 x2 x3 x4 x5 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t11r1U1R2u10Z2W3D3a2n9l3m3	224
8	[x1 x2 x3 -x2 x5 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R3u10Z1W3D3a2n9l3m3	176
8	[x1 x2 x3 x2 x5 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t3r1U1R3u19Z1W3D3a2n9l3m3	72
8	[x1 x2 x3 x4 -x1 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R2u19Z2W3D3a2n9l3m3	92
8	[x1 x2 x3 x4 0 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R2u11Z2W3D3a2n9l3m3	254
8	[x1 x2 x3 x4 x1 x6 x7 x8 x9]	p3g1f1s1b1B1c2T2t10r1U1R3u11Z2W3D3a2n9l3m3	64
8	[x1 x2 x3 x4 x5 x6 x7 -x6 x9]	p3g1f1s3b1B1c1T2t11r1U1R3u11Z1W3D3a2n9l3m3	88
8	[x1 x2 x3 x4 x5 x6 x7 x6 x9]	p3g1f1s3b1B1c1T2t10r1U1R3u10Z1W3D3a2n9l3m3	36
8	[x1 x2 x3 x4 x5 x6 x7 x8 0]	p3g1f1s3b1B1c1T2t12r1U1R2u19Z2W3D3a2n9l3m3	30
		TOTAL de cas pour 8 paramètres	1036

En faisant le même type d'expérience, dans le cas de la projection orthographique, sur les cas
p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n9l3m3

et

p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n9l3m3

parmi ces 37440 cas, 4968 correspondent à des cas de matrice fondamentale nulle, et 14098 à des cas de matrices fondamentales plus simples que la matrice affine de type :

$$\begin{pmatrix} 0 & 0 & x3 \\ 0 & 0 & x6 \\ x7 & x8 & x9 \end{pmatrix}$$

Les différentes formes simplifiées de matrices sont répertoriées dans le tableau suivant :

nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
1	[0 0 0 0 0 x6 0 -x6 -x6]	p1g1f1s1b1B1c1T2t5r1U1R3u1Z1W3D3a2n9l3m3	612
1	[0 0 0 0 0 x6 0 -x6 0]	p1g1f1s1b1B1c1T1t13r1U1R3u1Z1W3D3a2n9l3m3	396
1	[0 0 x3 0 0 0 -x3 0 -x3]	p1g1f1s1b1B1c1T2t1r1U1R3u17Z1W3D3a2n9l3m3	102
1	[0 0 x3 0 0 0 -x3 0 0]	p1g1f1s1b1B1c1T1t13r1U1R3u17Z1W3D3a2n9l3m3	66
		TOTAL de cas pour 1 paramètre	1176
2	[0 0 0 0 0 x6 0 -x6 x9]	p1g1f1s1b1B1c1T2t10r1U1R3u1Z2W3D3a2n9l3m3	936
2	[0 0 0 0 0 x6 0 x8 -x6]	p1g1f1s1b1B1c1T2t5r1U1R2u18Z2W3D3a2n9l3m3	48
2	[0 0 0 0 0 x6 0 x8 0]	p1g1f1s1b1B1c1T1t13r1U1R2u18Z1W3D3a2n9l3m3	402
2	[0 0 0 0 0 x6 x7 -x6 -x6]	p1g1f1s1b1B1c1T2t5r1U1R3u3Z2W3D3a2n9l3m3	22
2	[0 0 0 0 0 x6 x7 -x6 0]	p1g1f1s1b1B1c1T1t13r1U1R3u3Z1W3D3a2n9l3m3	22
2	[0 0 x3 0 0 0 -x3 0 x9]	p1g1f1s1b1B1c1T2t11r1U1R3u17Z2W3D3a2n9l3m3	156
2	[0 0 x3 0 0 0 -x3 x8 -x3]	p1g1f1s1b1B1c1T2t1r1U1R3u18Z2W3D3a2n9l3m3	11
2	[0 0 x3 0 0 0 -x3 x8 0]	p1g1f1s1b1B1c1T1t13r1U1R3u18Z1W3D3a2n9l3m3	11
2	[0 0 x3 0 0 0 x7 0 -x3]	p1g1f1s1b1B1c1T2t1r1U1R2u3Z2W3D3a2n9l3m3	8
2	[0 0 x3 0 0 0 x7 0 0]	p1g1f1s1b1B1c1T1t13r1U1R2u3Z1W3D3a2n9l3m3	167
2	[0 0 x3 0 0 x6 -x3 -x6 -x3]	p1g1f1s1b1B1c1T2t1r1U1R3u19Z2W3D3a2n9l3m3	198
2	[0 0 x3 0 0 x6 -x3 -x6 -x6]	p1g1f1s1b1B1c1T2t5r1U1R3u19Z2W3D3a2n9l3m3	48
2	[0 0 x3 0 0 x6 -x3 -x6 0]	p1g1f1s1b1B1c1T1t13r1U1R3u19Z1W3D3a2n9l3m3	186
		TOTAL de cas pour 2 paramètres	2215
3	[0 0 0 0 0 x6 0 x8 x9]	p1g1f1s1b1B1c1T2t10r1U1R2u18Z2W3D3a2n9l3m3	2682
3	[0 0 0 0 0 x6 x7 -x6 x9]	p1g1f1s1b1B1c1T2t10r1U1R3u3Z1W3D3a2n9l3m3	154
3	[0 0 0 0 0 x6 x7 x8 -x6]	p1g2f1s1b1B1c1T2t5r1U1R3u3Z2W3D3a2n9l3m3	44
3	[0 0 0 0 0 x6 x7 x8 0]	p1g1f3s1b1B1c1T1t13r1U1R3u3Z1W3D3a2n9l3m3	110
3	[0 0 x3 0 0 0 -x3 x8 x9]	p1g1f1s1b1B1c1T2t11r1U1R3u18Z1W3D3a2n9l3m3	77
3	[0 0 x3 0 0 0 x7 0 x9]	p1g1f1s1b1B1c1T2t11r1U1R2u3Z2W3D3a2n9l3m3	1193
3	[0 0 x3 0 0 0 x7 x8 0]	p1g1f1s3b1B1c1T1t13r1U1R3u18Z1W3D3a2n9l3m3	33

suite page suivante

<i>suite de la page précédente</i>			
nombre de paramètres	forme simplifiée de matrice fondamentale	exemple de cas pour ce type de matrice	nombre de cas
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ -x_3\ -x_6\ x_9]$	p1g1f1s1b1B1c1T2t10r1U1R3u19Z2W3D3a2n9l3m3	1080
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ -x_3\ x_8\ -x_3]$	p1g1f1s1b1B1c1T2t1r1U1R4u18Z2W3D3a2n9l3m3	139
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ -x_3\ x_8\ -x_6]$	p1g1f1s1b1B1c1T2t5r1U1R4u18Z2W3D3a2n9l3m3	8
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ -x_3\ x_8\ 0]$	p1g1f1s1b1B1c1T1t13r1U1R4u18Z1W3D3a2n9l3m3	115
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ -x_6\ -x_3]$	p1g1f1s1b1B1c1T2t1r1U1R4u3Z2W3D3a2n9l3m3	8
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ -x_6\ -x_6]$	p1g1f1s1b1B1c1T2t5r1U1R4u3Z2W3D3a2n9l3m3	52
3	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ -x_6\ 0]$	p1g1f1s1b1B1c1T1t13r1U1R4u3Z1W3D3a2n9l3m3	50
		TOTAL de cas pour 3 paramètres	5745
4	$[0\ 0\ 0\ 0\ 0\ x_6\ x_7\ x_8\ x_9]$	p1g1f3s1b1B1c1T2t10r1U1R3u3Z1W3D3a2n9l3m3	836
4	$[0\ 0\ x_3\ 0\ 0\ 0\ x_7\ x_8\ x_9]$	p1g1f1s3b1B1c1T2t11r1U1R3u18Z1W3D3a2n9l3m3	264
4	$[0\ 0\ x_3\ 0\ 0\ x_6\ -x_3\ x_8\ x_9]$	p1g1f1s1b1B1c1T2t10r1U1R4u18Z2W3D3a2n9l3m3	1448
4	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ -x_6\ x_9]$	p1g1f1s1b1B1c1T2t10r1U1R4u3Z1W3D3a2n9l3m3	880
4	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ x_8\ -x_3]$	p1g1f1s1b1B1c1T2t1r1U1R2u20Z2W3D3a2n9l3m3	128
4	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ x_8\ -x_6]$	p1g1f1s1b1B1c1T2t5r1U1R2u20Z2W3D3a2n9l3m3	72
4	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ x_8\ 0]$	p1g1f1s1b1B1c1T1t13r1U1R2u20Z1W3D3a2n9l3m3	1334
		TOTAL de cas pour 4 paramètres	4962
5	$[0\ 0\ x_3\ 0\ 0\ x_6\ x_7\ x_8\ x_9]$	p1g1f1s1b1B1c1T2t10r1U1R2u20Z2W3D3a2n9l3m3	18374
		TOTAL de cas pour 5 paramètres	18374

2.8.2 Les formes particulières de matrices d'homographie.

2.8.2.1 Les cas de mouvements induisant des homographies.

Dans le cas de la projection perspective, on est en présence d'une relation homographique entre les points dans les images dans le cas de la rotation pure. Nous allons donc étudier les 4320 cas suivants :

p3g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]R[1-5]r1a[0;2]u[1-4;9-12;17-20]W3T1t13D3Z3d3n9l3m3

3219 cas présentent des formes simplifiées de matrice d'homographie que l'on présente dans le tableau suivant, où le nombre de degrés de liberté de chaque modèle est exactement le nombre de paramètres indiqués moins un, la matrice d'homographie étant définie à un facteur d'échelle près.

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
<i>suite page suivante</i>			

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
1	[x1 0 0 0 -x1 0 0 0 -x1]	p3g1f1s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	2
1	[x1 0 0 0 -x1 0 0 0 x1]	p3g1f1s1b1B1c1T1t13r1U1R2u1Z3W3D3a1n9l3m3	2
1	[x1 0 0 0 x1 0 0 0 -x1]	p3g1f1s1b1B1c1T1t13r1U1R2u9Z3W3D3a1n9l3m3	4
1	[x1 0 0 0 x1 0 0 0 x1]	p3g1f1s1b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	240
		TOTAL de cas pour 1 paramètre	248
2	[x1 0 0 0 -x1 0 0 0 x9]	p3g1f3s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	4
2	[x1 0 0 0 -x1 x6 0 0 x1]	p3g1f1s1b1B1c2T1t13r1U1R2u1Z3W3D3a1n9l3m3	2
2	[x1 0 0 0 x1 0 0 0 x9]	p3g1f3s1b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	62
2	[x1 0 0 0 x1 x6 0 -x6 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u17Z3W3D3a4n9l3m3	1
2	[x1 0 0 0 x5 0 0 0 -x5]	p3g1f1s3b1B1c1T1t13r1U1R2u1Z3W3D3a1n9l3m3	4
2	[x1 0 0 0 x5 0 0 0 x5]	p3g1f1s3b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	62
2	[x1 0 x3 0 -x1 0 0 0 -x1]	p3g1f1s1b1B1c2T1t13r1U1R2u17Z3W3D3a1n9l3m3	2
2	[x1 0 x3 0 x1 0 -x3 0 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u1Z3W3D3a4n9l3m3	1
2	[x1 x2 0 -x2 x1 0 0 0 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u9Z3W3D3a4n9l3m3	1
2	[x1 x2 0 0 -x1 0 0 0 -x1]	p3g2f1s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	4
2	[x1 x2 0 0 -x1 0 0 0 x1]	p3g2f1s1b1B1c1T1t13r1U1R2u1Z3W3D3a1n9l3m3	4
2	[x1 x2 0 0 x1 0 0 0 -x1]	p3g4f1s1b1B1c1T1t13r1U1R2u9Z3W3D3a1n9l3m3	2
2	[x1 x2 0 0 x1 0 0 0 x1]	p3g4f1s1b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	60
		TOTAL de cas pour 2 paramètres	209
3	[x1 0 0 0 x5 0 0 0 x9]	p3g1f3s3b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	66
3	[x1 0 0 0 x5 x6 0 -x6 x5]	p3g1f1s1b1B1c1T1t13r1U1R4u17Z3W3D3a4n9l3m3	5
3	[x1 0 0 0 x5 x6 0 x6 -x5]	p3g1f1s1b1B1c1T1t13r1U1R2u10Z3W3D3a1n9l3m3	8
3	[x1 0 x3 0 -x1 x6 0 0 -x1]	p3g1f1s1b1B1c3T1t13r1U1R2u17Z3W3D3a1n9l3m3	2
3	[x1 0 x3 0 -x1 x6 0 0 x1]	p3g1f1s1b1B1c3T1t13r1U1R2u1Z3W3D3a1n9l3m3	2
3	[x1 0 x3 0 x1 0 -x3 x8 x1]	p3g2f1s1b1B1c1T1t13r1U1R3u1Z3W3D3a4n9l3m3	1
3	[x1 0 x3 0 x1 x6 -x3 -x6 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u9Z3W3D3a4n9l3m3	2
3	[x1 0 x3 0 x1 x6 0 -x6 x1]	p3g2f1s1b1B1c1T1t13r1U1R3u17Z3W3D3a4n9l3m3	1
3	[x1 0 x3 0 x1 x6 0 0 -x1]	p3g1f1s1b1B1c2T1t13r1U1R2u9Z3W3D3a1n9l3m3	8
3	[x1 0 x3 0 x1 x6 0 0 x1]	p3g1f1s1b1B1c3T1t13r1U1R1u10Z3W3D3a1n9l3m3	120
3	[x1 0 x3 0 x5 0 -x3 0 x1]	p3g1f1s1b1B1c1T1t13r1U1R4u1Z3W3D3a4n9l3m3	2
3	[x1 0 x3 0 x5 0 0 0 x5]	p3g1f1s3b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	62
3	[x1 0 x3 0 x5 0 x3 0 -x1]	p3g1f1s1b1B1c1T1t13r1U1R2u11Z3W3D3a1n9l3m3	4
3	[x1 x2 0 -x2 x1 0 0 0 x9]	p3g1f1s1b1B1c1T1t13r1U1R4u9Z3W3D3a4n9l3m3	5
3	[x1 x2 0 -x2 x1 x6 0 -x6 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u11Z3W3D3a4n9l3m3	2
3	[x1 x2 0 0 -x1 0 0 0 x9]	p3g2f3s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	8

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
3	[x1 x2 0 0 x1 0 0 0 x9]	p3g2f3s1b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	124
3	[x1 x2 0 0 x5 0 0 0 -x5]	p3g2f1s3b1B1c1T1t13r1U1R2u1Z3W3D3a1n9l3m3	8
3	[x1 x2 0 0 x5 0 0 0 x5]	p3g2f1s3b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	124
3	[x1 x2 0 x2 -x1 0 0 0 x9]	p3g1f1s1b1B1c1T1t13r1U1R2u19Z3W3D3a1n9l3m3	8
3	[x1 x2 x3 -x2 x1 0 -x3 0 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u10Z3W3D3a4n9l3m3	2
3	[x1 x2 x3 0 -x1 0 0 0 -x1]	p3g2f1s1b1B1c2T1t13r1U1R2u17Z3W3D3a1n9l3m3	4
3	[x1 x2 x3 0 x1 0 0 0 x1]	p3g4f1s1b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	60
		TOTAL de cas pour 3 paramètres	628
4	[x1 0 0 0 x1 x6 0 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R3u17Z3W3D3a4n9l3m3	1
4	[x1 0 x3 0 -x1 x6 0 0 x9]	p3g1f3s1b1B1c2T1t13r1U1R2u17Z3W3D3a1n9l3m3	8
4	[x1 0 x3 0 x1 0 x7 0 x9]	p3g1f3s1b1B1c1T1t13r1U1R3u1Z3W3D3a4n9l3m3	1
4	[x1 0 x3 0 x1 x6 0 0 x9]	p3g1f3s1b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	125
4	[x1 0 x3 0 x5 x6 0 0 -x5]	p3g1f1s3b1B1c2T1t13r1U1R2u1Z3W3D3a1n9l3m3	8
4	[x1 0 x3 0 x5 x6 0 0 x5]	p3g1f1s3b1B1c3T1t13r1U1R1u10Z3W3D3a1n9l3m3	62
4	[x1 x2 0 0 x5 0 0 0 x9]	p3g2f3s3b1B1c1T1t13r1U1R1u10Z3W3D3a1n9l3m3	132
4	[x1 x2 0 x4 -x1 0 0 0 x9]	p3g2f1s1b1B1c1T1t13r1U1R2u19Z3W3D3a1n9l3m3	4
4	[x1 x2 0 x4 x5 0 0 0 x5]	p3g1f1s3b1B1c1T1t13r1U1R3u9Z3W3D3a4n9l3m3	1
4	[x1 x2 x3 -x2 x1 x6 -x3 -x6 x1]	p3g1f1s1b1B1c1T1t13r1U1R3u12Z3W3D3a4n9l3m3	3
4	[x1 x2 x3 -x2 x1 x6 0 0 x1]	p3g1f1s1b1B1c2T1t13r1U1R3u9Z3W3D3a4n9l3m3	2
4	[x1 x2 x3 0 -x1 x6 0 0 -x1]	p3g2f1s1b1B1c3T1t13r1U1R2u17Z3W3D3a1n9l3m3	4
4	[x1 x2 x3 0 -x1 x6 0 0 x1]	p3g2f1s1b1B1c2T1t13r1U1R2u1Z3W3D3a1n9l3m3	8
4	[x1 x2 x3 0 x1 0 -x3 x8 x1]	p3g4f1s1b1B1c1T1t13r1U1R3u1Z3W3D3a4n9l3m3	1
4	[x1 x2 x3 0 x1 x6 0 -x6 x1]	p3g4f1s1b1B1c1T1t13r1U1R3u17Z3W3D3a4n9l3m3	1
4	[x1 x2 x3 0 x1 x6 0 0 -x1]	p3g4f1s1b1B1c2T1t13r1U1R2u9Z3W3D3a1n9l3m3	4
4	[x1 x2 x3 0 x1 x6 0 0 x1]	p3g4f1s1b1B1c3T1t13r1U1R1u10Z3W3D3a1n9l3m3	60
4	[x1 x2 x3 0 x5 0 0 0 x5]	p3g2f1s3b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	124
		TOTAL de cas pour 4 paramètres	549
5	[x1 0 0 0 x5 x6 0 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R2u10Z3W3D3a1n9l3m3	13
5	[x1 0 x3 0 x1 x6 x7 x8 x1]	p3g2f1s1b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	2
5	[x1 0 x3 0 x5 0 x7 0 x9]	p3g1f1s3b1B1c1T1t13r1U1R2u11Z3W3D3a1n9l3m3	19
5	[x1 0 x3 0 x5 x6 0 0 x9]	p3g1f3s3b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	132
5	[x1 0 x3 0 x5 x6 x7 -x6 x5]	p3g1f1s3b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	2
5	[x1 x2 0 -x2 x1 x6 0 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R3u11Z3W3D3a4n9l3m3	2
5	[x1 x2 0 x4 x5 0 0 0 x9]	p3g1f1s3b1B1c1T1t13r1U1R2u19Z3W3D3a1n9l3m3	65
5	[x1 x2 0 x4 x5 x6 0 -x6 x5]	p3g1f1s3b1B1c1T1t13r1U1R3u11Z3W3D3a4n9l3m3	2

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
5	[x1 x2 x3 -x2 x1 0 x7 0 x9]	p3g1f3s1b1B1c1T1t13r1U1R3u10Z3W3D3a4n9l3m3	2
5	[x1 x2 x3 -x2 x1 x6 0 0 x9]	p3g1f1s1b1B1c2T1t13r1U1R4u9Z3W3D3a4n9l3m3	10
5	[x1 x2 x3 0 -x1 x6 0 0 x9]	p3g2f3s1b1B1c2T1t13r1U1R2u17Z3W3D3a1n9l3m3	16
5	[x1 x2 x3 0 x1 x6 0 0 x9]	p3g2f3s1b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	248
5	[x1 x2 x3 0 x5 0 -x3 x8 x1]	p3g2f1s1b1B1c1T1t13r1U1R4u1Z3W3D3a4n9l3m3	4
5	[x1 x2 x3 0 x5 0 x3 x8 -x1]	p3g2f1s1b1B1c1T1t13r1U1R2u11Z3W3D3a1n9l3m3	8
5	[x1 x2 x3 0 x5 x6 0 -x6 x5]	p3g2f1s1b1B1c1T1t13r1U1R4u17Z3W3D3a4n9l3m3	10
5	[x1 x2 x3 0 x5 x6 0 0 -x5]	p3g2f1s3b1B1c2T1t13r1U1R2u1Z3W3D3a1n9l3m3	16
5	[x1 x2 x3 0 x5 x6 0 0 x5]	p3g2f1s3b1B1c3T1t13r1U1R1u10Z3W3D3a1n9l3m3	124
5	[x1 x2 x3 0 x5 x6 0 x6 -x5]	p3g2f1s1b1B1c1T1t13r1U1R2u10Z3W3D3a1n9l3m3	16
5	[x1 x2 x3 x2 -x1 x6 0 0 x9]	p3g1f1s1b1B1c2T1t13r1U1R2u19Z3W3D3a1n9l3m3	16
		TOTAL de cas pour 5 paramètres	707
6	[x1 0 x3 0 x1 x6 x7 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	2
6	[x1 0 x3 x4 x5 x6 x7 0 -x1]	p3g1f1s1b1B1c2T1t13r1U1R2u11Z3W3D3a1n9l3m3	4
6	[x1 x2 x3 -x2 x5 x6 -x3 x6 x9]	p3g1f1s1b1B1c1T1t13r1U1R4u10Z3W3D3a4n9l3m3	4
6	[x1 x2 x3 -x2 x5 x6 x3 -x6 x9]	p3g1f1s1b1B1c1T1t13r1U1R4u11Z3W3D3a4n9l3m3	4
6	[x1 x2 x3 0 x1 0 x7 x8 x9]	p3g2f3s1b1B1c1T1t13r1U1R3u1Z3W3D3a4n9l3m3	4
6	[x1 x2 x3 0 x1 x6 x7 x8 x1]	p3g4f1s1b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	2
6	[x1 x2 x3 0 x5 0 x7 x8 x5]	p3g2f1s3b1B1c1T1t13r1U1R3u1Z3W3D3a4n9l3m3	2
6	[x1 x2 x3 0 x5 x6 0 0 x9]	p3g2f3s3b1B1c2T1t13r1U1R1u10Z3W3D3a1n9l3m3	264
6	[x1 x2 x3 0 x5 x6 0 x8 -x5]	p3g1f1s1b1B1c2T1t13r1U1R2u10Z3W3D3a1n9l3m3	24
6	[x1 x2 x3 x2 x5 x6 -x3 -x6 x9]	p3g1f1s1b1B1c1T1t13r1U1R4u19Z3W3D3a4n9l3m3	4
6	[x1 x2 x3 x2 x5 x6 x3 x6 x9]	p3g1f1s1b1B1c1T1t13r1U1R2u12Z3W3D3a1n9l3m3	6
6	[x1 x2 x3 x4 -x1 x6 0 0 x9]	p3g2f1s1b1B1c2T1t13r1U1R2u19Z3W3D3a1n9l3m3	8
6	[x1 x2 x3 x4 x5 0 x7 0 x5]	p3g1f1s3b1B1c1T1t13r1U1R3u10Z3W3D3a4n9l3m3	4
		TOTAL de cas pour 6 paramètres	332
7	[x1 0 x3 0 x5 x6 x7 x8 x9]	p3g1f3s3b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	2
7	[x1 0 x3 x4 x5 x6 x7 0 x9]	p3g1f1s1b1B1c2T1t13r1U1R3u1Z3W3D3a4n9l3m3	52
7	[x1 x2 0 x4 x5 x6 0 x8 x9]	p3g1f3s3b1B1c1T1t13r1U1R3u11Z3W3D3a4n9l3m3	2
7	[x1 x2 x3 -x2 x1 x6 x7 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R3u12Z3W3D3a4n9l3m3	3
7	[x1 x2 x3 0 x1 x6 x7 x8 x9]	p3g2f3s1b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	4
7	[x1 x2 x3 0 x5 0 x7 x8 x9]	p3g2f1s3b1B1c1T1t13r1U1R2u11Z3W3D3a1n9l3m3	38
7	[x1 x2 x3 0 x5 x6 0 x8 x9]	p3g1f1s1b1B1c2T1t13r1U1R3u17Z3W3D3a4n9l3m3	70
7	[x1 x2 x3 0 x5 x6 x7 x8 x5]	p3g2f1s3b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	4
7	[x1 x2 x3 x4 x5 0 -x3 x8 x9]	p3g2f1s1b1B1c1T1t13r1U1R3u10Z3W3D3a4n9l3m3	4

<i>suite de la page précédente</i>			
nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
7	[x1 x2 x3 x4 x5 0 x7 0 x9]	p3g1f3s3b1B1c1T1t13r1U1R3u10Z3W3D3a4n9l3m3	2
7	[x1 x2 x3 x4 x5 x6 0 -x6 x9]	p3g2f1s1b1B1c1T1t13r1U1R3u11Z3W3D3a4n9l3m3	8
7	[x1 x2 x3 x4 x5 x6 0 0 x9]	p3g1f1s3b1B1c2T1t13r1U1R2u19Z3W3D3a1n9l3m3	230
7	[x1 x2 x3 x4 x5 x6 x7 -x6 x5]	p3g1f1s3b1B1c1T1t13r1U1R3u12Z3W3D3a4n9l3m3	3
		TOTAL de cas pour 7 paramètres	422
8	[x1 x2 x3 -x2 x5 x6 x7 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R4u10Z3W3D3a4n9l3m3	8
8	[x1 x2 x3 0 x5 x6 x7 x8 x9]	p3g2f3s3b1B1c1T1t13r1U1R3u19Z3W3D3a4n9l3m3	4
8	[x1 x2 x3 x2 x5 x6 x7 x8 x9]	p3g1f3s1b1B1c1T1t13r1U1R2u12Z3W3D3a1n9l3m3	10
8	[x1 x2 x3 x4 x5 0 x7 x8 x9]	p3g2f1s3b1B1c1T1t13r1U1R3u10Z3W3D3a4n9l3m3	12
8	[x1 x2 x3 x4 x5 x6 0 x8 x9]	p3g1f1s1b1B1c2T1t13r1U1R3u11Z3W3D3a4n9l3m3	56
8	[x1 x2 x3 x4 x5 x6 x7 -x6 x9]	p3g1f1s3b1B1c1T1t13r1U1R4u11Z3W3D3a4n9l3m3	8
8	[x1 x2 x3 x4 x5 x6 x7 0 x9]	p3g1f1s1b1B1c2T1t13r1U1R3u10Z3W3D3a4n9l3m	16
8	[x1 x2 x3 x4 x5 x6 x7 x6 x9]	p3g1f1s3b1B1c1T1t13r1U1R2u12Z3W3D3a1n9l3m3	10
		TOTAL de cas pour 8 paramètres	124

Dans le cas de la projection para-perspective, nous posons $_r = [X_0 \ Y_0 \ Z_0]^T$ utilisons les contraintes **W1** et **W2** pour exprimer le parallélisme ou l'orthogonalité du vecteur de rotation par rapport à l'angle du regard (voir paragraphe 1.7) et utilisons les relations concernant les paramètres β_u et β_v donnés par les équations 1.4. Nous étudions donc les cas de la formes:

pas de translation et toutes rotations sauf R2 (=> W1, r//(X0 Y0 Z0))

p2g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R[1;3-5]u12Z3W1D3a[1;4]n9l3m3

pas de translation et R2 (=> W2, r perp (X0 Y0 Z0))

p2g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R2u[4;12;20]Z3W2D3a1n9l3m3

toutes translations et toutes rotations sauf R2 (=> W1, r//(X0 Y0 Z0))

p2g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[1;3-5]u12Z[1-3]W1D3a[1;4]n9l3m3

toutes translations et R2 (=> W2, r perp (X0 Y0 Z0))

p2g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R2u[4;12;20]Z[1-3]W2D3a1n9l3m3

ce qui fait en tout 14652 cas parmi lesquels, 10440 cas présentent des formes simplifiées de matrice d'homographie et 4212 sont composés de contraintes contradictoires.

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
1	[x1 0 0 0 -x1 -x1 0 0 -x1]	p2g1f1s1b1B1c1T2t5r1U1R2u4Z1W2D3a1n9l3m3	1
1	[x1 0 0 0 x1 0 0 0 x1]	p2g1f1s1b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	60
<i>suite page suivante</i>			

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
1	$[x1\ 0\ 0\ 0\ x1\ x1\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t5r1U1R1u12Z2W1D3a1n9l3m3	20
1	$[x1\ 0\ x1\ 0\ -x1\ 0\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	1
1	$[x1\ 0\ x1\ 0\ x1\ 0\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	40
		TOTAL de cas pour 1 paramètre	122
2	$[x1\ 0\ 0\ 0\ x1\ 0\ 0\ 0\ x9]$	p2g1f3s1b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	20
2	$[x1\ 0\ 0\ 0\ x5\ 0\ 0\ 0\ x5]$	p2g1f1s3b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	20
2	$[x1\ 0\ 0\ 0\ x5\ x5\ 0\ 0\ x5]$	p2g1f1s3b1B1c1T2t5r1U1R1u12Z2W1D3a1n9l3m3	11
2	$[x1\ 0\ x1\ 0\ -x1\ x6\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	1
2	$[x1\ 0\ x1\ 0\ x1\ x6\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t3r1U1R1u12Z2W1D3a1n9l3m3	20
2	$[x1\ 0\ x3\ 0\ -x1\ -x1\ 0\ 0\ -x1]$	p2g1f1s1b1B1c1T2t5r1U1R2u4Z1W2D3a1n9l3m3	1
2	$[x1\ 0\ x3\ 0\ x1\ 0\ 0\ 0\ x1]$	p2g2f1s1b1B1c1T2t4r1U1R1u12Z1W1D3a1n9l3m3	20
2	$[x1\ 0\ x3\ 0\ x1\ x1\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t6r1U1R1u12Z2W1D3a1n9l3m3	60
2	$[x1\ x2\ 0\ 0\ x1\ 0\ 0\ 0\ x1]$	p2g4f1s1b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	10
2	$[x1\ x2\ x1\ 0\ -x1\ 0\ 0\ 0\ x1]$	p2g2f1s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	2
2	$[x1\ x2\ x1\ 0\ x1\ 0\ 0\ 0\ x1]$	p2g4f1s1b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	10
		TOTAL de cas pour 2 paramètres	175
3	$[x1\ 0\ 0\ 0\ -x1\ x6\ 0\ 0\ x9]$	p2g1f3s1b1B1c1T2t5r1U1R2u4Z1W2D3a1n9l3m3	1
3	$[x1\ 0\ 0\ 0\ x1\ x6\ 0\ 0\ x9]$	p2g1f3s1b1B1c1T2t5r1U1R1u12Z2W1D3a1n9l3m3	10
3	$[x1\ 0\ 0\ 0\ x5\ 0\ 0\ 0\ x9]$	p2g1f3s3b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	20
3	$[x1\ 0\ x3\ 0\ -x1\ 0\ 0\ 0\ x9]$	p2g1f3s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	1
3	$[x1\ 0\ x3\ 0\ -x1\ x6\ 0\ 0\ -x1]$	p2g1f1s1b1B1c1T2t5r1U1R2u4Z1W2D3a1n9l3m3	1
3	$[x1\ 0\ x3\ 0\ -x1\ x6\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	1
3	$[x1\ 0\ x3\ 0\ x1\ 0\ 0\ 0\ x9]$	p2g1f3s1b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	10
3	$[x1\ 0\ x3\ 0\ x1\ x6\ 0\ 0\ x1]$	p2g1f1s1b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	620
3	$[x1\ 0\ x3\ 0\ x5\ 0\ 0\ 0\ -x5]$	p2g1f1s3b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	1
3	$[x1\ 0\ x3\ 0\ x5\ 0\ 0\ 0\ x5]$	p2g1f1s3b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	40
3	$[x1\ 0\ x3\ 0\ x5\ x5\ 0\ 0\ x5]$	p2g1f1s3b1B1c1T2t6r1U1R1u12Z2W1D3a1n9l3m3	31
3	$[x1\ x2\ 0\ 0\ x1\ 0\ 0\ 0\ x9]$	p2g2f3s1b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	20
3	$[x1\ x2\ 0\ 0\ x5\ 0\ 0\ 0\ x5]$	p2g2f1s3b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	20
3	$[x1\ x2\ x3\ 0\ -x1\ -x1\ 0\ 0\ -x1]$	p2g2f1s1b1B1c1T2t5r1U1R2u4Z1W2D3a1n9l3m3	4
3	$[x1\ x2\ x3\ 0\ x1\ 0\ 0\ 0\ x1]$	p2g4f1s1b1B1c1T2t4r1U1R1u12Z1W1D3a1n9l3m3	40
3	$[x1\ x2\ x3\ 0\ x1\ x1\ 0\ 0\ x1]$	p2g4f1s1b1B1c1T2t5r1U1R1u12Z2W1D3a1n9l3m3	40
		TOTAL de cas pour 3 paramètres	860
4	$[x1\ 0\ 0\ 0\ x5\ x6\ 0\ 0\ x9]$	p2g1f3s3b1B1c1T2t5r1U1R1u12Z2W1D3a1n9l3m3	11
4	$[x1\ 0\ x3\ 0\ -x1\ x6\ 0\ 0\ x9]$	p2g1f3s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	4

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
4	[x1 0 x3 0 x1 x6 0 0 x9]	p2g1f3s1b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	380
4	[x1 0 x3 0 x5 0 0 0 x9]	p2g1f3s3b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	11
4	[x1 0 x3 0 x5 x6 0 0 -x5]	p2g1f1s3b1B1c2T2t1r1U1R2u20Z1W2D3a1n9l3m3	2
4	[x1 0 x3 0 x5 x6 0 0 x5]	p2g1f1s3b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	321
4	[x1 x2 0 0 x5 0 0 0 x9]	p2g2f3s3b1B1c1T1t13r1U1R1u12Z3W1D3a1n9l3m3	20
4	[x1 x2 x3 0 -x1 0 0 0 x9]	p2g2f3s1b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	2
4	[x1 x2 x3 0 -x1 x6 0 0 -x1]	p2g2f1s1b1B1c3T2t5r1U1R2u4Z1W2D3a1n9l3m3	2
4	[x1 x2 x3 0 -x1 x6 0 0 x1]	p2g2f1s1b1B1c2T2t1r1U1R2u20Z1W2D3a1n9l3m3	4
4	[x1 x2 x3 0 x1 0 0 0 x9]	p2g2f3s1b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	40
4	[x1 x2 x3 0 x1 x6 0 0 x1]	p2g4f1s1b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	320
4	[x1 x2 x3 0 x5 0 0 0 -x5]	p2g2f1s3b1B1c1T2t1r1U1R2u20Z1W2D3a1n9l3m3	2
4	[x1 x2 x3 0 x5 0 0 0 x5]	p2g2f1s3b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	100
4	[x1 x2 x3 0 x5 x5 0 0 x5]	p2g2f1s3b1B1c1T2t5r1U1R1u12Z2W1D3a1n9l3m3	84
4	[x1 x2 x3 x2 -x1 x6 0 0 x3]	p2g1f1s1b1B1c1T2t3r1U1R2u20Z1W2D3a1n9l3m3	2
4	[x1 x2 x3 x2 -x1 x6 0 0 x6]	p2g1f1s1b1B1c1T2t6r1U1R2u20Z1W2D3a1n9l3m3	2
		TOTAL de cas pour 4 paramètres	1307
5	[x1 0 x3 0 x5 x6 0 0 x9]	p2g1f3s3b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	384
5	[x1 0 x3 x4 x5 x6 0 0 -x5]	p2g1f1s1b1B1c1T2t11r1U1R2u20Z1W2D3a1n9l3m3	18
5	[x1 x2 0 x4 x5 0 0 0 x9]	p2g1f1s1b1B1c1T1t13r1U1R2u20Z3W2D3a1n9l3m3	120
5	[x1 x2 0 x4 x5 x6 0 0 x6]	p2g1f1s1b1B1c1T2t5r1U1R2u20Z2W2D3a1n9l3m3	20
5	[x1 x2 x3 0 -x1 x6 0 0 x9]	p2g2f3s1b1B1c1T2t5r1U1R2u4Z1W2D3a1n9l3m3	10
5	[x1 x2 x3 0 x1 x6 0 0 x9]	p2g2f3s1b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	780
5	[x1 x2 x3 0 x5 0 0 0 x9]	p2g2f3s3b1B1c1T2t1r1U1R1u12Z2W1D3a1n9l3m3	42
5	[x1 x2 x3 0 x5 x6 0 0 -x1]	p2g1f1s1b1B1c1T2t10r1U1R2u4Z1W2D3a1n9l3m3	27
5	[x1 x2 x3 0 x5 x6 0 0 -x5]	p2g2f1s3b1B1c2T2t1r1U1R2u20Z1W2D3a1n9l3m3	4
5	[x1 x2 x3 0 x5 x6 0 0 x5]	p2g2f1s3b1B1c1T2t10r1U1R1u12Z2W1D3a1n9l3m3	642
5	[x1 x2 x3 x2 -x1 x6 0 0 x9]	p2g1f1s1b1B1c2T2t3r1U1R2u20Z1W2D3a1n9l3m3	20
5	[x1 x2 x3 x4 -x1 x6 0 0 x6]	p2g2f1s1b1B1c1T2t6r1U1R2u20Z1W2D3a1n9l3m3	2
5	[x1 x2 x3 x4 x5 0 0 0 x3]	p2g1f1s1b1B1c1T2t1r1U1R2u20Z2W2D3a1n9l3m3	30
		TOTAL de cas pour 5 paramètres	2099
6	[x1 0 x3 x4 x5 x6 0 0 x9]	p2g1f1s1b1B1c1T2t11r1U1R2u12Z1W2D3a1n9l3m3	30
6	[x1 x2 0 x4 x5 x6 0 0 x9]	p2g1f3s1b1B1c1T2t5r1U1R2u20Z2W2D3a1n9l3m3	20
6	[x1 x2 x3 0 x5 x6 0 0 x9]	p2g1f1s1b1B1c1T2t10r1U1R2u12Z1W2D3a1n9l3m3	907
6	[x1 x2 x3 x4 -x1 x6 0 0 x9]	p2g2f1s1b1B1c1T2t3r1U1R2u20Z1W2D3a1n9l3m3	10
6	[x1 x2 x3 x4 x5 0 0 0 x9]	p2g1f1s3b1B1c1T2t1r1U1R2u20Z2W2D3a1n9l3m3	138

<i>suite de la page précédente</i>			
nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
6	[x1 x2 x3 x4 x5 x6 0 0 x3]	p2g1f1s1b1B1c1T2t3r1U1R2u20Z2W2D3a1n9l3m3	10
6	[x1 x2 x3 x4 x5 x6 0 0 x6]	p2g1f1s1b1B1c1T2t6r1U1R2u20Z2W2D3a1n9l3m3	108
		TOTAL de cas pour 6 paramètres	1223
7	[x1 x2 x3 x4 x5 x6 0 0 x9]	p2g1f1s1b1B1c1T2t10r1U1R2u20Z2W2D3a1n9l3m3	4654
		TOTAL de cas pour 7 paramètres	4654

Et, enfin, dans le cas orthographique, 8424 cas présentent des formes simplifiées de matrice d'homographie.

pas de translation et toutes rotations (ux=uy=0)

p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R[1-5]u9Z3W3D3a[0;2]n9l3m3

et :

pas de translation et retournement R2 (uz = 0)

p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R2u[1-2;17;19]Z3W3D3a[0;2]n9l3m3

et :

toutes translations et toutes rotations

p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[1-5]u9Z3W3D3a[0;2]n9l3m3

et :

toutes translations et retournement

p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R2u[1-2;17;19]Z3W3D3a[0;2]n9l3m3

ce qui fait en tout 8424 cas étudiés.

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
1	[x1 0 -x1 0 -x1 0 0 0 -x1]	p1g1f1s1b1B1c1T2t1r1U1R2u1Z3W3D3a1n9l3m3	4
1	[x1 0 -x1 0 x1 0 0 0 -x1]	p1g1f1s1b1B1c1T2t1r1U1R2u9Z3W3D3a1n9l3m3	8
1	[x1 0 0 0 -x1 -x1 0 0 -x1]	p1g1f1s1b1B1c1T2t5r1U1R2u1Z3W3D3a1n9l3m3	4
1	[x1 0 0 0 -x1 0 0 0 -x1]	p1g1f1s1b1B1c1T1t13r1U1R2u1Z3W3D3a1n9l3m3	4
1	[x1 0 0 0 -x1 0 0 0 x1]	p1g1f1s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	4
1	[x1 0 0 0 -x1 x1 0 0 x1]	p1g1f1s1b1B1c1T2t5r1U1R2u17Z3W3D3a1n9l3m3	4
1	[x1 0 0 0 x1 -x1 0 0 -x1]	p1g1f1s1b1B1c1T2t5r1U1R2u9Z3W3D3a1n9l3m3	4
1	[x1 0 0 0 x1 0 0 0 -x1]	p1g1f1s1b1B1c1T1t13r1U1R2u9Z3W3D3a1n9l3m3	8
1	[x1 0 0 0 x1 0 0 0 x1]	p1g1f1s1b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	40
1	[x1 0 0 0 x1 x1 0 0 x1]	p1g1f1s1b1B1c1T2t5r1U1R1u9Z3W3D3a1n9l3m3	20

suite page suivante

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
1	$[x1\ 0\ x1\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t1r1U1R2u17Z3W3D3a1n9l3m3	4
1	$[x1\ 0\ x1\ 0\ x1\ 0\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t1r1U1R1u9Z3W3D3a1n9l3m3	40
		TOTAL de cas pour 1 paramètre	144
2	$[x1\ 0\ -x1\ 0\ -x1\ x6\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t3r1U1R2u1Z3W3D3a1n9l3m3	4
2	$[x1\ 0\ -x1\ 0\ x1\ x6\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t3r1U1R2u9Z3W3D3a1n9l3m3	4
2	$[x1\ 0\ 0\ 0\ -x1\ 0\ 0\ 0\ x9]$	p1g1f3s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	8
2	$[x1\ 0\ 0\ 0\ -x1\ x6\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t10r1U1R2u1Z3W3D3a1n9l3m3	2
2	$[x1\ 0\ 0\ 0\ -x1\ x6\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t10r1U1R2u17Z3W3D3a1n9l3m3	12
2	$[x1\ 0\ 0\ 0\ x1\ 0\ 0\ 0\ x9]$	p1g1f3s1b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	14
2	$[x1\ 0\ 0\ 0\ x1\ x6\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t10r1U1R2u9Z3W3D3a1n9l3m3	2
2	$[x1\ 0\ 0\ 0\ x1\ x6\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	10
2	$[x1\ 0\ 0\ 0\ x5\ -x5\ 0\ 0\ -x5]$	p1g1f1s3b1B1c1T2t5r1U1R2u17Z3W3D3a1n9l3m3	8
2	$[x1\ 0\ 0\ 0\ x5\ 0\ 0\ 0\ -x5]$	p1g1f1s3b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	8
2	$[x1\ 0\ 0\ 0\ x5\ 0\ 0\ 0\ x5]$	p1g1f1s3b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	14
2	$[x1\ 0\ 0\ 0\ x5\ x5\ 0\ 0\ x5]$	p1g1f1s3b1B1c1T2t5r1U1R1u9Z3W3D3a1n9l3m3	14
2	$[x1\ 0\ x1\ 0\ -x1\ x6\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t3r1U1R2u17Z3W3D3a1n9l3m3	12
2	$[x1\ 0\ x1\ 0\ x1\ x6\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t3r1U1R1u9Z3W3D3a1n9l3m3	20
2	$[x1\ 0\ x3\ 0\ -x1\ -x1\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t6r1U1R2u1Z3W3D3a1n9l3m3	12
2	$[x1\ 0\ x3\ 0\ -x1\ 0\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t11r1U1R2u1Z3W3D3a1n9l3m3	12
2	$[x1\ 0\ x3\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t11r1U1R2u17Z3W3D3a1n9l3m3	2
2	$[x1\ 0\ x3\ 0\ -x1\ x1\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t6r1U1R2u17Z3W3D3a1n9l3m3	4
2	$[x1\ 0\ x3\ 0\ x1\ -x1\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t6r1U1R2u9Z3W3D3a1n9l3m3	12
2	$[x1\ 0\ x3\ 0\ x1\ 0\ 0\ 0\ -x1]$	p1g1f1s1b1B1c1T2t11r1U1R2u9Z3W3D3a1n9l3m3	4
2	$[x1\ 0\ x3\ 0\ x1\ 0\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	20
2	$[x1\ 0\ x3\ 0\ x1\ x1\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t6r1U1R1u9Z3W3D3a1n9l3m3	60
2	$[x1\ x2\ -x1\ 0\ -x1\ 0\ 0\ 0\ -x1]$	p1g2f1s1b1B1c1T2t1r1U1R2u1Z3W3D3a1n9l3m3	8
2	$[x1\ x2\ -x1\ 0\ x1\ 0\ 0\ 0\ -x1]$	p1g4f1s1b1B1c1T2t1r1U1R2u9Z3W3D3a1n9l3m3	4
2	$[x1\ x2\ 0\ -x2\ x1\ 0\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T1t13r1U1R3u9Z3W3D3a4n9l3m3	2
2	$[x1\ x2\ 0\ -x2\ x1\ x1\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t5r1U1R3u9Z3W3D3a4n9l3m3	2
2	$[x1\ x2\ 0\ 0\ -x1\ 0\ 0\ 0\ -x1]$	p1g2f1s1b1B1c1T1t13r1U1R2u1Z3W3D3a1n9l3m3	8
2	$[x1\ x2\ 0\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1g2f1s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	8
2	$[x1\ x2\ 0\ 0\ x1\ 0\ 0\ 0\ -x1]$	p1g4f1s1b1B1c1T1t13r1U1R2u9Z3W3D3a1n9l3m3	4
2	$[x1\ x2\ 0\ 0\ x1\ 0\ 0\ 0\ x1]$	p1g4f1s1b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	10
2	$[x1\ x2\ x1\ -x2\ x1\ 0\ 0\ 0\ x1]$	p1g1f1s1b1B1c1T2t1r1U1R3u9Z3W3D3a4n9l3m3	2
2	$[x1\ x2\ x1\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1g2f1s1b1B1c1T2t1r1U1R2u17Z3W3D3a1n9l3m3	8

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
2	[x1 x2 x1 0 x1 0 0 0 x1]	p1g4f1s1b1B1c1T2t1r1U1R1u9Z3W3D3a1n9l3m3	10
		TOTAL de cas pour 2 paramètres	324
3	[x1 0 0 0 -x1 x6 0 0 x9]	p1g1f3s1b1B1c1T2t10r1U1R2u17Z3W3D3a1n9l3m3	12
3	[x1 0 0 0 x1 x6 0 0 x9]	p1g1f3s1b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	21
3	[x1 0 0 0 x5 0 0 0 x9]	p1g1f3s3b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	22
3	[x1 0 0 0 x5 x6 0 0 -x5]	p1g1f1s3b1B1c1T2t10r1U1R2u17Z3W3D3a1n9l3m3	4
3	[x1 0 0 0 x5 x6 0 0 x5]	p1g1f1s3b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	7
3	[x1 0 x3 0 -x1 0 0 0 x9]	p1g1f3s1b1B1c1T2t11r1U1R2u17Z3W3D3a1n9l3m3	12
3	[x1 0 x3 0 -x1 x6 0 0 -x1]	p1g1f1s1b1B1c1T2t12r1U1R2u1Z3W3D3a1n9l3m3	36
3	[x1 0 x3 0 -x1 x6 0 0 x1]	p1g1f1s1b1B1c1T2t12r1U1R2u17Z3W3D3a1n9l3m3	36
3	[x1 0 x3 0 x1 0 0 0 x9]	p1g1f3s1b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	21
3	[x1 0 x3 0 x1 x6 0 0 -x1]	p1g1f1s1b1B1c1T2t12r1U1R2u9Z3W3D3a1n9l3m3	114
3	[x1 0 x3 0 x1 x6 0 0 x1]	p1g1f1s1b1B1c1T2t12r1U1R1u9Z3W3D3a1n9l3m3	180
3	[x1 0 x3 0 x5 -x5 0 0 -x5]	p1g1f1s3b1B1c1T2t6r1U1R2u17Z3W3D3a1n9l3m3	8
3	[x1 0 x3 0 x5 0 0 0 -x5]	p1g1f1s3b1B1c1T2t11r1U1R2u17Z3W3D3a1n9l3m3	12
3	[x1 0 x3 0 x5 0 0 0 x5]	p1g1f1s3b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	56
3	[x1 0 x3 0 x5 x5 0 0 x5]	p1g1f1s3b1B1c1T2t6r1U1R1u9Z3W3D3a1n9l3m3	42
3	[x1 x2 0 -x2 x1 0 0 0 x9]	p1g1f1s1b1B1c1T1t13r1U1R4u9Z3W3D3a4n9l3m3	10
3	[x1 x2 0 -x2 x1 x6 0 0 x1]	p1g1f1s1b1B1c1T2t10r1U1R3u9Z3W3D3a4n9l3m3	1
3	[x1 x2 0 -x2 x1 x6 0 0 x6]	p1g1f1s1b1B1c1T2t5r1U1R4u9Z3W3D3a4n9l3m3	4
3	[x1 x2 0 0 -x1 0 0 0 x9]	p1g2f3s1b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	16
3	[x1 x2 0 0 x1 0 0 0 x9]	p1g2f3s1b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	28
3	[x1 x2 0 0 x5 0 0 0 -x5]	p1g2f1s3b1B1c1T1t13r1U1R2u17Z3W3D3a1n9l3m3	16
3	[x1 x2 0 0 x5 0 0 0 x5]	p1g2f1s3b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	28
3	[x1 x2 0 x2 -x1 0 0 0 x9]	p1g1f1s1b1B1c1T1t13r1U1R2u19Z3W3D3a1n9l3m3	16
3	[x1 x2 0 x2 -x1 x6 0 0 x6]	p1g1f1s1b1B1c1T2t5r1U1R2u19Z3W3D3a1n9l3m3	8
3	[x1 x2 x1 -x2 x1 x6 0 0 x1]	p1g1f1s1b1B1c1T2t3r1U1R3u9Z3W3D3a4n9l3m3	2
3	[x1 x2 x3 -x2 x1 0 0 0 x1]	p1g1f1s1b1B1c1T2t11r1U1R3u9Z3W3D3a4n9l3m3	1
3	[x1 x2 x3 -x2 x1 0 0 0 x3]	p1g1f1s1b1B1c1T2t1r1U1R4u9Z3W3D3a4n9l3m3	4
3	[x1 x2 x3 -x2 x1 x1 0 0 x1]	p1g1f1s1b1B1c1T2t6r1U1R3u9Z3W3D3a4n9l3m3	2
3	[x1 x2 x3 0 -x1 -x1 0 0 -x1]	p1g2f1s1b1B1c1T2t5r1U1R2u1Z3W3D3a1n9l3m3	32
3	[x1 x2 x3 0 -x1 0 0 0 -x1]	p1g2f1s1b1B1c1T2t11r1U1R2u1Z3W3D3a1n9l3m3	24
3	[x1 x2 x3 0 -x1 0 0 0 x1]	p1g2f1s1b1B1c1T2t11r1U1R2u17Z3W3D3a1n9l3m3	4
3	[x1 x2 x3 0 -x1 x1 0 0 x1]	p1g2f1s1b1B1c1T2t5r1U1R2u17Z3W3D3a1n9l3m3	16
3	[x1 x2 x3 0 x1 -x1 0 0 -x1]	p1g4f1s1b1B1c1T2t5r1U1R2u9Z3W3D3a1n9l3m3	8

nombre de paramètres	forme simplifiée de matrice homographie	exemple de cas pour ce type de matrice	nombre de cas
3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ 0 \ 0 \ -x_1]$	p1g4f1s1b1B1c1T2t11r1U1R2u9Z3W3D3a1n9l3m3	2
3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ 0 \ 0 \ x_1]$	p1g4f1s1b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	30
3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_1 \ 0 \ 0 \ x_1]$	p1g4f1s1b1B1c1T2t5r1U1R1u9Z3W3D3a1n9l3m3	40
3	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ 0 \ 0 \ 0 \ x_3]$	p1g1f1s1b1B1c1T2t1r1U1R2u19Z3W3D3a1n9l3m3	8
		TOTAL de cas pour 3 paramètres	883
4	$[x_1 \ 0 \ 0 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f3s3b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	33
4	$[x_1 \ 0 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f3s1b1B1c1T2t12r1U1R2u17Z3W3D3a1n9l3m3	124
4	$[x_1 \ 0 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f3s1b1B1c1T2t12r1U1R1u9Z3W3D3a1n9l3m3	217
4	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1g1f3s3b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	33
4	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ -x_5]$	p1g1f1s3b1B1c1T2t12r1U1R2u17Z3W3D3a1n9l3m3	116
4	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T2t12r1U1R1u9Z3W3D3a1n9l3m3	140
4	$[x_1 \ x_2 \ 0 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f1s1b1B1c1T2t10r1U1R4u9Z3W3D3a4n9l3m3	11
4	$[x_1 \ x_2 \ 0 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1g2f3s3b1B1c1T1t13r1U1R1u9Z3W3D3a1n9l3m3	44
4	$[x_1 \ x_2 \ 0 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f1s1b1B1c1T2t10r1U1R2u19Z3W3D3a1n9l3m3	16
4	$[x_1 \ x_2 \ 0 \ x_4 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1g2f1s1b1B1c1T1t13r1U1R2u19Z3W3D3a1n9l3m3	8
4	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T1t13r1U1R3u9Z3W3D3a4n9l3m3	2
4	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_5 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T2t5r1U1R3u9Z3W3D3a4n9l3m3	2
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ 0 \ 0 \ 0 \ x_9]$	p1g1f1s1b1B1c1T2t11r1U1R4u9Z3W3D3a4n9l3m3	11
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_1]$	p1g1f1s1b1B1c1T2t12r1U1R3u9Z3W3D3a4n9l3m3	27
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_3]$	p1g1f1s1b1B1c1T2t3r1U1R4u9Z3W3D3a4n9l3m3	4
4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_6]$	p1g1f1s1b1B1c1T2t6r1U1R4u9Z3W3D3a4n9l3m3	4
4	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1g2f3s1b1B1c1T2t11r1U1R2u17Z3W3D3a1n9l3m3	24
4	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ -x_1]$	p1g2f1s1b1B1c1T2t10r1U1R2u1Z3W3D3a1n9l3m3	84
4	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ x_1]$	p1g2f1s1b1B1c1T2t10r1U1R2u17Z3W3D3a1n9l3m3	120
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ 0 \ 0 \ x_9]$	p1g2f3s1b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	42
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ -x_1]$	p1g4f1s1b1B1c1T2t10r1U1R2u9Z3W3D3a1n9l3m3	60
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ x_1]$	p1g4f1s1b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	105
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ -x_5 \ 0 \ 0 \ -x_5]$	p1g2f1s3b1B1c1T2t5r1U1R2u17Z3W3D3a1n9l3m3	32
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ -x_5]$	p1g2f1s3b1B1c1T2t11r1U1R2u17Z3W3D3a1n9l3m3	24
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_5]$	p1g2f1s3b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	112
4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_5 \ 0 \ 0 \ x_5]$	p1g2f1s3b1B1c1T2t5r1U1R1u9Z3W3D3a1n9l3m3	112
4	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1g1f1s1b1B1c1T2t11r1U1R2u19Z3W3D3a1n9l3m3	16
4	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_3]$	p1g1f1s1b1B1c1T2t3r1U1R2u19Z3W3D3a1n9l3m3	16
4	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ 0 \ 0 \ 0 \ x_3]$	p1g2f1s1b1B1c1T2t1r1U1R2u19Z3W3D3a1n9l3m3	8
		TOTAL de cas pour 4 paramètres	1547

nombre de paramètres	forme simplifiée de matrice homographe	exemple de cas pour ce type de matrice	nombre de cas
5	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f3s3b1B1c1T2t12r1U1R1u9Z3W3D3a1n9l3m3	341
5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1g1f1s3b1B1c1T1t13r1U1R2u19Z3W3D3a1n9l3m3	130
5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T2t10r1U1R3u9Z3W3D3a4n9l3m3	1
5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_6]$	p1g1f1s3b1B1c1T2t5r1U1R2u19Z3W3D3a1n9l3m3	12
5	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f1s1b1B1c1T2t12r1U1R4u9Z3W3D3a4n9l3m3	147
5	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g2f3s1b1B1c1T2t10r1U1R2u17Z3W3D3a1n9l3m3	272
5	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g2f3s1b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	476
5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1g2f3s3b1B1c1T2t11r1U1R1u9Z3W3D3a1n9l3m3	66
5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ -x_5]$	p1g2f1s3b1B1c1T2t10r1U1R2u17Z3W3D3a1n9l3m3	240
5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1g2f1s3b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	294
5	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f1s1b1B1c1T2t12r1U1R2u19Z3W3D3a1n9l3m3	232
5	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1g2f1s1b1B1c1T2t11r1U1R2u19Z3W3D3a1n9l3m3	4
5	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ x_6 \ 0 \ 0 \ x_6]$	p1g2f1s1b1B1c1T2t5r1U1R2u19Z3W3D3a1n9l3m3	16
5	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_3]$	p1g2f1s1b1B1c1T2t1r1U1R3u9Z3W3D3a4n9l3m3	20
5	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T2t11r1U1R3u9Z3W3D3a4n9l3m3	3
5	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_5 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T2t6r1U1R3u9Z3W3D3a4n9l3m3	2
		TOTAL de cas pour 5 paramètres	2256
6	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f1s3b1B1c1T2t10r1U1R2u19Z3W3D3a1n9l3m3	27
6	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1g2f3s3b1B1c1T2t10r1U1R1u9Z3W3D3a1n9l3m3	748
6	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1g2f1s1b1B1c1T2t10r1U1R2u19Z3W3D3a1n9l3m3	120
6	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1g1f1s3b1B1c1T2t11r1U1R2u19Z3W3D3a1n9l3m3	175
6	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1g1f1s3b1B1c1T2t12r1U1R3u9Z3W3D3a4n9l3m3	29
6	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_6]$	p1g1f1s3b1B1c1T2t6r1U1R2u19Z3W3D3a1n9l3m3	108
		TOTAL de cas pour 6 paramètres	1207
7	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1g1f1s3b1B1c1T2t12r1U1R2u19Z3W3D3a1n9l3m3	2063
		TOTAL de cas pour 7 paramètres	2063

Si nous comptabilisons toutes les formes d'homographies, sachant que certaines apparaissent plusieurs fois, cela nous fait 197 formes différentes, dont

nombre de paramètres	nombre de formes
1	12
2	35
3	46
4	35
5	29
6	19
7	13
8	8

2.9 Conclusion

Nous avons étudié comment générer et gérer un inventaire exhaustif de cas particuliers de projection, mouvement et structure. Cependant, la complexité du problème est un obstacle à un examen exhaustif de l'ensemble des cas particuliers. Ceci nous a conduit à établir des hiérarchies de cas et nous conduit à plusieurs pistes :

- parcourir de façon adéquate l'arbre issu du produit cartésien des arbres élémentaires de cas particuliers en utilisant des critères statistiques ou des mécanismes de types algorithmes génétiques (voir chapitre suivant)
- s'intéresser aux formes particulières des matrices fondamentales ou homographiques, ce qui nous permet de scinder en deux parties équitables la recherche du cas particulier correspondant aux données

Il nous faut maintenant valider expérimentalement l'apport de ces cas particuliers, et pour cela, nous allons maintenant étudier les techniques de minimisation et d'estimation de paramètres. On pourra alors s'attaquer à la validation expérimentale ainsi qu'aux applications qui en découlent.

Chapitre 3

Estimation

Quels outils? Quelles techniques?

Ce chapitre a pour but de faire le tour des méthodes utilisées pour estimer des paramètres ainsi que les méthodes permettant d'évaluer la précision de l'estimation et l'adéquation du modèle ainsi pa-

ramétré. Nous présenterons alors les techniques qui ont été utilisées dans les expérimentations du chapitre suivant tout en mentionnant les modifications qu'il serait bon d'essayer afin d'accroître la qualité des résultats.

*« Oh thésarde ma soeur
Songe à la douceur
D'une biblio tendre
Ecris à loisir
Relue à mourir
Et qui ne fait plus attendre »
Thierry Viéville*

Estimation (english abstract)

Yeah, great, we have some interesting models of data. But, now, how to fit a model to the data? How to measure the adequation? How to compare several models with different numbers of parameters?

This chapter has the answer. Just read it.

3.1 Introduction

Dans ce chapitre, nous cherchons à analyser les méthodes qui permettent de déterminer les paramètres du modèle qui correspond à nos données. Nos données correspondent à des appariements de points entre deux images. Pour chaque couple de points en correspondance, nous disposons d'un coefficient d'erreur. Le modèle à estimer à partir de ces données est l'un de ceux vus au chapitre 2. Pour ce faire, nous allons tout d'abord nous intéresser aux techniques de détermination des paramètres d'un modèle donné. Ensuite, nous examinerons les techniques de validation des modèles pour déterminer si un modèle correspond ou non aux données ainsi que les techniques de comparaison de différents modèles.

Comme dans toute application réelle, les données sont bruitées et contiennent des artefacts (points mal localisés, erreurs d'appariements, ...). On cherchera dans un premier temps à étudier ce bruit et ces erreurs afin de l'éliminer au mieux. Différentes techniques, dites robustes, seront alors proposées afin d'éliminer le bruit et les données aberrantes. Ne pouvant connaître la valeur exacte des paramètres recherchés, on s'attachera, dans tous les cas, à obtenir un maximum de vraisemblance.

Enfin, après avoir passé en revue les principales méthodes statistiques, nous développerons notre choix et analyserons les spécificités des problèmes de vision en cause qui justifient notre approche.

3.2 Le bruit

3.2.1 Origine des incertitudes.

L'acquisition des images est effectuée par l'intermédiaire de capteurs visuels qui engendrent des déformations optiques. Elles sont ensuite numérisées, ce qui a pour conséquence l'ajout d'un bruit électronique ainsi que d'une quantification aussi bien en espace (résolution des pixels) qu'en intensité (niveaux de gris de 0 à 255).

Dans notre cas, les points d'intérêt ou coins correspondant aux points de plus grande courbure sont déterminés selon la méthode de Harris (Harris et Stephens, 1988). La localisation de ces points a une précision liée à la résolution de l'image ou à certaines configurations géométriques des points frontières entre plusieurs zones, dont le coin est issu. Par ailleurs, certains points d'intérêts déterminés ne correspondent pas à un point physique de la scène mais correspondent soit à des contours d'occultation, c'est-à-dire des contours apparents liés au point de vue, soit à des zones d'ombre ou des reflets qui dépendent du point de vue d'observation et des sources de lumière. Par conséquent, leur mouvement projeté ne correspond pas au mouvement rigide de l'objet.

A l'étape suivante, étant donné un point d'intérêt dans une image, on utilise une fenêtre de corrélation centrée sur ce point et on sélectionne un espace rectangulaire de recherche autour de ce point dans la seconde image. On effectue une opération de corrélation entre la fenêtre centrée dans la première image et toutes les fenêtres centrées autour d'un point de l'espace de recherche dans la seconde image. On sélectionne alors les positions pour lesquelles le score de corrélation est supérieur à un seuil. On dispose ainsi de plusieurs candidats à l'appariement. On élimine les ambiguïtés

en utilisant un mécanisme de relaxation consistant à propager des contraintes sur les candidats à l'appariement : (i) on impose que chaque point n'ait qu'un et un seul correspondant dans l'autre image et (ii) que si $(\mathbf{m}_{1i}, \mathbf{m}_{2j})$ et $(\mathbf{m}_{1k}, \mathbf{m}_{2l})$ sont deux couples de points en correspondances tels que \mathbf{m}_{1k} soit dans un voisinage de \mathbf{m}_{1i} et \mathbf{m}_{2l} dans un voisinage de \mathbf{m}_{2j} alors la position relative de \mathbf{m}_{1k} par rapport à \mathbf{m}_{1i} est similaire à celle de \mathbf{m}_{2l} par rapport à \mathbf{m}_{2j} . En pratique, on contraint l'homothétie et la rotation entre les points d'une image à l'autre à être en dessous d'un certain seuil. Pour plus de détails, le lecteur pourra se référer à (Zhang et al., 1995) par exemple. Cependant, des erreurs concernant des points proches mais non en correspondances physiques peuvent subsister car nous n'avons pas utilisé ici toutes les contraintes du problème, à savoir la rigidité des objets ou les contraintes épipolaires détaillées dans le chapitre 1. En effet, sachant que nous nous intéressons justement aux cas singuliers, nous ne pouvons prendre en compte ces contraintes générales évidemment invalides dans un cas singulier. Par contre, dans les séquences vidéo que nous allons traiter, la disparité entre deux images consécutives est faible et donc, on pourra, à l'aide des mécanismes de corrélation et relaxation, mais aussi par un mécanisme de suivi, obtenir des correspondances avec moins d'appariements aberrants. On pourra alors traiter les images une à une ou bien ne conserver que l'image initiale et la finale sur lesquelles les correspondances seront bien meilleures que si on les avait obtenues uniquement en analysant ces deux images.

Dans une même scène, certains ensembles de points suivent un modèle tandis que d'autres suivent un autre modèle (géométries différentes, mouvements différents, ...). De plus, on doit considérer que pour un modèle donné, par exemple celui correspondant à une rotation pure, un sous-ensemble des données vérifie ce modèle avec un paramétrage et qu'un autre sous-ensemble de points suit également ce modèle ou un autre mais avec un paramétrage différent (exemple, deux objets en rotation à des vitesses de rotation différentes). On va donc rechercher des modèles à travers leurs paramètres, satisfaisant le plus de points possibles. Les autres points seront alors considérés comme aberrants car ne satisfaisant pas le modèle choisi ou car provenant de données aberrantes. Nous verrons ultérieurement que nous pourrions alors segmenter les points en sous-ensembles répondant à des modèles ou des paramétrages de modèles différents. Cependant, pour les points considérés comme non aberrants, on observe des erreurs dues aux étapes précédentes ainsi que des erreurs dues au modèle qui est forcément une approximation de la réalité (voir chapitre 1) et aux erreurs de résolution numérique.

3.2.2 Modélisation du bruit

Au delà des points aberrants pour un modèle donné, les sources d'erreurs sont vues comme un bruit qui provient de plusieurs facteurs mais se décompose généralement en un terme d'erreur systématique ou biais et un terme d'erreur statistique, qui disparaît par moyennage quand on prend suffisamment de données. Un biais apparaît lorsque le modèle ne correspond pas aux données. En ce qui concerne les erreurs statistiques, celles-ci sont modélisées par la plupart des auteurs sous la forme d'un bruit gaussien de moyenne nulle et de variance σ . En notant n le nombre de variables

indépendantes et \mathbf{x} le vecteur de paramètres, la densité de probabilité s'écrit :

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{\mathbf{x}^T \Sigma^{-1} \mathbf{x}}{2}\right)$$

Différents auteurs ont essayé de modéliser le bruit autrement que par un bruit gaussien mais Kanatani (Kanatani, 1996) justifie l'approche du bruit gaussien par le fait que c'est la plus réaliste (loi des grands nombres) et reproche à ces auteurs de finalement effectuer le même raisonnement sous une forme déguisée.

Un argument supplémentaire en faveur de la loi gaussienne est que, sachant qu'on s'intéresse généralement à la moyenne et à la variance d'une variable aléatoire, la loi qui minimise l'information restante :

$$I[\mathbf{x}|\bar{x} = \mu, \Sigma_{\mathbf{xx}} = \Sigma]$$

est la loi gaussienne de moyenne μ et de variance Σ .

Une loi gaussienne possède également une autre propriété très intéressante, à savoir que sa variance suit une loi du χ^2 (chi-deux) à $(n - p)$ degrés de liberté représentant la différence entre le nombre de données et le nombre de paramètres : $\chi^2(n - p)$. La densité de probabilité d'une loi du χ^2 à r degrés de liberté s'exprime par :

$$\gamma\left(\frac{r}{2}, \frac{1}{2}\right) = p_r(\xi^2) = \frac{1}{2^{r/2} \Gamma(r/2)} (\xi^2)^{r/2-1} e^{-(\xi^2)/2} \text{ avec } \begin{cases} E[\xi^2] &= r \\ \arg \max_{\xi^2} [\Gamma_r(\xi^2)] &= r - 2 \end{cases}$$

définie sur $[0, \infty[$ où $\Gamma(r) = \int_0^\infty t^{r-1} e^{-t} dt$.

Test d'hypothèses à partir de l'interprétation probabiliste. Pour une valeur donnée de $\xi^2(\mathbf{x})$, $P(\xi_0^2; r) = \int_0^{\xi_0^2} \gamma_r(\xi^2) d\xi^2$ est la probabilité que, pour une modélisation correcte, la valeur observée soit plus petite que le seuil ξ_0^2 tandis que $1 - P(\xi_0^2; n)$ est la probabilité que, même pour une modélisation correcte, la valeur observée soit plus grande que le seuil ξ_0^2 . Cela revient à fixer un seuil statistique, i.e. une probabilité d'erreur. On peut alors faire des tests qualitatifs sur une valeur \mathbf{x} , estimée sur un échantillon donné, de la manière suivante :

\mathcal{H}_1 si $\xi^2(\mathbf{x}) > \xi_0^2$ alors ce point est significativement différent de $\bar{\mathbf{x}}$, compte-tenu de la précision évaluée à travers \mathbf{X} et de la probabilité d'erreur choisie.

\mathcal{H}_0 si $\xi^2(\mathbf{x}) < \xi_0^2$ alors, soit ce point n'est pas différent de $\bar{\mathbf{x}}$, soit la précision évaluée à travers \mathbf{X} ne permet pas de voir cette différence.

On peut donc uniquement différencier deux valeurs données à travers ces tests, mais il n'est pas possible de décider que deux valeurs sont égales.

La théorie probabiliste permet d'éliminer le bruit gaussien mais suppose que l'on dispose d'un nombre infini de données. Cela étant irréalisable en pratique, nous utilisons un modèle statistique qui demande quant à lui un nombre fini de données.

3.2.3 Elimination du bruit: les techniques robustes

En dehors du bruit gaussien, certains points sont aberrants car sont très éloignés du modèle. Ils perturbent le résultat de façon significative surtout lorsque les données sont proches d'une configuration dégénérée. Ils sont généralement modélisés par une distribution normale.

Les méthodes robustes sont les méthodes qui permettent d'éliminer les données aberrantes tout en minimisant partiellement le bruit additif gaussien. Ces méthodes sont basées sur le principe de maximum de vraisemblance (*MLE: maximum likelihood estimators*) qui consiste à identifier intuitivement la probabilité de données vis-à-vis de paramètres et la probabilité de paramètres vis-à-vis de données.

Un estimateur robuste est caractérisé par :

son efficacité relative (*relative efficiency*) : égale au rapport entre la plus petite variance atteignable (borne de Cramér-Rao) et la variance de la méthode.

un point d'arrêt (*breakdown point*) : représentant la plus petite proportion de données aberrantes qui mette en péril l'algorithme.

Les estimateurs robustes modernes sont de trois familles différentes : les estimateurs par destruction de points (*Case Deletion Diagnostics*), les M-estimateurs (*M-estimators*), les estimateurs par échantillonnage aléatoire (*Random Sampling Algorithms*). Nous détaillerons ces familles dans la suite de ce chapitre.

3.3 Les techniques de minimisation.

Il existe de nombreuses techniques de minimisation dans la littérature et nous allons développer ici (i) dans un but didactique, les plus employées en vision par ordinateur, (ii) dans un but pragmatique, celles présentant un intérêt pour notre problème.

Ces différentes méthodes peuvent être classées en méthodes concernant des problèmes linéaires ou non-linéaires et parmi ces méthodes, on distingue également les méthodes robustes des méthodes non-robustes. Nous ne parlerons pas ici des méthodes qui prennent en compte des contraintes car, dans notre cas, celles-ci sont éliminées au niveau de ce que nous avons appelé les cas particuliers. De même, nous nous limiterons aux méthodes robustes.

La plupart des algorithmes nécessite le calcul d'une estimée du résultat sur un sous-ensemble minimal de données. Cette estimée n'est donc pas forcément robuste et les méthodes varient s'il s'agit d'un problème linéaire ou non.

Un problème est considéré comme linéaire, si, pour N variables indépendantes X_1, x_2, \dots, x_N , on maximise la fonction suivante :

$$z = a_{01} x_1 + a_{02} x_2 + \dots + a_{0N} x_N$$

Les problèmes linéaires sont mis sous forme matricielle et leur solution exacte, quant elle existe, est connue. Un algorithme standard est l'algorithme des moindres carrés linéaires.

Parmi les problèmes non-linéaires, on distingue ceux pour lesquels on sait calculer directement une estimée des paramètres en inversant les équations du modèle et ceux pour lesquels l'équation du modèle n'est pas inversible. Par exemple, à partir de deux points, on sait calculer les paramètres d'une droite. Pour des structures plus compliquées ce n'est pas forcément le cas et on doit alors recourir à des méthodes itératives de résolution numérique tels que l'utilisation du gradient, du Hessien, ...

3.3.1 Une méthode robuste historique : la transformée de Hough

C'est une des méthodes les plus anciennes en traitement d'image et vision par ordinateur qui consiste à découper l'espace des paramètres en un nombre fini de cellules de tailles identiques et de faire voter pour une cellule chaque sous-ensemble de n données permettant le calcul des paramètres. La solution est dans la cellule qui a obtenu le plus de vote et sa précision est liée à la taille de la cellule.

Cette méthode, bien qu'ayant pour avantage crucial sa robustesse, a pour désavantage majeur sa complexité qui est exponentielle en la dimension de l'espace de recherche, c'est-à-dire le nombre de paramètres.

3.3.2 Méthodes de régression (*Case Deletion Diagnostic*).

Le principe de ces méthodes est que l'on détermine les paramètres sur un ensemble initial de données. Puis, pour chaque donnée, on détermine son influence en comparant le résultat obtenu avec et sans cette donnée. Si son influence est statistiquement significative la donnée est rejetée (c'est une donnée aberrante). L'ensemble des données est alors mis-à-jour.

Ces méthodes convergent vers la solution si l'ensemble initial des données est bien choisi. S'il est très pauvre en information, on aura des paramètres très loin de leur vraie valeur et la signification des résidus sera pauvre.

De nombreuses variantes existent mais ont toutes pour inconvénient majeur de nécessiter une initialisation correcte des paramètres.

3.3.3 Moindres carrés linéaires.

Nous présentons cette méthode comme étant une des méthodes les plus classiques mais possédant l'inconvénient majeur de ne pas être robuste dans sa forme générale. Par contre, si on lui ajoute des coefficients de pondération choisis judicieusement, on peut rendre robuste cette méthode, par exemple, en utilisant la méthode précédente (3.3.2). Cette méthode a ensuite été modifiée pour donner par exemple les M-estimateurs (3.3.8).

Supposons que l'on veuille ajuster les m paramètres a_j d'un modèle avec n données (x_i, y_i) disponibles. Le modèle s'exprime par $y(x) = y(x, a_1, \dots, a_m)$ et on cherche à minimiser :

$$\sum_{i=1}^n [y_i - y(x_i, a_1, \dots, a_m)]^2$$

Si de plus on a accès aux erreurs σ_i de mesures des y_i , on va minimiser :

$$\chi^2 = \frac{1}{n} \sum_{i=1}^n \frac{[y_i - y(x_i, a_1, \dots, a_m)]^2}{\sigma_i^2}$$

Dans le cas où les fonctions y sont linéaires en les paramètres a_1, \dots, a_m , on peut exprimer la quantité à minimiser sous forme matricielle :

$$|\mathbf{A} \mathbf{a} - \mathbf{b}|^2$$

où la linéarité de la fonction y s'exprime par :

$$y(x_i, a_1, \dots, a_m) = \sum_{j=1}^m a_j Y_j(x_i)$$

tandis que les quantités suivantes sont introduites :

$$\mathbf{A} = \left(\frac{Y_j(x_i)}{\sigma_i} \right)_{ij} \quad \text{et} \quad \mathbf{b} = \frac{y_i}{\sigma_i}$$

On décompose ensuite la matrice \mathbf{A} en valeurs singulières :

$$\mathbf{A} = \mathbf{U} \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \dots & & \\ & & & \dots & \\ & & & & \sigma_n \end{pmatrix} \mathbf{V}$$

La solution est alors :

$$\mathbf{a} = \mathbf{V} \begin{pmatrix} \frac{1}{\sigma_1} & & & & \\ & \frac{1}{\sigma_2} & & & \\ & & \dots & & \\ & & & \dots & \\ & & & & \frac{1}{\sigma_n} \end{pmatrix} \mathbf{U}^T \mathbf{b}$$

3.3.4 Moindres carrés non-linéaires: Levenberg-Marquardt

Cette méthode consiste à minimiser la même expression que dans le cas linéaire $f(\mathbf{a})$ mais de façon itérative. Nous présentons ici la méthode de Levenberg-Marquardt qui est la méthode de moindres carrés non-linéaires la plus répandue dans le domaine.

Tout comme dans le cas linéaire, on va utiliser le χ^2 mais on suppose que son comportement est quadratique au voisinage de la solution.

En supposant qu'il est possible de calculer à la fois le gradient \mathbf{b} et le Hessien \mathbf{A} au point \mathbf{a} , si le Hessien est une bonne approximation (on est proche de la solution), alors on itère en utilisant ces informations :

$$\mathbf{a} \mapsto \mathbf{a} - \mathbf{A}^{-1} \mathbf{b}$$

sinon, si on est loin de la solution, on utilise une descente du gradient :

$$\mathbf{a} \mapsto \mathbf{a} - \beta \mathbf{b}$$

L'algorithme de Levenberg-Marquardt combine ces deux alternatives en transformant les éléments diagonaux de \mathbf{A} de la façon suivante :

$$a_{ij} \mapsto (1 + \lambda) a_{ij} = a'_{ij}$$

où λ est une constante de régularisation que l'on va faire décroître au fur et à mesure de la convergence. L'algorithme se comporte alors comme suit :

- calculer \mathbf{x} , solution de $\mathbf{A}' \mathbf{x} - \mathbf{b} = \mathbf{0}$
- calculer $f(\mathbf{a} + \mathbf{x})$
- si $f(\mathbf{a} + \mathbf{x}) \geq f(\mathbf{a})$ multiplier λ par un facteur 10 (on est loin de la solution)
- si $f(\mathbf{a} + \mathbf{x}) \leq f(\mathbf{a})$ diviser λ par un facteur 10 (on s'est approché de la solution)

L'algorithme s'arrête lorsque le critère de convergence est satisfait.

3.3.5 Méthode de la moindre médiane des carrés (*LMedS: Least-Median Square*)

Cette méthode utilise la technique d'échantillonnage aléatoire : au lieu d'estimer les paramètres pour chaque sous-ensemble de données, on ne les estime que sur certains sous-ensembles, tirés aléatoirement, ceci afin de réduire la complexité de l'algorithme. On minimise ici non pas la somme des carrés des distances mais la médiane de ces mêmes carrés, ce qui permet d'avoir jusqu'à 50% de données erronées. L'algorithme est le suivant :

1. On réalise aléatoirement m sous-ensembles de données de taille minimale (la plus petite taille qui permette d'estimer les paramètres).
2. Pour chaque sous-ensemble j , on calcule les paramètres $A_j = (a_i)_{1 \leq i \leq m}$ (il peut exister plusieurs solutions).
3. Pour chaque solution, on calcule la médiane du résidu au carré M_j sur toutes les données (voir figure 3.1).
4. On conserve la solution pour laquelle M_j est minimale.
5. On affine finalement le résultat : on compense la non-optimalité de la méthode au bruit gaussien en utilisant l'estimée de la déviation standard robuste: $\hat{\sigma} = 1,4826 [1 + \frac{5}{(n-p)}] \sqrt{M_j}$. On élimine ainsi les données aberrantes que l'on qualifie ainsi si, r_k étant le résidu de la donnée k , : $r_k^2 \geq (2,5 \hat{\sigma})^2$.

Pour estimer le nombre m de sous-échantillons de taille p que l'on doit tirer à la première étape, on calcule la probabilité qu'au moins un des sous-échantillons fournisse une bonne estimation. En notant ε la fraction de points aberrants,

$$P = 1 - [1 - (1 - \varepsilon)^p]^m \implies m = \frac{\log(1 - P)}{\log(1 - (1 - \varepsilon)^p)}$$

Si on pose, par exemple, $\varepsilon = 40\%$, $p = 7$ et $P = 0,99$, on obtient $m = 163$.

Certains auteurs affinent par la suite les paramètres par des techniques de Monte-Carlo comme par exemple Zhengyou Zhang dans (Zhang et al., 1995).

3.3.6 Méthode de la ν -iane (ν -trimmed Square).

Cette méthode est une généralisation de la méthode de la médiane (qui consiste à prendre $\nu = \frac{1}{2}$). On peut alors avoir jusqu'à $100 * \nu \%$ de données erronées. A l'étape 3, on classe les résidus et on prend le $\nu^{\text{ième}}$ résidu pour N_j (voir figure 3.1).

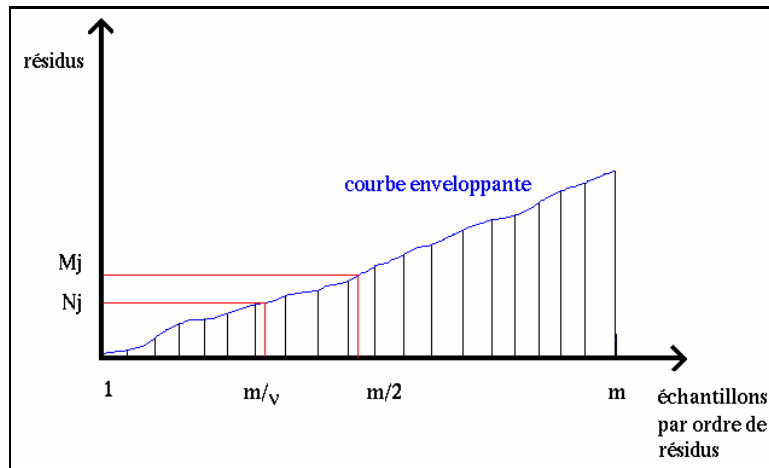


FIG. 3.1 – Détermination de la médiane et ν -iane

3.3.7 RANSAC (*RAN*d*om SA*m*ple Co*n*sen*sus)

Le principe de cet algorithme (Bolles et Fischler, 1981) consiste à trouver un sous-ensemble minimal de points qui permette d'estimer les paramètres de façon cohérente avec le plus grand ensemble de données. Cet algorithme est très proche de celui de la moindre médiane des carrés (LMedS) ou de la ν -iane aux différences que :

- RANSAC utilise un seuil sur l'erreur tandis que la ν -iane utilise un pourcentage de données correctes à priori,
- la ν -iane s'intéresse à la valeur d'erreur du pourcentage donné ν alors que RANSAC s'intéresse au nombre de données en accord,
- RANSAC est moins coûteux car il peut quitter la boucle d'échantillonnage aléatoire dès qu'une solution consistante est trouvée
- la ν -iane est instable lorsque plus de ν données sont dans une configuration de solutions multiples (exemple: un plan pour la détermination de la matrice fondamentale)

3.3.8 M-Estimateurs

La méthode des moindres carrés standard consiste à minimiser la somme des carrés des résidus :

$$\sum_{i=1}^n [r_i]^2$$

qui est très instable en présence de données aberrantes.

Les M-estimateurs ont pour but d'éliminer l'influence des données aberrantes en remplaçant le carré du résidu par une fonction ρ du résidu, symétrique, définie positive avec un unique minimum en zéro, et de variation plus faible que la fonction carré.

$$\sum_{i=1}^n \rho(r_i)$$

Le minimum est atteint lorsque les dérivées en chaque paramètre a_j s'annulent :

$$\sum_{i=1}^n \psi(r_i) \frac{\partial r_i}{\partial a_j} = 0 \text{ où } \psi(x) = \frac{\partial \rho(x)}{\partial x}$$

La fonction ψ est appelée fonction d'influence et représente l'influence d'une donnée sur la valeur des paramètres.

Si la fonction d'influence de ψ , dérivée de ρ n'est pas bornée, alors des points fortement aberrants auront une influence non négligeable sur le résultat: la méthode n'est alors pas robuste.

Différentes fonctions ρ existent dans la littérature. Citons pour exemple (ces exemples ainsi que d'autres encore sont détaillés dans (Zhang, 1997b)) :

$$\begin{array}{l} L_p \\ \text{Huber} \\ \text{Cauchy} \\ \text{Tukey} \end{array} \begin{cases} \frac{|r|^\nu}{\nu} \\ \text{si } |r| \leq k \quad \frac{r^2}{2} \\ \text{si } |r| \geq k \quad k(|r| - \frac{k}{2}) \\ \frac{c^2}{2} \log(1 + (\frac{r}{c})^2) \\ \text{si } |r| \leq c \quad \frac{c^2}{6} (1 - [1 - (\frac{r}{c})^2]^3) \\ \text{si } |r| \geq c \quad \frac{c^2}{6} \end{cases} \quad (3.1)$$

La fonction de Huber propose les meilleurs résultats dans une large gamme de situations à quelques exceptions près où des difficultés apparaissent, dues à l'instabilité du gradient (la dérivée seconde est non continue en k).

Pour les fonctions de la famille L_p (*least-powers*), plus le paramètre ν est petit, moins les données aberrantes de grand résidu auront d'influence mais pas trop petit non plus afin d'estimer correctement les paramètres; un bon compromis est réalisé avec $\nu = 1.2$ (Rey, 1983) mais on rencontre tout de même des difficultés de calcul.

Les fonctions de Cauchy et de Tukey présentent l'intérêt de ne pas subir l'influence des données aberrantes de fort résidu mais ont pour inconvénient de ne pas garantir l'unicité de la solution.

Une solution proposée par Huber (Huber, 1981) est alors de commencer le calcul itératif avec une fonction convergente mais sensible aux données aberrantes (par exemple celle de Huber) et d'affiner le résultat avec une autre fonction robuste (par exemple celle de Tukey).

Le problème général est que cette classe de méthodes ne supporte qu'une faible proportion de données aberrantes (faible point d'arrêt).

3.3.9 Comparaison des méthodes

Sachant qu'on veut pouvoir éliminer une proportion de données aberrantes souvent plus grande que 50% (plusieurs objets en mouvement, ...), seules les méthodes de la ν -iane et RANSAC répondent à nos besoins. Dans (Torr et Murray, 1997), Torr compare différentes méthodes et préfère la méthode RANSAC suivie d'une estimation-M. D'après les données utilisées dans ce papier, la méthode de la moindre médiane des carrés propose toutefois des résultats quasiment similaires que nous améliorons par la ν -iane dans les cas où la proportion de données aberrantes dépasse 50 %, suivie d'une minimisation aux moindres carrés non-linéaire pour laquelle les données aberrantes sont éliminées.

3.4 Evaluation du résultat

Nous allons maintenant supposer que nous savons comment, à partir de données et d'un modèle paramétré, évaluer les paramètres de ce modèle. Le problème que nous nous posons maintenant est de savoir comment évaluer la qualité des paramètres estimés pour ce modèle, et, d'autre part, de pouvoir comparer différents modèles entre eux.

Nous distinguons dans la suite les méthodes qui consistent à évaluer un modèle de celles qui tentent de comparer différents modèles les uns aux autres.

3.4.1 Evaluation d'un modèle

Pour évaluer un modèle, c'est-à-dire calculer son résidu ou taux d'erreur, on se doit de définir une grandeur physique que l'on va chercher à mesurer (voir paragraphe 3.7.2). Se pose alors la question de prédictabilité du modèle. Commençons par une approche simpliste, l'estimation par resubstitution que l'on cherchera ensuite à améliorer.

3.4.1.1 Estimation par resubstitution

C'est la méthode la plus triviale pour laquelle les paramètres sont déterminés avec un échantillon de données qui est ensuite réutilisé pour le calcul d'erreur. Le résultat, appelé "taux apparent d'erreur", n'est pas très bon, surtout si l'échantillon est petit et le modèle complexe. On dit qu'il y a un fort biais d'optimisme.

3.4.1.2 Estimation par échantillon test

Dans ce cas, l'échantillon de test est indépendant de l'échantillon qui a servi à estimer les paramètres; on élimine ainsi le biais d'optimisme de la méthode précédente. Par contre, sachant que la précision sur les paramètres est fonction du nombre de données et que cette taille a justement diminué, on y perd en précision. Globalement, on ne gagne pas beaucoup en précision sur le taux d'erreur mais on a pu vérifier les paramètres. Alors, après le test, on évalue à nouveaux les paramètres sur toutes les données.

3.4.1.3 Estimation par validation croisée ou *Cross Validation*

Cette méthode de calcul d'erreur repose sur le même principe que la méthode de régression vue auparavant (3.3.2).

On prélève un à un chaque élément i de l'échantillon d'apprentissage et on estime les paramètres par rapport à la base privée de l'élément i . On calcule alors l'erreur entre élément prélevé et les paramètres estimés. On pose $T_i = 1$ si l'élément i est correct, $T_i = 0$ sinon. Une estimation du taux d'erreur est alors donnée par :

$$\frac{1}{n} \sum_{i=1}^n T_i$$

Cette méthode est apparemment bien plus coûteuse (n fois) mais on réduit grandement ce coût en considérant que de nombreux calculs sont parallélisables. Cependant, cette méthode reste moins utilisée que la méthode suivante qui est un peu moins précise dans le cas général mais bien plus facile à mettre en oeuvre. D'autre part, on peut obtenir une grande variance lorsque le taux d'erreur est élevé.

3.4.1.4 Estimation par canif ou méthode de Quenouille-Tukey (*jackknife*)

Cette méthode, inventée par Quenouille (Quenouille, 1956) et reprise par Tukey (Tukey, 1958), ressemble beaucoup à la précédente à la différence près que le biais d'optimisme est estimé plus en détail.

Quenouille s'est inspiré du fait que plus l'échantillon est petit, plus le biais est grand. Entre les tailles d'échantillons n et $n - 1$, une estimation de la variation du biais est donnée en calculant le taux apparent d'erreur à la taille n ainsi que la moyenne des taux apparents des n échantillons privés de l'élément i . On multiplie alors par $(n - 1)$ pour obtenir une estimation du biais.

La qualité est cependant équivalente à la méthode par validation croisée qui est de plus, plus simple à mettre en oeuvre.

3.4.1.5 Estimation par bretelles (*bootstrap*)

La méthode des bretelles est due à Efron (Efron, 1982) et consiste à construire N échantillons de taille M . Sur chaque échantillon j , on calcule le taux apparent d'erreur sur l'échantillon \hat{T}_b et l'erreur sur l'ensemble des données.

Le lecteur désireux de plus amples détails pourra se référer à (Chavance, 1992) ou bien encore (Celeux et Nakache, 1994).

Ces méthodes d'évaluation ont pour inconvénients de ne considérer qu'un seul modèle pris isolément. Dans le cadre de notre étude, ce qui nous intéresse, c'est de trouver le modèle le plus simple et dont l'erreur est minimisée. Ceci nous a amené à nous intéresser aux méthodes de comparaison de modèles ce qui est l'objet du paragraphe suivant.

3.4.2 Comparaison de différents modèles.

Intuitivement, on va s'intéresser au modèle dont le résidu est le plus faible. Mais peut-on comparer les résidus de différents modèles? Cela n'est pas si simple.

Par exemple, supposons qu'on cherche à déterminer les paramètres de l'équation d'une courbe à partir de points. Les différents modèles vont être différenciés par le nombre de paramètres. Il est alors évident qu'on pourra toujours trouver une courbe qui passe par tous les points, y compris les points aberrants; cette courbe ne sera évidemment pas la solution qu'on cherche. De plus, s'agissant de points bruités, tout nouveau point de la courbe pris en considération ne sera pas sur la courbe décrite.

Une autre façon de voir les choses est de choisir le modèle qui comporte le moins de données aberrantes. Cela est contraire à notre approche qui voudrait pouvoir segmenter une scène en différents objets ayant des mouvements différents ou des structures géométriques différentes.

On peut alors introduire une autre idée et considérer la validité d'un modèle par rapport à sa capacité à prédire de nouvelles données. Ceci nous mène plus généralement à vouloir définir une prédictibilité d'un modèle. Pratiquement, cela va se traduire par la comparaison des différents résidus en faisant intervenir d'autres paramètres comme le nombre de données, le nombre de paramètres à estimer et le degré de liberté du modèle.

Différents auteurs se sont intéressés à ce problème comme Akaike, Kanatani, Hannan et bien d'autres encore.

3.4.2.1 Le Critère d'Akaike (AIC: *Akaike Information Criterion*)

Ce critère porte le nom de son auteur, Akaike (Akaike, 1972) qui est un précurseur dans le domaine de l'évaluation de critère d'information. Le critère d'Akaike peut être vu comme une approximation du critère d'entropie de Kullback-Leibler (Kullback, 1959).

Akaike a montré que le test du χ^2 est biaisé pour de faibles valeurs car le nombre de degrés de liberté du modèle intervient dans la prise en compte du bruit. Ce biais peut être compensé en ajoutant un terme correctif et le terme choisi par Akaike représente une pénalité fonction de la complexité du modèle. Notons dans la suite S le maximum de vraisemblance logarithmique, k le nombre de paramètres à estimer.

$$\text{AIC} = -2 \log(S) + 2k$$

On cherchera donc le modèle qui minimise ce critère, le MAICE (*Minimum AIC Estimate*), qui a pour propriété d'être le modèle le moins complexe qui contienne le plus d'information.

Cependant, ce critère ne sait pas tenir compte du nombre de données (donc du fait que plus les données sont nombreuses meilleur est le résultat) et même s'il fonctionne généralement bien, il ne sait pas traiter les données aberrantes. On est alors tenté d'effectuer une estimation robuste des paramètres puis d'appliquer le critère d'Akaike sur les données non aberrantes.

Un autre inconvénient est qu'on ne peut comparer entre eux que des modèles découlant les uns des autres (pour lesquels un modèle à $(k - 1)$ paramètres est obtenu à partir d'un modèle à k paramètres en fixant un paramètre). Ainsi, on pourra comparer le modèle avec translation et celui avec translation dans le plan rétinien mais pas le modèle projectif et rotation pure (nécessitant le calcul homographique) avec le modèle projectif et rotation additionnée de translation (passage par calcul de la matrice \mathbf{F}) car on a affaire à des calculs sur des variétés de dimensions différentes.

3.4.2.2 Justification du critère d'Akaike.

Suivant Akaike ((Kanatani, 1996) pour une récente discussion), nous considérons M vecteurs de mesure $\bar{\mathbf{x}}_i \in \mathbb{R}^p$ dont nous possédons des valeurs $\mathbf{x}_i \in \mathbb{R}^p$. Chaque mesure vérifie q contraintes $f(\mathbf{x}_i, \mathbf{c}) = 0 \in \mathbb{R}^q$ par rapport à un vecteur de paramètres $\mathbf{c} \in \mathbb{R}^k$. Nous supposons que les mesures sont bruitées par un bruit gaussien additif $\mathbf{x}_i = \bar{\mathbf{x}}_i + \nu_i$ de variance $E[\nu_i \nu_i^T] = \sigma_i^2 \mathbf{I}$. Nous cherchons alors à estimer \mathbf{c} et des mesures $\hat{\mathbf{x}}_i \in \mathbb{R}^p$ optimales et compatibles par rapport à \mathbf{c} , c'est-à-dire minimiser $J = \sum \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_{\Lambda^{-1}}^2$ sous les contraintes $f(\hat{\mathbf{x}}_i, \mathbf{c}) = 0$ où la distance est celle de Mahalanobis : $\|\mathbf{b} - \mathbf{a}\|_{\Lambda^{-1}}^2 = (\mathbf{b} - \mathbf{a})^T \Lambda^{-1} (\mathbf{b} - \mathbf{a})$.

Pour tester la prédictabilité du modèle, on introduit de nouveaux vecteurs "virtuels" $\mathbf{x}_i^* \in \mathbb{R}^p$ ayant la même distribution que les données initiales utilisées pour estimer le modèle (voir figure 3.2).

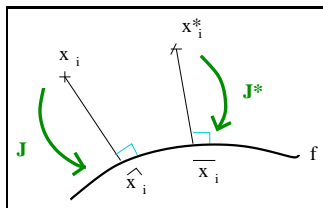


FIG. 3.2 – Notations pour la justification du critère d'Akaike

Nous considérons alors comme critère de performance un estimateur non biaisé de la distance de Mahalanobis entre les deux estimées :

$$J^* = \sum_i \|\mathbf{x}_i^* - \bar{\mathbf{x}}_i\|^2$$

Une estimation non biaisée de ce critère est donnée par : $J^* = J + E[J^*] - E[J]$, cette formule permettant d'évaluer J^* à partir des estimations.

Or, par une généralisation du théorème de Pythagore (voir figure 3.2), $\hat{\mathbf{x}}_i$ étant la projection de \mathbf{x}_i sur la variété de f , donc orthogonale par Λ à $(\bar{\mathbf{x}}_i - \hat{\mathbf{x}}_i)$ (de même pour $\bar{\mathbf{x}}_i$, projeté de \mathbf{x}_i^* :

$$\begin{aligned}\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|_{\Lambda^{-1}}^2 &= \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_{\Lambda^{-1}}^2 + \|\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\|_{\Lambda^{-1}}^2 \\ \|\mathbf{x}_i^* - \hat{\mathbf{x}}_i\|_{\Lambda^{-1}}^2 &= \|\mathbf{x}_i^* - \bar{\mathbf{x}}_i\|_{\Lambda^{-1}}^2 + \|\bar{\mathbf{x}}_i - \hat{\mathbf{x}}_i\|_{\Lambda^{-1}}^2\end{aligned}$$

L'estimée des carrés des normes correspondant au nombre de degrés de liberté. La différence $\mathbf{m}_i - \bar{\mathbf{m}}_i = \nu_i$ possède autant de degrés de liberté que de composantes, soit p . En considérant M mesures, on obtient $M p$. Il en est de même pour $\mathbf{m}_i^* - \bar{\mathbf{m}}_i$. Les quantités $\hat{\mathbf{m}}_i - \bar{\mathbf{m}}_i$, qui dépendent d'un vecteur de paramètre \mathbf{c} , possèdent chacune p composantes soumises à q contraintes ce qui donne, pour M mesures, $M(p - q)$ degrés de liberté. Il faut cependant également ajouter k degrés correspondant aux paramètres des contraintes, soit $M(p - q) + k$. On obtient alors :

$$\begin{aligned}E[J^*] &= \underbrace{\sum_i E[\|\mathbf{x}_i^* - \bar{\mathbf{x}}_i\|_{\Lambda^{-1}}^2]}_{M p} + \underbrace{\sum_i E[\|\bar{\mathbf{x}}_i - \hat{\mathbf{x}}_i\|_{\Lambda^{-1}}^2]}_{M(p-q)+k} \\ E[J] &= \underbrace{\sum_i E[\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|_{\Lambda^{-1}}^2]}_{M p} - \underbrace{\sum_i E[\|\bar{\mathbf{x}}_i - \hat{\mathbf{x}}_i\|_{\Lambda^{-1}}^2]}_{M(p-q)+k}\end{aligned}$$

ainsi que :

$$\begin{aligned}J^* &= J + M p + M(p - q) + k - (M p - M(p - q) + k) \\ &= J + 2 M(p - q) + 2 k\end{aligned}$$

Le cas historique d'Akaike consiste à prendre $p = q$. On obtient alors :

$$J^* = J + 2 k$$

Il faut noter qu'on exprime souvent J sous la forme $-2 \log(S)$.

3.4.2.3 Akaike et ses variantes

De nombreuses variantes existent; citons ici les plus rencontrées.

Le critère AIC a tendance à sélectionner des modèles ayant un trop grand nombre de paramètres, c'est pourquoi on peut lui préférer le critère BIC (*Bayesian Information Criterion*). Celui-ci provient de l'analyse Bayésienne (voir (Akaike, 1977), (Rissanen, 1978) et (Schwarz, 1978) pour plus de détails) et s'exprime sous la forme :

$$\text{BIC} = -2 \log(S) + k \frac{\log(N)}{N}$$

Mais ce dernier possède alors l'inconvénient de choisir des modèles ayant un nombre trop faible de paramètres. Cet inconvénient a été corrigé par la méthode ICL (*Integrated Classification Likelihood*)

(Biernacki et al., 1998). Cette méthode possède également l'avantage de proposer une justification théorique approfondie.

Hannan (Hannan et Quinn, 1979) a développé un critère φ dans le cadre d'analyse de modèles ARMA (*Auto-Regressive Mobile Average*) (Hannan, 1980) :

$$\varphi(k) = -\log(S) + k c \log\left(\frac{\log(N)}{N}\right) \text{ with } c > 2$$

qui a pour avantage que le second terme décroît très vite lorsque l'on veut contraindre k .

Kanatani a apporté une modification au critère d'Akaike dans (Kanatani, 1998) (GIC: *Geometric Information Criterion*) en considérant un nombre fini N de données, en faisant intervenir le nombre de paramètres k du modèle quand le nombre de données n'est pas trop élevé, ainsi qu'en tenant compte également de la dimension d de la variété du modèle.

$$\text{GIC} = -2 \log(S) + 2(dN + k) \tag{3.2}$$

Torr fournit une explication intuitive intéressante de ce critère dans (Torr, 1998) et, sachant que le GIC a tendance à sous-estimer la dimension de la variété du modèle et sur-estimer le nombre de paramètres, propose également une nouvelle généralisation :

$$\text{GIC} = -2 \log(S) + \lambda_1 dN + \lambda_2 k \text{ avec } 1 \leq \lambda_1 \leq 2 \text{ et } \lambda_2 > 2$$

D'autres variantes sont également présentées dans (Torr, 1998) et diffèrent les unes des autres par leur second membre.

L'inconvénient majeur de toutes ces méthodes est qu'aucune n'est robuste.

3.4.2.4 Un critère robuste, le GRIC: *Geometric Robust Information Criterion*

Ronchetti (Hampel et al., 1986) a proposé de modifier le critère d'Akaike en un critère robuste, c'est-à-dire prenant en compte les données aberrantes, qui prend la forme :

$$\text{AICR} = \sum_i \rho(e_i^2) + \alpha k$$

où e_i est le résidu de la donnée i . Ce critère dépend d'une fonction ρ dont le rôle est d'apporter de la robustesse et d'un coefficient de pondération α . Si l'on suppose que le bruit est gaussien pour les données de faible résidu et de distribution uniforme pour les données de fort résidu, ρ peut être définie, par analogie à la fonction d'Huber (voir équation 3.1) :

$$\rho(e_i^2) = \left\{ \begin{array}{ll} \frac{e_i^2}{\sigma^2} & \frac{e_i^2}{\sigma^2} < \lambda_3 (p - d) \\ \lambda_3 (p - d) & \frac{e_i^2}{\sigma^2} \geq \lambda_3 (p - d) \end{array} \right\} = \min\left(\frac{e_i^2}{\sigma^2}, \lambda_3 (p - d)\right)$$

Ici, p est toujours la dimension des données et $(p - d)$ la codimension de la variété du modèle. Torr choisit arbitrairement $\lambda_3 = 2$ sachant que sa valeur ne change pas grand chose expérimentalement. Cette fonction possède l'avantage de clairement isoler les données aberrantes des autres données. De même, Akaike et Torr choisissent $\alpha = 2$.

Avec un premier terme robuste issu de ce critère, Torr (Torr, 1997) l'a agrémenté du second terme généralisé du GIC pour donner lieu à un critère cumulant les deux avantages : la robustesse et la possibilité de comparer différents modèles de différentes dimensions entre eux. Il s'agit du GRIC (*Geometric Robust Information Criterion*) :

$$\text{GRIC} = \sum \rho(e_i^2) + \lambda_1 d N + \lambda_2 k \text{ avec } 1 \leq \lambda_1 \leq 2 \text{ et } \lambda_2 > 2 \quad (3.3)$$

prenant en compte l'erreur, la dimension et le nombre de paramètres du modèle. On considère dans ce critère que l'erreur de déviation standard σ , nécessaire au calcul, est fournie par le détecteur de points d'intérêts.

Toujours dans (Torr, 1997), Torr propose des variantes du GRIC en extension de (Schwarz, 1978) sous la forme de GRIC2 et GRIC3 :

$$\begin{aligned} \text{GRIC2} &= \sum \rho(e_i^2) + 2 d N + 2 k \log(N) \\ \text{GRIC3} &= \sum \rho(e_i^2) + 2 d N + 2 k \log(n) + \log(|\Sigma|) \end{aligned}$$

mais montre que c'est le GRIC1 (équation 3.3) qui fournit les meilleurs résultats.

A ce stade, il nous manque encore la possibilité de comparer des modèles de variétés de dimensions différentes. Torr propose un critère obtenu en développant les probabilités à posteriori de chaque donnée pour le modèle donné. Mais la complexité de la minimisation du critère ainsi obtenu fait qu'il utilise plutôt un algorithme de type tirage aléatoire d'échantillons (*random sampling*). Le lecteur intéressé trouvera tous les détails dans (Torr, 1998).

3.5 Une autre alternative : Les algorithmes génétiques.

Dans le cadre de cette thèse, nous avons exposé notre problème d'estimation à une autre équipe de recherche de l'INRIA Sophia Antipolis : le projet SINUS (SIMulation NUMérique pour les Sciences de l'ingénieur) afin d'estimer si les algorithmes génétiques pouvaient résoudre notre problème.

3.5.1 Présentation du problème : résumé

Notre problème est le choix du cas particulier le plus adapté à un cas expérimental donné. Une première solution consisterait à calculer dans chaque cas le résidu correspondant et de comparer par la suite les résidus. Or, nous disposons d'un nombre de cas très élevé (environ 2^{17} pour deux vues).

« Diane, 2^{17} ? Mais c'est énorme ! »

Rachid Nour-Eddine Deriche, voyage Nice-Graz, 1996

On va donc examiner en quoi les algorithmes génétiques peuvent nous aider.

3.5.2 Présentation des algorithmes génétiques

Les algorithmes génétiques (AGs) sont des algorithmes stochastiques permettant de résoudre des problèmes NP-complets. Ils sont inspirés des mécanismes de la sélection naturelle et de la génétique; ils utilisent à la fois les principes de la survie des individus les mieux adaptés et ceux de la propagation du patrimoine génétique. Ils furent développés par Holland (Holland, 1975) et son équipe à l'Université du Michigan vers les années 60 et repris par Goldberg (Goldberg, 1991). Les algorithmes génétiques présentent 4 particularités principales par rapport aux algorithmes dits classiques :

- utilisation d'un codage de paramètres et non des paramètres eux-mêmes
- analyse simultanée d'un ensemble de points et non pas point par point
- seule la valeur de la fonction en ces points est utilisée (pas d'utilisation de dérivée ou autre grandeur)
- le choix des points se fait de façon stochastique et non pas déterministe

Précisons un peu de vocabulaire avant de poursuivre : une chaîne ou *chromosome* est constituée d'éléments ou *gènes* possédant une valeur ou *allèle* (binaire, décimale, hexadécimale, ...) localisés en des sites ou *locus*. Ces chaînes forment une population dont la génération est une image à un instant donné.

Un algorithme génétique manipule des chaînes contenant le codage des paramètres à l'aide de trois opérateurs afin de maximiser la fonction d'adaptation ou *fitness* :

opérateur de reproduction : une chaîne est dupliquée proportionnellement à son adéquation au problème, la proportion pouvant être exacte ou proportionnelle.

opérateur de croisement : cet opérateur prend 2 chaînes en entrée pour fournir deux nouvelles chaînes, principalement selon deux types de croisement :

(1) **croisement par rapport à un site :** un site est tiré aléatoirement et les fins de chaînes à partir de ce site sont échangées

(2) **croisement uniforme :** la première chaîne est construite en tirant aléatoirement entre les deux chaînes pour chaque site; la seconde chaîne est complémentaire de la première.

opérateur de mutation : des valeurs en des sites tirés aléatoirement sont modifiées. Cet opérateur, utilisé en faible proportion, permet de palier à la perte ou non-présence de certaines informations.

Ces mécanismes sont illustrés sur la figure 3.3.

Le principe de l'algorithme est le suivant :

population initiale
faire
 reproduction
 croisement
 mutation
évaluation de la population

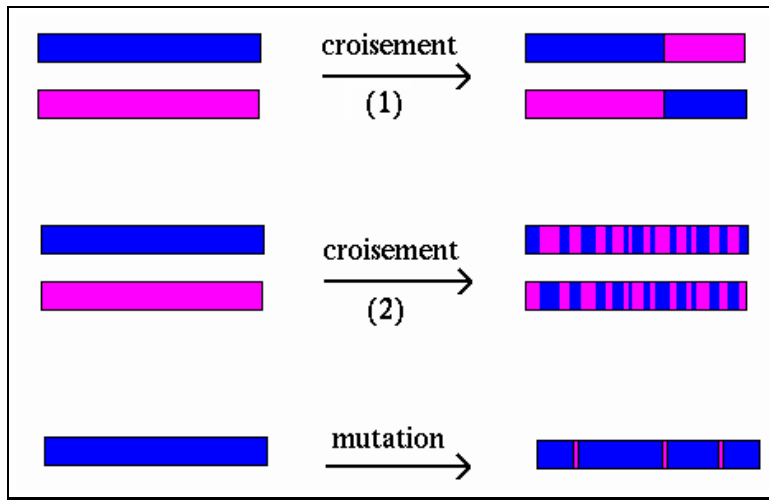


FIG. 3.3 – Opérateurs de croisement et de mutation

tant que (taille de la population < nombre max.) et (fitness)

3.5.3 Adaptation à notre problème

Nos cas particuliers sont des combinaisons de cas particuliers atomiques et s'expriment, comme expliqué au chapitre 2, selon la forme suivante :

$p[1-3]g[1-4]f[1-3]s[1-3]b[1-4]B[1-4]c[1-3]R[1-5]a[0-2]u[1-24]W[1-3]T[1-2]t[1-12]D[1-3]Z[1-3]d[1-3]N[1-13]l[1-3]m[1-3]$

Nous allons ainsi considérer des chromosomes composés des gènes p , g , f , s , b , B , c , R , a , u , W , T , t , D , Z , d , N , l et m dont les allèles sont des entiers situés dans les intervalles donnés ci-dessus. La figure figure 3.4 illustre la composition d'un tel chromosome constitué de quatre sous-parties correspondant au type de projection, aux paramètres intrinsèques et extrinsèques ainsi qu'à la géométrie.

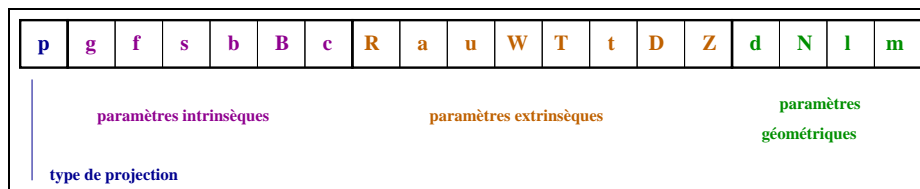


FIG. 3.4 – Description d'une chaîne ou chromosome correspondant à notre problème.

On considère l'arbre de recherche comme étant le produit cartésien des arbres de chaque famille (notée précédemment par une lettre et définie au chapitre 2). L'intérêt des algorithmes génétiques est de nous permettre de sauter aléatoirement d'une branche à l'autre, c'est-à-dire de ne pas parcourir l'arbre dans son entier mais seulement en partie.

Le point difficile des algorithmes génétiques concerne la convergence qui dépend évidemment des opérateurs de croisement et de mutation choisis. Dans notre cas, il va falloir définir ces opérateurs en fonction des parties des chromosomes qu'ils affectent. Une première idée consiste à définir plusieurs ensembles d'opérateurs de croisement, correspondant à des croisements sur les sous-chromosomes (paramètres intrinsèques, extrinsèques et géométriques). Mais c'est une étude bien plus vaste qui est nécessaire pour aborder ce problème.

Ainsi, nous nous sommes arrêté à une phase de réflexion et avons décidé, dans un premier temps, de trouver une autre solution pour échapper à la combinatoire trop importante de cet arbre de recherche.

3.6 Approche bayésienne

La loi de Bayes s'exprime de la façon suivante, où \mathbf{x} représente une estimée des paramètres et \mathbf{y} un vecteur de mesures :

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x}) p(\mathbf{x})}{p(\mathbf{y})}$$

Dans cette expression, $p(\mathbf{x})$ est un terme confiance aux paramètres tandis que $p(\mathbf{y})$ ne dépend pas des paramètres mais peut être estimé dans notre cas. Généralement, chercher le maximum de $p(\mathbf{x}|\mathbf{y})$ en faisant varier \mathbf{x} revient donc à maximiser la fonction de vraisemblance $p(\mathbf{y}|\mathbf{x})$.

L'approche bayésienne pour la sélection de modèle évalue la vraisemblance à posteriori pour chaque modèle basée sur l'intégrale du produit de la vraisemblance avec le prior. Sachant que les cas particuliers étudiés dans cette thèse ont une signification physique de projection, de mouvement ou de géométrie, nous pouvons assigner une distribution à priori pour les paramètres (par exemple, généralement, le centre optique est situé vers le centre de l'image, la translation est nulle ou presque, l'angle de rotation est faible ...) (Torr, 1999).

C'est une voie sûrement prometteuse qu'il faudra étudier à l'avenir. Le lecteur désireux de plus de détails pourra se référer à (Torr, 1999), (Bolle et Cooper, 1984), (Bolle et Cooper, 1985), (Sorenson, 1900) et (Chang et Chatterjee, 1990).

3.7 Calcul de l'erreur dans le cadre de notre problème de vision

Pour appliquer les méthodes précédentes, il nous faut encore mesurer une erreur. Nous allons examiner les différentes possibilités.

3.7.1 Erreur algébrique et erreur géométrique

Hartley (Hartley, 1998) note $\mathbf{m}_i = [u_i \ v_i \ w_i]^T$ un point 2D de l'image et $\hat{\mathbf{m}}_i = [\hat{u}_i \ \hat{v}_i \ \hat{w}_i]^T$ le point 2D estimé.

Il définit l'erreur algébrique par :

$$d_{\text{alg}}(\mathbf{m}_i, \hat{\mathbf{m}}_i) = \sqrt{((\hat{w}_i v_i - w_i \hat{v}_i)^2 + (\hat{w}_i u_i - w_i \hat{u}_i)^2)} = \|\mathbf{M} \mathbf{p}\|^2 = \|\varepsilon\|^2$$

c'est-à-dire euclidienne dans l'image et donnée en pixels, et l'erreur géométrique par :

$$d_{\text{géom}}(\mathbf{m}_i, \hat{\mathbf{m}}_i) = \sqrt{\left(\frac{u_i}{w_i} - \frac{\hat{u}_i}{\hat{w}_i}\right)^2 + \left(\frac{v_i}{w_i} - \frac{\hat{v}_i}{\hat{w}_i}\right)^2}$$

reliée à la précédente par la relation suivante :

$$d_{\text{géom}}(\mathbf{m}_i, \hat{\mathbf{m}}_i) = \frac{d_{\text{alg}}(\mathbf{m}_i, \hat{\mathbf{m}}_i)}{w_i \hat{w}_i}$$

En général, l'erreur géométrique est préférable à l'erreur algébrique car elle prend mieux en compte les contraintes physiques de la mesure dans l'image. Cependant, Hartley, toujours dans (Hartley, 1998), montre que l'erreur algébrique fournit de très bons résultats si les données sont correctement normalisées et propose un algorithme où les contraintes physiques sont prises en compte.

3.7.2 Calcul d'un résidu du problème

En supposant qu'on est dans un cas où le calcul de la matrice fondamentale est possible (où celle-ci n'est pas théoriquement nulle), les principaux critères utilisés dans la littérature (Zhang et al., 1995; Zhang, 1998b) sont :

1. distances des points aux lignes épipolaires (1 coordonnée, distance orthogonale)
2. erreurs épipolaires pondérées par le gradient
3. distances des points aux retroprojections

3.7.2.1 Distances aux droites épipolaires.

Comme expliqué par Zhang dans (Zhang et al., 1995), et utilisé par de nombreux auteurs comme par exemple (Luong et al., 1993), un critère efficient est la moyenne des distances euclidiennes entre chaque point et sa droite épipolaire correspondante. La somme des moindres carrés $\min_{\mathbf{F}} \varepsilon_{\mathbf{F}}^2(\mathbf{F})$ est minimisée :

$$\varepsilon_{\mathbf{F}}^2(\mathbf{F}) = \frac{\left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \underbrace{[d(\mathbf{m}', \mathbf{F} \mathbf{m})^2 + d(\mathbf{m}, \mathbf{F}^T \mathbf{m}')^2]}_{f_{\mathbf{m}}(\mathbf{F})^2} \right]}{\left[2 \sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \right]} \quad (3.4)$$

et $w_{\mathbf{m}}$ est un poids correspondant à la précision de l'appariement, en réalité, l'inverse de la variance de la précision de l'appariement. La quantité $w_{\mathbf{m}}$ est donnée en pixel^{-2} , tandis que $\varepsilon_{\mathbf{F}}(\mathbf{F})$, la *distance moyenne à l'épipole*, est en pixel.

La distance d'un point \mathbf{m} (resp. \mathbf{m}') à sa droite épipolaire $\mathbf{F}^T \mathbf{m}'$ (resp. $\mathbf{F} \mathbf{m}$) est donnée par :

$$d(\mathbf{m}', \mathbf{F} \mathbf{m}) = \frac{|\mathbf{m}'^T \mathbf{F} \mathbf{m}|}{\sqrt{(\mathbf{F} \mathbf{m})_0^2 + (\mathbf{F} \mathbf{m})_1^2}}$$

$$\left(\text{resp. } d(\mathbf{m}, \mathbf{F}^T \mathbf{m}') = \frac{|\mathbf{m}^T \mathbf{F}^T \mathbf{m}'|}{\sqrt{(\mathbf{F}^T \mathbf{m}')_1^2 + (\mathbf{F}^T \mathbf{m}')_2^2}} \right)$$

3.7.3 Erreurs épipolaires pondérées par le gradient

Ce critère se base sur le fait qu'en présence de bruit, l'équation fondamentale $\mathbf{m}'^T \mathbf{F} \mathbf{m} = 0$ n'est pas vérifiée. L'erreur est donc :

$$f_m(\mathbf{F}) = \frac{|\mathbf{m}'^T \mathbf{F} \mathbf{m}|}{\sqrt{(\mathbf{F}^T \mathbf{m}')_1^2 + (\mathbf{F}^T \mathbf{m}')_2^2}}$$

3.7.4 Distances des points aux retroprojections

Ce critère nécessite le calcul des points 3D projectifs et leurs reprojections, donc la détermination des matrices de projection, ce qui n'est pas possible dans notre cas.

3.7.5 Comparaison des différents critères

Zhengyou Zhang note \mathcal{J}_1 le critère basé sur les distances des points aux lignes épipolaires, \mathcal{J}_2 , le critère basé sur l'erreur du gradient pondéré et \mathcal{J}_3 le critère basé sur la distance des points aux reprojections. Statistiquement, \mathcal{J}_3 est préférable aux autres critères mais Zhengyou (Zhang, 1998b) a démontré que les critères 2 et 3 sont équivalents si les épipoles sont à l'infini (géométrie standard) et sont très proches même si les épipoles sont dans l'image. Sachant de plus que le critère 3 prend beaucoup de temps, on recommande l'usage du critère 2. Les critères 1 et 2 sont théoriquement équivalents au niveau des paramètres estimés si les épipoles sont à l'infini et que les objets ont même facteur d'échelle dans les deux images. Sinon, si les épipoles sont dans les images, la différence entre ces deux critères reste tout de même très faible en pratique (voir également (Zhang, 1997a) et (Zhang, 1998a)).

Dans notre cas, les images sont suffisamment proches pour que même lors d'un mouvement d'approche ou de zoom d'un objet, les hypothèses d'équivalences des critères soient vérifiées.

Torr (Torr et Murray, 1997) présente une autre classification :

OR : Moindres carrés orthogonaux

S1 : Repondération itérative (Weng et al., 1989) par l'inverse du gradient (adaptation de la méthode de Sampson).

S2 : Repondération itérative (Luong et Faugeras, 1993) utilisant les distances des points aux droites épipolaires.

En terme de probabilité, S1 est une approximation du χ^2 au premier ordre tandis que S2 est un χ^2 .

3.7.6 Autres cas

Ce paragraphe concerne les cas où il n'est pas possible de calculer la matrice fondamentale car celle-ci est théoriquement nulle. On se ramène alors à d'autres grandeurs telle l'homographie ou bien les épipoles.

Couramment, dans le cas de l'homographie, la distance considérée est la distance géométrique entre \mathbf{m}_2 et $\mathbf{H} \mathbf{m}_1$, \mathbf{m}_1 et \mathbf{m}_2 étant normalisés :

$$\left\| \mathbf{m}_2 - \frac{\mathbf{H} \mathbf{m}_1}{((\mathbf{h}^2)^T \mathbf{m}_1)} \right\|^2$$

où \mathbf{h}^2 représente la dernière ligne de la matrice d'homographie \mathbf{H} .

3.8 Conclusion de cette étude : notre choix

Dans les expérimentations présentées au paragraphe 4.2.2 le mécanisme suivant a été utilisé :

pour chaque modèle étudié :	
	Minimisation sur l'ensemble des données par la méthode de la ν -iane (3.3.6) afin d'éliminer les données aberrantes. Lorsque la scène ou la caméra possède un mouvement rigide, une valeur de 50% pour ν est suffisante. Il faut faire varier cette valeur dans le cas où la scène est rigide par morceaux. Les erreurs sont calculées en utilisant les distances des points aux droites épipolaires pour la matrice fondamentale ou les distances géométriques du paragraphe précédent pour l'homographie.
	Raffinement du résultat par minimisation non linéaire sur les données non aberrantes. La méthode de minimisation utilisée est celle de la librairie MINPACK selon l'algorithme de Levenberg-Marquardt (More, 1977).
	Calcul de l'erreur du modèle selon le critère d'Akaike modifié par Kanatani et présenté par l'équation (3.2), en utilisant la distance des points aux droites épipolaires.
comparaison des résidus ainsi obtenus.	

Il serait intéressant par la suite d'utiliser comme critère, le critère GRIC de Torr, équation (3.3) sachant que l'étape de minimisation n'élimine pas forcément tous les points aberrants et de comparer alors les résultats obtenus avec les critères GIC et GRIC. En effet, il n'a pas été tenu compte du fait que l'on compare ici des résidus obtenus sur des données différentes car les points aberrants d'un modèle ne sont pas forcément aberrants pour un autre modèle.

Une autre amélioration de nos résultats pourrait être obtenue en utilisant le critère de Torr permettant de comparer des résidus de modèles provenant de branches différentes dans la hiérarchie des cas particuliers.

Enfin, il serait également intéressant d'étudier de façon plus approfondie l'approche bayésienne ainsi que l'approche des algorithmes génétiques. Ces approches feront l'objet d'une étude ultérieure.

Chapitre 4

Expérimentations et applications

On a construit de belles théories avec de belles équations mais arrive maintenant l'instant cruel de la confrontation aux données réelles.

Nous allons donc, dans ce chapitre, valider les idées de cette thèse par plusieurs expérimentations sur une plate-forme robotique Argès et évoquerons alors différentes applications.

« Pour un stage de DEA, il faut que ça marche avec une image.

Pour une thèse, il faut que ça marche avec deux images. »

Mike Brady

Experimentations (english abstract)

Here, we present some experimental results.

First, we experimentally demonstrate that when comparing several models with the data, the model with fewer parameters gives the best results.

Then, we also demonstrate that given the data, we are able to retrieve which cases of displacement, projection parameters or scene geometry to which they correspond.

Thus, we have focused on one particular case to experiment an active vision algorithm on a robotic system called Argès: we use the residual error to converge to a displacement for which the equations allow us to do a 3D reconstruction.

Finally, we finally present some applications using hierarchical reconstruction and video sequence segmentation.

4.1 Présentation de la plate-forme expérimentale Argès.

4.1.1 Présentation du système robotique Argès.

Le système robotique Argès (figures 4.1 et 4.4) est une plate-forme expérimentale destinée à l'expérimentation d'algorithmes de vision active comme (Viéville et al., 1993; Viéville et al., 1995a; Viéville et al., 1996b; Lingrand et Viéville, 1996; Enciso et al., 1996a; Enciso et Viéville, 1995).



FIG. 4.1 – Photo du système robotique Argès

Ce système est composé de :

- une caméra couleur CCD Acom1 PAL** de Imagic dont le zoom, le focus et l'iris sont télécommandables via une liaison série et possède un mécanisme d'auto-focus. Les caractéristiques techniques de cette caméra sont données sur la figure 4.2.
- une tourelle RobotSoft** pouvant effectuer des mouvements de roulis et tangage ("pan" et "tilt") avec une résolution de 3.086 minutes d'arc, une capacité de 4 lbs et une vitesse pouvant aller jusqu'à 300 deg/sec, utilisant des moteurs à courants continus, via une interface rs232c
- un rail CharlyRobot** permettant des mouvements de translation d'un degré de liberté avec une résolution de 0.1 mm
- une carte d'acquisition et de traitement d'images SunVideo** basée sur la librairie micro-programmée "xil" de Sun permettant d'implémenter des algorithmes de vision temps-réel (Murcia et al., 1997)
- une commande électronique de pilotage** de un à trois moteurs pas à pas biphasés C242 Charlyrobot, équipée d'une carte de pilotage 3 axes dans laquelle on introduit les données par l'intermédiaire d'une liaison série rs232c provenant d'une station de travail.
- une carte ArtePort** de ArteconTM permettant à une station Sun de communiquer via trois liaisons séries rs232c.


 ACom1 Computer Controlled CCD-Camera	
Manuell	/ 3 Pin Klippen socket */ / by RS-232 model not used */
Camera Parameter:	
Output :	Y/C (Super-VHS) / FBAS (PAL)
ImageSensor :	1/3" IT CCD 752(H) x 582(V) Pixel
Objektiv:	motor. 8x Zoom f=5.9 bis 47.2mm F1.4
Lens:	9 elements in 6 Groups (incl. 2 asphärischen lenses)
Angle of view horiz:	44.3°(W-W) to 5.8°(Tele)
Sichtwinkel vertikal:	34.9°(W-W) to 4.4°(Tele)
Shortest Distance:	10 mm (W-W) / 900 mm (Tele)
Illumination Adjustment:	automatic AGC 18dB and motor. Iris F1.4 to >F22
Focus:	automatic with 10 bit DSP and high-speed motor
Video-Output Y/C:	Y: VS 1.0 V _{pp} Synch negativ 75Ω terminated C: Burst 0.3 V _{pp}
Video-Output FBAS:	VBS: 1.0 V _{pp} Composite 75Ω terminated
H-Resolution (centre):	>450 TV-lines
V-Resolution (centre):	>400 TV-lines
Min. Illumination:	6 lux F1.4
S/N-Ratio:	>46dB
Wight-Balance :	TTL-Auto tracing, One push Hold, In/Out door Preset
Electronic Shutter:	28 steps to 1/10000 sec.
Power Supply:	DC 7 bis 9V (3W)
Dimensions:	55x 57x 107 mm
FilterMount:	37 mm Sony
Working-Conditions:	0 to +50 °C / 30%-80%humidity
Storage-Conditions:	-10 to +60 °C / 20%-90%humidity

FIG. 4.2 – *Caractéristiques techniques de la caméra utilisée dans le système robotique Argès (d'après le constructeur, "I magic").*

La figure 4.3 présente le câblage de ce système. La caméra à zoom et focus télécommandables est montée sur une tourelle (roulis et tangage) elle-même montée sur un rail à un degré de liberté.

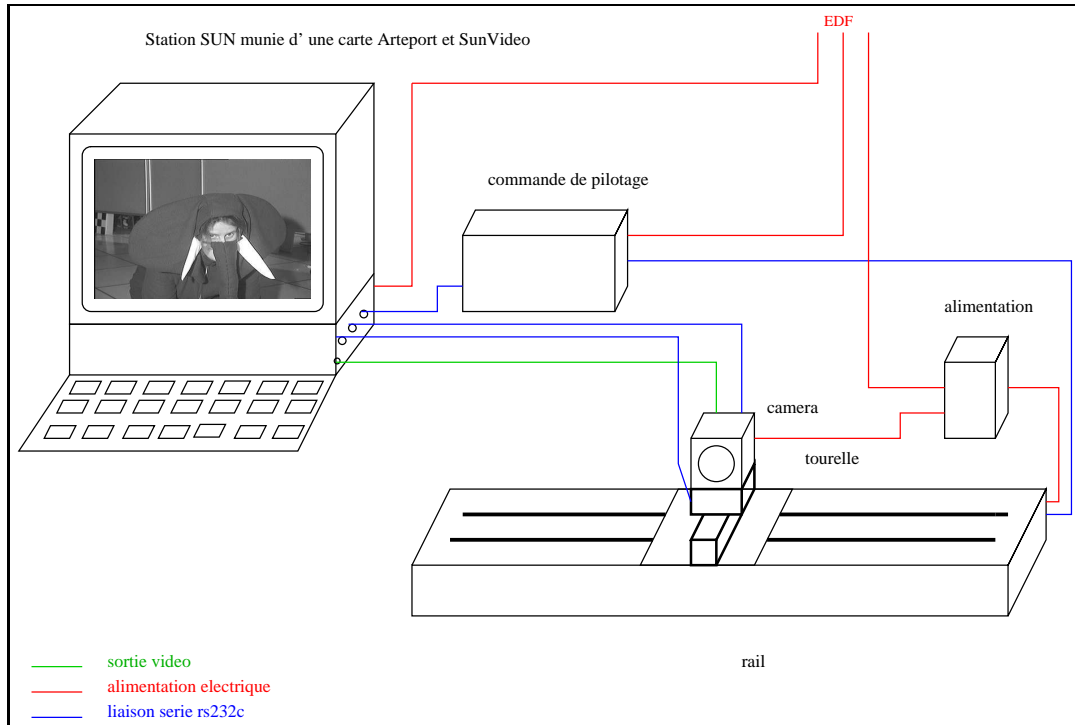


FIG. 4.3 – schéma de câblage des éléments constituant le système robotique Argès

Ce système a été conçu afin de pouvoir réaliser différents mouvements simples, résumés sur la figure 4.4 :

- un zoom
- une rotation pure en roulis ("pan")
- une rotation pure en tangage ("tilt")
- une translation pure, parallèle au plan rétinien ou parallèle à l'axe optique ou bien encore dans toute direction oblique

A ce système peut être couplée une table tournante de précision possédant deux degrés de liberté : un degré de liberté en rotation autour d'un axe vertical et un degré de liberté en translation le long de ce même axe. Ainsi, l'ajout de cette table porte le nombre de degrés de liberté à 3 en rotation et 2 en translation.

Un film montrant Argès en mouvement ainsi qu'une présentation générale de ce système est disponible sur Internet à l'adresse suivante :

<http://www-sop.inria.fr/robotvis/personnel/vthierry/Demos/Arges/>

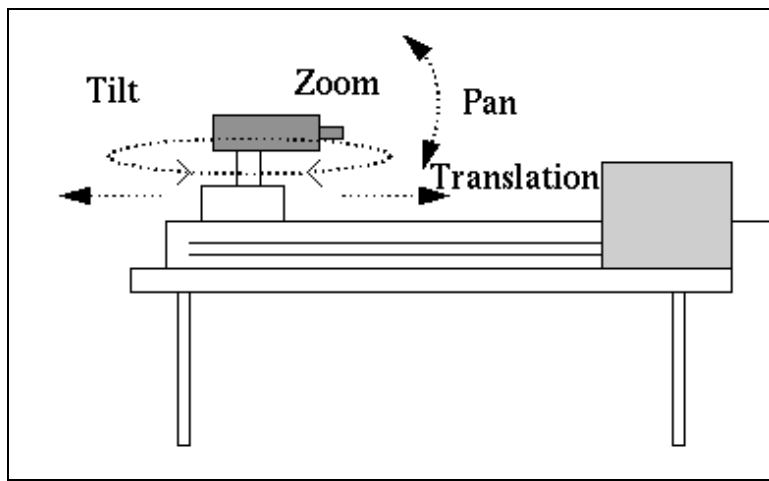


FIG. 4.4 – Schéma synthétique du système robotique Argès

4.1.2 Présentation de l'environnement logiciel Argès

Les drivers permettant de communiquer via les liaisons séries avec la tourelle, le rail et la caméra ont été codés par nos soins. Nous avons réalisé en Java une interface conviviale de communication avec le système. Cette interface (voir figure 4.5) permet non seulement d'effectuer des commandes sur le système mais présente aussi tout un ensemble de fonctionnalités sous forme d'un package Java allant de la vision bas-niveau (détection de points d'intérêts, appariements de ces éléments) à nos algorithmes récemment développés.

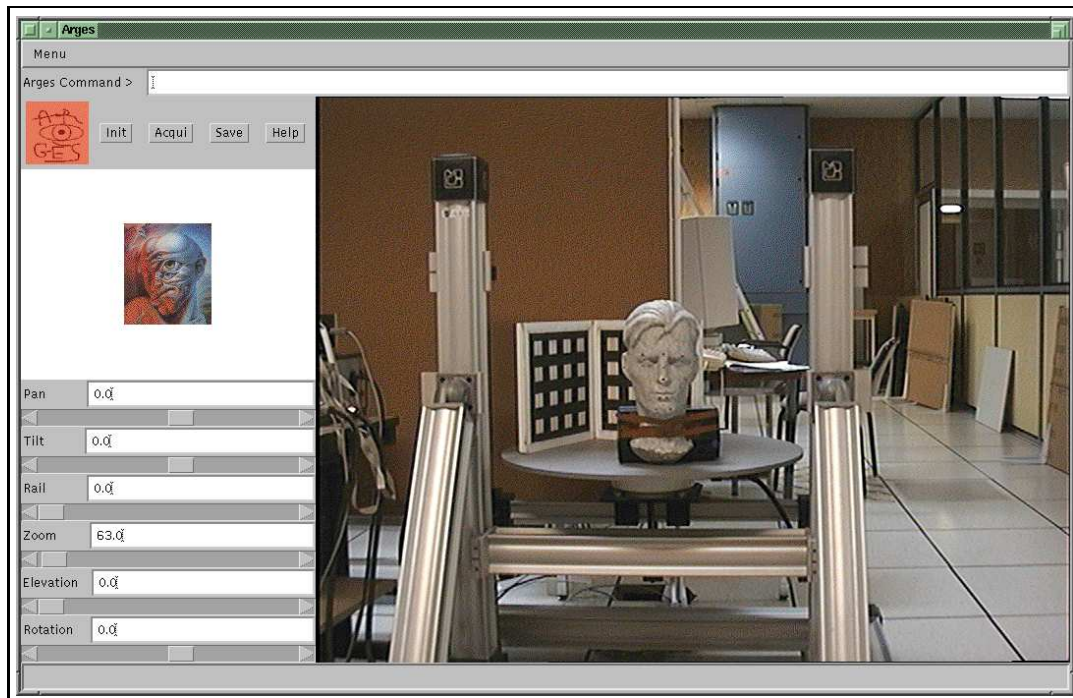


FIG. 4.5 – Interface du système robotique Argès

Pour plus de détails, on pourra se référer à la page suivante :

<http://www-sop.inria.fr/robotvis/personnel/vthierry/Demos/Arges/doc/>

4.2 Prototype d'expérimentations d'un échantillon de cas particuliers concernant deux images

Ce prototype concerne deux images consécutives ou non d'une séquence d'images pour lesquelles on a déjà effectué la détection puis l'appariement des points d'intérêt.

4.2.1 Présentation de l'organisation logicielle: implémentation multi-langages

Notre problème se décompose en plusieurs étapes de natures très différentes. Premièrement, la gestion des cas particuliers qui comporte le codage des cas atomiques, la construction des cas moléculaires, les éliminations des cas redondants et inutiles et la génération des équations simplifiées du modèle particulier. Pour cela, nous avons besoin d'un outil de calcul symbolique.

En deuxième étape viennent les calculs mettant en évidence l'adéquation ou non d'un modèle avec les données en simultané avec l'estimation des paramètres du modèle. Cette étape est englobée dans un algorithme chargé de l'ordre d'examen des cas particuliers. Nous avons alors besoin d'un langage permettant d'effectuer des calculs numériques lourds de façon rapide.

Reste en dernière étape l'interface utilisateur que l'on veut généralement le plus conviviale et éventuellement, fonctionner à travers un réseau.

Pour les trois étapes de natures différentes, nous avons choisi 3 langages adaptés.

Maple : pour l'aspect symbolique

langage C : pour les calculs numériques

Java : pour l'interface

4.2.1.1 Maple

Maple est un langage destiné aux mathématiques : calculs symboliques, calculs numériques et représentations graphiques. Ce langage a été développé conjointement par l'Université de Waterloo (Ontario, Canada) et l'Institut Fédéral de Technologie de Zürich (ETH, Suisse). La première version commerciale (3.3) est apparue en 1985.

L'aspect calcul symbolique de Maple nous intéresse particulièrement pour traiter les cas particuliers, d'autant plus que Maple nous permet ensuite de générer du code C. En supplément de Maple, nous utilisons une bibliothèque "Mascotte" (Viéville, 1996) adaptée aux problèmes de la vision et qui comporte, entre autres, une extension du package "linalg" ("linalg1") et une amélioration du code C généré.

4.2.1.2 Java

Ce langage a été conçu à l'origine par une équipe de Sun, dirigée par J. Gosling, afin d'avoir un langage qui puisse permettre de créer des logiciels qui s'adaptent à n'importe quel nouveau processeur dans le domaine du matériel électronique consommable grand public (télévision, toaster, ...). Ainsi naissait le langage connu initialement sous le nom de "Oak" au début des années 90. Mais le spectateur téléphage végétatif n'était pas près à une telle interactivité et Oak fut voué à l'échec. Or, on était alors en plein développement d'Internet dans lequel Gosling trouva une nouvelle application bien plus prometteuse de ses idées. Ainsi, Oak évolua pour permettre aux machines d'architectures différentes en présence sur le réseau d'utiliser un même langage : **Java**, dont la première version sortit en 1995.

Sun décrit Java comme un langage simple, orienté objet, distribué, interprété, robuste, sécurisé, indépendant de l'architecture, portable, hautement performant, multithread et dynamique.

L'interface de pilotage du système robotique Argès est écrite en Java et une bibliothèque en Java a été créée afin de répondre plus spécifiquement à nos problèmes, notamment en ce qui concerne la gestion des images et séquences d'images, ainsi que l'état d'un processus, par exemple Maple (en cours, terminé ou arrêté sur une erreur). Il a donc été naturel de prévoir l'intégration de notre prototype d'expérimentation au sein de l'environnement d'Argès.

4.2.1.3 Langage C

Le langage **C** a été écrit en 1971 et 1974 par Brian Kernighan et Dennis Ritchie au laboratoire Bell d'ATT. Ce langage se veut un langage général, indépendant du matériel (au niveau du code, mais doit être compilé pour chaque nouvelle architecture) mais suffisamment proche pour être performant (et éviter le recours à l'assembleur).

L'intérêt de son utilisation est triple :

- Maple sait générer du C
- Java sait intégrer du C via les "Native"
- les routines de minimisation des "Numerical Recipes" sont disponibles en C

4.2.1.4 Organisation

L'organisation est présentée sur le schéma de la figure 4.2.1.4 dont l'algorithme se décompose en les étapes suivantes :

- Choix des contraintes (par un utilisateur, voir automatiquement, de façon exhaustive).
- Test sur la compatibilité des contraintes. Si oui, continuer, si non, revenir au choix des contraintes.
- Détermination des paramètres libres et liés - Mise en équations réduites.
- Minimisation à l'aide des données du critère d'adéquation en faisant varier les paramètres libres.

L'interface utilisateur est présentée sur la figure 4.2.1.4.

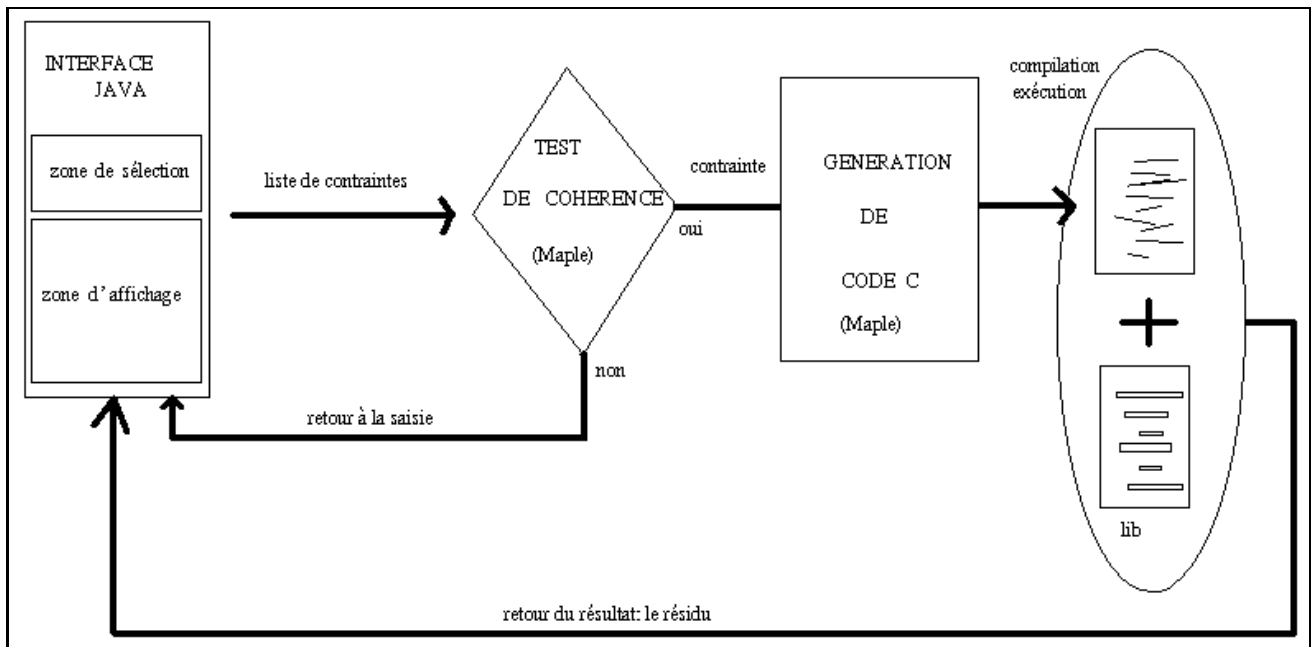


FIG. 4.6 – Organisation logicielle

4.2.2 Expérimentations

Nous présentons ici des résultats concernant deux cas particuliers ainsi qu'un ensemble de plusieurs déterminations automatiques de mouvement singulier.

4.2.2.1 Expérimentation pour un mouvement de translation pure

Voici les résultats correspondant à un mouvement réel de translation pure sur le rail effectué par Argès. On compare les résidus obtenus pour plusieurs séries d'hypothèses sur le mouvement.

La figure 4.8 montre les images prises par Argès, la figure 4.9 les points détectés et la figure 4.10 les vecteurs de disparité entre les deux images.

cas particulier	résidu
rotation nulle, zoom fixe, γ nul	9.03948
translation en x pure	4.9927
translation en z pure	530.13
rotation en x, translation en z, γ nul	37.8845
zoom et translation purs	11.0012

Ces résultats confirment 2 points:

- *le meilleur résultat correspond au mouvement effectué* : cette méthode permet donc de retrouver quel type de mouvement a été effectué,

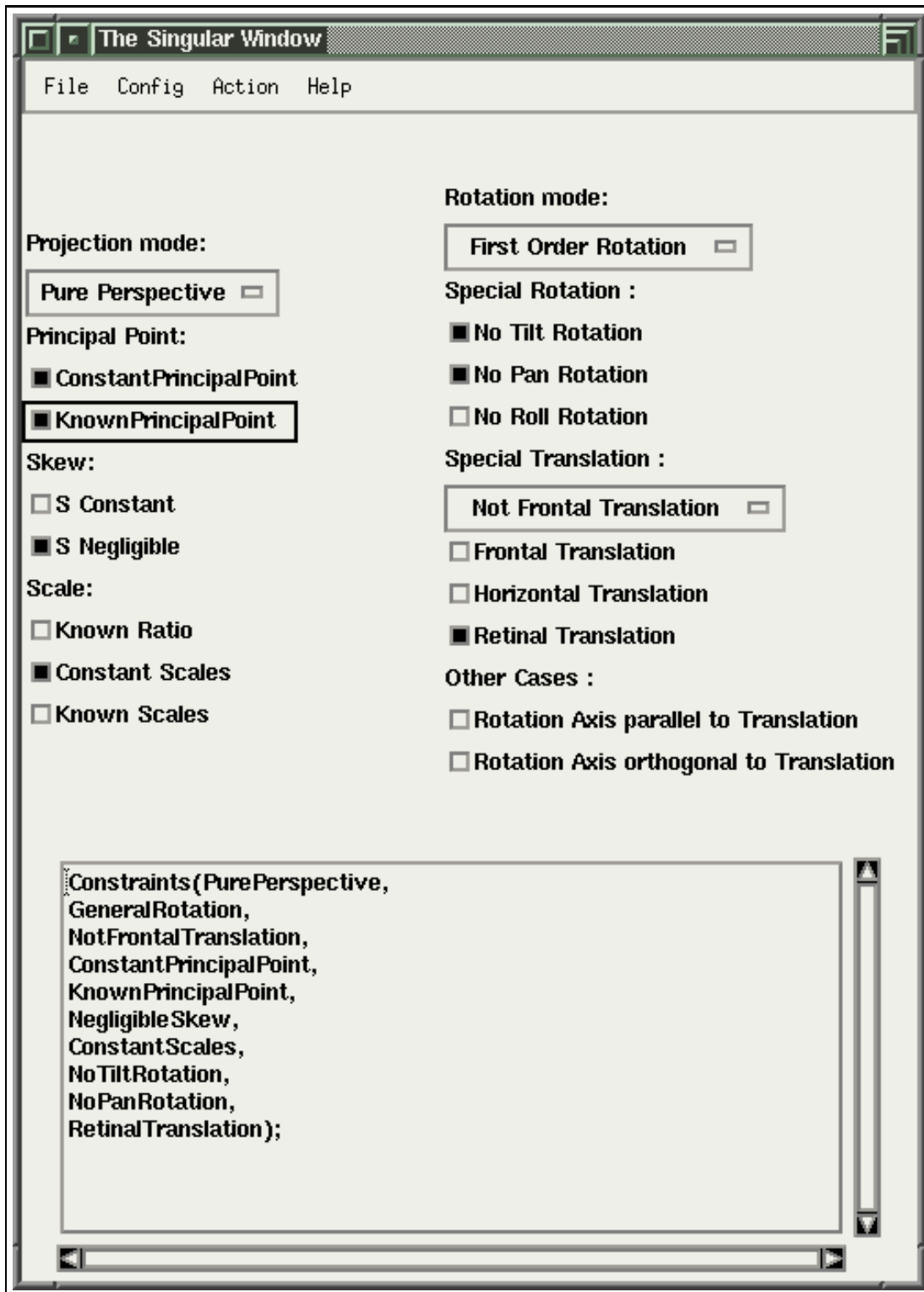


FIG. 4.7 – Interface du module de test de cas particuliers.

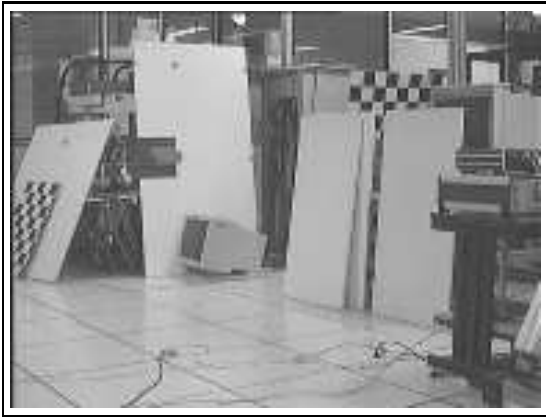


FIG. 4.8 – images prises par Argès lors d'un mouvement de translation pure

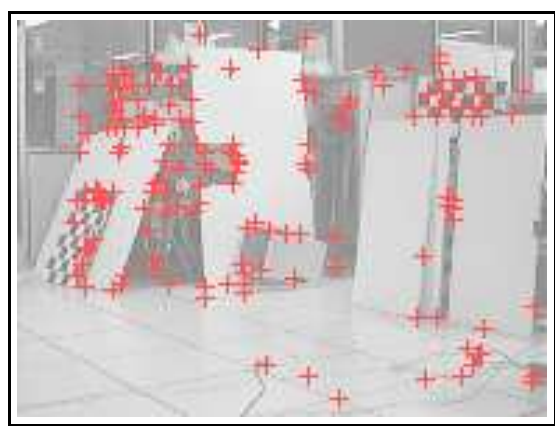
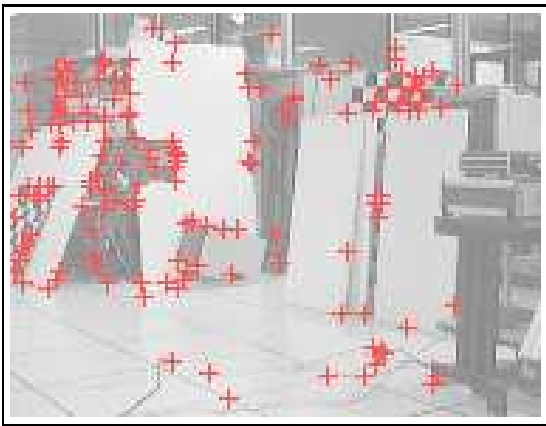


FIG. 4.9 – 154 points ont été détectés et appariés entre les 2 images

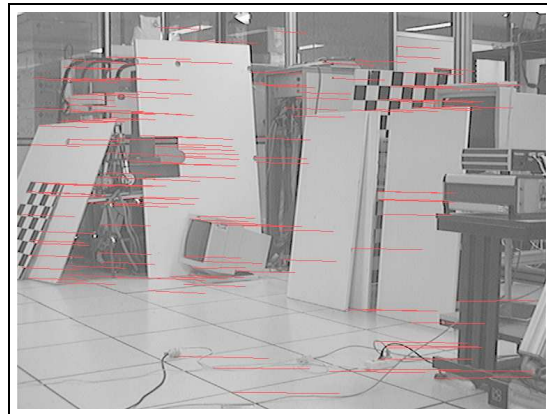


FIG. 4.10 – disparité vectorielle entre les deux images de la figure 4.8.

– le cas de translation pure fournit un résultat moins bon que translation en x pure : moins on a de paramètres, plus la méthode est précise.

4.2.2.2 Expérimentation pour une variation de zoom.

On a renouvelé l'expérience précédente pour un mouvement de zoom. Pour cela, on a bloqué l'auto-focus afin que ce mécanisme n'intervienne pas dans cette expérience.

Les images acquises par Argès se trouvent en figure 4.11, les points détectés sur la figure 4.12 et les vecteurs de disparité vectorielle en figure 4.13.

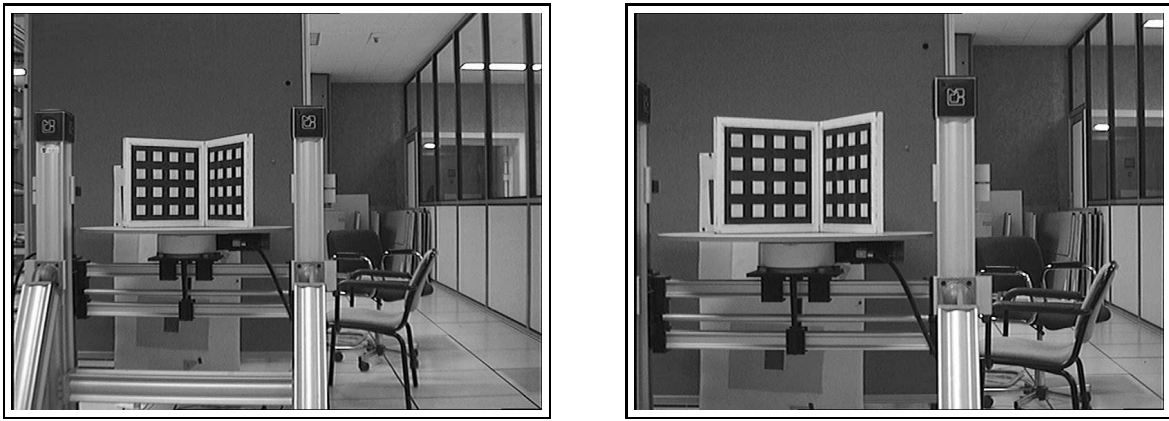


FIG. 4.11 – images prises par Argès lors d'un zoom et translation en x

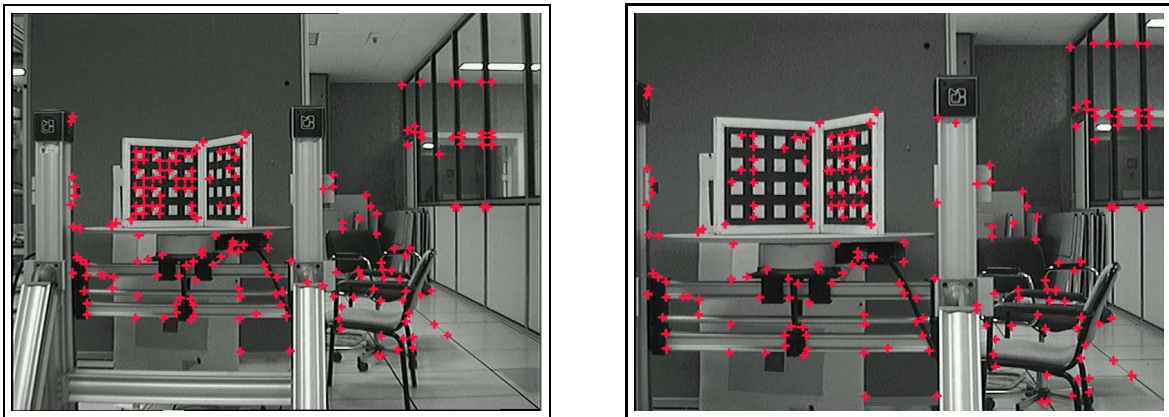


FIG. 4.12 – 196 points ont été détectés et appariés entre les 2 images

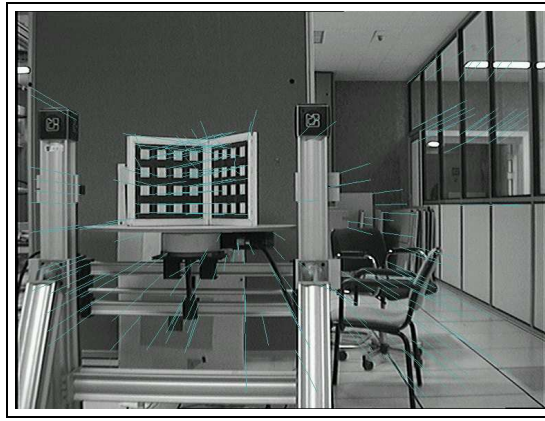


FIG. 4.13 – *disparité vectorielle entre les deux images de la figure 4.11.*

cas particulier	résidu
rotation et translation en z	168.212
translation pure	843.401
translation pure en x	937.84
translation pure en y	1185.67
translation pure en z	2958.12
rotation, zoom et translation en z	12.1572
zoom et translation en z	9.14224
zoom et translation en x	6.14666

Les conclusions sont identiques à celles du cas précédent. On notera cependant que le facteur prépondérant dans l'expérience précédente concerne la variation ou non du facteur de zoom et point principal.

4.2.2.3 Récapitulation d'expériences sur plusieurs types de mouvements singuliers.

On a effectué plusieurs expériences avec des types de mouvements différents et, pour chacune de ces expériences, on a cherché le modèle pour lequel le résidu est le plus faible afin de d'établir

dans quel cas on se trouve. Les résultats sont regroupés dans le tableau suivant :

mouvement réalisé	mouvement de plus faible résidu	erreur résiduelle dans le meilleur cas	erreur résiduelle dans le cas général
cas stationnaire	cas stationnaire	0.056	0.078
pure translation rétinienne	pure translation rétinienne	0.456	0.879
déplacement rétinien	déplacement rétinien	0.766	0.947
mouvement planaire	mouvement planaire	1.766	2.947
translation pure	translation pure	0.342	1.023
zoom	zoom	0.342	1.023

qui montre là encore la validité de la méthode.

4.3 Expérimentations sur les différentes formes d'homographies à l'infini.

Nous reprenons les formes particulières de matrice d'homographies étudiées au chapitre 1, à savoir :

numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
1	1	$[x1\ 0\ -x1\ 0\ -x1\ 0\ 0\ 0\ -x1]$	p1
2	1	$[x1\ 0\ -x1\ 0\ x1\ 0\ 0\ 0\ -x1]$	p1
3	1	$[x1\ 0\ 0\ 0\ -x1\ -x1\ 0\ 0\ -x1]$	p1, p2
4	1	$[x1\ 0\ 0\ 0\ -x1\ 0\ 0\ 0\ -x1]$	p1, p3
5	1	$[x1\ 0\ 0\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1, p3
6	1	$[x1\ 0\ 0\ 0\ -x1\ x1\ 0\ 0\ x1]$	p1
7	1	$[x1\ 0\ 0\ 0\ x1\ -x1\ 0\ 0\ -x1]$	p1
8	1	$[x1\ 0\ 0\ 0\ x1\ 0\ 0\ 0\ -x1]$	p1, p3
9	1	$[x1\ 0\ 0\ 0\ x1\ 0\ 0\ 0\ x1]$	p1, p2, p3
10	1	$[x1\ 0\ 0\ 0\ x1\ x1\ 0\ 0\ x1]$	p1, p2
11	1	$[x1\ 0\ x1\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1, p2
12	1	$[x1\ 0\ x1\ 0\ x1\ 0\ 0\ 0\ x1]$	p1, p2
13	2	$[x1\ 0\ -x1\ 0\ -x1\ x6\ 0\ 0\ -x1]$	p1
14	2	$[x1\ 0\ -x1\ 0\ x1\ x6\ 0\ 0\ -x1]$	p1
15	2	$[x1\ 0\ 0\ 0\ -x1\ 0\ 0\ 0\ x9]$	p1, p3
16	2	$[x1\ 0\ 0\ 0\ -x1\ x6\ 0\ 0\ -x1]$	p1
17	2	$[x1\ 0\ 0\ 0\ -x1\ x6\ 0\ 0\ x1]$	p1, p3

suite page suivante

numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
18	2	[x1 0 0 0 x1 0 0 0 x9]	p1, p2, p3
19	2	[x1 0 0 0 x1 x6 0 -x6 x1]	p3
20	2	[x1 0 0 0 x1 x6 0 0 -x1]	p1
21	2	[x1 0 0 0 x1 x6 0 0 x1]	p1
22	2	[x1 0 0 0 x5 -x5 0 0 -x5]	p1
23	2	[x1 0 0 0 x5 0 0 0 -x5]	p1, p3
24	2	[x1 0 0 0 x5 0 0 0 x5]	p1, p2, p3
25	2	[x1 0 0 0 x5 x5 0 0 x5]	p1, p2
26	2	[x1 0 x1 0 -x1 x6 0 0 x1]	p1, p2
27	2	[x1 0 x1 0 x1 x6 0 0 x1]	p1, p2
28	2	[x1 0 x3 0 -x1 -x1 0 0 -x1]	p1, p2
29	2	[x1 0 x3 0 -x1 0 0 0 -x1]	p1, p2
30	2	[x1 0 x3 0 -x1 0 0 0 x1]	p1
31	2	[x1 0 x3 0 -x1 x1 0 0 x1]	p1
32	2	[x1 0 x3 0 x1 -x1 0 0 -x1]	p1
33	2	[x1 0 x3 0 x1 0 -x3 0 x1]	p3
34	2	[x1 0 x3 0 x1 0 0 0 -x1]	p1
35	2	[x1 0 x3 0 x1 0 0 0 x1]	p1, p2
36	2	[x1 0 x3 0 x1 x1 0 0 x1]	p1, p2
37	2	[x1 x2 -x1 0 -x1 0 0 0 -x1]	p1
38	2	[x1 x2 -x1 0 x1 0 0 0 -x1]	p1
39	2	[x1 x2 0 -x2 x1 0 0 0 x1]	p1, p3
40	2	[x1 x2 0 -x2 x1 x1 0 0 x1]	p1
41	2	[x1 x2 0 0 -x1 0 0 0 -x1]	p1, p3
42	2	[x1 x2 0 0 -x1 0 0 0 x1]	p1, p3
43	2	[x1 x2 0 0 x1 0 0 0 -x1]	p1, p3
44	2	[x1 x2 0 0 x1 0 0 0 x1]	p1, p2, p3
45	2	[x1 x2 x1 -x2 x1 0 0 0 x1]	p1
46	2	[x1 x2 x1 0 -x1 0 0 0 x1]	p1, p2
47	2	[x1 x2 x1 0 x1 0 0 0 x1]	p1, p2
48	3	[x1 0 0 0 -x1 x6 0 0 x9]	p1, p2
49	3	[x1 0 0 0 x1 x6 0 0 x9]	p1, p2
50	3	[x1 0 0 0 x5 0 0 0 x9]	p1, p2, p3
51	3	[x1 0 0 0 x5 x6 0 -x6 x5]	p3
52	3	[x1 0 0 0 x5 x6 0 0 -x5]	p1

numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
53	3	$[x1\ 0\ 0\ 0\ x5\ x6\ 0\ 0\ x5]$	p1
54	3	$[x1\ 0\ 0\ 0\ x5\ x6\ 0\ x6\ -x5]$	p3
55	3	$[x1\ 0\ x3\ 0\ -x1\ 0\ 0\ 0\ x9]$	p1, p2
56	3	$[x1\ 0\ x3\ 0\ -x1\ x6\ 0\ 0\ -x1]$	p1, p2, p3
57	3	$[x1\ 0\ x3\ 0\ -x1\ x6\ 0\ 0\ x1]$	p1, p2, p3
58	3	$[x1\ 0\ x3\ 0\ x1\ 0\ -x3\ x8\ x1]$	p3
59	3	$[x1\ 0\ x3\ 0\ x1\ 0\ 0\ 0\ x9]$	p1, p2
60	3	$[x1\ 0\ x3\ 0\ x1\ x6\ -x3\ -x6\ x1]$	p3
61	3	$[x1\ 0\ x3\ 0\ x1\ x6\ 0\ -x6\ x1]$	p3
62	3	$[x1\ 0\ x3\ 0\ x1\ x6\ 0\ 0\ -x1]$	p1, p3
63	3	$[x1\ 0\ x3\ 0\ x1\ x6\ 0\ 0\ x1]$	p1, p2, p3
64	3	$[x1\ 0\ x3\ 0\ x5\ -x5\ 0\ 0\ -x5]$	p1
65	3	$[x1\ 0\ x3\ 0\ x5\ 0\ -x3\ 0\ x1]$	p3
66	3	$[x1\ 0\ x3\ 0\ x5\ 0\ 0\ 0\ -x5]$	p1, p2
67	3	$[x1\ 0\ x3\ 0\ x5\ 0\ 0\ 0\ x5]$	p1, p2, p3
68	3	$[x1\ 0\ x3\ 0\ x5\ 0\ x3\ 0\ -x1]$	p3
69	3	$[x1\ 0\ x3\ 0\ x5\ x5\ 0\ 0\ x5]$	p1, p2
70	3	$[x1\ x2\ 0\ -x2\ x1\ 0\ 0\ 0\ x9]$	p1, p3
71	3	$[x1\ x2\ 0\ -x2\ x1\ x6\ 0\ -x6\ x1]$	p3
72	3	$[x1\ x2\ 0\ -x2\ x1\ x6\ 0\ 0\ x1]$	p1
73	3	$[x1\ x2\ 0\ -x2\ x1\ x6\ 0\ 0\ x6]$	p1
74	3	$[x1\ x2\ 0\ 0\ -x1\ 0\ 0\ 0\ x9]$	p1, p3
75	3	$[x1\ x2\ 0\ 0\ x1\ 0\ 0\ 0\ x9]$	p1, p2, p3
76	3	$[x1\ x2\ 0\ 0\ x5\ 0\ 0\ 0\ -x5]$	p1, p3
77	3	$[x1\ x2\ 0\ 0\ x5\ 0\ 0\ 0\ x5]$	p1, p2, p3
78	3	$[x1\ x2\ 0\ x2\ -x1\ 0\ 0\ 0\ x9]$	p1, p3
79	3	$[x1\ x2\ 0\ x2\ -x1\ x6\ 0\ 0\ x6]$	p1
80	3	$[x1\ x2\ x1\ -x2\ x1\ x6\ 0\ 0\ x1]$	p1
81	3	$[x1\ x2\ x3\ -x2\ x1\ 0\ -x3\ 0\ x1]$	p3
82	3	$[x1\ x2\ x3\ -x2\ x1\ 0\ 0\ 0\ x1]$	p1
83	3	$[x1\ x2\ x3\ -x2\ x1\ 0\ 0\ 0\ x3]$	p1
84	3	$[x1\ x2\ x3\ -x2\ x1\ x1\ 0\ 0\ x1]$	p1
85	3	$[x1\ x2\ x3\ 0\ -x1\ -x1\ 0\ 0\ -x1]$	p1, p2
86	3	$[x1\ x2\ x3\ 0\ -x1\ 0\ 0\ 0\ -x1]$	p1, p3
87	3	$[x1\ x2\ x3\ 0\ -x1\ 0\ 0\ 0\ x1]$	p1

numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
88	3	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_1 \ 0 \ 0 \ x_1]$	p1
89	3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ -x_1 \ 0 \ 0 \ -x_1]$	p1
90	3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ 0 \ 0 \ -x_1]$	p1
91	3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ 0 \ 0 \ x_1]$	p1, p2, p3
92	3	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_1 \ 0 \ 0 \ x_1]$	p1, p2
93	3	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ 0 \ 0 \ 0 \ x_3]$	p1
94	4	$[x_1 \ 0 \ 0 \ 0 \ x_1 \ x_6 \ 0 \ x_8 \ x_9]$	p3
95	4	$[x_1 \ 0 \ 0 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2
96	4	$[x_1 \ 0 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2, p3
97	4	$[x_1 \ 0 \ x_3 \ 0 \ x_1 \ 0 \ x_7 \ 0 \ x_9]$	p3
98	4	$[x_1 \ 0 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2, p3
99	4	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1, p2
100	4	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ -x_5]$	p1, p2, p3
101	4	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1, p2, p3
102	4	$[x_1 \ x_2 \ 0 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1
103	4	$[x_1 \ x_2 \ 0 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1, p2, p3
104	4	$[x_1 \ x_2 \ 0 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1
105	4	$[x_1 \ x_2 \ 0 \ x_4 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1, p3
106	4	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_5]$	p1, p3
107	4	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_5 \ 0 \ 0 \ x_5]$	p1
108	4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ 0 \ 0 \ 0 \ x_9]$	p1
109	4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ -x_3 \ -x_6 \ x_1]$	p3
110	4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_1]$	p1, p3
111	4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_3]$	p1
112	4	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_6]$	p1
113	4	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1, p2
114	4	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ -x_1]$	p1, p2, p3
115	4	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ x_1]$	p1, p2, p3
116	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ -x_3 \ x_8 \ x_1]$	p3
117	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ 0 \ 0 \ 0 \ x_9]$	p1, p2
118	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ -x_6 \ x_1]$	p3
119	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ -x_1]$	p1, p3
120	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ x_1]$	p1, p2, p3
121	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ -x_5 \ 0 \ 0 \ -x_5]$	p1
122	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ -x_5]$	p1, p2

numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
123	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_5]$	p1, p2, p3
124	4	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_5 \ 0 \ 0 \ x_5]$	p1, p2
125	4	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1
126	4	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_3]$	p1, p2
127	4	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_6]$	p2
128	4	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ 0 \ 0 \ 0 \ x_3]$	p1
129	5	$[x_1 \ 0 \ 0 \ 0 \ x_5 \ x_6 \ 0 \ x_8 \ x_9]$	p3
130	5	$[x_1 \ 0 \ x_3 \ 0 \ x_1 \ x_6 \ x_7 \ x_8 \ x_1]$	p3
131	5	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ 0 \ x_7 \ 0 \ x_9]$	p3
132	5	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2, p3
133	5	$[x_1 \ 0 \ x_3 \ 0 \ x_5 \ x_6 \ x_7 \ -x_6 \ x_5]$	p3
134	5	$[x_1 \ 0 \ x_3 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ -x_5]$	p2
135	5	$[x_1 \ x_2 \ 0 \ -x_2 \ x_1 \ x_6 \ 0 \ x_8 \ x_9]$	p3
136	5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1, p2, p3
137	5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_6 \ 0 \ -x_6 \ x_5]$	p3
138	5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1
139	5	$[x_1 \ x_2 \ 0 \ x_4 \ x_5 \ x_6 \ 0 \ 0 \ x_6]$	p1, p2
140	5	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ 0 \ x_7 \ 0 \ x_9]$	p3
141	5	$[x_1 \ x_2 \ x_3 \ -x_2 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1, p3
142	5	$[x_1 \ x_2 \ x_3 \ 0 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2, p3
143	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2, p3
144	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ -x_3 \ x_8 \ x_1]$	p3
145	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ 0 \ 0 \ x_9]$	p1, p2
146	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ 0 \ x_3 \ x_8 \ -x_1]$	p3
147	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ -x_6 \ x_5]$	p3
148	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ -x_1]$	p2
149	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ -x_5]$	p1, p2, p3
150	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ 0 \ x_5]$	p1, p2, p3
151	5	$[x_1 \ x_2 \ x_3 \ 0 \ x_5 \ x_6 \ 0 \ x_6 \ -x_5]$	p3
152	5	$[x_1 \ x_2 \ x_3 \ x_2 \ -x_1 \ x_6 \ 0 \ 0 \ x_9]$	p1, p2, p3
153	5	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ 0 \ 0 \ 0 \ x_9]$	p1
154	5	$[x_1 \ x_2 \ x_3 \ x_4 \ -x_1 \ x_6 \ 0 \ 0 \ x_6]$	p1, p2
155	5	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_3]$	p1, p2
156	5	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ 0 \ 0 \ 0 \ x_5]$	p1
157	5	$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_5 \ 0 \ 0 \ x_5]$	p1

numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
158	6	[x1 0 x3 0 x1 x6 x7 x8 x9]	p3
159	6	[x1 0 x3 x4 x5 x6 0 0 x9]	p2
160	6	[x1 0 x3 x4 x5 x6 x7 0 -x1]	p3
161	6	[x1 x2 0 x4 x5 x6 0 0 x9]	p1, p2
162	6	[x1 x2 x3 -x2 x5 x6 -x3 x6 x9]	p3
163	6	[x1 x2 x3 -x2 x5 x6 x3 -x6 x9]	p3
164	6	[x1 x2 x3 0 x1 0 x7 x8 x9]	p3
165	6	[x1 x2 x3 0 x1 x6 x7 x8 x1]	p3
166	6	[x1 x2 x3 0 x5 0 x7 x8 x5]	p3
167	6	[x1 x2 x3 0 x5 x6 0 0 x9]	p1, p2, p3
168	6	[x1 x2 x3 0 x5 x6 0 x8 -x5]	p3
169	6	[x1 x2 x3 x2 x5 x6 -x3 -x6 x9]	p3
170	6	[x1 x2 x3 x2 x5 x6 x3 x6 x9]	p3
171	6	[x1 x2 x3 x4 -x1 x6 0 0 x9]	p1, p2, p3
172	6	[x1 x2 x3 x4 x5 0 0 0 x9]	p1, p2
173	6	[x1 x2 x3 x4 x5 0 x7 0 x5]	p3
174	6	[x1 x2 x3 x4 x5 x6 0 0 x3]	p2
175	6	[x1 x2 x3 x4 x5 x6 0 0 x5]	p1
176	6	[x1 x2 x3 x4 x5 x6 0 0 x6]	p1, p2
177	7	[x1 0 x3 0 x5 x6 x7 x8 x9]	p3
178	7	[x1 0 x3 x4 x5 x6 x7 0 x9]	p3
179	7	[x1 x2 0 x4 x5 x6 0 x8 x9]	p3
180	7	[x1 x2 x3 -x2 x1 x6 x7 x8 x9]	p3
181	7	[x1 x2 x3 0 x1 x6 x7 x8 x9]	p3
182	7	[x1 x2 x3 0 x5 0 x7 x8 x9]	p3
183	7	[x1 x2 x3 0 x5 x6 0 x8 x9]	p3
184	7	[x1 x2 x3 0 x5 x6 x7 x8 x5]	p3
185	7	[x1 x2 x3 x4 x5 0 -x3 x8 x9]	p3
186	7	[x1 x2 x3 x4 x5 0 x7 0 x9]	p3
187	7	[x1 x2 x3 x4 x5 x6 0 -x6 x9]	p3
188	7	[x1 x2 x3 x4 x5 x6 0 0 x9]	p1, p2, p3
189	7	[x1 x2 x3 x4 x5 x6 x7 -x6 x5]	p3
190	8	[x1 x2 x3 -x2 x5 x6 x7 x8 x9]	p3
191	8	[x1 x2 x3 0 x5 x6 x7 x8 x9]	p3
192	8	[x1 x2 x3 x2 x5 x6 x7 x8 x9]	p3

<i>suite de la page précédente</i>			
numéro du cas	nombre de paramètres	forme simplifiée de matrice homographie	type de projection générant un tel cas
193	8	[x1 x2 x3 x4 x5 0 x7 x8 x9]	p3
194	8	[x1 x2 x3 x4 x5 x6 0 x8 x9]	p3
195	8	[x1 x2 x3 x4 x5 x6 x7 -x6 x9]	p3
196	8	[x1 x2 x3 x4 x5 x6 x7 0 x9]	p3
197	8	[x1 x2 x3 x4 x5 x6 x7 x6 x9]	p3
198	9	[x1 x2 x3 x4 x5 x6 x7 x8 x9]	p3

Dans un premier temps, nous avons effectué différents mouvements avec le système Argès afin de valider expérimentalement notre approche.

Dans la séquence suivante (figure 4.14), on a modifié le système Argès en tournant la caméra d'un quart de tour afin de pouvoir effectuer un mouvement de rotation selon un axe approximativement parallèle à l'axe optique. Entre chaque acquisition, la caméra a tourné de 5 degrés.

Pour chaque couple d'images consécutives, nous avons calculé les résidus pour chaque cas d'homographie et avons trouvé que le cas de résidu le plus faible est, pour chaque couple d'images, le cas numéro 110, correspondant à la forme [x1 x2 x3 -x2 x1 x6 0 0 x1]:

$$\mathbf{H} = \begin{pmatrix} x_1 & x_2 & x_3 \\ -x_2 & x_1 & x_6 \\ 0 & 0 & x_1 \end{pmatrix}$$

Les cas permettant d'obtenir une telle forme de matrice d'homographie sont les suivants :

```
p[1-3]g1f1s1b1B1c[2-3]T1t13r1U1R3u9Z3W3D3a4n913m3
p1g1f1s1b1B1c1T2t12r1U1R3u9Z3W3D3a4n913m3
p1g1f1s1b1B1c[2-3]T2t[1-12]r1U1R3u9Z3W3D3a4n913m3
```

Dans tous ces cas, la rotation est approximée au premier ordre et l'axe est parallèle à l'axe optique, les paramètres intrinsèques sont fixes sauf le point principal qui est autorisé à varier. La projection peut être perspective auquel cas la translation est nulle, ou orthographique, la translation étant alors quelconque.

Nous avons ensuite effectué une expérience sur une séquence concernant un mouvement approximatif de rotation autour de l'axe optique en tenant une caméra dans la main et en la faisant pivoter autour de son axe optique. Nous montrons les images de début et de fin de séquence sur la figure 4.15 et avons obtenu pour forme de plus faible résidu la forme numéro 110 qui correspond effectivement au mouvement effectué comme nous venons de le voir.



FIG. 4.14 – Rotation de la caméra d'Argès autour de son axe optique.



FIG. 4.15 – *Mouvement approximatif de rotation autour de l'axe optique et translation*

4.4 Une application de vision active sur un cas particulier

Nous présentons dans cette section une application en vision active concernant le cas de la projection orthographique et qui a déjà été présentée dans (Lingrand et Viéville, 1996) et qui constitue un prolongement de (Viéville et Faugeras, 1995).

Dans cette expérimentation, on a utilisé le modèle de projection orthographique et supposé les paramètres de projection constants mais non connus. On a également supposé le facteur γ nul, ce qui est une hypothèse tout à fait raisonnable, compte tenu des approximations précédentes ainsi que, suivant (Enciso, 1995), le rapport $\frac{\alpha_u}{\alpha_v}$ est constant et connu. On appliquera donc tout d'abord une transformation aux coordonnées images afin de pouvoir exprimer le modèle de projection de la façon suivante :

$$\begin{cases} u = u_0 + f X \\ v = v_0 + f Y \end{cases} \quad (4.1)$$

Suivant (Waxman et Ullman, 1985), l'expression de la disparité vectorielle, appelée également flot optique, sur une surface plane d'équation $Z = N_x X + N_y Y + Z_N$, pour un mouvement de translation $[t_x \ t_y \ t_z]^T$ et rotation de vecteur $[r_0 \ r_1 \ r_2]^T$, approximée au premier ordre ($\mathbf{R} = \mathbf{I} + \tilde{\mathbf{r}}$) est alors la suivante :

$$\begin{cases} du = C_u + a u + b v \\ dv = C_v + c u + d v \end{cases} \quad (4.2)$$

avec :

$$\begin{cases} C_u = f t_0 + f r_1 Z_N - r_1 N_x u_0 + (r_2 - r_1 N_y) v_0 \\ C_v = f t_1 - f r_0 Z_N + (r_0 N_x - r_2) u_0 + r_0 N_y v_0 \\ a = r_1 N_x \\ b = r_1 N_y - r_2 \\ c = r_2 - r_0 N_x \\ d = -r_0 N_y \end{cases} \quad (4.3)$$

Les six inconnues (C_u, C_v, a, b, c, d) du problème sont à l'évidence déterminées par trois points non alignés, à condition, bien entendu, de considérer ces quantités constantes pour ces trois points. Il est alors naturel d'utiliser, pour gérer le réseau de ces triplets de points, la triangulation de Delaunay (Devillers et al., 1992) que l'on propagera d'image en image par les appariements de points. De ces inconnues, liées aux composantes du mouvement, à la normale à la facette et aux paramètres intrinsèques, on obtient d'une part, des équations quadratiques permettant de déterminer, à une constante près, les deux premières composantes de la normale à la facette et de façon similaire le vecteur de rotation :

$$a N_y^2 + d N_x^2 - N_x N_y (b + c) = 0 \quad (4.4)$$

$$a w_y^2 + d w_x^2 + w_x w_y (b + c) = 0 \quad (4.5)$$

Dans la pratique, le modèle qui convient le mieux à nos caméras est un modèle projectif, dont la validité est accrue dans la zone fovéale. Or, nous avons observé, au chapitre 1, que si le mouvement est rétinien, les équations de la disparité vectorielle sont les mêmes dans le cas perspectif et orthographique, à condition de supposer les équivalences suivantes (voir (Lingrand et Viéville, 1995) pour plus de détails) :

$$r_0 \leftrightarrow \frac{t_1}{Z_0} \text{ et } r_1 \leftrightarrow -\frac{t_0}{Z_0}$$

L'expérience permettant de valider ces résultats a consisté en la prise de plusieurs séquences d'images avec un mouvement du rail, ces séquences se distinguant par l'orientation de l'axe optique de la caméra par rapport au rail. Pour une valeur d'élévation fixée, on teste différentes valeurs de rotation horizontale (roulis), ce qui revient à ajouter de la translation selon l'axe optique. L'allure de la fonction erreur obtenue au cours de chacune des séquences est en accord avec les résultats théoriques (voir 4.16).

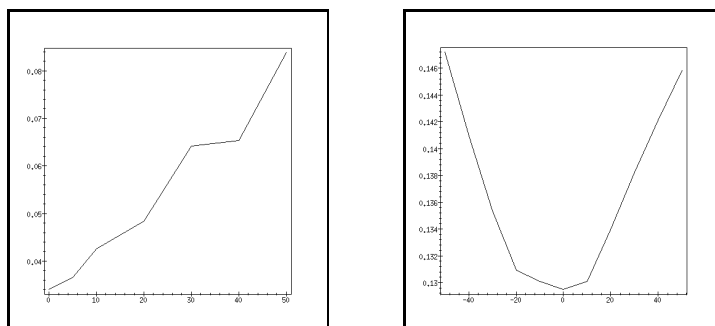


FIG. 4.16 – Evolution de l'erreur en mouvement en fonction de l'orientation en roulis de la caméra, modèle théorique et expérimentation (scène de la mire).

On montre ainsi que les mouvements validant nos équations sont bien ceux laissant le plan rétinien invariant et permettent alors une reconstruction optimale. Néanmoins, la reconstruction

est plus sensible aux aléas expérimentaux que la fonction erreur car d'autres perturbations rentrent en ligne de compte (qualité de la vision précoce, variations d'éclairage, erreurs d'appariements, ...). La qualité moyenne des reconstructions (voir 4.18) est due, dans notre cas, plus aux autres sources d'erreurs qu'au mouvement proprement dit.

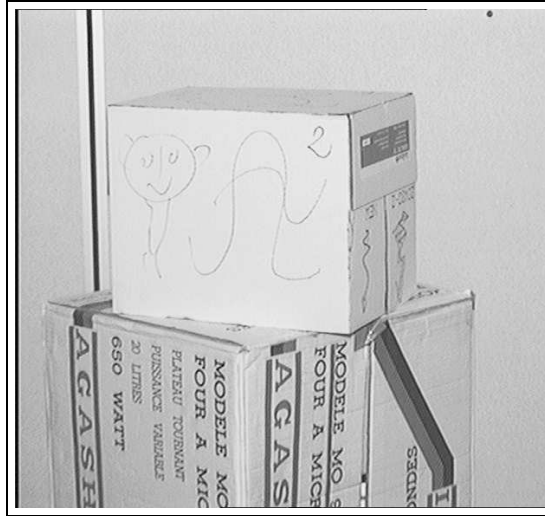


FIG. 4.17 – Scène réelle vue par Argès.

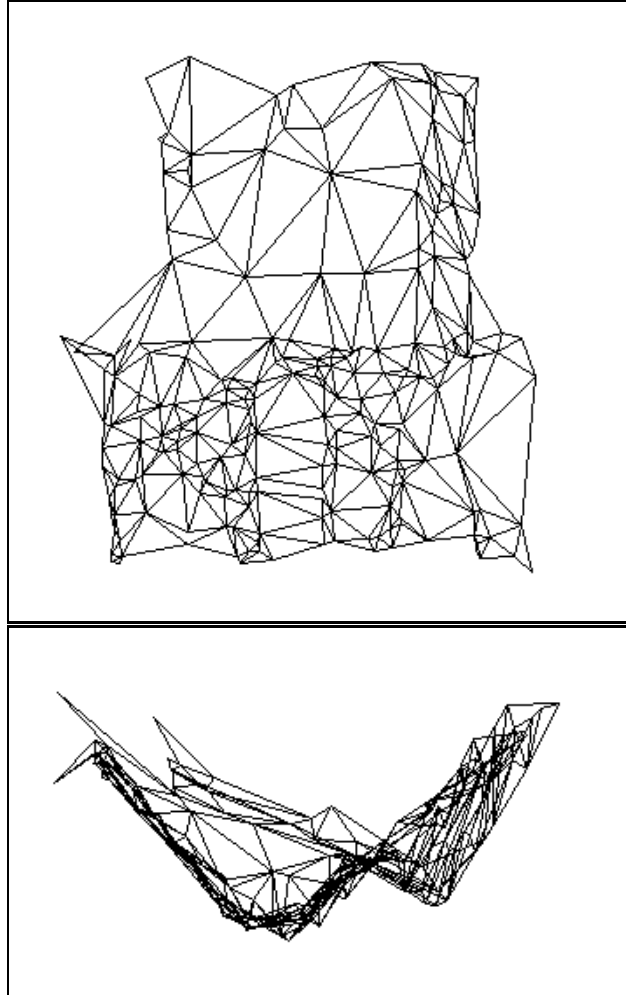


FIG. 4.18 – *Vues de face et de dessus de la reconstruction.*

4.5 Application: reconstruction hiérarchique.

L'intérêt immédiat de ce travail est de permettre de recueillir plus d'informations sur la caméra, la scène et les déplacements. On peut ensuite envisager de se placer volontairement et successivement dans des cas particuliers pour obtenir un maximum d'informations et une reconstruction "hiérarchique": Le robot effectue un déplacement particulier pour déterminer tel ou tel paramètre qu'il utilisera ensuite lors d'un autre déplacement pour reconstruire la scène, ... les critères permettant de déterminer de façon automatique l'adéquation du mouvement aux cas particuliers, tout comme nous l'avons vu dans le paragraphe précédent 4.4.

4.6 Application: segmentation d'une séquence vidéo.

Une autre application consiste en la segmentation d'une séquence vidéo. On va pouvoir segmenter les objets de la scène en objets rigides ou objets de même mouvement, puis objets de même type de mouvement (en rotation, en translation, de même axe, d'axes orthogonaux, en mouvements sur un plan, ...) ou encore objets de structures géométriques particulières, par exemple planaire.

Conclusion et Perspectives

Pour conclure, je dirais qu'il reste encore beaucoup à faire.

« Je sens qu'ce soir, j'veis conclure »

In "Les bronzés font du ski"

Conclusion (english abstract)

In this thesis, we have experimentally demonstrated the interest of the parametrization of singular cases. In particular:

- we handle singularities in equations for which the general equations are degenerate,*
- we exploit as best as possible the available information, allowing us to determine in some cases that are not available in the general equations,*
- we obtain more accurate numerical results than in the general case, because of the reduce number of unknowns.*

Furthermore, we have implemented these particular cases on a system including the Argès robotic platform, allowing us to determine automatically from images which particular case was performed. The control of such a particular displacement has been implemented with success.

We have then proposed several applications such as 3D reconstruction, video sequences segmentation and, moreover, perspectives about human 3D motion perception using EEG and MEG data.

1 Conclusion

Dans cette thèse, nous nous sommes tout d'abord intéressé à la modélisation des caméras et du mouvement et avons choisi la modélisation la plus adéquate à notre problème. Nous avons également montré, dans le cadre de la géométrie épipolaire entre deux vues, que, suivant le modèle de caméra adopté, l'existence d'une relation homographique entre les points images dépend de conditions à la fois géométriques et de mouvement.

Nous avons étudié les singularités des équations reliant les points projetés dans les images, tout d'abord en considérant deux images (matrices fondamentales et homographies) puis différentes vues d'une séquence d'images. L'ensemble des cas particuliers ainsi obtenus étant numériquement extrêmement élevé, nous nous sommes alors intéressé à la génération automatique de l'ensemble de ces contraintes ainsi qu'à sa gestion. Ainsi, nous avons généré toutes les formes de matrices fondamentales et d'homographies possibles entre deux vues afin d'étudier les simplifications apportées.

Après avoir ensuite étudié les méthodes à notre disposition pour évaluer numériquement les paramètres des différents modèles et déterminer l'adéquation de ces modèles aux données, nous nous sommes intéressé à la validation expérimentale.

Nous avons ainsi pu valider expérimentalement l'intérêt de la paramétrisation de ces cas particuliers en montrant que :

- on évite les singularités des équations pour lesquelles les équations générales sont dégénérées,
- on exploite au mieux les informations disponibles en permettant dans certains cas de déterminer des paramètres qu'il n'est pas possible de déterminer dans le cas général,
- on obtient une précision numérique supérieure au cas général, étant donné le nombre réduit de paramètres.

Nous avons alors implémenté un module gérant différents cas particuliers sur un système robotique, permettant de déterminer automatiquement dans quel cas singulier on se trouve. L'asservissement d'un robot sur un cas particulier a ensuite été implémenté et testé avec succès.

Nous proposons finalement diverses applications, par exemple la reconstruction hiérarchique par asservissement successifs sur différents cas spécifiques permettant chacun de retrouver des informations complémentaires et ensuite de reconstruire précisément un environnement.

2 Perspectives

Plusieurs familles de perspectives s'offrent maintenant.

La première concerne une implémentation de plus grande envergure concernant un système de reconstruction hiérarchique. Il faut alors repenser l'organisation logicielle afin d'obtenir un système plus rapide et paralléliser les calculs. Mais surtout exhiber plus en détails les différents cas qui vont s'emboîter afin de réaliser une reconstruction hiérarchique, et ceci suivant des contraintes liées aux degrés de liberté du système robotique mais aussi à la scène (sauf pour les passe-murailles et les oiseaux volants).

La seconde consiste à appliquer ces résultats sur la segmentation de scènes dans des séquences vidéo. Il s'agit alors de segmenter les objets de mouvements rigides différents du reste de la scène. On pourra alors détecter, par différence les objets non rigides. On segmentera également les objets de structures différents. François Gaspard a déjà réutilisé avec succès une partie de ces travaux pour la détermination de plans d'orientations différentes. Il reste encore beaucoup de perspectives dans ce domaine.

Cependant, une autre perspective alléchante et plus éloignée de ces travaux consiste en la compréhension des mécanismes de perception du mouvement. La question que l'on peut se poser est de savoir si la perception du mouvement n'est pas incrémentale, partant d'une évaluation grossière obtenue par un modèle très simple puis affinée par des modèles de plus en plus précis, en fonction de la précision nécessaire. Il est évident que pour obtenir une vue globale d'une scène ou pour effectuer une manoeuvre délicate, on n'a pas besoin de la même précision. Cette perspective est présentée en annexe C.

ANNEXES

Annexe A

Maple : un outil adapté aux calculs lourds, longs et fastidieux

Ici sont présentés tous les calculs que vous auriez aimé voir dans la thèse sans oser les demander.

A.1 Code Maple :coefficients de la disparité vectorielle dans le cas de la projection perspective

```

> r:=matrix([[r11,r12,r13],[r21,r22,r23],[r31,r32,r33]]):
> Z := Nx * X + Ny * Y + ZN:
> equ := au*X/Z + gamma*Y/Z + u_0 - u:
> eqv := av*Y/Z + v0 - v:
> assign(solve({equ,eqv},{X,Y})):
bytes used=23029936, alloc=2096768, time=17.88
> X2:= r[1,1] * X + r[1,2] * Y + r[1,3] * Z + t0:
> Y2:= r[2,1] * X + r[2,2] * Y + r[2,3] * Z + t1:
> Z2:= r[3,1] * X + r[3,2] * Y + r[3,3] * Z + t2:
> u2 := au2 * X2/Z2 + g2 * Y2/Z2 + u0p:
> v2 := av2 * Y2/Z2 + v0p:
> du := factor(u2 - u):
> dv := factor(v2 - v):
> Cu := coeff(coeff(numer(du),u,0),v,0);

Cu := -au2 t0 u_0 Nx av - g2 t1 au v0 Ny + g2 t1 gamma Nx v0 - g2 t1 u_0 Nx av
      - u0p r31 ZN v0 gamma + u0p r31 ZN av u_0 + u0p r32 au ZN v0
      - u0p r33 ZN au av - u0p t2 au v0 Ny + u0p t2 gamma Nx v0 - u0p t2 u_0 Nx av
      - g2 r21 ZN v0 gamma + g2 r21 ZN av u_0 + g2 r22 au ZN v0 - g2 r23 ZN au av
      - g2 t1 au av - u0p t2 au av - au2 r11 ZN v0 gamma + au2 r11 ZN av u_0
      + au2 r12 au ZN v0 - au2 r13 ZN au av - au2 t0 au v0 Ny + au2 t0 gamma Nx v0
      - au2 t0 au av

> Cv := coeff(coeff(numer(dv),u,0),v,0);
Cv := -v0p t2 u_0 Nx av - av2 r21 ZN v0 gamma + av2 r21 ZN av u_0
      + av2 r22 au ZN v0 - av2 r23 ZN au av - av2 t1 au v0 Ny + av2 t1 gamma Nx v0
      - av2 t1 u_0 Nx av - v0p r31 ZN v0 gamma + v0p r31 ZN av u_0
      + v0p r32 au ZN v0 - v0p r33 ZN au av - v0p t2 au v0 Ny + v0p t2 gamma Nx v0
      - av2 t1 au av - v0p t2 au av

> Cd := coeff(coeff(denom(du),u,0),v,0);
Cd := -r31 ZN v0 gamma + r31 ZN av u_0 + r32 au ZN v0 - r33 ZN au av - t2 au av
      - t2 au v0 Ny + t2 gamma Nx v0 - t2 u_0 Nx av

> a := coeff(numer(du),u);
a := g2 t1 Nx av - u0p r31 ZN av + u0p t2 Nx av - r31 ZN gamma v + r31 ZN v0 gamma
      - r31 ZN av u_0 - r32 au ZN v0 + r32 au ZN v + r33 ZN au av + t2 au v0 Ny
      - t2 au v Ny - t2 gamma Nx v0 + t2 gamma Nx v + t2 u_0 Nx av - g2 r21 ZN av
      + t2 au av + au2 t0 Nx av - au2 r11 ZN av

> b := coeff(numer(du),v);
b := g2 r21 ZN gamma + g2 t1 au Ny - g2 t1 gamma Nx + u0p r31 ZN gamma - u0p r32 au ZN
      + u0p t2 au Ny - u0p t2 gamma Nx - u r31 ZN gamma + u r32 au ZN - u t2 au Ny
      + u t2 gamma Nx - g2 r22 au ZN + au2 r11 ZN gamma - au2 r12 au ZN
      + au2 t0 au Ny - au2 t0 gamma Nx

> c := coeff(numer(dv),u);
c := v0p t2 Nx av + v r31 ZN av - v t2 Nx av - av2 r21 ZN av + av2 t1 Nx av
      - v0p r31 ZN av

> d := coeff(numer(dv),v);
d := r31 ZN v0 gamma - r31 ZN av u_0 + r31 ZN av u - r32 au ZN v0 + r33 ZN au av

```

```

+ t2 au v0 Ny - t2 gamma Nx v0 + t2 u_0 Nx av - t2 u Nx av + av2 r21 ZN gamma
- av2 r22 au ZN + av2 t1 au Ny - av2 t1 gamma Nx + v0p r31 ZN gamma
- v0p r32 au ZN + v0p t2 au Ny - v0p t2 gamma Nx + t2 au av

```

```

> e := coeff(numer(du),u^2);
      e := r31 ZN av - t2 Nx av

```

```

> g := coeff(numer(dv),v^2);
      g := r32 au ZN + t2 gamma Nx - t2 au Ny - r31 ZN gamma

```

A.2 Code Maple :coefficients de la disparité vectorielle dans le cas de la projection para-perspective

```

a := coeff(numer(du),u);
b := coeff(numer(du),v);
c := coeff(numer(dv),u);
d := coeff(numer(dv),v);
> r:=matrix([[r11,r12,r13],[r21,r22,r23],[r31,r32,r33]]);
> Z := Nx * X + Ny * Y + ZN:
> equ := au*X + g*Y + bu*Z + u_0 - u:
> eqv := av*Y + bv*Z + v0 - v:
> assign(solve({equ,eqv},{X,Y})):
bytes used=1000120, alloc=786288, time=0.54
> X2:= r[1,1] * X + r[1,2] * Y + r[1,3] * Z + t0:
> Y2:= r[2,1] * X + r[2,2] * Y + r[2,3] * Z + t1:
> Z2:= r[3,1] * X + r[3,2] * Y + r[3,3] * Z + t2:
>
> u2 := au2 * X2 + g2 * Y2 + bu2 * Z2 + u0p:
> v2 := av2 * Y2 + bv2 * Z2 + v0p:
> du := factor(u2 - u):
> dv := factor(v2 - v):
> Cu := coeff(coeff(numer(du),u,0),v,0);
Cu := au2 r11 av u_0 - au2 r11 v0 g + au2 r12 au v0 - au2 t0 au av
      + g2 r21 av u_0 - g2 r21 v0 g + g2 r22 au v0 - g2 t1 au av
      + bu2 r31 av u_0 - bu2 r31 v0 g + bu2 r32 au v0 - bu2 t2 au av
      - u0p au bv Ny + u0p g bv Nx - u0p bu Nx av - au2 r11 Ny bu v0
      - au2 r11 bv ZN g + au2 r11 bv Ny u_0 + au2 r11 ZN bu av
      + au2 r12 au bv ZN + au2 r12 bu Nx v0 - au2 r12 u_0 bv Nx
      + au2 r13 Nx av u_0 - au2 r13 Nx v0 g + au2 r13 Ny au v0
      - au2 r13 ZN au av - au2 t0 au bv Ny - u0p au av + au2 t0 g bv Nx
      - au2 t0 bu Nx av - g2 r21 Ny bu v0 - g2 r21 bv ZN g + g2 r21 bv Ny u_0
      + g2 r21 ZN bu av + g2 r22 au bv ZN + g2 r22 bu Nx v0 - g2 r22 u_0 bv Nx
      + g2 r23 Nx av u_0 - g2 r23 Nx v0 g + g2 r23 Ny au v0 - g2 r23 ZN au av
      - g2 t1 au bv Ny + g2 t1 g bv Nx - g2 t1 bu Nx av - bu2 r31 Ny bu v0
      - bu2 r31 bv ZN g + bu2 r31 bv Ny u_0 + bu2 r31 ZN bu av
      + bu2 r32 au bv ZN + bu2 r32 bu Nx v0 - bu2 r32 u_0 bv Nx
      + bu2 r33 Nx av u_0 - bu2 r33 Nx v0 g + bu2 r33 Ny au v0
      - bu2 r33 ZN au av - bu2 t2 au bv Ny + bu2 t2 g bv Nx - bu2 t2 bu Nx av

> Cv := coeff(coeff(numer(dv),u,0),v,0);
Cv := -v0p au av + av2 r21 av u_0 - av2 r21 v0 g + av2 r22 au v0
      - av2 t1 au av + bv2 r31 av u_0 - bv2 r31 v0 g + bv2 r32 au v0

```

```

- bv2 t2 au av - v0p au bv Ny + v0p g bv Nx - v0p bu Nx av
- av2 r21 Ny bu v0 - av2 r21 bv ZN g + av2 r21 bv Ny u_0
+ av2 r21 ZN bu av + av2 r22 au bv ZN + av2 r22 bu Nx v0
- av2 r22 u_0 bv Nx + av2 r23 Nx av u_0 - av2 r23 Nx v0 g
+ av2 r23 Ny au v0 - av2 r23 ZN au av - av2 t1 au bv Ny + av2 t1 g bv Nx
- av2 t1 bu Nx av - bv2 r31 Ny bu v0 - bv2 r31 bv ZN g
+ bv2 r31 bv Ny u_0 + bv2 r31 ZN bu av + bv2 r32 au bv ZN
+ bv2 r32 bu Nx v0 - bv2 r32 u_0 bv Nx + bv2 r33 Nx av u_0
- bv2 r33 Nx v0 g + bv2 r33 Ny au v0 - bv2 r33 ZN au av - bv2 t2 au bv Ny
+ bv2 t2 g bv Nx - bv2 t2 bu Nx av

> a := coeff(numer(du),u);
a := -au2 r11 av - g2 r21 av - bu2 r31 av + au bv Ny - g bv Nx + bu Nx av
- au2 r11 bv Ny + au2 r12 bv Nx - au2 r13 Nx av + au av - g2 r21 bv Ny
+ g2 r22 bv Nx - g2 r23 Nx av - bu2 r31 bv Ny + bu2 r32 bv Nx
- bu2 r33 Nx av
> b := coeff(numer(du),v);
b := au2 r11 g - au2 r12 au + g2 r21 g - g2 r22 au + bu2 r31 g - bu2 r32 au
+ au2 r11 Ny bu - au2 r12 bu Nx + au2 r13 Nx g - au2 r13 Ny au
+ g2 r21 Ny bu - g2 r22 bu Nx + g2 r23 Nx g - g2 r23 Ny au
+ bu2 r31 Ny bu - bu2 r32 bu Nx + bu2 r33 Nx g - bu2 r33 Ny au
> c := coeff(numer(dv),u);
c := -av2 r21 bv Ny - av2 r21 av - bv2 r31 av + av2 r22 bv Nx - av2 r23 Nx av
- bv2 r31 bv Ny + bv2 r32 bv Nx - bv2 r33 Nx av

> d := coeff(numer(dv),v);
d := bu Nx av + au av + au bv Ny - g bv Nx + av2 r21 g - av2 r22 au
+ bv2 r31 g - bv2 r32 au + av2 r21 Ny bu - av2 r22 bu Nx + av2 r23 Nx g
- av2 r23 Ny au + bv2 r31 Ny bu - bv2 r32 bu Nx + bv2 r33 Nx g
- bv2 r33 Ny au

```

A.3 Code Maple : Expressions de la matrice fondamentale \mathbf{F}

Le code Maple utilisé pour le calcul explicite de la matrice \mathbf{F} :

```

#bibliothèques utilisées
with(linalg): #standard
with(linalg1): # ajouts par Thierry.Vieville@sophia.inria.fr

#définition des variables et expressions
m := matrix([[u],[v],[1]]);
mp := matrix([[up],[vp],[1]]);
mtp := transpose(mp);
w :=vector([r_0, r_1, 2]);
pR :=rot3d(w);
t :=vector([t_0, t_1, t_2]);
R :=stackmatrix(concat(pR,t),array([0,0,0,1]));
Ap := matrix([
    [alpha_u_2, gamma_2, lambda*u_0_2, (1-lambda)*u_0_2],
    [0, alpha_v_2, lambda*v_0_2, (1-lambda)*v_0_2],
    [0, 0, mu, 1-mu]]);
q := map(evalc,multiply(Ap, R));

B1 := matrix([
    [u, 0, alpha_u, gamma_1, lambda*u_0, (1-lambda)*u_0],

```

```

[v, 0, 0, alpha_v, lambda*v_0, (1-lambda)*v_0],
[1, 0, 0, 0, mu, 1 - mu],
[0, up, q11, q12, q13, q14],
[0, vp, q21, q22, q23, q24],
[0, 1, q31, q32, q33, q34]];

F := matrix([[f11, f12, f13],
            [f21, f22, f23],
            [f31, f32, f33]]);

#calcul effectif
eqFond := expand(multiply(mtp,multiply(F, m))[1,1]);
eq := expand(sort(det(B1)-eqFond));
maListe:= [coeffs(eq, {u,v,up,vp})];
assign(solve({op(maListe)},{f11,f12,f13,f21,f22,f23,f31,f32,f33}));

#Ecriture sous forme LaTeX
latex(f11,'theF1');latex(f12,'theF2');latex(f13,'theF3');
latex(f21,'theF4');latex(f22,'theF5');latex(f23,'theF6');
latex(f31,'theF7');latex(f32,'theF8');latex(f33,'theF9');
#latex(q,'theQ');
quit;

```

et le résultat obtenu (certes compliqué) :

$$\begin{aligned}
F_{11} &= q31 q23 \alpha_v - q33 q21 \alpha_v - q22 q31 \lambda v_0 + q34 \mu q21 \alpha_v + q22 q31 \mu v_0 + q32 \lambda v_0 q21 + q33 \mu q21 \alpha_v - \\
& q31 \mu q23 \alpha_v - q24 q31 \mu \alpha_v - q32 \mu v_0 q21 \\
F_{12} &= q32 \alpha_u q23 - q22 q31 \mu u_0 - q31 q23 \gamma_1 + q22 q33 \alpha_u \mu + q33 q21 \gamma_1 - q22 q33 \alpha_u - q33 \mu q21 \gamma_1 + q22 q31 \lambda u_0 + \\
& q34 q22 \alpha_u \mu - q34 \mu q21 \gamma_1 + q32 \mu u_0 q21 + q24 q31 \mu \gamma_1 + q31 \mu q23 \gamma_1 - q32 \alpha_u \mu q23 - q24 q32 \alpha_u \mu - \\
& q32 \lambda u_0 q21 \\
F_{13} &= q32 \alpha_u \lambda v_0 q23 + q22 q33 \alpha_u v_0 - q22 q33 \alpha_u \lambda v_0 - q31 \lambda v_0 q23 \gamma_1 + q34 \alpha_u q23 \alpha_v + q24 q31 \lambda u_0 \alpha_v + \\
& q24 q32 \alpha_u \lambda v_0 - q32 \alpha_u v_0 q23 + q34 \lambda v_0 q21 \gamma_1 + q33 u_0 q21 \alpha_v - q33 v_0 q21 \gamma_1 + q31 v_0 q23 \gamma_1 - \\
& q24 q33 \alpha_u \alpha_v + q31 \lambda u_0 q23 \alpha_v - q24 q31 \lambda v_0 \gamma_1 - q31 u_0 q23 \alpha_v - q34 \lambda u_0 q21 \alpha_v + q33 \lambda v_0 q21 \gamma_1 - \\
& q34 q22 \alpha_u \lambda v_0 - q33 \lambda u_0 q21 \alpha_v \\
F_{21} &= q12 q31 \lambda v_0 - q11 q32 \lambda v_0 + q11 q32 \mu v_0 + q14 q31 \mu \alpha_v - q11 q33 \mu \alpha_v + q11 q33 \alpha_v - q31 q13 \alpha_v - \\
& q12 q31 \mu v_0 + q31 \mu q13 \alpha_v - q34 q11 \mu \alpha_v \\
F_{22} &= -q11 q33 \gamma_1 + q31 q13 \gamma_1 - q32 \alpha_u q13 - q34 q12 \alpha_u \mu + q12 q33 \alpha_u - q14 q31 \mu \gamma_1 - q11 q32 \mu u_0 + q34 q11 \mu \gamma_1 - \\
& q12 q33 \alpha_u \mu - q12 q31 \lambda u_0 + q14 q32 \alpha_u \mu + q32 \alpha_u \mu q13 + q12 q31 \mu u_0 - q31 \mu q13 \gamma_1 + q11 q33 \mu \gamma_1 + \\
& q11 q32 \lambda u_0 \\
F_{23} &= q31 u_0 q13 \alpha_v + q32 \alpha_u v_0 q13 + q14 q31 \lambda v_0 \gamma_1 - q12 q33 \alpha_u v_0 + q11 q33 \lambda u_0 \alpha_v - q31 v_0 q13 \gamma_1 - \\
& q14 q31 \lambda u_0 \alpha_v + q11 q33 v_0 \gamma_1 - q14 q32 \alpha_u \lambda v_0 - q11 q33 \lambda v_0 \gamma_1 + q34 q12 \alpha_u \lambda v_0 - q31 \lambda u_0 q13 \alpha_v - \\
& q34 \alpha_u q13 \alpha_v - q34 q11 \lambda v_0 \gamma_1 + q31 \lambda v_0 q13 \gamma_1 + q12 q33 \alpha_u \lambda v_0 - q32 \alpha_u \lambda v_0 q13 + q14 q33 \alpha_u \alpha_v + \\
& q34 q11 \lambda u_0 \alpha_v - q11 q33 u_0 \alpha_v \\
F_{31} &= -q22 q11 \mu v_0 + q22 q11 \lambda v_0 - q11 q23 \alpha_v + q13 q21 \alpha_v - q12 \lambda v_0 q21 - \mu q13 q21 \alpha_v + q11 q24 \mu \alpha_v - \\
& q14 \mu q21 \alpha_v + q12 \mu v_0 q21 + q11 \mu q23 \alpha_v \\
F_{32} &= q11 q23 \gamma_1 - q11 \mu q23 \gamma_1 + \mu q13 q21 \gamma_1 + q22 \alpha_u q13 + q14 \mu q21 \gamma_1 - q13 q21 \gamma_1 - q22 q11 \lambda u_0 - q22 \alpha_u \mu q13 - \\
& q22 q14 \alpha_u \mu - q12 \alpha_u q23 - q11 q24 \mu \gamma_1 + q22 q11 \mu u_0 + q12 \lambda u_0 q21 - q12 \mu u_0 q21 + q24 q12 \alpha_u \mu + \\
& q12 \alpha_u \mu q23 \\
F_{33} &= -q12 \alpha_u \lambda v_0 q23 + \lambda u_0 q13 q21 \alpha_v - \lambda v_0 q13 q21 \gamma_1 + v_0 q13 q21 \gamma_1 - q11 v_0 q23 \gamma_1 + q22 q14 \alpha_u \lambda v_0 - \\
& q14 \alpha_u q23 \alpha_v + q22 \alpha_u \lambda v_0 q13 + q11 q24 \lambda v_0 \gamma_1 - q22 \alpha_u v_0 q13 + q12 \alpha_u v_0 q23 + q11 u_0 q23 \alpha_v - \\
& q11 q24 \lambda u_0 \alpha_v + q24 \alpha_u q13 \alpha_v + q14 \lambda u_0 q21 \alpha_v - u_0 q13 q21 \alpha_v - q24 q12 \alpha_u \lambda v_0 + q11 \lambda v_0 q23 \gamma_1 - \\
& q14 \lambda v_0 q21 \gamma_1 - q11 \lambda u_0 q23 \alpha_v
\end{aligned}$$

avec la matrice q définie comme suit :

$$\left\{ \begin{array}{l} q_{11} = \frac{\alpha'_u \left(1 + \frac{-\frac{1}{2}r_2^2 - \frac{1}{2}r_1^2}{D}\right) + \frac{\gamma_2 \left(r_2 + \frac{1}{2}r_1 r_0\right)}{D} + \frac{\lambda u'_0 \left(-r_1 + \frac{1}{2}r_2 r_0\right)}{D}}{D} \\ q_{12} = \frac{\alpha'_u \left(-r_2 + \frac{1}{2}r_1 r_0\right)}{D} + \frac{\gamma_2 \left(1 + \frac{-\frac{1}{2}r_2^2 - \frac{1}{2}r_0^2}{D}\right) + \frac{\lambda u'_0 \left(r_0 + \frac{1}{2}r_2 r_1\right)}{D}}{D} \\ q_{13} = \frac{\alpha'_u \left(r_1 + \frac{1}{2}r_2 r_0\right)}{D} + \frac{\gamma_2 \left(-r_0 + \frac{1}{2}r_2 r_1\right)}{D} + \frac{\lambda u'_0 \left(1 + \frac{-\frac{1}{2}r_1^2 - \frac{1}{2}r_0^2}{D}\right)}{D} \\ q_{14} = \alpha'_u t_0 + \gamma_2 t_1 + \lambda u'_0 t_2 + u'_0 - \lambda u'_0 \\ q_{21} = \frac{\alpha'_v \left(r_2 + \frac{1}{2}r_1 r_0\right)}{D} + \frac{\lambda v'_0 \left(-r_1 + \frac{1}{2}r_2 r_0\right)}{D} \\ q_{22} = \frac{\alpha'_v \left(1 + \frac{-\frac{1}{2}r_2^2 - \frac{1}{2}r_0^2}{D}\right) + \frac{\lambda v'_0 \left(r_0 + \frac{1}{2}r_2 r_1\right)}{D}}{D} \\ q_{23} = \frac{\alpha'_v \left(-r_0 + \frac{1}{2}r_2 r_1\right)}{D} + \frac{\lambda v'_0 \left(1 + \frac{-\frac{1}{2}r_1^2 - \frac{1}{2}r_0^2}{D}\right)}{D} \\ q_{24} = \alpha'_v t_1 + \lambda v'_0 t_2 + v'_0 - \lambda v'_0 \\ q_{31} = \frac{\mu \left(-r_1 + \frac{1}{2}r_2 r_0\right)}{D} \\ q_{32} = \frac{\mu \left(r_0 + \frac{1}{2}r_2 r_1\right)}{D} \\ q_{33} = \mu \left(1 + \frac{-\frac{1}{2}r_1^2 - \frac{1}{2}r_0^2}{D}\right) \\ q_{34} = 1 + \mu t_2 - \mu \end{array} \right.$$

avec :

$$D = 1 + \frac{1}{4}r_0^2 + \frac{1}{4}r_1^2 + \frac{1}{4}r_2^2$$

A.4 Condition d'existence d'une relation homographique dans le cas de la projection para-perspective

On se place ici dans le cas où le paramètre γ n'est pas nul.
Le calcul s'effectue en Maple :

```
tasmanie-31% maple
|\~/|   Maple V Release 5 (INRIA)
._|\|   |/_|. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
\ MAPLE / reserved. Maple and Maple V are registered trademarks of
<_--- _---> Waterloo Maple Inc.
|           Type ? for help.

>
> with(linalg):
> with(linalg1):
>
> w :=vector([r0, r1, r2]):
> pR :=rot3d(w): # calcul de la matrice de rotation (exponentielle de w)
> t :=vector([t0, t1, t2]):
> R :=stackmatrix(concat(pR,t),array([0,0,0,1])):
bytes used=1000036, alloc=851812, time=0.74
> A1 := matrix([
>         [au, g1, bu, u0],
>         [0,  av, bv, v0],
>         [0,  0,  0,  1]]):
> Ap := matrix([
>         [au2, g2,  bu2, u02],
```

```

> [0, av2, bv2, v02],
> [0, 0, 0, 1]]):
> A2 := map(evalc,evalm(Ap&*R)):
>
> eq:=evalm((concat(col(A2,1..2),col(A2,\
> 4))&*inverse(concat(col(A1,1..2),col(A1,4))&*col(A1,3) - col(A2,3)):
> eq1:= subs(bv=-av*Y0/Z0, bu=-au*X0/Z0-\
> g1*X0/Y0, bv2=-av2*Y0/Z0, bu2=-au2*X0/Z0-g2*X0/Y0,evalm(eq)):
> ## cas general, on va chercher une solution proche de la solution
> ## correspondant au cas gamma1=gamma2=0
> qw1:=numer(factor(subs(r0=t*(X0/Z0+e0),r1=t*(Y0/Z0+e1),r2=t,evalm(eq1[1])))):
> qw2:=numer(factor(subs(r0=t*(X0/Z0+e0),r1=t*(Y0/Z0+e1),r2=t,evalm(eq1[2])))):
> se0:=solve(qw1,e0):
> se11:=solve(subs(e0=se0[1],qw2),e1):
> se12:=solve(subs(e0=se0[2],qw2),e1):
> se011:=factor(subs(e1=se11[1],se0[1])):
> se012:=factor(subs(e1=se11[2],se0[1])):
> se021:=factor(subs(e1=se12[1],se0[2])):
> se022:=factor(subs(e1=se12[2],se0[2])):

```

On obtient ainsi 4 solutions différentes ce auquel on pouvait s'attendre vue la définition de e_0 et e_1 . L'introduction de γ induit une perturbation sur la direction du regard.

A.5 Génération des contraintes moléculaires dans le cas de 2 vues

```

#####
# dlingran@sophia.inria.fr, Copyright (C) 1999. All rights reserved. #
#                                                                    #
# 2 frames --                                                         #
#                                                                    #
#####

with(linalg):
with(linalg1):

#
# Specific uncalibrated motion models : N vues, i = {0 .. N-1}
# - Known values are prefixed with _
# - The last item corresponds to a DEFAULT value and is the most general case
#
'spec/Cts' := subs(
{x = vector([1,0,0]), y = vector([0,1,0]), z = vector([0,0,1])},

[

# Projection models : [1] ortho - [2] para - [3] persp.
p = [
{lambda=0, mu=0},
{lambda=1, mu=0},
{lambda=1, mu=1}],

# Intrinsic parameters

```



```

# Skew-Factor gama(i) : [1] known (zero) - [2] constant and known -
# [3] constant (unknown) - [3] general
g = [
  {gama=(i->0)},
  {gama=(i->_gama_0)},
  {gama=(i->gama_0)},
  {gama=(i->gama(i))}],

# Focal length alpha_v(i) = f(i) : [1] known (one) -
# [2] constant (unknown) - [3] general
f = [
  {alpha_v=(i->1)},
  {alpha_v=(i->f_0)},
  {alpha_v=(i->f(i))}],

# Scale factor alpha_u(i)/alpha_v(i) : [1] known (one) -
# [2] constant (unknown) - [3] general
s = [
  {alpha_u=(i->alpha_v(i))},
  {alpha_u=(i->ka*alpha_v(i))},
  {alpha_u=(i->au(i))}],

# Beta parameter (for the paraperspective projection) : [1] nul (other proj)
# [2] known and constant [3] constant (unknown) [4] general
b = [
  {beta_v=(i->0)},
  {beta_v=(i->_b_0)},
  {beta_v=(i->b_0)},
  {beta_v=(i->bv(i))}],

# Scale factor beta_u(i)/beta_v(i) : [1] known (one) -
# [2] constant (unknown) - [3] general
B = [
  {beta_u=(i->beta_v(i))},
  {beta_u=(i->_kb*beta_v(i))},
  {beta_u=(i->kb*beta_v(i))},
  {beta_u=(i->bu(i))}],

# Image center i.e. principal point (u0(i), v0(i)) : [1] known (zero) -
# [2] constant (unknown) - [3] general
c = [
  {u0=(i->0), v0=(i->0)},
  {u0=(i->u0_0), v0=(i->v0_0)},
  {u0=(i->u0(i)), v0=(i->v0(i))}],

# Extrinsic parameters : rigid displacement and (optional) planar normal
# - With  $M(i) = R(i) M(0) + t(i)$  we have:
# *  $R(i) = I + (2 a)/(1 + a^2) [u]_x/||u|| + a^2/(1 + a^2) [u]_x^2/||u||^2$ 
# *  $t(i)$  defined up to a scale factor
# - When  $M(i)$  in  $P(i) \Leftrightarrow n(i)^T M(i) = 1$  while  $||n(i)||=1/d(P(i),0)$  we have:
# *  $n(i) = R(i) n(0) / (1 + (t(i)^T R(i) n(0)))$ 

# Translation t(i) : [1] zero -
# [2] general
T = [
  {t=(i->vector(3,0))},
  {t=(i->if i=0 then vector(3,0))}

```

```

    else vector([tx(i), ty(i), tz(i)])
    fi]],

# Translation direction (Focus of expansion) constraints
t = [
  {'&.'(t(i),x)=1,'&.'(t(i),y)=0,'&.'(t(i),z)=0},
  {'&.'(t(i),x)=1,'&.'(t(i),y)=0},
  {'&.'(t(i),x)=1,'&.'(t(i),z)=0},
  {'&.'(t(i),x)=1},
  {'&.'(t(i),y)=1,'&.'(t(i),z)=0,'&.'(t(i),x)=0},
  {'&.'(t(i),y)=1,'&.'(t(i),z)=0},
  {'&.'(t(i),y)=1,'&.'(t(i),x)=0},
  {'&.'(t(i),y)=1},
  {'&.'(t(i),z)=1,'&.'(t(i),x)=0,'&.'(t(i),y)=0},
  {'&.'(t(i),z)=1,'&.'(t(i),x)=0},
  {'&.'(t(i),z)=1,'&.'(t(i),y)=0},
  {'&.'(t(i),z)=1},
  {}],

# Rotation axis r(i) : [1] general
r = [{r = (i->2*tan(a(i)/2)/sqrt(evalc(dotprod(u(i),u(i)))))*u(i)}],

# Rotation axis u(i) : [1] general
U = [{u = (i->vector([ux(i), uy(i),uz(i)]))}],

# Rotation matrices : [1] identity - [2] symmetry - [3] first order -
# [4] second order - [5] general
R = [
  {R=(i->evalm(array(1..3,1..3,identity)))},
  {R=(i->evalm(array(1..3,1..3,identity)
    +2/evalc(dotprod(u(i),u(i)))*evalm(tilde(evalm(u(i)))&*tilde(evalm(u(i))))))},
  {R=(i->evalm(array(1..3,1..3,identity)+tilde(evalm(r(i)))))},
  {R=(i->evalm(array(1..3,1..3,identity)
    +tilde(evalm(r(i)))+evalm(tilde(evalm(r(i)))&*tilde(evalm(r(i))))/2))},
  {R=(i->evalm(array(1..3,1..3,identity)
    +(tilde(evalm(r(i)))+evalm(tilde(evalm(r(i)))&*tilde(evalm(r(i))))/2)
    /(1+evalc(dotprod(r(i),r(i))))))}],

# Rotation axis direction constraints
u = [
  {'&.'(u(i),y)=1,'&.'(u(i),z)=0,'&.'(u(i),x)=0},
  {'&.'(u(i),y)=1,'&.'(u(i),z)=0},
  {'&.'(u(i),y)=1,'&.'(u(i),x)=0},
  {'&.'(u(i),y)=1},
  {'&.'(u(i),y)=-1,'&.'(u(i),z)=0,'&.'(u(i),x)=0},
  {'&.'(u(i),y)=-1,'&.'(u(i),z)=0},
  {'&.'(u(i),y)=-1,'&.'(u(i),x)=0},
  {'&.'(u(i),y)=-1},
  {'&.'(u(i),z)=1,'&.'(u(i),x)=0,'&.'(u(i),y)=0},
  {'&.'(u(i),z)=1,'&.'(u(i),x)=0},
  {'&.'(u(i),z)=1,'&.'(u(i),y)=0},
  {'&.'(u(i),z)=1},
  {'&.'(u(i),z)=-1,'&.'(u(i),x)=0,'&.'(u(i),y)=0},
  {'&.'(u(i),z)=-1,'&.'(u(i),x)=0},
  {'&.'(u(i),z)=-1,'&.'(u(i),y)=0},
  {'&.'(u(i),z)=-1},

```

```

{'&.'(u(i),x)=1,'&.'(u(i),y)=0,'&.'(u(i),z)=0},
{'&.'(u(i),x)=1,'&.'(u(i),y)=0},
{'&.'(u(i),x)=1,'&.'(u(i),z)=0},
{'&.'(u(i),x)=1},
{'&.'(u(i),x)=-1,'&.'(u(i),y)=0,'&.'(u(i),z)=0},
{'&.'(u(i),x)=-1,'&.'(u(i),y)=0},
{'&.'(u(i),x)=-1,'&.'(u(i),z)=0},
{'&.'(u(i),x)=-1}],

# Relative orientation between rotation and translation :
# [1] fixed axis rotation - [2] screw rotation - [3] general
Z = [
{'&x'(u(i),t(i))=0},
{'&.'(u(i),t(i))=0},
{}],

# Direction of rotation axis [1] parallel - [2] ortho - [3] general
W = [
{'&x'(u(i),vector([-ux,_uy,_uz]))=0},
{'&.'(u(i),vector([-ux,_uy,_uz]))=0},
{}],

# Direction of translation [1] parallel - [2] ortho - [3] general
D = [
{'&x'(t(i),vector([-tx,_ty,_tz]))=0},
{'&.'(t(i),vector([-tx,_ty,_tz]))=0},
{}],

# Angle a(i) = tan(theta/2) : [1] theta=zero -[2] known
# [3] general
a = [
{a=(i->0)},
{a=(i->if i=0 then 0 else _a(i) fi)},
{a=(i->if i=0 then 0 else a(i) fi)},
{a=(i->1)}],

# (Optional) plane normal direction constraints in view 0
n = [
{nx=0,ny=0,nz=0},
{ny=0,nz=0},
{nx=0,nz=0},
{nx=0,ny=0},
{nx=0},
{ny=0},
{nz=0},
{nx=_Nx,ny=_Ny,nz=_Nz},
{}],

l = [
{'&x'(array([nx,ny,nz]),t(i))=0},
{'&.'(array([nx,ny,nz]),t(i))=0},
{}],

m = [
{'&x'(array([nx,ny,nz]),u(i))=0},

```

```
{'&.(array([nx,ny,nz]),u(i))=0},  
{}],
```

```
NULL]):
```

```
# Test the coherence of the data structure  
if not type('spec/Cts', list(name=list(set('=')))) then  
  ERROR('BUG in data structure !!!!!')  
fi:  
  
#  
# Hierarchical tree of constraints  
# - The upper is the most general model  
# - The lower is the most specific model (no retinal displacement)  
# Equations lower=upper are on each model item index  
# - Default items are linearly sorted up to the most general case  
  
'spec/Tree/default' := proc(u)  
  global 'spec/Cts':  
  u = {'i=i+1'$i=1..nops(  
    op(1, map((c,u)->if op(1,c)=u then op(2,c) fi, 'spec/Cts', u))-1}  
end:  
  
'spec/Tree' := [  
  
  'spec/Tree/default'(p),  
  
  'spec/Tree/default'(g),  
  
  'spec/Tree/default'(f),  
  
  'spec/Tree/default'(s),  
  
  'spec/Tree/default'(b),  
  
  'spec/Tree/default'(B),  
  
  'spec/Tree/default'(c),  
  
  'spec/Tree/default'(T),  
  
  f = {  
    1=2, 1=3, 2=4, 3=4,  
    5=6, 5=7, 6=8, 7=8,  
    9=10, 9=11, 10=12, 11=12,  
    13=14, 13=15, 14=16, 15=16,  
    17=18, 17=19, 18=20, 19=20,  
    21=22, 21=23, 22=24, 23=24},  
  
  R = {  
    1=2, 2=5,  
    1=3, 3=4, 4=5},  
  
  'spec/Tree/default'(r),  
  'spec/Tree/default'(U),
```

```

u = {
  1=2, 1=3, 2=4, 3=4,
  5=6, 5=7, 6=8, 7=8,
  9=10, 9=11, 10=12, 11=12,
  13=14, 13=15, 14=16, 15=16,
  17=18, 17=19, 18=20, 19=20,
  21=22, 21=23, 22=24, 23=24},

Z = {1=3, 2=3},

W = {1=3, 2=3},

D = {1=3, 2=3},

'spec/Tree/default'(a),

n = {1=2, 1=3, 1=4, 2=5, 2=6, 3=6, 3=7, 4=5, 4=7, 5=9, 6=9, 7=9, 8=9},

l = {1=3, 2=3},

m = {1=3, 2=3},

NULL]:

#
# Basic functions on the model hierarchy
# - Models are referenced by their index, a list of integers

# [1] Model name and index functions

# - Return the number of defined models
'spec/nops' := proc()
  global 'spec/Cts':
  convert(map(c->nops(op(2,c)), 'spec/Cts'), '*')
end:

lprint(cat('They are ', 'spec/nops'(), ' models : oh la la !')):

# - Return true if m is a correct model index or name
'spec/type' := proc(m)
  option remember:
  global 'spec/Cts':
  type(m, list(integer)) and (nops(m) = nops('spec/Cts')) and
  not member(false, maps2((m,c)->(m>0) and (m<=nops(op(2,c))), m, 'spec/Cts'))
end:

'type/spec/index' := op('spec/type'):

# - Return the index of a given model name or NULL if undefined
'spec/index' := proc(m_name :: string)
  global 'spec/Cts':
  local m, j, p, k, v:
  m := 'spec/top'():
  j := 1: while j <= length(m_name) do
    p := sscanf(substring(m_name,j..length(m_name)), '%[a-zA-Z]%d');
    if nops(p) = 2 and

```

```

    member(convert(p[1],symbol), map(u->op(1,u), 'spec/Cts', 'k') and
    p[2] > 0 and p[2] <= nops(op(2,'spec/Cts'[k])) then
    m[k] := p[2]:
    j := j + length(cat(p[1],p[2]))
    else RETURN(NULL) fi:
od:
m
end:

# - Return the name of a model for a given index
'spec/name' := proc(m :: 'spec/index')
global 'spec/Cts':
cat(op(maps2((m,c)->if m<nops(op(2,c)) then op(1,c),m fi, m, 'spec/Cts'))
end:

# [2] Acces functions in the model hierarchy

# - Return the lower model index
'spec/bottom' := proc()
global 'spec/Cts':
[1$nops('spec/Cts')]
end:

# - Return the upper model index
'spec/top' := proc()
global 'spec/Cts':
['nops(op(2,'spec/Cts'[k]))'$k=1..nops('spec/Cts')]
end:

# - Return the set of the next more general models indexes in the hierarchy
'spec/up' := proc(m :: 'spec/index')
global 'spec/Tree':
maps2((i,b)->map(
(b,i)->if op(1,b)=i then op(2,b) fi, op(2,b) union {i=i}, i),
m, 'spec/Tree')
end:

# - Return the set of the next less general models indexes in the hierarchy
'spec/down' := proc(m :: 'spec/index')
global 'spec/Tree':
maps2((i,b)->map(
(b,i)->if op(2,b)=i then op(1,b) fi, op(2,b) union {i=i}, i),
m, 'spec/Tree')
end:

# [3] Functions of a given model

'spec/resolEq' := proc(u)
local v:
v:=evalm(op(1,u)-op(2,u)):
if type(v, array) then
op(convert(v,set));
else
v;
fi:
end:

```

```

# - Return the substitution equations for a given model
'spec/eqs' := proc(m :: 'spec/index')
global 'spec/Cts':
option remember:
local e, f, c, v, s, l, s2:
# Equations for this model
e := map(evalc,{op(maps2((m,c)->op(op(m,op(2,c))), 'm', 'spec/Cts'))}):
# Distinction between Functional substitutions and Constraints
f := map(u->if type(u, '=' and type(op(2,u), procedure) then u fi, e):
c := e minus f:
c := subs(f, c):
c := map('spec/resolEq', c):
# Resolution of the constraints
v := map(u->if substring(convert(u,string), 1)<>"_" then u fi, indets(c)):
s := {solve(c, v)}:
if nops(s)<>1 or nops(op(1,s))<>nops(v) then
RETURN(NULL);
# ERROR(cat('Unable to compute equations of model ', 'spec/name'(m),
# ' running: ', convert(Solve(c,v), string)))
fi:
s:= op(1,s);
# Maintenant on remplace les ux(i) par exemple de s dans f (3 fois pour les substitutions)
s2:=map(u->if(type(op(1,u),function)) then unapply(op(1,u),i) = unapply(op(2,u),i) fi,s);
f:=subs(s2,f);
f:=subs(s2,f);
f:=subs(s2,f);

l := {op(f),op(s)};
# ici, on compose les fonctions (2 fois car il existe R ( r ( u ) ) )
l := map(u->op(1,u) = subs(l,op(2,u)),l);
l := map(u->op(1,u) = subs(l,op(2,u)),l);
RETURN(l);
end:

##
#map(u->op(1,u) = subs(q,op(2,u)),q);
##

# - Map a given model on an object (matrix, vector, expression)
'spec/map' := proc(m :: 'spec/index', object)
option remember:
local r:
# Dealing with named arrays
if type(object, string) and type(object, array) then
lprint('in spec map, just return');
RETURN(procname(m, op(object)))
fi:
# Substitute and eval
r := subs('spec/eqs'(m), object):
if type(object, array) then
map(eval, r)
else
eval(r)
fi

```

```

end:

# - Return the free variables of an object for a given model
'spec/free' := proc(m :: 'spec/index', object)
  indets('spec/map'(m, convert(object, set)))
end:

# Intrinsic parameters matrix

'spec/Pi' := i->array([
  [ alpha_u(i),   gama(i), lambda * beta_u(i) + mu * u0(i), (1-mu) * u0(i)],
  [           0, alpha_v(i), lambda * beta_v(i) + mu * v0(i), (1-mu) * v0(i)],
  [           0,           0,                               mu,          (1-mu)]
]):

# Extrinsic parameters matrix

'spec/Pe' := i->stackmatrix(
  concat(evalm(R(i)),evalm(t(i))),
  array([0,0,0,1])):

###
###
###
'spec/gener/persp' := proc()
  local ig, iff, is, iC, it, iR, iu, iZ, theindex:

  for ig in {1,2,4} do
    for iff in {1,3} do
      for is in {1,3} do
        for iC to 3 do
          for it to 12 do
            for iR to 5 do
              for iu in {1,2,3,4,9,10,11,12,17,18,19,20} do
                for iZ to 3 do
                  theindex := cat('p3g',ig,'f',iff,'s',is,'b1B1c',iC,'T2t',it,'r1U1R',iR,
                    'u',iu,'Z',iZ,'W3D3a2n9l3m3'):
                  print(theindex);
                  'spec/analyse'(convert(theindex,string)):
                od
              od
            od
          od
        od
      od
    od
  od:
end:

'spec/gener/para' := proc()
  local ig, iff, is, iC, it, iR, iu, iZ, theindex:

  for ig in {1,2,4} do
    for iff in {1,3} do
      for is in {1,3} do
        for iC to 3 do

```



```

for it to 12 do
  for iR to 5 do
    for iu in {1,2,3,4,9,10,11,12,17,18,19,20} do
      for iZ to 3 do
        theindex := cat('p2g',ig,'f',iff,'s',is,'b1B1c',iC,'T2t',it,'r1U1R',iR,
          'u',iu,'Z',iZ,'W3D3a2n9l3m3'):
        print(theindex);
        'spec/analyse'(convert(theindex,string)):
      od
    od
  od
od
od:
end:

'spec/gener/ortho' := proc()
  local ig, iff, is, iC, it, iR, iu, iZ, theindex:

  for ig in {1,2,4} do
    for iff in {1,3} do
      for is in {1,3} do
        for iC to 3 do
          for iR from 2 to 5 do
            for iu in {1,2,3,4,17,18,19,20} do
              for iZ to 3 do
                theindex := cat('p1g',ig,'f',iff,'s',is,'b1B1c',iC,'T1t13r1U1R',iR,
                  'u',iu,'Z',iZ,'W3D3a2n9l3m3'):
                print(theindex);
                'spec/analyse'(convert(theindex,string)):
              od
            od
          od
        od
      od
    od
  od:
  for ig in {1,2,4} do
    for iff in {1,3} do
      for is in {1,3} do
        for iC to 3 do
          for it to 12 do
            for iR from 2 to 5 do
              for iu in {1,2,3,4,17,18,19,20} do
                for iZ to 3 do
                  theindex := cat('p1g',ig,'f',iff,'s',is,'b1B1c',iC,'T2t',it,'r1U1R',iR,
                    'u',iu,'Z',iZ,'W3D3a2n9l3m3'):
                  print(theindex);
                  'spec/analyse'(convert(theindex,string)):
                od
              od
            od
          od
        od
      od
    od
  od
end:

```

```

    od
  od
od:end:

'spec/analyse' := proc(m_name :: string)
local m, R1, t1, Pe1, Pi0, Pi1, q, mtp, B1, eq, maListe, F, Fs, fd0,fd1,i,j,u,v,up,vp, dof,
      x1, x2, x3, x4, x5, x6, x7, x8, x9, f11, f12, f13, f21, f22, f23, f31, f32, f33:

lprint('in analyse');
fd0 := fopen("Fnull-persp2.test",APPEND):
fd1 := fopen("Fother-persp2.test",APPEND):
m := 'spec/index'(m_name):
lprint('after spec index');
if m = NULL then
  lprint(m_name,' Desole ce nom est invalide !')
else
  if 'spec/eqs'(m) <> NULL then
    R1 := 'spec/map'(m, R(1)):
    t1 := 'spec/map'(m, t(1)):
    Pe1 := stackmatrix(concat(R1,t1),array([0,0,0,1])):
    Pi0 := 'spec/map'(m, 'spec/Pi'(0)):
    Pi1 := 'spec/map'(m, 'spec/Pi'(1)):
    q := multiply(Pi1,Pe1):
    mtp := transpose(matrix([[up],[vp],[1]])):
    B1 := stackmatrix(concat(concat(vector([u,v,1]),vector([0,0,0])),Pi0
      ,concat(concat(vector([0,0,0]),vector([up,vp,1])),q)):
    eq:= expand(sort(det(B1) - expand(multiply(mtp,
      multiply(matrix([[f11, f12, f13], [f21, f22, f23], [f31, f32, f33]],
        vector([u,v,1]))[1])))):
    maListe:= [coeffs(eq, {u,v,up,vp})]:
    assign(solve({op(maListe)},{f11,f12,f13,f21,f22,f23,f31,f32,f33})):
    F := array([f11, f12, f13, f21, f22, f23, f31, f32, f33]):
    Fs := array([x1, x2, x3, x4, x5, x6, x7, x8, x9]):
    for i to 9 do
      if F[i] = 0 then
        Fs[i]:=0;
      else
        if i < 9 then
          for j from i+1 to 9 do
            if factor(expand(F[i]-F[j]))=0
              then Fs[j]:=Fs[i];
            else if factor(expand(F[i]+F[j]))=0
              then Fs[j]:=-Fs[i];
            fi
          od
        fi
      fi
    od:
    dof:=nops(indets({op(convert(Fs,list))})):
    if dof = 0 then
      fprintf(fd0,"%s\n", m_name);
    else if dof < 9 then
      fprintf(fd1,"%d :[%s %s %s %s %s %s %s %s %s ] ## %s\n", dof,
        convert(Fs[1],string),convert(Fs[2],string),convert(Fs[3],string),
        convert(Fs[4],string),convert(Fs[5],string),convert(Fs[6],string),

```

```

convert(Fs[7],string),convert(Fs[8],string),convert(Fs[9],string), m_name);

fi
fi

fi

fi:
# fi:
fclose(fd0):
fclose(fd1):
end:

'spec/gener/persp'():
# 'spec/gener/ortho'():
#'spec/gener/test'():
#'spec/gener/para'():

```

Les particuliers correspondants à des matrices fondamentales nulles sont répertoriés dans le fichier "Fnull.test" tandis que les cas pour lesquels on obtient des formes simplifiées non nulles sont dans le fichier "Fother.test", pour lequel chaque ligne correspond à un cas et contient le nombre de variables mise en jeu, le nom et la forme simplifiée.

Pour les cas concernant la projection perspective et plus particulièrement formés de la sorte :

$p3g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[1-5]u[1-4;9-12;17-20]Z[1-3]W3D3a2n913m3$

on obtient 77760 cas particuliers.

Pour des raisons de vitesses de calcul, on a considéré le sous-ensemble de cas suivant :

$p3g1f1s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[1-5]u[1-4;9-12;17-20]Z[1-3]W3D3a2n913m3$

soit, 1296 cas, parmi lesquels 7087 proposent une matrice fondamentale simplifiée.

Nous montrons ci-dessous un court extrait du fichier obtenu :

```

1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g2f1s1b1B1c1T2t1r1U1R1u2Z2W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u1Z2W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u1Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u2Z2W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u2Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u19Z2W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u19Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u20Z1W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u20Z2W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R1u20Z3W3D3a2n913m3
2 : [0 0 0 0 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u1Z2W3D3a2n913m3
2 : [0 0 0 0 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u1Z3W3D3a2n913m3
3 : [0 0 0 0 0 x6 x7 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u2Z2W3D3a2n913m3
3 : [0 0 0 0 0 x6 x7 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u2Z3W3D3a2n913m3
3 : [0 0 0 0 0 x5 x6 0 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u3Z2W3D3a2n913m3
3 : [0 0 0 0 0 x5 x6 0 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u3Z3W3D3a2n913m3
5 : [0 0 0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u4Z2W3D3a2n913m3
5 : [0 0 0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u4Z3W3D3a2n913m3
2 : [0 0 0 0 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u9Z2W3D3a2n913m3
2 : [0 0 0 0 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u9Z3W3D3a2n913m3
3 : [0 0 0 0 0 x5 x6 0 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u10Z2W3D3a2n913m3
3 : [0 0 0 0 0 x5 x6 0 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u10Z3W3D3a2n913m3
3 : [0 0 0 0 0 x4 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u11Z2W3D3a2n913m3
3 : [0 0 0 0 0 x4 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u11Z3W3D3a2n913m3
5 : [0 0 0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u12Z2W3D3a2n913m3
5 : [0 0 0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u12Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u17Z1W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u17Z2W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u17Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u18Z1W3D3a2n913m3
3 : [0 0 0 0 0 x4 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u18Z2W3D3a2n913m3

```

```

3 : [0 0 0 x4 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u18Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u19Z1W3D3a2n913m3
3 : [0 0 0 0 0 x6 x7 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u19Z2W3D3a2n913m3
3 : [0 0 0 0 0 x6 x7 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u19Z3W3D3a2n913m3
1 : [0 0 0 0 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u20Z1W3D3a2n913m3
5 : [0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u20Z2W3D3a2n913m3
5 : [0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R2u20Z3W3D3a2n913m3
2 : [0 0 0 x4 0 x6 0 -x6 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R3u1Z2W3D3a2n913m3
4 : [0 0 0 x4 x5 x6 x7 -x6 x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R3u20Z2W3D3a2n913m3
4 : [0 0 0 x4 x5 x6 x7 -x6 x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R3u20Z3W3D3a2n913m3
3 : [0 0 0 x4 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u1Z2W3D3a2n913m3
3 : [0 0 0 x4 0 x6 0 x8 0 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u1Z3W3D3a2n913m3
5 : [0 0 0 x4 x5 x6 x7 x8 x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u2Z2W3D3a2n913m3
5 : [0 0 0 x4 x5 x6 x7 x8 x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u2Z3W3D3a2n913m3
5 : [0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u3Z2W3D3a2n913m3
5 : [0 0 0 x4 x5 x6 x7 x8 -x5 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u3Z3W3D3a2n913m3
6 : [0 0 0 x4 x5 x6 x7 x8 x9 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u4Z2W3D3a2n913m3
6 : [0 0 0 x4 x5 x6 x7 x8 x9 ] ## p3g1f1s1b1B1c1T2t1r1U1R4u4Z3W3D3a2n913m3

```

Pour les cas concernant la projection orthographique, et plus précisément les cas de la forme :

$$p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T1t13r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n913m3$$

et

$$p1g[1-2;4]f[1;3]s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n913m3$$

on obtient 44928 cas. Pour les mêmes raisons de temps de calcul que précédemment, on ne considère que les cas suivants :

$$p1g1f1s[1;3]b1B1c[1-3]T1t13r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n913m3$$

et

$$p1g1f1s[1;3]b1B1c[1-3]T2t[1-12]r1U1R[2-5]u[1-4;17-20]Z[1-3]W3D3a2n913m3$$

soit 7488 cas, parmi lesquels 971 présentent une matrice fondamentale nulle (ce sont donc des cas d'homographies) et 6421 une matrice fondamentale simplifiée.

Annexe B

Analyse détaillée de quelques cas particuliers.

Cette annexe présente une analyse détaillée de quelques cas particuliers et qui est une partie de (Viéville et Lingrand, 1999).

B.1 Considering screw rotations with fixed camera parameters.

Surprisingly perhaps it would be useful to start analyzing the case of a “screw rotation” i.e. the situation in which **the translation is aligned with rotation axis**.

As easily demonstrated¹ this situation is characterized by the fact that :

$$\mathbf{F} \mathbf{e} = 0 \quad (\text{B.1})$$

In other words $\mathbf{e} = \mathbf{e}'$, i.e. the two epipoles are the same. This is a sufficient and necessary condition.

Let us complete this analysis by studying the rank of the F -matrix, Why such a choice? Because, since $\mathbf{F} = \mathbf{A}'^T \tilde{\mathbf{t}} \mathbf{R} \mathbf{A}^{-1}$ its rank must be the same as for the *essential matrix* $\mathbf{E} = \tilde{\mathbf{t}} \mathbf{R}$ studied in (Maybank, 1990) since it differs only up to a change of coordinates. Therefore these rank properties must be only related to the rigid displacement parameters and not the calibration parameters, and all results acquired by (Maybank, 1990) are still true in our case.

For instance, for a screw rotation, the symmetric part of the F -matrix is not of full rank since $(\mathbf{F} + \mathbf{F}^T) \mathbf{e} = 0$. More precisely, as demonstrated in (Maybank, 1990) in the calibrated case, this result:

$$\det(\mathbf{F} + \mathbf{F}^T) = 0 \Leftrightarrow \begin{cases} \mathbf{u} & - & \mathbf{t} \\ \mathbf{u} & || & \mathbf{t} \end{cases} \quad (\text{B.2})$$

in other words a necessary and sufficient for the F -matrix to correspond either a “fixed axis” or a “screw rotation” is that the rank of the symmetric part of \mathbf{F} is less than three.

A step further, the matrix \mathbf{F} is entirely symmetric if and only if², we have a screw rotation for which the rotation has an angle of $\Pi/2$.

Summarizing these different possibilities we have the following results, depending on the rank of \mathbf{F} , its symmetric part $\mathbf{F}_s = \frac{\mathbf{F} + \mathbf{F}^T}{2}$ and its skew-symmetric part $\mathbf{F}_a = \frac{\mathbf{F} - \mathbf{F}^T}{2}$:

Rank	\mathbf{F}	case
≤ 2	\mathbf{F} \mathbf{F}_s \mathbf{F}_a	always true fixed axis or screw rotation (always true !)
≤ 1	\mathbf{F} \mathbf{F}_s \mathbf{F}_a	pure rotation pure translation quarter-turn screw rotation
$= 0$	\mathbf{F} \mathbf{F}_s \mathbf{F}_a	pure rotation pure translation quarter-turn screw rotation

1. If we write $\mathbf{t} = \alpha \mathbf{u} + \mathbf{v}$ with $\mathbf{v} = \mathbf{t} - \underbrace{(\mathbf{t}^T \mathbf{u})}_{\alpha} \mathbf{u}$ while $\mathbf{v} \perp \mathbf{u}$, which is a unique decomposition of \mathbf{t} , we obtain $\mathbf{F} \mathbf{e} = \mathbf{A}^{-1T} \tilde{\mathbf{t}} \mathbf{R} \mathbf{t} \mathbf{A}^{-1} = \alpha \mathbf{A}^{-1T} [(1 - \cos(\theta)) \mathbf{u} \wedge \mathbf{v} - \sin(\theta) \mathbf{v}] \mathbf{A}^{-1}$. It is easy to see that it vanishes if and only if $\mathbf{v} = 0$, as expected.

2. If we write $\mathbf{a} = (1 + \cos(\theta)) \mathbf{t} + \sin(\theta) \mathbf{t} \wedge \mathbf{u} - (1 - \cos(\theta)) (\mathbf{t}^T \mathbf{u}) \mathbf{u}$ we obtain: $\mathbf{F} - \mathbf{F}^T = \mathbf{A}^{-1T} \tilde{\mathbf{a}} \mathbf{A}^{-1}$ and \mathbf{F} is entirely symmetric if and only if $\mathbf{a} = 0$ which in turn is true if and only if $\mathbf{t} \parallel \mathbf{u}$ and $\theta = \Pi/2$.

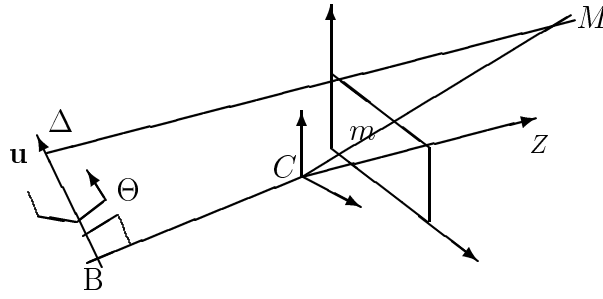


FIG. B.1 – *Notations for a fixed axis rotation*

One important thing is to mention here: these properties are true only considering **real** displacements, i.e. there exist translations and rotations with complex components which violate these rules.

Unfortunately, if $\mathbf{A} \neq \mathbf{A}'$, these results are not true anymore. Furthermore, in this case, the F -matrix is not constrained by the screw-rotation equation but still in a general form.

Considering \mathbf{H}_∞ i.e. the case of a pure rotation, the constraint that the translation is orthogonal to the axis of rotation has no effect on its parameterization since it does not depends on the translation.

Considering a H -matrix this unique constraints does not induce any constraint on its components already function of many other parameters.

B.2 Considering fixed axis rotations with fixed camera parameters.

Let us now consider that the elementary displacement of the camera is a fixed axis rotation, while intrinsic parameters are constant. It is known (Beardsley et al., 1995; Viéville, 1994; Viéville et al., 1995b) that this situation is characterized by the fact that the translation is perpendicular to the axis of rotation.

We use the notations of figure B.1. We consider a vector \mathbf{u} aligned with the rotation axis Δ and the point B on the axis Δ , such that $\mathbf{C} = \vec{CB} - \mathbf{u}$. The rotation angle is θ .

From equations (1.7), (1.14) and (1.17) in this case, the F -matrix is of the rather heavy form:

$$\mathbf{F} \equiv \|\mathbf{C}\| \begin{bmatrix} \sin(\theta) \tilde{\mathbf{f}}_0 \\ +(1 - \cos(\theta)) [\mathbf{f}_1 \mathbf{f}_2^T + \mathbf{f}_2 \mathbf{f}_1^T] \end{bmatrix} \quad (\text{B.3})$$

with :

$$\begin{aligned} \mathbf{f}_0 &= \lambda^2 \frac{2}{\det(\mathbf{A})} \mathbf{A} \frac{\mathbf{C} \wedge \mathbf{u}}{\|\mathbf{C}\|} \\ \mathbf{f}_1 &= \lambda \mathbf{A}^{-1T} \mathbf{u} \\ \mathbf{f}_2 &= \lambda \mathbf{A}^{-1T} \frac{\mathbf{C} \wedge \mathbf{u}}{\|\mathbf{C}\|} \end{aligned}$$

while we have:

$$\mathbf{f}_0^T \mathbf{f}_1 = 0 \quad (\text{B.4})$$

which is the only constraint between these three vectors³. Knowing the rotation angle θ and the matrix \mathbf{F} up to a scale factor, it is then straightforward to recover \mathbf{f}_0 which is the vector associated to the skew-symmetric part of \mathbf{F} .

Similarly $\mathbf{u}_1 = \left(\frac{\mathbf{f}_1}{\|\mathbf{f}_1\|} + \frac{\mathbf{f}_2}{\|\mathbf{f}_2\|} \right)$ and $\mathbf{u}_2 = \left(\frac{\mathbf{f}_1}{\|\mathbf{f}_1\|} - \frac{\mathbf{f}_2}{\|\mathbf{f}_2\|} \right)$ are the eigen-vectors associated respectively with the positive eigen-value and negative eigen-value of the symmetric part of \mathbf{F} , the third eigen-value being zero as induced by equation (B.2). We then obtain $\mathbf{f}_1 \equiv \mathbf{u}_1 + \mathbf{u}_2$ and $\mathbf{f}_2 \equiv \mathbf{u}_1 - \mathbf{u}_2$, but up to a scale factor only.

We can recover the intrinsic calibration parameters:

$$\begin{cases} u_0 &= f_0^0 - \gamma f_2^0 / f_0^2 \\ v_0 &= f_0^1 - \gamma f_2^1 / f_0^2 \\ f &= \frac{\sqrt{\gamma(f_0^0 f_2^0 + f_0^1 f_2^1 + f_0^2 f_2^2) - \gamma^2((f_2^0)^2 + (f_2^1)^2)}}{f_0^2} \end{cases} \quad (\text{B.5})$$

up to a 1D-indetermination related to γ .

A step further, we obtain two equations for the intrinsic parameters:

$$\begin{cases} f_2^1 (f_0^2 u_0 - f_0^0) - f_2^0 (f_0^2 v_0 - f_0^1) &= 0 \\ f^2 + u_0^2 + v_0^2 - (f_0^0 u_0 + f_0^1 v_0 + \gamma f_2^2) / f_0^2 &= 0 \end{cases} \quad (\text{B.6})$$

with

$$\gamma = \frac{f_2^0 (f_0^2 u_0 - f_0^0) + f_2^1 (f_0^2 v_0 - f_0^1)}{(f_2^0)^2 + (f_2^1)^2}$$

so that we have one linear equation in u_0 and v_0 and direct estimation of f^2 as soon as u_0 and v_0 are estimated.

In addition, we recover the extrinsic calibration parameters:

$$\mathbf{u} \equiv \mathbf{A}^T \mathbf{f}_1 \quad \text{and} \quad \mathbf{C} \equiv \mathbf{u} \wedge \mathbf{A}^{-1} \mathbf{f}_0 \quad (\text{B.7})$$

except $\|\mathbf{C}\| = \|\mathbf{t}\| / \sqrt{2(1 - \cos(\theta))}$ which cannot be recovered since we use a monocular system with a scale factor indetermination.

3. Consider three generic vectors \mathbf{f}_0 , \mathbf{f}_1 and \mathbf{f}_2 such that $\mathbf{f}_0 - \mathbf{f}_1$ and introduce a vector $\mathbf{b} = \gamma \frac{\mathbf{f}_1 \wedge \mathbf{f}_0}{\mathbf{f}_1^T \mathbf{f}_1} - (\frac{1}{2} + \gamma) \frac{\mathbf{f}_2 \wedge \mathbf{f}_0}{\mathbf{f}_1^T \mathbf{f}_2}$ so that $\mathbf{b}^T \mathbf{f}_0 = 0$ and $\mathbf{b}^T (\mathbf{f}_1 \wedge \mathbf{f}_2) = \frac{(f_0^T \mathbf{f}_2)}{2}$ for any γ .

There is always a solution of equation (B.3) with respect to \mathbf{f}_0 , \mathbf{f}_1 and \mathbf{f}_2 for:

$$\begin{aligned} \lambda &= 1 \\ \mathbf{u} &= \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ \frac{\mathbf{C} \wedge \mathbf{u}}{\|\mathbf{C}\|} &= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ A^{-1T} &= (\mathbf{b}, \mathbf{f}_1, \mathbf{f}_2) \end{aligned}$$

as the reader can easily verify.

All these equations are singular if and only if:

$$f_0^2 = \frac{2}{f^2} [C^0 u^1 - C^1 u^0] = 0 \quad (\text{B.8})$$

It corresponds to the fact that (i) the intersection of the optical axis with retinal plane, (ii) the projection of \mathbf{C} and (iii) the principal point are collinear, which can be detected in practice.

Considering the case where the calibration parameters vary, no constraint can be generated to \mathbf{F} , as discussed before.

Considering \mathbf{H}_∞ i.e. the case of a pure rotation, the constraint that the translation is orthogonal to the axis of rotation has no effect on its parameterization since it does not depend on the translation.

Considering a H -matrix this unique constraint does not induce any constraint on its components already function of many other parameters.

B.3 Considering pure translations with fixed camera parameters.

Most of the analysis of this case is very simple and widely known (see for instance in (Shashua et Navab, 1994; Van Gool et al., 1994)), since :

$$\mathbf{F} = \tilde{\mathbf{e}} \quad \text{and} \quad \mathbf{H}_\infty \equiv \mathbf{I} \quad (\text{B.9})$$

and the affine calibration is obvious.

However the characterization the H -matrix has not yet been made. Let us do it now. Using the same equations we easily see that we are looking for a matrix \mathbf{H} of the form:

$$\mathbf{H} \equiv \mathbf{I} + \frac{\mathbf{e}}{\|\mathbf{e}\|} \nu^T \quad (\text{B.10})$$

Considering \mathbf{H}^T we have the following eigen-value decomposition :

$$\begin{cases} \mathbf{u} - \mathbf{e} \Rightarrow \mathbf{H}^T \mathbf{u} = \mathbf{u} \\ \mathbf{u} \equiv \nu \Rightarrow \mathbf{H}^T \mathbf{u} = (1 + \nu^T \frac{\mathbf{e}}{\|\mathbf{e}\|}) \mathbf{u} \end{cases} \quad (\text{B.11})$$

Such a collineation is only defined by five parameters (2 for the projection of the translation defined up to a scale factor and 3 for the parameter of the plane) and can thus be characterized as soon as three generic points are given, since each match generates two equations given by: $\mathbf{m}' \wedge \mathbf{H} \mathbf{m} = 0$, or more precisely:

$$\begin{aligned} & \mathbf{m}' \wedge \mathbf{m} + (\nu^T \mathbf{m}) \left[\mathbf{m}' \wedge \frac{\mathbf{e}}{\|\mathbf{e}\|} \right] = 0 \\ \Leftrightarrow & \begin{cases} |\mathbf{m}, \mathbf{m}', \frac{\mathbf{e}}{\|\mathbf{e}\|}| = 0 \\ (\nu^T \mathbf{m}) = \frac{\mathbf{e}}{\|\mathbf{e}\|}^T \frac{(\mathbf{m}'^T \mathbf{m}') \mathbf{m} - (\mathbf{m}'^T \mathbf{m}) \mathbf{m}'}{\|\mathbf{m}' \wedge \mathbf{m}\|^2} \end{cases} \end{aligned} \quad (\text{B.12})$$

so that given at least two non-stationary points we can calculate $\frac{\mathbf{e}}{\|\mathbf{e}\|}$ and then, given at least three non collinear points we can calculate ν knowing $\frac{\mathbf{e}}{\|\mathbf{e}\|}$.

Reciprocally, *the corresponding H-matrix is characterized by the fact that it has two eigen-values which are equal.*

In order to clarify this last point consider any matrix \mathbf{H} with \mathbf{u}_{1a} and \mathbf{u}_{1b} the eigen-vectors corresponding to the same eigen-value λ_1 of \mathbf{H}^T and \mathbf{u}_0 the eigen-vector corresponding to another eigen-value λ_0 of \mathbf{H}^T . Inverting equation (B.11) we easily obtain :

$$\mathbf{e} \equiv \mathbf{u}_{1a} \wedge \mathbf{u}_{1b} \quad ; \quad \nu = \frac{\lambda_0/\lambda_1 - 1}{\frac{\mathbf{e}}{\|\mathbf{e}\|}^T \mathbf{u}_0} \mathbf{u}_0 \quad (\text{B.13})$$

and verify that \mathbf{H} is of the form of equation (B.10). Therefore \mathbf{H} is of the form of equation (B.10) if and only if it has two eigen-values which are equal.

It is thus possible to identify if a given H -matrix corresponds to a pure translation (testing if two eigen-values are equal) and if true, to recover the elements of the collineation (see (Faugeras, 1993) for a review of the calibrated case).

B.4 Considering a zoom of the camera.

Another important specific displacement is the zoom already studied by some authors (Pollefeys et al., 95; Enciso et al., 1996b). Here we would like to provide a complete analysis of the form of the F -matrix and H_∞ -matrix and H -matrix in this case.

A real zoom does not correspond to a simple variation of the intrinsic parameters since the position of the optical center varies, and a translation occurs⁴. Let us now model this effect.

From equation (1.7) and (1.17) we have a F -matrix of the form :

$$\mathbf{F} = \begin{pmatrix} 0 & -\lambda t^2 & \lambda f t^1 - F^{01} v_0 \\ \lambda t^2 & 0 & -\lambda f t^0 + F^{01} u_0 \\ -\lambda f' t^1 + F^{01} v'_0 & \lambda f' t^0 - F^{01} u'_0 & F^{22} \end{pmatrix} \quad (\text{B.14})$$

where :

$$\begin{aligned} F^{22} &= (u_0 v'_0 - v_0 u'_0) F^{01} \\ &\quad - u_0 F^{20} - u'_0 F^{02} - v_0 F^{21} - v'_0 F^{12} \end{aligned}$$

which is thus constrained by:

$$F^{00} = 0, F^{11} = 0, F^{01} = -F^{10} \quad (\text{B.15})$$

Reciprocally, using equation (B.23) we can easily verify that this is a sufficient condition.

4. We consider that a zoom is the combination of a *variation of the intrinsic parameters, plus a general translation, but without any rotation.* We must assume that this translation is general, and in particular not related to the displacement of the principal point because, we are in the case of a thick lens and the focus mechanism induces variations of the latter without variations of the former

Here, we have 4 independent parameters (since a general F -matrix has 7 parameters and here 3 constraints occur). More precisely, knowing (u_0, v_0, f) we can recover, if $F^{01} = -\lambda t^2 \neq 0$, which is expected for a zoom:

$$\mathbf{t} \equiv \begin{pmatrix} F^{12} - F^{01} u_0 \\ -F^{02} - F^{01} v_0 \\ f F^{01} \end{pmatrix}$$

and

$$(B.16)$$

$$\begin{cases} u'_0 &= \frac{f'}{f} u_0 - \frac{\frac{f'}{f} F^{12} + F^{21}}{F^{01}} \\ v'_0 &= \frac{f'}{f} v_0 + \frac{\frac{f'}{f} F^{02} + F^{20}}{F^{01}} \end{cases}$$

i.e. \mathbf{t} up to a scale factor and the new location of the principal point (u'_0, v'_0) while if $t^2 = 0$ we are in the case of an retinal displacement, as detailed before.

Here, we cannot recover f'/f because this expansion factor is an indetermination of the motion parameterization as discussed in (Viéville et al., 1996b) while equation (B.14) with $\det(\mathbf{F}) = 0$ is equivalent to equation (B.16). This is somehow a very negative result, since the “zoom” effect is precisely a variation of f , non-measurable! This is due to the fact that this is an affine quantity, as discussed in (Viéville et al., 1996b).

If we have a “perfect” zoom, i.e. $t^0 = t^1 = 0$, so that there is a pure translation in Z we obtain very simply: $u_0 = F^{12}/F^{01}, v_0 = F^{02}/F^{01}, u'_0 = F^{21}/F^{01}, v'_0 = F^{20}/F^{01}$ but no information on f and f' . The F -matrix has the same form as for a general zoom, unless $u_0 = u'_0$ and $v_0 = v'_0$ and in such a case the form is similar to the case of a pure translation.

If now, we consider that the affine calibration is given, we have:

$$\mathbf{H}_\infty \equiv \begin{pmatrix} \frac{f'}{f} & 0 & u'_0 - u_0 \frac{f'}{f} \\ 0 & \frac{f'}{f} & v'_0 - v_0 \frac{f'}{f} \\ 0 & 0 & 1 \end{pmatrix} \quad (B.17)$$

We are here in a situation where we recover, the new intrinsic calibration parameters from the old one, and the “zoom” effect since we have a knowledge of the affine calibration.

For a general planar structure, we have a H -matrix of the form:

$$\mathbf{H} \equiv \begin{pmatrix} \frac{f'}{f} + s^0 n^0 & s^0 n^1 & u'_0 - u_0 \frac{f'}{f} + s^0 n^2 \\ s^1 n^0 & \frac{f'}{f} + s^1 n^1 & v'_0 - v_0 \frac{f'}{f} + s^1 n^2 \\ s^2 n^0 & s^2 n^y & 1 + s^2 n^2 \end{pmatrix} \quad (B.18)$$

subject to the following constraint:

$$\begin{aligned} & H^{01} H^{20} H^{20} - H^{00} H^{21} H^{20} \\ & - H^{10} H^{21} H^{21} + H^{11} H^{21} H^{20} = 0 \end{aligned}$$

It thus yields 7 independent equations which allow to compute:

$$\mathbf{e} \equiv (H^{01} H^{20}, H^{10} H^{21}, H^{20} H^{21})$$

with a constraint on the plane parameter: $n^0 H^{21} = n^1 H^{20}$ and extract two equations for the intrinsic parameters:

$$\begin{cases} u'_0 &= \frac{f'}{f} u_0 + \frac{H^{01}}{H^{21}} + \frac{f'}{f} \frac{H^{01} H^{22} - H^{02} H^{21}}{H^{01} H^{20} - H^{00} H^{21}} \\ v'_0 &= \frac{f'}{f} v_0 + \frac{H^{10}}{H^{20}} + \frac{f'}{f} \frac{H^{10} H^{22} - H^{12} H^{20}}{H^{10} H^{21} - H^{11} H^{20}} \end{cases}$$

as for a general rigid object, but again no information on f'/f .

B.5 Considering retinal displacements.

Let us consider the particular type of displacement for which the retinal plane is invariant. Clearly this corresponds to rigid motion with: $t^2 = u^0 = u^1 = 0$, i.e. we have a rotation around the z axis only and of angle θ and a translation with $t^2 = 0$, while the calibration parameters might be constant or not.

In that case, we have a F -matrix of the form:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & \lambda t^1 \\ 0 & 0 & -\lambda t^0 \\ F^{20} & F^{21} & F^{22} \end{pmatrix} \quad (\text{B.19})$$

with :

$$\begin{aligned} F^{20} &= \frac{f'}{f} (-F^{02} \cos(\theta) - F^{12} \sin(\theta)) \\ F^{21} &= \frac{f'}{f} (F^{02} \sin(\theta) - F^{12} \cos(\theta)) \\ F^{22} &= -(F^{02} u'_0 + F^{20} u_0 + F^{12} v'_0 + F^{21} v_0) \end{aligned}$$

This F -matrix is constrained by:

$$F^{00} = F^{01} = F^{10} = F^{11} = 0 \quad (\text{B.20})$$

and we observed, from equation (B.19), that:

$$\frac{f'}{f} = \sqrt{\frac{(F^{20})^2 + (F^{21})^2}{(F^{02})^2 + (F^{12})^2}} \quad (\text{B.21})$$

so that we can estimate the variation of the focal length, while we recover, from equation (B.19), the angle of the rotation as:

$$\theta = \arctan\left(\frac{F^{21}}{F^{20}}\right) - \arctan\left(\frac{F^{12}}{F^{02}}\right) \quad (\text{B.22})$$

We also recover the translation up to a scale factor $\mathbf{t} \equiv \mathbf{e} \equiv (F^{02}, F^{12}, 0)^T$, and obtain a linear equation for the location of the principal point, using F^{22} .

On the reverse, considering any general matrix F , while the calibration parameters are either constant or varying, the displacement is a displacement for which the retinal plane is invariant, if and only if $F^{00} = F^{01} = F^{10} = F^{11} = 0$ since if we define $\mathbf{v} = (v^0, v^1, v^2) = 2 \tan\left(\frac{\theta}{2}\right) \frac{\mathbf{u}}{\|\mathbf{u}\|}$ we have:

$$\begin{aligned}
F^{00} &= (2v^0v^2 - 4v^1)t^1 - (4v^2 + 2v^0v^1)t^2 \\
F^{01} &= (2v^1v^2 + 4v^0)t^1 \\
&\quad + (v^0v^0 + v^2v^2 - v^1v^1 - 4)t^2 \\
F^{10} &= (-2v^0v^2 + 4v^1)t^0 \\
&\quad + (v^0v^0 - v^2v^2 - v^1v^1 + 4)t^2 \\
F^{11} &= (-2v^1v^2 - 4v^0)t^0 + (-4v^2 + 2v^0v^1)t^2
\end{aligned} \tag{B.23}$$

which real solutions are only either $t^0 = t^1 = t^2 = 0$ or $v^0 = v^1 = v^2 = 0$, as easily verified with a symbolic calculator.

Therefore a necessary and sufficient condition to be in this situation is, for the F -matrix, to be of the form

$$F = \begin{pmatrix} 0 & 0 & F^{02} \\ 0 & 0 & F^{12} \\ F^{20} & F^{21} & F^{22} \end{pmatrix}$$

Furthermore, if in addition we either know that (1) calibration parameters are constant or that (2) the rotation angle, we obtain additional constraints (respectively quadratic and linear) on the F -matrix components easily made explicit with equations (B.21) and (B.22).

If now, we assume that we have a knowledge of the plane at infinity, the related matrix \mathbf{H}_∞ is given by:

$$\mathbf{H}_\infty \equiv \begin{pmatrix} \frac{f'}{f} \cos(\theta) & -\frac{f'}{f} \sin(\theta) & \mathbf{H}_{\infty 02} \\ \frac{f'}{f} \sin(\theta) & \frac{f'}{f} \cos(\theta) & \mathbf{H}_{\infty 12} \\ 0 & 0 & 1 \end{pmatrix} \tag{B.24}$$

where :

$$\begin{aligned}
\mathbf{H}_{\infty 02} &= \frac{f'}{f} (-\cos(\theta)u_0 + \sin(\theta)v_0) + u'_0 \\
\mathbf{H}_{\infty 12} &= \frac{f'}{f} (-\cos(\theta)v_0 - \sin(\theta)u_0) + v'_0
\end{aligned}$$

In this case, we also directly obtain the rotation angle θ , the variation of the focal length f'/f , and can estimate the new location of the optical center (u'_0, v'_0) in function of the previous value (u_0, v_0) , as visible in equation (B.24). If the calibration parameters are constant, we directly obtain the location of the principal point. In any case, no information on the absolute value of the focal

length is obtained. The particular H_∞ -matrix to be estimated is of the form: $H_\infty = \begin{pmatrix} a & b & c \\ -b & a & d \\ 0 & 0 & 1 \end{pmatrix}$.

Let us now consider the fact that we are observing a planar structure, under an retinal displacement. In this case we do not have to estimate the eight parameters of the collineation, but only six of them since the matrix \mathbf{H} is of the form:

$$\mathbf{H} = \begin{pmatrix} a & b & d_u \\ c & d & d_v \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{B.25})$$

i.e. we can assume that $\mathbf{h}^2 \equiv (0,0,1)^T$ which leads to a retinal displacement.

This fact can be easily verified with the model of equation (1.7) since we have:

$$\begin{aligned} H^{20} &\equiv u^0 u^2 (1 - \cos(\theta)) - u^1 \sin(\theta) + t^2 n^0 \\ H^{21} &\equiv u^1 u^2 (1 - \cos(\theta)) + u^0 \sin(\theta) + t^2 n^1 \end{aligned} \quad (\text{B.26})$$

whatever the calibration parameters are. The unary vector aligned with the rotation axis is $\mathbf{u} = (u^0, u^1, u^2)$ and the normal of the plane $\mathbf{n} = (n^0, n^1, n^2)$.

Solving these equations we easily demonstrate that these two values vanish if and only if *there is no rotation except around the optical axis (i.e. $\mathbf{u} \equiv (0,0,1)^T$) and either the translation is parallel to the retinal plane (i.e. $s^2 \equiv t^2 = 0$) or the plane is a fronto-parallel plane (i.e. the normal of the plane $\mathbf{n} \equiv (0,0,1)^T$).*

In the general case of variable calibration parameters, it is not possible to recover the Euclidean parameters, since we still have 11 parameters to estimate. However, if the calibration parameters are constant, as given in (Viéville, 1994), we can recover the direction of the translation, the angle of the rotation and the plane normal up to an indetermination. In (Viéville, 1994) the equations have been given for small displacements and they can be easily generalized as given now:

$$\begin{cases} \theta &= 2 \arctan(Z) \\ t^0 &= \lambda \cos(\alpha) \\ t^1 &= \lambda \sin(\alpha) \\ n^0 &= \frac{t^0(H^{00} - H^{11}) + t^1(H^{01} - H^{10})}{(t^0)^2 + (t^1)^2} \\ n^1 &= \frac{t^1(H^{11} - H^{00}) + t^0(H^{01} - H^{10})}{(t^0)^2 + (t^1)^2} \end{cases} \quad (\text{B.27})$$

with :

$$\begin{aligned} (1 + H^{00} + H^{11} + H^{11}H^{00} - H^{01}H^{10})Z^2 \\ + 2(H^{01} - H^{10})Z \\ + (1 - H^{00} - H^{11} + H^{11}H^{00} - H^{01}H^{10}) &= 0 \end{aligned}$$

and $\alpha = \frac{1}{2} \arctan(2 \frac{\alpha_1}{\alpha_2})$ with :

$$\begin{aligned} \alpha_1 &= (H^{11}H^{01} + H^{00}H^{10}) \\ &\quad - \cos(\theta)(H^{01} + H^{10}) + \sin(\theta)(H^{11} - H^{00}) \\ \alpha_2 &= ((H^{00})^2 + (H^{01})^2 - (H^{10})^2 - (H^{11})^2) \\ &\quad - 2(\cos(\theta)(H^{11} - H^{00}) + \sin(\theta)(H^{10} + H^{01})) \end{aligned}$$

We have two solutions for the displacement, as in the continuous case. The interpretation is discussed in (Viéville, 1994). Furthermore, the translation is also given up to an indeterminate λ , as expected.

Furthermore, we have one additional linear equation for the intrinsic parameter (u_0, v_0) :

$$\begin{aligned} & \sin(\alpha) (H^{02} - u_0 (1 - H^{00}) + v_0 H^{01}) \\ & = \cos(\alpha) (H^{12} + u_0 H^{10} - v_0 (1 - H^{11})) \end{aligned} \tag{B.28}$$

as in the continuous case again. We do not have any constraint on the focal length f , since this quantity always appears in the equation as a factor of the indeterminate λ .

An even more degenerated form of the retinal planar displacement is to consider a *constant retinal motion*, i.e. $a = d = 1$ and $c = b = 0$ in equation (B.25). This extreme situation is sometimes used for panoramic displacements. It can be easily demonstrated that this situation corresponds to a *pure retinal translation of a fronto-parallel plane, with constant calibration parameters*, with $\mathbf{e} \equiv (d_u, d_v, 0)$.

Annexe C

Ebauche de reflexion à propos de la préhension du mouvement chez l'être humain.

Nous présentons dans cette annexe une suite des travaux présentés dans cette thèse pour l'étude des mécanismes de vision effectués par le cerveau humain. Ceci est bien-sûr un domaine très vaste et

déjà bien étudié mais ce pourrait être une nouvelle piste.

À suivre.

C.1 Introduction

Et si l'étude des singularités pouvait nous mettre sur la voie de la compréhension de la perception du mouvement ? Et si le fonctionnement du cerveau consistait (ou une partie de son fonctionnement) à examiner les cas de situations les plus simples pour déterminer le mouvement avant de considérer le mouvement comme général ? Et si notre perception travaillait de façon incrémentale en déterminant le mouvement de façon grossière puis affinait cette perception en utilisant des modèles plus complets, et ceci, uniquement dans les cas où une perception fine est nécessaire (reconstruction 3D de l'environnement précise, déplacement dans un univers étroit où la manoeuvre est délicate, ...

C.2 Présentation du sujet

La question que l'on peut se poser est de savoir si la perception du mouvement n'est pas incrémentale, partant d'une évaluation grossière obtenue par un modèle très simple puis affinée par des modèles de plus en plus précis, en fonction de la précision nécessaire. Pour cela, on s'intéresse aux mesures de l'activité cérébrale par l'intermédiaire, d'une part, de l'imagerie par résonance magnétique fonctionnelle (IRMf), d'autre part, par électro-encéphalographie (EEG) et magnéto-encéphalographie (MEG).

D'une part, l'Imagerie par Résonance Magnétique fonctionnelle (IRMf) rend compte de l'activité du cerveau de manière indirecte au travers de la mesure des débits sanguins. Sa résolution spatiale est satisfaisante, mais sa résolution temporelle, de l'ordre de quelques secondes, est insuffisante. La résolution nécessaire à une étude convenable des phénomènes cérébraux est, en effet, de l'ordre de la milliseconde.

D'autre part, si l'Electro-encéphalographie (EEG) et la Magnéto-encéphalographie (MEG) qui mesurent respectivement les champs électrique et magnétique en un certain nombre de points extérieurs au cerveau ont une résolution temporelle satisfaisante, leur résolution spatiale est, en revanche, inadaptée. De plus, la reconstruction tridimensionnelle du phénomène à partir des sources de signal observées est un problème inverse difficile qui semble pouvoir être posé de manière efficace en terme d'optimisation multi-modèle et robuste, comme développé dans nos travaux antérieurs.

Il s'agit donc, à ce niveau, de rechercher une modélisation géométrique et physique plus fine des phénomènes liés à l'activité cérébrale en vue d'une augmentation de la résolution spatiale en reconstruction EEG et MEG, basée sur des données anatomiques IRM servant d'information à priori pour le problème étudié.

A partir de ces travaux, on pourra alors étudier les structures cérébrales liées à la perception du mouvement, dans le cadre de l'hypothèse actuellement en vigueur de stratification dans la perception du mouvement.

Ce travail est donc une suite logique des travaux présentés dans cette thèse de manière :

- applicative, par l'usage des méthodes développées à un nouveau problème d'estimation
- réflexive, en ce qu'elle vise à vérifier si les modèles développés en vision artificielle peuvent corespondre à des phénomènes perceptifs observés en vision biologique

Ce travail s'inscrit dans l'action coopérative 3D-MEG d'Olivier Faugeras et dans la collaboration avec le Massachusetts General Hospital au niveau du développement d'activités concernant les signaux MEG, EEG et fMRI, avec notamment l'arrivée d'une nouvelle tête MEG, donc de moyens plus performants mis à disposition de cette équipe de chercheurs de renommée internationale.

Annexe D

Le pot de thèse

Cette annexe présente les recettes du pot de thèse. Nous espérons que même si cette annexe

risque d'être bien utile, elle ne sera cependant la partie la plus lue de la thèse.



Ces recettes sont, pour certaines, très largement inspirées de deux ouvrages:

- LES FRÈRES ROUX *Pâtisserie* Editions Macdonald
- LES SOEURS SCOTTO *Desserts de rêve* Editions Olivier Orban

D.1 Le cake aux olives.

Ingrédients : 150 g d'olives vertes dénoyautées
1/2 sachet de levure
15 cl de lait
250 g de jambon blanc en dés
4 oeufs
150 g de comté en dés ou rapé
sel, poivre
4 c. à soupe d'huile d'olive
150 g de farine

Préchauffer le four à th. 6 (180°C).

Tamiser ensemble la farine et la levure. Ajouter le sel et le poivre et faire un puits au centre. Ajouter les oeufs un à un en remuant puis le lait. Lorsque le mélange est homogène, ajouter l'huile, les olives préalablement ébouillantées, le jambon et le fromage.

Laisser cuire 1h15. Vérifier la cuisson.

D.2 La tarte au fromage de Michoko.

Alors, La quiche de Michoko, n'en est plus tout à fait une ... pas plus que ça n'est une tarte au fromage... Disons juste que c'est un audacieux mélange des deux!

Ingrédients : 4 oeufs
petits fromages de chèvre frais
1 rouleau de pâte brisée toute prête
ciboulette finement ciselée
1 pot 30 cl crème fraîche (+un peu de lait)
sel, poivre, noix de muscade râpée
1 talon de jambon blanc ou fumé découpé en dés
gruyère râpé

Préchauffer le four à th. 6 (180°C).

Dans un saladier, bien mélanger les oeufs, la crème fraîche, le sel, le poivre, la noix de muscade, la ciboulette.

Etaler la pâte dans la tourtière, répartir dessus les dés de jambon, le gruyère râpé, des morceaux de fromage de chèvre (coupés en deux, c'est bien).

Recouvrir du mélange oeufs+crème.

Enfourner pour 1/2 heure.

C'est fini!

Céline.

D.3 Le cake au citron.

Ingrédients : 1 beau citron non traité	2-3 oeufs
75 ml de crème fraîche épaisse	gelée de fruits (abricot, pomme, ...)
50 g de beurre très très mou	sucre glace
175 g de sucre	1 bouchon de rhum
140 g de farine	sel
2,5 g de levure	

Préchauffer le four à th. 7 (200°C).

Raper le zeste du citron dans un bol. Y ajouter le sucre, le sel et les oeufs. Battre au fouet sans trop faire mousser. Ajouter la crème et mélanger à nouveau. Tamiser la farine et la levure au dessus du mélange puis ajouter le beurre fondu et le rhum. Verser la pâte dans un moule à cake beurré et faire cuire 7 minutes à 200°C puis 33 minutes à 180°C.

A la fin de la cuisson, démouler et peindre aussitôt au rhum. Faire fondre la gelée de fruits à feu très doux dans une casserole. En peindre le cake refroidi. Mélanger le jus du citron et du sucre glace. Peindre à nouveau le cake avec ce mélange puis passer moins d'une minute au four à température maximale pour faire tenir le glaçage.

D.4 Le nougat glacé.

Ingrédients : 3 oeufs	2 c. à soupe d'eau de fleur d'oranger
90 g de sucre	150 g d'amandes mondées
100 g de miel liquide	fruits confits ou deshydrates
4 dl crème liquide	

Faire griller les amandes à la poêle puis ajouter 2 grosses cuillères à soupe de sucre. Dès que celui-ci est fondu, retirer du feu et faire refroidir. Piller grossièrement.

Faire chauffer le miel tout doucement. Fouetter les blancs d'oeufs en neige ferme puis, tout en continuant de fouetter, ajouter le miel bouillant et l'eau de fleur d'oranger.

Fouetter les jaunes d'oeufs avec le restant de sucre jusqu'à ce que le mélange blanchisse. Ajouter les blancs en continuant de fouetter.

Pendant ce temps, battre la crème liquide en chantilly et l'incorporer au mélange précédent en soulevant la masse avec une spatule.

Incorporer alors au mélange les amandes et fruits coupés en petits dés. Mettre dans des moules et laisser glacer pendant au moins 12 heures.

D.5 Le tiramisu.

Ingrédients : 6 jaunes d'oeufs	24 biscuits
140 g de sucre	1 l de café bien fort
500 ml de mascarpone	cacao non sucré en poudre
rhum, amareto et/ou marsala	

Battre les jaunes d'oeufs avec le sucre. Lorsque le mélange est bien blanchi, ajouter la mascarpone et continuer de battre. Ajouter un peu d'amareto ou de marsala.

Sucrer et aromatiser le café avec le rhum. Imbiber un premières couches de biscuits avec le café et disposez une couche de crème. Renouveler l'opération puis finir en tamisant du cacao sur le dessus. Déguster ou mettre au frais.

D.6 Gateau au chocolat, amandes et gingembre.

Ingrédients : 200 g de chocolat noir	30 g de gingembre confit
75 g d'amandes émondées	3 oeufs
125 g de beurre très mou	1 c. à café de Ricoré
80 g de sucre	rhum, sucre glace
75 g de farine	

Couper le gingembre en très petits morceaux et mettre à macérer dans le rhum pendant au moins 2 heures.

Préchauffer le four à th. 5 (175°C).

Colorer légèrement les amandes à la poêle et les écraser grossièrement.

Faire fondre le chocolat au bain marie.

Mettre le beurre dans le bol d'un mixer. Battre pendant 1 minute, puis ajouter le Ricoré et le sucre. Casser un oeuf puis ajouter 1/3 de la farine. Recommencer l'opération 2 fois. Incorporer le chocolat. Arrêter de battre. Ajouter alors le gingembre et le rhum.

Faire cuire 20 minutes. Au bout de ce temps, laisser reposer le gâteau 10 minutes avant de le démouler (il est moelleux au centre). Laisser refroidir 4 heures. Poudrer de sucre glace juste avant de servir.

Bibliographie

- AKAIKE, H. (1972). « Use of an information theoretic quantity for statistical model identification ». Dans *5th Hawaiï Int. Conf. System Sciences*, pages 249–250.
- AKAIKE, H. (1977). « On entropy maximisation principle. ». *Applications of Statistics*, pages 27–41.
- ALOIMONOS, J. (1990). « Perspective Approximations ». *Image and Vision Computing*, 8(3):179–192.
- ALTMANN, S. A. (1989). « Hamilton, Rodrigues, and the Quaternion Scandal ». *Mathematics Magazine*, 62(3):291–308.
- ARMSTRONG, M., ZISSERMAN, A., et BEARDSLEY, P. (1994). « Euclidean structure from uncalibrated images ». Dans HANCOCK, E., éditeur, *Proceedings of the 5th British Machine Vision Conference*, pages 508–518, York, UK. BMVA Press.
- ARMSTRONG, M., ZISSERMAN, A., et HARTLEY, R. (1996). « Self-calibration from Image Triplets ». Dans *Fourth European Conference on Computer Vision*, pages 3–16.
- ÅSTRÖM, K. et HEYDEN, A. (1998). « Continuous Time Matching Constraints for Image Streams ». *I.J.C.V.*, 28(1):85–96.
- AYACHE, N. (1989). *Vision Stéréoscopique et Perception Multisensorielle: Applications à la Robotique Mobile*. InterEditions, Paris.
- AZARBAYEJANI, A. et P., P. A. (1995). « Recursive estimation of motion, structure, and focal length. ». Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17 (6), pages 562–575.
- BEARDSLEY, P. A., REID, I. D., ZISSERMAN, A., et MURRAY, D. W. (1995). « Active visual navigation using non-metric structure ». Dans *Proceedings of the 5th International Conference on Computer Vision*, pages 58–64, Boston, MA. IEEE Computer Society Press.
- BENOSMAN, R. (1997). « *Reconstruction d'environnements perçus par un capteur panoramique de stéréovision et détermination des modèles CAO.* ». Thèse de Doctorat, Université de Paris VI.
- BIERNACKI, C., CELEUX, G., et GOVAERT, G. (1998). « Assessing a Mixture Model for Clustering with the Integrated Classification Likelihood ». Rapport technique RR-3521, INRIA Rhône-Alpes.
- BOBET, P. (1994). « *Tête stéréoscopique. Réflexes oculaires et Vision.* ». Thèse de Doctorat, Institut National Polytechnique de Grenoble, France.

- BOLLE, R. et COOPER, D. (1985). « On parallel Bayesian Estimation and Recognition for Large Data Sets, with Application to Estimating 3D Complex Object Position from Range Data ». Dans *Proceedings SPIE Conference on Vision for Robots, Cannes, France*.
- BOLLE, R. M. et COOPER, D. B. (1984). « Bayesian Recognition of Local 3-D Shape by Approximating Image Intensity Functions With Quadric Polynomials ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):418–429.
- BOLLES, R. C. et FISCHLER, M. A. (1981). « A RANSAC-based approach to model fitting and its application to finding cylinders in range data ». Dans *International Joint Conference on Artificial Intelligence*, pages 637–643, Vancouver, Canada.
- BOUFAMA, B., WEINSHALL, D., et WERMAN, M. (1994). « Shape from motion algorithms: a comparative analysis of scaled orthography and perspective ». Dans EKLUNDH, J.-O., éditeur, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800 de *Lecture Notes in Computer Science*, pages 199–204, Stockholm, Sweden. Springer-Verlag.
- BRAND, P., MOHR, R., et BOBET, P. (1993). « Distorsions optiques : correction dans un modèle projectif ». Rapport technique 1933, LIFIA–INRIA Rhône-Alpes.
- BROOKS, M. J., de AGAPITO, L., HUYNH, D. Q., et BAUMELA, L. (1996). « Direct Methods for Self-Calibration of a Moving Stereo Head ». Dans *Fourth European Conference on Computer Vision*, volume II, pages 415–426.
- CAO, Z. L., OH, S. J., et HALL, E. L. (1986). « Dynamic omnidirectional vision for mobile robots ». *Journal of Robotics systems*, 3(1):5–17.
- CELEUX, G. et NAKACHE, J.-P. (1994). *Analyse discriminante sur variables qualitatives*. Editions Polytechnica.
- CHANG, C. et CHATTERJEE, S. (1990). « Multiresolution Stereo – A Bayesian Approach ». Dans *Proceedings of the 10th International Conference on Pattern Recognition*, pages 908–912, Atlantic City, N.J. Computer Society Press.
- CHAVANCE, M. (1992). « Le canif et les bretelles ». *Rev. Epidem. et Santé Publique*, 40:209–218.
- CHRISTY, S. et HORAUD, R. (1994). « Euclidian Shape and Motion from Multiple Perspective Views by Affine Iterations ». Rapport technique 2421, INRIA Rhones-Alpes.
- CLARKE, J. C. (1997). « *Applications of Sequence Geometry to Visual Motion* ». Thèse de Doctorat, University of Oxford.
- CUI, N., WENG, J., et COHEN, P. (1990). « Extended structure and motion analysis from monocular image sequences ». Dans *Proceedings of the 3rd International Conference on Computer Vision*, pages 222–229, Osaka, Japan. IEEE Computer Society Press.
- de AGAPITO, L., HAYMAN, E., et REID, I. (1998). « Self-calibration of a rotating camera with varying intrinsic parameters ». Dans *British Machine Vision Conference*, Southampton, UK. BMVA Press.
- DEMENTHON, D. et DAVIS, L. S. (1989). « Exact and approximate solutions to the three-point pers-

- pective problem ». Rapport technique CAR-TR-471, Computer Vision Laboratory, University of Maryland.
- DEVERNAY, F. (97). « *Vision stéréoscopique et propriétés différentielles des surfaces* ». Thèse de Doctorat, École Polytechnique.
- DEVILLERS, O., MEISER, S., et TEILLAUD, M. (1992). « Fully dynamic Delaunay triangulation in logarithmic expected time per operation ». *Comput. Geom. Theory Appl.*, 2(2):55–80.
- EFRON, B. (1982). *The jackknife, the bootstrap and other resampling plans*. Philadelphia.
- ENCISO, R. (1995). « *Auto-Calibration des Capteurs Visuels Actifs. Reconstruction 3D Active.* ». Thèse de Doctorat, Université Paris XI Orsay.
- ENCISO, R. et VIÉVILLE, T. (1995). « Comment simplifier le processus de calibration? ». Dans *4emes Journées Internationales d'Informatique, "L'Interface des Mondes Réels et Virtuels"*, Montpellier.
- ENCISO, R., ZHANG, Z., et VIÉVILLE, T. (1996a). « Dense Reconstruction using Fixation and Stereo Cues ». Dans *World Automation Congress, ISIA C Symposia*, Montpellier.
- ENCISO, R., ZISSERMAN, A., et VIÉVILLE, T. (1996b). « An affine solution to the Euclidean calibration while using a zoom lens ». Dans *Workshop ALCATECH, 21-27, Danemark*.
- FAUGERAS, O. (1993). *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press.
- FAUGERAS, O. et MOURRAIN, B. (1995a). « About the correspondences of points between n images ». Dans *Proceedings of the Workshop on the representation of visual scenes*, Cambridge, Massachusetts, USA.
- FAUGERAS, O. et MOURRAIN, B. (1995b). « On the geometry and algebra of the point and line correspondences between n images ». Dans *Proceedings of the 5th International Conference on Computer Vision*, pages 951–956, Boston, MA. IEEE Computer Society Press.
- FAUGERAS, O., QUAN, L., et STURM, P. (98). « Self-Calibration of a 1D Projective Camera and Its Application to the Self-Calibration of a 2D Projective Camera ». Dans *Proceedings of the 5th European Conference on Computer Vision*, pages 36–52, Freiburg, Germany.
- FAUGERAS, O. et ROBERT, L. (1993). « What can two images tell us about a third one? ». Rapport technique 2018, INRIA.
- GASPARD, F. et VIÉVILLE, T. (1996). « Hierarchical Visual Perception without Calibration ». RR 3002, INRIA Sophia-Antipolis.
- GASPARD, F., ZISSERMAN, A., et VIÉVILLE, T. (1997). « Le zoom comme outil de calibration affine d'une caméra ». Dans *Journées ORASIS'97*, pages 27–38.
- GOLDBERG, D. E. (1991). *Genetic algorithms*. Addison-Wesley, Reading, MA.
- HAMPEL, J. P., RONCHETTI, E. M., et ROUSSEEUW, P. J. (1986). *Robust statistics: an approach based on influence fonctions*. Wiley, New-York.
- HANNAN, E. J. (1980). « The estimation of the order of an arma process ». *The Annals of Statistics*, 8(5):1071–1081.

- HANNAN, E. J. et QUINN, B. G. (1979). « The determination of the order of an autoregression ». *J. Roy. Statist. Soc.*, B(41):190–195.
- HARRIS, C. et STEPHENS, M. (1988). « A combined Corner and Edge Detector ». Dans *Proc. 4th Alvey Vision Conf.*, pages 189–192.
- HARTLEY, R. (1994a). « Lines and points in three views-an integrated approach ». Dans *Proceedings of the ARPA Image Understanding Workshop*. Defense Advanced Research Projects Agency, Morgan Kaufmann Publishers, Inc.
- HARTLEY, R. (1994b). « Self-Calibration from Multiple Views with a Rotating Camera ». Dans EKLUNDH, J.-O., éditeur, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800-801 de *Lecture Notes in Computer Science*, pages 471–478, Stockholm, Sweden. Springer-Verlag.
- HARTLEY, R. I. (1998). « Minimizing Algebraic Error in Geometric Estimation Problems ». Dans *Proceedings of the 6th International Conference on Computer Vision*, Bombay, India. IEEE Computer Society, IEEE Computer Society Press.
- HEYDEN, A. (1995). « *Geometry and Algebra of Multiple Projective Transformations* ». Thèse de Doctorat, Lund University.
- HEYDEN, A. et ASTRÖM, K. (1996). « Algebraic varieties in multiple view geometry ». Dans *Fourth European Conference on Computer Vision*, volume II, pages 671–682.
- HEYDEN, A. et ÅSTRÖM, K. (1997). « Euclidean reconstruction from image sequences with varying and unknown focal length and principal point ». Dans *Comp. Vision and Pattern Rec.*, pages 438–443. IEEE Computer Society Press.
- HOCHSCHILD, G. (1968). *La Structure des Groupes de Lie*. Dunod, Paris.
- HOLLAND, J. H. (1975). « *Adaptation in Naturel and Artificial Systems* ». Thèse de Doctorat, university of Michigan.
- HORAUD, R., CHRISTY, S., et DORNAIKA, F. (1994). « Object Pose: The Link between Weak Perspective, Para Perspective, and Full Perspective ». Rapport technique 2356, INRIA.
- HORAUD, R., CHRISTY, S., et MOHR, R. (1997). « Euclidean Reconstruction and Affine Camera Calibration Using Controlled Robot Motions ». Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, France.
- HORAUD, R., DORNAIKA, F., LAMIROY, B., et CHRISTY, S. (97). « Object Pose: The Link between Weak Perspective, Paraperspective, and Full Perspective ». *IJCV*, 22(2).
- HUBER, P. (1981). *Robust Statistics*. John Wiley & Sons, New York.
- JARVIS, R. A. et BYRNE, J. C. (1988). « An automated guided vehicle with map building and path finding capabilities ». Dans *4th ISPP*, pages 497–504.
- JENKINS, F. A. et WHITE, H. E. (1976). *Fundamentals of Optics*. McGraW-Hill Book Company, fourth edition.
- KANADE, T., SAITO, H., et VEDULA, S. (1998). « The 3D Room: Digitizing Time-Varying 3D

- Events by Synchronized Multiple Video Streams . ». Rapport technique CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA.
- KANATANI, K. (1996). *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science.
- KANATANI, K. (1998). « Statistical optimization and geometric inference in computer vision ». *Phil. Trans. R. Soc. Lond.*, A(356):1303–1320.
- KOENDERINCK (84). « The structure of images ». *biol. Cybern.*, 50:363–370.
- KOENDERINK, J. J. et van DOORN, A. J. (1991). « Affine Structure from Motion ». *Journal of the Optical Society of America*, A8:377–385.
- KULLBACK, S. (1959). *Information Theory and Statistics*. Wiley, New-York.
- LAVEAU, S. (96). « *Géométrie d'un système de N caméras. Théorie, estimation et applications* ». Thèse de Doctorat, École Polytechnique.
- LAVEST, J. (1992). « *Stéréovision axiale par zoom pour la robotique* ». Thèse de Doctorat, Université Blaise Pascal de Clermont-Ferrand, France.
- LINGRAND, D. et VIÉVILLE, T. (1995). « Dynamic Foveal 3D Sensing Using Affine Models ». Rapport technique RR-2687, INRIA Sophia-Antipolis.
- LINGRAND, D. et VIÉVILLE, T. (1996). « Dynamic Foveal 3D Sensing Using Affine Models ». Dans *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 810–814, Vienna, Austria. Computer Society Press.
- LORUSSO, A., EGGERT, D., et FISCHER, R. (1995). « A comparison of Four Algorithmes for Estimation 3D-Rigid Transformations ». Dans *6th British Machine Vision Conference (BMVC'95)*, pages 237– 240.
- LUONG, Q. et VIÉVILLE, T. (1996). « Canonical representations for the geometries of multiple projective views ». *Computer Vision and Image Understanding*, 64(2):193–229.
- LUONG, Q.-T., DERICHE, R., FAUGERAS, O., et PAPADOPOULO, T. (1993). « On Determining the Fundamental Matrix: Analysis of Different Methods and Experimental Results ». Rapport technique 1894, INRIA.
- LUONG, Q. T. et FAUGERAS, O. (1993). « Self-calibration of a stereo rig from unknown camera motions and point correspondences ». Rapport technique RR-2014, INRIA Sophia Antipolis.
- MACLAUHLAN, P. F. et MURRAY, D. W. (1995). « A unifying framework for structure and motion recovery from image sequences ». Dans *Proceedings of the 5th International Conference on Computer Vision*, pages 314–320, Boston, MA. IEEE Computer Society Press.
- MANIÈRE, T. (1997). « *Etude et réalisation d'un système de prise de vues panoramiques binoculaire dédié 'a la reconstruction 3d de sc'enes* ». Thèse de Doctorat, Université Pierre et Marie Curie, Paris.
- MAYBANK, S. (1990). « Properties of Essential Matrices ». *International Journal of Imaging Systems and technology*, 2:380–384.

- MOHR, R. (1993). Projective Geometry and Computer Vision. Dans CHEN, C., PAU, L., et WANG, P., éditeurs, *Handbook of Pattern Recognition and Computer Vision*, pages 369–393. World Scientific Publishing Company.
- MORE, J. (1977). The Levenberg-Marquardt Algorithm, Implementation and Theory. Dans WATSON, G. A., éditeur, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag.
- MURCIA, C. D., NIEMAZ, M., et VIÉVILLE, T. (1997). « Détection et suivi de cibles sur une durée indéterminée ». Dans *Journées Orasis'97*.
- OHTA, Y.-i., MAENOBU, K., et SAKAI, T. (1981). « Obtaining Surface Orientation from Texels Under Perspective Projection ». Dans *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 746–751, Vancouver, Canada.
- OLAGUE, G. et MOHR, R. (1998). « Optimal Camera Placement for Accurate Reconstruction ». Rapport technique RR-3338, INRIA - Rhone-Alpes.
- PAHLAVAN, K., UHLIN, T., et EKHLUND, J.-O. (1993). « Dynamic Fixation ». Dans *Proceedings of the 4th International Conference on Computer Vision*, pages 412–419, Berlin, Germany. IEEE Computer Society Press.
- POELMAN, C. J. et KANADE, T. (1993). « A Paraperspective Factorization method for Shape and Motion Recovery ». Rapport technique CMU-CS-93-219, Carnegie Mellon University, School of Computer Science.
- POELMAN, C. J. et KANADE, T. (1994). « A Paraperspective Factorization for Shape and Motion Recovery ». Dans EKLUNDH, J.-O., éditeur, *Proceedings of the 3rd European Conference on Computer Vision*, volume B de *Lecture Notes in Computer Science*, pages 97–108, Stockholm, Sweden. Springer-Verlag.
- POLLEFEYS, M., KOCH, R., et VAN GOOL, L. (1997). « Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters ». Rapport technique KUL/ESAT/MI2/9707, Katholieke Universiteit Leuven.
- POLLEFEYS, M., VAN GOOL, L., et MOONS, T. (95). « Euclidean 3D reconstruction from stereo sequences with variable focal lengths ». Dans *Proceedings of the 2nd Asian Conference on Computer Vision*, volume II, page 6, Singapore.
- POLLEFEYS, M., VAN GOOL, L., et PROESMANS, M. (1996). « Euclidean 3D reconstruction from stereo sequences with variable focal lengths ». Dans BUXTON, B., éditeur, *Proceedings of the 4th European Conference on Computer Vision*, volume I, pages 31–42, Cambridge, UK.
- QUAN, L. (1996). « Self-calibration of an affine camera from multiple views ». *IJCV*, 19(1):93–105.
- QUENOUILLE, M. (1956). « Notes on bias in estimation ». *Biometrika*, 43:353–360.
- REY, W. J. (1983). *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer, Berlin, Heidelberg.
- RISSANEN, J. (1978). « Modeling by shortest data description ». *Automatica*, 14:465–471.

- ROBERT, L. (1993). « *Perception stéréoscopique de courbes et de surfaces tridimensionnelles. Applications à la robotique mobile* ». Thèse de Doctorat, École Polytechnique, Paris, France.
- RODRIGUES, O. (1840). « Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire ». *Journal de Mathématiques Pures et Appliquées*, 5. pp. 380–440.
- SCHWARZ, G. (1978). « Estimating the dimension of a model ». *Ann. Statist.*, 6:461–464.
- SHAPIRO, L. S., ZISSERMAN, A., et BRADY, M. (1995). « 3D Motion Recovery via Affine Epipolar Geometry ». *IJCV*, 16(2):147 – 182.
- SHASHUA, A. (1994). « Trilinearity in Visual Recognition by Alignment ». Dans EKLUNDH, J.-O., éditeur, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800-801 de *Lecture Notes in Computer Science*, pages 479–484, Stockholm, Sweden. Springer-Verlag.
- SHASHUA, A. et NAVAB, N. (1994). « Relative Affine Structure: Theory and Application to 3D Reconstruction from Perspective Views. ». Dans *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Seattle, WA. IEEE.
- SLAMA, C. C., éditeur (1980). *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition.
- SOATTO, S. et PERONA, P. (1995). « Dynamic Rigid Motion Estimation From Weak Perspective ». Dans *Proceedings of the 5th International Conference on Computer Vision*, pages 321–328, Boston, MA. IEEE Computer Society Press.
- SORENSEN, H. (1900). « *Comparison of Kalman, Bayesian and Maximum Likelihood Estimation Techniques* », Chapitre 6. Unknown.
- SOUTHWELL, C. A., BASU, A., FIALA, M., et REYDA, J. (1996). « Panoramic stereo ». Dans *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 378–382, Vienna, Austria. Computer Society Press.
- SPETSAKIS et ALOIMONOS (1991). « A multi-frame approach to visual motion perception ». *ijcv*, 6(3):245–255.
- STURM, P. (1997a). « Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction ». Dans *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 1100–1105.
- STURM, P. (1997b). « *Vision 3D non calibrée. Contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage* ». Thèse de Doctorat, INPG, Grenoble, France.
- TOMASI, C. et KANADE, T. (1991). « Factoring Image Sequences into Shape and Motion ». Dans *Proc. IEEE Workshop on Visual Motion*, pages 21–28, Princeton, NJ.
- TOMASI, C. et KANADE, T. (1992). « Shape and motion from image streams under orthography: a factorization method ». *The International Journal of Computer Vision*, 9(2):137–154.
- TORR, P. (1995). « *Motion Segmentation and Outlier Detection* ». Thèse de Doctorat, Department of Engineering Science, University of Oxford.

- TORR, P. (1997). « An assessment of information criteria for motion model selection ». Dans PRESS, I. C. S., éditeur, *Comp. Vision and Pattern Rec.*, pages 47–52.
- TORR, P., FITZGIBBON, A. W., et ZISSERMAN, A. (1998). « Maintaining Multiple Motion Model Hypotheses Over Many Views to Recover Matching and Structure ». Dans SOCIETY, I. C., éditeur, *ICCV'98*, pages 485–491, Bombay, India. IEEE Computer Society, Narosa Publishing House.
- TORR, P., ZISSERMAN, A., et MAYBANK, S. (1995). « Robust Detection of Degenerate Configurations for the Fundamental Matrix ». Dans *Proceedings of the 5th International Conference on Computer Vision*, pages 1037–1042, Boston, MA. IEEE Computer Society Press.
- TORR, P. H. S. (1998). « Geometric motion segmentation and model selection ». *Phil. Trans. R. Soc. Lond. A*, 356:1321–1340.
- TORR, P. H. S. (1999). « Model Selection for Structure and Motion Recovery from Multiple Images ». Rapport technique MSR-TR-99-16, Microsoft Research.
- TORR, P. H. S. et MURRAY, D. W. (1997). « The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix ». *IJCV*, 24(3):271–300.
- TRIGGS, B. (1995). « Matching Constraints and the Joint Image ». Dans *Proceedings of the 5th International Conference on Computer Vision*, pages 338–343, Boston, MA. IEEE Computer Society Press.
- TRIGGS, B. (1998a). « Autocalibration from Planar Scenes ». Dans SPRINGER, éditeur, *ECCV'98*, volume I, pages 89–105.
- TRIGGS, B. (1998b). « Optimal Estimation of Matching Constraints ». Dans KOCH, R. et GOOL, L. V., éditeurs, *Workshop on 3D Structure from Multiple Images of Large-scale Environments SMILE'98*, Lecture Notes in Computer Science.
- TUKEY, J. (1958). « Bias and confidence in not quite large samples ». *Ann. Math. Stat.*, 29:164.
- VAN GOOL, L., MOONS, T., PROESMANS, M., et VAN DIEST, M. (1994). « Affine reconstruction from perspective image pairs obtained by a translating camera ». Dans *Proceedings of the International Conference on Pattern Recognition*, Jerusalem, Israel. Computer Society Press.
- VIÉVILLE, T. (1994). « Autocalibration of Visual Sensor Parameters on a Robotic Head ». *Image and Vision Computing*, 12.
- VIÉVILLE, T. (1996). « Mascotte : A few Maple Routines for Real-Time Code Generation ». Rapport technique RR-2826, INRIA.
- VIÉVILLE, T., CLERGUE, E., ENCISO, R., et MATHIEU, H. (1994). « Experimenting 3-D Vision on a Robotic Head ». Dans *Proceedings of the International Conference on Pattern Recognition*, pages 739–743, Jerusalem, Israel. Computer Society Press.
- VIÉVILLE, T., CLERGUE, E., ENCISO, R., et MATHIEU, H. (1995a). « Experimentating with 3-D vision on a robotic head ». *Robotics and Autonomous Systems*. 14(1).
- VIÉVILLE, T., FACAO, P., et CLERGUE, E. (1995b). « Computation of ego-motion using the Vertical Cue ». *Machine Vision and Applications*, 8(1):41–52.

- VIÉVILLE, T. et FAUGERAS, O. (1995). « Motion analysis with a camera with unknown, and possibly varying intrinsic parameters ». Dans *The 5th Int. Conf. Comp. Vision*, pages 750–756, Cambridge, Massachusetts, USA.
- VIÉVILLE, T. et FAUGERAS, O. (1996). « The First Order Expansion of Motion Equations in the Uncalibrated Case ». *CVGIP: Image Understanding*, 64(1):128–146.
- VIÉVILLE, T., FAUGERAS, O. D., et LUONG, Q.-T. (1996a). « Motion of Points and Lines in the Uncalibrated Case ». *The International Journal of Computer Vision*, 17(1):7–42.
- VIÉVILLE, T. et LINGRAND, D. (1995). « Using Singular Displacements for Uncalibrated Monocular Visual Systems ». Rapport technique RR-2678, INRIA Sophia-Antipolis.
- VIÉVILLE, T. et LINGRAND, D. (1999). « Using Specific Displacements to analyze Motion without Calibration ». *IJCV*, 31(1):5–29.
- VIÉVILLE, T., ROMANN, F., HOTZ, B., MATHIEU, H., BUFFA, M., ROBERT, L., FACAO, P., FAUGERAS, O., et AUDREN, J. (1993). « Autonomous navigation of a mobile robot using inertial and visual cues ». Dans KIKODE, M., SATO, T., et TATSUNO, K., éditeurs, *Intelligent Robots and Systems*, Yokohama.
- VIÉVILLE, T., ZELLER, C., et ROBERT, L. (1996b). « Using Collineations to Compute Motion and Structure in an Uncalibrated Image Sequence ». *The International Journal of Computer Vision*, 20(3):213–242.
- WAXMAN, A. M. et ULLMAN, S. (1985). « Surface structure and three-dimensional motion from imageflow kinematics ». *Int. J. of Robot. Res.*, 4.
- WENG, J., HUANG, T., et AHUJA, N. (1989). « Motion and Structure from Two Perspective Views: Algorithms, Error Analysis and Error Estimation ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476.
- WILES, C. et BRADY, M. (1996). « Ground plane motion camera models. ». Dans BUXTON, B., éditeur, *Proceedings of the 4th European Conference on Computer Vision*, pages 234–247, Cambridge, UK.
- WILLSON, R. (1994). « *Modeling and Calibration of Automated Zoom Lenses* ». Thèse de Doctorat, Department of Electrical and Computer Engineering, Carnegie Mellon University.
- YAGI, Y., KAWATO, S., et TSUJI, S. (1994). « Real-time omnidirectional image sensor (COPIS) for vision-guided navigation ». *IEEE Trans. on Robotics and Automation*, 10(1).
- ZELLER, C. (1996). « *Calibration Projective Affine et Euclidienne en Vision par Ordinateur* ». Thèse de Doctorat, École Polytechnique.
- ZHANG, Z. (1997a). « On the optimization criteria used in two-view motion analysis. ». Rapport technique TR-H-228, ATR Human Information Processing Research Labs., Kyoto, Japan.
- ZHANG, Z. (1997b). « Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting ». *Image and Vision Computing Journal*, 15(1):59–76.
- ZHANG, Z. (1998a). « Determining the epipolar geometry and its uncertainty: a review ». *The International Journal of Computer Vision*, 27(2):161–195.

- ZHANG, Z. (1998b). « Understanding the Relationship Between the Optimization Criteria in Two-View Motion Analysis ». Dans The Institute of Electrical and ELECTRONICS ENGINEERS, I., éditeur, *International Conference on Computer Vision*, pages 772–777, Bombay, India. IEEE Computer Society, Narosa Publishing HOUSE.
- ZHANG, Z., DERICHE, R., FAUGERAS, O., et LUONG, Q.-T. (1995). « A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry ». *Artificial Intelligence Journal*, 78:87–119.
- ZISSERMAN, A., LIEBOWITZ, D., et ARMSTRONG, M. (1998). « Resolving Ambiguities in Auto-Calibration ». *Philosophical Transactions of the Royal Society of London, SERIES A*, 356(1740):1193–1211.

Index

– Symboles –

α_u	22, 62, 66, 79
α_v	22, 62, 66, 79
β_u	24, 66
β_v	24, 66
γ	22, 62, 66
u_0	22, 66
v_0	22, 66

– A –

aberration	14
AIC	<i>voir</i> Akaike
Akaike	118
algorithmes génétiques	122
Argès	131
auto-calibration	66

– B –

barillet	32
barrette CCD	9
biais	108
BIC	120

– C –

C	136
calibration	66
caméra couleur	9
camera obscura	11
caméra sténopée	10
cas particuliers	
géométriques (2 vues)	72
caméra (2 vues)	65
de mouvement (2 vues)	68
de mouvement (N vues)	79

gestion de l’inventaire	81
Case Deletion Diagnostic	111
collinéation	20
contrainte	
d’ordre	41
d’unicité	41
de continuité	41
contraintes épipolaires	41
coussinet	32

– D –

dioptrique	12
disparité vectorielle	42
distorsion	15, 32

– E –

épipolaire	
droite épipolaire	41
épipôle	41
espace projectif	20
estimation par	
échantillon test	117
bretelles	117
resubstitution	116
validation croisée	117
Euler (angles d’)	37

– G –

gamma-correction	9
Gauss	13
géométrie	
épipolaire	41
projective	20
GIC	121

– H –	
homographie	20, 50, 51, 55, 73
– J –	
Java	136
– K –	
Képler	13
– L –	
lentille	13
lentille épaisse	14
lentille mince	14
Levenberg-Marquardt	112
– M –	
MAICE	119
Maple	135
matrice fondamentale	46
miroir	15
modèle projectif linéaire	19
modèle des lentilles épaisses	31
moindre médiane des carrés	113
moindres carrés	111
– N –	
Nautilus	10
– O –	
optique	12
– P –	
paramètres	
extrinsèques	40
intrinsèques	30
modaux	30
pinhole	<i>voir</i> caméra sténopée
pixel	9
PLUNDER	64
point principal	66
principe de dualité	21
prisme	12, 15

problème linéaire	110
projection	9, 21
affine	<i>voir</i> orthographique
ortho-perspective	26
orthographique	27, 150
para-perspective	23
perspective	22
– Q –	
quaternion	39
– R –	
Rodrigues	38–40
– S –	
Snell-Descartes	12, 13