

وزارة التعليم العالي والبحث العلمي

Badji Mokhtar Annaba University  
Université Badji Mokhtar Annaba

جامعة باجي مختار - عنابة



Faculté des Sciences de l'Ingénieur

Département d'Informatique

**THÈSE**

Présentée en vue de l'obtention  
du diplôme de Doctorat 3<sup>ème</sup> Cycle

# Estimation de la Forme Volumique à Partir des Images

**Domaine:** Mathématiques et Informatique

**Filière:** Informatique

**Spécialité:** Reconnaissance des Formes et Intelligence Artificielle -RFIA-

Par: **Mr ZOUBIR TALAI**

**Devant le Jury**

**Président**

Mme Labiba SOUCI-MESLATI

Professeur University Badji Mokhtar Annaba

**Rapporteur**

Mme Yamina MOHAMED BEN ALI

Professeur University Badji Mokhtar Annaba

**Examineurs**

Mr Abdelouahab MOUSSAOUI

Professeur University Sétif

Mme Farida-Hayet MEROUANI

Professeur University Badji Mokhtar Annaba

**Année 2014-2015**

# Remerciements

**T**out d'abord je commence par remercier **Allah** pour ma santé, pour cette opportunité précieuse de pouvoir faire la recherche scientifique, ainsi pour tout les faveurs qui m'étaient attribuées "al hamdoulillah".

**J**E remercie ma directrice de thèse Pr. MOHAMED BEN ALI Yamina, pour son soutien durant cette longue période du travail, pour ces conseils précieux, pour sa présence, son guide, et surtout de nous avoir accordé la chance de travailler avec elle.

**J**E remercie également les membres de jury pour avoir accepté de donner leur précieuse évaluation et valider ce modeste travail.

**J**E n'oublie pas de remercier le "Laboratoire de Recherche en Informatique" pour nous avoir accepté comme des membres, ainsi que les moyens et les facilités qui nous ont offerts afin de faire nos expérimentations. Sans oublier le département d'informatique de l'Université Badji Mokhtar Annaba pour la formation précieuse durant tout mon cursus universitaire.

**J**E termine par remercier tout le monde qui a contribué de près ou de loin dans la réalisation de ce projet.

Merci tous le monde.



# Dédicaces

Je dédie ce modeste travail à :

**M**A chère mère, ainsi que mon très précieux père et tous les membres de ma famille : Had, Islam, Mouad, Haïtem, Mima. Merci pour le soutien moral, et d'avoir été là à mes cotés durant des périodes délicates.

**A** La personne la plus importante dans ma vie, ma femme qui ma soutenue et aider pour finir mon projet, qui a été présente à mes cotés pour le bien et le pire. JE TE REMERCIE INFINIMENT.

**A** Ma belle mère Pr. Houria et mon beau père Pr. Nacer qui je les remercie pour leurs conseils très précieux et leur soutien.

**M**ES précieux amis : Bilel, Isam, Farhou, Baydoo, Oualid, Hamza, Fayçal, Mounir, Mohamed, Walid, Wiwa, Youba, Alla, Badri, Zinou.

**S**éncialement Touma, Mika, Rabia, Farida et toutes mes tantes et mes oncles BOU-NAIDJA RACHEDI et TALAI.



# Résumé

La problématique de la reconstruction géométrique est un domaine de recherche très vaste. Depuis le début des années 70, il a suscité beaucoup d'intérêt aussi bien dans les domaines de la recherche que pour des applications industrielles. Le travail présenté dans cette thèse regroupe plusieurs étapes qui sont englobées dans le processus de la reconstruction tridimensionnelle.

D'abord, une méthode alternative basée sur l'optimisation par essaim de particules est étudiée pour résoudre le problème de la transformation projective qu'on appelle l'Homographie. Plusieurs résultats sont illustrés dans ce travail, notamment pour la géométrie épipolaire, ainsi que l'étalonnage des caméras pour finir par le modèle 3D reconstitué à partir d'une paire d'images d'une scène donnée. L'approche proposée pour le calcul de l'Homographie a bien montré son efficacité, en terme de précision et de robustesse face aux données de taille importante, comparant à l'approche mathématique analytique qui repose sur la décomposition en valeur singulière puis la minimisation de l'erreur. L'approche a été testée sur des données artificielles et réelles et a prouvé une amélioration importante. Finalement une reconstruction projective suivie d'une reconstruction affine est présentée comme une dernière partie du travail pour valider l'approche proposée.

**Mot clés :** Reconstruction 3D, Homographie, Optimisation par essaim de particule, Stéréovision, Rectification, Calibration.

# Abstract

The problem of geometrical reconstruction is a very large area of research. Since the early 70s, it had a lot of interest in both fields of research and applications in industry . The work presented in this thesis consists of several steps that are encompassed in the 3D reconstruction process.

In the first place, an alternative method based on particle swarm optimization is designed to solve the projective transformation problem that is called Homography. Several results are illustrated in this work, including the epipolar geometry and calibration of cameras to finally ending with a 3D model reconstituted from a pair of images of a given scene. The proposed approach for the calculation of the Homography has shown its effectiveness in terms of accuracy and robustness in dealing with large datasets comparing to the analytical mathematical approach based on singular value decomposition then minimizing the error. The approach was tested on artificial and real data showed significant improvement. Finally a projective reconstruction followed by an affine reconstruction is presented as a final part of the work to validate the proposed approach.

**Keywords** :3D Reconstruction, Homography, Particle swarm optimisation, Stereovision, Rectification, Calibration.

## الملخص

إن إشكالية إعادة بناء الأشكال ثلاثية الأبعاد الهندسية تعد من أكبر مجالات البحث حديثا. منذ وقت مبكر وفي السبعينيات، حظيت بالكثير من الإهتمام في كل من مجالات البحوث والتطبيقات الصناعية. العمل المقدم في هذه الأطروحة يتكون من الخطوات العدة التي تكون عملية إعادة البناء الثلاثي الأبعاد. اقتراح طريقة بديلة تعتمد على استمثال عناصر السرب لحل مشكلة الإسقاط التحويلي والتي تسمى بالهوموغرافيا ستدرس وتوضح. سيتم عرض عدة نتائج في هذا العمل، بما في ذلك الرؤية المزدوجة ومعايرة الكاميرات للإتهاء أخيرا مع نموذج ثلاثي الأبعاد مستنتج من زوج من الصور لمشهد معين. قد أظهرت الطريقة المقترحة لحساب الهوموغرافيا فعاليتها من حيث الدقة والفعالية في التعامل مع مجموعات البيانات الكبيرة بالمقارنة مع الطريقة التحليلية الرياضية الكلاسيكية المعتمدة على أساس تجزئة القيم الأولية التي يليها التقليل من الخطأ. تم اجراء عدة اختبارات لهذه الطريقة على بيانات اصطناعية وحقيقية وأظهرت تحسنا ملحوظا. في اخر هذه الأطروحة سيتم عرض بعض النتائج المحصل عليها من ناحية إعادة البناء ثلاثية الأبعاد.

**كلمات المفتاح** - اعادة البناء ثلاثية الأبعاد، الهوموغرافيا، استمثال عناصر السرب، الرؤية المزدوجة، تصحيح الصور، معايرة الكاميرات.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problématique . . . . .	2
1.2	Motivation . . . . .	2
<b>2</b>	<b>Heuristique et Métaheuristique</b>	<b>4</b>
2.1	Création et origine . . . . .	5
2.2	Classification des métaheuristiques . . . . .	6
2.2.1	Les approches à base de solution unique “trajectoire” . . . . .	6
2.2.2	Les approches à base de population de solutions “évolutionnaires” . . . . .	6
2.3	Approches trajectoires . . . . .	7
2.3.1	Le recuit simulé (simulated annealing) . . . . .	7
2.3.2	La recherche Tabou (Tabu Search) . . . . .	9
2.3.3	La recherche à voisinage variable . . . . .	11
2.4	Approches populations “évolutionnaires” . . . . .	12
2.4.1	Les algorithmes génétiques . . . . .	12
2.4.2	Les colonies de fourmis . . . . .	17
2.4.3	L’optimisation par colonies d’abeilles . . . . .	19
2.4.4	L’optimisation par essaim de particules . . . . .	23
<b>3</b>	<b>Stéréovision</b>	<b>30</b>
3.1	Géométrie Épipolaire . . . . .	32
3.2	La Matrice Fondamentale . . . . .	34
3.2.1	Dérivation Géométrique . . . . .	35
3.2.2	Dérivation Algébrique . . . . .	36
3.2.3	Représentation algébrique de la géométrie épipolaire – La matrice fondamentale . . . . .	37
3.2.4	Condition de Correspondance . . . . .	39
3.2.5	Propriétés de la Matrice Fondamentale . . . . .	40

3.2.6	Quelques détails sur la matrice fondamentale . . . . .	41
3.2.7	L’Homographie de la ligne épipolaire . . . . .	42
3.3	Géométrie épipolaire calibrée et matrice essentielle . . . . .	42
3.3.1	La matrice essentielle . . . . .	43
3.3.2	Les propriétés de la matrice essentielle . . . . .	44
3.4	Estimation de la géométrie épipolaire – Méthode de base . . . . .	45
3.4.1	Combien de correspondances faut-il avoir ? . . . . .	46
3.5	Rectification des images . . . . .	46
3.5.1	Préliminaires . . . . .	47
3.5.2	Projection de l’épipôle à l’infini . . . . .	49
3.5.3	La mise en correspondances des transformations . . . . .	50
3.5.4	Transformations Quasi affine . . . . .	53
3.5.5	Rééchantillonnage . . . . .	56
3.6	Expérimentations . . . . .	57
<b>4</b>	<b>Homography</b>	<b>62</b>
4.1	Définition Générale . . . . .	63
4.2	Description Mathématique . . . . .	63
4.3	L’estimation de l’Homography . . . . .	65
4.4	Résolution analytique de l’Homographie . . . . .	65
4.5	Résolution de l’Homographie basée sur les essais de particules . . . . .	66
4.5.1	Principe . . . . .	66
4.5.2	Première estimation de la matrice H . . . . .	66
4.6	Expérimentations . . . . .	68
4.6.1	Application sur des images synthèses . . . . .	68
4.6.2	Application sur l’échiquier . . . . .	75
4.6.3	Application sur la base de données INRIA . . . . .	78
4.6.4	Discussion . . . . .	83
4.7	Conclusion . . . . .	83
<b>5</b>	<b>Calibration de Caméra</b>	<b>84</b>
5.1	Définition . . . . .	85
5.1.1	Calibration Photogrammétrique . . . . .	85
5.1.2	Auto-Calibrage . . . . .	86
5.2	Modèle de caméra Sténopé . . . . .	87
5.3	Équations de base . . . . .	88
5.3.1	Notation . . . . .	88

5.3.2	Homographie entre le plan du modèle et son image . . . . .	89
5.3.3	Contraintes sur les paramètres intrinsèques . . . . .	89
5.4	Résolution du calibrage de caméra . . . . .	90
5.4.1	Solution de forme fermée . . . . .	90
5.4.2	Extraction des paramètres intrinsèques depuis la matrice B . . . . .	92
5.4.3	Estimation du maximum de vraisemblance . . . . .	92
5.4.4	Traitement de la distorsion radiale . . . . .	93
5.5	Expérimentations . . . . .	94
<b>6</b>	<b>Reconstruction 3D</b>	<b>104</b>
6.1	Méthodes actives . . . . .	105
6.1.1	Utilisation du Scanner Laser pour la reconstruction . . . . .	105
6.2	Méthodes passives . . . . .	107
6.2.1	Méthodes monoculaires . . . . .	107
6.2.2	Approches multi-vues . . . . .	111
6.2.3	Reconstruction à partir du mouvement . . . . .	121
6.3	Reconstruction 3D à partir des caméras non calibrées . . . . .	123
6.4	Reconstruction à partir des caméras calibrées . . . . .	125
6.4.1	Interprétation géométrique des quatre solutions . . . . .	126
6.5	Théorème de la reconstruction projective . . . . .	127
6.6	Reconstruction stratifié . . . . .	129
6.6.1	L'étape de reconstruction métrique . . . . .	133
6.6.2	Reconstruction métrique directe en utilisant $\omega$ . . . . .	137
6.6.3	La reconstruction directe - en utilisant la réalité de terrain . . . . .	138
6.6.4	Expérimentations . . . . .	139
	<b>Conclusion</b>	<b>145</b>
	Perspectives . . . . .	146

## Bibliographie

# Table des figures

2.1	Sélection par roulette . . . . .	14
2.2	Croisement uniforme . . . . .	16
2.3	Croisement 1-point . . . . .	16
2.4	Croisement 2-point . . . . .	16
2.5	Principe de mouvement des fourmis . . . . .	18
2.6	Comportement des abeilles butineuses pour le nectar . . . . .	21
2.7	Trois topologies différentes . . . . .	25
3.1	Principe de la stéréoscopie . . . . .	31
3.2	Correspondances des points . . . . .	32
3.3	Entités géométriques impliqués dans la géométrie épipolaire . . . . .	33
3.4	Exemple de la géométrie épipolaire . . . . .	33
3.5	Mouvement parallèle au plan d'image . . . . .	35
3.6	Transfert de point via un plan . . . . .	36
3.7	Homographie d'une ligne épipolaire . . . . .	43
3.8	Un rééchantillonnage non-quasi-affine d'un peigne présenté dans [54] . . . . .	54
3.9	Capture d'écran de l'interface d'accueil du logiciel . . . . .	58
3.10	L'interface dédiée à la StéreoVision . . . . .	58
3.11	Affichage des résultats du calcul de la matrice fondamentale et projection des lignes épipolaires (exemple 1) . . . . .	59
3.12	Affichage des résultat de la rectification et projection des lignes épipolaires (exemple 1) . . . . .	60
3.13	Affichage des résultats du calcul de la matrice fondamentale et projection des lignes épipolaires (exemple 2) . . . . .	60
3.14	Affichage des résultat de la rectification et projection des lignes épipolaires (exemple 2) . . . . .	61
4.1	Les différents cas possibles de l'Homographie . . . . .	64
4.2	Échiquier synthétisé avec 4 points . . . . .	69

4.3	Échiquier synthétisé avec 80 points . . . . .	69
4.4	L'évolution de l'essaim en utilisant 4 points . . . . .	71
4.5	L'évolution de l'essaim en utilisant 80 points . . . . .	74
4.6	Sélection des quatre coins qui correspondent aux axes et l'origine . . . .	76
4.7	L'utilisation d'Homographie pour trouver tous les corners de l'échiquier	76
4.8	Sélection des quatre coins qui correspondent aux axes et l'origine . . . .	77
4.9	L'utilisation d'Homographie pour trouver tous les corners de l'échiquier	78
4.10	Comparaison de la projection des erreurs dans l'image Lipton . . . . .	79
4.11	Comparaison de la projection des erreurs dans l'image INRIA . . . . .	80
4.12	Comparaison de la projection des erreurs dans l'image Rubik . . . . .	81
4.13	Comparaison de la projection des erreurs dans l'image Boat . . . . .	82
5.1	Modèle de caméra Sténopé . . . . .	88
5.2	Capture d'écran du logiciel élaboré pour la calibration des caméras . . .	96
5.3	Résultats des approches comparés : exemple numéro 1 . . . . .	98
5.4	Résultats des approches comparés : exemple numéro 2 . . . . .	100
5.5	Résultats des approches comparés : exemple numéro 3 . . . . .	102
5.6	Résultats . . . . .	103
6.1	Résultats du recalage . . . . .	106
6.2	Résultat de l'approche Shape From Shading présenté dans [86] . . . . .	108
6.3	Résultat de l'approche Shape From Texture présenté dans [84] . . . . .	109
6.4	Éstimation d'une carte de profondeur par stéréo-vision . . . . .	113
6.5	Reconstruction 3D à partir des silhouettes présenté dans [90] . . . . .	116
6.6	Un objet 2D représenté avec une vue, sa silhouette et le cône correspondant.	117
6.7	$v_\infty$ et $v_R$ pour un objet 2D : $v_\infty$ est l'enveloppe convexe de l'objet . . .	118
6.8	Silhouette d'un objet 3D : elle est incluse dans la projection de l'enveloppe convexe. . . . .	119
6.9	Principe du fonctionnement du Space carving . . . . .	120
6.10	L'ensemble des pixels photo-consistants . . . . .	121
6.11	Représentation du Photohull . . . . .	121
6.12	Résultat de la modélisation temporelle présenté dans [44] . . . . .	122
6.13	Reconstruction de la structure d'un objet à partir du champ vectoriel .	123
6.14	Les quatre solutions possibles pour une reconstruction calibrée depuis $E$	127
6.15	Reconstruction projective . . . . .	129
6.16	Reconstruction affine . . . . .	132
6.17	Reconstruction métrique . . . . .	137

6.18	Reconstruction directe . . . . .	139
6.19	Reconstruction 3D projective de la scène des livres en utilisant une caméra non-calibrée . . . . .	141
6.20	Reconstruction 3D Affine de la scène des livres en utilisant la contrainte des lignes parallèles . . . . .	142
6.21	Deuxième exemple d'une reconstruction projective 3D de la scène du médicament . . . . .	143
6.22	Reconstruction 3D Affine de la scène du médicament en utilisant la contrainte des lignes parallèles . . . . .	144

# Liste des tableaux

3.1	Résumé des propriétés de la matrice fondamentale . . . . .	41
4.1	Résultat d'un test avec 4 points (choisi parmi plusieurs tests) . . . . .	70
4.2	Résultat d'un autre test avec 4 points (choisi parmi plusieurs tests) . . .	70
4.3	Résultats d'un test avec 80 points (choisi parmi plusieurs tests) . . . . .	72
4.4	Résultats d'un autre test avec 80 points (choisi parmi plusieurs tests) . .	73
4.5	L'évolution de l'essaim en utilisant 80 points et la Normalisation . . . . .	75
4.6	Résultats de l'expérimentation 1 (Temps en millisecondes) . . . . .	76
4.7	Résultats de l'expérimentation 2 (Temps en millisecondes) . . . . .	77
6.1	Résumé des étapes pour avoir une reconstruction métrique à partir d'une reconstruction projective . . . . .	140

# Chapitre 1

## Introduction

Depuis l'invention de la caméra, les chercheurs se sont intéressés à l'image numérique. Le traitement d'image devient alors l'un des plus grands domaines de recherche en informatique. L'axe de recherche le plus important dans ce domaine est l'interprétation de l'image. Plusieurs techniques d'intelligence artificielle sont placées pour achever un but supérieur qui est la construction d'un système aussi complexe et performant que l'œil humain. Pour cela, ce dernier doit se servir de plusieurs informations acquises depuis des différentes sources pour pouvoir analyser la scène perçue et interagir d'une façon optimale selon les besoins.

La vision humaine est obtenue selon un apprentissage que l'homme acquiert depuis sa naissance. L'information visuelle est fusionnée avec les connaissances stockées dans son cerveau. Ces dernières sont issues de son apprentissage durant des expériences antérieures. Cela lui permet de bien interagir avec son environnement.

La vision par ordinateur (la vision artificielle) est la discipline qui essaye de reproduire le système visuel humain. En utilisant de l'intelligence artificielle, le principe est de permettre à une machine de voir. On dit qu'une machine voit si elle est dotée des capacités suivantes :

- L'Analyse d'une scène représentée par une ou plusieurs images.
- Le Traitement de données acquises d'une scène.
- La compréhension du contenu de cette scène.

## 1.1 Problématique

Les formes volumiques perçues par l'être humain se constituent dans le cerveau par une reconstitution d'une image à partir de la perception des deux images provenant de chaque œil. Pour la vision artificielle, cette information ne provient pas d'une seule image de la scène prise par la caméra. Malgré la richesse d'informations contenues dans une image pareille, la profondeur d'un point de la scène n'est pas directement accessible dans une unique capture. Le calcul de la position spatiale des points observés dans une scène n'est possible sauf si elle est prise, en au moins deux images, sous des angles de vues légèrement (au moins) différents. La combinaison de plusieurs images d'une même scène est appelée la Stéréovision. L'utilisation de deux images seulement est la Stéréovision la plus simple appelée binoculaire.

## 1.2 Motivation

Dans ce contexte, ce travail de thèse a pour but de réaliser une des tâches assurée par le système visuel humain. On travaille sur les trois capacités de la vision par ordinateur pour analyser une scène et pouvoir estimer la forme des objets présents et les distances qui les sépare. L'objectif est d'arriver a une reconstruction tridimensionnelle d'une scène observée par une ou plusieurs caméras. Le modèle utilisé dans tout ce document est le modèle sténopé (Pinhole model).

Le processus de reconstruction des formes volumiques est composé de plusieurs sous-tâches. La thèse est composée d'une introduction, de cinq chapitre dont quatre explique ces différentes étapes de la reconstruction et une conclusion. Le document se résume comme suit :

- Ayant utilisé des métaheuristiques, le deuxième chapitre introduit ces approches appelées *approches bio-inspirées* qui s'inspirent des systèmes biologiques, et des comportements des êtres vivants. Cette partie va se concentrer sur l'optimisation par essaims de particules (PSO) qui nous intéressent dans ce travail. On a utilisé cette dernière pour résoudre l'Homographie qui est un des problèmes rencontrés durant le processus la reconstruction 3D.
- Le troisième chapitre consiste a la Stéréovision : C'est l'analyse de deux ou plusieurs images d'une même scène prises sous des différents angles. Le chapitre explique tous ce qui est nécessaire pour calculer les correspondances entre ces

images, faire la liaison entre les points d'intérêts, ensuite corriger ces images pour faciliter le processus d'appariement et de reconstruction. Des résultats sont présentés illustrant les sorties du logiciel élaboré à partir de ce travail.

- Le quatrième chapitre est dédié à une transformation projective très importante utilisée dans toute les étapes du processus de reconstruction. Cette projection est dite *Homographie*. On expliquera le concept d'homographie, ainsi que la méthode classique analytique du calcul de cette projection. Ensuite, on propose une méthode pour résoudre ce problème en utilisant l'optimisation par essaim de particules. Une comparaison est alors effectuée entre les résultats traditionnels et ceux obtenus par notre approche.
- La calibration des caméras est l'étape suivante dans le processus de reconstruction. Dans le cinquième chapitre on explique le procédé de calcul des paramètres de la caméra. Ces paramètres sont utilisé par l'appareil à photo pour capturer l'image (les paramètres intrinsèque, les paramètres extrinsèques). Le chapitre explique essentiellement la méthode présenté par Zhang dans [104]. Des résultats seront inclus.
- Le dernier chapitre est l'utilisation de tout les résultats présentés auparavant pour effectuer la reconstruction 3D de la scène à partir de plusieurs images prises de la même scène.
- On conclu notre travail avec quelques points et observations concernant la reconstruction 3D et les difficultés rencontrés durant ce processus. On propose quelques perspectives qui reste à réaliser dans le future.

# Chapitre 2

## Heuristique et Métaheuristique

### Sommaire

---

<b>2.1</b>	<b>Création et origine . . . . .</b>	<b>5</b>
<b>2.2</b>	<b>Classification des métaheuristiques . . . . .</b>	<b>6</b>
2.2.1	Les approches à base de solution unique “trajectoire” . . . . .	6
2.2.2	Les approches à base de population de solutions “évolutionnaires” . . . . .	6
<b>2.3</b>	<b>Approches trajectoires . . . . .</b>	<b>7</b>
2.3.1	Le recuit simulé (simulated annealing) . . . . .	7
2.3.2	La recherche Tabou (Tabu Search) . . . . .	9
2.3.3	La recherche à voisinage variable . . . . .	11
<b>2.4</b>	<b>Approches populations “évolutionnaires” . . . . .</b>	<b>12</b>
2.4.1	Les algorithmes génétiques . . . . .	12
2.4.2	Les colonies de fourmis . . . . .	17
2.4.3	L’optimisation par colonies d’abeilles . . . . .	19
2.4.4	L’optimisation par essaim de particules . . . . .	23

---

Après avoir cadré notre objectif de travail ainsi que le plan de résolution proposé, il est important d'introduire au premier lieu l'utilisation des métaheuristiques, vu que l'une des approches détaillées sera à la base de notre proposition sur l'homographie. Ainsi, ce premier chapitre est dédié aux approches bio-inspirées, leurs principe de fonctionnement et ces domaines d'applications. Aussi pour chaque algorithme décrit des avantages et des inconvénients sont présentés.

## 2.1 Création et origine

Afin d'améliorer le comportement d'un algorithme dans son exploration de l'espace des solutions d'un problème donné, le recours à une méthode heuristique (du verbe grec *heuriskein*, qui signifie "trouver") permet de guider le processus dans sa recherche des solutions optimales. Feigenbaum et Feldman (1963) [34] définissent une heuristique comme une règle d'estimation, une stratégie, une astuce, une simplification, ou tout autre sorte de système qui limite drastiquement la recherche des solutions dans l'espace des configurations possibles. Newell, Shaw et Simon (1957) [81] précisent qu'un processus heuristique peut résoudre un problème donné, mais n'offre pas la garantie de le faire. Dans la pratique, certaines heuristiques sont connues et ciblées sur un problème particulier.

La métaheuristique, elle, se place à un niveau plus général encore, et intervient dans toutes les situations où l'ingénieur ne connaît pas d'heuristique efficace pour résoudre un problème donné, ou lorsqu'il estime qu'il ne dispose pas du temps nécessaire pour en déterminer une.

En 1996, I.H. Osman et G. Laporte [83] définissaient la métaheuristique comme "un processus itératif qui subordonne et qui guide une heuristique, en combinant intelligemment plusieurs concepts pour explorer et exploiter tout l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions optimales, ou presque-optimales". En 2006, le réseau Metaheuristics ([metaheuristics.org](http://metaheuristics.org)) définit les métaheuristiques comme "un ensemble de concepts utilisés pour définir des méthodes heuristiques, pouvant être appliqués à une grande variété de problèmes. On peut voir la métaheuristique comme une "boîte à outils" algorithmique, utilisable pour résoudre différents problèmes d'optimisation, et ne nécessitant que peu de modifications pour qu'elle puisse s'adapter à un problème particulier". Elle a donc pour objectif de pouvoir être programmée et testée rapidement

sur un problème. Comme l’heuristique, la métaheuristique n’offre généralement pas de garantie d’optimalité, bien qu’on ait pu démontrer la convergence de certaines d’entre elles. Non déterministe, elle incorpore souvent un principe stochastique pour surmonter l’explosion combinatoire. Elle fait parfois usage de l’expérience accumulée durant la recherche de l’optimum, pour mieux guider la suite du processus de recherche[8].

## 2.2 Classification des métaheuristicques

Les métaheuristicques regroupe une très grande variété d’approche. Ces approches sont selon la taille de la population se divisent en deux grande catégories :

### 2.2.1 Les approches à base de solution unique “trajectoire”

Ces algorithmes partent d’une solution initiale (obtenue de façon exacte, ou par tirage aléatoire) et s’en éloignent progressivement, pour réaliser une trajectoire, un parcours progressif dans l’espace des solutions. Dans cette catégorie, se rangent :

- La méthode de descente.
- Le recuit simulé.
- La méthode Tabou.
- La recherche par voisinage variable.

### 2.2.2 Les approches à base de population de solutions “évolutionnaires”

Elles consistent à travailler avec un ensemble de solutions simultanément, que l’on fait évoluer graduellement. L’utilisation de plusieurs solutions simultanément permet naturellement d’améliorer l’exploration de l’espace des configurations. cette catégorie regroupe :

- Les algorithmes génétiques.
- Les algorithmes par colonies de fourmis.
- L’optimisation par essaim de particules.
- Les algorithmes à estimation de distribution.
- Le path relinking (ou chemin de liaison).

On doit dire que ces métaheuristiques évolutionnaires sont probablement plus gourmandes en calcul, mais on peut dire aussi qu’elles se prêteront bien à leur parallélisation. Certains algorithmes peuvent se ranger dans les deux catégories à la fois, comme la méthode GRASP, qui construit un ensemble de solutions, qu’elle améliore ensuite avec une recherche locale. La plupart des méthodes développées ces dernières années sont d’ailleurs souvent à cheval sur ces deux approches.

Une autre manière, plus intuitive, de classer les métaheuristiques consiste à les séparer celles qui se sont inspirées d’un phénomène naturel appelées **Bio-Inspirée**, de celles qui ne le sont pas. Les algorithmes génétiques ou les algorithmes de colonies de fourmis entrent clairement dans le bio-inspiré, tandis que la méthode de descente, ou la recherche Tabou, entrent dans la seconde catégorie.

On peut également raisonner par rapport à l’usage que font les métaheuristiques de la fonction objectif. Certaines la laissent “telle quelle” d’un bout à l’autre du processus de calcul, tandis que d’autres la modifient en fonction des informations collectées au cours de l’exploration – l’idée étant toujours de “s’échapper” d’un minimum local, pour avoir davantage de chance de trouver l’optimal. La recherche locale guidée est un exemple de métaheuristique qui modifie la fonction objectif.

Enfin, il faut distinguer les métaheuristiques qui ont la faculté de mémoriser des informations à mesure que leur recherche avance, de celles qui fonctionnent sans mémoire, en aveugle, et qui peuvent revenir sur des solutions qu’elles ont déjà examinées. On distingue la mémoire à court terme (celles des derniers mouvements effectués), et la mémoire à long terme (qui concerne des paramètres synthétiques plus généraux). Le meilleur représentant des métaheuristiques avec mémoire reste la recherche Tabou. Dans les “sans mémoire”, on trouve le recuit simulé par exemple [8].

## 2.3 Approches trajectoires

### 2.3.1 Le recuit simulé (simulated annealing)

Le recuit simulé (SA) a été introduit par [58] et [109] comme une méthode de recherche locale normale, utilisant une stratégie pour éviter les minima locaux. Cette métaheuristique est basée sur une technique utilisée depuis longtemps par les métallurgistes qui, pour obtenir un alliage sans défaut, faisant alterner les cycles de réchauffage (ou de recuit) et de refroidissement lent des métaux. Le recuit simulé s’appuie sur des

travaux faites par [74], qui ont pu décrire l'évolution d'un système en thermodynamique [29]. Le principe du recuit simulé est de parcourir de manière itérative l'espace des solutions. On part avec une solution notée  $s_0$  initialement générée de manière aléatoire dont correspond une énergie initiale  $E_0$ , et une température initiale  $T_0$  généralement élevée. A chaque itération de l'algorithme, un changement élémentaire est effectué sur la solution, cette modification fait varier l'énergie du système  $\delta E$ . Si cette variation est négative (la nouvelle solution améliore la fonction objective, et permet de diminuer l'énergie du système), elle est acceptée. Si la solution trouvée est moins bonne que la précédente alors elle sera acceptée avec une probabilité  $P$  calculée suivant la distribution de Boltzmann suivante :

$$P(E, T) = \exp^{-\frac{\delta E}{T}}$$

En fonction du critère de Metropolis, un nombre  $\epsilon \in [0, 1]$  est comparé à  $\delta E$  la probabilité  $p = \exp^{-\frac{\delta E}{T}}$ . Si  $p \leq \epsilon$  la nouvelle solution est acceptée. Le fonctionnement du critère de Metropolis est interprété par

- Si  $\delta E = f(s') - f(s) < 0$  alors  $e^{-T} > 1$ , donc est toujours inférieur à cette valeur, et on accepte la solution  $s'$
- Si  $\delta E > 0$ 
  - et  $T$  est très grande, alors  $e^{-\frac{\delta E}{T}} \cong 1$ , tout voisin est systématiquement accepté.
  - et  $T$  est très petite, alors  $e^{-\frac{\delta E}{T}} \cong 0$ , une dégradation a peu de chances d'être acceptée.

Le choix de la température est primordial pour garantir l'équilibre entre l'intensification et la diversification des solutions dans l'espace de recherche. Premièrement, le choix de la température initiale dépend de la qualité de la solution de départ. Si cette solution est choisie aléatoirement, il faut prendre une température relativement élevée. On utilise souvent la règle suivante :

$$T_{k+1} \leftarrow T_k \cdot \alpha$$

où  $\alpha \in [0, 1]$ , un paramètre qui exprime la diminution de la température de l'itération  $k$  à  $k + 1$ . La décroissance de la température peut également être réalisée par paliers (en d'autres termes, elle ne change qu'après un certain nombre d'itérations). Certains préconisent l'utilisation de stratégies non monotones. On peut ainsi rehausser la température lorsque la recherche semble bloquée dans une région de l'espace de recherche. On peut alors considérer une grande augmentation de la température comme un processus de diversification alors que la décroissance de la température correspond à un processus d'intensification.

## Avantages

- La méthode du recuit simulé a l'avantage d'être souple vis-à-vis des évolutions du problème et facile à implémenter.
- Donne généralement de bonnes solutions par rapport aux algorithmes de recherche classiques.
- Peut être utilisé dans la plupart des problèmes d'optimisation.
- Contrairement aux méthodes de descente, SA évite le piège des optima locaux,
- L'approche converge vers un optimum global (lorsque le nombre d'itérations tend vers l'infini). Cela fait de lui une option attrayante pour les problèmes d'optimisation difficiles.

**Inconvénients** Le principal inconvénient du recuit simulé est qu'une fois l'algorithme piégé à basse température dans un minimum local, il lui est impossible de s'en sortir. Plusieurs solutions ont été proposées pour tenter de résoudre ce problème, par exemple en acceptant une brusque remontée de la température de temps en temps, pour relancer la recherche sur d'autres régions plus éloignées. Mise à part cela, on peut citer quelques autres inconvénients comme :

- La difficulté de déterminer la température initiale :
  - Si elle est trop basse, la qualité de recherche sera mauvaise.
  - Si elle est trop haute, le temps de calcul sera élevé.
- Définir les voisinages permettant un calcul efficace de  $\delta E$ .
- L'impossibilité de savoir si la solution trouvée est optimale.

### 2.3.2 La recherche Tabou (Tabu Search)

La recherche tabou (TS) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. La recherche tabou est introduite principalement par Glover [39], Hansen [48], Glover et Laguna dans [40]. Cette méthode a montré une grande efficacité pour la résolution des problèmes d'optimisation difficiles. En effet, à partir d'une solution initiale  $s$  dans un ensemble de solutions local  $S$ , des sous-ensembles de solution  $N(s)$  appartenant au voisinage  $S$  sont générés. Par l'intermédiaire de la fonction d'évaluation nous retenons la solution qui améliore la valeur de  $f$ , choisie parmi l'ensemble de solutions voisines  $N(s)$ . L'algorithme accepte parfois des solutions qui n'améliorent pas toujours la solution courante. Nous mettons en oeuvre une liste

tabou (tabu list)  $T$  de longueur  $k$  contenant les  $k$  dernières solutions visitées, ce qui ne donne pas la possibilité à une solution déjà trouvée d'être acceptée et stockée dans la liste tabou. Alors le choix de la prochaine solution est effectué sur un ensemble des solutions voisines en dehors des éléments de cette liste tabou. Quand le nombre  $k$  est atteint, chaque nouvelle solution sélectionnée remplace la plus ancienne dans la liste. La construction de la liste tabou est basée sur le principe FIFO, c'est-à-dire le premier entré est le premier sorti. Comme critère d'arrêt on peut par exemple fixer un nombre maximum d'itérations sans amélioration de  $s^*$ , ou bien fixer un temps limite après lequel la recherche doit s'arrêter [29].

La méthode Tabou est une technique de recherche dont le principe a été proposé pour la première fois par Fred Glover dans les années 80, et est devenue très classique en optimisation combinatoire. Elle se distingue des méthodes de recherche locale simples par le recours à un historique des solutions visitées, de façon à rendre la recherche un peu moins "aveugle". Il devient donc possible de s'extraire d'un minimum local, mais, pour éviter d'y retomber périodiquement, certaines solutions sont bannies, elles sont rendues "taboues".

A l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine à chaque itération, Tabou examine un échantillonnage de solutions de  $N(s)$  et retient la meilleure  $s'$  même si  $f(s') > f(s)$ . La recherche Tabou ne s'arrête donc pas au premier optimum trouvé.

Le danger serait alors de revenir à  $s$  immédiatement, puisque  $s$  est meilleure que  $s'$ . Pour éviter de tourner ainsi en rond, on crée une liste  $T$  qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste  $T$  est appelée liste Tabou [8].

**Avantages et inconvénients** La méthode Tabou est une méthode de recherche locale, et la structure de son algorithme de base est finalement assez proche de celle du recuit simulé, donc on passe facilement de la première heuristique vers celle-ci, avec l'avantage, par rapport au recuit simulé, d'avoir un paramétrage simplifié : dans un premier temps, le paramétrage consistera d'abord à trouver une valeur indicative  $t$  d'itérations pendant lesquelles les mouvements sont interdits. Il faudra également décider d'une stratégie de mémorisation à long terme – sur la qualité des solutions, sur leur recense, ou sur leur qualité...

En revanche, la méthode Tabou exige une gestion de la mémoire de plus en plus

lourde à mesure que l'on voudra raffiner le procédé en mettant en place des stratégies de mémorisation complexe.

L'efficacité de la méthode Tabou fait qu'elle est largement employée dans les problèmes d'optimisation combinatoire : elle a été testée avec succès sur les grands problèmes classiques (voyageur de commerce, ordonnancement d'ateliers) et elle est fréquemment appliquée sur les problèmes de constitution de planning, de routage, d'exploration géologique, etc [8].

### 2.3.3 La recherche à voisinage variable

La recherche à voisinage variable est une méthode basée sur la recherche locale. Elle a été introduite en 1997 par Mladenovic et Hansen [77]. La recherche à voisinage variable est notée par RVV ou encore VNS (de l'anglais : Variable Neighbourhood Search). Elle se base sur un simple principe de recherche fondé sur le changement systématique du voisinage. A l'opposé d'autres méthodes basées sur la recherche locale, la VNS n'utilise pas un seul voisinage pour exploiter la solution courante. Par contre, elle utilise plusieurs voisinages dans un ordre prédéfini. Son principe de base consiste à entamer la recherche avec un voisinage  $V_1$  d'une solution initiale  $s$ . Lorsque la recherche se bloque dans un optimum local, on continue la recherche avec un autre voisinage puis on revient au premier voisinage après chaque amélioration. La recherche continue en passant d'un voisinage à un autre à chaque blocage dans la vallée de l'optimum local du voisinage utilisé.

L'idée de parcourir un ensemble de voisinage permet de sauver la recherche du blocage causé par le piège de l'optimum local. Cette stratégie permet de diversifier l'exploration de l'espace de recherche et donne au processus de recherche plus de chances de rencontrer l'optimum global. Cependant, le choix et l'ordre de parcours des différents voisinages jouent un rôle non négligeable dans la performance de l'algorithme. En fait, s'il est bien étudié, la recherche converge vers la solution optimale (l'optimum global), sinon la convergence de la solution optimale ne sera pas garantie.

## 2.4 Approches populations “évolutionnaires”

### 2.4.1 Les algorithmes génétiques

Les algorithmes génétiques (AG) sont des algorithmes d’optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils ont été adaptés à l’optimisation par John Holland [55], également les travaux de David Goldberg ont largement contribué à les enrichir [42]. Le vocabulaire utilisé est le même que celui de la théorie de l’évolution et de la génétique, on emploie le terme individu (solution potentielle), population (ensemble de solutions), génotype (une représentation de la solution), gène (une partie du génotype), parent, enfant, reproduction, croisement, mutation, génération, etc. Leur fonctionnement est extrêmement simple : on part d’une population de solutions potentielles (chromosomes) initiales, arbitrairement choisies [29], on évalue leurs performances (Fitness) relative et à la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples tels que : la sélection, le croisement et la mutation. Quelques individus se reproduisent, d’autres disparaissent et seuls les individus les mieux adaptés sont supposés survivre. On recommence ce cycle jusqu’à ce qu’on trouve une solution satisfaisante. En effet, l’héritage génétique à travers les générations permet à la population d’être adaptée et donc répondre au critère d’optimisation. Un algorithme génétique recherche le ou les extrema d’une fonction définie sur un espace de données. Son mise en oeuvre nécessite :

- **le codage des données** La première étape est de définir et coder convenablement le problème. Cette étape associe à chaque point de l’espace de recherche une structure de données spécifique, appelée génotype ou ensemble de chromosomes, qui caractérisera chaque individu de la population. Le codage de chaque individu en séquence est essentielle dans l’élaboration d’un algorithme génétique dont dépend notamment l’implémentation des opérateurs de transformations. Ainsi, cette phase détermine la structure de données qui sera utilisée pour coder le génotype des individus de la population. Le codage doit donc être adapté au problème traité. Plusieurs types de codages sont utilisés dans la littérature, les premiers résultats théoriques sur les algorithmes génétiques ont opté pour un codage par une séquence binaire de longueur fixe à travers la notion de schéma [42]. L’efficacité de l’algorithme génétique dépend donc du choix convenable du type de codage.
- **Génération de la population initiale** La génération de la population ini-

tiale, c'est-à-dire le choix des dispositifs de départ que nous allons faire évoluer, ce choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Néanmoins, une initialisation aléatoire est plus simple à réaliser : les valeurs des gènes sont tirées au hasard selon une distribution uniforme. Toutefois, il peut être utile de guider la génération initiale vers des sous domaines intéressants de l'espace de recherche. Par exemple lors d'une recherche d'optima dans un problème d'optimisation sous contraintes, il est préférable de produire des éléments satisfaisant les contraintes. La population initiale doit être suffisamment diversifiée et de taille assez importante pour que la recherche puisse parcourir l'espace d'état dans un temps limité.

- **Fonction d'adaptation (Fitness)** L'évaluation de la Fitness est généralement l'étape dans laquelle on mesure la performance de chaque individu. Pour pouvoir juger la qualité d'un individu et ainsi le comparer aux autres, il faut établir une mesure commune d'évaluation. Aucune règle n'existe pour définir cette fonction, son calcul peut ainsi être quelconque, que ce soit une simple équation ou une fonction affine. La manière la plus simple est de poser la fonction d'adaptation comme la formalisation du critère d'optimisation.
- **Sélection** La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais, pendant le passage d'une génération à une autre, ce processus est basé sur la performance de l'individu. L'opérateur de sélection doit être conçu pour donner également une chance aux mauvais éléments, car ces éléments peuvent, par croisement ou mutation, engendrer une descendance pertinente par rapport au critère d'optimisation. Il existe différentes techniques de sélection, on propose quelques unes.

**Sélection par tournoi :** Cette méthode est celle avec laquelle on obtient les résultats les plus satisfaisants. Le principe de cette méthode est le suivant : on effectue un tirage avec remise de deux individus de  $P$ , et on les fait "combattre". Celui qui a la fitness la plus élevée l'emporte avec une probabilité  $p$  comprise entre 0.5 et 1. On répète ce processus  $n$  fois de manière à obtenir les  $n$  individus de  $P'$  qui serviront de parents. La variance de cette méthode est élevée et le fait d'augmenter ou de diminuer la valeur de  $p$  permet respectivement de diminuer ou d'augmenter la pression de la sélection [6].

**Élitisme :** Cette méthode consiste à sélectionner les  $n$  individus dont on a besoin pour la nouvelle génération  $P'$  en prenant les  $n$  meilleurs individus de la population  $P$  après l'avoir triée de manière décroissante selon la fitness de ses individus. Il est inutile de préciser que cette méthode est encore pire que celle

de la loterie biaisée dans le sens où elle amènera à une convergence prématurée encore plus rapidement et surtout de manière encore plus sûre que la méthode de sélection de la loterie biaisée ; en effet, la pression de la sélection est trop forte, la variance nulle et la diversité inexistante, du moins le peu de diversité qu'il pourrait y avoir ne résultera pas de la sélection mais plutôt du croisement et des mutations. Là aussi il faut opter pour une autre méthode de sélection [6].

**Sélection par roulette :** Cette méthode est la plus connue et la plus utilisée. Avec cette méthode chaque individu a une chance d'être sélectionné proportionnelle à sa performance, donc plus les individus sont adaptés au problème, plus ils ont de chances d'être sélectionnés. Pour utiliser l'image de la "roue du forain", chaque individu se voit attribué un secteur dont l'angle est proportionnel à son adaptation, sa "fitness", voir la figure 2.1.

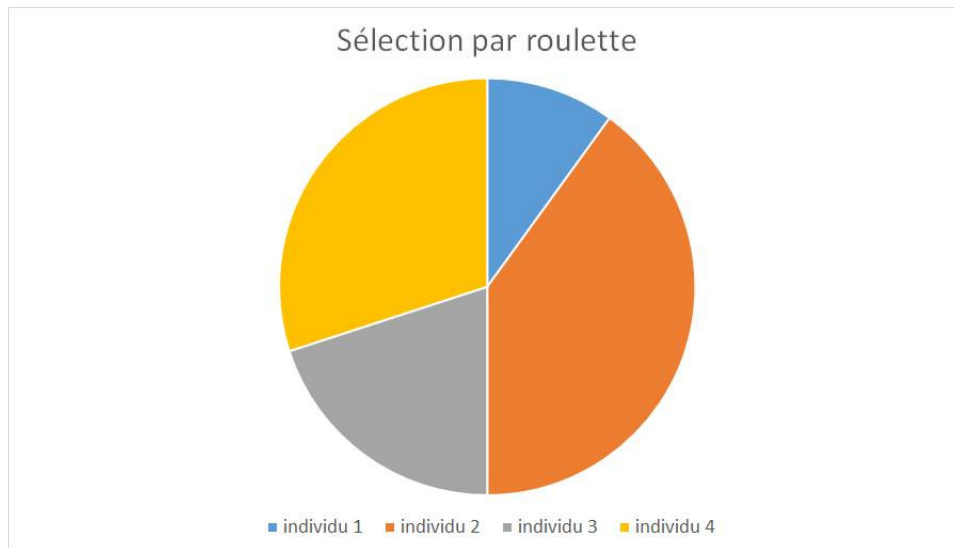


FIGURE 2.1: Sélection par roulette

On fait tourner la roue et quand elle cesse de tourner on sélectionne l'individu correspondant au secteur désigné par une sorte de "curseur", curseur qui pointe sur un secteur particulier de celle-ci après qu'elle se soit arrêté de tourner [6]. Cette méthode, bien que largement répandue, a pas mal d'inconvénients :

En effet, elle a une forte variance. Il n'est pas impossible que sur  $n$  sélections successives destinées à désigner les parents de la nouvelle génération  $P'$ , la quasi-totalité, voire pire la totalité des  $n$  individus sélectionnés soient des individus ayant une fitness vraiment mauvaise et donc que pratiquement aucun individu voire aucun individu a forte fitness ne fasse partie des parents de la nouvelle génération. Ce phénomène est bien sûr très dommageable car cela

va complètement à l'encontre du principe des algorithmes génétiques qui veut que les meilleurs individus soient sélectionnés de manière à converger vers une solution la plus optimale possible.

A l'inverse, on peut arriver à une domination écrasante d'un individu "localement supérieur". Ceci entraînant une grave perte de diversité. Imaginons par exemple qu'on ait un individu ayant une fitness très élevée par rapport au reste de la population, disons dix fois supérieure, il n'est pas impossible qu'après quelques générations successives on se retrouve avec une population ne contenant que des copies de cet individu. Le problème est que cet individu avait une fitness très élevée, mais que cette fitness était toute relative, elle était très élevée mais seulement en comparaison des autres individus. On se retrouve donc face à problème connu sous le nom de "convergence prématurée"; l'évolution se met donc à stagner et on atteindra alors jamais l'optimum, on restera bloqué sur un optimum local. Il existe certaines techniques pour essayer de limiter ce phénomène, comme par exemple le "scaling", qui consiste à effectuer un changement d'échelle de manière à augmenter ou diminuer de manière forcée la fitness d'un individu par rapport à un autre selon leur écart de fitness. Malgré tout, il est conseillé d'opter plutôt pour une autre méthode de sélection.

**Reste stochastique sans remplacement :** Cette méthode semble être très peu utilisée et qui plus est possède une variance faible, donc introduit peu de diversité, nous n'entrerons donc pas dans les détails, on se contentera d'exposer sa mise en oeuvre : On prend l'image d'un segment découpé en autant de sous-segments qu'il y a d'individus. Les individus sélectionnés sont désignés par un ensemble de points équidistants [6].

- **Croisement** L'opérateur de croisement favorise l'exploration de l'espace de recherche et enrichit la diversité de la population en manipulant la structure des chromosomes, le croisement fait avec deux parents et génère deux enfants, en espérant qu'un des deux enfants au moins héritera de bons gènes des deux parents et sera mieux adapté qu'eux. Il existe plusieurs méthodes de croisement par exemple le croisement en un point, ou en multiples points voir les figures 2.3,2.4,2.2.

**Croisement uniforme :** Ce type de croisement est fondé sur la probabilité. En fait, il permet la génération d'un enfant en échangeant chaque gène des deux parents avec une probabilité égale à 0.5, comme le montre la figure 2.2.

**Croisement 1-point :** consiste à choisir un seul point de coupure, puis échanger les fragments situés après ce point de coupure. Comme le montre la

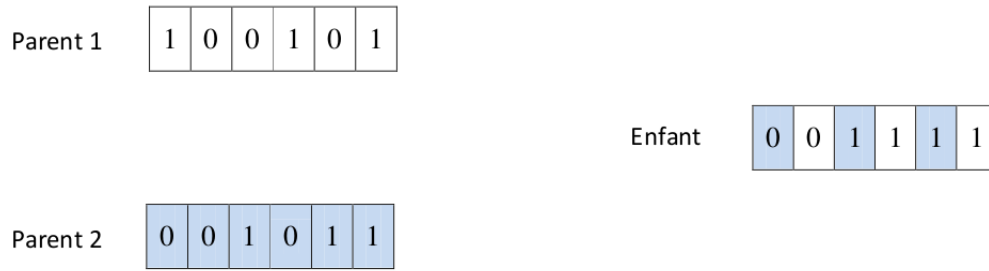


FIGURE 2.2: Croisement uniforme

figure 2.3.

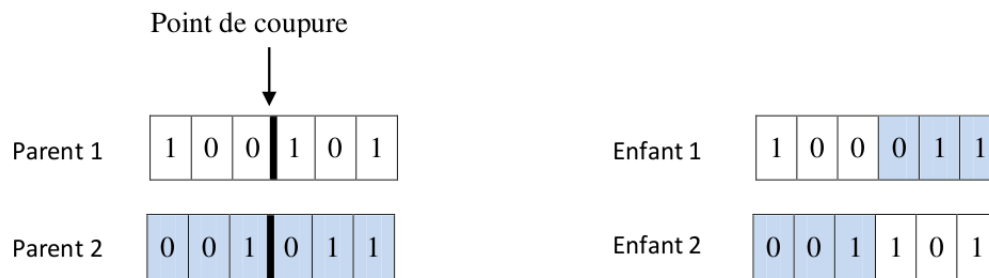


FIGURE 2.3: Croisement 1-point

**Le croisement 2-point** : consiste à choisir deux points de coupure, puis échanger les fragments situés entre ces deux points comme le montre la figure 2.4.

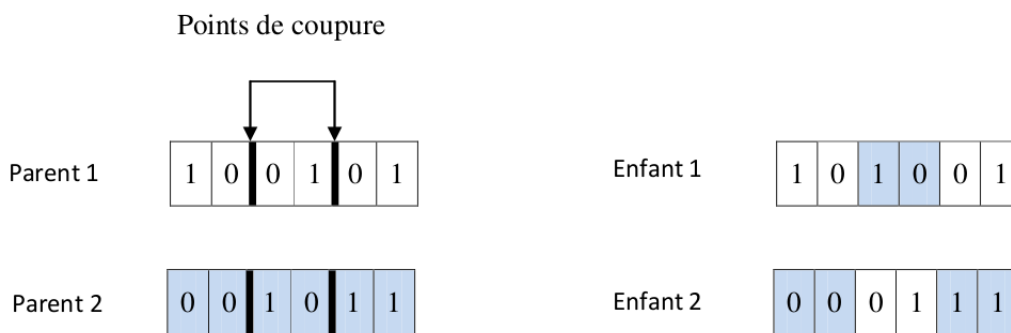


FIGURE 2.4: Croisement 2-point

- **Mutation** L'opérateur de mutation est un processus où un changement mineur du code génétique appliqué à un individu pour introduire de la diversité et ainsi d'éviter de tomber dans des optimums locaux. Cet opérateur est appliqué avec une probabilité  $P_m$  généralement inférieure à celle du croisement  $P_c$ .

L'efficacité des algorithmes génétiques dépend fortement du réglage des différents paramètres caractérisant ces algorithmes, et qui sont parfois difficiles à déterminer. Des paramètres comme la taille de la population, le nombre maximal des générations, la probabilité de mutation  $p_m$ , et la probabilité de croisement  $p_c$ . Les deux premiers paramètres dépendent directement de la nature du problème et de sa complexité, et leurs choix doit représenter un compromis entre la qualité des solutions et le temps d'exécution. La probabilité de croisement  $p_c$  est liée à la forme de la fonction d'évaluation. Son choix est en général heuristique. Plus sa valeur est élevée, plus la population subit des changements importants. La probabilité de mutation  $p_m$  est généralement faible puisqu'un taux élevé risque de conduire vers un optimum local. En revanche, une probabilité faible permet d'assurer une bonne exploration de l'espace de recherche sans perturber la convergence. Le succès des algorithmes génétiques dépend aussi de la manière du codage des individus. Dans les problèmes combinatoires, ce codage est souvent suggéré par la nature même du problème, ce qui peut induire de meilleurs résultats.

## 2.4.2 Les colonies de fourmis

Les algorithmes de colonies de fourmis ont été proposés par Colomni, Dorigo et Maniezzo en 1992 [23] et appliquées la première fois au problème du voyageur de commerce. Ce sont des algorithmes itératifs à population où tous les individus partagent un savoir commun qui leur permet d'orienter leurs futurs choix et d'indiquer aux autres individus des choix à suivre ou à éviter. Le principe de cette métaheuristique repose sur le comportement particulier des fourmis, elles utilisent pour communiquer une substance chimique volatile particulière appelée phéromone grâce à une glande située dans leur abdomen. En quittant leur nid pour explorer leur environnement à la recherche de la nourriture, les fourmis arrivent à élaborer des chemins qui s'avèrent fréquemment être les plus courts pour aller du nid vers une source de nourriture. Chaque fourmi dépose alors une quantité de phéromones sur ces pistes qui deviendront un moyen de communication avec leurs congénères, les fourmis choisissent ainsi avec une probabilité élevée les chemins contenant les plus fortes concentrations de phéromones à l'aide des récepteurs situés dans leurs antennes [29]. La figure 2.5 illustre et confirme ce constat, une expérience a été faite par gauss et deneubourg en 1989 appelée expérience du pont à double branche, les fourmis qui retournent au nid rapidement, après avoir visité la source de nourriture, sont celles qui ont choisi la branche courte et les fourmis empruntant cette branche faisant plus d'aller retour, et par conséquent la quantité de phéromones déposée sur la plus courte branche est relativement supérieure que celle

présente sur la plus longue branche. Puisque les fourmis sont attirées plus vers les pistes de plus grande concentration en phéromones, alors la branche courte sera la plus empruntée par la majorité des fourmis. Cette métaheuristique a permis de résoudre différents problèmes d'optimisation combinatoire à forte complexité, comme le problème du voyageur de commerce [28], le problème de coloration de graphe [24], le problème d'affectation quadratique [36] ou le problème de routage de véhicules [73], etc.

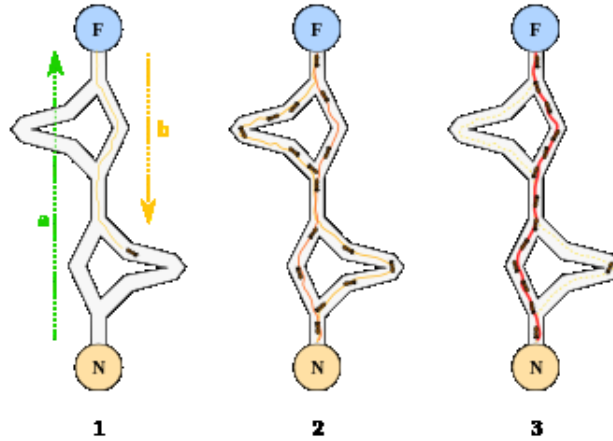


FIGURE 2.5: Principe de mouvement des fourmis

**Principe de l'algorithme** Dans ce paragraphe on décrit l'implémentation d'un algorithme de colonie de fourmis original dit 'Ant System' (AS) [22], orienté pour résoudre le problème de voyageur de commerce (TSP), ce problème consiste à trouver le plus court cycle hamiltonien dans un graphe, où chaque sommet du graphe représente une ville. La distance entre les villes  $i$  et  $j$  est représentée par  $d_{ij}$ , et le couple  $(i, j)$  représente l'arête entre ces deux villes. Nous initialisons d'abord la quantité de phéromone sur les arêtes à  $\tau_{init} > 0$ , chaque fourmi parcourt le graphe et construit un trajet complet (une solution). A chaque étape de la construction des solutions, la fourmi doit décider à quel sommet elle va se déplacer, cette décision est prise d'une manière probabiliste fondée sur les valeurs de phéromone et d'une information statistique qui permet notamment de trouver une bonne solution. La probabilité pour qu'une fourmi  $k$  se déplace du sommet  $i$  au sommet  $j$ , qui appartient à un ensemble de sommets qui ne sont pas encore visités par la fourmi  $k$  noté  $S_i^k$ , est :

$$p_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in S_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta}$$

$\alpha, \beta$  sont deux paramètres qui influencent sur l'importance de l'intensité de phéromone  $\tau_{ij}$ , et l'information statistique dite visibilité  $\eta_{ij}$ . Cette valeur guide le choix des fourmis vers des villes proches, et éviter le plus possible celles trop lointaines ( $\tau_{ij} = \frac{1}{d_{ij}}$ ). Pour  $\alpha = 0$ , on prend en compte juste la visibilité c'est-à-dire que le choix sera tombé à chaque fois sur la ville la plus proche. Si  $\beta = 0$ , seules les pistes de phéromones jouent sur le choix. Pour éviter une sélection trop rapide d'un chemin, un compromis convenable entre ces deux paramètres est obligatoire.

**Mise à jour de phéromones** Lorsque toutes les fourmis ont construit une solution, une quantité de phéromones  $\delta\tau_{ij}^k$  est déposée par chaque fourmi  $k$  sur son trajet. Pour toute itération  $t$ , si le chemin  $(i, j)$  est dans la tournée de la fourmi  $k$  la quantité de phéromones déposée sur ce trajet est :

$$\delta\tau_{ij}^k(t) = \frac{Q}{L^k(t)}$$

Où  $L^k(t)$  est la longueur totale de la tournée de la fourmi  $k$ , et  $Q$  est une constant. Donc l'ajout de la quantité de phéromones dépend certainement de la qualité de la solution obtenue c'est-à-dire plus la tournée parcourue est petite plus l'ajout de la quantité de phéromones est important. Dans le but de ne pas négliger toutes les mauvaises solutions obtenues, et ainsi éviter la convergence vers des optima locaux de mauvaise qualité, le concept d'évaporation des pistes de phéromones est simulé à travers un paramètre  $\rho$  appelé le taux d'évaporation ( $0 < \rho < 1$ ) comme suit :

$$\tau_{ij}(t+1) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \delta\tau_{ij}(t)$$

Où  $\delta\tau_{ij}(t) = \sum_{k=1}^m \delta\tau_{ij}^k(t)$ ,  $t$  représente une itération donnée et  $m$  le nombre de fourmis [29].

### 2.4.3 L'optimisation par colonies d'abeilles

#### Comportement réel des abeilles

Un modèle de comportement alimentaire d'une colonie d'abeilles a été développé à partir d'équations de réaction-diffusion[38]. Ce modèle qui conduit à l'émergence de l'intelligence collective des essaims d'abeilles se compose de trois composants essential : sources de nourriture, les fouilleurs de salariés et chômeurs butineuses, et définit deux

principaux modes de comportement de la colonie des abeilles : le recrutement à une source de nourriture et de l'abandon d'une source. Les principales composantes du modèle sont expliquées comme ci-dessous :

1. Sources alimentaires : Pour sélectionner une source de nourriture, une abeille ouvrière évalue plusieurs propriétés liées à la source de nourriture tels que sa proximité de la ruche, la richesse de l'énergie, le goût de son nectar, et la facilité ou la difficulté d'extraire cette énergie. Pour la simplicité, la qualité d'une source d'alimentation peut être représenté par une seule quantité bien que cela dépend de divers les paramètres mentionnés ci-dessus.
2. butineuses salariées : Une butineuse employée est affectée à une source de nourriture spécifique qu'elle exploite actuellement. elle carie les informations sur cette source et la partage avec d'autres abeilles qui attendent dans la ruche. L'information comprend la distance, la direction et la rentabilité de la source de nourriture.
3. butineuses chômeurs : une abeille butineuse qui cherche une source de nourriture pour l'exploiter est appelée chômeur. elle peut être soit un scout qui cherche l'environnement de façon aléatoire ou un spectateur qui tente de trouver une source de nourriture par le biais de l'information donnée par l'abeille employée. Le nombre moyen de scouts est environ 5-10%.

L'échange d'informations entre les abeilles est le cas le plus important dans la formation de la connaissance collective. En examinant l'ensemble de la ruche, il est possible de distinguer certaines parties qui existent généralement dans toutes les ruches. La plus importante partie de la ruche en ce qui concerne l'échange d'informations est le domaine de la danse. La communication entre les abeilles liées à la qualité des sources de nourriture se produit dans la zone de la danse. Cette danse est appelée danse frétillante. Étant donné que les informations sur toutes les sources riches est disponible pour un spectateur sur la piste de danse, sans doute, elle pourrait regarder nombreuses danses et choisit de s'engager à la source la plus rentable. Il ya une grande probabilité que les spectateurs choisissent des sources plus rentables car il ya plus d'informations qui circulent sur les sources les plus rentables.

Les butineuses salariées partagent leurs informations avec une probabilité qui est proportionnelle à la rentabilité de la source de nourriture, et le partage de cette information par la danse frétillante est plus long en durée. Ainsi, le recrutement est proportionnel à la rentabilité d'une source d'alimentation.

Afin de mieux comprendre les caractéristiques de comportement de base des butineuses, on examine la figure 2.6. Supposons qu'il y a deux sources alimentaires découvertes : A et B. Au tout début, une butineuse potentielle commencera comme une butineuse chômeuse. Cette abeille butineuse n'aura pas de connaissances sur les sources de nourriture autour du nid. Il ya deux options possibles pour une telle abeille :

1. Elle peut être un éclaireur et commence à rechercher spontanément autour du nid pour la nourriture en raison d'une motivation interne ou possiblement un indice externe (S sur la figure 2.6).
2. Elle peut être une recrue après avoir regardé les danses frétillantes et commence à rechercher une source de nourriture (R sur la figure 2.6).

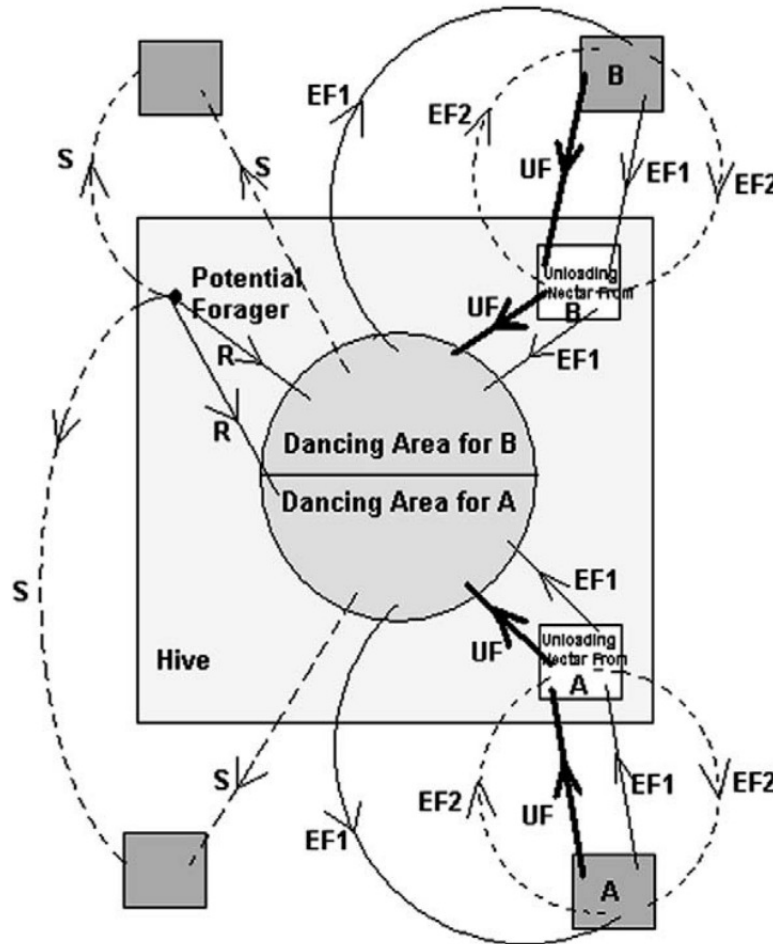


FIGURE 2.6: Comportement des abeilles butineuses pour le nectar

Après avoir trouvé la source de nourriture, l'abeille utilise sa propre capacité à mémoriser l'emplacement, puis commence immédiatement à l'exploiter. Ainsi, l'abeille butineuse deviendra une salariée. La butineuse prend une charge de nectar de la source

et retourne à la ruche pour la décharger dans un magasin d'alimentation. Après le déchargement de la nourriture, l'abeille a les options suivantes :

1. Elle peut devenir un disciple engagé après avoir abandonné la source de nourriture (UF).
2. Elle peut danser puis recruter des camarades pour revenir à la même source alimentaire (EF1).
3. Elle peut continuer à se nourrir de la source de nourriture sans recruter des abeilles (EF2).

Il est important de noter que toutes les abeilles ne commencent pas le butinage simultanément. Les expériences ont confirmé que les nouvelles abeilles commencent la recherche de nourriture à une vitesse proportionnelle à la différence entre le nombre total éventuelle des abeilles et le nombre d'abeilles actuellement à la recherche de nourriture.

### **Algorithme de colonie d'abeilles artificielles**

Dans l'algorithme ABC, la position d'une source de nourriture représente une solution possible du problème de l'optimisation et la quantité de nectar d'une source alimentaire correspond à la qualité (fitness) de la solution associée. Le nombre des abeilles salariées ou les abeilles spectatrices est égale au nombre de solutions dans la population. À la première étape, l'ABC génère une population initiale distribuée au hasard  $P(C = 0)$  de  $SN$  solutions (positions de source de nourriture), où  $SN$  désigne la taille des abeilles salariés ou abeilles spectatrices. Chaque solution  $x_i (i = 1, 2, \dots, SN)$  est un vecteur de dimension  $D$ . Ici,  $D$  est le nombre de paramètres d'optimisation.

Après l'initialisation, la population des positions (solutions) est soumise à des cycles répétés,  $C = 1, 2, \dots, MCN$ , des processus de recherche des abeilles employées, des abeilles spectatrices et des éclaireuses. Une abeille employée produit une modification de la position (solution) dans sa mémoire en fonction de l'information locale (information visuelle) et teste la quantité de nectar (valeur de fitness) de la nouvelle source (nouvelle solution). Si la quantité de nectar de la nouvelle position est plus élevée que celle de la précédente, l'abeille mémorise la nouvelle position et oublie l'ancienne. Sinon, elle maintient la position de la précédente dans sa mémoire. Après que toutes les abeilles salariés complètent le processus de recherche, elles partagent les informations du nectar des sources de nourriture et leurs informations de position avec les abeilles spectatrices. Une abeille spectatrice évalue les informations du nectar tirées de toutes les abeilles employées et choisit une source de nourriture avec une probabilité liée à son montant

de nectar. Comme dans le cas de l'abeille employée, elle produit une modification de la position dans sa mémoire et vérifie la quantité de nectar de la source candidate. Si le nectar est plus élevée que celle de la précédente, l'abeille mémorise la nouvelle position et oublie l'ancienne. L'algorithme 1 résume le comportement des abeilles artificielles décrit ci-dessus.

**repeat**

- Placer les abeilles employées sur leurs sources de nourriture.
- Placer les abeilles spectatrices sur les sources de nourriture selon la quantité de nectar correspondante.
- Envoyer les scouts à la zone de recherche pour découvrir des nouvelles sources alimentaires.
- Mémoriser la meilleure source d'alimentation jusque ici.

**until** Critère d'arrêt;

**Algorithm 1:** Algorithme de base

#### 2.4.4 L'optimisation par essaim de particules

L'Optimisation par Essaim Particulaire (OEP, ou PSO en anglais) a été proposée par Kennedy et Eberhart [56]. Cette méthode est inspirée du comportement social des animaux évoluant en essaim[29]. Au départ, ils cherchaient à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale. Les particules sont les individus et elles se déplacent dans l'hyperespace de recherche en se basant sur des informations limitées :

1. Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a tendance à retourner vers ce point.
2. Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point.

Chaque individu utilise donc, non seulement, sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement. Des règles simples, telles que "aller à la même vitesse que les autres", "se déplacer dans la même direction" ou encore "rester proche de ses voisins" sont des exemples de comportements qui suffisent à maintenir la cohésion de l'essaim. Le déplacement d'une particule est influencé par les trois types de comportement :

1. Une composante physique : la particule tend à suivre sa propre voie.
2. Une composante cognitive : la particule tend à revenir vers le meilleur site par lequel elle est déjà passée.

3. Une composante sociale : la particule tend à se diriger vers le meilleur site déjà atteint par ses voisins.

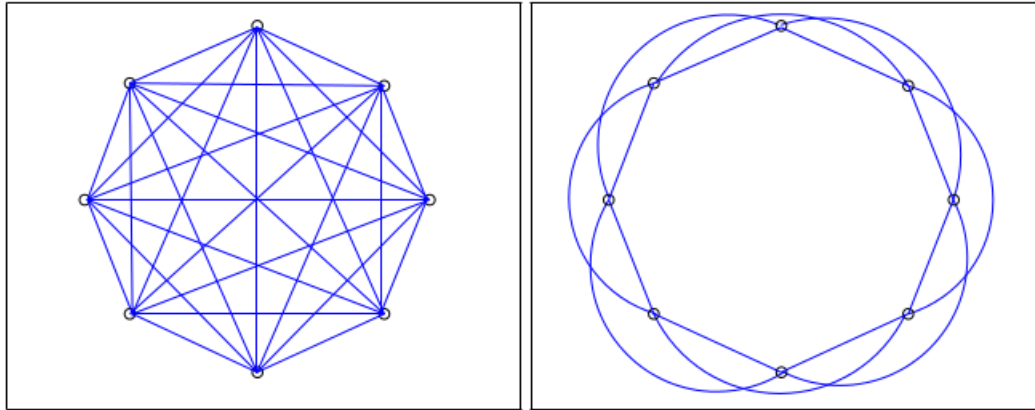
## Le voisinage

[30] Le voisinage constitue la structure du réseau social. Les particules à l'intérieur d'un voisinage communiquent entre-elles. Différents voisinages ont été étudiés [57] et sont considérés en fonction des identificateurs des particules et non des informations topologiques comme les distances euclidiennes dans l'espace de recherche :

- Topologie en étoile (figure 2.7(a)) : le réseau social est complet, chaque particule est attirée vers la meilleure particule notée  $g_{best}$  et communique avec les autres.
- Topologie en anneau (figure 2.7(b)) : chaque particule communique avec  $n$  ( $n = 3$ ) voisines immédiates. Chaque particule tend à se déplacer vers la meilleure dans son voisinage local notée  $l_{best}$ .
- Topologie en rayon (figure 2.7(c)) : une particule "centrale" est connectée à toutes les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration l'information est propagée aux autres.

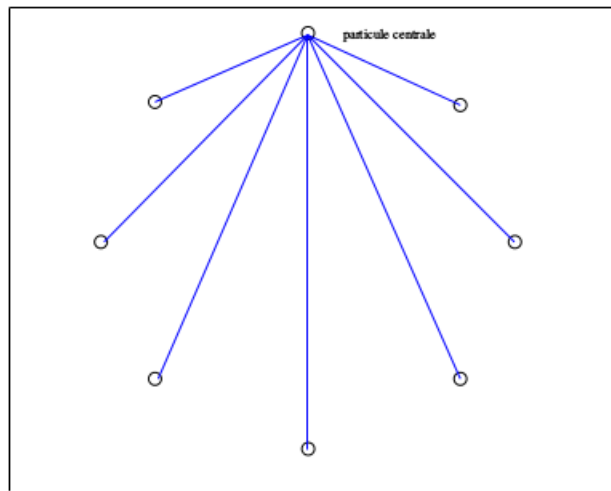
## L'OEP basique

L'OEP est une méthode inspirée du comportement sociologique des animaux qui évoluent en essaims, tels que les poissons qui se déplacent en bancs ou les oiseaux migrateurs. L'OEP est une méthode stochastique d'optimisation qui déplace un essaim de "particules" candidates, qui "volent" au-dessus de l'espace de recherche. Le déplacement de chaque particule est influencé par sa propre vitesse, la meilleure position qu'elle a atteinte ainsi que la meilleure position connue de toutes les particules de l'essaim. On note  $x_i$  le vecteur position de la  $i^{\text{ème}}$  particule de l'essaim et  $v_i$  son vecteur vitesse. Si  $D$  est la dimension du problème,  $x_i$  et  $v_i$  sont des vecteurs à  $D$  composantes dont les  $j^{\text{èmes}}$  composantes sont respectivement notées  $x_{i,j}$  et  $v_{i,j}$ . De même, on note  $p_i$  le vecteur de dimension  $D$  qui correspond à la meilleure position atteinte par la particule  $i$  et  $p_{i,j}$  sa coordonnée sur la dimension  $j$ . Enfin, on note  $g$  le vecteur de dimension  $D$  qui correspond à la meilleure position connue de l'essaim et  $g_j$  sa coordonnée sur la dimension  $j$ . Les déplacements des particules entre les itérations  $t$  et  $t + 1$  sont calculés à l'aide des équations 2.3 et 2.4.



(a) Etoile

(b) Anneau



(c) Rayon

FIGURE 2.7: Trois topologies différentes

### Algorithme

Chaque particule représente une solution potentielle dans l'espace de recherche. La nouvelle position d'une particule est déterminée en fonction de sa propre valeur et celle de ses voisins. Soit  $\vec{x}_i(t)$  la position de la particule  $P_i$  au temps  $t$ , sa position est modifiée en ajoutant une vitesse  $\vec{v}_i(t)$  à sa position courante. C'est le vecteur vitesse qui dirige le processus de recherche et reflète la "sociabilité" des particules. Si l'on considère  $N$  particules et que chaque particule compare sa nouvelle position à sa meilleure position

obtenue, cela donne l'algorithme 2,  $F$  étant la fonction de fitness[30].

**Data:**  $nb$  : *integer*  $> 0$  //Nombre de particules  
 $\vec{x}_i, \vec{v}_i$  : *Vectors*(Positions, Vélocités)  
 $pbest_i$  : *Real* (Meilleure fitness locale des particules)  
 $\vec{x}_{pbest_i}$  : *Vectors* (Meilleure position locale des particules)  
 $\rho_1$  : *Real*  
initialisation;  
**repeat**

**for**  $i=1$  to  $nb$  **do**  
| **if**  $F(\vec{x}_i) > pbest_i$  **then**  
| |  $pbest_i = F(\vec{x}_i)$   
| |  $\vec{x}_{pbest_i} = \vec{x}_i$   
| **end**  
**end**

$$\vec{v}_i(t) = \vec{v}_i(t-1) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) \quad (2.1)$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (2.2)$$

**until** *Critère d'arrêt*;

**Algorithm 2:** Algorithme de base

Ce premier algorithme ne prend pas en compte le voisinage, puisqu'on utilise uniquement l'amélioration obtenue sur la particule elle-même. En considérant un voisinage

en étoile l'algorithme 2 devient 3.

**Data:**  $nb$  : *integer*  $> 0$  //Nombre de particules

$\vec{x}_i, \vec{v}_i$  : *Vectors*(Positions, Vélocités)

$pbest_i, gbest_i$  : *Real* (Meilleure fitness locale des particules, Meilleure fitness globale des particules)

$\vec{x}_{pbest_i}, \vec{x}_{gbest_i}$  : *Vectors* (Meilleure position locale des particules, Meilleure position globale des particules)

$\rho_1, \rho_2$  : *Real*

initialisation;

**repeat**

**for**  $i=1$  to  $nb$  **do**

**if**  $F(\vec{x}_i) > pbest_i$  **then**

$pbest_i = F(\vec{x}_i)$

$\vec{x}_{pbest_i} = \vec{x}_i$

**end**

**if**  $F(\vec{x}_i) > gbest_i$  **then**

$gbest_i = F(\vec{x}_i)$

$\vec{x}_{gbest_i} = \vec{x}_i$

**end**

**end**

**for**  $i=1$  to  $nb$  **do**

$$\vec{v}_i(t) = \vec{v}_i(t-1) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest_i} - \vec{x}_i(t)) \quad (2.3)$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (2.4)$$

**end**

**until** Critère d'arrêt;

**Algorithm 3:** Algorithme avec prise en charge du voisinage

Plus une particule est éloignée de la meilleure solution globale et de sa meilleure solution, plus sera importante la variation de sa vitesse afin de faire bouger la particule vers les meilleures solutions. Les variables aléatoires  $\rho_1$  et  $\rho_2$  peuvent être définie de la façon suivante :

$$\begin{cases} \rho_1 = r_1 c_1 \\ \rho_2 = r_2 c_2 \end{cases}$$

$r_1$  et  $r_2$  suivent une loi uniforme sur  $[0..1]$  et  $c_1$  et  $c_2$  sont constantes et représentent une accélération positive, avec  $c_1 + c_2 \leq 4$ . L'algorithme s'exécute tant qu'un critère de convergence n'a pas été atteint. Cela peut être :

- Un nombre fixe d'itérations ;
- En fonction de la fitness ;
- Lorsque la variation de vitesse est proche de 0.

Cet algorithme peut être facilement modifié pour utiliser un voisinage en anneau par exemple. Les boucles devront se faire sur le voisinage et on déterminera  $\vec{x}_{lbest}$  qui remplacera  $\vec{x}_{gbest}$ .

## Paramètres de l'algorithme

Six paramètres rentrent en ligne de compte :

1. La dimension du problème ;
2. Le nombre de particules ;
3. Les valeurs des coefficients  $\rho$  ;
4. La taille du voisinage ;
5. La vitesse maximale ;
6. L'inertie.

Nous allons nous intéresser plus particulièrement aux deux derniers.

**Vitesse maximale** Pour éviter que les particules se déplacent trop rapidement d'une région à une autre dans l'espace de recherche, on fixe une vitesse maximale  $V_{max}$  [30]. Ainsi si  $v_{ij}(t)$  est la vitesse de la particule  $P_i$  au temps  $t$  dans la dimension  $j$ ,

$$v_{ij} = V_{max} \text{si } v_{ij}(t) > V_{max} \text{ et } v_{ij}(t) = -V_{max} \text{ si } v_{ij}(t) < -V_{max}$$

$V_{max}$  est généralement dépendant de l'échelle du problème.  $V_{max}$  n'est pas obligatoire (Clerc and Kennedy, 2002) si on utilise un coefficient de constriction (resserrement)  $\kappa$ .

$$\vec{v}_i(t) = \kappa(\vec{v}_i(t-1) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_i(t))) \quad (2.5)$$

avec :  $\kappa = 1 - \frac{1}{\rho} + \frac{\sqrt{|\rho^2 - 4\rho|}}{2}$  et  $\rho = \rho_1 + \rho_2 > 4$

**Facteur d'inertie** Le coefficient d'inertie  $\Phi$  joue un rôle important dans la convergence de l'algorithme. Il est utilisé pour contrôler l'impact que peut avoir la vitesse d'une particule sur le comportement de celle-ci, régulant ainsi le compromis entre exploration et exploitation. Une valeur élevée de  $w$  permet une exploration globale de

l'espace de recherche alors qu'une faible valeur favorise une recherche locale [30]. De ce fait, il peut être envisagé de commencer le processus avec une valeur élevée de  $w$ , puis de diminuer  $w$  au fil des itérations, en vue d'affiner la recherche au fur et à mesure de la convergence.

Pour contrôler l'influence de la vitesse obtenue au pas précédent on peut introduire un facteur d'inertie  $\Phi$  qui décroît en fonction du temps.

$$\vec{v}_i(t) = \Phi \vec{v}_i(t-1) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest_i} - x_i(t)) \quad (2.6)$$

Un grand facteur d'inertie provoque une grande exploration de l'espace de recherche alors qu'un petit facteur d'inertie concentre la recherche sur un petit espace. La convergence n'est pas assurée quelques soient les valeurs de  $\Phi$  et également de  $c_1$  et  $c_2$  [98], les études expérimentales montrent

$$\frac{1}{2}(c_1 + c_2) - 1 < \Phi \leq 1 \quad (2.7)$$

La diversité des approches bio-inspirées rends le choix entre elles délicat, pour cela, on a présenté les plus importantes et on a essayé de donner une idée sur les avantages et inconvénients de chaque approche. Cette étude explique notre choix pour l'optimisation par essaim de particules. Cette métaheuristique combine les avantages de plusieurs autres méthodes bio-inspirées et donne bien des meilleurs résultats avec une très bonne maniabilité et flexibilité. Après avoir donné une idée générale sur ces approches, on entame les différentes étapes du processus de la reconstruction 3D. Le chapitre suivant sera consacré à la Stéréovision, et tous ce qui est lié à la stéréoscopie.

# Chapitre 3

## Stéréovision

### Sommaire

---

<b>3.1</b>	<b>Géométrie Épipolaire . . . . .</b>	<b>32</b>
<b>3.2</b>	<b>La Matrice Fondamentale . . . . .</b>	<b>34</b>
3.2.1	Dérivation Géométrique . . . . .	35
3.2.2	Dérivation Algébrique . . . . .	36
3.2.3	Représentation algébrique de la géométrie épipolaire – La ma- trice fondamentale . . . . .	37
3.2.4	Condition de Correspondance . . . . .	39
3.2.5	Propriétés de la Matrice Fondamentale . . . . .	40
3.2.6	Quelques détails sur la matrice fondamentale . . . . .	41
3.2.7	L’Homographie de la ligne épipolaire . . . . .	42
<b>3.3</b>	<b>Géométrie épipolaire calibrée et matrice essentielle . . . . .</b>	<b>42</b>
3.3.1	La matrice essentielle . . . . .	43
3.3.2	Les propriétés de la matrice essentielle . . . . .	44
<b>3.4</b>	<b>Estimation de la géométrie épipolaire – Méthode de base . . . . .</b>	<b>45</b>
3.4.1	Combien de correspondances faut-il avoir? . . . . .	46
<b>3.5</b>	<b>Rectification des images . . . . .</b>	<b>46</b>
3.5.1	Preliminaires . . . . .	47
3.5.2	Projection de l’épipôle à l’infini . . . . .	49
3.5.3	La mise en correspondances des transformations . . . . .	50
3.5.4	Transformations Quasi affine . . . . .	53
3.5.5	Rééchantillonnage . . . . .	56
<b>3.6</b>	<b>Expérimentations . . . . .</b>	<b>57</b>

---

La stéréoscopie est un procédé de reconstruction de profondeur à partir de deux images de la même scène. Cette méthode est apparue presque en même temps que la photographie. Le procédé de la stéréoscopie est calqué sur la perception humaine du relief grâce aux deux images planes que l'on perçoit de chaque œil. Expliquons son fonctionnement : si deux images de la même scène sont acquises avec des angles différents, alors la profondeur de la scène crée une disparité géométrique entre elles. Notons  $\epsilon$  cette fonction de disparité. Alors, si l'altitude des caméras est suffisamment haute pour que le modèle de projection parallèle soit précis,  $\epsilon$  et la fonction de profondeur  $z$  sont liées par la relation :

$$z = \frac{\epsilon}{b/h}$$

où  $b/h$  est un coefficient de stéréoscopie dépendant seulement des conditions d'acquisition.  $b$  est la distance entre les centres des caméras et  $h$  est la distance entre la scène et le système de caméras. La précision  $dz$  de la mesure de profondeur est donc liée à la précision  $d\epsilon$  de la mesure de disparité :

$$dz = \frac{d\epsilon}{b/h}$$

La figure Fig 3.1 représente tous ça.

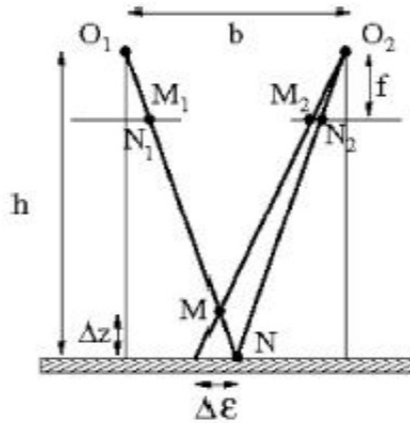


FIGURE 3.1: Principe de la stéréoscopie

Pour une précision  $d\epsilon$  donnée, plus le coefficient  $b/h$  est grand et plus l'erreur de profondeur  $dz$  est petite. Mais un grand coefficient  $b/h$  signifie aussi qu'il y a beaucoup de changements entre les deux images de la paire stéréoscopique, ce qui implique plus de difficultés dans le processus d'appariement. Il faut donc faire un compromis entre une erreur de profondeur petite et une erreur de disparité petite, c'est-à-dire un processus

d'appariement fiable. Cependant, quand les deux vues sont séparées par un angle très petit, les disparités observées sur les images sont très faibles en comparaison avec la taille d'un pixel. C'est pourquoi les méthodes d'appariement pixelliques ne sont pas en mesure de trouver une information de profondeur intéressante.

### 3.1 Géométrie Épipoilaire

La géométrie épipoilaire entre deux points de vue est essentiellement la géométrie de l'intersection des plans d'image avec le faisceau de plans ayant la ligne de base comme axe (la ligne de base est la ligne qui joint les centres des caméras). Cette géométrie est généralement motivée par considération de la recherche des points correspondants à l'appariement stéréo, et nous allons commencer à partir de cet objectif [50]. Supposons un point  $X$  dans l'espace 3D est imagée en deux points de vues, à  $x$  dans le premier, et  $x'$  dans le deuxième. Quelle est la relation entre les points correspondant  $x$  et  $x'$  de l'image ? comme illustré dans la figure 3.2, les points d'image  $x$  et  $x'$ , le point d'espace  $X$ , et les centres des caméras sont coplanaires. Notons ce plan  $\pi$ . Il est clair que les rayons rétro-projetés de  $x$  et  $x'$  sont coplanaires et s'intersectent en  $X$  et se situent dans  $\pi$ . C'est cette dernière propriété qui est la plus importante dans la recherche des correspondances.

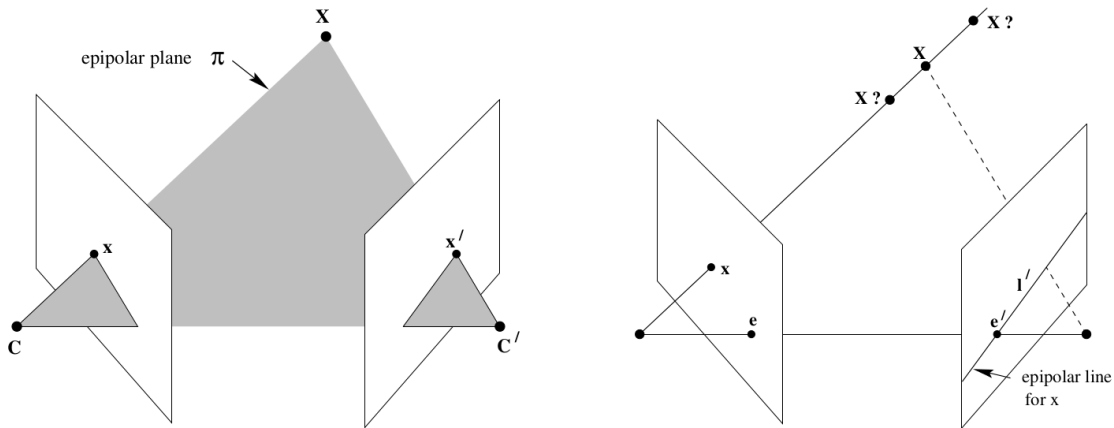


FIGURE 3.2: Correspondances des points

Supposons maintenant qu'on a seulement le point  $x$ , on peut se demander sous quelle contrainte on peut avoir le point correspondant  $x'$ . Le plan  $\pi$  est déterminée par la ligne de base et le rayon défini par  $x$ . A partir de ci-dessus, nous savons que le rayon correspondant au (inconnu) point  $x'$  réside dans  $\pi$ , donc le point  $x'$  se trouve sur la

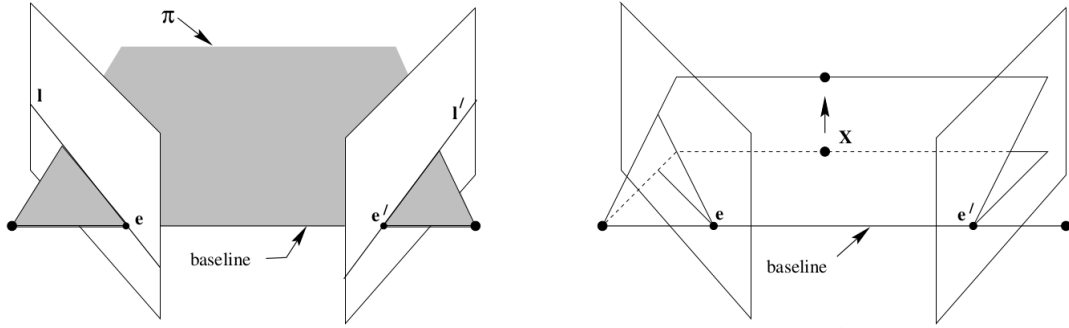


FIGURE 3.3: Entités géométriques impliqués dans la géométrie épipolaire

ligne d'intersection  $l'$  de  $\pi$  avec le deuxième plan de l'image. Cette ligne  $l'$  est l'image de la deuxième vue du rayon rétro-projetée à partir de  $x$ . C'est la ligne épipolaire correspondante à  $x$ . En termes d'un algorithme de stéréo-correspondance, le bénéfice est que la recherche pour le point correspondant à  $x$  ne nécessite pas de couvrir tout le plan d'image mais peut être restreinte à la ligne  $l'$ .

Les entités géométriques impliqués dans la géométrie épipolaire sont illustrés dans la figure 3.3.

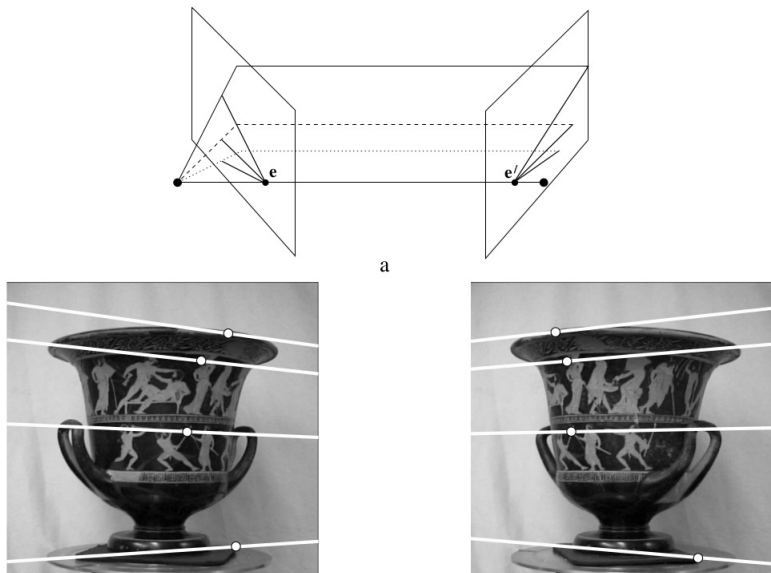


FIGURE 3.4: Exemple de la géométrie épipolaire

La terminologie est [50] :

- **L'épipôle** est le point d'intersection de la ligne joignant les centres des caméras (la ligne de base) avec le plan de l'image. De manière équivalente, l'épipôle est

l'image dans une vue du centre de la caméra de l'autre point de vue. C'est aussi le point vanishing de la direction de la ligne de base (translation).

- Un **plan épipolaire** est un plan qui contient la ligne de base. Il y a une famille d'un paramètre (un faisceau) des plans épipolaires.
- Une **ligne épipolaire** est l'intersection d'un plan épipolaire avec le plan d'image. Tous les lignes épipolaires s'intersectent au épipôle. Un plan épipolaire s'intersecte avec les plans d'image gauche et droite sur les lignes épipolaires, et définit les correspondances entre les lignes.

Des exemples de la géométrie épipolaire sont données dans la figure 3.4 et la figure 3.5. La géométrie épipolaire de ces paires d'images, et en faite tous les exemples de ce chapitre, sont calculées directement à partir des images comme décrit à la section 3.4.

## 3.2 La Matrice Fondamentale

La matrice fondamentale est la représentation algébrique de la géométrie épipolaire. Dans ce qui va suit, on dérive la matrice fondamentale à partir de la correspondance entre un point et sa ligne épipolaire, ensuite on spécifie les propriétés de la matrice[50].

Étant donné une paire d'images, on a vu dans la figure 3.2 que pour tout point  $x$  dans une image, il existe une ligne épipolaire  $l'$  correspondante dans l'autre image. Tout point  $x'$  dans la deuxième image correspondant au point  $x$  doit se trouver sur la ligne épipolaire  $l'$ . La ligne épipolaire est la projection dans la deuxième image du rayon à partir du point  $x$  par rapport au centre  $C$  de la première caméra. Ainsi, il y a une carte

$$x \mapsto l'$$

d'un point dans une image à sa ligne épipolaire correspondante dans l'autre image. C'est la nature de cette carte qui va maintenant être explorée. Il va s'avérer que cette cartographie est une corrélation (singulière), qui est une transformation projective des points vers des lignes, qui est représenté par une matrice  $F$  i.e. la matrice fondamentale.

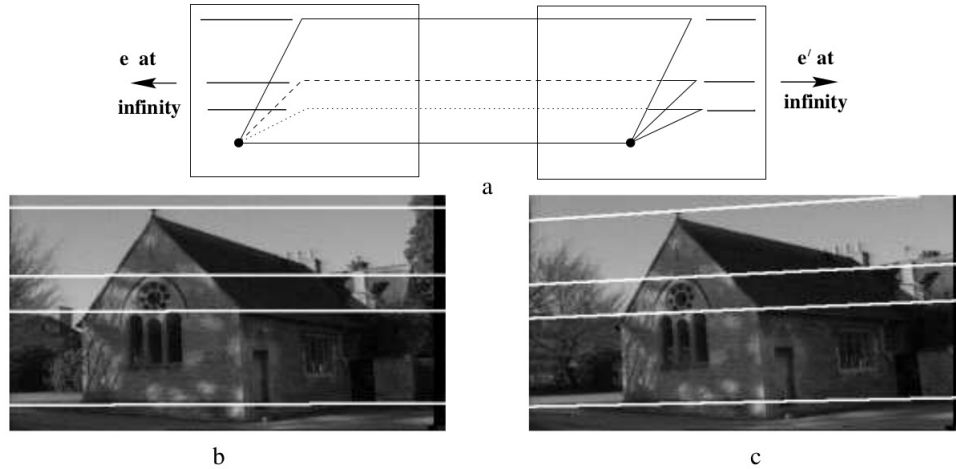


FIGURE 3.5: Mouvement parallèle au plan d'image

### 3.2.1 Dérivation Géométrique

Nous commençons par une dérivation géométrique de la matrice fondamentale. La cartographie d'un point dans une image à une ligne épipolaire correspondante dans l'autre image peut être décomposé en deux étapes. Dans la première étape, le point  $x$  est mappé à un point  $x'$  dans l'autre image se trouvant sur la ligne épipolaire  $l'$ . Ce point  $x'$  est un match potentiel pour le point  $x$ . Dans la deuxième étape, la ligne épipolaire  $l'$  est obtenue comme la ligne joignant  $x'$  à l'épipôle  $e'$ [50].

#### Étape 1 : Transfert de point via un plan

Voir la figure 3.6. Considérons un plan  $\pi$  dans l'espace ne passant pas par les deux centres de caméra. Le rayon à travers le centre la première caméra correspondant au point  $x$  rencontre le plan  $\pi$  dans un point  $X$ . Ce point  $X$  est ensuite projeté à un point  $x'$  dans la deuxième image. Cette procédure est connue en tant que transfert par l'intermédiaire du plan  $\pi$ . Tant que  $X$  est sur le rayon correspondant à  $x$ , le point projetée  $x'$  doit se situer sur la ligne épipolaire  $l'$  correspondante à l'image de ce rayon, comme illustré dans la figure 3.2. Les points  $x$  et  $x'$  sont les deux des images du point 3D  $X$  reposant sur le plan. L'ensemble de tels points  $x_i$  dans la première image et les points correspondants  $x'_i$  dans la deuxième image sont projectivement équivalent, car ils sont chacun projectivement équivalent à l'ensemble des points planes  $X_i$ . Il y a donc une homographie 2D  $H_\pi$  qui cartographie chaque  $x_i$  à  $x'_i$ .

## 9.2 The fundamental matrix F

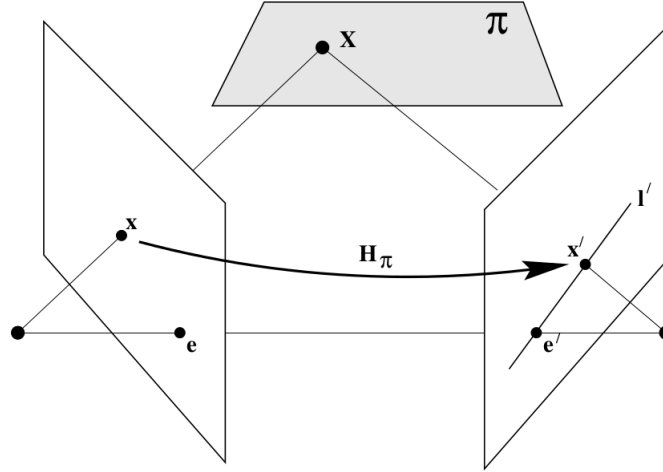


FIGURE 3.6: Transfert de point via un plan

### Étape 2 : Construction de la ligne épipolaire

Étant donné le point  $x'$ , la ligne épipolaire passant par  $x'$  et l'épipôle  $e'$  peut être écrit comme  $l' = e' \times x' = [e']_x x'$  (la notation  $[e']_x$  est défini dans [50] (eq 4.5, p 581)). Tant que  $x'$  peut être écrit comme  $x' = H_\pi x$ , nous avons

$$l' = [e']_x H_\pi X = FX$$

où on définit  $F = [e']_x H_\pi$ , la matrice fondamentale.

**Résultat** La matrice fondamentale  $F$  peut s'écrire comme suit :  $F = [e'] \times H_\pi$ , où  $H_\pi$  est le transformation projective d'une image à une autre via n'importe quel plan  $\pi$ . En outre, depuis  $[e']_x$  est de rang 2 et  $H_\pi$  de rang 3,  $F$  est une matrice de rang 2.

### 3.2.2 Dérivation Algébrique

La forme de la matrice fondamentale en termes des matrices de projection de deux caméras  $P, P'$ , peut être dérivée algébriquement. La formulation suivante est due à Xu et Zhang [102]. Le rayon rétro-projetée de  $x$  par  $P$  est obtenu en résolvant  $PX = x$ . La famille d'un seul paramètre de solutions est de la forme donnée par [50] (eq 6.13, p162)

comme :

$$X(\lambda) = P^+x + \lambda C$$

où  $P^+$  est la pseudo-inverse de  $P$ , i.e.  $PP^+ = I$ , et  $C$  son vecteur nul, à savoir le centre de la caméra, défini par  $PC = 0$ . Le rayon est paramétré par le scalaire  $\lambda$ . En particulier deux points sur le rayon sont  $P^+x$  (à  $\lambda = 0$ ), et le premier centre de caméra  $C$  (à  $\lambda = \infty$ ). Ces deux points sont imagés par la deuxième caméra  $P'$  à  $P'P^+x$  et  $P'C$  respectivement dans la deuxième vue. La ligne épipolaire est la ligne joignant ces deux points projetés, à savoir  $l' = (P'C) \times (P'P^+x)$ . Le point  $P'C$  est l'épipôle dans la deuxième image, c'est à dire la projection du premier centre de l'appareil, et peut être dénoté par  $e'$ . Ainsi,  $l' = [e'] \times (P'P^+x) = Fx$ , où  $F$  est la matrice

$$F = [e']_x P'P^+ \quad (3.1)$$

C'est essentiellement la même formule pour la matrice fondamentale que celle dérivée dans la section précédente, l'homographie  $H_\pi$  ayant la forme explicite  $H_\pi = P'P^+$  en termes des deux matrices de la caméra. Notez que cette dérivation se décompose dans le cas où le centre des deux appareil photo est le mêmes, dans ce cas,  $C$  est le centre de la caméra commun à la fois de  $P$  et  $P'$ , et ainsi de  $P'C = 0$ . Il s'ensuit que  $F$  défini dans 3.1 est la matrice zéro.

### 3.2.3 Représentation algébrique de la géométrie épipolaire – La matrice fondamentale

Soient  $P_1$  et  $P_2$  les matrices de projection des deux caméras :

$$P_1 \sim K_1 R_1 (I - t_1)$$

$$P_2 \sim K_2 R_2 (I - t_2)$$

Étant donné un point  $q_1$  dans la première image, nous voulons calculer la droite épipolaire  $l_2$  associée, dans la deuxième image. Nous procédons comme suit : d'abord, nous calculons le rayon de projection associé à  $q_1$  ; le rayon peut être représenté par deux points  $3D$  ; nous projetons ces points sur la deuxième image ; la droite  $l_2$  peut alors être donnée par le produit vectoriel des vecteurs de coordonnées des deux points image. Le choix naturel pour le premier point sur le rayon de projection est le centre

de projection de la première caméra :

$$\begin{pmatrix} t_1 \\ 1 \end{pmatrix}$$

Comme deuxième point nous choisissons de part son expression algébrique simple, le point à l'infini du rayon :

$$\begin{pmatrix} R_1^T K_1^{-1} q_1 \\ 0 \end{pmatrix}$$

(il est facile de vérifier que ce point est projeté sur  $q_1$  par  $P_1$ ). Ces deux points définissent entièrement le rayon de projection. Nous les projetons sur la deuxième image :

$$\begin{aligned} a &\sim P_2 \begin{pmatrix} t_1 \\ 1 \end{pmatrix} \sim K_2 R_2 (t_1 - t_2) \\ b &\sim P_2 \begin{pmatrix} R_1^T K_1^{-1} q_1 \\ 0 \end{pmatrix} \sim K_2 R_2 R_1^T K_1^{-1} q_1 \end{aligned}$$

La droite épipolaire est alors donnée par le produit vectoriel  $a \times b$ . En utilisant la règle  $(Mx) \times (My) \sim M^{-T}(x \times y)$ , nous obtenons :

$$\begin{aligned} l_2 &\sim a \times b \\ &\sim K_2 R_2 (t_1 - t_2) \times K_2 R_2 R_1^T K_1^{-1} q_1 \\ &\sim (K_2 R_2)^{-T} (t_1 - t_2) \times (R_1^T K_1^{-1} q_1) \end{aligned}$$

En utilisant la notation  $[\ ]_x$ , on peut finalement écrire :

$$\begin{aligned} l_2 &\sim (K_2 R_2)^{-T} [t_1 - t_2]_x (R_1^T K_1^{-1} q_1) \\ &\sim (K_2 R_2)^{-T} [t_1 - t_2]_x (R_1^T K_1^{-1}) q_1 \end{aligned} \tag{3.2}$$

L'équation 3.2 représente la transformation qui nous donne la droite épipolaire  $l_2$  associée au point  $q_1$ , à partir des coordonnées de celui-ci. La matrice qui représente cette transformation est appelée la matrice fondamentale  $F_{12}$  :

$$F_{12} \sim (K_2 R_2)^{-T} [t_1 - t_2]_x (R_1^T K_1^{-1}) \tag{3.3}$$

Elle représente entièrement la géométrie épipolaire (la transformation épipolaire

et la position des épipôles peuvent en être extraites). La matrice fondamentale est “orientée” – elle donne des droites épipolaires dans la deuxième image à partir de points dans la première image. Qu’en est-il pour le chemin inverse ? Si nous permutons les indices 1 et 2 dans l’équation 3.3, nous obtenons la matrice fondamentale qui donne des droites dans la première image, à partir de points dans la deuxième :

$$F_{21} \sim (K_1 R_1)^{-T} [t_2 - t_1]_x (R_2^T K_2^{-1})$$

Nous pouvons observer que  $F_{21}$  n’est rien d’autre que la transposée de  $F_{12}$  (éventuellement à un facteur scalaire près) :

$$F_{21} \sim F_{12}^T$$

Dans la suite, nous allons simplement écrire  $F$ , en sousentendant que la “direction” de la transformation découle du contexte. Nous pouvons maintenant exprimer la contrainte épipolaire évoquée au §3.1. Le point  $q_2$  correspondant à  $q_1$  doit se trouver sur la droite épipolaire  $l_2$ , c’est-à-dire que le produit scalaire de  $q_2$  et  $l_2$  est nul, ou bien  $q_2^T l_2 = 0$ . En remplaçant la définition de  $l_2$ , nous obtenons finalement la contrainte épipolaire :

$$q_2^T F q_1 = 0 \tag{3.4}$$

### 3.2.4 Condition de Correspondance

Jusqu’à ce point, on a considéré la carte  $x \mapsto l'$  défini par  $F$ . On peut maintenant souligner les propriétés les plus basiques de la matrice fondamentale.

**Résultat** La matrice fondamentale satisfait la condition suivante : pour tout couple de points correspondent  $x \leftrightarrow x'$  dans deux images, on a :

$$x'^T F x = 0$$

C’est vrai, parce que si les points  $x$  et  $x'$  correspondent, alors  $x'$  se trouve sur la ligne épipolaire  $l' = Fx$  correspondante au point  $x$ . En d’autres termes  $0 = x'^T l' = x'^T Fx$ . Inversement, si les points d’image satisfont la relation  $x'^T Fx = 0$  alors les rayons définis par ces points sont coplanaires. C’est une condition nécessaire pour que les points correspondent.

L'importance de la relation du résultat de 3.2.4, c'est qu'il donne une façon de caractériser la matrice fondamentale sans référence aux matrices de la caméra, c'est à dire seulement en termes de points d'image correspondants. Cela permet à  $F$  d'être calculé seulement à partir des correspondances de l'image. On a vu à partir de 3.1 que  $F$  peut être calculé à partir des deux matrices des appareils photo  $P, P'$ , et en particulier que  $F$  est déterminée de manière unique par les caméras, jusqu'à une mise à l'échelle d'ensemble. Cependant, nous pouvons maintenant se demander combien de correspondances sont nécessaires au calcul de  $F$  à partir de  $x'^T Fx = 0$ , et les circonstances sous lesquelles la matrice est spécifiquement défini par ces correspondances. Ces détails sont reportées jusqu'à la section 3.4, où on voit qu'en général au moins 7 correspondances sont nécessaire pour calculer  $F$ .

### 3.2.5 Propriétés de la Matrice Fondamentale

**Définition** Supposons que nous avons deux images acquises par des caméras avec des centres non-coïncident, alors **la matrice fondamentale**  $F$  est une matrice homogène unique de taille  $3 \times 3$  de rang 2 qui satisfait

$$x'^T Fx = 0 \tag{3.5}$$

pour tous les points correspondants  $x \leftrightarrow x'$ . Maintenant Nous listons brièvement un certain nombre de propriétés de la matrice fondamentale. les plus importantes des propriétés sont également résumées dans le tableau 3.1.

1. **Transposé** : Si  $F$  est la matrice fondamentale de la paire de caméras  $(P, P')$ , alors  $F^T$  est la matrice fondamentale de la paire dans l'ordre inverse :  $(P', P)$ .
2. **Les lignes épipolaires** : Pour tout point  $x$  dans la première image, la ligne épipolaire correspondante est  $l' = Fx$ . De même,  $l = F^T x'$  représente la ligne épipolaire correspondante à  $x'$  dans la deuxième image.
3. **L'épipôle** : Pour tout point  $x$  (autre que  $e$ ) la ligne épipolaire  $l' = Fx$  contient l'épipôle  $e'$ . Ainsi  $e'$  satisfait  $e'^T(Fx) = (e'^T F)x = 0$  pour tout  $x$ . Il s'ensuit que  $e'^T F = 0$ , i.e.  $e'$  est le vecteur nul gauche de  $F$ . De même  $Fe = 0$ , i.e.  $e$  est le vecteur droit nul de  $F$ .
4.  **$F$  a sept degrés de libertés** : une matrice homogène de taille  $3 \times 3$  a huit ratios indépendant (il y a neuf éléments, et de la mise à l'échelle commune n'est pas significatif) ; Toutefois,  $F$  satisfait également à la contrainte  $\det F = 0$  qui élimine un degré de liberté.

TABLE 3.1: Résumé des propriétés de la matrice fondamentale

<ul style="list-style-type: none"> <li>— <math>F</math> est une matrice homogène de rang 2 avec 7 degrés de libertés.</li> <li>— <b>Correspondance des points</b> : Si <math>x</math> et <math>x'</math> sont des points de l'image qui correspondent, alors <math>x'^T F x = 0</math>.</li> <li>— <b>Les lignes épipolaires</b> : <ul style="list-style-type: none"> <li><math>l' = F x</math> est la ligne épipolaire correspondante à <math>x</math>.</li> <li><math>l = F^T x'</math> est la ligne épipolaire correspondante à <math>x'</math>.</li> </ul> </li> <li>— <b>Les épipôles</b> : <ul style="list-style-type: none"> <li><math>F e = 0</math>.</li> <li><math>F^T e' = 0</math>.</li> </ul> </li> <li>— <b>Calcul à partir des matrices des caméras <math>P, P'</math></b> : <ul style="list-style-type: none"> <li>Caméras générales, <math>F = [e']_x P' P^+</math>, où <math>P^+</math> est la pseudo-inverse de <math>P</math>, et <math>e' = P' C</math>, avec <math>P C = 0</math>.</li> <li>Caméras canoniques, <math>P = [I 0], P' = [M m], F = [e']_x M = M^{-T} [e]_x</math>, où <math>e' = m</math> et <math>e = M^{-1} m</math>.</li> <li>Caméras pas à l'infini <math>P = K[I 0], P' = K'[R t], F = K'^{-T} [t]_x R K^{-1} = [K't]_x K' R K^{-1} = K'^{-T} R K^T [K R^T t]_x</math>.</li> </ul> </li> </ul>
---

5.  **$F$  est une corrélation** : une transformation projective d'un point à une ligne. Dans ce cas, un point  $x$  dans la première image définit une ligne dans la deuxième  $l' = F x$ , qui est la ligne épipolaire de  $x$ . Si  $l$  et  $l'$  sont les lignes épipolaires correspondants alors n'importe quel point  $x$  sur  $l$  est projeté sur la même ligne  $l'$ . Ça veut dire qu'il n'y a pas de transformation inverse, et que  $F$  n'est pas full-rank. Pour cette raison,  $F$  n'est pas une corrélation propre (qui va être inversible).

### 3.2.6 Quelques détails sur la matrice fondamentale

La matrice fondamentale est une transformation entre points et droites, ce qui est souvent appelée une corrélation. La propriété algébrique principale de la matrice fondamentale est qu'elle est singulière, ce qui est équivalent au fait que son déterminant est nul ou que son rang est inférieur à 3. Il y a plusieurs manières de constater ceci – nous en expliquons deux. Premièrement, la matrice  $[t_1 - t_2]_x$  dans l'équation 3.2 est singulière (toute matrice anti-symétrique l'est) et par conséquent, la matrice fondamentale l'est aussi (le déterminant d'un produit de matrices carrées équivaut au produit des déterminants). Le deuxième argument découle de l'observation de la figure 3.4. La matrice fondamentale est une fonction entre deux espaces bi-dimensionnels – l'espace des points dans la première image et l'espace des droites dans la deuxième. Pourtant, seul

le faisceau de droites épipolaires – un espace uni-dimensionnel – est atteint. La fonction n’est pas bijective ce qui se manifeste par la singularité de la matrice fondamentale. Une matrice singulière a un noyau, c’est-à-dire qu’il existe des vecteurs non-nuls pour qui le produit matrice-vecteur donne le vecteur nul. Le noyau de la matrice fondamentale n’est rien d’autre que l’épipôle  $e_1$ , i.e. nous avons :

$$F e_1 = 0$$

Ceci peut être prouvé de manière analytique, mais il y a aussi une explication intuitive : la droite épipolaire associée à l’épipôle n’est pas définie, ce qui correspond bien au fait que le vecteur nul  $0 = F e_1$  n’est pas admis pour représenter des points ou des droites en coordonnées homogènes. Pour le deuxième épipôle il existe une relation analogue à celle donnée ci-dessus :

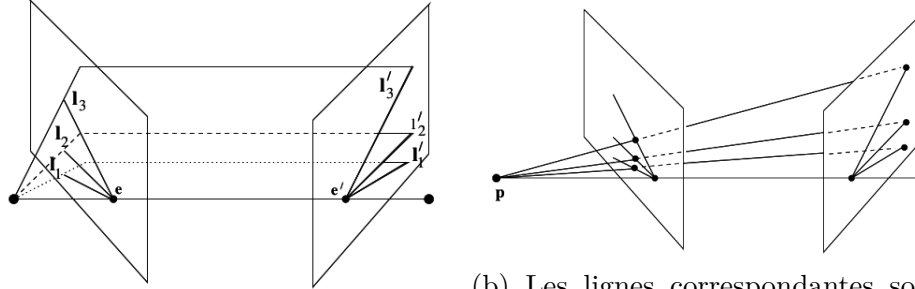
$$F^T e_2 = 0$$

### 3.2.7 L’Homographie de la ligne épipolaire

L’ensemble de lignes épipolaires dans chacune des images forme un faisceau de droites passant par l’épipôle. Un tel faisceau de lignes peut être considéré comme un espace projectif de dimension 1. Il est clair de la figure 3.7 que les lignes épipolaires correspondantes sont liés en perspective, de sorte qu’il existe une homographie entre le faisceau de droites épipolaires centrée à  $e$  dans la première vue, et le faisceau centrée à  $e'$  dans la seconde. Une homographie entre deux espaces projectifs 1-Dimensionnel a trois degrés de liberté. Les degrés de liberté de la matrice fondamentale peuvent donc être comptabilisés comme suit : 2 pour  $e$ , 2 pour  $e'$ , et 3 pour l’homographie de la ligne épipolaire qui mappe une ligne par  $e$  à une ligne par  $e'$ . Ici, on donne une formule explicite pour cette cartographie.

## 3.3 Géométrie épipolaire calibrée et matrice essentielle

La matrice fondamentale dépend du positionnement des deux caméras ainsi que de leurs paramètres intrinsèques, c’est-à-dire de leurs matrices de calibrage  $K_1$  et  $K_2$ . Dans cette section, nous supposons que les caméras sont calibrées[93], c’est-à-dire que nous connaissons  $K_1$  et  $K_2$ . Nous pouvons donc, à partir de la matrice fondamentale,



(a) Il s'agit d'un faisceau de droites épipolaires dans chaque image centrée sur l'épipôle. La correspondance entre les lignes épipolaires,  $l_i \leftrightarrow l'_i$ , est définie par le faisceau de plans avec la ligne de base comme axe.

(b) Les lignes correspondantes sont liés par une perspective avec le centre n'importe quel point  $p$  sur la ligne de base. Il s'ensuit que la correspondance entre les lignes épipolaires dans les faisceaux est une homographie de dimension 1.

FIGURE 3.7: Homographie d'une ligne épipolaire

calculer la matrice essentielle  $E$  :

$$E \sim K_2^T F K_1 \sim R_2 [t_1 - t_2]_x R_1^T \quad (3.6)$$

**Que représente la matrice essentielle ?** En effet, la matrice fondamentale, elle, représente la géométrie épipolaire de deux images si les repères utilisés pour les représenter sont les repères pixels. La matrice essentielle, quant à elle, représente également la géométrie épipolaire, mais cette fois-ci exprimée par rapport aux repères image. On dit souvent que la matrice essentielle représente la géométrie épipolaire calibrée, puisque c'est l'information sur le calibrage des caméras qui permet de remonter aux repères image. On constate que la matrice essentielle ne dépend que de la position et de l'orientation des caméras, d'où son utilité pour l'estimation du mouvement d'une caméra, respectivement du placement relatif de deux caméras.

### 3.3.1 La matrice essentielle

La matrice essentielle est la spécialisation de la matrice fondamentale pour le cas de coordonnées d'image normalisés (voir ci-dessous). Historiquement, la matrice essentielle était introduite (par Longuet-Higgins [68]) avant la matrice fondamentale, et la matrice fondamentale peut être considéré comme la généralisation de la matrice essentielle dans lequel (inessentiel) l'hypothèse de caméras calibrées est retirée[50]. La matrice essentielle a moins de degrés de libertés, et des propriétés supplémentaires, par

rapport à la matrice fondamentale. Ces propriétés sont décrites ci-dessous.

**Coordonnées normalisés** Considérons une matrice de la caméra décomposée en  $P = K[R|t]$ , et  $x = PX$  un point de l'image. Si la matrice d'étalonnage  $K$  est connue, alors on peut appliquer son inverse pour avoir le point  $\hat{x} = K^{-1}x$ . Alors  $\hat{x} = [R|t]X$ , où  $\hat{x}$  est le point d'image exprimé en coordonnées normalisées. Il peut être considéré comme l'image du point  $X$  par rapport à un appareil photo  $[R|t]$  ayant la matrice d'identité  $I$  comme matrice d'étalonnage. La matrice de la caméra  $K^{-1}P = [R|t]$  est appelé une matrice normalisée de caméra, avec la supposition que la matrice de calibrage n'est pas connue. Maintenant, envisager une paire de matrices normalisée de caméra  $P = [I|0]$  et  $P' = [R|t]$ . La matrice fondamentale correspondante à la paire normalisés de caméras est habituellement appelée matrice essentielle, et selon [50] (eq 9,2 -p 244), elle a la forme :

$$E = [t]_x R = R[R^T t]_x$$

**Définition** L'équation qui définit la matrice essentielle est la suivante :

$$\hat{x}'^T E \hat{x} = 0 \tag{3.7}$$

en termes des coordonnées normalisée de l'image pour les points correspondants  $x \leftrightarrow x'$ . La substitution de  $\hat{x}$  et  $\hat{x}'$  donne  $x'^T K'^{-T} E K^{-1} x = 0$ . En comparant ceci avec le relation  $x'^T F x = 0$  pour la matrice fondamentale, il s'ensuit que la relation entre les matrices fondamentales et essentielles est

$$E = K'^T F K \tag{3.8}$$

### 3.3.2 Les propriétés de la matrice essentielle

La matrice essentielle,  $E = [t]_x R$ , n'a que cinq degrés de liberté : les deux matrices de rotation et translation  $R$  et  $t$  ont trois degrés de libertés, mais il y a une échelle d'ambiguïté globale - comme la matrice fondamentale, la matrice essentielle est une quantité homogène. Le nombre réduit de degrés de libertés se traduit par des contraintes supplémentaires qui sont satisfaites par une matrice essentielle, par rapport à la matrice fondamentale.

### 3.4 Estimation de la géométrie épipolaire – Méthode de base

Dans la section 3.2.3, nous avons vu comment calculer la matrice fondamentale, donc la représentation algébrique de la géométrie épipolaire, à partir du positionnement et des paramètres intrinsèques de deux caméras en question. Un des usages de la géométrie épipolaire, discuté précédemment, consiste à contraindre la mise en correspondance des images. Réciproquement, la connaissance de suffisamment de correspondances permet d'estimer la géométrie épipolaire. Ceci est très important en pratique, où l'on ne connaît souvent pas le positionnement des caméras[93]. Dans cette section, nous traitons la méthode de base pour le calcul de la matrice fondamentale, à partir de correspondances de points. La méthode de base s'appuie sur l'équation 3.4 :

$$q_2^T F q_1 = 0$$

En explicitant les coefficients de la matrice fondamentale  $F$ , ceci s'écrit comme :

$$\begin{aligned} F_{11}q_{1,1}q_{2,1} + F_{12}q_{1,2}q_{2,1} + F_{13}q_{1,3}q_{2,1} + \\ F_{21}q_{1,1}q_{2,2} + F_{22}q_{1,1}q_{2,2} + F_{23}q_{1,3}q_{2,2} + \\ F_{31}q_{1,1}q_{2,3} + F_{32}q_{1,2}q_{2,3} + F_{33}q_{1,3}q_{2,3} = 0 \end{aligned}$$

Cette équation est linéaire en coefficients de  $F$ . Si l'on prend en compte  $n$  correspondances de points, chacune donne une équation du type ci-dessus. Toutes ces équations peuvent être regroupées en un système matriciel :

$$A f = 0 \tag{3.9}$$

avec un vecteur  $f$  contenant les coefficients de  $F$  (les inconnues) :

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^T$$

et la matrice  $A$  de dimension  $n \times 9$  qui est de la forme :

$$A = \begin{bmatrix} q_{1,1}q_{2,1} & q_{1,2}q_{2,1} & q_{1,3}q_{2,1} & q_{1,1}q_{2,2} & q_{1,1}q_{2,2} & q_{1,3}q_{2,2} & q_{1,1}q_{2,3} & q_{1,2}q_{2,3} & q_{1,3}q_{2,3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{n \times 9}$$

A cause du bruit dans les données, il n'y a en général pas de solution exacte pour l'équation 3.9, donc on résout le système aux moindres carrés, c'est-à-dire on détermine  $\hat{f}$  tel que :

$$\hat{f} = \arg \min_f \| Af \|^2 \text{ sous la contrainte } \| f \| = 1$$

### 3.4.1 Combien de correspondances faut-il avoir ?

L'équation 3.9 est un système d'équations linéaire et homogène (sur le côté droit se trouve un vecteur nul). De manière générale, en présence de  $m$  inconnues, il y a une solution unique (à l'échelle près) si l'on dispose de  $m - 1$  équations. Avec moins d'équations, le système est sous-contraint. Si plus d'équations sont disponibles, il n'y a en général, en présence de bruit dans les données, pas de solution exacte, c'est pourquoi la méthode des moindres carrés est appliquée[93]. Dans notre cas alors, chaque correspondance de points fournit une seule équation, donc il nous faut a priori un minimum de 8 correspondances pour estimer la géométrie épipolaire. C'est la raison pourquoi cet algorithme est typiquement appelée la méthode des 8 points (8 point method) dans la littérature.

**Remarque.** Grâce au fait que la matrice fondamentale doit être singulière, nous pouvons en fait l'estimer avec seulement 7 correspondances de points. Avec les 7 équations linéaires et homogènes sur les 9 inconnues (qui sont définies à un facteur multiplicatif près), il existe une famille de solutions pour la matrice fondamentale qui peut être exprimée par la combinaison linéaire de deux matrices de base :

$$F = F_1 + \lambda F_2$$

La matrice  $F$  étant singulière se traduit par le fait que son déterminant est nul. Le déterminant de la matrice ci-dessus, est un polynôme cubique en  $\lambda$ . Donc, au lieu d'avoir une infinité de solutions, il n'en restent plus que 3 (correspondant aux 3 racines du polynôme).

## 3.5 Rectification des images

Dans cette partie du travail, on va utiliser la méthode de rectification présenté par Hartley dans [54].

### 3.5.1 Préliminaires

Les vecteurs colonnes seront dénotés par des lettres minuscules, comme  $x$ . Les vecteurs lignes sont des vecteurs colonnes transposés, tels que  $x^T$ . Ainsi, le produit scalaire de deux vecteurs est représenté par  $a^T b$ . En d'autre part,  $ab^T$  est une matrice de rang 1. Les matrices seront désignés par des lettres majuscules. Le signe  $\cong$  est utilisée pour indiquer l'égalité des vecteurs ou des matrices jusqu'à une multiplication par facteur d'échelle non nul.

Si  $A$  est une matrice carrée alors sa matrice des cofacteurs est notée  $A^*$ . Les identités suivantes sont bien connus :  $A^* A = A A^* = \det(A) I$  où  $I$  est la matrice d'identité. En particulier, si  $A$  est une matrice inversible, alors  $A^* \approx (A^T)^{-1}$ .

Ayant un vecteur,  $t = (t_x, t_y, t_z)^T$ , on introduit la matrice antisymétrique

$$[t]_x \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (3.10)$$

Pour tout vecteur non nul  $t$ , la matrice  $[t]_x$  est de rang 2. En outre, l'espace nul de  $[t]_x$  est généré par le vecteur  $t$ . Cela signifie que  $t^T [t]_x = [t]_x t = 0$  et que tout autre vecteur éliminé par  $[t]_x$  est scalaire multiple de  $t$ .

La matrice  $[t]_x$  est étroitement liée au produit vectoriel et ce, pour tous vecteurs  $s$  et  $t$ , on a  $s^T [t]_x = s \times t$  et  $[t]_x s = t \times s$ . Une propriété utile des produits vectoriels peut être exprimée en termes de la matrice  $[t]_x$ .

**proposition** Pour toute matrice  $M$  de taille  $3 \times 3$  et vecteur  $t$ , on a :

$$M^* [t]_x = [Mt]_x M \quad (3.11)$$

*La géométrie projective.* Le  $n$ -espace projectif réel se compose de l'ensemble des classes d'équivalence de  $(n+1)$  vecteurs réel non-nuls, où deux vecteurs sont considérés comme équivalents, s'ils se diffèrent par un facteur constant. De cette manière, un vecteur représentant un point dans  $P^n$  est connu comme une représentation homogène du point.

Le  $n$ -espace projectif réel contient un  $n$ -espace euclidien comme l'ensemble de tous

les vecteurs homogènes dont la coordonnée finale n'est pas égal à zéro. Par exemple, un point  $P^2$  est représenté par un vecteur  $u = (u, v, w)^T$ . Si  $w \neq 0$ , alors cela représente le point dans  $R^2$  exprimé en coordonnées euclidiennes comme suit :  $(u/w, v/w)^T$ .

Les lignes dans  $P^2$  sont également représentés en coordonnées homogènes. En particulier, la ligne  $\lambda$  avec les coordonnées  $(\lambda, \mu, \nu)^T$  est la ligne formée de points qui satisfont l'équation  $\lambda u + \mu v + \nu w = 0$ . En d'autres termes, un point  $u$  se trouve sur une ligne  $\lambda$  si et seulement si  $\lambda^T u = 0$ . La ligne qui lie les deux points  $u_1$  et  $u_2$  est donnée par le produit vectoriel  $u_1 \times u_2$ .

Une application projective (ou transformation) à partir de  $P^n$  vers  $P^m$  est une projection représentée par une transformation linéaire des coordonnées homogènes. Les applications projectives peuvent être représentés par des matrices de dimension  $(m + 1) \times (n + 1)$ . Le mot projectivité sera également utilisé pour désigner une application projective inversible.

Si  $A$  est une matrice non singulière de taille  $3 \times 3$  qui représente une transformation projective de  $P^2$ , alors  $A^*$  est la projection de la ligne correspondante. En d'autres termes, si  $u_1$  et  $u_2$  se trouvent sur une ligne  $\lambda$ , alors  $Au_1$  et  $Au_2$  se trouvent sur la ligne  $A^*\lambda$  :

$$A^*(u_1 \times u_2) = (Au_1) \times (Au_2)$$

Cette formule peut être déduit facilement de la proposition 3.5.1.

La plupart des vecteurs et matrices utilisées dans cette section seront définis seulement jusqu'à la multiplication par un facteur non nul. Normalement, nous allons ignorer les facteurs de multiplications et nous allons utiliser le signe d'égalité ( $=$ ) pour désigner l'égalité jusqu'à un facteur constant. Les exceptions à cette règle seront notés spécifiquement.

*Modèle de la caméra.* Le modèle de l'appareil photo considéré dans cette section est celui de la projection centrale, autrement connu comme le modèle sténopé ou perspective. Une telle caméra projette une région de  $R^3$  qui se trouve en face d'elle dans une région du plan d'image  $R^2$ . Pour plus de commodité mathématique, nous étendons cette projection à une correspondance entre les espaces projectifs  $P^3$  (l'espace d'objet) et  $P^2$  (l'espace d'image). La projection est définie partout dans  $P^3$ , sauf au centre de projection de la caméra (ou le centre de la caméra).

les points dans l'espace d'objet seront donc désignés par quatre vecteurs homogènes  $x = (x, y, z, t)^T$ , ou plus généralement par  $(x, y, z, 1)^T$ . Les points de l'espace

d'image seront représentés par  $u = (u, v, w)^T$ .

La projection de l'espace objet vers l'espace d'image est une application projective représenté par une matrice  $P$  de taille  $3 \times 4$  de rang 3, connue comme la matrice de la caméra (paramètres intrinsèques). La matrice de la caméra transforme les points d'objet dans l'espace à 3 dimensions à des points dans l'espace de l'image en 2 dimensions selon l'équation  $u = Px$ . La matrice de la caméra  $P$  est définie à un facteur d'échelle seulement, et a donc 11 entrées indépendantes. Ce modèle permet la modélisation de plusieurs paramètres, notamment : la position et l'orientation de la caméra ; les principaux décalages des points dans l'espace image ; et des facteurs d'échelle inégales dans deux directions orthogonales, celles-ci ne sont pas nécessairement parallèles aux axes dans l'espace d'image.

Supposons que le centre de la caméra n'est pas à l'infini, et que ces coordonnées euclidiennes soit  $t = (t_x, t_y, t_z)^T$ . La projection de la caméra n'est pas défini à  $t$  donc  $P(t_x, t_y, t_z, 1)^T = 0$ . Si  $P$  est écrit sous forme de bloc en tant que  $P = (M|v)$ , alors  $Mt + v = 0$ , et ainsi  $v = -MT$ . Donc, la matrice de la caméra peut être écrite sous la forme :

$$P = (M | -Mt)$$

où  $t$  est le centre de la caméra. Étant donné que  $P$  est de rang 3, Donc  $M$  est inversible (non-singulière).

### 3.5.2 Projection de l'épipôle à l'infini

Dans cette section, on va discuter de la question de trouver une transformation projective  $H$  d'une image projetant un épipôle à un point à l'infini. En fait, si les lignes épipolaires vont être transformées en lignes parallèles à l'axe  $x$ , alors l'épipôle doit être projeté au point de l'infinie  $(1, 0, 0)^T$ . Cela laisse beaucoup de degrés de liberté ouverts pour  $H$  (en fait quatre), et si une  $H$  inappropriée est choisie, des déformations projective sévères de l'image peuvent avoir lieu. Pour que l'image rééchantillonnée ressemble un peu à l'image d'origine, on peut imposer des restrictions plus étroites sur le choix de  $H$ .

Une condition qui mène à de bons résultats est d'insister sur le fait que la transformation  $H$  doit agir autant que possible comme une transformation rigide dans le voisinage d'un certain point sélectionné  $u_0$  de l'image. On entend par là, qu'au premier ordre le voisinage de  $u_0$  peut seulement subir une rotation et une translation, et donc

aura le même aspect dans les deux image originale et rééchantillonnée. Un choix approprié du point  $u_0$  peut être le centre de l'image. Par exemple, ce serait un bon choix dans le contexte de la photographie aérienne si on sait que la vue n'est pas trop oblique.

Pour l'instant, supposons  $u_0$  est l'origine et l'épipôle  $p = (f, 0, 1)$  se trouve sur l'axe  $x$ . Considérons maintenant la transformation suivante :

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \quad (3.12)$$

Cette transformation projette l'épipôle  $(f, 0, 1)^T$  vers le point à l'infini  $(f, 0, 0)^T$  selon les besoins. Un point  $(u, v, 1)^T$  est projeté par  $G$  au point  $(\hat{u}, \hat{v}, 1)^T = (u, v, 1 - u/f)^T$ . Si  $|u/f| < 1$ , alors on peut écrire

$$\begin{aligned} (\hat{u}, \hat{v}, 1)^T &= (u, v, 1 - u/f) \\ &= (u(1 + u/f + \dots), \\ &\quad v(1 + u/f + \dots), 1)^T \end{aligned}$$

Le Jacobien est :

$$\frac{\delta(\hat{u}, \hat{v})}{\delta(u, v)} = \begin{bmatrix} 1 + 2u/f & 0 \\ v/f & 1 + u/f \end{bmatrix}$$

Maintenant, si  $u = v = 0$  alors c'est la projection d'identité. En d'autres termes,  $G$  est approximée (au premier ordre) à l'origine par la projection d'identité.

Pour un point d'intérêt arbitrairement placé  $u_0$  et l'épipôle  $p$ , la projection nécessaire  $H$  est un produit  $H = GRT$  où  $T$  est une translation qui prend le point  $u_0$  à l'origine,  $R$  est une rotation autour de l'origine prendre l'épipôle  $p'$  vers un point  $(f, 0, 1)^T$  sur l'axe  $x$ , et  $G$  est la projection simplement envisagée pour prendre  $(f, 0, 1)^T$  à l'infini. La projection composite est au premier ordre une transformation rigide dans le voisinage de  $u_0$ .

### 3.5.3 La mise en correspondances des transformations

Nous considérons deux images  $I$  et  $I'$ . L'intention est de rééchantillonner les deux images en fonction des transformations  $H$  à appliquer à  $I$  et  $H'$  à appliquer à  $I'$ . Le rééchantillonnage doit être fait d'une manière telle qu'une ligne épipolaire dans  $I$  est

assortie avec sa ligne épipolaire correspondante dans  $I'$ . Plus précisément, si  $\lambda$  et  $\lambda'$  sont n'importe quelle paire de lignes épipolaires correspondantes dans les deux images, alors  $H^*\lambda = H'^*\lambda'$ . (Rappelons que  $H^*$  est la projection ligne correspondante à la projection point  $H$ .) Toute paire de transformations qui remplit cette condition sera appelée une paire correspondante de transformations.

La stratégie dans le choix d'une paire correspondante de transformations est de choisir premièrement  $H'$  à être une transformation qui envoie l'épipôle  $p'$  à l'infini, comme décrit dans la section précédente. On cherche alors une transformation correspondante  $H$  choisie d'une manière à minimiser la distance des sommes des carrés :

$$\sum_i d(Hu_i, H'u'_i)^2 \quad (3.13)$$

La première question est de savoir comment trouver une transformation correspondant  $H'$ . Cette question est répondue par le théorème suivant.

**Théorème 3.5.1** *Soit  $I$  et  $I'$  des images avec une matrice fondamentale  $F = [p']_x M$ , et soit  $H'$  une transformation projective de  $I'$ . Une transformation projective  $H$  de  $I$  correspond à  $H'$  si et seulement si  $H$  est de la forme :*

$$H = (I + H'p'a^T)H'M \quad (3.14)$$

pour certain vecteur  $a$ .

**Preuve.** Si  $u$  est un point dans  $I$ , alors  $p \times u$  est la ligne épipolaire dans la première image, et  $Fu$  est la ligne épipolaire dans la deuxième image. Les transformations  $H$  et  $H'$  sont une paire correspondante si et seulement si  $H^*(p \times u) = H'^*Fu$ . Comme ceci doit être maintenu pour tous les  $u$ , on peut écrire d'une façon équivalente  $H^*[p]_x = H'^*F = H'^*[p']_x M$ , ou, en appliquant la proposition 3.5.1,

$$[Hp]_x H = [H'p']_x H'M \quad (3.15)$$

qui est une condition nécessaire et suffisante pour que  $H$  et  $H'$  peuvent corres-

pondre. Pour prouver l'inverse, si 3.14 est valide, alors :

$$\begin{aligned}
Hp &= (I + H'p'a^T)H'Mp \\
&= (I + H'p'a^T)H'p' \\
&= (I + a^T H'p')H'p' \\
&\approx H'p'
\end{aligned}$$

■

On est particulièrement intéressé par le cas où  $H'$  est une transformation qui prends l'épipôle  $p'$  à un point à l'infini  $(1, 0, 0)^T$ . Dans ce cas,  $I + H'p'a^T = I + (1, 0, 0)^T a^T$  est de la forme :

$$A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

ce qui représente une transformation affine. Ainsi, un cas particulier du théorème 3.5.1 est.

**Corollaire** Soit  $I$  et  $I'$  deux images avec une matrice fondamentale  $F = [p']_x M$ , et soit  $H'$  une transformation projective de  $I'$  projetant l'épipôle  $p'$  vers le point à l'infini  $(1, 0, 0)^T$ . Une transformation  $H$  de  $I$  correspond  $H'$  si et seulement si  $H$  est de la forme  $H = AH_0$ , où  $H_0 = H'M$  et  $A$  est une transformation affine de la forme 3.16.

Ayant  $H'$  qui projette l'épipôle à l'infini, on peut utiliser ce corollaire pour faire le choix d'une transformation correspondante  $H$  pour minimiser la disparité. Ayant  $\hat{u}'_i = H'u'_i$  et  $\hat{u}_i = H_0u_i$ , le problème de minimisation 3.13 est de trouver  $A$  de la forme 3.16 de telle sorte que :

$$\sum_i d(A\hat{u}_i, \hat{u}'_i)^2 \quad (3.17)$$

est minimisé.

En particulier, soit  $\hat{u}_i = (\hat{u}_i, \hat{v}_i, 1)$ , et soit  $\hat{u}'_i = (\hat{u}'_i, \hat{v}'_i, 1)$ . Étant donné que  $H'$  et  $M$  sont connus, ces vecteurs peuvent être calculés à partir des points correspondants  $u'_i \leftrightarrow u_i$ . Ensuite, la quantité à minimiser 3.17 peut être écrite :

$$\sum_i (a\hat{u}_i + b\hat{v}_i + c - \hat{u}'_i)^2 + (\hat{v}_i - \hat{v}'_i)^2$$

Étant donné que  $(\hat{v}_i - \hat{v}'_i)^2$  est une constante, ce qui est équivalent à minimiser :

$$\sum_i (a\hat{u}_i + b\hat{v}_i + c - \hat{u}'_i)^2$$

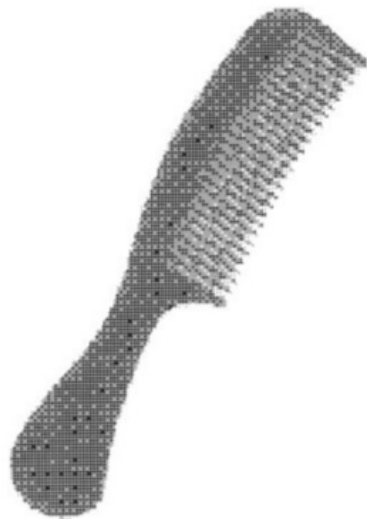
C'est un problème simple de minimisation linéaire des moindres carrés et est facilement résolu en utilisant des techniques linéaires connus pour trouver  $a, b$  et  $c$ . Alors  $A$  est calculée à partir de 3.16 et  $H$  à partir de 3.14. A noter qu'une solution linéaire est possible parce que  $A$  est une transformation affine. Si c'était tout simplement une transformation projective, ce ne serait pas un problème linéaire.

### 3.5.4 Transformations Quasi affine

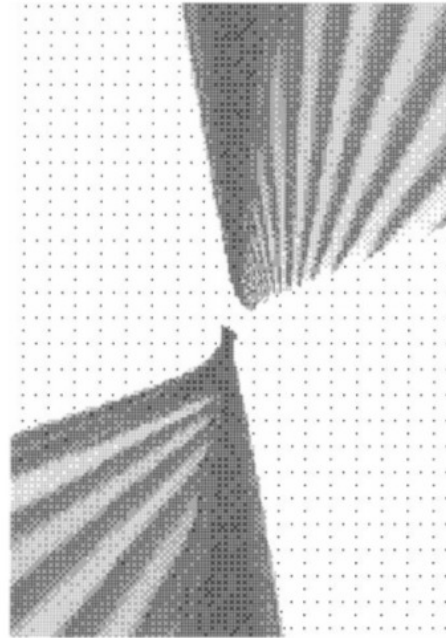
Dans le rééchantillonnage d'une image par l'intermédiaire d'une transformation projective  $2D$ , il est possible de diviser l'image de sorte que les régions connectées ne le seront plus. Une telle transformation projective est appelé projectivité non-quasi-affine. (Une définition plus précise est donnée ci-dessous.) Un exemple est donné dans la figure 3.8 qui correspond à une image d'un peigne et l'image rééchantillonnée selon une projectivité non-quasi-affine. Le rééchantillonnage d'une image via une telle projectivité est évidemment pas souhaitable. On va donc examiner des méthodes pour éviter tels cas.

Dans des images réelles, l'ensemble du plan de l'image n'est pas visible, mais en général seulement une région rectangulaire de celui-ci. En conséquence, on introduit le concept d'une fenêtre de vue. La fenêtre de vue est la partie du plan d'image qui contient toutes les images disponibles, y compris les points appariés (mais pas nécessairement l'épipôle). Dans le rééchantillonnage d'une image, seulement les points dans la fenêtre de vue seront rééchantillonnés. La fenêtre de vue est supposée être un sous-ensemble convexe du plan d'image.

La ligne à l'infini  $L_\infty$  dans le plan projectif  $P^2$  se compose de tous les points avec la dernière coordonnée égal à 0. Soit  $W$  une région convexe du plan. Une transformation projective  $H$  est dite quasi-affine par rapport à  $W$  si  $H(W) \cap L_\infty = \emptyset$ . Il est clair que si l'épipôle  $p$  se trouve en dehors de la fenêtre de vue convexe  $W$ , alors il existe une projectivité quasi-affine par rapport à  $W$ , qui prend  $p$  au point  $(0, 0, 1)^T$ . En fait, n'importe quelle ligne passante par  $p$ , qui ne croise pas  $W$  peut être choisi comme la ligne  $H^{-1}(L_\infty)$ . La perspective 3.12 projette la ligne  $u = f$  à l'infini, c'est la ligne à travers l'épipôle parallèle à l'axe verticale de l'image. Si l'épipôle est suffisamment



(a) Peigne



(b) Peigne après rééchantillonnage

FIGURE 3.8: Un rééchantillonnage non-quasi-affine d'un peigne présenté dans [54]

éloigné de la fenêtre de vue, alors cette projection sera quasi-affine.

Si l'épipôle se trouve à l'intérieur de la fenêtre de vue d'une image, les techniques de cette section peuvent encore être appliquées en considérant une fenêtre de vue plus petite. Il est possible que la projectivité  $H'$  construite au paragraph §3.5.5 n'est pas quasi-affine, et dans ce cas la fenêtre de vue doit être réduite, ou une autre projectivité doit être choisie.

On passe maintenant à la question de savoir quand il est possible de trouver une paire de projectivités appariées  $H$  et  $H'$ , chacune quasi-affine par rapport à la fenêtre de vue de l'image respective. Il n'est pas à prévoir que ce sera toujours possible même lorsque les épipôles se situent en dehors des deux fenêtres de vues. Cependant, on peut faire presque aussi bien, comme le montre le théorème suivant.

**Théorème 3.5.2** *Considérez deux images  $I$  et  $I'$  avec vue fenêtres  $W$  et  $W'$ . Supposons que l'épipôle  $p'$  de l'image  $I'$  n'est pas dans  $W'$ . Soit  $H'$  une projectivité de  $I'$ , quasi-affine par rapport à  $W'$ , et projetant  $p'$  à l'infini, et soit  $H$  toute projectivité correspondante. Alors il existe une sous-fenêtre convexe  $W_+ \subseteq W$  tel que  $H$  est quasi-affine par rapport à  $W_+$  et tel que  $W_+$  contient tous les points de  $W$  qui correspondent à un point en  $W'$ .*

Si notre but dans le rééchantillonnage est de faciliter l'appariement de points, alors  $W_+$  contient toute la partie intéressante de l'image  $I$ . Le théorème affirme que  $H$  est quasi-affine par rapport à  $W_+$ , et ainsi  $W_+$  peut être rééchantillonnée selon  $H$  avec des résultats satisfaisants. Avant de prouver théorème précédant certain matériel préalable est nécessaire.

Un ensemble d'images correspondantes est appelé un ensemble définissant des correspondances d'image si les équations 3.5 ont une solution unique. Ainsi, un ensemble définissant des correspondances est celui qui contient suffisamment de points appariés pour déterminer la structure épipolaire de la paire d'image.

Ayant un ensemble de correspondances d'image  $u'_i \leftrightarrow u_i$ . Soit  $P$  et  $P'$  les matrices de caméra et  $x_i$  un ensemble de points dans  $P^3$ . La triple  $(P, P', x_i)$  est appelé une réalisation de l'ensemble des correspondances  $u'_i \leftrightarrow u_i$  si  $u_i = Px_i$  et  $u'_i = P'x_i$ . Notez qu'il n'est pas nécessaire que  $P$  et  $P'$  soient les matrices réelles des caméras, ou que les points  $x_i$  représentent les emplacements des points réels dans l'espace. En effet, sans calibrage de la caméra, la structure réelle de l'ensemble de points  $x_i$  peut être déterminée à partir de l'image correspondante uniquement à une transformation projective 3D.

Les théorèmes suivants sont basés sur une analyse des *cheirality* : c'est la détermination si les points sont derrière ou en face de la caméra. Les preuves sont donnés dans [51], qui contient une enquête approfondie sur *cheirality*.

Notez que lorsque l'égalité de vecteurs ( $=$ ) est considéré dans les théorèmes suivants, nous entendons égalité exacte, et non l'égalité à un facteur.

**Théorème 3.5.3** *Soit  $H$  une projectivité 2D et  $H(u_i, v_i, 1)^T = \alpha_i(\hat{u}_i, \hat{v}_i, 1)^T$  pour un certain ensemble de points  $u_i = (u_i, v_i, 1)^T \in \mathbb{R}^2$ . Alors  $H$  est quasi-affine par rapport à  $u_i$  si et seulement si tout  $\alpha_i$  ont le même signe.*

**Théorème 3.5.4** *Soit  $(P, P', x_i)$  une réalisation d'un ensemble définissant de correspondances d'images  $u'_i \leftrightarrow u_i$  dérivé d'un ensemble de points 3D physiquement réalisable. Soit  $x_i \simeq (x_i, y_i, z_i, t_i)^T$ ,  $u_i \simeq (u_i, v_i, w_i)^T$  et  $(u_i, v_i, w_i)^T = P(x_i, y_i, z_i, t_i)^T$ . Soit les quantités amorcées définis de façon similaire. Alors le signe de  $w_i w'_i$  est constant pour tout  $i$ .*

Maintenant, on peut prouver le théorème 3.5.2.

**Preuve.** Considérons les projectivités correspondantes  $H$  et  $H'$  pour une paire d'images  $I$  et  $I'$ , et supposons que  $H'$  est quasi-affine par rapport à  $W'$ . Soit  $u'_i \leftrightarrow u_i$  un ensemble définissant des correspondances avec  $u_i = (u_i, v_i, 1)^T \in W$  et  $u'_i = (u'_i, v'_i, 1)^T \in$

$W'$  pour tout  $i$ . Soit  $H(u_i, v_i, 1)^T = \alpha_i(\hat{u}_i, \hat{v}_i, 1)^T$  et  $H'(u'_i, v'_i, 1)^T = \alpha'_i(\hat{u}_i + \delta_i, \hat{v}_i, 1)^T$ . Étant donné que  $H'$  est quasi-affine, tout  $\alpha'_i$  ont le même signe. On veut prouver que tout  $\alpha_i$  ont le même signe. ■

Une réalisation pour  $u'_i \leftrightarrow u_i$  est donnée en mettant  $P = H^{-1}(I|0)$ ,  $P' = H'^{-1}(I|(1, 0, 0)T)$  et  $x_i = (\hat{u}_i, \hat{v}_i, 1, \delta_i)^T$ . En effet, on vérifie que :

$$\begin{aligned} Px_i &= H^{-1}(I|0)(\hat{u}_i, \hat{v}_i, 1, \delta_i)^T \\ &= H^{-1}(\hat{u}_i, \hat{v}_i, 1)^T = \alpha_i^{-1}(u_i, v_i, 1)^T \end{aligned}$$

et

$$\begin{aligned} P'x_i &= H'^{-1}(I|(1, 0, 0)T)(\hat{u}_i, \hat{v}_i, 1, \delta_i)^T \\ &= H'^{-1}(\hat{u}_i + \delta_i, \hat{v}_i, 1)^T = \alpha'^{-1}(u'_i, v'_i, 1)^T \end{aligned}$$

Il suit du théorème 3.5.4 que  $(\alpha_i \alpha'_i)^{-1}$  et donc  $\alpha_i \alpha'_i$  a le même signe pour tout  $i$ . Cependant, par hypothèse,  $H'$  est quasi-affine par rapport à tous les  $u'_i$ , et donc tout  $\alpha'_i$  possèdent le même signe, et donc la même chose pour tout  $\alpha_i$ . Cela signifie que tous les  $u_i$  se trouvent d'un côté de  $H^{-1}(L_\infty)$ . On définit  $W_+$  pour être la partie de  $W$  qui se trouve sur le même côté de  $H^{-1}(L_\infty)$  comme tous les  $u_i$ , et la preuve est complète.

### 3.5.5 Rééchantillonnage

Une fois que les deux transformations de rééchantillonnage  $H$  et  $H'$  ont été déterminées, la paire d'images peut être rééchantillonnée. Il ya deux étapes, d'abord déterminer l'étendue des images rééchantillonnées, et deuxièmement, réaliser le rééchantillonnage.

#### Déterminer les dimensions de l'image de sortie

Supposons que l'intervalle des projectivités de rééchantillonnage  $H$  et  $H'$  est le même plan  $\hat{J}$ . Supposons que la projectivité  $H'$  est quasi-affine par rapport à la fenêtre,  $W'$ . Alors  $H'(W')$  sera une région convexe dans  $\hat{J}$ . En fait, si  $W$  est une région polygonale, il en est de  $H'(W')$ , et elle peut être facilement calculée. Quant à  $H(W)$ , si  $H$  est quasi-affine par rapport à  $W$ , alors  $H(W)$  sera également un ensemble convexe délimitée.

En d'autre part, si  $H$  n'est pas quasi-affine par rapport à toute la fenêtre  $W$ ,  $H(W)$  sera divisée en deux parties,  $H(W_+)$  et  $H(W_-)$  qui s'étend à l'infini, avec des points à l'infini dans  $\hat{J}$ . Selon le théorème 3.5.2, une seule des deux régions  $W_-$  et  $W_+$  (supposons  $W_+$ ) contient des points qui correspondent avec les points dans  $W'$ . C'est une question simple de programmation géométrique de déterminer l'intersection  $H'(W') \cap H(W_+)$ .

La fenêtre  $\hat{W}$  de rééchantillonnage dans  $\hat{J}$  peut alors être choisie. Normalement,  $\hat{W}$  doit être une région rectangulaire alignée avec les axes de coordonnées dans  $\hat{J}$ . C'est une question de choix si  $\hat{W}$  est choisie comme le plus petit rectangle contenant  $H'(W') \cap H(W_+)$  ou si  $\hat{W}$  doit être un rectangle contenu dans  $H'(W') \cap H(W_+)$ .

### Rééchantillonnage

Une fois la fenêtre  $\hat{W}$  est choisie, il est facile de rééchantillonner chacune des images. Considérez le rééchantillonnage de la première image. Pour chaque position de pixel (c'est le point avec des coordonnées entières)  $\hat{u}$  dans  $\hat{W}$ , la position correspondante  $u = H^{-1}\hat{u}$  dans  $J$  est calculée, et la «couleur» ou l'intensité d'image à  $u$  est déterminée. Le pixel  $\hat{u}$  est ensuite coloré avec cette couleur. Si  $u$  se trouve en dehors de la fenêtre de vue  $W$  ou  $W_+$ , alors le pixel  $\hat{u}$  est coloré de la couleur du fond choisi. Comme le point  $u$  n'aura pas les coordonnées entières, il est nécessaire dans la détermination de la couleur de  $u$  d'interpoler. Dans les images présentées dans cette section, l'interpolation bicubic a été utilisée, et ça a donné des résultats satisfaisants. Dans d'autres cas, comme si aliasing devient une question importante, une méthode plus sophistiquée de l'interpolation doit être utilisé.

## 3.6 Expérimentations

Dans cette partie, nous allons présenter le logiciel de la Stéréovision / Calibration

la première partie de ce logiciel sert à calculer la matrice fondamentale à partir d'un ensemble de pair points choisis dans deux images prises de la même scène vue en deux différents points d'observation.

Tous les résultats de cette partie sont exécuté sur une machine avec un processeur **intel core i7** quatre coeur et une mémoire de **8 Go**. L'application a été compilé sous **windows 8** avec en utilisant le langage de programmation **C#** sous l'environnement **Visual Studio 2012**.

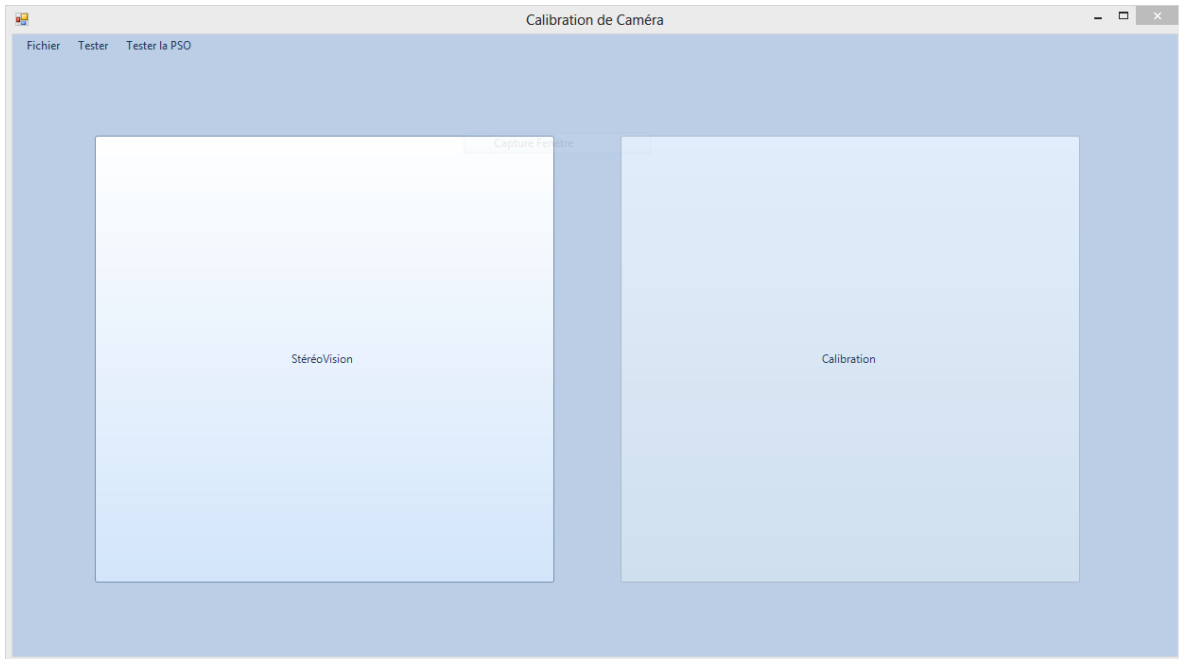


FIGURE 3.9: Capture d'écran de l'interface d'accueil du logiciel

La figure 3.9 montre l'interface principale du logiciel, ce dernier est composé de deux parties essentielles, la première est dédiée à la stéréovision qui va être décrite ici, la deuxième est conçue pour la calibration des caméras (voir la section 5.5).

Une fois l'utilisateur a choisi son objectif, dans ce cas la Stéréovision, l'interface présenté dans la figure 3.10 est affichée et l'utilisateur peut ouvrir la pair d'images sujet.

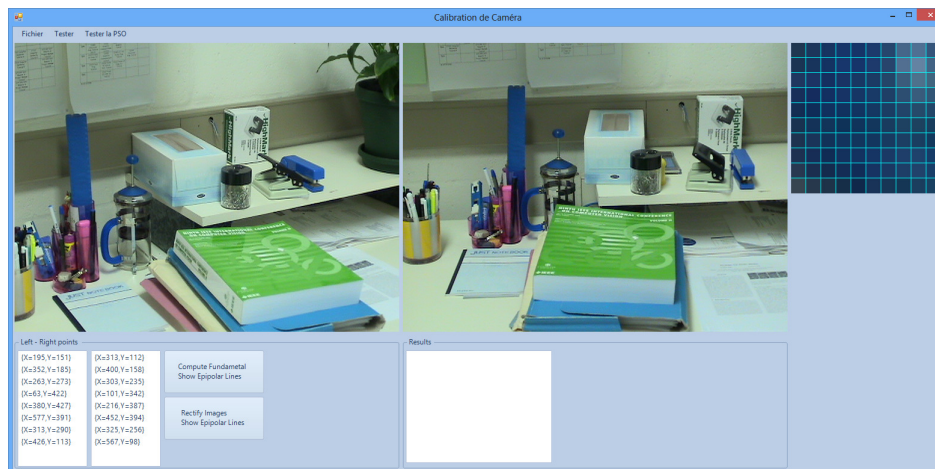


FIGURE 3.10: L'interface dédiée à la StéréoVision

L'interface est composée de plusieurs régions, on les explique comme suivant :

- Les deux zones d'images où on affiche les images sujets gauche et droite.

- Left - Right points : les points choisis dans les deux images gauche et droite dans l'ordre des clics de l'utilisateur.
- Compute Fundamental - Show epipolar lines : sert à calculer la matrice fondamentale et à afficher les lignes épipolaires correspondantes aux points choisis par l'utilisateur.
- Rectify Images - Show epipolar lines : ce bouton est dédié au calcul des homographies pour la rectification des deux images, et affiche les images réctifié ainsi que les nouvelles lignes épipolaires correspondantes après la rectification.
- Results : cette zone est consacré à l'affichage des résultats numériques.
- La zone d'image à l'extrême coin supérieur de la fenêtre est un zoom local à la zone parcourue par le pointeur, qui sert à faciliter le choix des points.

Les figures 3.11,3.12,3.13,3.14 montre les étapes que l'utilisateur parcourt durant le processus de la stéréovision et la rectification des images.

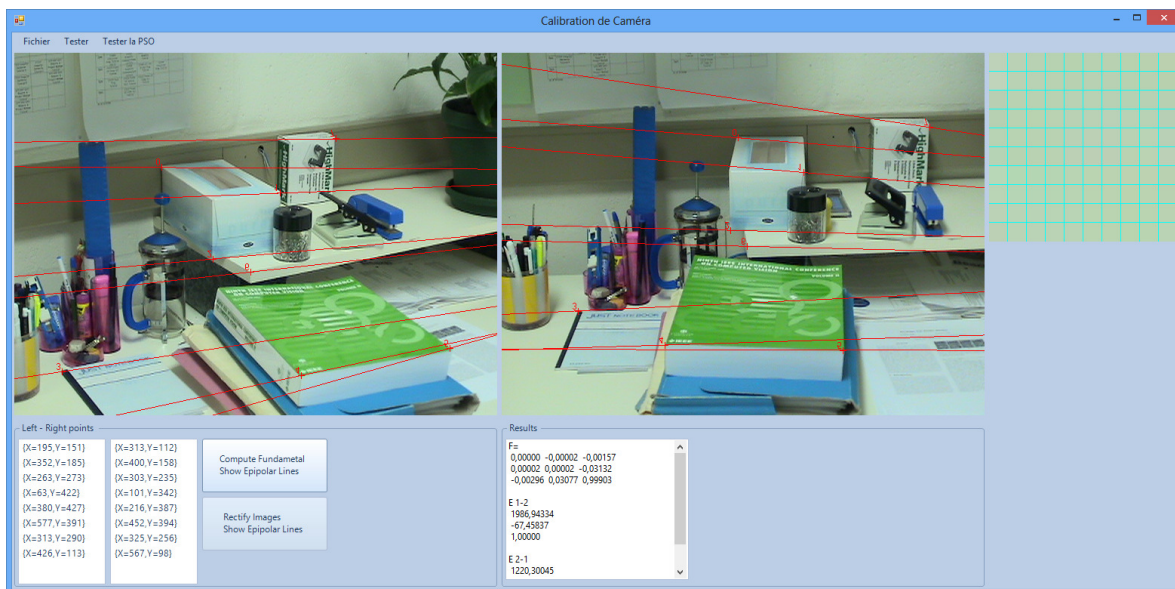


FIGURE 3.11: Affichage des résultats du calcul de la matrice fondamentale et projection des lignes épipolaires (exemple 1)

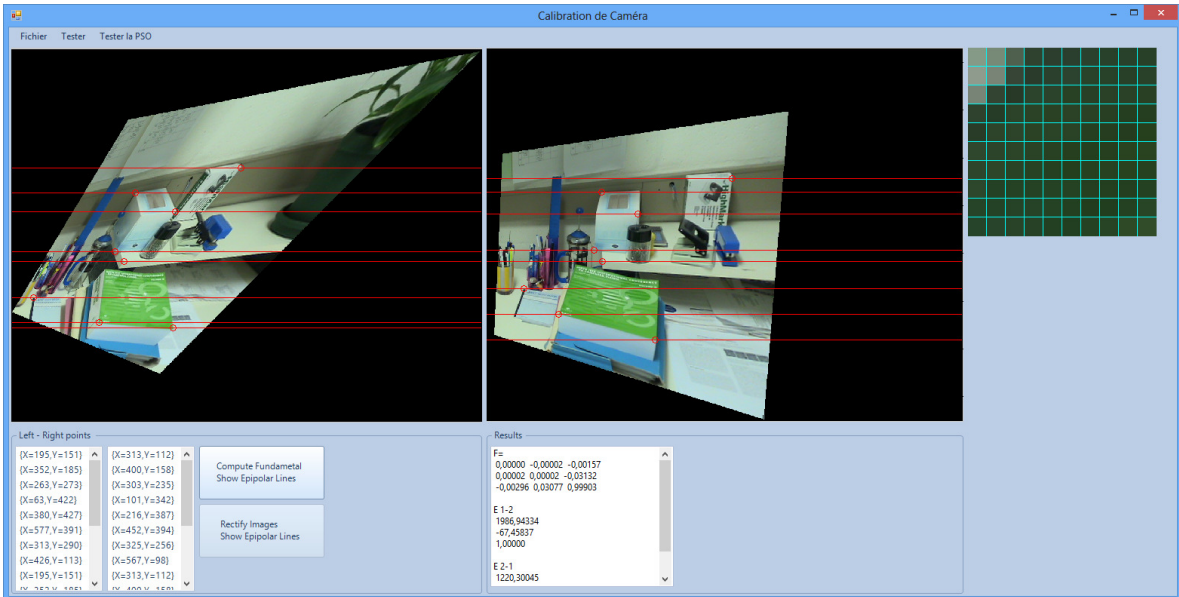


FIGURE 3.12: Affichage des résultat de la rectification et projection des lignes épipolaires (exemple 1)

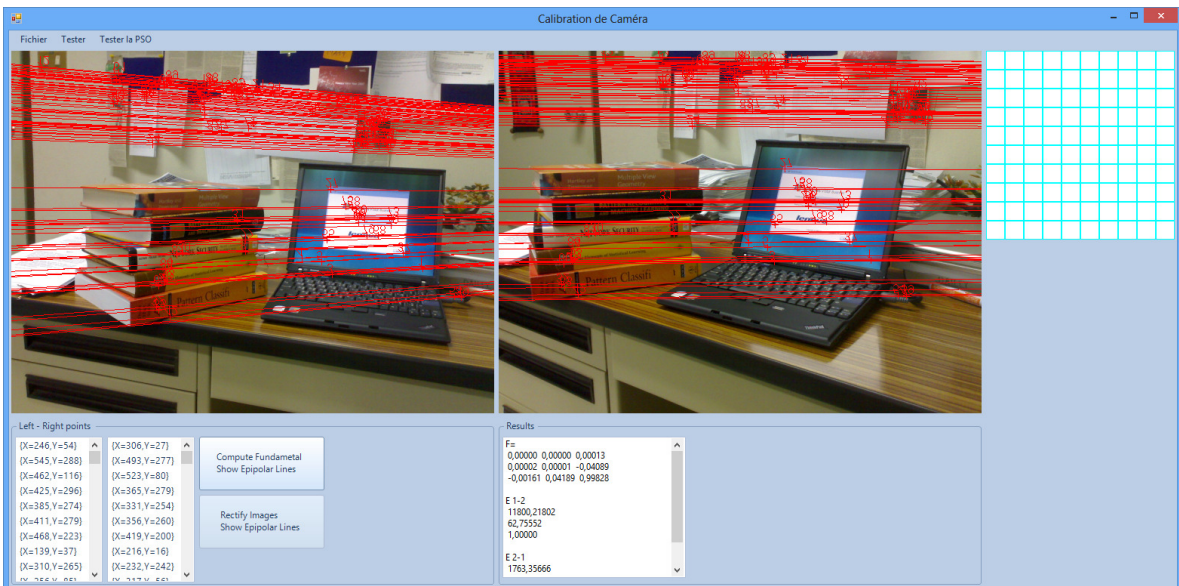


FIGURE 3.13: Affichage des résultats du calcul de la matrice fondamentale et projection des lignes épipolaires (exemple 2)

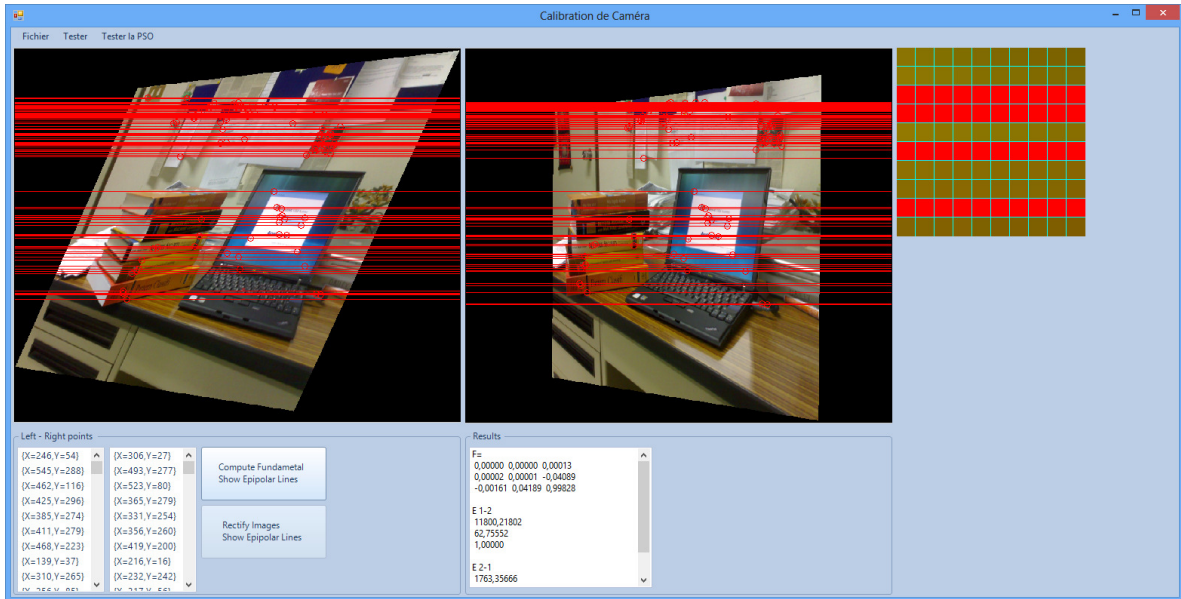


FIGURE 3.14: Affichage des résultat de la rectification et projection des lignes épipolaires (exemple 2)

Dans cette partie du travail, on a parcouru les différentes situations de la Stériovision, et expliqué le calcul de la matrice fondamentale qui appliquent une transformation projective à deux images de la même scène dans le cas des caméras non calibrées. Ensuite, on a calculé la matrice essentielle pour la scène dans le cas où les caméras ont été calibrées. La rectification de ces images est une étape qui a été aussi expliquée et implémentée pour faciliter l'appariement des points d'intérêts et faciliter le processus de reconstruction 3D. Des résultats de notre implémentation de cette partie ont été illustrés à la fin.

# Chapitre 4

## Homography

### Sommaire

---

<b>4.1</b>	<b>Définition Générale</b>	<b>63</b>
<b>4.2</b>	<b>Description Mathématique</b>	<b>63</b>
<b>4.3</b>	<b>L'estimation de l'Homography</b>	<b>65</b>
<b>4.4</b>	<b>Résolution analytique de l'Homographie</b>	<b>65</b>
<b>4.5</b>	<b>Résolution de l'Homographie basée sur les essaims de particules</b>	<b>66</b>
4.5.1	Principe	66
4.5.2	Première estimation de la matrice H	66
<b>4.6</b>	<b>Expérimentations</b>	<b>68</b>
4.6.1	Application sur des images synthèses	68
4.6.2	Application sur l'échiquier	75
4.6.3	Application sur la base de données INRIA	78
4.6.4	Discussion	83
<b>4.7</b>	<b>Conclusion</b>	<b>83</b>

---

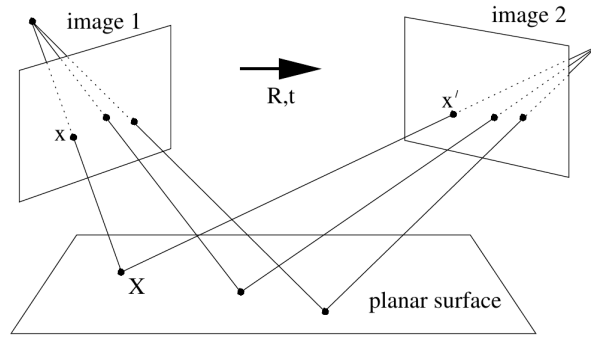
Tout le processus de la reconstruction 3D utilise une certaine transformation projective dite Homographie. On a mentionné dans le chapitre précédant cette transformation, sans l'expliquer exactement. L'utilité de l'homographie dans la rectification ainsi que la projection de l'épipôle vers l'infini est une application parmi plusieurs dans ce travail. Ainsi vu l'importance de l'homographie dans la reconstruction ainsi que les complications liés aux définitions mathématiques et analytiques de cette projection et son calcul, l'introduction des métaheuristiques pour remplacer et simplifier la méthode classique est clairement nécessaire. L'approche bio-inspirée proposée dans cette partie sera la PSO. Des résultats numériques et visuels seront présentés et discutés à la fin pour valider cette partie du travail.

## 4.1 Définition Générale

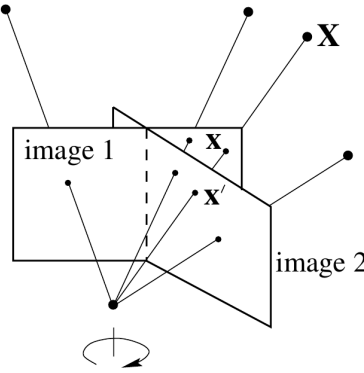
L'Homography est une relation non-singulière linéaire entre des points sur des plans [2]. L'Homography entre deux vues joue un rôle très important dans la géométrie des vues multiples. Les images des points d'un plan dans une vue sont reliées à leurs correspondantes dans une autre vue par une Homography planaire en utilisant une représentation homogène. Cette relation est dite projective car elle ne dépend que de l'intersection entre les plans avec les lignes. L'Homography projette les points d'une vue vers une autre comme si c'était des images de points d'un plan. L'Homography induite d'un plan est unique jusqu'à un facteur d'échelle déterminé par huit (8) paramètres ou degrés de liberté. L'Homography dépend des paramètres intrinsèques et extrinsèques des caméras utilisés pour les deux vues, et les paramètres du plan 3D, Fig. 4.1 montre les différents cas de l'Homography.

## 4.2 Description Mathématique

Dans le plan complexe, une transformation Mobius est généralement appelée une Homography. Ces transformations linéaires et fractionnelles sont des expressions des transformations projective sur la ligne projective complexe, qui est une extension du plan complexe. Dans le cas des dimensions élevés, les coordonnées homogènes sont utilisées pour représenter les transformations projectives par le moyen de multiplications des matrices. Avec les coordonnées cartésiennes, la multiplication des matrices ne peut pas exécuter la division requise pour la projection perspective. Avec d'autres mots, en



(a) Deux ou plusieurs caméras qui prennent des images du même plan



(b) Une caméra qui tourne sur son axe et prend des images

FIGURE 4.1: Les différents cas possibles de l'Homographie

utilisant les coordonnées cartésiennes, la projection perspective est une relation non-linéaire. On a :

$$m = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} M' = \begin{bmatrix} w \times X \\ w \times Y \\ w \end{bmatrix} H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Alors :

$$m = H \times M'$$

Ou :

$$M = \frac{M'}{w} = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Avec H : Matrice d'Homography. M : Les points du modèle dans l'espace. m : La projection des points du modèle The modèle sur l'image.

### 4.3 L'estimation de l'Homography

Les points sont les caractéristiques fondamentales les plus simples qui peuvent être utilisés pour estimer l'Homography. Ils ont été toujours le centre d'intérêt des chercheurs pour l'estimation de l'Homography. Ces méthodes récupèrent l'Homography comme une relation linéaire des points et lignes. On cite quelques une :

- Transformation Linéaire Directe(TLD) utilisant la correspondances entres points.
- TLD utilisant la correspondances entres lignes.
- Estimation robuste utilisant RANdom SAmple Consensus (RANSAC).
- Optimisation statistique.

### 4.4 Résolution analytique de l'Homographie

Il y a tellement de méthodes pour estimer l'Homographie entre le plan du modèle et son image. Ici, on présente une technique basée sur le critère du maximum de vraisemblance. Soit  $M_i$  et  $m_i$  les points du modèle et d'images respectivement. Idéalement, ils doivent satisfaire l'équation 5.2. Dans la pratique, ils le font pas à cause du bruit présent dans les images extraites. On assume que  $m_i$  est corrompue par un bruit Gaussien avec une moyenne de 0 et une matrice de covariance  $\Lambda_{m_i}$ . Alors, l'estimation du maximum de vraisemblance de  $H$  est obtenue en minimisant l'équation suivante :

$$\sum_i (m_i - \hat{m}_i)^T \Lambda_{m_i}^{-1} (m_i - \hat{m}_i)$$

Où :

$$\hat{m}_i = \frac{1}{\bar{h}_3^T M_i} \begin{bmatrix} \bar{h}_1^T M_i \\ \bar{h}_2^T M_i \end{bmatrix}$$

Avec  $\bar{h}_i$ , la  $i^{\text{ème}}$  ligne de  $H$ . Dans la pratique, on assume simplement que  $\Lambda = \sigma^2 I$  pour tous les  $i$ . C'est raisonnable si les points sont extraits indépendemment avec la même procédure. Dans ce cas, le problème ci-dessus devient un problème de non-linéaire des moindres carrés, i.e.  $\min_H \sum_i \| m_i - \hat{m}_i \|^2$ . La minimisation non-linéaire est conduite par l'algorithme de Levenberg-Marquardt comme implémenté dans Minipack [79]. ça requis une estimation initiale qui peut être obtenue comme suivant : Soit  $x =$

$[\bar{h}_1^T, \bar{h}_2^T, \bar{h}_3^T]$ . Alors l'équation 5.2 peut être écrite comme ça :

$$\begin{bmatrix} \tilde{M}^T & 0^T & -u\tilde{M}^T \\ 0^T & \tilde{M}^T & -v\tilde{M}^T \end{bmatrix} x = 0$$

Quand on a  $n$  points, on a  $n$  équations comme ci-dessus, qui peuvent être écrites sous forme de matrice d'équation comme  $Lx = 0$ , où  $L$  est une matrice de  $2n \times 9$ . Comme  $x$  est défini à un facteur d'échelle, la solution est bien connue comme le vecteur singulier droit de la matrice  $L$  associé à la plus petite valeur singulière.

Dans  $L$ , quelques éléments sont constants =1, d'autres sont des pixels, d'autres sont des coordonnées du monde réel, et d'autres sont la multiplication des deux. ça va rendre  $L$  numériquement pauvrement conditionnée. Des meilleurs résultats peuvent être obtenues en effectuant une normalisation simple de données, comme il a été proposé dans [53], avant d'exécuter la procédure ci-dessus.

## 4.5 Résolution de l'Homographie basée sur les essais de particules

### 4.5.1 Principe

Le processus de la résolution de l'Homographie est devisé en deux parties, la première c'est de supposer qu'il n'y a pas de distorsions appliqué sur le plan de l'image, et procéder simplement par calculer l'Homographie des quatre coins du plan du modèle, ce qui va être discuté dans le prochain paragraphe.

### 4.5.2 Première estimation de la matrice H

Ayant les coordonnées des points de modèle dans l'espace (les axes  $(\vec{X}, \vec{Y})$  et l'origine choisis dans l'image) et leur correspondant dans les pixels de l'image, on suit l'opération inverse de la multiplication des matrices.

$$H \times M = m \tag{4.1}$$

Chaque particule va représenter un vecteur de neuf positions réelles qui représentent la solution de la matrice de l'Homographie. Durant l'étape prochaine, chaque particule

essayera de trouver une solution pour :

$$m - H \times M = 0$$

### Description de l'algorithme

La résolution de l'Homographie basée sur la PSO passe par plusieurs étapes pour trouver les meilleures valeurs de la matrice dites optimaux. Ainsi, l'algorithme utilisé va évoluer avec un nombre maximum de générations après quelques expérimentations.

**Etape 1. Initialisation** Pour chaque particule, une position et une vitesse aléatoires vont être données, ces derniers doivent être dans un intervalle spécifié. Le choix de cet intervalle a un effet important sur la convergence, un grand intervalle peut causer une convergence prématurée, et un qui est petit rend l'opération trop longue.

**Etape 2. Evaluation** Le succès de la PSO est équivalent le bon choix de la fonction d'évaluation qui appelée fonction de fitness. La raison principale de cette fonction est de maximiser/minimiser son résultat. Pour ce problème, la fonction utilisé est :

$$m - H \times M = 0$$

Avec "m", "M" sont des matrices de  $3 \times n$  (n nombre de points choisis dans l'image), et "H" est une matrice de  $3 \times 3$  décrits comme suit :

$$H = \begin{bmatrix} x_1 & x_4 & x_6 \\ x_2 & x_5 & x_7 \\ x_3 & x_6 & x_9 \end{bmatrix}$$

Avec  $X_i$  ( $i=1\dots9$ ) sont les vecteurs de positions de la particule. Donc pour calculer la fitness, la particule doit avoir une position qui satisfais l'équation Eq.4.1 avec la condition suivante : La première et deuxième lignes de ma matrice résultat sont obtenues par leur divisions sur la troisième ligne (Cordonnées homogènes). Pour avoir une bonne contrainte sur le vecteur de position, l'équation Eq.4.1 a été modifié à :

$$m - y = m - (M^T \times H^T) = 0 \tag{4.2}$$

Avec :

$$y_{i,1} = x_1 \times M_{i,1} + x_2 \times M_{i,2} + x_3 \times M_{i,3} \quad (4.3)$$

$$y_{i,2} = x_4 \times M_{i,1} + x_5 \times M_{i,2} + x_6 \times M_{i,3} \quad (4.4)$$

$$y_{i,3} = x_7 \times M_{i,1} + x_8 \times M_{i,2} + x_9 \times M_{i,3} \quad (4.5)$$

La minimisation de la distance Euclidienne (Sans la racine carré) doit être fait après le calcul d' $y$  :

$$fitness = \sum_{i=1, j=1}^{n,3} (m_{i,j} - y_{i,j})^2 \quad (4.6)$$

Avec une contrainte en plus :  $y_{i,j} = m_{i,j}$  est vraie si et seulement si  $y_{i,j}$  est multiplié par  $y_{i,3}$  :

$$y_{i,j} = y_{i,j} \times y_{i,3}$$

La dernière chose à faire dans cette partie est de voir si la fitness d'une particule donnée est sa meilleure pour l'enregistrer ou garder la valeur ancienne, la même opération est faite pour la fitness des voisins.

**Étape 3. La mise à jour** dans cette étape, tous les positions et vitesses de chaque particule doit être mise à jour en utilisant les équations standard 2.3,2.4. Pour cela, toute particule va mettre à jour sa position en respectant l'intervalle de l'espace de recherche.

## 4.6 Expérimentations

### 4.6.1 Application sur des images synthèses

L'algorithme a été testé plusieurs fois sur des images de synthèse ainsi que sur d'autres réelles, avec un ensemble de paramètres qu'on change à chaque fois. Les tableaux 4.1, 4.2, 4.3, 4.4 montrent l'évolution de l'essaim avec différentes valeurs des paramètres et données. Une fois la meilleure combinaison des paramètres est obtenue, d'autres tests ont été faits (approximativement 5 pour chaque ensemble de données) pour comparer la méthode proposée avec l'approche classique. Premièrement, les mêmes points sont sélectionnés dans les deux images. Ensuite, l'essaim démarre avec les positions des pixels comme des données initiales. Le résultat finale sera la matrice d'Homographie, qui n'est d'autre que la meilleure position de la particule avec la fitness

la plus basse.

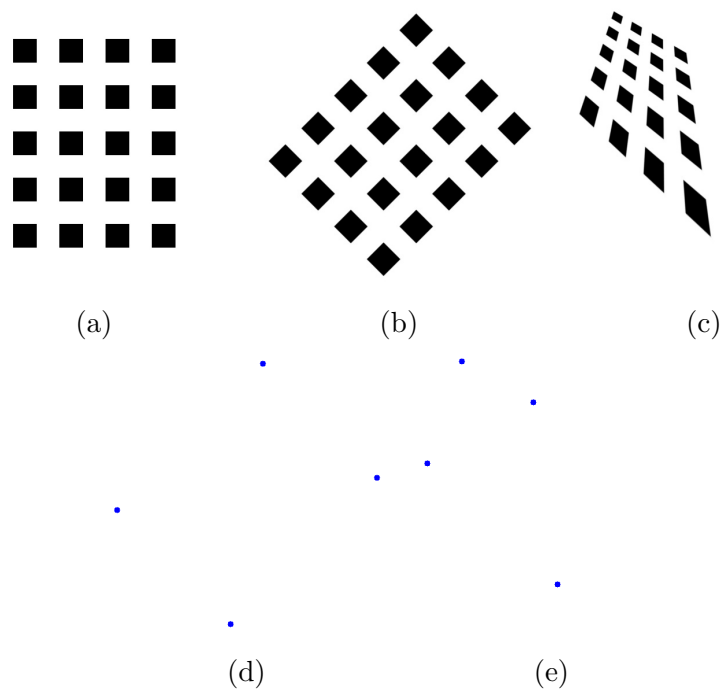


FIGURE 4.2: Échiquier synthétisé avec 4 points

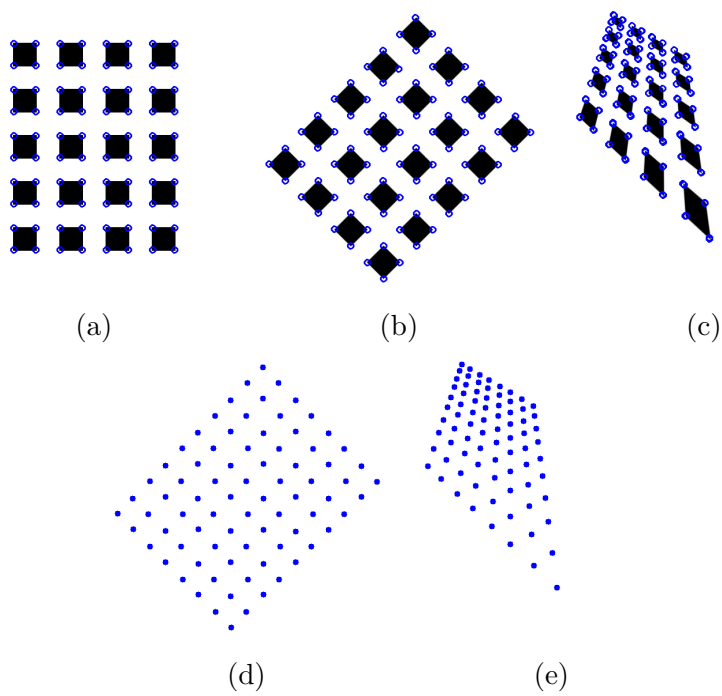


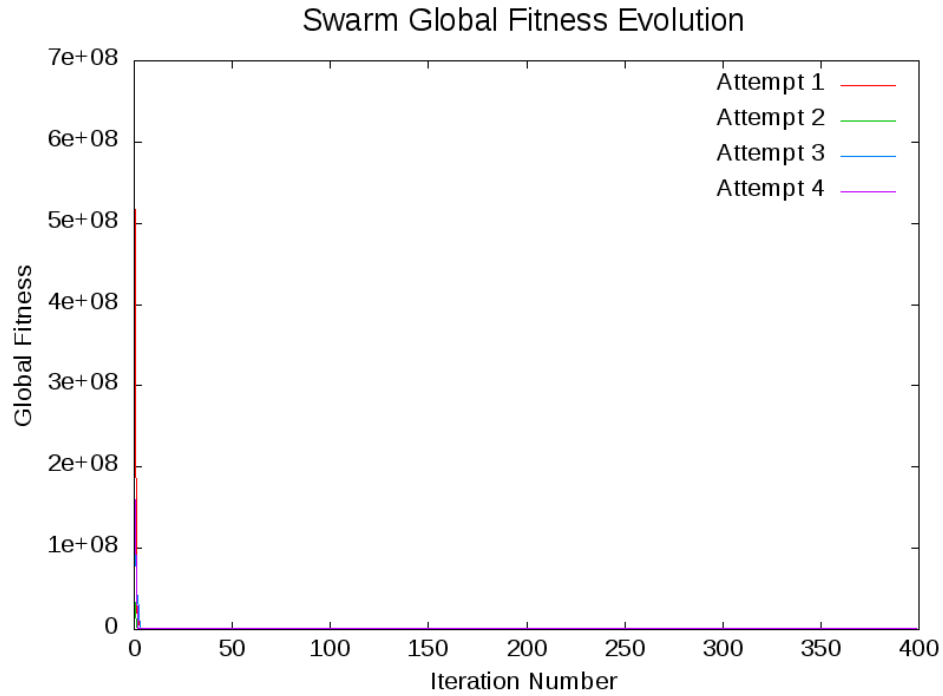
FIGURE 4.3: Échiquier synthétisé avec 80 points

TABLE 4.1: Résultat d'un test avec 4 points (choisi parmi plusieurs tests)

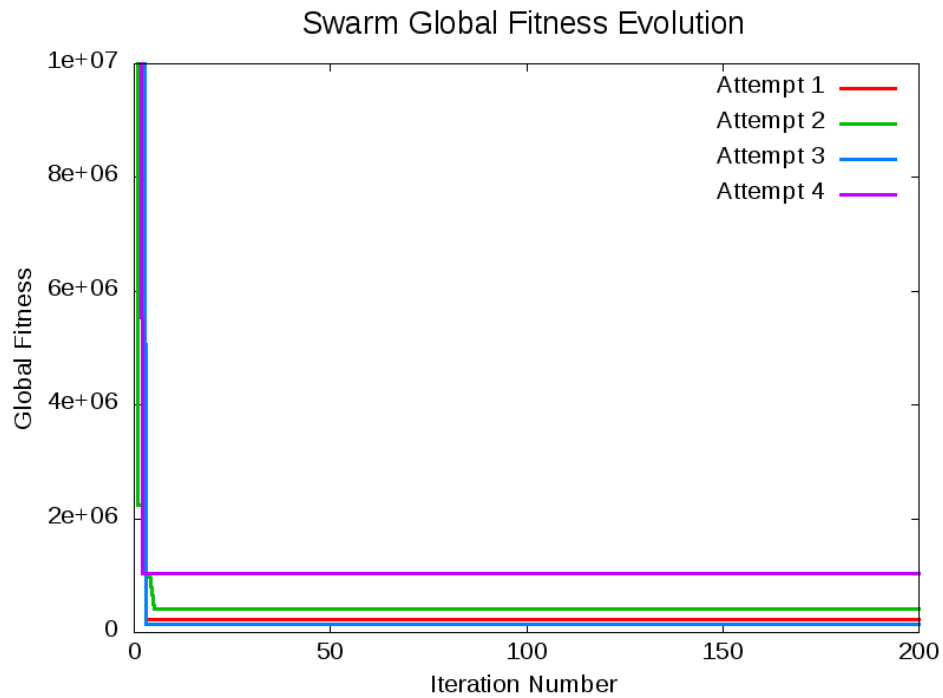
<b>W</b>	<b>C1</b>	<b>C2</b>	<b>Iterations</b>	<b>Global Fitness</b>
0.1	0.5	0.5	400	8.18323e+08
0.1	1	1	400	7.03306e+07
0.1	1.5	1.5	600	96085.2
0.1	2	2	1200	3324.06
0.3	0.5	0.5	400	7.63895e+07
0.3	1	1	400	90475.5
0.3	1.5	1.5	600	81440.8
0.3	2	2	1000	107.077
0.5	0.5	0.5	400	94988.7
0.5	1	1	800	49684.2
0.5	1.5	1.5	1000	40456.9
0.5	2	2	2440	9.83057e-12
0.7	0.5	0.5	800	133532
0.7	1	1	1400	56081.5
0.7	1.5	1.5	1198	9.09916e-12
0.7	2	2	800	66475.8
0.9	0.5	0.5	4600	13619.4
0.9	1	1	10000	8.2621e-07
0.9	1.5	1.5	600	91244.5
0.9	2	2	400	229683

TABLE 4.2: Résultat d'un autre test avec 4 points (choisi parmi plusieurs tests)

<b>W</b>	<b>C1</b>	<b>C2</b>	<b>Iterations</b>	<b>Global Fitness</b>
0.1	0.5	0.5	400	4.8759e+08
0.1	1	1	400	3.33552e+08
0.1	1.5	1.5	800	80183
0.1	2	2	600	8693.51
0.3	0.5	0.5	400	3.98901e+08
0.3	1	1	400	84847.1
0.3	1.5	1.5	600	82581.1
0.3	2	2	4000	0.00344849
0.5	0.5	0.5	400	149947
0.5	1	1	800	98642.7
0.5	1.5	1.5	800	31321.9
0.5	2	2	3235	9.88459e-12
0.7	0.5	0.5	800	96266.2
0.7	1	1	1600	44893.8
0.7	1.5	1.5	1340	9.94529e-12
0.7	2	2	800	14928.4
0.9	0.5	0.5	5400	26.6547
0.9	1	1	8123	9.92649e-12
0.9	1.5	1.5	400	102442
0.9	2	2	400	410869



(a) The hole graph



(b) A zoom in region

FIGURE 4.4: L'évolution de l'essaim en utilisant 4 points

Les tableaux 4.1, 4.2 montrent que les paramètres optimaux sont  $W=0.7$ ,  $C1=C2=1.5$ . Une autre combinaison peut être utilisées :  $W=2$  and  $C1=C2=2.0$ , mais le

nombre d'itérations requis a augmenté. La courbe 4.4 montre l'évolution de la meilleure fitness de l'essaim dans le temps. Comme on peut le constater, la convergence de l'algorithme est très rapide en général, mais elle nécessite quelques itérations en plus pour avoir une bonne précision. L'algorithme change un peu son comportement dans le cas d'un grand set de données, comme on peut le voir dans le tableau 4.3 et 4.4, l'essaim converge vers la solution, mais se stagne avec une fitness globale de 350. Ainsi, la meilleure solution pour résoudre ce problème est la normalisation des données, le tableau 4.5 montre le résultat après une normalisation affine des données : Le nombre d'itérations a diminué presque à la moitié, et la fitness globale est moins que 1.0 ce qui vraiment bien. La courbe 4.5 montre que même avec un grand set de données, la vitesse de convergence de l'essaim est très rapide. L'algorithme a été testé plusieurs fois, et chaque fois le résultat est obtenu avec la même approximation.

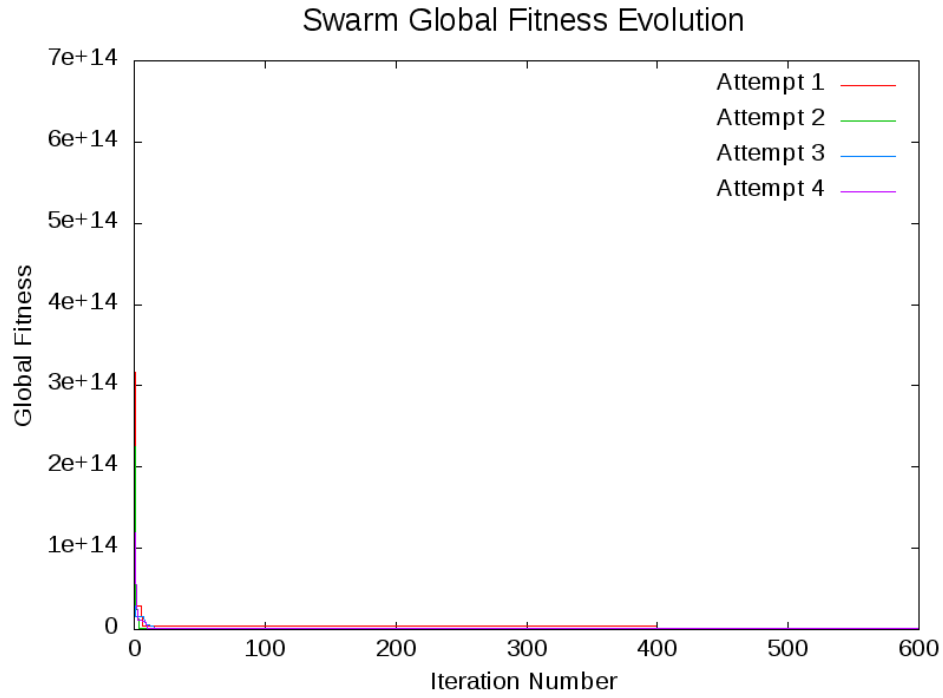
**Remarque :** 20 particules ont été utilisé pour tous les tests, ce choix sera justifié plus tard.

TABLE 4.3: Résultats d'un test avec 80 points (choisi parmi plusieurs tests)

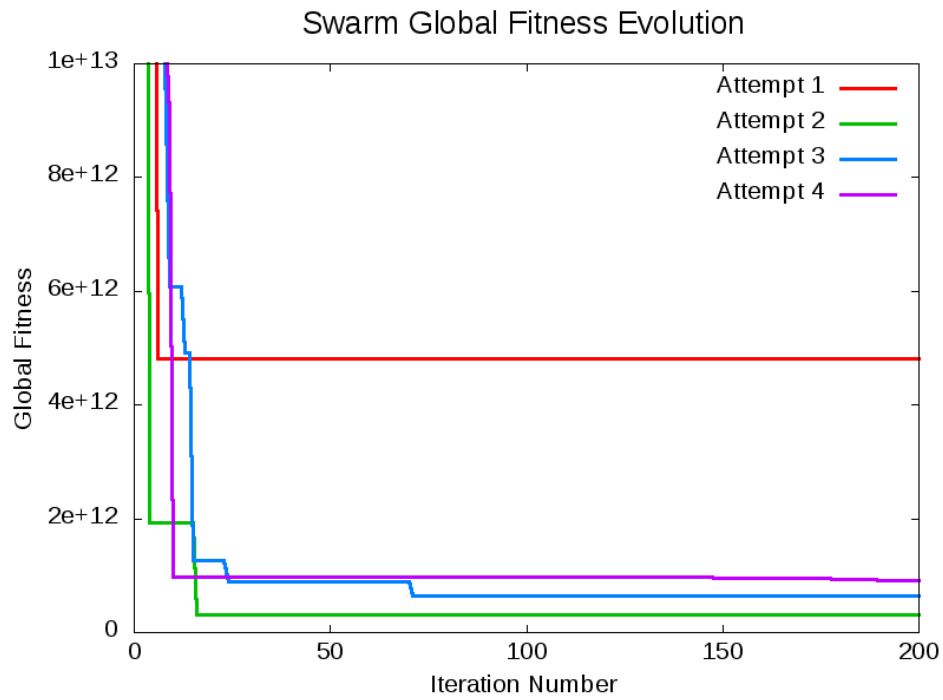
W	C1	C2	Iterations	Global Fitness
0.1	0.5	0.5	400	1.73597e+15
0.1	1	1	400	3.35456e+14
0.1	1.5	1.5	600	4.67195e+09
0.1	2	2	800	6.69405e+08
0.3	0.5	0.5	400	2.37662e+14
0.3	1	1	400	1.10319e+09
0.3	1.5	1.5	800	4.50737e+08
0.3	2	2	3000	380.449
0.5	0.5	0.5	400	2.76751e+09
0.5	1	1	600	1.82068e+09
0.5	1.5	1.5	1000	3.36625e+09
0.5	2	2	4200	389.062
0.7	0.5	0.5	800	8.99348e+09
0.7	1	1	1400	1.0974e+09
0.7	1.5	1.5	3800	353.663
0.7	2	2	1400	2.21473e+09
0.9	0.5	0.5	6400	7159.37
0.9	1	1	10000	418174
0.9	1.5	1.5	400	3.80877e+10
0.9	2	2	400	4.8225e+12

TABLE 4.4: Résultats d'un autre test avec 80 points (choisi parmi plusieurs tests)

<b>W</b>	<b>C1</b>	<b>C2</b>	<b>Iterations</b>	<b>Global Fitness</b>
0.1	0.5	0.5	400	1.76293e+15
0.1	1	1	400	8.65689e+14
0.1	1.5	1.5	600	9.52011e+13
0.1	2	2	1000	8.91141e+07
0.3	0.5	0.5	400	4.80177e+14
0.3	1	1	400	1.93128e+10
0.3	1.5	1.5	800	6.82383e+09
0.3	2	2	2400	13549.1
0.5	0.5	0.5	400	1.73609e+13
0.5	1	1	600	2.94359e+09
0.5	1.5	1.5	1600	8.58555e+08
0.5	2	2	10000	353.663
0.7	0.5	0.5	800	7.33597e+09
0.7	1	1	1200	1.2126e+08
0.7	1.5	1.5	3800	353.663
0.7	2	2	400	9.5797e+10
0.9	0.5	0.5	6000	30012.4
0.9	1	1	2800	5.90055e+06
0.9	1.5	1.5	600	2.23666e+11
0.9	2	2	400	3.06725e+11



(a) The hole graph



(b) A zoom in region

FIGURE 4.5: L'évolution de l'essaim en utilisant 80 points

TABLE 4.5: L'évolution de l'essaim en utilisant 80 points et la Normalisation

W	C1	C2	Iterations	Global Fitness
0.1	0.5	0.5	400	1.71322e+10
0.1	1	1	400	1.25109e+10
0.1	1.5	1.5	600	5.45375e+09
0.1	2	2	1000	510726
0.3	0.5	0.5	400	2.0747e+10
0.3	1	1	400	4.72099e+09
0.3	1.5	1.5	800	7.59147e+08
0.3	2	2	2000	282.193
0.5	0.5	0.5	400	6.06712e+09
0.5	1	1	600	179843
0.5	1.5	1.5	800	151690
0.5	2	2	3000	0.0351003
0.7	0.5	0.5	800	1.3761e+06
0.7	1	1	1400	35.4015
0.7	1.5	1.5	1600	0.0351003
0.7	2	2	800	3.59181e+06
0.9	0.5	0.5	7200	45707.9
0.9	1	1	6600	0.0351003
0.9	1.5	1.5	800	8.60617e+08
0.9	2	2	400	9.74705e+09

## 4.6.2 Application sur l'échiquier

Dans cette section, seulement 4 points ont été utilisés (pas nécessairement). Dans ce qui va suivre, les paramètres de l'algorithme vont être changés pour estimer le temps d'exécution et sa performance en générale.

### Premier Exemple

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$m = \begin{bmatrix} 467.1576 & 509.9683 & 138.1927 & 172.3374 \\ 131.4335 & 390.1121 & 379.1524 & 108.2834 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$$

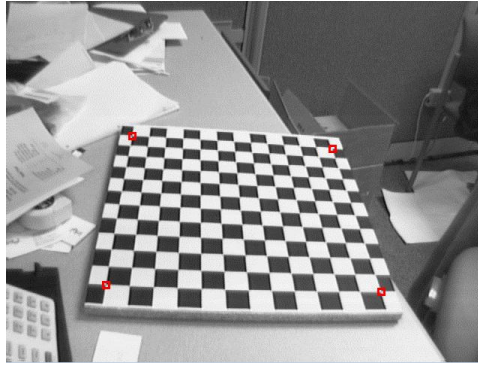


FIGURE 4.6: Sélection des quatre coins qui correspondent aux axes et l'origine

TABLE 4.6: Résultats de l'expérimentation 1 (Temps en millisecondes)

Nombre de Particules	10	20	30	50
Temps d'exécution	171	106	110	127
Meilleure fitness	$4.92 \times 10^{-4}$	$2.15 \times 10^{-3}$	$1.50 \times 10^{-4}$	$8.50 \times 10^{-5}$

$$H_{PSO-Based} = \begin{bmatrix} -69.1738 & -274.3659 & 434.6555 \\ 85.952 & -25.5094 & 196.4818 \\ -0.2699 & -0.0498 & 1.000 \end{bmatrix}$$

Les résultats montrés en haut, ont été obtenues avec l'algorithme PSO en utilisant 20 particules, ce qui le meilleure choix de la taille de l'essaim pour notre problème.



FIGURE 4.7: L'utilisation d'Homographie pour trouver tous les corners de l'échiquier

## Deuxième Exemple

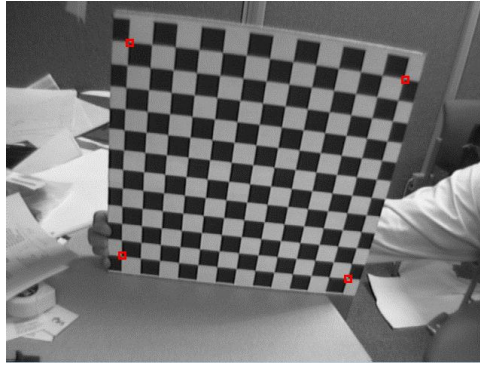


FIGURE 4.8: Sélection des quatre coins qui correspondent aux axes et l'origine

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$m = \begin{bmatrix} 528.8588 & 454.0497 & 154.4934 & 163.6237 \\ 105.2864 & 369.8911 & 340.3457 & 56.9960 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$$

$$H_{PSO-Based} = \begin{bmatrix} 25.3399 & -372.2964 & 528.8588 \\ 346.1909 & -50.7501 & 105.2864 \\ 0.2205 & -0.0431 & 1.000 \end{bmatrix}$$

TABLE 4.7: Résultats de l'expérimentation 2 (Temps en millisecondes)

Nombre de Particules	10	20	30	50
Temps d'exécution	59	68	88	104
Meilleure fitness	$4.62 \times 10^{-4}$	$3.56 \times 10^{-4}$	$8.46 \times 10^{-5}$	$3.00 \times 10^{-5}$

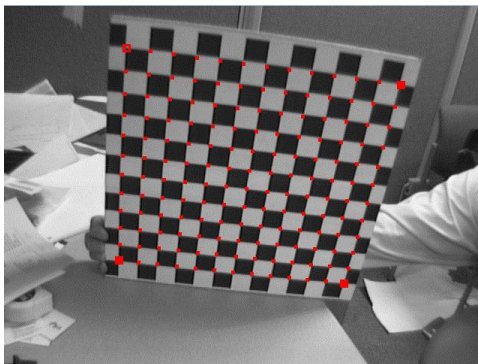


FIGURE 4.9: L'utilisation d'Homographie pour trouver tous les corners de l'échiquier

Comme il est affiché dans les tableaux 4.6,4.7, à condition qu'un nombre élevé de particules est considéré, mieux sera la meilleure fitness de l'essaim, malheureusement, on perd un peu de gain dans le temps d'exécution, ce qui va être encore considérable si on traite un grand set de données. Les tableaux 4.7,4.6 montrent le résultat en utilisant seulement 4 points dans l'ensemble de données, avec une différence de 20 à 40 millisecondes entre 20 et 50 particules. Considérant cet effet, le choix intelligent est d'utiliser peu de particules (15-20) pour traiter un grand ensemble de données, et doubler ce nombre pour les petits ensembles.

### 4.6.3 Application sur la base de données INRIA

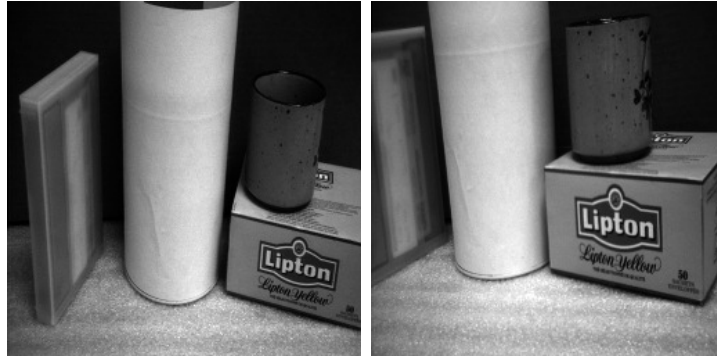
On a utilisé l'applet de Mr Krystian Mikolajczyk pour la stéréovision, et la base de données d'INRIA pour tester notre approche, l'applet est disponible suivant ce lien <sup>1</sup>, et la base de données peut être téléchargée ici <sup>2</sup>. Les figures 4.10, 4.11, 4.12, 4.13 montrent la supériorité de l'Homographie de la PSO par rapport à celle de l'applet. La différence entre les résultats est facilement distinguée : On remarque que les croix vertes sont plus proches des croix bleues que les rouges, ces résultats sont encore plus clairs dans les courbes (d) sur les figures 4.10, 4.11, 4.12, 4.13. La comparaison est basée sur l'efficacité et la précision car le temps d'exécution est pratiquement le même. L'erreur est calculée à partir de la distance cartésienne entre les points choisis et ceux projetés par l'Homographie :

$$error = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad (4.7)$$

---

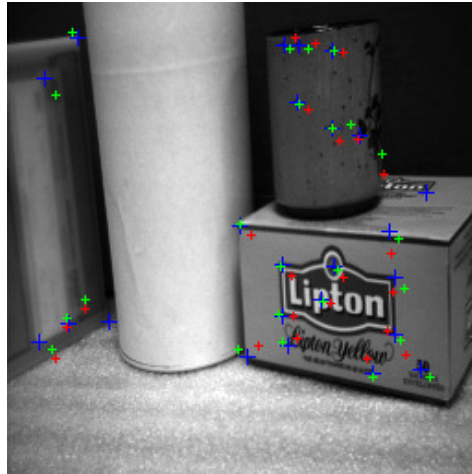
1. <http://lear.inrialpes.fr/people/mikolajczyk/>

2. <http://lear.inrialpes.fr/people/mikolajczyk/Database/rotzoom.html>

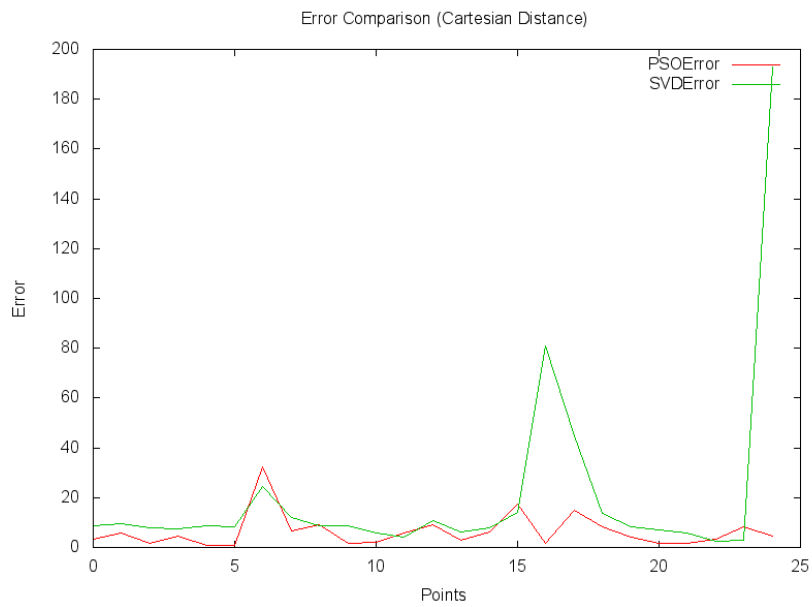


(a) Left pose

(b) Right pose



(c) Résultats : Les croix bleues sont les points choisis, les verts sont la projection en utilisant l'Homographie de la PSO, les rouges sont ceux de l'applet de Mikolajczyk

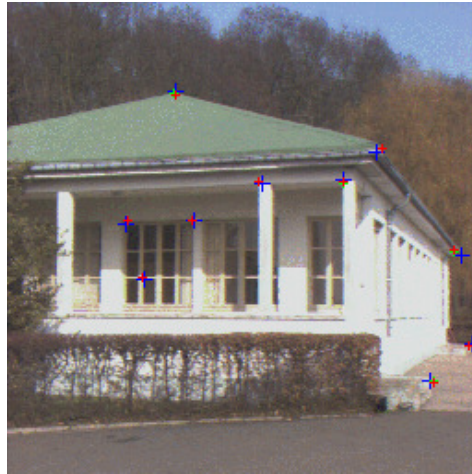


(d) Représentation des erreurs de la PSO et SVD dans l'image Lipton

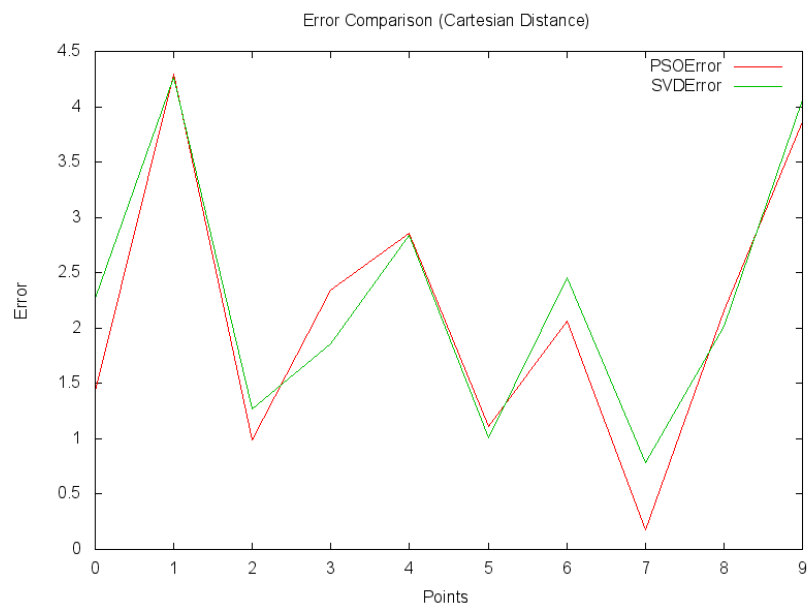


(a) Left pose

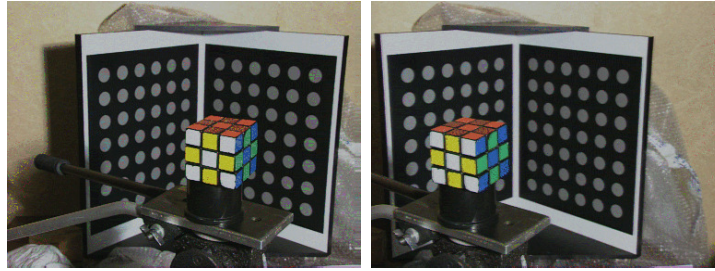
(b) Right pose



(c) Résultats : Les croix bleues sont les points choisis, les verts sont la projection en utilisant l'Homographie de la PSO, les rouges sont ceux de l'applet de Mikolajczyk

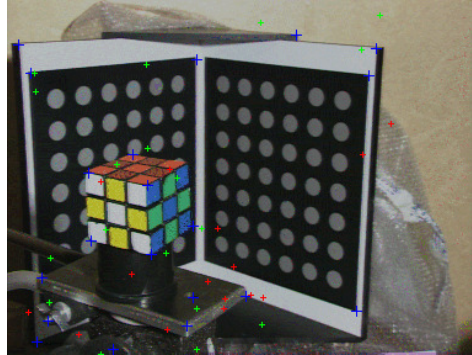


(d) Représentation des erreurs de la PSO et SVD dans l'image INRIA

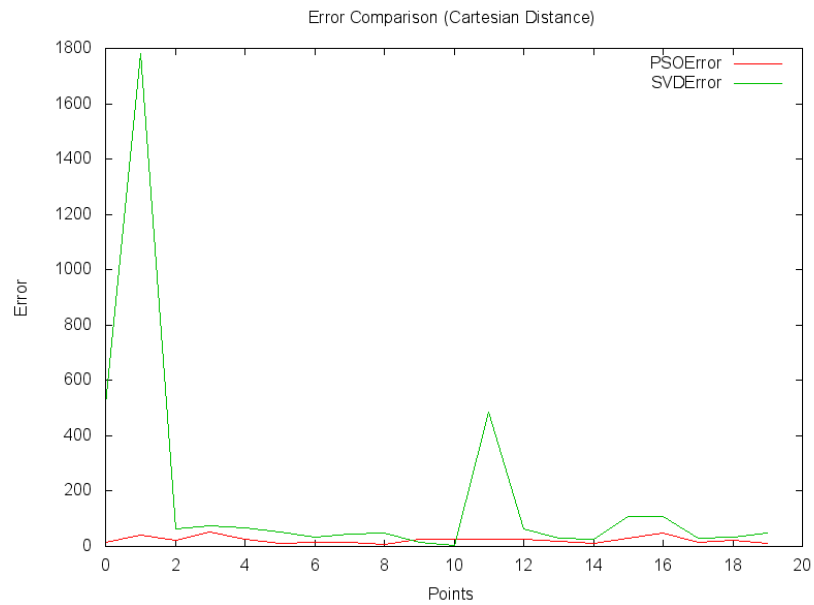


(a) Left pose

(b) Right pose



(c) Résultats : Les croix bleues sont les points choisis, les verts sont la projection en utilisant l'Homographie de la PSO, les rouges sont ceux de l'applet de Mikolajczyk



(d) Représentation des erreurs de la PSO et SVD dans l'image Rubik

FIGURE 4.12: Comparaison de la projection des erreurs dans l'image Rubik

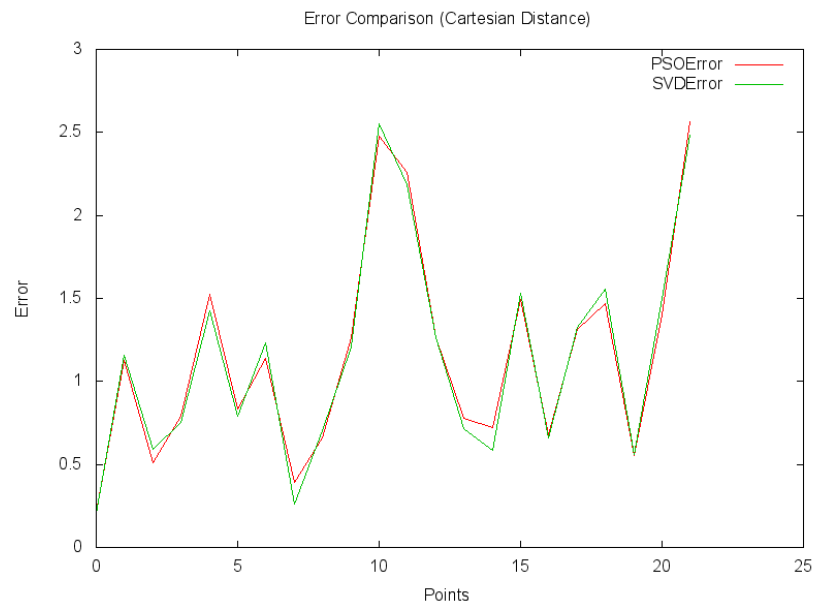


(a) Left pose

(b) Right pose



(c) Résultats : Les croix bleues sont les points choisis, les verts sont la projection en utilisant l'Homographie de la PSO, les rouges sont ceux de l'applet de Mikolajczyk



(d) Représentation des erreurs de la PSO et SVD dans l'image Boat

FIGURE 4.13: Comparaison de la projection des erreurs dans l'image Boat

#### 4.6.4 Discussion

Les expérimentations ont donné quelques points positives qu'on peut discuter comme suivant :

- Les résultats de la PSO s'il ne sont pas meilleure, sont égaux à ceux du SVD.
- Le temps d'exécution est presque identique.
- La simplicité d'implémentation de l'algorithme PSO.
- La flexibilité de l'algorithme PSO qui peut être utilisé pour améliorer la vitesse de convergence.

### 4.7 Conclusion

L'importance de l'Homographie dans le processus de la reconstruction 3D a nécessité une étude approfondie de son calcul, ce qui explique cette dernière partie. On a vu comment était plus simple d'utiliser la PSO pour résoudre ce problème non linéaire, et comment l'approche proposée était plus performante par rapport à la précision et facilité d'implémentation. L'approche proposée est aussi performante dans le cas d'un grand ensemble de données à traiter, et cela sans nécessiter la modification de ces derniers ou plus précisément la normalisation. Après avoir expliqué le coeur de ce travail, on enchaîne avec la calibration des caméras dans la partie suivante.

# Chapitre 5

## Calibration de Caméra

### Sommaire

---

<b>5.1</b>	<b>Définition . . . . .</b>	<b>85</b>
5.1.1	Calibration Photogrammétrique . . . . .	85
5.1.2	Auto-Calibrage . . . . .	86
<b>5.2</b>	<b>Modèle de caméra Sténopé . . . . .</b>	<b>87</b>
<b>5.3</b>	<b>Équations de base . . . . .</b>	<b>88</b>
5.3.1	Notation . . . . .	88
5.3.2	Homographie entre le plan du modèle et son image . . . . .	89
5.3.3	Contraintes sur les paramètres intrinsèques . . . . .	89
<b>5.4</b>	<b>Résolution du calibrage de caméra . . . . .</b>	<b>90</b>
5.4.1	Solution de forme fermée . . . . .	90
5.4.2	Extraction des paramètres intrinsèques depuis la matrice B . . . . .	92
5.4.3	Estimation du maximum de vraisemblance . . . . .	92
5.4.4	Traitement de la distorsion radiale . . . . .	93
<b>5.5</b>	<b>Expérimentations . . . . .</b>	<b>94</b>

---

Un des paramètres que nous avons employé jusque là, est le paramètres de la caméra  $P$  ou  $K$ . Cette matrice qu'on a utilisé pour calculer la matrice essentielle dans la StéréoVision, désigne les paramètres intrinsèques de la caméra utilisé pour avoir les images sujets du traitement. Dans ce chapitre, on va définir et expliquer le processus qui nous permet d'obtenir cette matrice. Ce processus est appelé *calibration de caméra* ou *étalonnage de caméra* ou *calibrage de caméra*, tous ces termes désigne le processus inverse de l'obtention de l'image, en d'autres termes, la projection inverse de l'image d'un espace 2D vers l'espace 3D. Plusieurs méthodes ont été proposés par les chercheurs qu'on trouve dans la littérature, mais on a choisi de travailler avec la méthode de Zhang proposé dans [104]. Ce choix est dû à la facilité de cette approche et à son efficacité surtout dans sa grande flexibilité. Cette dernière se matérialise dans le fait qu'un simple échiquier soit suffisent pour pouvoir calibrer la caméra. Plusieurs résultats seront présentés à la fin de ce chapitre, et seront comparés avec les résultats produits par l'approche présentée dans le chapitre précédant i.e. l'Homographie basée sur la PSO.

## 5.1 Définition

### 5.1.1 Calibration Photogrammétrique

Le calibrage de la caméra est effectué en observant un objet de calibrage, qui a une géométrie bien connue et précise dans l'espace 3D [105]. Cet objet se consiste généralement de deux ou trois plans orthogonales les uns sur les autres. Parfois, un plan qui subit une translation connue et précise peut aussi être utilisé. . Suivant les dimensions des objets de calibrage, on peut classifier ces techniques en trois catégories :

#### **Calibrage en utilisant un objet de référence 3D**

Le calibrage de caméra est effectué en observant un objet de calibrage dont on connaît avec précision sa géométrie 3D dans l'espace. Cet objet se consiste généralement de deux ou trois plans orthogonales les uns sur les autres. Des fois, un plan qui subit une translation bien précise est aussi utilisé [97], qui fournit l'équivalent des points de références 3D. Ces approches nécessite un appareil de calibrage trop cher, et élaborer une installation dédié au processus.

## Calibrage basé sur un plan 2D

Les techniques de cette catégorie nécessitent d'observer un objet planaire montré à plusieurs différentes orientations [94],[104]. Différente de la technique de Tsai [97], la connaissance du mouvement du plan n'est pas nécessaire. Parce que n'importe qui peut avoir ce motif de calibration, l'installation est facile pour le calibrage de caméra.

## Calibrage basé sur une ligne 1D

Les objets de calibration utilisés dans cette catégorie sont composés d'un ensemble de points col-linéaire [105]. Une caméra peut être calibrée en observant une ligne qui bouge autour d'un point fixe, comme un fil de boule suspendu au plafond.

### 5.1.2 Auto-Calibrage

Les techniques dans cette catégorie n'utilisent aucun objet de calibration. Juste en bougeant la caméra dans une scène statique, la rigidité de la scène donne en générale deux contraintes sur les paramètres internes de la caméra d'un seul déplacement de la caméra en utilisant seulement les informations de l'image [105]. Alors, si les images ont été prises par la même caméra avec des paramètres internes fixés, les correspondances entre trois images sont suffisantes pour récupérer les deux paramètres internes et externes, ce qui va permettre de reconstruire les structures 3D. Malgré que cette approche est tellement flexible, mais elle n'est pas encore mature. Car il y a tellement de paramètres à estimer, qu'on ne peut pas toujours avoir des résultats fiables.

D'autres techniques existent : points de fuite pour les directions orthogonales [17],[66], et calibration à partir d'une pure rotation [52],[92]. Avant de continuer, on doit éclaircir un point : aucune technique de calibration est meilleure que les autres. Ça dépend vraiment de la situation que l'utilisateur a besoin de faire avec. Les recommandations suivantes ont été faites par Zengyou Zhang dans [106] :

- Calibrage avec appareil vs. Auto-calibrage. Dès que possible, si on peut pré-calibrer une caméra, on doit le faire avec un appareil de calibration. L'auto-calibrage ne peut pas généralement aboutir à un résultat précis comparable à celui du pré-calibrage car il a besoin d'estimer un grand nombre de paramètres, ce qui va donner un problème mathématique plus difficile. Quand le pré-calibrage est

impossible (e.g. reconstruction d'une scène à partir d'un film ancien), l'auto-calibrage est la seule option.

- Auto-calibrage partiel vs. complet. L'auto-calibrage partiel fait référence au cas où seulement un sous-ensemble de paramètres intrinsèques de la caméra va être calibré. En suivant le même conseil du premier cas, dès que possible, l'auto-calibrage partiel est recommandé car le nombre de paramètres à estimer est plus petit. On prends l'exemple de la reconstruction 3D avec une caméra qui a une distance focale variable. Il est préférable de pré-calibrer le rapport d'aspect et l'asymétrie du pixel.
- Calibrage avec un appareil 3D vs. un appareil 2D. Une meilleure précision est obtenue généralement en utilisant un appareil 3D, donc ça doit être utiliser si la précision est indispensable et quand c'est abordable d'utiliser un tel appareil. D'après le feedback que Zenghyou Zhang a reçu des chercheurs de vision par ordinateur autour du monde dans les dernières années, le calibrage avec un appareil 2D semble le meilleur choix dans la plus part des situations en raison de la facilité de l'utilisation et une bonne précision.
- Calibrage avec un appareil 1D. Cette technique est relativement nouvelle, et c'est difficile pour le moment de prédire si elle va être populaire ou non. Elle va cependant être utile spécialement pour calibrer un réseau de caméras. Pour calibrer la géométrie relative entre plusieurs caméras aussi bien que ses paramètres intrinsèques, il est nécessaire pour tous les caméras impliqués d'observer simultanément un nombre de points. C'est vraiment difficile de réaliser ça avec un appareil de calibrage 3D ou 2D si une caméra est montée en face de la chambre et l'autre en derrière. Ce n'est pas un problème des objets 1D. On peut par exemple utiliser un fil des boules suspendues au plafond.

## 5.2 Modèle de caméra Sténopé

Un point 2D est dénoté par  $m = [u, v]^T$ . Un point 3D est dénoté par  $M = [X, Y, Z]^T$ . on utilise  $\tilde{x}$  pour dénoter le vecteur augmenté en ajoutant 1 comme dernier élément :  $\tilde{m} = [u, v, 1]^T$  et  $\tilde{M} = [X, Y, Z, 1]^T$ . Une caméra est modelée par le modèle habituel sténopé (Fig. 5.1) : L'image du point 3D  $M$ , dénoté par  $m$  est formé par un rayon optique depuis  $M$  passant par le centre optique  $C$  et fais une intersection avec le plan de l'image. Les trois points  $M, m$  et  $C$  sont collinéaire. Dans la figure Fig. 5.1, pour des raisons d'illustration, le plan d'image est positionné entre le point de la scène

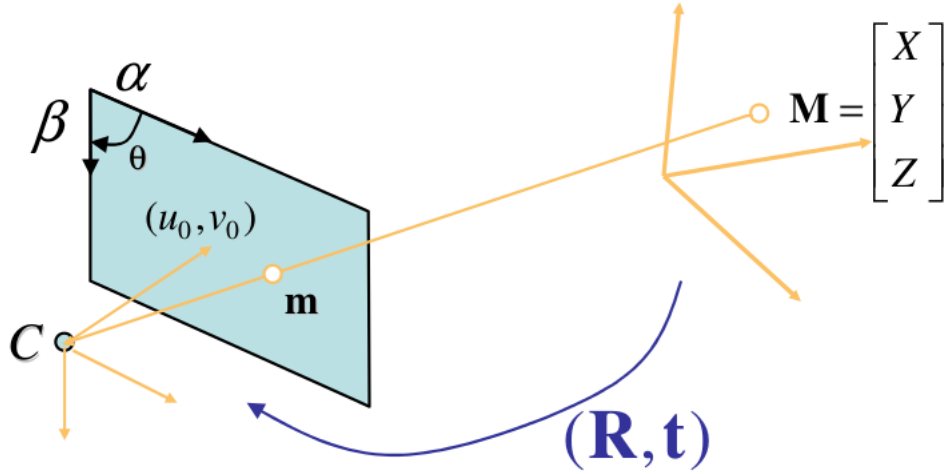


FIGURE 5.1: Modèle de caméra Sténopé

et le centre optique, qui est mathématiquement équivalent à l'installation physique où le plan de l'image est sur l'autre côté par rapport au centre optique.

## 5.3 Équations de base

On examine les contraintes sur les paramètres intrinsèques de la caméra qui proviennent en observant un seul plan. On commence par les notations dans ce manuscrit.

### 5.3.1 Notation

Un point 2D est dénoté par  $m = [u, v]^T$ . Un point 3D est dénoté par  $M = [X, Y, Z]^T$ . On utilise  $\tilde{x}$  pour dénoté le vecteur augmenté en ajoutant 1 comme le dernier élément :  $\tilde{m} = [u, v, 1]^T$  et  $\tilde{M} = [X, Y, Z, 1]^T$ . Une caméra est modélisée par le modèle sténopé : la relation entre un point 3D  $M$  et sa projection image  $m$  est donné par :

$$s\tilde{m} = A[R \ t]\tilde{M} \quad (5.1)$$

où  $s$  est un facteur d'échelle arbitraire,  $(R, t)$ , appelé paramètres extrinsèques, est la rotation et translation qui relie le système de coordonnées de la scène au système de coordonnées de la caméra, et  $A$ , la matrice des paramètres intrinsèques, qui est donnée

par :

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

avec  $(u_0, v_0)$  les coordonnées du point principale,  $\alpha$  et  $\beta$  les facteurs d'échelle dans les axes  $u$  et  $v$  dans l'image, et  $\gamma$  le paramètre qui décrit la distorsion des axes des deux images. On utilise l'abréviation  $A^{-T}$  pour  $(A^{-1})^T$  ou  $(A^T)^{-1}$ .

### 5.3.2 Homographie entre le plan du modèle et son image

Sans se perdre dans les généralités, on assume que le plan du modèle est sur  $Z = 0$  du système de coordonnées de la scène. On dénote le  $i^{\text{ème}}$  colonne de la matrice de rotation  $R$  par  $r_i$ . De l'équation 5.1, on a :

$$s = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Pour faciliter la lisibilité, on utilise toujours  $M$  pour dénoter un point du plan du modèle, mais  $M = [X, Y]^T$  car  $Z$  est toujours égale à 0. A son tour,  $\tilde{M} = [X, Y, 1]^T$ . Alors, un point du modèle  $M$  et son image  $m$  sont reliés par une Homographie  $H$  :

$$s\tilde{m} = H\tilde{M} \quad \text{avec} \quad H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \quad (5.2)$$

### 5.3.3 Contraintes sur les paramètres intrinsèques

Ayant une image du plan du modèle, une Homographie peut être estimée. On la dénote par  $H = [h_1 \ h_2 \ h_3]$ . De l'équation 5.2 on a :

$$H = [h_1 \ h_2 \ h_3] = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

où  $\lambda$  est un scalaire arbitraire. Utilisant la connaissance que  $r_1$  et  $r_2$  sont orthogonaux, on a :

$$h_1^T A^{-T} A^{-1} h_2 = 0 \quad (5.3)$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \quad (5.4)$$

Ces deux là sont les contraintes de base sur les paramètres intrinsèques, ayant une seule Homographie. Car une Homographie a 8 degrés de libertés et il y a 6 paramètres intrinsèques (3 pour la rotation et 3 pour la translation), on peut seulement obtenir 2 contraintes sur les paramètres intrinsèques. Remarque que  $A^{-T} A^{-1}$  décrit en faite l'image de la conique absolue.

## 5.4 Résolution du calibrage de caméra

Cette section fournit les détails comment résoudre efficacement le problème du calibrage de caméra. On commence par une solution analytique, suivie d'une technique d'optimisation non-linéaire basée sur le maximum d'un critère de vraisemblance. En fin, on prends en considération les distorsions de la lentille, en utilisant deux solutions analytique et non-linéaire.

### 5.4.1 Solution de forme fermée

On a :

$$\begin{aligned} B = A^{-T} A^{-1} &\equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \end{aligned} \quad (5.5)$$

Remarque que  $B$  est symétrique, définie par un vecteur 6D :

$$b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \quad (5.6)$$

Soit l' $i^{\text{ème}}$  vecteur colonne de  $H$  :  $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$ . Alors on a :

$$h_i^T B h_j = v_{ij}^T b \quad (5.7)$$

Avec :

$$v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$$

Donc, les deux contraintes fondamentales 5.3,5.4, d'une quelconque Homographie peuvent être écrites comme 2 équations homogènes dans  $b$  :

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (5.8)$$

Si  $n$  images du plan du modèle sont observés, en empilant  $n$  équations comme 5.8 on aura :

$$Vb = 0 \quad (5.9)$$

où  $V$  est une matrice de  $2n \times 6$ . Si  $n \geq 3$ , on va avoir une solution unique  $b$  défini à facteur d'échelle. Si  $n = 2$ , on peut imposer une contrainte de non-distorsion  $\gamma = 0$ , i.e.,  $[0, 1, 0, 0, 0, 0]b = 0$ , qui va être ajouté comme équation additionnelle à 5.9. (Si  $n = 1$ , on peut résoudre seulement deux paramètres intrinsèques de la caméra, e.g.,  $\alpha$  et  $\beta$ , en assumant que  $u_0$  et  $v_0$  sont connus (e.g., au centre de l'image) et  $\gamma = 0$ , et ça a été effectivement réalisé dans [89] pour l'estimation de la pose de la tête en se basant sur le fait que les yeux et la bouche sont quasiment coplanaires). La solution de l'équation 5.9 est bien connue comme le vecteur singulier de  $V^T V$  associé à la valeur singulière la plus petite (en d'autres termes, le vecteur singulier de la droite de  $V$  associé à la valeur singulière la plus petite). Une fois  $b$  est estimé, on peut calculer la matrice des paramètres intrinsèques  $A$  de la caméra.

Une fois  $A$  est connue, les paramètres extrinsèques de chaque image peut alors facilement être calculé. De l'équation 5.2, on a :

$$r_1 = \lambda A^{-1} h_1 r_2 = \lambda A^{-1} h_2 r_3 = r_1 \times r_2 t = \lambda A^{-1} h_3$$

Avec  $\lambda = 1 / \| A^{-1} h_1 \| = 1 / \| A^{-1} h_2 \|$ . Dans l'évidence, à cause du bruit présent dans les données, la matrice calculée  $R = [r_1, r_2, r_3]$  ne satisfais toujours pas les propriétés d'une matrice de rotation.

## 5.4.2 Extraction des paramètres intrinsèques depuis la matrice B

La matrice B décrite ci-dessus est estimée à facteur d'échelle, i.e.  $B = \lambda A^{-T} A$  avec  $\lambda$  un facteur arbitraire. Sans difficultés, on peut extraire les paramètres intrinsèques de la matrice B.

$$\begin{aligned}
 v_0 &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\
 \lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\
 \alpha &= \sqrt{\lambda / B_{11}} \\
 \beta &= \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}} \\
 \gamma &= -B_{12}\alpha^2\beta / \lambda \\
 u_0 &= \gamma v_0 / \beta - B_{13}\alpha^2 / \lambda
 \end{aligned}$$

## 5.4.3 Estimation du maximum de vraisemblance

La solution en dessus est obtenue par la minimisation d'une distance algébrique qui n'est pas physiquement significative. On peut l'affiner à travers une inférence de maximum de vraisemblance. On a  $n$  images du plan du modèle et  $m$  points sur ce plan. Assumant que les points d'image sont corrompus par un bruit indépendant et distribué identiquement. L'estimation du maximum de vraisemblance peut être obtenue en minimisant l'équation suivante :

$$\sum_{i=1}^n \sum_{j=1}^m \| m_{ij} - \hat{m}(A, R_i, t_i, M_j) \|^2 \quad (5.10)$$

où  $\hat{m}(A, R_i, t_i, M_j)$  est la projection du point  $M_j$  dans l'image  $i$ , selon l'équation 5.2. Une rotation  $R$  est paramétrée par un vecteur de 3 paramètres, dénoté par  $r$ , qui est parallèle aux axes de rotation et sa magnitude est égale à l'angle de rotation.  $R$  et  $r$  sont reliées par la formule de Rodrigues [32]. La minimisation de l'équation 5.10 est un problème non-linéaire, qui peut être résolu avec l'algorithme de Levenberg-Marquardt comme implémenté dans Minipack [79]. ça nécessite une estimation initiale de la matrice  $A, R_i, t_i | i = 1..n$  qui obtenue en utilisant la technique décrit dans la section précédente.

#### 5.4.4 Traitement de la distorsion radiale

Jusqu'à maintenant, on n'a pas considéré la distorsion de la lentille de la caméra. Cependant, une caméra de bureau présente habituellement une distorsion de lentille, et spécialement une distorsion radiale. Dans cette section, on considère seulement les deux premiers termes de la distorsion radiale. On se basant sur les rapports dans la littérature [15],[100],[97], il est probable que la fonction de distorsion est totalement dominée par les composantes radiales, et spécialement par le premier terme. Il a aussi été trouvé que n'importe quel autres modèles mieux élaborés ne va pas seulement ne pas aider, mais aussi peut causer des instabilités numériques [100],[97]. Soit  $(u, v)$  les coordonnées idéales d'un pixel (pas de distorsion), et  $(\check{u}, \check{v})$  les coordonnées réelles correspondantes observées. Les points idéales sont la projection des points du modèle selon le modèle sténopé. La même chose pour  $(x, y)$  et  $(\check{x}, \check{y})$  sont les coordonnées idéales (pas de distorsion) et réelles normalisés (distordu). On a [15],[100] :

$$\begin{aligned}\check{x} &= x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \check{y} &= y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]\end{aligned}\tag{5.11}$$

Où  $k_1$  et  $k_2$  sont les coefficients de la distorsion radiale. Le centre de la distorsion radiale est le même que le point principal. A partir de  $\check{u} = u_0 + \alpha\check{x} + \gamma\check{y}$  et  $\check{v} = v_0 + \beta\check{y}$  et en assumant que  $\gamma = 0$  on a :

$$\check{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]\tag{5.12}$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]\tag{5.13}$$

**Estimation de la distorsion radiale par alternance** Comme la distorsion radiale est censé être petite, on peut espérer d'estimer les cinq paramètres intrinsèques raisonnablement bien en ignorant la distorsion. Une stratégie est alors d'estimer  $k_1$  et  $k_2$  après avoir estimé les autres paramètres, qui va nous donner les coordonnées idéales des pixels  $(u, v)$ . Alors, de l'équation 5.12,5.13, on a deux équations pour chaque point de chaque image :

$$\begin{bmatrix} (u - u_0)(x^2 + y^2) & (u - u_0)(x^2 + y^2)^2 \\ (v - v_0)(x^2 + y^2) & (v - v_0)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix}$$

Ayant  $m$  points dans  $n$  images, on raffine l'estimation des autres paramètres en trouvant une solution à l'équation 5.10 avec  $\hat{m}(A, R_i, t_i, M_j)$  remplacé par les équations 5.12,5.13. On peut alterner ces deux procédures jusqu'à la convergence.

**Estimation complète du maximum de vraisemblance** Une extension de l'équation 5.10 est d'estimer l'ensemble total des paramètres en minimisant l'équation suivante :

$$\sum_{i=1}^n \sum_{j=1}^m \| m_{ij} - \tilde{m}(A, k_1, k_2, R_i, t_i, M_j) \|^2 \quad (5.14)$$

Où  $\tilde{m}(A, k_1, k_2, R_i, t_i, M_j)$  est la projection du point  $M_j$  dans l'image  $i$  par l'équation 5.2, suivie par une distorsion par les équations 5.12,5.13. Ceci est un problème de minimisation non linéaire, qui est résolu avec l'algorithme de Levenberg-Marquardt comme implémenté dans Minipack[79]. Une rotation est encore paramétrée par un vecteur  $r$  de taille 3 comme dans la section 5.4.3. Une estimation initiale de  $A$  et  $R_i, t_i | i = 1..n$  peut être trouvée en utilisant la technique décrite dans la section 5.4.1 ou 5.4.3. Une estimation initiale de  $k_1$  et  $k_2$  peuvent être obtenues avec la techniques décrite dans le dernier paragraphe, ou simplement en les donnant la valeur 0.

## 5.5 Expérimentations

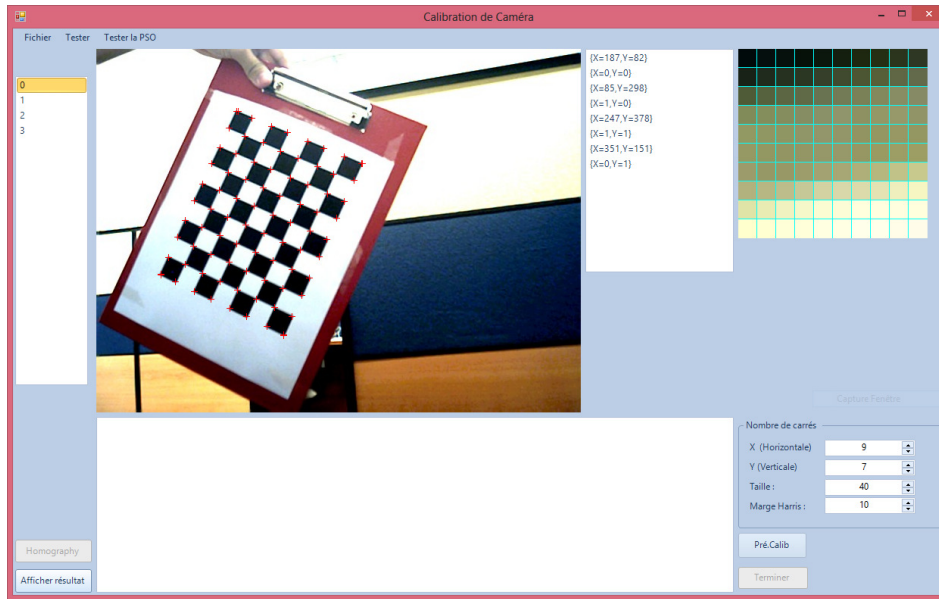
Dans cette partie, nous allons présenter la partie calibration du logiciel que nous avons implémenté, cette partie sert à calculer l'Homographie et calibrer la caméra utilisée à partir d'une série d'images prises d'un pattern dédié au processus i.e. un échiquier. Ce choix est due à la facilité d'avoir ce pattern et le taux élevé de réussite de l'opération. Nous rappelons que tous les résultats de cette partie ont été obtenus sur une machine équipée d'un processeur **intel core i7** quatre coeur et une mémoire de **8 Go**. Comme mentionné avant, l'application a été compilé sous **windows 8** en utilisant le langage de programmation **C#** sous l'environnement **Visual Studio 2012**.

La figure 5.2 montre l'interface dédié à l'étalonnage, cette dernière est composée d'une liste qui contient la série d'image de l'échiquier, une zone d'image qui sert à afficher ainsi qu'à choisir manuellement les quatre premiers coins de l'échiquier. Une autre zone est dédié au paramètres des algorithmes utilisés :

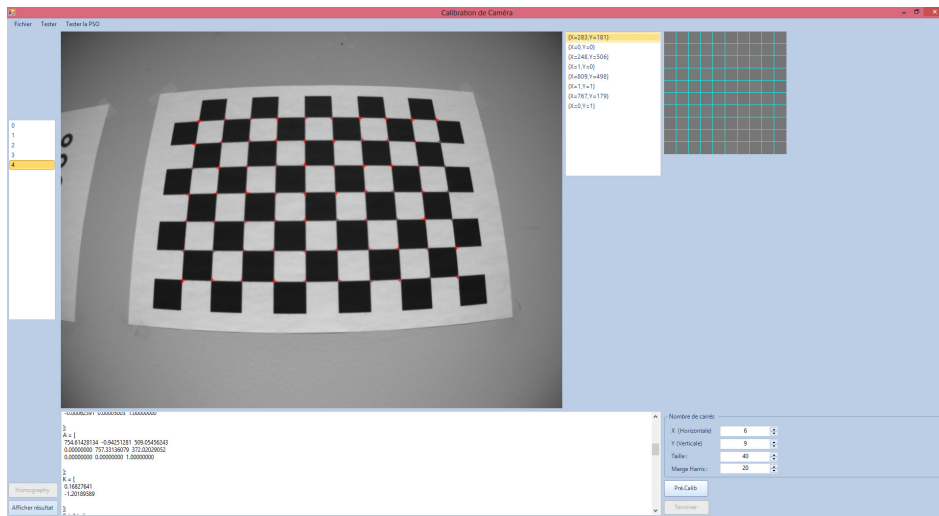
- X et Y : le nombre de carrés dans les deux direction  $\vec{X}, \vec{Y}$
- Taille : la taille des carrés qui composent l'échiquier.

- La zone à l'extrême droite : cette zone effectue *un Zoom* local de la région du curseur, ceci aide l'utilisateur à indiquer les quatre coins de l'échiquier.
- La zone en bas : les résultats seront affichés dans cette zone là, voir la figure 5.2(b).
- Les boutons Afficher résultats et Pré.Calib : l'utilisateur doit appuyer sur ces deux boutons par ordre pour, premièrement afficher le résultat de l'Homographie et projeter tous les points d'intersection des carrés du pattern, deuxièmement, la calibration est effectuée et l'affichage des résultats en bas et les sauvegarder pour une utilisation ultérieure.

**Remarque** Le choix initial des quatre coins de l'échiquier peut ne pas être précis, l'utilisateur peut indiquer les points de façon rapide car le programme va corriger ses erreurs et imprécisions par le détecteur de Harris [49].



(a) Première étape du processus : le marquage manuel des quatre coins de l'échiquier



(b) Deuxième étape du processus : l'affichage des résultats de l'étalonnage

FIGURE 5.2: Capture d'écran du logiciel élaboré pour la calibration des caméras

On a testé la méthode d'étalonnage de Zhang et on l'a comparé avec notre travail qui utilise l'optimisation par essaim de particule pour résoudre l'Homographie. Pour pouvoir valider l'efficacité des deux approches, on a choisi un pattern de calibration qui a subi une distorsion radiale importante. On dit qu'une approche est meilleure si en appliquant l'inverse de la distorsion (*Undistortion*), les lignes qui étaient arrondies par la loupe de la caméra seront plutôt droites.

La figure 5.3 montre les résultats appliqués à la première série d'images en utilisant

les sorties du logiciel de calibration suivants :

$$A = \begin{bmatrix} 938.89709440 & 11.30399201 & 341.85098246 \\ 0.00000000 & 933.86276998 & 239.99674738 \\ 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix} K = \begin{bmatrix} 0.04641685 \\ -0.93843019 \end{bmatrix}$$

Comme on peut le constater, la première série n'a pas subi une grande distorsion, les lignes apparaissent droites ce qui implique que la correction n'est pas significatif. Pour compléter les résultats, voilà les paramètres extrinsèques de la première série, ces paramètres seront utilisés pour l'estimation de pose et la reconstruction plus tard :

$$T(1) = \begin{bmatrix} -234.76193037 \\ -240.35584412 \\ 1417.11178930 \end{bmatrix} R(1) = \begin{bmatrix} 0.01132839 & 0.99992434 & -0.00479435 \\ 0.47535151 & -0.00116701 & 0.87979519 \\ -0.87972303 & 0.01224566 & 0.47532876 \end{bmatrix}$$

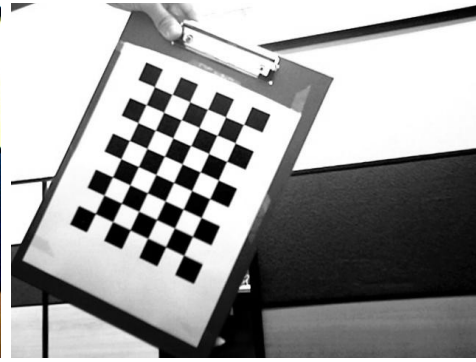
$$T(2) = \begin{bmatrix} -245.88399448 \\ -168.45737211 \\ 1010.55287372 \end{bmatrix} R(2) = \begin{bmatrix} 0.01132839 & 0.99992434 & -0.00479435 \\ 0.47535151 & -0.00116701 & 0.87979519 \\ -0.87972303 & 0.01224566 & 0.47532876 \end{bmatrix}$$

$$T(3) = \begin{bmatrix} 341.06968623 \\ -75.00233886 \\ 1486.70756800 \end{bmatrix} R(3) = \begin{bmatrix} 0.27761889 & 0.85003058 & 0.44763352 \\ 0.85694984 & -0.42972755 & 0.28455440 \\ 0.43424040 & 0.30460180 & -0.84773405 \end{bmatrix}$$

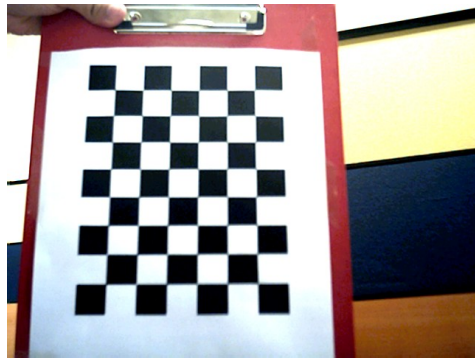
$$T(4) = \begin{bmatrix} -274.56453930 \\ -80.25476534 \\ 1407.35962938 \end{bmatrix} R(4) = \begin{bmatrix} 0.02408411 & 0.99962275 & 0.01320291 \\ 0.83942366 & -0.02739347 & 0.54278680 \\ 0.54294371 & -0.00198970 & -0.83976674 \end{bmatrix}$$



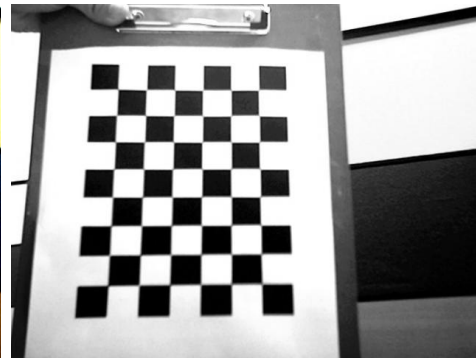
(a) Image initiale



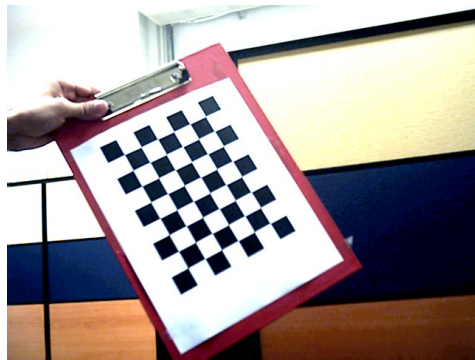
(b) Image corrigée



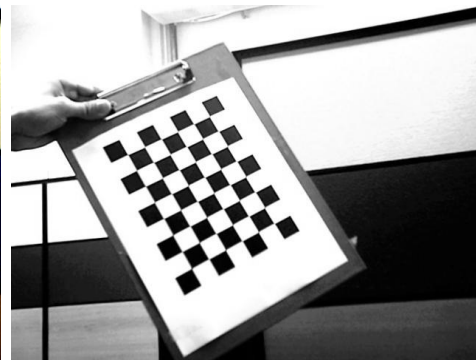
(c) Image initiale



(d) Image corrigée



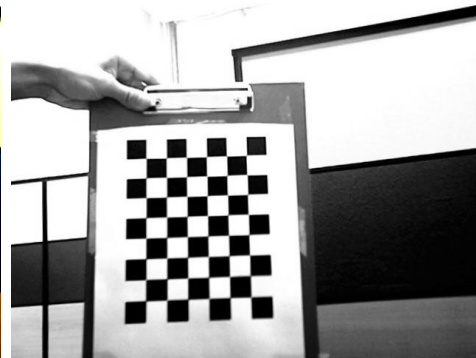
(e) Image initiale



(f) Image corrigée



(g) Image initiale



(h) Image corrigée

FIGURE 5.3: Résultats des approches comparés : exemple numéro 1

La deuxième série d'images présente une distorsion encore plus importante comme le montre la figure 5.4, on constate que le pattern d'étalonnage est grièvement déformé à cause de la distorsion de la loupe. Les résultats tirés du logiciel d'étalonnage le montre très bien. La série (b) de la figure 5.4 montre les résultats après la correction en utilisant les paramètres suivants :

$$A = \begin{bmatrix} 754.61428134 & -0.94251281 & 509.05456243 \\ 0.00000000 & 757.33136079 & 372.02029052 \\ 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix} K = \begin{bmatrix} 0.16827641 \\ -1.20189589 \end{bmatrix}$$

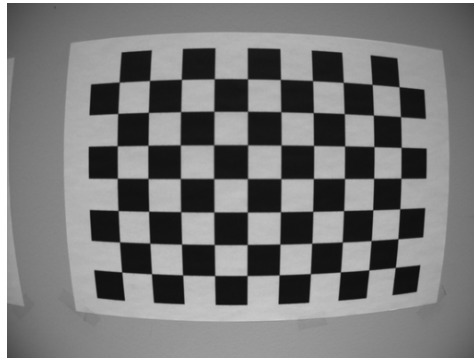
Comme pour la première série, voici les paramètres extrinsèques tirées de la deuxième série d'images :

$$T(1) = \begin{bmatrix} -150.83699011 \\ -115.24537377 \\ 414.23051651 \end{bmatrix} R(1) = \begin{bmatrix} 0.05122834 & 0.99716086 & -0.05518952 \\ 0.74377169 & -0.00121516 & 0.66843264 \\ -0.66646780 & 0.07529110 & 0.74172227 \end{bmatrix}$$

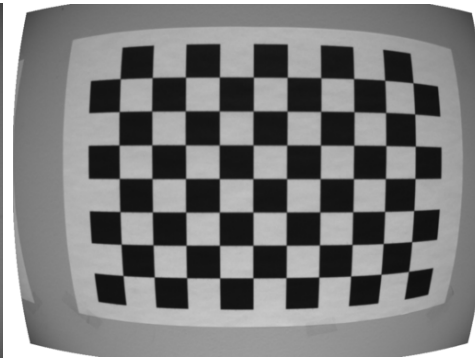
$$T(2) = \begin{bmatrix} -138.19256560 \\ -134.48241827 \\ 420.26752432 \end{bmatrix} R(2) = \begin{bmatrix} 0.22124960 & 0.94574294 & -0.23794726 \\ 0.75351902 & -0.01089569 & 0.65733582 \\ -0.61907811 & 0.32473307 & 0.71504596 \end{bmatrix}$$

$$T(3) = \begin{bmatrix} -216.00655364 \\ -122.72410716 \\ 518.68869558 \end{bmatrix} R(3) = \begin{bmatrix} 0.26572276 & 0.92738455 & -0.26334258 \\ 0.66646681 & 0.02066049 & 0.74524837 \\ 0.69657261 & -0.37353854 & -0.61258106 \end{bmatrix}$$

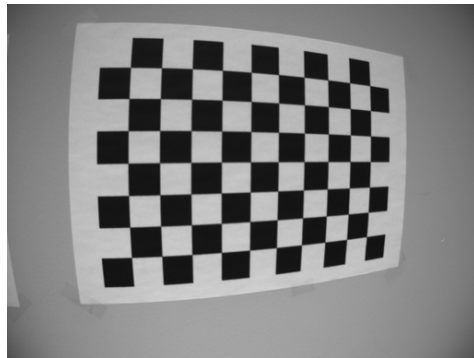
$$T(4) = \begin{bmatrix} -146.73394755 \\ -106.27178743 \\ 460.93577691 \end{bmatrix} R(4) = \begin{bmatrix} 0.01005767 & 0.99990866 & -0.00902850 \\ 0.88319846 & -0.00464919 & 0.46897641 \\ -0.46889160 & 0.01269077 & 0.88316455 \end{bmatrix}$$



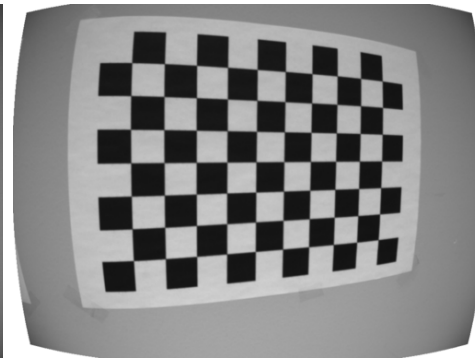
(a) Image initiale



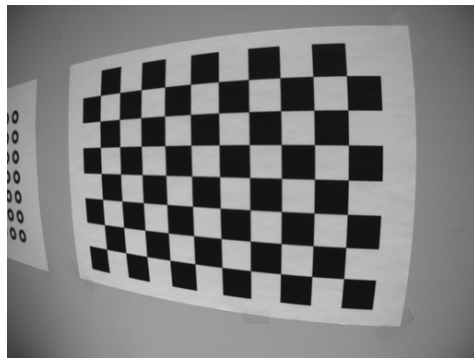
(b) Image corrigée



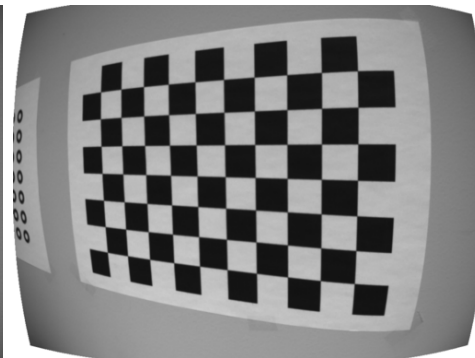
(c) Image initiale



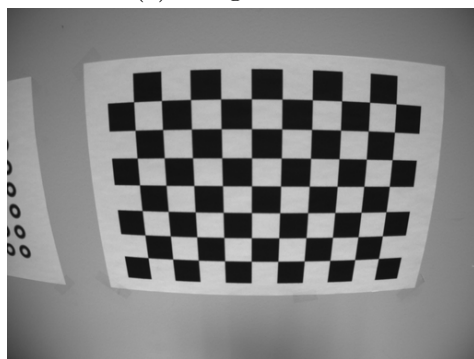
(d) Image corrigée



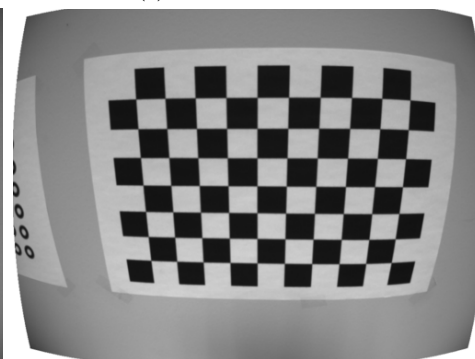
(e) Image initiale



(f) Image corrigée



(g) Image initiale



(h) Image corrigée

FIGURE 5.4: Résultats des approches comparés : exemple numéro 2

La troisième série d'images présente encore une distorsion importante comme le montre la figure 5.5, l'échiquier est déformé à cause de la distorsion de la caméra, et les résultats de la calibration le prouve. La série (b) de la figure 5.5 est la correction de la série (a) en utilisant les paramètres suivants :

$$A = \begin{bmatrix} 816.64336738 & 1.55444355 & 529.66323786 \\ 0.00000000 & 820.70376364 & 387.61539890 \\ 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix} K = \begin{bmatrix} 0.23272514 \\ -3.29828740 \end{bmatrix}$$

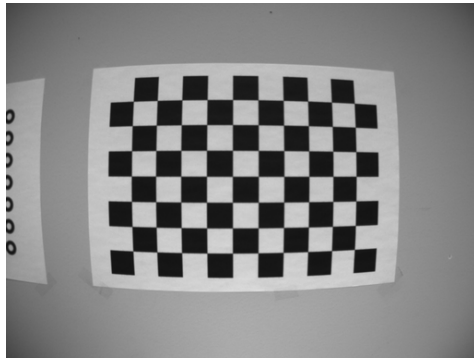
Comme pour les séries précédentes, voici les paramètres extrinsèques tirées de la troisième série d'images :

$$T(1) = \begin{bmatrix} -187.43254263 \\ -128.73636500 \\ 593.82656730 \end{bmatrix} R(1) = \begin{bmatrix} 0.02594026 & 0.99923541 & -0.02925236 \\ 0.71056398 & 0.00215225 & 0.70362930 \\ -0.70315427 & 0.03903800 & 0.70996486 \end{bmatrix}$$

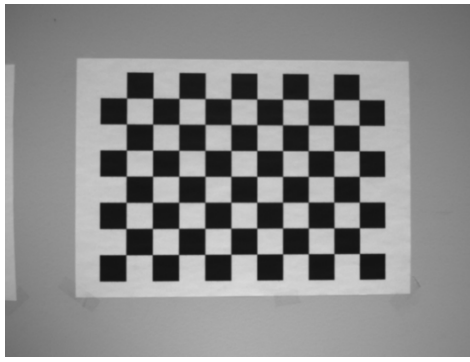
$$T(2) = \begin{bmatrix} -157.89665427 \\ -132.57513813 \\ 599.58337216 \end{bmatrix} R(2) = \begin{bmatrix} -0.24643148 & 0.92377632 & 0.29310210 \\ 0.73772469 & -0.01733095 & 0.67487918 \\ 0.62851715 & 0.38254013 & -0.67722171 \end{bmatrix}$$

$$T(3) = \begin{bmatrix} -180.66832248 \\ -130.84151300 \\ 735.43128397 \end{bmatrix} R(3) = \begin{bmatrix} 0.26596524 & 0.93293896 & -0.24266725 \\ 0.70939147 & -0.01897231 & 0.70455929 \\ 0.65270685 & -0.35953436 & -0.66686491 \end{bmatrix}$$

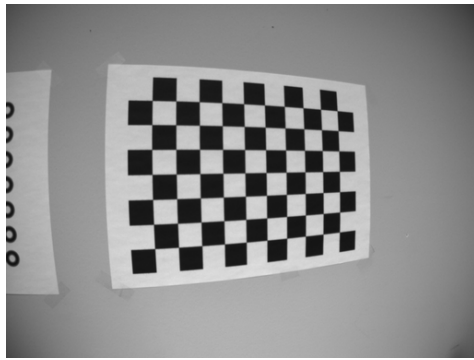
$$T(4) = \begin{bmatrix} -175.03474115 \\ -90.14228087 \\ 605.35573761 \end{bmatrix} R(4) = \begin{bmatrix} -0.03798125 & 0.99842097 & 0.04138825 \\ 0.41943779 & -0.02166433 & 0.90752554 \\ 0.90698918 & 0.05182875 & -0.41795265 \end{bmatrix}$$



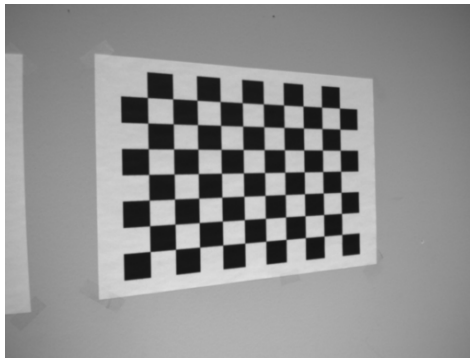
(a) Image initiale



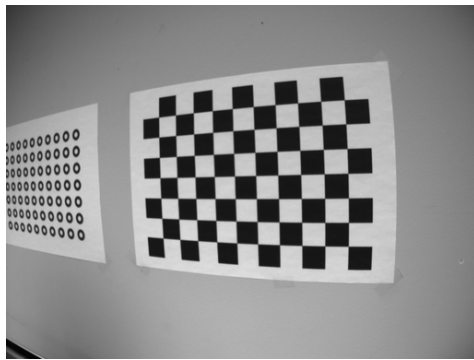
(b) Image corrigée



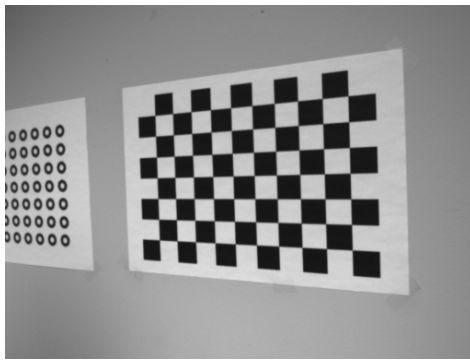
(c) Image initiale



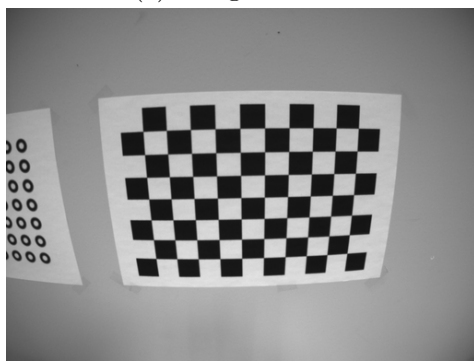
(d) Image corrigée



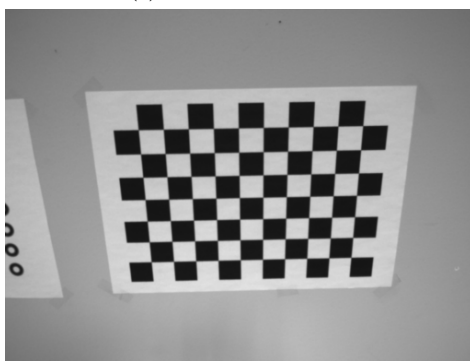
(e) Image initiale



(f) Image corrigée



(g) Image initiale



(h) Image corrigée

FIGURE 5.5: Résultats des approches comparés : exemple numéro 3

D'autres résultats sont présents dans la figure 5.6.

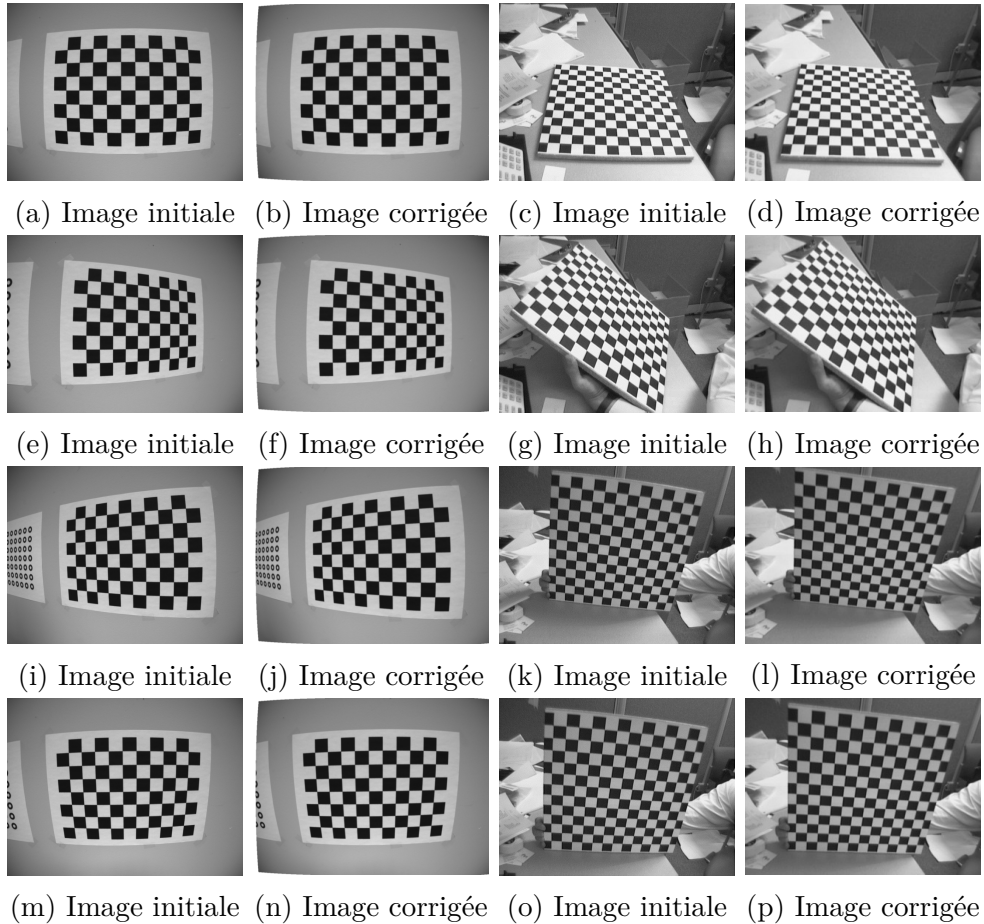


FIGURE 5.6: Résultats

Dans cette partie, nous avons expliquer comment calibrer une caméra à partir d'un pattern simple qui est l'échiquier. Cette méthode a été proposée par Zhang dans [104], est une parmi plusieurs autres méthodes de calibration. Zhang a proposé plusieurs autres approches dans [105], et les a classifiés selon le pattern utilisé. On a aussi comparé les résultats produit par l'approche de Zhang avec nos résultats qui permutent le calcul classique de l'Homographie basé sur les moindres carrés avec l'Homographie issue de l'optimisation par essaim de particules. Les expérimentations ont montré que la PSO donne des résultats acceptable et peut donc remplacer l'approche classique.

# Chapitre 6

## Reconstruction 3D

### Sommaire

---

<b>6.1 Méthodes actives . . . . .</b>	<b>105</b>
6.1.1 Utilisation du Scanner Laser pour la reconstruction . . . . .	105
<b>6.2 Méthodes passives . . . . .</b>	<b>107</b>
6.2.1 Méthodes monoculaires . . . . .	107
6.2.2 Approches multi-vues . . . . .	111
6.2.3 Reconstruction à partir du mouvement . . . . .	121
<b>6.3 Reconstruction 3D à partir des caméras non calibrées . .</b>	<b>123</b>
<b>6.4 Reconstruction à partir des caméras calibrées . . . . .</b>	<b>125</b>
6.4.1 Interprétation géométrique des quatre solutions . . . . .	126
<b>6.5 Théorème de la reconstruction projective . . . . .</b>	<b>127</b>
<b>6.6 Reconstruction stratifié . . . . .</b>	<b>129</b>
6.6.1 L'étape de reconstruction métrique . . . . .	133
6.6.2 Reconstruction métrique directe en utilisant $\omega$ . . . . .	137
6.6.3 La reconstruction directe - en utilisant la réalité de terrain .	138
6.6.4 Expérimentations . . . . .	139

---

La reconstruction 3D à partir des images était l'objectif fixé au départ. Après avoir détailler les différentes étapes qui procèdent la reconstruction, citons : la Stéréovision, le calcul de l'Homographie, et l'étalonnage des caméras ; on vient récolter le fruit des résultats obtenus jusque là. Effectivement, la reconstruction 3D à partir des images n'est autre que l'utilisation de la Stéréovision et ses résultats combinés aux paramètres des caméras déduit lors de la calibration. La seule étape qui reste pour avoir le modèle 3D de la scène sujet est le calcul de la carte de profondeur. Dans cette partie on finalisera notre travail avec les résultats obtenus de la projection 3D des images sujets.

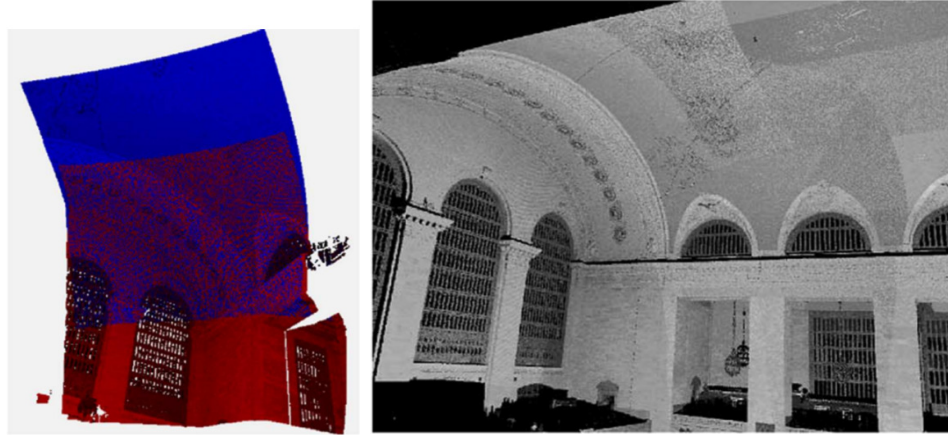
La problématique de la reconstruction géométrique est un domaine de recherche très vaste. Depuis le début des années 70, il a suscité beaucoup d'intérêt aussi bien dans les domaines de la recherche que pour des applications industrielles. Cette discipline est construite sur le développement d'analyse de données numériques provenant de diverses modalités : les approches par contacts physiques et les approches sans contact. Parmi les méthodes sans contact, les caméras numériques sont les capteurs les plus couramment utilisés. L'environnement est ainsi perçu à partir d'images numériques qu'il convient d'analyser. Plusieurs approches à partir de caméras sont apparues : les approches monoculaires qui estiment la géométrie de la scène à partir d'une seule vue ; les approches multi-vues construites sur l'utilisation de plusieurs vues prises simultanément ; ou encore les approches construites sur l'analyse d'une ou plusieurs vues prises à instants différents. Selon les applications visées, ces techniques œuvrent en lumière ambiante, nécessitent une lumière contrôlée qui peut parfois projeter un motif structuré. Dans la suite nous revenons sur les principales méthodes d'estimation de la géométrie à partir d'images[76].

## **6.1 Méthodes actives**

### **6.1.1 Utilisation du Scanner Laser pour la reconstruction**

Dans les dernières années, la technologie scanner laser a été émergé comme une approche utile et compétitive pour créer des reconstitutions en 3D[59]. Les avantages de base des méthodes qui utilisent cette technologie sont :

1. La vitesse.
2. L'exactitude.
3. La résolution de la reconstruction.



(a) Deux intervalles de scans (b) plusieurs intervalles de scans formant un modèle 3D

FIGURE 6.1: Résultats du recalage

De plus, le champ de vision des scanners permet la reconstruction d'objets, dont la taille varie de quelques centimètres à plusieurs mètres et existent dans une distance courte ou longue. Par conséquent, cette technologie est adaptée pour les scènes de grande taille, comme l'intérieur et l'extérieur des bâtiments et donc, elle est généralement acceptée par la communauté comme un support valable pour la documentation et la conservation des bâtiments historiques, des monuments et des sites archéologiques. Épigramme, la reconstruction d'une scène à l'aide d'un scanner nécessite la étapes suivantes :

1. Acquisition d'un nombre approprié de scans gamme de couleurs de manière à couvrir adéquatement la scène 3D.
2. Recalage des analyses d'intervalle dans le même système de coordonnées (fig. 6.1(a)).
3. Traitement des données pour le raffinement du modèle de surface 3D final (fig. 6.1 (b)).

L'étape de traitement comprend les étapes suivantes :

Élimination des informations redondantes (c'est à dire les données doubles créés à partir du recalage des régions qui se chevauchent) à partir du nuage de points enregistrés et le retrait du bruit .

Construction d'un modèle 3D qui comprend des facettes polygonales du nuage de points et ses données manquantes complètes (remplissage de trous).

La mise en place de l'équipement, afin d'acquérir les analyses d'intervalle est une tâche triviale, alors qu'il existe de nombreux logiciels commerciaux qui sont en mesure de

traiter des données de manière à affiner le modèle 3D final. Par conséquent, le problème principal lors de la procédure de reconstruction 3D réside dans le calcul automatique des transformations en trois dimensions qui alignent les ensembles de données de gamme afin d'extraire le modèle 3D complet [59]. Le recalage des nuages de points dans le même système de coordonnées suppose l'exploitation d'une méthode de recalage 3D-3D. Là, il y a beaucoup de méthodes relatives au recalage des données de plage. Certaines méthodes sont automatiques et s'appuient sur une procédure d'appariement automatisé des caractéristiques, tandis que d'autres nécessitent l'utilisation de marqueurs. Cette section présente les procédures les plus efficaces qui peuvent effectuer un recalage précis sans avoir besoin de marqueurs spéciaux. La procédure de recalage de scan de gamme peut être divisé en deux étapes : (1) le recalage initial qui fournit un bon deviner de départ de la transformation de l'alignement et (2) de recalage affine qui donne la transformation précise d'alignement.

## 6.2 Méthodes passives

### 6.2.1 Méthodes monoculaires

Les approches de reconstruction qui utilisent une seule caméra, s'appuient sur l'extraction de certaines caractéristiques de l'image afin de déterminer l'information de profondeur[76]. Cette information caractéristique peut être l'éclairément (Shape From Shading), la déformation de texture (Shape From Texture), la variation des paramètres de mise au point de l'objectif utilisé (Shape from Focus/Defocus), la détection de points de fuites [47] ou encore des approches par temps de vol de la lumière (Caméras à temps de vol). L'ensemble de ces approches utilisent certaines hypothèses fortes qui permettent de lever l'ambiguïté de profondeur, information en partie perdue lors du processus de formation de l'image.

#### Shape from shading

L'approche Shape From Shading estime la forme d'objets éclairés, à partir des variations graduelles de l'éclairément observé (voir figure 6.2). Ayant une image en niveau de gris, le but est de déterminer la position des sources de lumière, ainsi que la surface de l'objet pour chaque pixel de l'image. Même si le modèle d'éclairément est supposé lambertien, que la direction des sources de lumière est connue et que l'information de

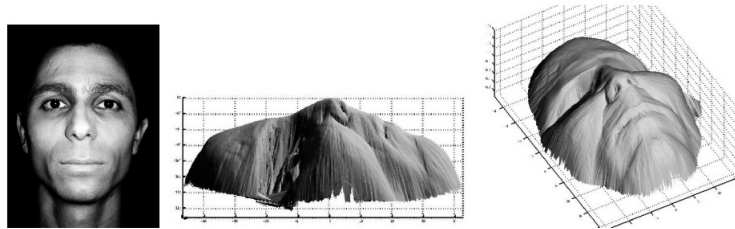


FIGURE 6.2: Résultat de l'approche Shape From Shading présenté dans [86]

luminance peut être décrite en fonction des sources de lumière et de la surface, le problème n'en est pas moins difficile. Si la surface de la forme est décrite par l'information de normale, cela revient à résoudre pour chaque pixel une équation linéaire à trois inconnues. Si la surface est décrite en terme de gradient, cela revient à résoudre une équation non linéaire à deux inconnues. Ainsi trouver une unique solution pour Shape From Shading est difficile et il est souvent nécessaire de définir des contraintes supplémentaires [85],[75]. En dehors des problèmes de leur robustesse de ce type d'approche, les contraintes algorithmiques nécessaires à la résolution de Shape From Shading sont encore loin des contraintes temps réel.

### Shape From Texture

En général les méthodes d'analyse de l'éclairage ne prennent pas en compte les variations de réflectance de la surface à estimer, c'est-à-dire les changements de couleur dus aux matériaux. Dans le monde réel, les changements de réflectance participent à l'apparence perçue des matériaux : le bois, la fourrure, l'herbe, les éraflures et les bosselures conduisent à des variations de l'éclairage perçu, mais ne relèvent pas du champ d'application des approches Shape From Shading. Ce groupe de phénomènes est généralement appelé texture. Les approches Shape from texture estiment la forme des objets en fonction des variations observées dans la texture [5], [37], [67]. Ainsi les objets à construire doivent être texturés que ce soit naturellement ou artificiellement (voir figure 6.3).

### Shape from Focus/Defocus

L'estimation de la profondeur Depth From Defocus ou de la forme Shape From Defocus consiste à reconstruire la forme 3D à partir de plusieurs images prises depuis une caméra fixe, avec différentes valeurs de mise au point. Avec certaines hypothèses

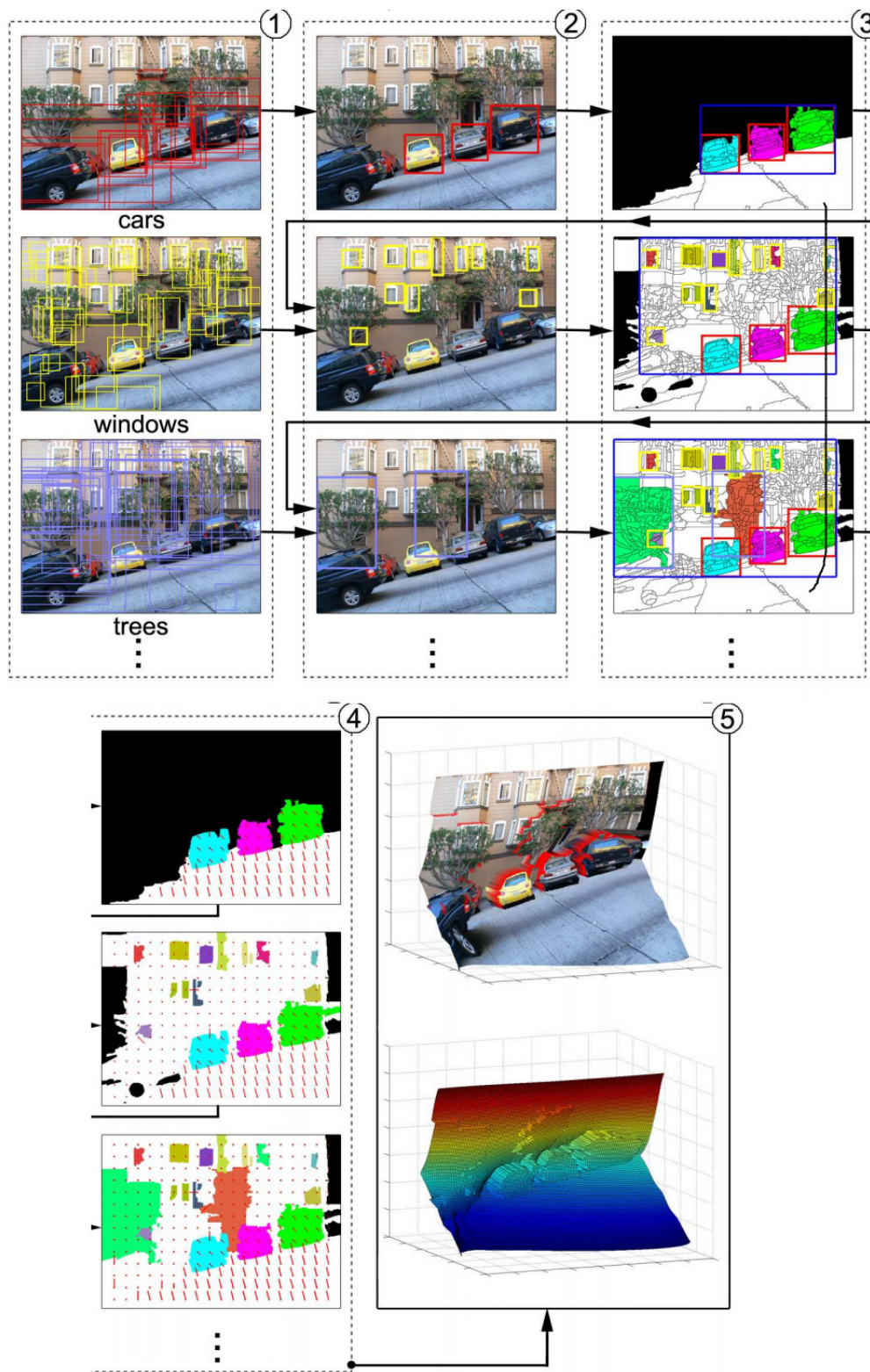


FIGURE 6.3: Résultat de l'approche Shape From Texture présenté dans [84]

de simplification sur la caméra, ce problème peut être posé comme étant l'inversion d'équations d'intégration qui décrivent le processus de création de l'image. Cette approche utilise le fait que, l'image de la scène sur le plan image, dépend de la radiance de la région, autant que de la forme de cette région. Cependant la valeur de chaque pixel provient de l'intégration d'une radiance inconnue par un noyau de convolution inconnu qui dépend aussi de la forme de la scène. Connaissant la valeur de l'intégrale en chaque pixel il faut alors estimer la valeur du noyau et de la radiance correspondante.

Plusieurs approches ont été proposées pour répondre à ce problème, en se basant sur l'hypothèse que la surface de la scène est localement plane et est parallèle au plan de la caméra (hypothèse de l'equifocal ) [7], [26]. Cette hypothèse permet ainsi de décrire l'image observée comme étant une convolution linéaire ; il en revient un compromis fondamental entre la précision et la robustesse au bruit. Pour être résistant aux bruits, certaines approches proposent de calculer l'intégration sur des fenêtres de l'image de taille maximale, cependant pour être en accord avec l'hypothèse de l'equifocal, il est nécessaire que ces fenêtres soient les plus petites possibles. En particulier cette hypothèse est violée sur les contours d'occultation. Pour répondre à cette limitation d'autres approches proposent d'utiliser des noyaux de convolution différents en fonction du voisinage du pixel [18].

Même si certains travaux ont proposé une approche temps réel, ce type d'approche reste cependant peu robuste aux bruits ambiants et sensible au manque de texture des objets à reconstruire.

## **Caméras 2,5D à temps de vol**

Les caméras 2,5D à temps de vol ont fait leur apparition au début des années 2000. Ce nouveau type de capteur offre un fonctionnement similaire à celui des radars. La différence provient du fait que l'onde envoyée n'est pas une onde acoustique, mais lumineuse.

L'appareil se compose d'une source de lumière infrarouge d'amplitude modulée et d'un capteur qui mesure l'intensité rétrodiffusée de la lumière infrarouge. La source infrarouge émet continuellement une lumière dont l'intensité varie de façon sinusoïdale. Ainsi les différents objets de la scène reflètent une intensité de lumière infrarouge dépendante de leur distance à la caméra. Cela provient du fait que des objets à différentes distances de la caméra sont atteints par des parties différentes de l'onde sinusoïdale. La lumière réfléchiée par les objets est ensuite comparée à la modulation d'amplitude

de référence. Le déphasage entre l'onde émise et l'onde reçue est fonction du temps de vol de la lumière réfléchiée par les objets de la scène. En mesurant l'intensité de lumière perçue en chaque pixel, il devient possible de calculer le déphasage correspondant et donc la distance entre les objets de la scène et la caméra.

Ces caméras estiment directement la carte de profondeur de la scène, mais se limitent pour le moment à de faibles résolutions, de l'ordre de  $160 \times 120$  pixels pour les capteurs actuels, comme par exemple pour la caméra SwissRanger 4000 proposée par la société MESA. Une description détaillée de l'approche par temps de vol est décrite dans [41].

Ces nouveaux capteurs amènent une solution alternative aux approches par stéréovision, mais leur coût reste important. De plus ceux-ci ne fournissent qu'une information 2,5D de la scène. Ainsi pour estimer la géométrie 3D, il est nécessaire d'utiliser plusieurs de ces périphériques, en utilisant par exemple la méthode proposée par [12]. Son coût élevé fait que ce type de capteur, quoique prometteur, reste trop luxurieux pour les chercheurs.

## 6.2.2 Approches multi-vues

Les approches monoculaires permettent l'estimation de la géométrie de la scène à partir de l'extraction de primitives depuis un seul point de vue [60]. La présence de conditions environnementales défavorables influe directement sur les résultats de ces approches. Afin de les rendre plus robustes et précises, une solution est de travailler sur l'analyse de plusieurs images prises simultanément depuis plusieurs points de vues.

### Stéréo

Le problème de reconstruction de forme à partir de deux images appelée stéréovision a suscité beaucoup d'intérêt dans les domaines de la vision par ordinateur [60]. La difficulté principale provient du problème de la mise en correspondance, c'est à dire de trouver les points dans chaque image, qui correspondent à un même point dans la scène 3D [33]. Une fois la correspondance trouvée pour tout couple de points issus des deux caméras, ces points peuvent être triangulés pour déterminer le point 3D de la scène correspondant. La mise en correspondance peut être facilitée en utilisant des contraintes géométriques et en faisant certaines hypothèses sur la scène [71], [9].

Supposant les paramètres de calibrage connus (voir chapitre 3), la contrainte épipolaire garantit qu'un point dans une image appartient à la droite épipolaire de ce point dans l'autre image.

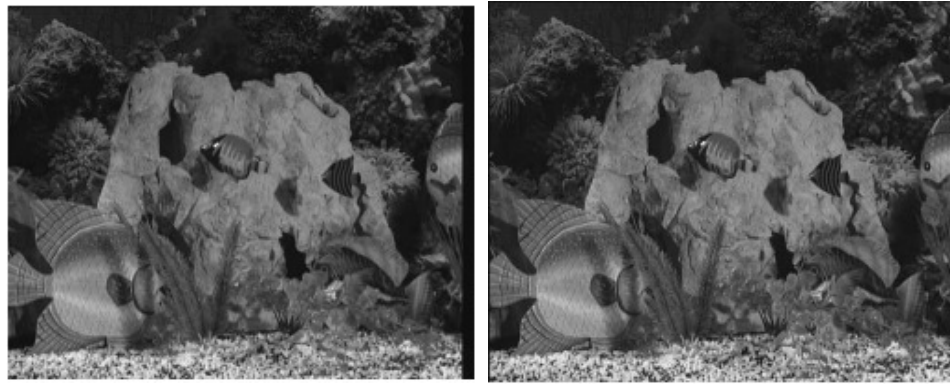
Avec un objet opaque la contrainte d'unicité impose qu'un point dans une image a un unique correspondant dans l'autre image. Cependant en pratique il est possible que les correspondances soient multiples, par exemple lorsque l'objet a une seule couleur. Ainsi cette seconde contrainte n'est pas suffisante pour garantir une mise en correspondance correcte, mais elle peut être utilisée pour vérifier une correspondance calculée par une autre méthode.

Lorsqu'un objet de couleur uniforme doit être reconstruit par une approche stéréo, il peut être intéressant d'utiliser la contrainte de continuité, qui suppose que la surface de l'objet est lisse. Des correspondances erronées qui produisent des profondeurs inconsistantes peuvent ainsi être supprimées.

Enfin la contrainte d'ordre impose que les correspondances entre pixels des images sont dans le même ordre sur la droite épipolaire de chaque image (excepté lorsque les parties à traiter contiennent des occultations ou différents objets). Ainsi les correspondances pour une seule surface peuvent être vérifiées.

Les associations denses résolvent le problème de mise en correspondance, en recherchant plusieurs associations entre les images, de façon à obtenir une carte de profondeur dense. Il existe deux principales méthodes pour estimer une mise en correspondance dense. La première recherche les correspondances pour chaque pixel par des approches d'auto-corrélation normalisée : en comparant une fenêtre autour de ce pixel dans l'autre image en utilisant une mesure de similarité pour décider de l'association [9], [80]. Cette première méthode s'avère sensible aux images bruitées et aux différences de conditions d'éclairage pour chaque vue. De plus elle calcule des estimations de profondeur de faible précision pour des régions d'intensité similaire. Enfin cette approche se révèle lente à cause du nombre important de comparaisons à réaliser pour chaque paire d'images (une par pixel). Le second type d'approche pour la stéréo dense, tire avantage de caractéristiques dans les images, telles que les régions homogènes, les contours [107], les droites ou encore les coins [13]. Ainsi pour que la carte de profondeur soit dense, il est nécessaire que les images contiennent beaucoup de primitives caractéristiques. Une analyse expérimentale des techniques de stéréo vision denses peut être trouvée dans [95]. La figure 6.4 présente l'estimation d'une carte de profondeur par stéréo-vision dense.

Les méthodes que nous venons de citer sont généralement utilisées pour des scènes



(a) Image de gauche

(b) Image de droite



(c) Résultat

FIGURE 6.4: Éstimation d'une carte de profondeur par stéréo-vision

statiques, ou bien en approche image par image pour des objets dynamiques. La stéréo-vision dynamique utilise l'information de mouvements dans les images pour aider à la reconstruction des cartes de profondeur, que ce soit à partir d'une caméra en mouvement, ou encore pour une scène en mouvement. Le mouvement peut être estimé en utilisant le flot optique entre les images successives, puis combiné à l'approche stéréo afin d'obtenir une carte de profondeur relative [46]. D'autres approches estiment les mouvements de la caméra entre les différents points de vue et ainsi raffinent la géométrie pour chaque nouvelle image [96], [108]. La navigation visuelle est une application typique des approches de stéréo-vision dynamiques, utilisée par des robots afin d'identifier et éviter les obstacles.

En dehors des difficultés liées à la mise en œuvre de telles approches, les méthodes intrinsèques de minimisation et d'optimisation n'offrent pas aujourd'hui, de reconstruction suffisamment dense et robuste pour le temps réel.

## Lumière Structurée

Pour répondre aux difficultés des approches de stéréo-vision liées à la mise en correspondance, certaines méthodes dites “actives” projettent de l’énergie lumineuse dans la scène. En remplaçant l’une des caméras d’un système de stéréo-vision par un périphérique qui projette un motif connu sur la scène, la mise en correspondance revient à calculer les correspondances entre les pixels et les points du motif projeté [27]. La principale difficulté de ce type d’approche provient du choix du motif à projeter, ainsi que de la stratégie de codage de ce motif. En effet la mise en correspondance dépend de la capacité à décoder les éléments du motif de façon à les localiser dans l’image.

Les méthodes de lumières structurées peuvent être classées en trois catégories [87] : les approches par multiplexage dans le temps, les méthodes par codage direct et les approches intégrant le voisinage spatial :

- Les approches basées sur des stratégies de multiplexage dans le temps sont faciles à implémenter et peuvent produire une reconstruction de très grande précision et à haute résolution. Cependant ce type d’approche n’est pas adapté à la reconstruction d’objets dynamiques.
- D’autres méthodes codent directement l’information spatiale, c’est à dire que chaque sous-motif est caractérisé par exemple par son intensité, ou par sa forme. Ces méthodes offrent une bonne résolution spatiale. Mais leur application reste limitée à des environnements peu bruités et dont les conditions d’éclairage sont contrôlées.
- Les méthodes de la dernière catégorie sont basées sur la projection d’un motif dans lequel est inséré un codage spatial. Le code de chaque composant du motif dépend à la fois de sa valeur et de la valeur de ses voisins. En général cela permet de travailler avec des scènes dynamiques, mais avec une résolution moindre que les deux précédentes classes de méthode [4].

L’information additionnelle du motif projeté permet en général une mise en correspondance plus robuste et rapide que les approches classiques de stéréo-vision. La projection d’un motif dans la scène revient à marquer la scène d’un ensemble de sous-motifs de couleur. Lorsque l’objet à reconstruire est un humain, cette projection peut être une gêne pour l’utilisateur. Ainsi d’autres approches projettent une lumière infra-rouge non visible par l’œil humain [78]. Si cette nouvelle solution permet d’éviter le problème d’artefacts, elle ajoute une contrainte de matériel difficile à satisfaire. Transformer un vidéo projecteur pour qu’il projette en infra-rouge est déjà une contrainte,

de plus le prix d'achat de ce matériel interdit les applications à domicile pour le grand public.

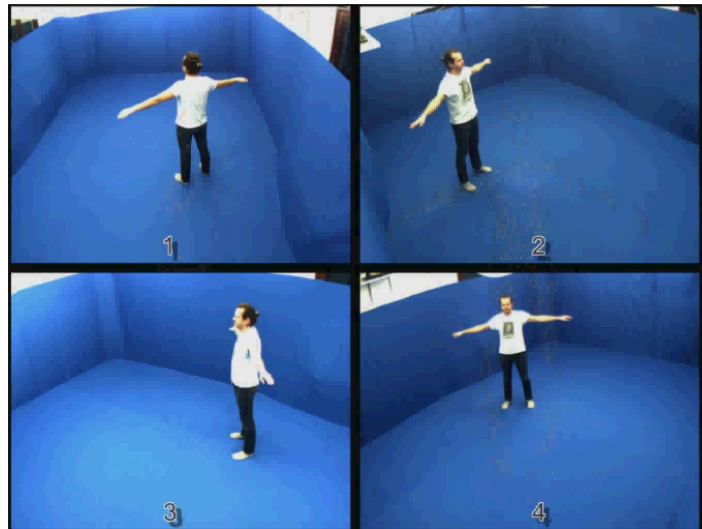
## Shape-From-Silhouette

Les approches par stéréo-vision sont construites sur l'extraction de primitives images, ensuite utilisées pour réaliser la mise en correspondance. Afin de fournir une estimation dense de la géométrie de la scène, il est souvent nécessaire d'extraire des caractéristiques de haut niveau, qui pénalisent les temps de reconstruction. De plus l'information obtenue s'apparente à une carte de profondeur 2,5D. Afin d'obtenir une représentation 3D, il est alors nécessaire de recourir à des approches de stéréo-vision à partir de plus de deux caméras, ce qui impacte à nouveaux les temps de calcul. Afin de diminuer la complexité algorithmique, d'autres approches sont construites sur l'extraction de primitives images plus bas niveau : les silhouettes. Celles-ci sont généralement représentées par un masque binaire qui indique pour tout pixel de chaque image, si celui-ci correspond à l'un des objets à reconstruire.

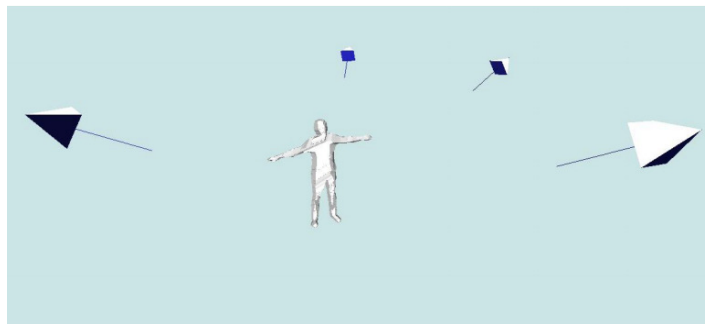
Les méthodes d'estimation de forme 3D à partir de plusieurs silhouettes sont très utilisées dans les environnements multi-caméras, en particulier lorsque cette estimation doit être faite sous la contrainte du temps réel. Ces approches, appelées Shape-From-Silhouette, estiment l'enveloppe visuelle [62](Visual Hull) des objets d'intérêt, qui est une estimation englobante de leur forme 3D, voir figure 6.5.

A partir d'un ensemble de  $n$  vues, l'information de silhouette correspondante à la projection des objets d'intérêt, est extraite des images capturées par une approche dite d'extraction de silhouette. Le cône de silhouette d'une caméra est défini par l'ensemble des demi droites issues du centre optique de la caméra à travers les pixels appartenants à la silhouette. L'enveloppe visuelle (EV) est définie par la forme tridimensionnelle générée par l'intersection des cônes de silhouette de toutes les caméras [62]. La forme produite fournit un volume englobant des objets d'intérêt. Les approches Shape-From-Silhouette ont été utilisées pour diverses applications, telles que la surveillance de foule [103], la modélisation 3D [1] ainsi que l'acquisition de mouvements sans marqueur [16], [21], [3], [20].

Il existe différentes implantations des approches Shape-From-Silhouette afin d'estimer l'enveloppe visuelle. Certaines offrent une estimation de l'enveloppe visuelle en temps réel, qu'elle soit volumique [70], [64], [45] ou surfacique [65], [72]. Dans la mesure où l'enveloppe visuelle définit une forme 3D englobante, cette reconstruction s'avère



(a) Étape 1 : prise de vues de différents angles



(b) Résultat

FIGURE 6.5: Reconstruction 3D à partir des silhouettes présenté dans [90]

peu précise lorsque le nombre de vues est réduit. D'autres approches proposent l'utilisation d'une information de couleur additionnelle afin de réduire la quantité d'artefacts construits.

**Définition** Les silhouettes ont été utilisées pour la première fois par Baumgart dans [10], où il propose de calculer des approximations de formes polyédrales en intersectant les cônes engendrés par les silhouettes [11]. Le terme "enveloppe visuelle" fut introduit plus tard par Laurentini [63] pour décrire le volume maximal compatible avec un ensemble de silhouettes. Le but de cette partie est d'utiliser l'information contenue dans les silhouettes pour distinguer des régions de l'espace où un objet est ou n'est pas présent. L'ultime résultat est une forme appelée enveloppe visuelle de l'objet. L'enveloppe visuelle contient toujours l'objet.

Supposons maintenant qu'un objet (2D ou 3D) est vu par un ensemble  $R$  de

vues de référence. L'objet se projette dans chaque vue  $r$  en une silhouette  $s_r$ . Pour chaque vue  $r$ , on considère le cône  $v_r$  défini par les rayons issus du centre de la caméra correspondante, et passant par les pixels de la silhouette  $s_r$ .

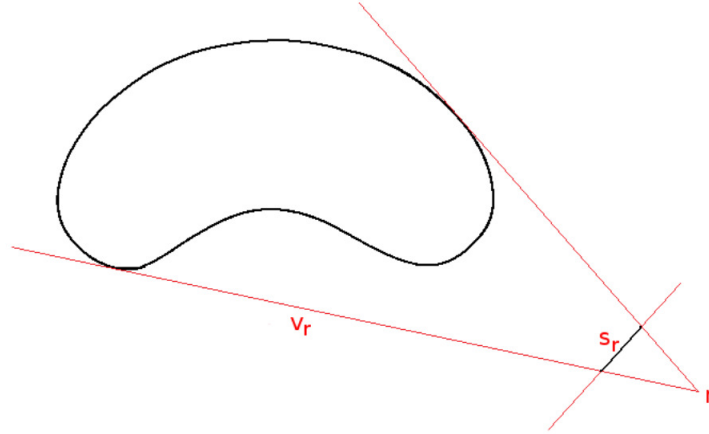


FIGURE 6.6: Un objet 2D représenté avec une vue, sa silhouette et le cône correspondant.

L'objet est toujours contenu dans le cône  $v_r$ . Ceci est vrai pour chaque vue  $r$ , donc l'objet est contenu dans  $v_R = \bigcap_r \in R^{v_r}$ . Quand la taille de  $R$  tend vers l'infini et inclut par conséquent toutes les vues possibles,  $v_R$  converge vers une forme appelée l'enveloppe visuelle de l'objet. L'enveloppe visuelle n'est pas égale à l'objet lui-même car certaines zones concaves ne peuvent pas être prises en compte quand on utilise seulement les silhouettes.

Plus concrètement, si la scène est de dimension 2, l'enveloppe visuelle est l'enveloppe convexe de l'objet, et si la scène est de dimension 3, l'enveloppe visuelle est contenue dans l'enveloppe convexe.

Dans la pratique, on ne peut construire que des approximations de l'enveloppe visuelle, en utilisant un nombre fini de vues. Étant donné un ensemble de vues  $R$ , l'approximation  $v_R$  est la meilleure description géométrique qui puisse être obtenue simplement avec les silhouettes de la scène. La figure 6.7 montre  $v_\infty$  et  $v_R$  dans un cas où  $R$  contient 4 vues différentes d'un unique objet 2D.

La figure 6.8 montre l'une des silhouettes d'un objet tridimensionnel. Cette silhouette est la projection de l'enveloppe visuelle (autant  $v_\infty$  que  $v_R$ ). On peut voir que la silhouette est contenue dans la projection de l'enveloppe convexe de l'objet.

Les enveloppes visuelles peuvent être calculées à partir des silhouettes des objets dans des images, et un certain nombre d'approches ont été proposées. Nous pouvons

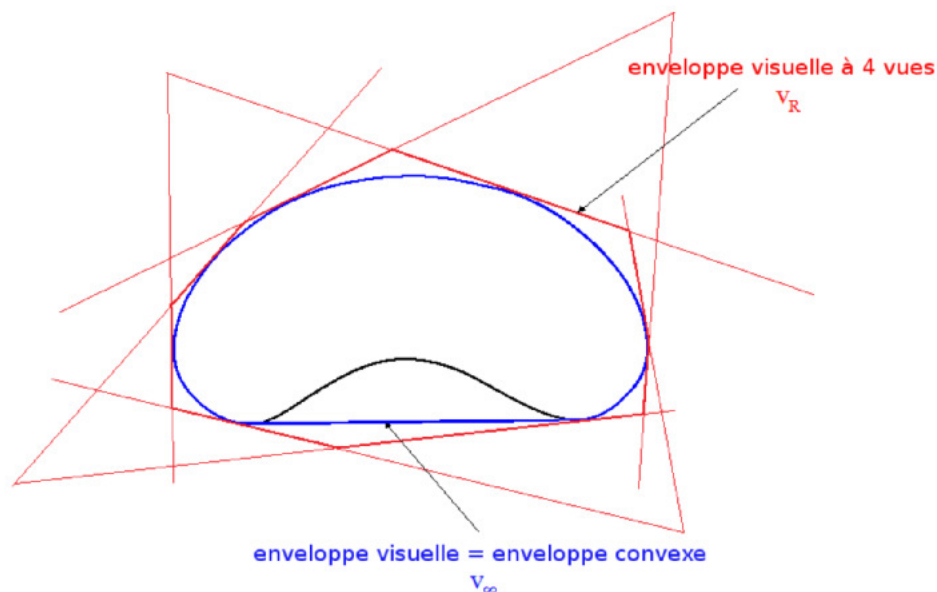


FIGURE 6.7:  $v_\infty$  et  $v_R$  pour un objet 2D :  $v_\infty$  est l'enveloppe convexe de l'objet

globalement les séparer en deux catégories : les volumétriques et les surfaciques. La première catégorie inclut des méthodes qui approximent l'enveloppe visuelle par un ensemble de voxels [91], [31]. Ces approches reposent sur une grille régulière de voxels et sont robustes à des topologies complexes. Néanmoins, la discrétisation de l'espace utilisée entraîne des approximations, et la précision des résultats est décevante comparée à la complexité des algorithmes. Les approches surfaciques estiment au contraire l'enveloppe visuelle en intersectant les cônes issus des silhouettes.

**Space carving** Les approches Shape-From-Silhouette volumiques construisent un volume qui contient l'ensemble des points des objets d'intérêt. Les approches de Space Carving utilisent l'information de couleur afin d'estimer la forme plus précisément, la figure 6.9 montre le principe de fonctionnement de cette approche.

Sous l'hypothèse que tout point de la surface des objets d'intérêt renvoie la même couleur dans toutes les images où il est observé, alors il devient possible d'affiner la reconstruction fournie par une approche Shape-From-Silhouette. Il suffit de supprimer les points reconstruits qui ne respectent pas cette hypothèse [88], [25].

Pour chaque voxel visible (non occulté) un test de validation compare la couleur des pixels sur lesquels ce voxel se projette. Si ces couleurs sont similaires, ce voxel est alors photo-consistant et est validé comme appartenant à la surface de l'objet [61]. Les

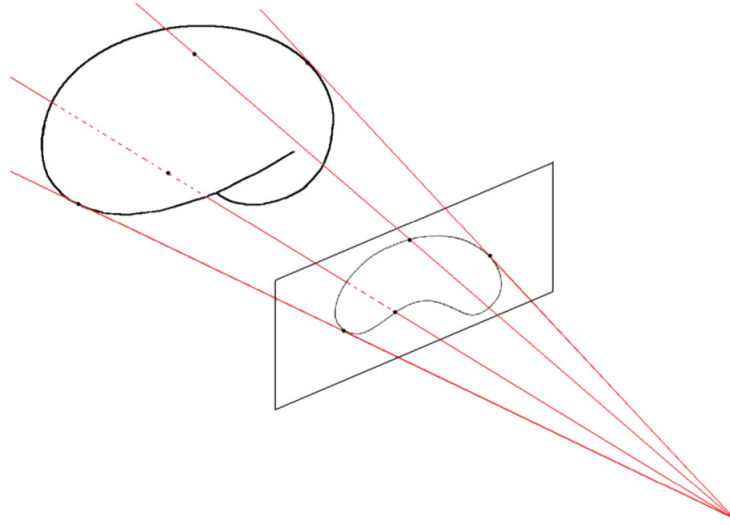


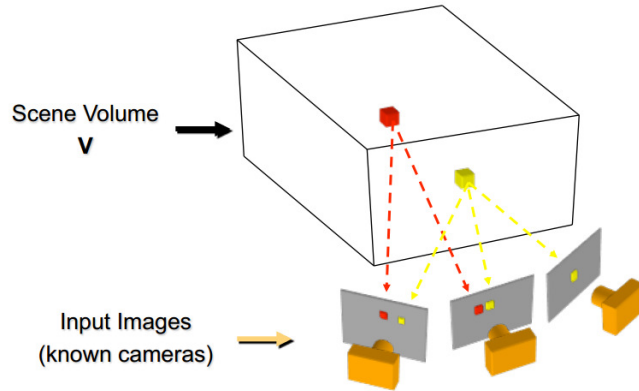
FIGURE 6.8: Silhouette d'un objet 3D : elle est incluse dans la projection de l'enveloppe convexe.

voxels qui ne sont pas photo-consistants sont supprimés, voir la figure 6.10.

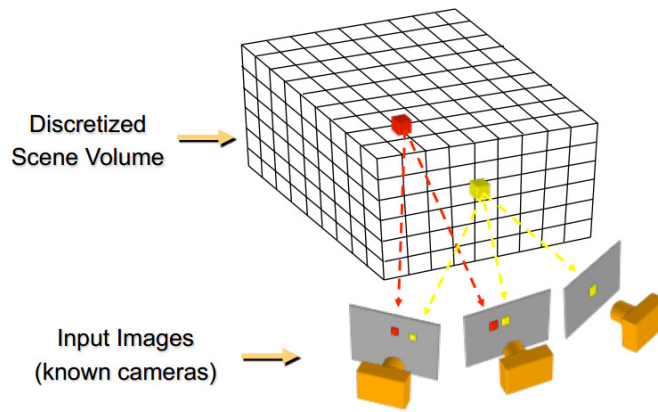
Ainsi la liste des voxels visibles est mise à jour et le processus est réitéré jusqu'à ce que tous les voxels restants soient photo-consistants. L'ensemble de ces voxels décrit la photohull des objets d'intérêt, voir figure 6.11.

Les approches de Space Carving produisent en général une estimation de la forme plus précise que celles proposées par Shape-From-Silhouette, au prix d'une forte sensibilité à l'apparence des objets d'intérêt et de calculs beaucoup plus lourds.

**Modélisation temporelle** En plus de l'information de silhouette, d'autres approches utilisent la continuité temporelle afin de corriger les formes calculées par les approches Shape-From-Silhouette. En utilisant une seule caméra filmant l'objet d'intérêt disposé sur un plateau tournant, il devient possible de construire l'enveloppe visuelle de cet objet à partir d'un grand nombre de vues [82], [35], [101] Ces approches fournissent une méthode intéressante de reconstruction précise de forme 3D, mais imposent que les objets ne soient pas articulés. La continuité temporelle a été aussi utilisée dans un contexte multi-caméras, afin de réduire les artefacts produits par Shape-From-Silhouette [99]. La forme d'objets d'intérêt dynamiques est calculée à chaque trame en utilisant une approche volumique. Ensuite le mouvement des objets est estimé en analysant les reconstructions successives (sceneflow ). La géométrie est synthétisée à un instant particulier en interpolant l'information issue des trames précédentes, courantes et sui-



(a) La scène initiale observée par plusieurs caméras



(b) Discrétisation de la scène en la divisant en voxels

FIGURE 6.9: Principe du fonctionnement du Space carving

vantes. La représentation englobante proposée par les méthodes Shape-From-Silhouette est souvent utilisée comme une initialisation des approches de reconstructions temporelles [19] afin de déterminer le mouvement rigide des objets de la scène et ainsi affiner leur reconstruction au cours du temps. D'autres approches proposent une représentation qui lie l'information spatiale et temporelle. Par exemple Goldluecke et Magnor [43] proposent une méthode qui modélise la scène comme étant une surface dans un espace spatio-temporel, qui est ensuite corrigée en utilisant la photo-consistance à travers la séquence complète. Cette approche fournit des résultats intéressants, voir la figure 6.12, mais impose des calculs particulièrement intensifs [44].

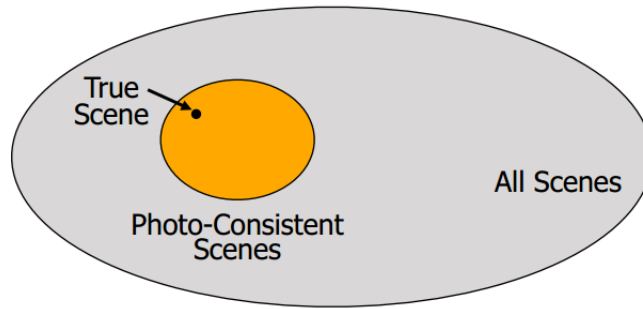


FIGURE 6.10: L'ensemble des pixels photo-consistants

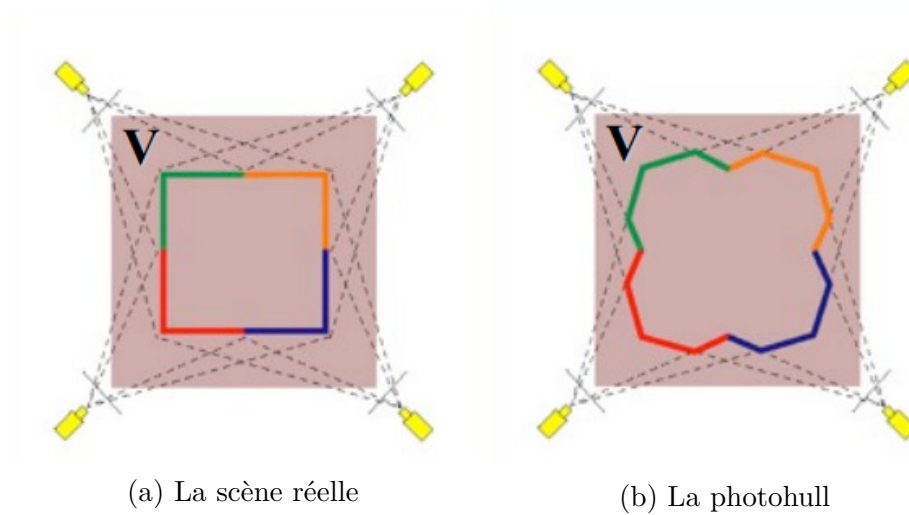


FIGURE 6.11: Représentation du Photohull

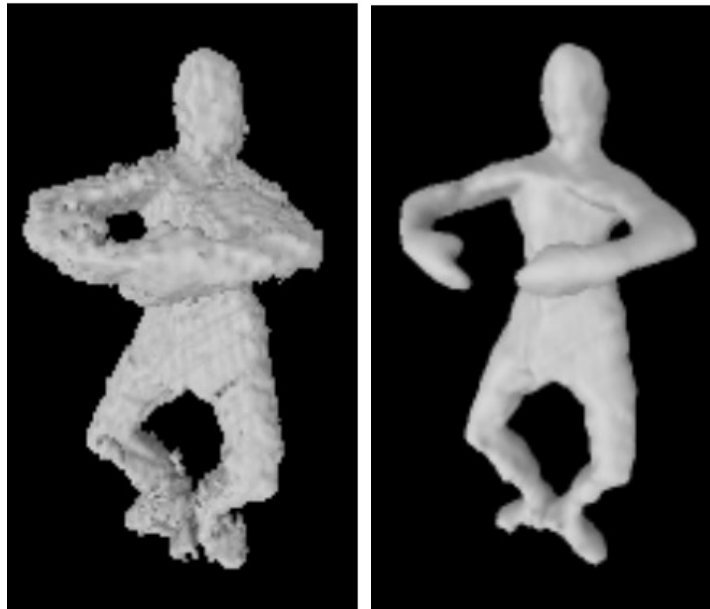
### 6.2.3 Reconstruction à partir du mouvement

Hormis certaines configurations dégénérées, le mouvement de la caméra dans l'espace tridimensionnel se projette en un mouvement 2D dans la séquence d'images capturées[14]. Ce mouvement 2D peut être représenté dans le plan image comme un champ vectoriel de déplacement. En mesurant ce champ de vecteurs on peut estimer la carte de disparité, et retrouver la structure de l'objet observé comme il est montré dans la figure 6.13, qui montre une reconstruction de la structure d'un objet (en haut à gauche) à partir du champ vectoriel, e(n haut à droite) permettant de définir une carte de profondeur (en bas à gauche) pouvant ensuite être représentée par un maillage 3D (en bas à droite). Les champs de vecteurs peuvent être approximés comme des champs de vitesse (flot optique).

Ainsi, la reconstruction de la structure à partir du mouvement (Shape from mo-



(a) Silhouettes sources



(b) L'enveloppe visuelle

(c) Le résultat final

FIGURE 6.12: Résultat de la modélisation temporelle présenté dans [44]

tion) consiste à prendre à l'aide d'une caméra, une séquence d'images de l'objet placé par exemple sur une table tournante, ce qui correspond à un environnement contrôlé. Ensuite pour chaque image, on estime le mouvement apparent de certains points (ou de tous les points de l'image), puis on retrouve la forme 3D à l'aide des vitesses estimées. Le mouvement est supposé très faible entre chaque image, et la scène est statique afin que le déplacement des points dans l'image ne soit dû qu'au mouvement de la caméra. L'avantage de ce genre de méthode est de pouvoir facilement mettre en correspondance les pixels entre les images par des méthodes de suivi visuel, pouvant exploiter l'information du mouvement effectué pour prédire les disparités dans l'image suivante. Mais l'inconvénient est une plus grande sensibilité au bruit par rapport à une méthode stéréo, car l'espacement entre deux images consécutives étant très faible, le calcul de la

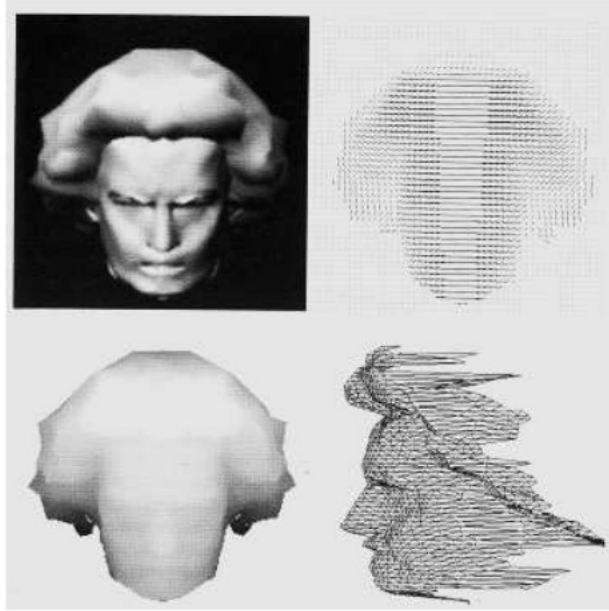


FIGURE 6.13: Reconstruction de la structure d'un objet à partir du champ vectoriel

disparité est instable.

### 6.3 Reconstruction 3D à partir des caméras non calibrées

On va maintenant dériver une formule spécifique pour une paire de caméras de formes canoniques ayant  $F$ . On utilisera la caractéristique suivante de la matrice fondamentale  $F$  correspondante à une paire de matrices de caméras :

**Résultat** Une matrice non-nulle  $F$  est la matrice fondamentale correspondante à une paire de matrices de caméras  $P$  et  $P'$  si et seulement si  $P'^T F P$  est antisymétrique.

**Preuve** La condition qui dit que  $P'^T F P$  est antisymétrique est équivalente à  $X^T P'^T F P X = 0$  pour tout  $X$ . Mettant  $x' = P'X$  et  $x = PX$ , ceci équivaut à  $x'^T F x = 0$ , ce qui est l'équation qui définit la matrice fondamentale.

D'après [50], une fois qu'on a calculé la géométrie épipolaire entre deux images, on peut extraire la forme trois dimensionnelle en utilisant la forme canonique des matrices de caméras, qui correspondent à la matrice fondamentale comme suit :

**Résultat** Soit  $F$  la matrice fondamentale, et  $S$  une matrice antisymétrique quelconque. On définit la paire de matrices de caméras comme suit :

$$P = [I|0] \quad \text{et} \quad P' = [SF|e']$$

Où  $e'$  est l'épipôle sachant que  $e'^T F = 0$ , et on assume que  $P'$  comme défini est une matrice de caméra valide (elle est de rang 3). Alors  $F$  est la matrice fondamentale correspondante à la paire  $(P, P')$ .

Pour illustrer ça, on invoque le résultat 6.3, et tout simplement vérifier que :

$$[SF|e']^T F [I|0] = \begin{bmatrix} F^T S^T F & 0 \\ e'^T F & 0 \end{bmatrix} = \begin{bmatrix} F^T S^T F & 0 \\ 0^T & 0 \end{bmatrix} \quad (6.1)$$

qui est une matrice antisymétrique.

La matrice antisymétrique  $S$  peut s'écrire en termes de son vecteur nul comme suit :  $S = [s]_x$ . Alors  $[[s]_x F|e']$  a le rang 3 produit par  $s^T e' \neq 0$ , suivant l'argument suivant. Sachant que  $s^T e' = 0$ , l'espace de colonne de  $F$  est perpendiculaire à  $e'$ . Mais, si  $s^T e' \neq 0$ , alors  $s$  n'est pas perpendiculaire à  $e'$ , et n'est donc dans l'espace colonne de  $F$ . Maintenant, l'espace de colonne de  $[s]_x F$  est traversé par le produit vectoriel de  $s$  avec les colonnes de  $F$ , et alors égale au plan perpendiculaire à  $s$ . Donc,  $[s]_x F$  a le rang 2. Étant donné que  $e'$  n'est perpendiculaire à  $s$ , il n'est pas sur ce plan, et alors  $[[s]_x F|e']$  a le rang 3, comme requis.

Comme l'est suggéré par [69], un bon choix pour  $S$  est  $S = [e']_x$ , pour ce cas  $e'^T e' \neq 0$ , qui donne le résultat suivant :

**Résultat** Les matrices de caméras correspondants à la matrice fondamentale  $F$  peuvent être choisies comme :  $P = [I|0]$  et  $P' = [[e']_x F|e']$ .

Notons que la matrice de caméra  $P'$  a la sous-matrice gauche  $[e']_x F$  qui est de taille  $3 \times 3$  et le rang 2. Ceci correspond à une caméra avec le centre qui sur  $\pi_\infty$ . Cependant, il n'y a pas de raison particulière d'éviter cette situation.

**Résultat** La formule générale d'une paire de matrices de caméras canoniques correspondantes à une matrice fondamentale  $F$  est donnée par :

$$\begin{aligned} P &= [I|0] \\ P' &= [[e']_x F + e'v^T | \lambda e'] \end{aligned} \tag{6.2}$$

Où  $v$  est vecteur aléatoire, et  $\lambda$  est un scalaire non-nul.

## 6.4 Reconstruction à partir des caméras calibrées

La matrice essentielle peut être directement calculée de 3.7 en utilisant les coordonnées normalisées de l'image, ou encore en utilisant la matrice fondamentale 3.8. Une fois la matrice essentielle est connue, les matrices de caméras peuvent être récupérés depuis  $E$  comme il est décrit ici. À l'inverse du cas de la matrice fondamentale, où il existe une ambiguïté projective, les matrices des caméras peuvent être récupérés depuis la matrice essentielle jusqu'à un facteur et une ambiguïté quadruple. Ceci est due à le fait qu'il y a quatre solutions possibles. On peut assumer que la matrice de la première caméra est  $P = [I|0]$ . Dans le but de calculer la matrice de la deuxième caméra  $P'$ , il est nécessaire de diviser  $E$  en un produit  $SR$  de matrice antisymétrique et une matrice de rotation.

**Résultat** On suppose que le SVD de  $E$  est  $Udiag(1, 1, 0)V^T$ . En utilisant la notation présente dans [50](9.6.2 (9.13)) il y a deux factorisations possibles  $E = SR$  (en ignorant les signes) comme suit :

$$S = UZU^T \quad R = UWV^T \quad UW^T v^T \tag{6.3}$$

**Preuve** On suppose que  $E = SR$ . La forme de  $S$  est déterminée par le fait que l'espace nul gauche est le même que celui de  $E$ . Alors  $S = UZU^T$ . La rotation  $R$  peut s'écrire comme  $UXV^T$ , où  $X$  est une matrice de rotation quelconque. Alors, on a :

$$Udiag(1, 1, 0)V^T = E = SR = (UZU^T)(UXV^T) = U(ZX)V^T$$

de cela on déduit que  $ZX = diag(1, 1, 0)$ . Étant donné que  $X$  est une matrice de rotation, donc  $X = W$  ou  $X = W^T$ , comme il est requis.

La factorisation 6.3 détermine la partie  $t$  de la matrice de caméra  $P'$  depuis  $S = [t]_x$ , jusqu'à un facteur d'échelle. Cependant, la norme Frobenius de  $S = UZU^T$  est  $\sqrt{2}$ , ce qui veut dire que si  $S = [t]_x$  alors  $\|t\| = 1$ , qui est une normalisation convenable pour la ligne de base des deux matrices de caméras. Étant donné que  $St = 0$ , alors  $t = U(0, 0, 2)^T = u_3$  la dernière colonne de  $U$ . Cependant, les signes de  $E$  et  $t$  ne peuvent pas être déterminés. Donc, en correspondance d'une matrice essentielle donnée, il y a quatre choix possibles pour la matrice de caméra  $P'$ , basé sur 2 choix possibles de  $R$  et deux signes possibles de  $t$ . Pour résumer :

**Résultat** Pour une matrice essentielle donnée  $E = Udiag(1, 1, 0)V^T$ , et une première caméra avec la matrice  $P = [I|0]$ , il y a quatre choix possibles pour la matrice de la deuxième caméra  $P'$ , à savoir :

$$\begin{aligned}
P' &= [UWV^T | +u_3] \quad ou \\
P' &= [UWV^T | -u_3] \quad ou \\
P' &= [UW^T V^T | +u_3] \quad ou \\
P' &= [UW^T V^T | -u_3]
\end{aligned} \tag{6.4}$$

### 6.4.1 Interprétation géométrique des quatre solutions

Il est clair que la différence entre les deux première solutions est simplement la direction du vecteur de translation de la première caméra vers la deuxième est inversée.

La relation entre la première et la troisième solution dans 6.4 est un peu compliquée. Cependant, on peut vérifier que :

$$[UW^T V^T | u_3] = [UWV^T | u_3] \begin{bmatrix} VW^T W^T V^T & \\ & 1 \end{bmatrix}$$

et  $VW^T W^T V^T = Vdiag(-1, -1, 1)V^T$  est une rotation de  $180^\circ$  autour de la ligne qui lie les deux centres des caméras. Les quatre solutions sont illustrés dans la figure 6.14, où il est montré qu'un point reconstruit  $X$  sera devant les deux caméras dans l'une de ces quatre situations. Donc, en essayant avec un seul point pour déterminer s'il est devant les deux caméras ou non est suffisant pour décider entre les quatre différentes solutions pour la matrice de caméra  $P'$ .

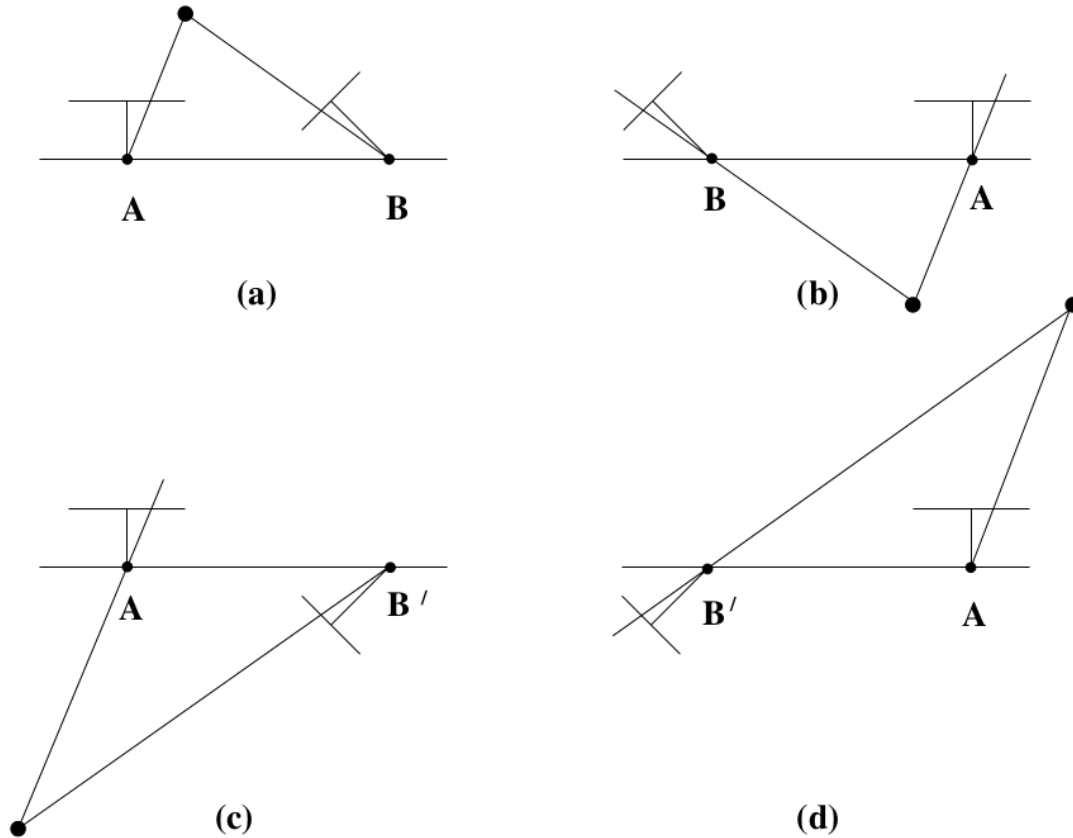


FIGURE 6.14: Les quatre solutions possibles pour une reconstruction calibrée depuis  $E$

## 6.5 Théorème de la reconstruction projective

On va maintenant discuter le théorème de la reconstruction projective qui dit que :

*Si un ensemble de correspondances de points dans deux vues détermine une matrice fondamentale unique, alors la scène et les caméras peuvent être reconstituées depuis ces correspondances seulement, et n'importe quelles deux reconstructions de ces correspondances sont projectivement équivalentes.*

Les points situés sur la ligne qui relie les deux centres des caméras doivent être exclus, puisque tels points ne peuvent pas être reconstruits de façon unique même si les matrices de la caméra sont déterminées.

**Théorème 6.5.1** (*théorème de la reconstruction projective*)

*Supposons que  $x_i \leftrightarrow x'_i$  est un ensemble de correspondances entre les points dans les deux images et que la matrice fondamentale  $F$  est déterminée de façon unique par la*

condition  $x_i^T F x_i = 0$  pour tout  $i$ . Soit  $(P_1, P'_1, X_{1i})$  et  $(P_2, P'_2, X_{2i})$  deux reconstructions des correspondances  $x_i \leftrightarrow x'_i$ . Alors il existe une matrice non singulière  $H$  tel que  $P_2 = P_1 H^{-1}$ ,  $P'_2 = P'_1 H^{-1}$  et  $X_{2i} = H X_{1i}$  pour tout  $i$ , sauf pour les tel que  $F x_i = x_i^T F = 0$ .

**Preuve.** Étant donné que la matrice fondamentale est déterminée de façon unique par les correspondances de points, on en déduit que  $F$  est la matrice fondamentale correspondante à la paire de caméras  $(P_1, P'_1)$  et aussi à  $(P_2, P'_2)$ . Selon [50] (théorème 9,10 (P254)), il existe une homographie  $H$  tel que  $P_2 = P_1 H^{-1}$  et  $P'_2 = P'_1 H^{-1}$  selon les besoins. Comme pour les points, on observe que  $P_2(H X_{1i}) = P_1 H^{-1} H X_{1i} = P_1 X_{1i} = x_i$ . D'un autre côté  $P_2 X_{2i} = x_i$ , donc  $P_2(H X_{1i}) = P_2 X_{2i}$ . Ainsi, les deux  $H X_{1i}$  et  $X_{2i}$  projettent au même point  $x_i$  sous l'action de la caméra  $P_2$ . Donc les deux  $H X_{1i}$  et  $X_{2i}$  sont situés sur le même vecteur passant par le centre de la caméra de  $P_2$ . De même, on peut en déduire que ces deux points sont situés sur le même rayon passant par le centre de la caméra de  $P'_2$ . Il y a deux possibilités : soit  $X_{2i} = H X_{1i}$  comme il est requis, ou ils sont des points distincts situés sur la ligne qui relie les deux centres des caméras. Dans ce dernier cas, les points d'image  $x_i$  et  $x'_i$  coïncident avec les épipôles dans les deux images, et ainsi  $F X_i = x_i^T F = 0$ . ■

C'est un résultat extrêmement important, car il implique que l'on peut calculer une reconstruction projective d'une scène de deux points de vue basés seulement sur les correspondances de l'image, sans rien connaître sur l'étalonnage ou les poses des deux caméras impliquées. En particulier, la reconstruction correcte est dans une transformation projective de la reconstruction projective. La figure 6.15 montre un exemple de structure 3D calculé dans le cadre d'une reconstruction projective de deux images. De manière plus détaillée, supposons que la reconstruction correcte euclidienne est  $(P_E, P'_E, X_{Ei})$  et la reconstruction projective est  $(P, P', X_i)$ , les reconstructions sont liées par une matrice non singulière  $H$  de telle sorte que

$$\begin{aligned} P_E &= P H^{-1} \\ P'_E &= P' H^{-1} \\ X_{Ei} &= H X_i \end{aligned} \tag{6.5}$$

où  $H$  est la matrice d'homographie de taille  $4 \times 4$ , qui est inconnue mais la même pour tous les points.

Pour certaines applications, la reconstruction projective est tout ce qui est nécessaire. Par exemple, des questions comme "à quel moment une ligne croise un plan?", "Quelle est la correspondance entre deux points de vues induites par des surfaces parti-

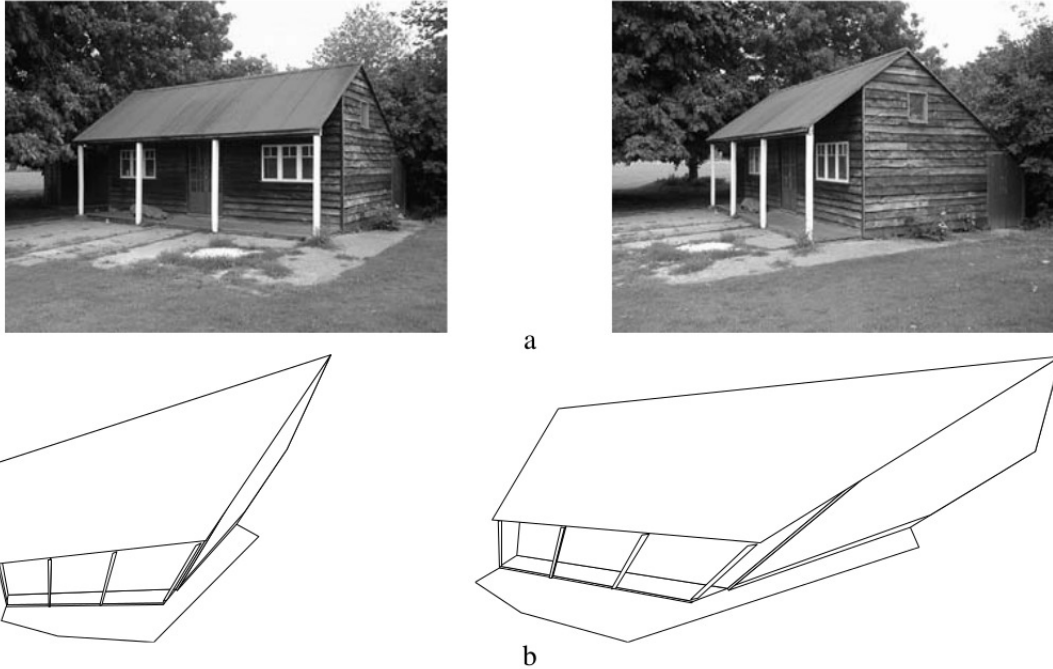


FIGURE 6.15: Reconstruction projective

culières, comme un plan ou un quadrique ?” peuvent être traitées directement à partir de la reconstruction projective. En outre, il sera vu par la suite que l’obtention d’une reconstruction projective d’une scène est la première étape vers la reconstruction affine ou métrique.

## 6.6 Reconstruction stratifié

L’approche «stratifié» pour la reconstruction est pour commencer une reconstruction projective, pour ensuite l’affiner progressivement à une affine et enfin une reconstruction métrique, si possible. Bien sûr, comme on vient de le voir, la reconstruction affine et métrique ne sont pas possibles sans des informations supplémentaires soit sur la scène, le mouvement ou l’étalonnage de la caméra.

**L’étape vers la reconstruction affine** L’essence de la reconstruction affine est de localiser le plan à l’infini par certains moyens, sachant que cette connaissance est équivalente à une reconstruction affine. Cette équivalence est expliqué dans le cas 2D à la section 2.7 (p47). Pour voir cette équivalence pour la reconstruction, supposons que nous avons déterminé une reconstruction projective d’une scène, constituée d’une

triple  $(P, P', X_i)$ . Supposons en outre que par des moyens un certain plan  $\pi$  a été identifié comme le plan correcte à l'infini. Le plan  $\pi$  est exprimée par un vecteur de taille 4 dans le système de coordonnées de la reconstruction projective. Dans la vraie reconstruction,  $\pi$  a les coordonnées  $(0, 0, 0, 1)^T$ , et on peut trouver une transformation projective qui projette  $\pi$  à  $(0, 0, 0, 1)^T$ . Compte tenu de la façon dont une transformation projective agit sur les plans, on veut trouver  $H$  tel que  $H^{-T}\pi = (0, 0, 0, 1)^T$ . Une telle transformation est donnée par

$$H = \begin{bmatrix} I & | & 0 \\ \hline & & \pi^T \end{bmatrix} \quad (6.6)$$

En effet, on vérifie immédiatement que  $H^T(0, 0, 0, 1)^T = \pi$ , et donc  $H^{-T}\pi = (0, 0, 0, 1)^T$ , comme on le souhaite. La transformation  $H$  s'applique maintenant à tous les points et les deux caméras. Notez cependant que cette formule ne fonctionnera pas si la coordonnée finale de  $\pi^T$  est égal à zéro. Dans ce cas, on peut calculer une  $H$  adaptée en calculant  $H^{-T}$  comme une matrice Householder de sorte que  $H^{-T}\pi = (0, 0, 0, 1)^T$ .

À ce stade, la reconstruction qu'on a, n'est pas nécessairement la véritable reconstruction. Tout ce qu'on sait, ce est que le plan à l'infini est correctement placé. La reconstruction actuelle diffère de la reconstruction véritable par une transformation projective qui fixe le plan à l'infini. Toutefois, selon [50] (résultat 3.7 (p80)), une transformation projective qui fixe le plan à l'infini est une transformation affine. Ainsi, la reconstruction diffère par une transformation affine de la vraie reconstruction - *c'est une reconstruction affine*.

Une reconstruction affine pourrait bien être suffisante pour certaines applications. Par exemple, le point au milieu de deux points et le centre de gravité d'un ensemble de points peuvent maintenant être calculés, et les lignes construites en parallèle à d'autres lignes et plans. Ces calculs ne sont pas possibles à partir d'une reconstruction projective.

Comme il a été dit, le plan à l'infini ne peut être identifié à moins que quelques informations supplémentaires sont données. On donnera maintenant plusieurs exemples du type d'informations qui suffisent pour cette identification.

**Translation** Considérons le cas où la caméra va subir un mouvement purement translationnelle. Dans ce cas, il est possible de procéder à la reconstruction affine à partir de deux points de vue. Une façon simple de voir ça, c'est d'observer qu'un point  $X$  sur le plan à l'infini va être projeté au même point dans deux images liées par une translation. Il fait également partie de notre expérience commune que lorsqu'on se déplace en ligne

droite (par exemple une voiture sur une route droite), les objets à une grande distance (tels que la lune) ne semblent pas bouger, seulement les objets proches se déplacent dans le champ de vision. Cela étant, on peut inventer n'importe quel nombre de points appariés  $x_i \leftrightarrow x_i$ , où un point dans une image correspond au même point dans l'autre image. Notez qu'on n'a pas besoin d'observer une telle correspondance dans les deux images - n'importe quel point et le même dans l'autre image fera l'affaire. Compte tenu d'une reconstruction projective, on peut alors reconstruire le point  $X_i$  correspondant à  $x_i \leftrightarrow x_i$ . Le point  $X_i$  sera sur le plan à l'infini. À partir de trois de ces points, on peut prendre les trois points sur le plan à l'infini - suffisants pour le déterminer de manière unique.

Bien que cet argument donne une preuve constructive que la reconstruction affine est possible à partir d'une translation d'une caméra, cela ne signifie pas que c'est la meilleure façon de procéder numériquement. En fait, dans ce cas, l'hypothèse d'un mouvement de translation implique une forme très limitée pour la matrice fondamentale - elle est anti-symétrique. Cette forme particulière devrait être prise en compte lors de la résolution de la matrice fondamentale.

**Résultat** Supposons que le mouvement de la caméra est une translation pure sans aucune rotation et aucun changement dans les paramètres internes. On a  $F = [e]_x = [e']_x$ , et pour une reconstruction affine on peut choisir les deux caméras comme  $P = [I|0]$  et  $P' = [I|e']$ .

**Contraintes de scène** Les contraintes ou conditions de scène peuvent également être utilisées pour obtenir une reconstruction affine. Aussi longtemps que trois points peuvent être identifiés qui sont connus pour être sur le plan à l'infini, alors ce plan peut être identifié, et la reconstruction peut être transformé à une reconstruction affine.

**Les lignes parallèles** La telle condition la plus évidente est la connaissance que les lignes 3D sont en réalité parallèle. L'intersection des deux lignes parallèles dans l'espace donne un point sur le plan à l'infini. L'image de ce point est le point de fuite de la ligne, et est le point d'intersection des deux lignes imagées. Supposons que les trois ensembles de lignes parallèles peuvent être identifiés dans la scène. Chaque ensemble se croise dans un point sur le plan à l'infini. À condition que chaque ensemble ait une direction différente, les trois points seront distincts. Sachant que trois points déterminent un plan, cette information est suffisante pour identifier le plan  $\pi$ .

La meilleure façon de calculer effectivement l'intersection de lignes dans l'espace est un problème assez délicat, car en présence de bruit, des lignes qui sont supposés s'intersecter le font rarement. Un exemple d'une reconstruction affine calculée à partir de trois ensembles de lignes parallèles de scène est donnée dans la figure 6.16.

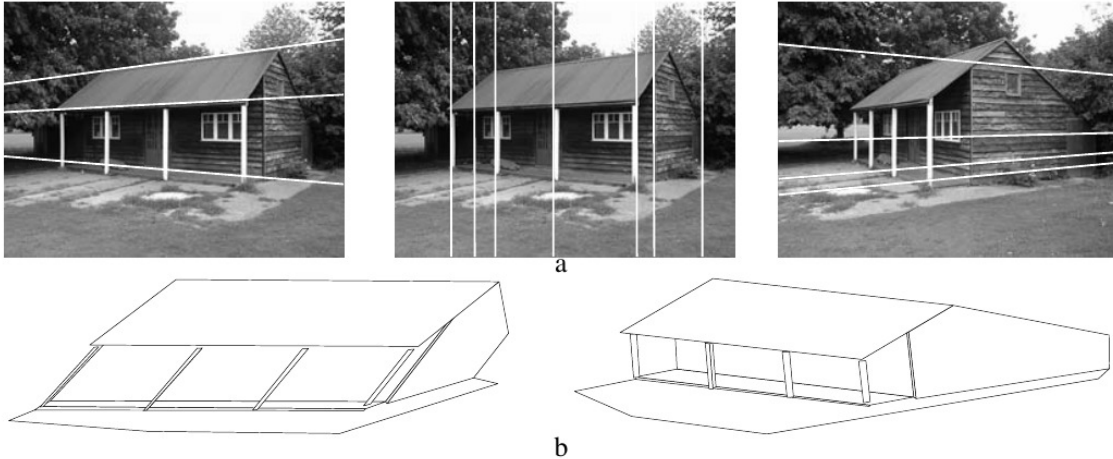


FIGURE 6.16: Reconstruction affine

Notez qu'il n'est pas nécessaire de trouver le point de fuite dans les deux images. Supposons que le point de fuite  $v$  est calculée à partir des lignes parallèles imagées dans la première image, et  $l$  est une ligne correspondante dans la deuxième image. Les points de fuite satisfont la contrainte épipolaire, de sorte que le point de fuite correspondant  $v'$  dans la deuxième image peut être calculés comme l'intersection de  $l'$  et la ligne épipolaire  $Fv$  de  $v$ . La construction du point 3D  $X$  peut être parfaitement exprimé algébriquement comme la solution des équations  $([v]_x P)X = 0$  et  $(l'^T P')X = 0$ . Ces équations expriment le fait que  $X$  se projette sur  $v$  dans la première image, et sur un point sur  $l'$  dans la deuxième image.

**Ratio de distance sur une ligne** Une alternative au calcul de points de fuite comme l'intersection des lignes parallèles de la scènes est d'utiliser les connaissances des ratios de longueur affines dans la scène. Par exemple, avec deux intervalles sur une ligne avec un rapport de longueur connu, le point à l'infini sur la ligne peut être déterminée. Cela signifie qu'à partir d'une image d'une ligne sur laquelle un rapport de distance du monde est connu, par exemple, que trois points sont espacés de façon égale, le point de fuite peut être déterminé. Ce calcul, et d'autres moyens de calcul de points de fuite et lignes de fuite, sont décrits dans [50].

**L’homographie infinie** Une fois que le plan à l’infini a été localisé, de sorte qu’on aura une reconstruction affine, et on aura aussi une projection image-vers-image appelée “homographie infinie”, qui est une homographie 2D. Bref, c’est la projection qui transfère des points de l’image de  $P$  à l’image de  $P'$  via le plan à l’infini comme suit : le rayon correspondant à un point  $x$  est étendu pour trouver le plan à l’infini dans un point  $X$  ; ce point est projeté à un point  $x'$  dans l’autre image. L’homographie de  $x$  à  $x'$  s’écrit  $x' = H_\infty x$ .

Ayant une reconstruction affine équivaut à savoir l’homographie infinie, comme il sera montré. Compte tenu de deux caméras  $P = [M|m]$  et  $P' = [M'|m']$  d’une reconstruction affine, l’homographie infinie est donnée par  $H_\infty = M'M^{-1}$ . C’est parce qu’un point  $X = (\tilde{X}^T, 0)^T$  sur le plan à l’infini se projette à  $x = M\tilde{X}$  dans une image et  $x' = M'\tilde{X}$  dans l’autre, donc  $x' = M'M^{-1}x$  pour les points sur  $\pi_\infty$ . En outre, il peut être vérifié que ça change pas pour une transformation affine des caméras dans l’espace 3D. Alors, l’homographie infinie peut être calculée explicitement à partir d’une reconstruction affine, et vice versa. :

**Résultat** Si une reconstruction affine a été obtenue dans laquelle les matrices de caméra sont  $P = [I|0]$  et  $P' = [M'|e']$ , alors l’homographie infinie est donnée par  $H_\infty = M'$ . Inversement, si homographie infinie  $H_\infty$  a été obtenue, alors les caméras d’une reconstruction affine peuvent être choisis comme  $P = [I|0]$  et  $P' = [H_\infty|e']$ .

L’homographie infinie peut être calculée directement à partir des entités correspondantes d’image, plutôt qu’indirectement à partir d’une reconstruction affine. Par exemple,  $H_\infty$  peut être calculée à partir de la correspondance de trois points de fuite ensemble avec  $F$ , ou la correspondance d’un point de fuite et une ligne de fuite, avec  $F$ . Cependant, de tels calculs directs sont totalement équivalents à déterminer  $\pi_\infty$  dans une reconstruction projective.

### 6.6.1 L’étape de reconstruction métrique

Tout comme la clé de la reconstruction affine est l’identification du plan à l’infini, la clé de la reconstruction métrique est l’identification de la conique absolue. Sachant que la conique absolue,  $\Omega_\infty$ , est une conique plane, située dans le plan à l’infini, en identifiant la conique absolue implique l’identification du plan à l’infini.

Dans une approche stratifiée, on procède à partir d’une reconstruction projective

vers une reconstruction affine à une reconstruction métrique, donc on connaît le plan à l'infini avant de trouver la conique absolue. Supposons qu'on a identifié la conique absolue sur le plan à l'infini. En principe, l'étape suivante consiste à appliquer une transformation affine à la reconstruction affine de sorte que la conique absolue identifiée est projetée à la conique absolue dans le cadre euclidienne standard (il aura alors l'équation  $X_1^2 + X_2^2 + X_3^2 = 0$ , sur  $\pi_\infty$ ). La reconstruction résultat est alors liée à la véritable reconstruction par une transformation projective qui fixe la conique absolue. Il résulte du résultat présenté dans [50] (résultat 3,9 (p82)) que la transformation projective est une transformation de similitude, donc on a réalisé une reconstruction métrique.

En pratique, la meilleure façon d'y parvenir est de considérer l'image de la conique absolue dans l'une des images. L'image de la conique absolue (comme tout conique) est une conique dans l'image. La projection inverse de cette conique est un cône, qui va trouver le plan à l'infini dans une seule conique, qui définit donc la conique absolue. On se rappelle que l'image de la conique absolue est une propriété de l'image elle-même, et comme n'importe quel point de l'image, une ligne ou une autre caractéristique, ne dépend pas de n'importe quelle reconstruction en particulier, elle est donc inchangée par des transformations 3D de la reconstruction.

Supposons que dans la reconstruction affine de l'image de la conique absolue telle que vue par la caméra avec la matrice  $P = [M|m]$  est une conique  $\omega$ . On va montrer comment  $\omega$  peut être utilisé pour définir l'homographie  $H$  qui transforme la reconstruction affine à une reconstruction métrique :

**Résultat** Supposons que l'image de la conique absolue est connue dans certains image comme étant  $\omega$ , et qu'on a une reconstruction affine dans laquelle la matrice correspondante de la caméra est donnée par  $P = [M|m]$ . Alors, la reconstruction affine peut être transformée en une reconstruction métrique en appliquant une transformation 3D de la forme

$$H = \begin{bmatrix} A^{-1} & \\ & 1 \end{bmatrix}$$

où  $A$  est obtenue par une factorisation de Cholesky de l'équation  $AA^T = (M^T\omega M)^{-1}$ .

**Preuve** Sous la transformation  $H$ , la matrice de caméra  $P$  est transformé à la matrice  $P_M = PH^{-1} = [M_M | m_M]$ . Si  $H^{-1}$  est de la forme suivante :

$$H^{-1} = \begin{bmatrix} A & 0 \\ 0^T & 1 \end{bmatrix}$$

alors  $M_M = MA$ . Cependant, l'image de la conique absolue est reliée à la matrice de la caméra  $P_M$  d'un cadre Euclidien par la relation :

$$\omega^* = M_M M_M^T$$

Cela est à cause que la matrice de caméra peut être décomposée en  $M_M = KR$ , et d'après [50] (résultat 8.11-p210)  $\omega^* = \omega^{-1} = KK^T$ . En combinant ça avec  $M_M = MA$  on aura  $\omega^{-1} = MAA^T M^T$ , qui peut être réarrangé en  $AA^T = (M^T \omega M)^{-1}$ . Une valeur particulière de  $A$  qui satisfait cette relation est obtenue par la factorisation de Cholesky de  $(M^T \omega M)^{-1}$ . Cette dernière est garantie d'être positive, autrement, aucune matrice telle  $A$  existera, et la reconstruction métrique ne sera pas possible.

Cette approche de la reconstruction métrique se base sur l'identification d'image de la conique absolue. Il existe plusieurs façons pour faire ça et ils seront discutés dans ce qui suit. Trois sources de contraintes sur l'image de la conique absolue seront données, et en pratique, la combinaison de ces contraintes est utilisée.

**Les contraintes relevées de l'orthogonalité de la scène** Les paires des points de fuites,  $v_1$  et  $v_2$ , résultants de l'orthogonalité des lignes de la scène placent une seule contrainte linéaire sur  $\omega$  :

$$v_1^T \omega v_2 = 0$$

De même, un point de fuite  $v$  et une ligne de fuite  $l$  qui résultent d'une direction et un plan qui sont orthogonaux placent deux contraintes sur  $\omega$  :

$$l = \omega v$$

Un exemple commun est le point de fuite pour la direction verticale et une ligne de fuite depuis le plan de base horizontale. Enfin un plan de la scène pris en image qui contient des informations métriques comme une grille carrée, place deux contraintes sur  $\omega$ .

**Contraintes relevées de la connaissance des paramètres internes** Si la matrice de calibration d'une caméra est égale à  $K$ , alors l'image de la conique absolue est  $\omega = K^{-T}K^{-1}$ . Donc, la connaissance des paramètres internes contenues dans  $K$  peut être utilisée pour contraindre ou déterminer les éléments de  $\omega$ . Dans le cas où  $K$  a une valeur nulle pour le *SKEW* ( $s = 0$ )

$$\omega_{12} = \omega_{21} = 0$$

et si les pixels sont carrés (*SKEW* égale à 0 et  $\alpha_x = \alpha_y$ ) alors

$$\omega_{11} = \omega_{22}$$

**Contraintes relevées à partir des même caméras dans toutes les images** Une des propriétés de la conique absolue est que sa projection dans une image dépend seulement sur la matrice d'étalonnage de la caméra, et non pas sur la position ou l'orientation de la caméra. Dans ce cas où les deux caméras  $P$  et  $P'$  ont la même matrice d'étalonnage (ce qui veut dire généralement que les deux images ont été prises par la même caméra avec différentes poses) une qui a  $\omega = \omega'$ , qui veut dire que l'image de la conique absolue est la même sur les deux images.

Ayant suffisamment plusieurs images, on peut utiliser cette propriété pour obtenir une reconstruction métrique d'une reconstruction affine. Sachant que la conique absolue se situe sur le plan à l'infini, son image peut être transféré d'une vue à l'autre par l'homographie infinie. Cela implique une équation

$$\omega' = H_{\infty}^{-T} \omega H_{\infty}^{-1} \tag{6.7}$$

où  $\omega$  et  $\omega'$  sont les images de  $\Omega_{\infty}$  dans les deux vues. En formant ces deux équations, il est nécessaire d'avoir déjà une reconstruction affine, étant donné que l'homographie infinie doit être connue. Si  $\omega = \omega'$ , alors 6.7 donne une ensemble d'équations linéaires dans les entités de  $\omega$ . En général cet ensemble d'équations linéaires place quatre contraintes sur  $\omega$ , et sachant que  $\omega$  a 5 degrés de libertés, il n'est pas complètement déterminé. Cependant, en combinant ces équations linéaires avec ceux en dessus qui proviennent de l'orthogonalité de la scène ou la connaissance des paramètres internes,  $\omega$  peut être déterminé de façon unique. Effectivement, l'équation 6.7 peut être utilisée pour transférer les contraintes sur  $\omega$  vers des contraintes sur  $\omega'$ . La figure 6.17 montre un exemple d'une reconstruction métrique calculée en combinant les contraintes de cette

manière.

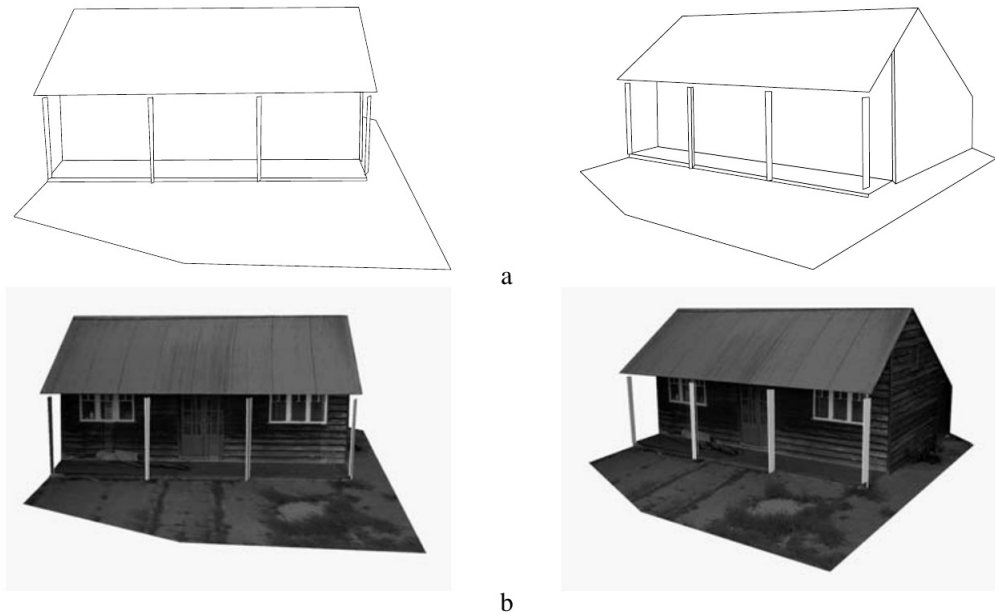


FIGURE 6.17: Reconstruction métrique

### 6.6.2 Reconstruction métrique directe en utilisant $\omega$

La discussion précédente a montré comment la connaissance de l'image de la conique absolue (IAC) peut être utilisée pour transformer une reconstruction affine à une reconstruction métrique. Cependant, sachant  $\omega$ , il est possible de passer directement à la reconstruction métrique, ayant au moins deux points de vue. Ceci peut être réalisé en au moins deux manières différentes. L'approche la plus évidente est d'utiliser l'IAC pour calculer l'étalonnage de chacune des caméras, puis procéder à une reconstruction calibrée. Cette méthode repose sur la connexion de  $\omega$  de la matrice d'étalonnage  $K$ , à savoir  $\omega = (KK^T)^{-1}$ . Ainsi on peut calculer  $K$  à partir de  $\omega$  en l'inversant et en appliquant ensuite la factorisation de Cholesky pour obtenir  $K$ . Si l'IAC est connue dans chaque image, les deux caméras peuvent être calibrées de cette manière. Ensuite avec des caméras calibrées, une reconstruction métrique de la scène peut être calculée en utilisant la matrice essentielle, comme montré dans la section 3.3.1.

Notez que quatre solutions possibles peuvent en résulter. Deux d'entre eux ne sont que des images miroirs, mais les deux autres sont différentes, formant une paire torsadée. Une approche plus conceptuelle à la reconstruction métrique est d'utiliser les connaissances de la IAC pour déterminer directement le plan à l'infini et la conique

absolue. Connaissant les matrices de la caméra  $P$  et  $P'$  dans un cadre projective, et une conique (plus précisément l'image de la conique absolue) dans chaque image, alors  $\Omega_\infty$  peut être calculée de façon explicite dans l'espace 3D.

Ce résultat est obtenu par rétro-projection des sections coniques en cônes, et qui doivent se croiser dans la conique absolue. Ainsi,  $\Omega_\infty$  et son plan de support  $\pi_\infty$  sont déterminées. Cependant, deux cônes seront généralement en intersection avec deux coniques de plans différents, chacun se trouvant dans un plan de support différent. Ainsi, il ya deux solutions possibles pour la conique absolue, une d'entre elles peut s'identifier comme appartenant à les deux reconstructions différentes qui constituent la paire torsadée d'ambiguïté.

### 6.6.3 La reconstruction directe - en utilisant la réalité de terrain

Il est possible de passer directement d'une reconstruction projective à une reconstruction métrique si des "points de contrôle au sol" sont donnés (c'est les points avec des emplacements 3D connus dans le cadre euclidien du monde réel). Supposons que nous avons un ensemble de  $n$  points de contrôle au sol  $X_{Ei}$  qui sont imagés au  $x_i \leftrightarrow x'_i$ . On souhaite utiliser ces points pour transformer la reconstruction projective en une métrique.

La position 3D de  $X_i$  des points de contrôles dans la reconstruction projective peut être calculée à partir de leurs correspondances d'image  $x_i \leftrightarrow x'_i$ . Étant donné que la reconstruction projective est liée par une homographie à la vraie reconstruction on a alors de l'équation 6.5 les équations :

$$X_{Ei} = HX_i, \quad i = 1, \dots, n.$$

Chaque correspondance de points fournit trois équations linéairement indépendantes sur les éléments de  $H$ , et puisque  $H$  a 15 degrés de liberté, une solution linéaire est obtenue à condition que  $n \geq 5$  (et il n'existe guère quatre points de contrôle qui sont coplanaires).

En variante, on peut contourner le calcul du  $X_i$  et calculer  $H$  en reliant les points de contrôle du sol directement à des mesures connues de l'image. Ainsi, comme dans

l'algorithme DLT pour la résection de la caméra, l'équation

$$x_i = PH^{-1}X_{Ei}$$

fournit deux équations linéairement indépendantes dans les entrées de  $H^{-1}$  qui est inconnue, toutes les autres grandeurs étant connues. De même, les équations peuvent être dérivées de l'autre image si  $x'_i$  est connu. Il n'est pas nécessaire que les points de contrôle au sol d'être visible dans les deux images. Notez cependant que si les deux  $x_i$  et  $x'_i$  sont visibles pour un point de contrôle donné  $X_{Ei}$  alors, à cause de la contrainte de coplanarité sur  $x$  et  $x'$ , les quatre équations générées de cette manière contiennent seulement trois équations indépendantes.

Une fois  $H$  a été calculée, elle peut être utilisée pour transformer les caméras  $P, P'$  de la reconstruction projective à leurs vraies homologues euclidiennes. Un exemple de la structure métrique calculée par cette méthode directe est montré dans la figure 6.18, où (a) montre les cinq points utilisés, (b) les points correspondants sur la reconstruction projective. (c) la reconstruction après que les cinq points ont été projetés vers leurs positions dans le monde réel.

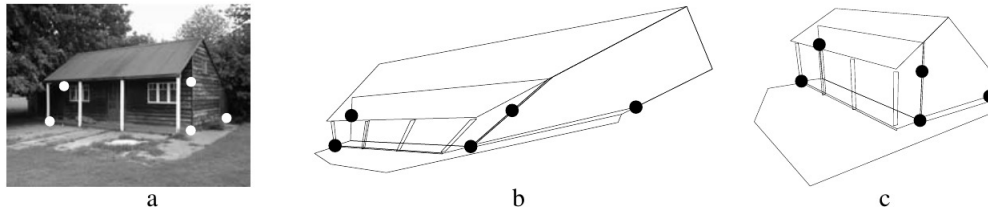


FIGURE 6.18: Reconstruction directe

Pour résumer [50] a élaborer l'algorithme qui est montré dans le tableau 6.1 qui illustre les étapes nécessaires pour passer d'une reconstruction projective vers une reconstruction métrique en utilisant des caméras non-calibrées.

## 6.6.4 Expérimentations

Les figures 6.19,6.21,6.20,6.22 montrent les résultat obtenus pour la reconstruction de quelques scènes prises en image par la même caméra depuis deux différents angles. La caméra n'est pas calibrée, donc la reconstruction 3D de la scène est une reconstruction stratifiée et passe par trois différente étape comme décrit dans le tableau

TABLE 6.1: Résumé des étapes pour avoir une reconstruction métrique à partir d'une reconstruction projective

- **Calculer une reconstruction projective**  $(P, P', X_i)$ 
  - Calculer la matrice fondamentale** depuis les correspondances des points  $x_i \leftrightarrow x'_i$  entre les images.
  - Déduction des caméras** : calculer les matrices de caméras  $P, P'$  depuis la matrice fondamentale.
  - La triangulation** : pour chaque correspondance de point  $x_i \leftrightarrow x'_i$ , calculer le point  $X_i$  dans l'espace qui se projette sur ces deux points.
- **Rectifier la reconstruction projective pour avoir une reconstruction métrique**
  - Soit utiliser **la méthode directe** : calculer l'homographie  $H$  pour que  $X_{Ei} = HX_i$  depuis cinq points de contrôles  $X_{Ei}$  ou plus qui ont une position Euclidienne connue. Ensuite la reconstruction métrique est :

$$P_M = PH^{-1}, \quad P'_M = P'H^{-1}, \quad X_{Mi} = HX_i$$

ou la méthode stratifiée :

**Une reconstruction affine** : Calculer le plan à l'infini  $\pi_\infty$ , comme décrit auparavant, ensuite mettre à jour la reconstruction projective vers une reconstruction affine avec l'homographie suivante :

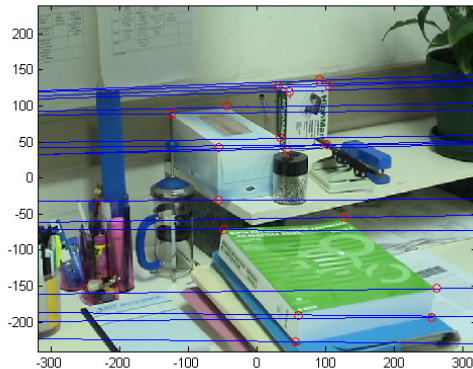
$$H = \begin{bmatrix} I & | & 0 \\ \hline & & \pi_\infty^T \end{bmatrix}$$

**La reconstruction métrique** : Calculer l'image de la conique absolue  $\omega$ , comme décrit auparavant, et ensuite transformer la reconstruction affine en une reconstruction métrique avec l'homographie suivante :

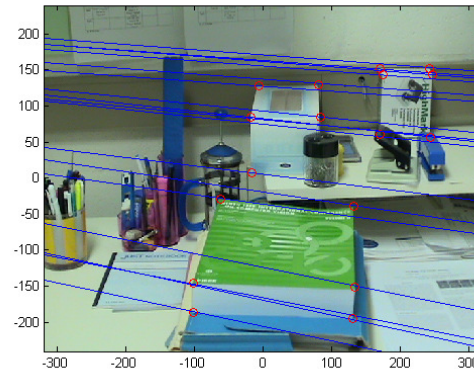
$$H = \begin{bmatrix} A^{-1} & \\ & 1 \end{bmatrix}$$

où  $A$  est obtenue avec la factorisation de Cholesky de l'équation  $AA^T = (M^T\omega M)^{-1}$ , et  $M$  est la première sous matrice  $3 \times 3$  de la caméra dans la reconstruction affine pour laquelle  $\omega$  est calculé

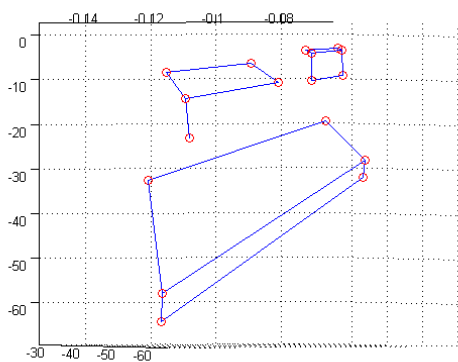
6.1. On a négliger la troisième partie car elle repose sur plus d'images pour avoir une reconstruction Euclidienne.



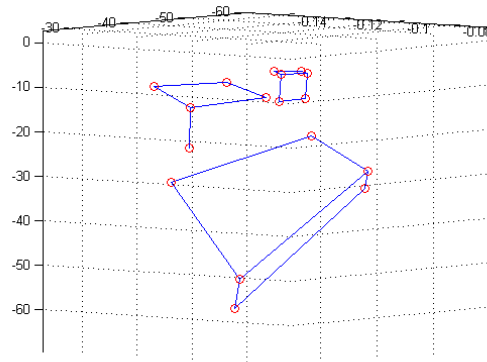
(a) Lignes épipolaires correspondants aux points de la deuxième image



(b) Lignes épipolaires correspondants aux points de la première image

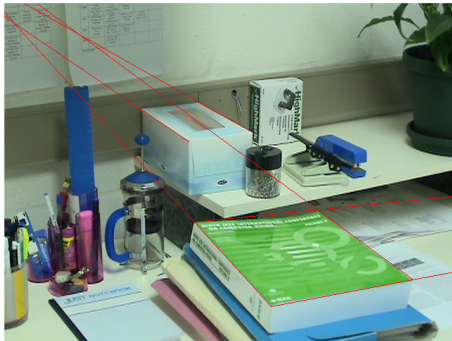


(c) Reconstruction Projective

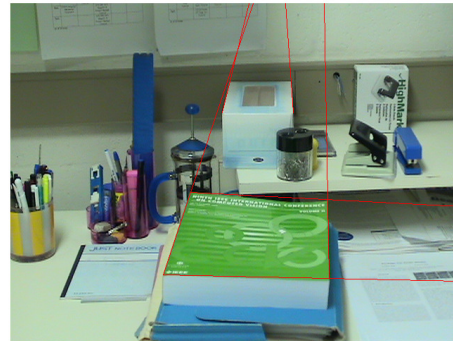


(d) Un autre point de vue de la Reconstruction Projective

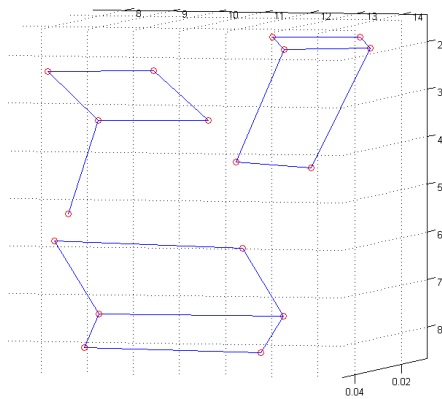
FIGURE 6.19: Reconstruction 3D projective de la scène des livres en utilisant une caméra non-calibrée



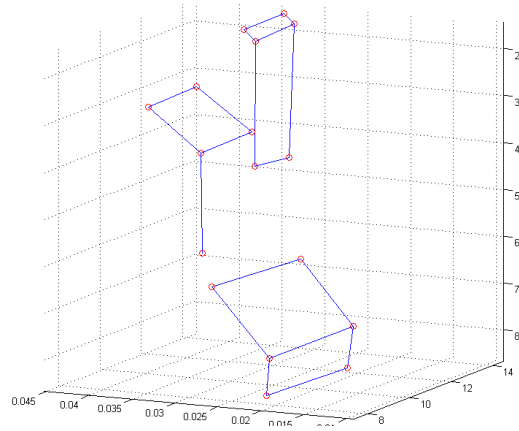
(a) Calcul et traçage des points de fuites des lignes parallèles dans la première image



(b) Calcul et traçage des points de fuites des lignes parallèles dans la deuxième image

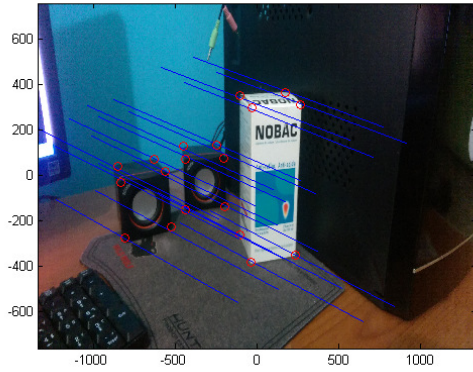


(c) Reconstruction Affine

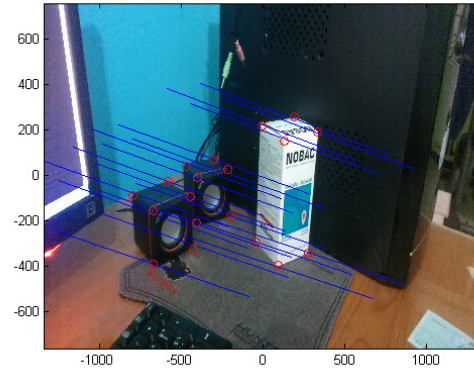


(d) Un autre point de vue de la Reconstruction Affine

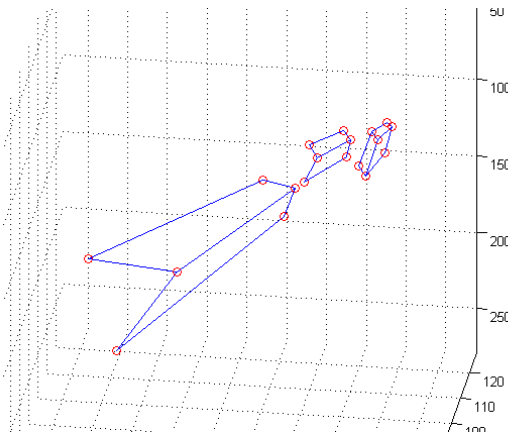
FIGURE 6.20: Reconstruction 3D Affine de la scène des livres en utilisant la contrainte des lignes parallèles



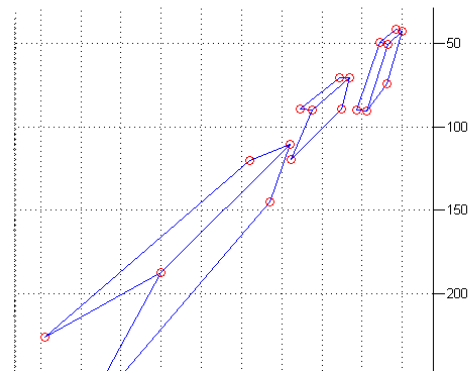
(a) Lignes épipolaires correspondant aux points de la deuxième image



(b) Lignes épipolaires correspondant aux points de la première image

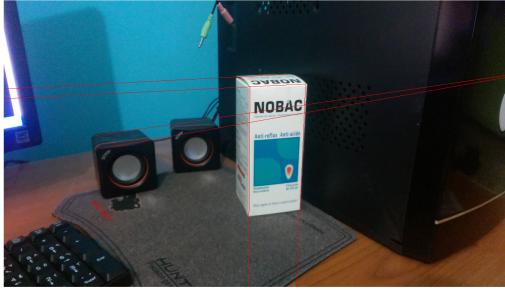


(c) Reconstruction Projective



(d) Un autre point de vue de la Reconstruction Projective

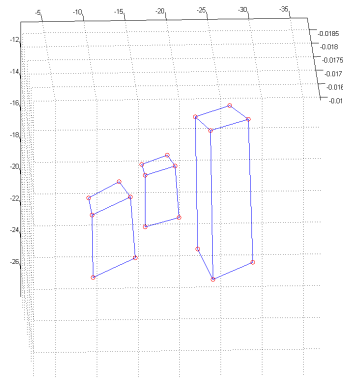
FIGURE 6.21: Deuxième exemple d'une reconstruction projective 3D de la scène du médicament



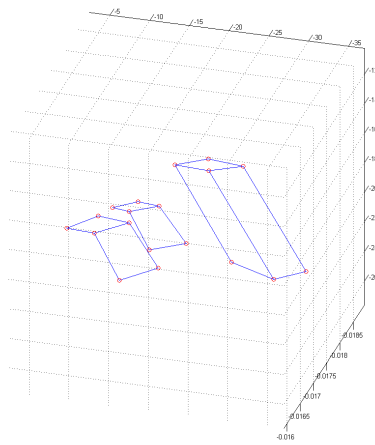
(a) Calcul et traçage des points de fuite des lignes parallèles dans la première image



(b) Calcul et traçage des points de fuite des lignes parallèles dans la deuxième image



(c) Reconstruction Affine



(d) Un autre point de vue de la Reconstruction Affine

FIGURE 6.22: Reconstruction 3D Affine de la scène du médicament en utilisant la contrainte des lignes parallèles

# Conclusion

Dans ce travail, nous avons essayé de proposer une méthode de reconstruction 3D à partir des images 2D. Notre objectif été d'introduire des techniques d'optimisation bio-inspirés pour résoudre un sous-problème pertinent . Étant donné que la reconstruction est un problème composé en plusieurs étapes, notamment la StéréoVision, le calcul de l'Homographie ainsi que la calibration de la caméra, nous avons réussi à améliorer le calcul de la transformation projectif dite Homographie en utilisant l'Optimisation par essaim de particules. Autrement dit, c'est une information présente dans tout le processus de reconstruction 3D pour : calculer les paramètres intrinsèques et extrinsèques de la caméras, pour calculer la projection nécessaire de l'épipôle vers l'infini dans la géométrie épipolaire, et aussi pour pouvoir faire la rectification des images stéréos. Les résultats présentés dans Le chapitre Homographie montrent une amélioration considérable par rapport à l'approche classique qui utilise la décomposition en valeurs singulière pour avoir une solution dans les moindres carrés.

L'optimisation par essaim de particules est une métaheuristique qui a prouver son efficacité pour plusieurs problèmes d'optimisation non-linéaires, et comme l'Homographie l'en est, on a essayer de tirer le meilleur de cette métaheuristique. En sachant que les méthodes analytiques proposés auparavant pour résoudre l'Homographie nécessitent une minimisation de l'erreur après le calcul initial de la matrice, la PSO ne le requiert pas due qu'elle est une optimisation qui utilise une fonction de minimisation dite fonction de fitness. Les résultats de la partie de la calibration des caméras montrent à elle une des utilisation de l'Homographie pour calculer les paramètres intrinsèques et extrinsèques des caméras. Les résultats illustrés dans cette partie valident l'approche proposée et font avancer le processus de la reconstruction 3D vers l'étape finale. Cette dernière n'est que la combinaison des trois premières parties. La seule tache qui restait à réaliser était l'utilisation des résultats obtenus auparavant pour produire une reconstruction 3D de la scène. Cette tâche doit améliorer une première reconstruction projective pour avoir une dite affine, pour ensuite la finaliser pour arriver à une reconstruction métrique

ou en d'autres termes une reconstruction Euclidienne.

Les résultats de ce travail ont été présenté à la fin pour conclure et mettre en évidence notre implémentation. On fini ce travail par dire qu'on a réussi à avoir une reconstruction 3D à partir des images obtenues de deux caméras différentes ou une caméra pour plusieurs points de vues, en utilisant un échiquier pour calibrer nos caméras, et l'optimisation par essaim de particule pour avoir l'Homographie, pour enfin utiliser la géométrie épipolaire pour calculer l'information perdue durant le processus de photographie qui est la profondeur.

## Perspectives

Le travail réalisé dans cette thèse ne fait que le début pour une recherche plus approfondie sur le domaine de la reconstruction 3D. Sachant que l'utilisation d'un système stéréo peut être un peu restreint due à l'équipement nécessaire et la mise en scène, des travaux future seront consacrés à l'élimination de ces contraintes en utilisant des équipement plus sophistiqués comme le Kinect, ou encore les smartphones étant donné qu'ils sont à la portée de tout le monde et qu'ils sont équipés de plusieurs capteurs, par exemple le gyroscope qui peut aider à calibrer la caméra.

Une suite de ce travail va être consacrée pour l'interprétation des contours flous. L'utilisation des règles d'inférences floues sur l'image nous a produit des contours épais pour les objets lointains et des contours fins pour les objets qui sont près de la caméra. On travaille actuellement sur une manière de quantifier cette information pour enfin l'utiliser afin d'avoir une carte de profondeur floue. La structure tridimensionnelle de la scène peut être déduite de cette carte. Une autre possibilité que nous envisageons est d'analyser les zones éclairées et sombres issues des sources de lumières, car celles-ci produit un effet de relief dans la scène observée.

ieetr

# Bibliographie

- [1] AGANJ, E. et al. “Spatio-temporal shape from silhouette using four-dimensional delaunay meshing”. In : *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, p. 1–8.
- [2] AGARWAL, A., JAWAHAR, C. V. et NARAYANAN, P. J. “A Survey of Planar Homography Estimation Techniques”. In : *CiteSeerX - Scientific Literature Digital Library and Search Engine* (2005).
- [3] AGUIAR, E. de et al. “M 3: marker-free model reconstruction and motion tracking from 3D voxel data”. In : *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*. IEEE. 2004, p. 101–110.
- [4] ALBITAR, C., GRAEBLING, P. et DOIGNON, C. “Robust Structured Light Coding for 3D Reconstruction.” In : *ICCV*. 2007, p. 1–6.
- [5] ALOIMONOS, J. “Shape from texture”. In : *Biological cybernetics* 58.5 (1988), p. 345–360.
- [6] AMÉDÉE, S. et FRANCOIS-GÉRARD, R. *ALGORITHMES GENETIQUES*. Rapp. tech. École normale supérieure de Lyon, 2004.
- [7] ASADA, N., FUJIWARA, H. et MATSUYAMA, T. “Edge and depth from focus”. In : *International Journal of Computer Vision* 26.2 (1998), p. 153–163.
- [8] AUTIN, B. “Les métaheuristiques en optimisation combinatoire”. Mém.de mast. Conservatoire National des Arts et Métiers, Paris, 2006.
- [9] BAKER, H. H. *Depth from Edge and Intensity Based Stereo*. Rapp. tech. DTIC Document, 1982.
- [10] BAUMGART, B. G. “A polyhedron representation for computer vision”. In : *Proceedings of the May 19-22, 1975, national computer conference and exposition*. ACM. 1975, p. 589–596.
- [11] BENOÎT, L. *Enveloppes Visuelles et Reconstruction 3D*. 2009.

- [12] BOGOMJAKOV, A. et GOTSMAN, C. “Reduced depth and visual hulls of complex 3D scenes”. In : *Computer Graphics Forum*. T. 27. 2. Wiley Online Library. 2008, p. 175–182.
- [13] BOUFAMA, B et JIN, K. “Towards a fast and reliable dense matching algorithm”. In : *TECHNICAL PAPERS-SOCIETY OF MANUFACTURING ENGINEERS-ALL SERIES-* (2003).
- [14] BRANDOU, V. “Stéréovision Locale et Reconstruction 3D/4D”. Thèse de doct. Université de Nice-Sophia Antipolis, 2008.
- [15] BROWN, D. C. “Close-range camera calibration”. In : *Photogrammetric Engineering* 37.8 (1971), p. 855–866.
- [16] CAILLETTE, F. et HOWARD, T. “Real-time markerless human body tracking using colored voxels and 3D blobs”. In : *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*. IEEE. 2004, p. 266–267.
- [17] CAPRILE, B. et TORRE, V. “Using Vanishing Points for Camera Calibration”. In : *Int. J. Comput. Vision* 4.2 (mai 1990), p. 127–140.
- [18] CHAUDHURI, S. et RAJAGOPALAN, A. N. *Depth from defocus - a real aperture imaging approach*. Springer, 1999, p. I–XVIII, 1–172.
- [19] CHEUNG, G. K., BAKER, S. et KANADE, T. “Visual hull alignment and refinement across time: A 3D reconstruction algorithm combining shape-from-silhouette with stereo”. In : *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. T. 2. IEEE. 2003, p. II–375.
- [20] CHEUNG, G. K. et al. “A real time system for robust 3D voxel reconstruction of human motions”. In : *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. T. 2. IEEE. 2000, p. 714–720.
- [21] CHEUNG, K., BAKER, S. et KANADE, T. “Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture”. In : *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. T. 1. IEEE. 2003, p. I–77.
- [22] COLORNI, A., DORIGO, M., MANIEZZO, V. et al. “Distributed optimization by ant colonies”. In : *Proceedings of the first European conference on artificial life*. T. 142. Paris, France. 1991, p. 134–142.

- [23] COLORNI, A., DORIGO, M., MANIEZZO, V. et al. “An Investigation of some Properties of an “Ant Algorithm”.” In : *PPSN 92* (1992), p. 509–520.
- [24] COSTA, D. et HERTZ, A. “Ants can colour graphs”. In : *Journal of the operational Research Society* 48.3 (1997), p. 295–305.
- [25] CULBERTSON, W. B., MALZBENDER, T. et SLABAUGH, G. “Generalized voxel coloring”. In : *Vision Algorithms: Theory and Practice*. Springer, 2000, p. 100–115.
- [26] DARELL, T. et WOHN, K. “Depth from focus using a pyramid architecture”. In : *Pattern Recognition Letters* 11.12 (1990), p. 787–796.
- [27] DIPANDA, A. et WOO, S. “Towards a real-time 3D shape reconstruction using a structured light system”. In : *Pattern recognition* 38.10 (2005), p. 1632–1650.
- [28] DORIGO, M., MANIEZZO, V. et COLORNI, A. “Ant system: optimization by a colony of cooperating agents”. In : *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 26.1 (1996), p. 29–41.
- [29] DOURI, S. M., ELBERNOUSSI, S. et LAKHBAB, H. *Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques*.
- [30] DUTOT, A. et OLIVIER, D. *Optimisation par essaim de particules Application au problème des n-Reines*. Rapp. tech. Laboratoire Informatique du Havre. Université du Havre, 2002.
- [31] DYER, C. R. “Volumetric scene reconstruction from multiple views”. In : *Foundations of image understanding*. Springer, 2001, p. 469–489.
- [32] FAUGERAS, O. *Three-dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA, USA : MIT Press, 1993.
- [33] FAUGERAS, O. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.
- [34] FEIGENBAUM, E. A. et FELDMAN, J. *Computers and Thought*. New York, NY, USA : McGraw-Hill, Inc., 1963.
- [35] FITZGIBBON, A. W., CROSS, G. et ZISSERMAN, A. “Automatic 3D model construction for turn-table sequences”. In : *3D Structure from Multiple Images of Large-Scale Environments*. Springer, 1998, p. 155–170.
- [36] GAMBARDELLA, L. M., TAILLARD, E. et DORIGO, M. “Ant colonies for the quadratic assignment problem”. In : *Journal of the operational research society* (1999), p. 167–176.

- [37] GÅRDING, J. “Shape from texture for smooth curved surfaces in perspective projection”. In : *Journal of Mathematical Imaging and Vision* 2.4 (1992), p. 327–350.
- [38] GHERBOUDJ, A. “Méthodes de résolution de problèmes difficiles académiques”. Thèse de doct. Université de Constantine 2, 2013.
- [39] GLOVER, F. “Future Paths for Integer Programming and Links to Artificial Intelligence”. In : *Comput. Oper. Res.* 13.5 (mai 1986), p. 533–549.
- [40] GLOVER, F. et LAGUNA, M. *Tabu Search*. Norwell, MA, USA : Kluwer Academic Publishers, 1997.
- [41] GOKTURK, S. B., YALCIN, H. et BAMJI, C. “A time-of-flight depth sensor-system description, issues and solutions”. In : *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*. IEEE. 2004, p. 35–35.
- [42] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1989.
- [43] GOLDLUECKE, B. et MAGNOR, M. “Space-time isosurface evolution for temporally coherent 3D reconstruction”. In : *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. T. 1. IEEE. 2004, p. I–350.
- [44] GOLDLUECKE, B. et MAGNOR, M. “Space-time isosurface evolution for temporally coherent 3D reconstruction”. In : *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. T. 1. IEEE. 2004, p. I–350.
- [45] GRAU, O. “Studio production system for dynamic 3D content”. In : *Visual Communications and Image Processing 2003*. International Society for Optics et Photonics. 2003, p. 80–89.
- [46] GROSSO, E. et TISTARELLI, M. “Active/dynamic stereo vision”. In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 17.9 (1995), p. 868–879.
- [47] GUILLOU, E. et al. “Using vanishing points for camera calibration and coarse 3D reconstruction from a single image”. In : *The Visual Computer* 16.7 (2000), p. 396–410.

- [48] HANSEN, P. “The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming”. In : *Proceedings of the Congress on Numerical Methods in Combinatorial Optimization*. Capri, Italy, 1986.
- [49] HARRIS, C. et STEPHENS, M. “A combined corner and edge detector.” In : *Alvey vision conference*. T. 15. Manchester, UK. 1988, p. 50.
- [50] HARTLEY, R. et ZISSERMAN, A. “Multiple View Geometry in computer vision”. In : *Cambridge University Press* (2003), p. 32–33.
- [51] HARTLEY, R. I. “Cheirality invariants”. In : *Proc. DARPA Image Understanding Workshop*. Citeseer. 1993, p. 745–753.
- [52] HARTLEY, R. I. “Self-calibration from Multiple Views with a Rotating Camera”. In : *Proceedings of the Third European Conference on Computer Vision (Vol. 1)*. ECCV '94. Stockholm, Sweden : Springer-Verlag New York, Inc., 1994, p. 471–478.
- [53] HARTLEY, R. I. “In Defense of the Eight-Point Algorithm”. In : *IEEE Trans. Pattern Anal. Mach. Intell.* 19.6 (juin 1997), p. 580–593.
- [54] HARTLEY, R. I. “Theory and practice of projective rectification”. In : *International Journal of Computer Vision* 35.2 (1999), p. 115–127.
- [55] HAYES-ROTH, F. “Review of "Adaptation in Natural and Artificial Systems by John H. Holland", The U. Of Michigan Press, 1975”. In : *SIGART Bull.* 53 (1975), p. 15–15.
- [56] KENNEDY, J et EBERHART, R. “Particle swarm optimization”. In : *Neural Networks, 1995. Proceedings., IEEE International Conference on*. T. 4. IEEE. 1995, p. 1942–1948.
- [57] KENNEDY, J. et MENDES, R. “Population structure and particle swarm performance”. In : (2002).
- [58] KIRKPATRICK, S., GELATT, C. D. et VECCHI, M. P. “Optimization by simulated annealing”. In : *SCIENCE* 220.4598 (1983), p. 671–680.
- [59] KORDELAS, G. et al. “State-of-the-art Algorithms for Complete 3D Model Reconstruction,”. In : *Engage” Summer School* (2010), p. 1–15.
- [60] KOSCHAN, A. et al. *What is new in computational stereo since 1989: A survey on current stereo papers*. Technische Universität Berlin, Fachbereich 20, Informatik, 1993.

- [61] KUTULAKOS, K. N. et SEITZ, S. M. “A theory of shape by space carving”. In : *International Journal of Computer Vision* 38.3 (2000), p. 199–218.
- [62] LAURENTINI, A. “The visual hull concept for silhouette-based image understanding”. In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16.2 (1994), p. 150–162.
- [63] LAURENTINI, A. “The visual hull concept for silhouette-based image understanding”. In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16.2 (1994), p. 150–162.
- [64] LEE, C., CHO, J. et OH, K. “Hardware-accelerated jaggy-free visual hulls with silhouette maps”. In : *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM. 2006, p. 87–90.
- [65] LI, M., MAGNOR, M. et SEIDEL, H.-P. “A hybrid hardware-accelerated algorithm for high quality rendering of visual hulls”. In : *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society. 2004, p. 41–48.
- [66] LIEBOWITZ, D. et ZISSERMAN, A. “Metric Rectification for Perspective Images of Planes”. In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR '98. Washington, DC, USA : IEEE Computer Society, 1998, p. 482–.
- [67] LOH, A. M. et HARTLEY, R. “Shape from Non-homogeneous, Non-stationary, Anisotropic, Perspective Texture.” In : *BMVC*. Citeseer. 2005.
- [68] LONGUET-HIGGINS, H. C. “Readings in Computer Vision: Issues, Problems, Principles, and Paradigms”. In : sous la dir. de FISCHLER, M. A. et FIRSCHEIN, O. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1987. Chap. A Computer Algorithm for Reconstructing a Scene from Two Projections, p. 61–62.
- [69] LUONG, Q.-T. et VIÉVILLE, T. “Canonical representations for the geometries of multiple projective views”. In : *Computer Vision and Image Understanding* 64.2 (1996), p. 193–229.
- [70] MAGNOR, M. et SEIDEL, H.-P. “Capturing the shape of a dynamic world-fast!” In : *Shape Modeling International, 2003*. IEEE. 2003, p. 3–9.
- [71] MARR, D. et POGGIO, T. “A theory of human stereo vision”. In : (1977).
- [72] MATUSIK, W. et al. “Image-based visual hulls”. In : *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 2000, p. 369–374.

- [73] MAZZEO, S. et LOISEAU, I. “An ant colony algorithm for the capacitated vehicle routing”. In : *Electronic Notes in Discrete Mathematics* 18 (2004), p. 181–186.
- [74] METROPOLIS, N. et al. “Equation of State Calculations by Fast Computing Machines”. In : *The Journal of Chemical Physics* 21.6 (1953), p. 1087–1092.
- [75] MEYER, A., BRICEÑO, H. M. et BOUAKAZ, S. “User-guided shape from shading to reconstruct fine details from a single photograph”. In : *Computer Vision–ACCV 2007*. Springer, 2007, p. 738–747.
- [76] MICHOD, B. “Reconstruction 3D à partir de séquences vidéo pour l’acquisition du mouvement de personnages en temps réel et sans marqueur”. Thèse de doct. Université Claude Bernard – Lyon 1, 2009.
- [77] MLADENOVIC, N. et HANSEN, P. “Variable neighborhood search”. In : *Computers & Operations Research* 24.11 (1997), p. 1097–1100.
- [78] MODROW, D. et al. “3d face scanning systems based on invisible infrared coded light”. In : *Advances in Visual Computing*. Springer, 2007, p. 521–530.
- [79] MORÉ, J. “The Levenberg-Marquardt algorithm: Implementation and theory”. English. In : *Numerical Analysis*. Sous la dir. de WATSON, G. T. 630. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1978, p. 105–116.
- [80] MÜHLMANN, K. et al. “Calculating dense disparity maps from color stereo images, an efficient implementation”. In : *International Journal of Computer Vision* 47.1-3 (2002), p. 79–88.
- [81] NEWELL, A., SHAW, J. C. et SIMON, H. A. “Empirical explorations of the logic theory machine: a case study in heuristic”. In : *Papers presented at the February 26-28, 1957, western joint computer conference: Techniques for reliability*. ACM, 1957, p. 218–230.
- [82] NIEM, W. “Robust and fast modeling of 3D natural objects from multiple views”. In : *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*. International Society for Optics et Photonics, 1994, p. 388–397.
- [83] OSMAN, I. H. et LAPORTE, G. “Metaheuristics: A bibliography”. In : *Annals of Operations research* 63.5 (1996), p. 511–623.
- [84] PAYET, N. et TODOROVIC, S. “Scene shape from texture of objects”. In : *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, p. 2017–2024.

- [85] PRADOS, E. et FAUGERAS, O. “Shape from shading: a well-posed problem?” In : *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. T. 2. IEEE. 2005, p. 870–877.
- [86] PRADOS, E. et FAUGERAS, O. “Shape from shading”. In : *Handbook of mathematical models in computer vision*. Springer, 2006, p. 375–388.
- [87] SALVI, J., PAGES, J. et BATLLE, J. “Pattern codification strategies in structured light systems”. In : *Pattern Recognition 37.4* (2004), p. 827–849.
- [88] SEITZ, S. M. et DYER, C. R. “Photorealistic scene reconstruction by voxel coloring”. In : *International Journal of Computer Vision 35.2* (1999), p. 151–173.
- [89] SHIMIZU, I. et al. “Head Pose Determination from One Image Using a Generic Model”. In : *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*. FG '98. Washington, DC, USA : IEEE Computer Society, 1998, p. 100–.
- [90] SINHA, S. N. “Silhouettes for Calibration and Reconstruction from Multiple Views”. Thèse de doct. UNC Chapel Hill, 2008.
- [91] SLABAUGH, G. et al. “A survey of methods for volumetric scene reconstruction from photographs”. In : *Proceedings of the 2001 Eurographics conference on Volume Graphics*. Eurographics Association. 2001, p. 81–101.
- [92] STEIN, G. “Accurate internal camera calibration using rotation, with analysis of sources of error”. In : *Computer Vision, 1995. Proceedings., Fifth International Conference on*. 1995, p. 230–236.
- [93] STURM, P. *Quelques notes pour le cours de Vision par Ordinateur*.
- [94] STURM, P. F. et MAYBANK, S. J. “On plane-based camera calibration: A general algorithm, singularities, applications”. In : 1999, p. 432–437.
- [95] SZELISKI, R. et ZABIH, R. “An experimental comparison of stereo algorithms”. In : *Vision Algorithms: Theory and Practice*. Springer, 2000, p. 1–19.
- [96] TIRUMALAI, A. P., SCHUNCK, B. G. et JAIN, R. C. “Dynamic stereo with self-calibration”. In : *Computer Vision, 1990. Proceedings, Third International Conference on*. IEEE. 1990, p. 466–470.
- [97] TSAI, R. Y. “Radiometry”. In : sous la dir. de WOLFF, L. B., SHAFER, S. A. et HEALEY, G. USA : Jones et Bartlett Publishers, Inc., 1992. Chap. A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses, p. 221–244.

- [98] VAN DEN BERGH, F. “An analysis of particle swarm optimizers”. Thèse de doct. University of Pretoria, 2006.
- [99] VEDULA, S., BAKER, S. et KANADE, T. “Spatio-temporal view interpolation”. In : *Proceedings of the 13th Eurographics workshop on Rendering*. Eurographics Association. 2002, p. 65–76.
- [100] WEI, G.-Q. et MA, S. D. “Implicit and Explicit Camera Calibration: Theory and Experiments.” In : *IEEE Trans. Pattern Anal. Mach. Intell.* 16.5 (1994), p. 469–480.
- [101] WONG, K.-Y. K., MENDONÇA, P. R. et CIPOLLA, R. “Head model acquisition from silhouettes”. In : *Visual Form 2001*. Springer, 2001, p. 787–796.
- [102] XU, G. et ZHANG, Z. *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Norwell, MA, USA : Kluwer Academic Publishers, 1996.
- [103] YANG, D. B., GONZÁLEZ-BAÑOS, H. H. et GUIBAS, L. J. “Counting people in crowds with a real-time network of simple image sensors”. In : *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003, p. 122–129.
- [104] ZHANG, Z. “A flexible new technique for camera calibration”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), p. 1330–1334.
- [105] ZHANG, Z. “Camera Calibration with One-Dimensional Objects”. In : *Proceedings of the 7th European Conference on Computer Vision-Part IV*. ECCV '02. London, UK, UK : Springer-Verlag, 2002, p. 161–174.
- [106] ZHANG, Z. “Camera calibration”. In : *Emerging topics in computer vision*. Prentice Hall Professional Technical Reference, 2004. Chap. 2, p. 4–43.
- [107] ZHAO, C. et MOHR, R. “Global three-dimensional surface reconstruction from occluding contours”. In : *Computer Vision and Image Understanding* 64.1 (1996), p. 62–96.
- [108] ZHAO, C. et MOHR, R. “Global three-dimensional surface reconstruction from occluding contours”. In : *Computer Vision and Image Understanding* 64.1 (1996), p. 62–96.
- [109] ČERNÝ, V. “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm”. English. In : *Journal of Optimization Theory and Applications* 45.1 (1985), p. 41–51.

# Productions Scientifiques

- TALAI, Z. ; TALAI, A. A fast edge detection using fuzzy rules. The International Arab Conference on Information Technology (ACIT). (13-16 Décembre 2010).
- TALAI, Z. ; TALAI, A. An Optimized Edge Detector Based on Fuzzy Logic. In Communications, Computing and Control Applications (CCCA), 2011 IEEE International Conference on (pp. 1-5), (Mars 2011).
- TALAI, Z. ; MOHAMED BEN ALI, Y. Homography Problem Resolution Using Particle Swarm optimization. Complex Systems (ICCS), 2012 International Conference on, (5-6 Novembre 2012).
- TALAI, Z. ; MOHAMED BEN ALI, Y. Homography Resolution Using Particle Swarm Optimization. International Journal of Robotics and Automation, 2015, vol. 30, no 2.