

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Image, Information et Hypermédia

Présentée et soutenue par :

M. LILIAN CALVET

le jeudi 23 janvier 2014

Titre :

METHODES DE RECONSTRUCTION TRIDIMENSIONNELLE INTEGRANT DES POINTS CYCLIQUES : APPLICATION AU SUIVI DE CAMERA

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

M. VINCENT CHARVILLAT

M. PIERRE GURDJOS

Rapporteurs :

M. ADRIEN BARTOLI, UNIVERSITE CLERMONT-FERRAND 1

M. RICHARD HARTLEY, AUSTRALIAN NATIONAL UNIVERSITY CANBERRA

Membre(s) du jury :

M. PETER STURM, INRIA GRENOBLE - RHONE ALPES, Président

M. DAVID FOFI, UNIVERSITE DE BOURGOGNE, Membre

M. PIERRE GURDJOS, CNRS, Membre

M. VINCENT CHARVILLAT, INP TOULOUSE, Membre

À mes parents.

Remerciements

Je remercie tout d'abord Vincent Charvillat pour m'avoir offert l'opportunité de réaliser cette thèse au sein de son équipe. Je remercie la région Midi-Pyrénées et la communauté de communes de Saint-Laurent-de-Neste pour avoir financé ces travaux. Je souhaite également remercier l'agence nationale de la recherche pour le soutien dont a pu bénéficier le laboratoire durant le projet ANR-ROM.

Je remercie mes encadrants Pierre Gurdjos et Vincent Charvillat pour le soutien dont ils ont fait preuve tout au long de cette thèse. C'est avec grand plaisir que j'ai bénéficié de leurs qualités prédagogiques, scientifiques et humaines qui m'ont permises de mener à bien ces travaux de recherche. Un immense merci à Pierre qui m'a guidé tout au long de cette thèse avec beaucoup d'enthousiasme. Je tiens à le remercier pour sa disponibilité, les nombreuses discussions que nous avons eues, son partage de connaissances et sa rigueur scientifique.

Je remercie Adrien Bartoli et Richard Hartley pour avoir accepté d'évaluer mon manuscrit, pour leurs remarques à la fois critiques et constructives. Je tiens également à remercier David Fofi et Peter Sturm pour l'intérêt qu'ils ont porté à ce travail en acceptant d'en être les examinateurs.

Je remercie les membres permanents : Simone Gasparini et Jean-Denis Durou (surnommé « Maître Capello ») pour leurs précieuses relectures, Sylvie Chambon et Géraldine Morin pour leur soutien. Je remercie Sylvie Eichen et Sylvie Armengaud pour leur support lors des missions et dossiers administratifs. Je remercie l'ensemble des membres de l'équipe pour la bonne humeur apportée au laboratoire : Jean-Denis D. pour ses immanquables pauses café, ses « disciples » Yvain Q., Bastien D., Benjamin R. et Nicolas B., mes collègues de bureau Axel C., Rabih A. et Florent B. avec qui j'ai partagé de très bons moments, Pauline J., Viorica P., Benoît B. et Jérome G.

Je tiens à remercier mes amis pour leur soutien et leurs encouragements : Marco K., Olive P., Mickou G., José P., Olive C., Fifou SBS., Mathieu B., Guillaume B., Nath B., Philou M., Marie M., Bastien C., Brice F., Margot L., Cyril D. et mes chers amis Salvetois.

Je souhaite enfin exprimer ma gratitude envers mes proches qui m'ont toujours encouragé, Monique et Louis, ma soeur Amandine, et une extrême reconnaissance envers mes parents pour m'avoir donné les moyens de réaliser mes études. Enfin, mes plus profonds remerciements vont à ma compagne Maeva pour sa patience, sa compréhension et son soutien.

Résumé

Cette thèse traite de la reconstruction tridimensionnelle d'une scène rigide à partir d'une collection de photographies numériques, dites vues. Le problème traité est connu sous le nom du « calcul de la structure et du mouvement » (structure-and/from-motion) qui consiste à « expliquer » des trajectoires de points dits d'intérêt au sein de la collection de vues par un certain mouvement de l'appareil (dont sa trajectoire) et des caractéristiques géométriques tridimensionnelles de la scène. Dans ce travail, nous proposons les fondements théoriques pour étendre certaines méthodes de calcul de la structure et du mouvement afin d'intégrer comme données d'entrée, des points d'intérêt réels et des points d'intérêt complexes, et plus précisément des images de points cycliques. Pour tout plan projectif, les points cycliques forment une paire de points complexes conjugués qui, par leur invariance par les similitudes planes, munissent le plan projectif d'une structure euclidienne. Nous introduisons la notion de marqueurs cycliques qui sont des marqueurs plans permettant de calculer sans ambiguïté les images des points cycliques de leur plan de support dans toute vue. Une propriété de ces marqueurs, en plus d'être très « riches » en information euclidienne, est que leurs images peuvent être appariées même si les marqueurs sont disposés arbitrairement sur des plans parallèles, grâce à l'invariance des points cycliques. Nous montrons comment utiliser cette propriété dans le calcul projectif de la structure et du mouvement via une technique matricielle de réduction de rang, dite de factorisation, de la matrice des données correspondant aux images de points réels, complexes et/ou cycliques. Un sous-problème critique abordé dans le calcul de la structure et du mouvement est celui de l'auto-calibrage de l'appareil, problème consistant à transformer un calcul projectif en un calcul euclidien. Nous expliquons comment utiliser l'information euclidienne fournie par les images des points cycliques dans l'algorithme d'auto-calibrage opérant dans l'espace projectif dual et fondé sur des équations linéaires. L'ensemble de ces contributions est finalement utilisé pour une application de suivi automatique de caméra utilisant des marqueurs formés par des couronnes concentriques (appelés C^2Tags), où il s'agit de calculer le mouvement tridimensionnel de la caméra dans la scène à partir d'une séquence vidéo. Ce type d'application est généralement utilisé dans l'industrie du cinéma ou de la télévision afin de produire des effets spéciaux. Le suivi de caméra proposé dans ce travail a été conçu pour proposer le meilleur compromis possible entre flexibilité d'utilisation et précision des résultats obtenus.

Abstract

The thesis deals with the problem of 3D reconstruction of a rigid scene from a collection of views acquired by a digital camera. The problem addressed, referred as the Structure-from-Motion (SfM) problem, consists in computing the camera motion (including its trajectory) and the 3D characteristics of the scene based on 2D trajectories of imaged features through the collection. We propose theoretical foundations to extend some SfM paradigms in order to integrate real as well as complex imaged features as input data, and more especially imaged circular points. Circular points of a projective plane consist in a complex conjugate point-pair which is fixed under plane similarity ; thus endowing the plane with an Euclidean structure. We introduce the notion of circular markers which are planar markers that allows to compute, without any ambiguity, imaged circular points of their supporting plane in all views. Aside from providing a very "rich" euclidean information, such features can be matched even if they are arbitrarily positioned on parallel planes thanks to their invariance under plane similarity ; thus increasing their visibility compared to natural features. We show how to benefit from this geometric property in solving the projective SfM problem via a rank-reduction technique, referred to as projective factorization, of the matrix whose entries are images of real, complex and/or circular features. One of the critical issues in such a SfM paradigm is the self-calibration problem, which consists in updating a projective reconstruction into an euclidean one. We explain how to use the euclidean information provided by imaged circular points in the self-calibration algorithm operating in the dual projective space and relying on linear equations. All these contributions are finally used in an automatic camera tracking application relying on markers made up of concentric circles (called C^2Tags). The problem consists in computing the 3D camera motion based on a video sequence. This kind of application is generally used in the cinema or TV industry to create special effects. The camera tracking proposed in this work in designed in order to provide the best compromise between flexibility of use and accuracy.

Table des matières

1 Introduction **21**

2 Notations et rappels géométriques **25**
2.1 Question de notations 26
2.2 Rappels de géométrie projective 27
2.2.1 L'espace projectif et son dual 28
2.2.2 Quadriques projectives 28
2.2.2.1 Conjugaison et polarité relatives à une quadrique 29
2.2.2.2 Tangence à une quadrique 29
2.2.3 Quadriques projectives duales 29
2.2.4 Quadriques dégénérées des espaces projectifs de dimensions deux et trois 30
2.2.4.1 Enveloppes des quadriques dégénérées lorsque $n \in \{2, 3\}$ . . . 31
2.2.5 Transformation d'une quadrique projective 31
2.2.6 Signature d'une quadrique projective 31
2.2.7 Stratifications projectives 33
2.2.7.1 Structure affine d'un espace projectif. Hyperplan à l'infini. . . 33
2.2.7.2 Structure affine euclidienne de l'espace projectif tridimensionnel. Conique absolue. 33

3 Calcul de la structure et du mouvement par factorisation projective incorporant des points cycliques **35**
3.1 Introduction 36
3.2 Calcul de la structure et du mouvement à partir de marqueurs cycliques 38
3.2.1 Marqueurs cycliques 39
3.2.2 Marqueurs cycliques appariés 41
3.3 État de l'art 42
3.4 Complexification de l'espace projectif réel 45
3.4.1 Pourquoi et comment complexifier un espace projectif réel ? 45
3.4.2 La paire de points cycliques d'un plan projectif 46
3.5 Le problème de la factorisation de points complexes 48

3.5.1 Les données . . . 49
3.5.2 Le problème . . . 50
3.5.3 Mise à l'échelle des blocs de la matrice des données . . . 51
3.5.3.1 Les grandes lignes . . . 51
3.5.3.2 Généralisation de la mise à l'échelle d'une donnée réelle à une donnée complexe . . . 52
3.5.4 Réduction du rang de la matrice des données . . . 52
3.6 Premier cas d'usage . . . 53
3.6.1 Factorisation d'images de $K > 1$ points cycliques . . . 53
3.6.1.1 Équation de mise à l'échelle . . . 54
3.6.1.2 Résolution . . . 56
3.6.1.3 Prédiction des blocs manquants de la matrice des données . . . 56
3.6.1.4 Heuristique de sélection de vues . . . 57
3.6.2 Reconstruction euclidienne post-factorisation de la structure et du mouvement . . . 57
3.7 Deuxième cas d'usage . . . 59
3.7.1 Factorisation d'images de points naturels et de points cycliques . . . 60
3.7.1.1 Équations de mise à l'échelle . . . 61
3.7.1.2 Résolution . . . 63
3.7.1.3 Algorithme de prédiction des blocs manquants de la matrice des données . . . 64
3.7.1.4 Heuristique de sélection de vues . . . 67
3.7.2 Reconstruction euclidienne post-factorisation de la structure et du mouvement . . . 67
3.8 Résultats expérimentaux . . . 68
3.8.1 Résultats du premier cas d'usage . . . 68
3.8.2 Résultats du deuxième cas d'usage . . . 68
3.9 Conclusion . . . 70

4 Rectification euclidienne d'une reconstruction projective 73
4.1 Introduction . . . 74
4.1.1 Notre problème spécifique . . . 75
4.2 État de l'art . . . 76
4.3 Autocalibrage dans l'espace projectif dual . . . 79
4.3.1 Formulation linéaire de [Pollefeys 1999] . . . 79
4.3.1.1 Les équations de base . . . 80
4.3.1.2 Résolution avec contrainte de signature $(3, 0)$ *a posteriori* . . . 80
4.3.2 Formation linéaire étendue intégrant une paire de points cycliques . . . 81
4.3.2.1 Points cycliques et contraintes d'autocalibrage . . . 81
4.3.2.2 Les équations du problème intégrant une paire de points cycliques 82
4.3.3 Équations de base revisitées . . . 85

4.3.3.1 Algorithme d'autocalibrage unifié 86
4.4 Résultats expérimentaux 89
4.4.1 Résultats sur données synthétiques 89
4.4.2 Résultats sur données réelles 90
4.5 Conclusion 95

5 Le système de marqueurs C²Tags 97
5.1 Introduction 98
5.2 État de l'art 99
5.2.1 Marqueurs 0D 99
5.2.2 Marqueurs 2D 101
5.2.3 Les codes-barres 102
5.3 Le problème et ses motivations 103
5.4 La solution proposée : le motif C²Tag 106
5.4.1 Un marqueur circulaire « idéal » 106
5.4.2 Le C²Tag, support des équipotentielles du motif « idéal » 110
5.4.3 Aperçu du système de détection 113
5.4.4 Détection 114
5.4.4.1 Système de vote pour l'ellipse interne 114
5.4.4.2 Regroupement des candidats en segments de contour de l'ellipse interne 115
5.4.4.3 Estimation initiale de l'ellipse externe 118
5.4.5 Optimisation de l'image du centre 122
5.4.6 Identification 126
5.5 Résultats expérimentaux 129
5.5.1 Résultats sur données de synthèse 129
5.5.1.1 Étude de validité de l'approximation des lignes de champ . . . 129
5.5.1.2 Résultats de l'algorithme de détection 130
5.5.2 Résultats sur données réelles 131
5.6 Conclusion 133
5.6.1 Calcul de la circonférence de l'ellipse dans l'image de contours 135

6 Application au suivi de caméra 139
6.1 Introduction 140
6.1.1 Contexte - le projet ANR ROM 140
6.1.2 État de l'art 142
6.1.3 Notre problème spécifique 145
6.1.4 Notations 145
6.1.5 Formulation du problème 147
6.2 Suivi de caméra hors ligne basé sur les C²Tags 147
6.2.1 Suivi de caméra à partir de deux C²Tags coplanaires 149

6.2.2 Suivi de caméra à partir de $N \geq 2$ C^{2}Tags 153
6.2.2.1 Suivi à partir des vues-clés 154
6.2.2.2 Suivi à partir des vues intermédiaires par resection 155
6.3 Suivi de caméra en ligne basé sur une base de connaissances 3D 157
6.4 Ajustement de faisceaux 158
6.4.1 Formulation générale du problème 158
6.4.2 Paramétrage 3D euclidien minimal 159
6.4.2.1 Paramétrage d'un C^{2}Tag 159
6.4.2.2 Paramétrage de deux C^{2}Tags coplanaires 161
6.4.3 Paramétrage 3D projectif minimal 162
6.5 Résultats expérimentaux 163
6.5.1 Résultats sur données de synthèse 163
6.5.2 Résultats sur données réelles 164
6.6 Conclusion 168

7 Conclusion **171**

Table des figures

1.1 Effets spéciaux 22

3.1 Exemples de marqueurs plans. 36
3.2 Restriction de la projection centrale à un plan de la scène 37
3.3 Apport d'un suivi de marqueurs cycliques appariés 43
3.4 Erreur 3D fournie par la méthode de factorisation projective des images de paires de points cycliques 69
3.5 Résultats sur données réelles du calcul de la structure et du mouvement reposant sur la factorisation de points cycliques 69
3.6 Résultats sur données réelles du calcul de la structure et du mouvement reposant sur la factorisation de points cycliques 70
3.7 Résultats sur données réelles du calcul de la structure et du mouvement reposant sur la factorisation de points naturels et de des points cycliques 71

4.1 Résultats sur des données réelles de l'utilisation des images des paires de points cycliques pour la calcul de la structure et du mouvement 77
4.2 Les droites, rétroprojections des images des points cycliques, sont tangentes à la conique absolue 82
4.3 Preuve de la proposition 29 83
4.4 Interprétation géométrique des équations pour l'autocalibrage intégrant les points cycliques 85
4.5 Erreur 3D fournie par la méthode proposée utilisant les contraintes liées aux images des points cycliques pour la factorisation projective et pour l'autocalibrage . 91
4.6 Reconstructions denses basées sur des séquences d'images calibrées via la paradigme de reconstruction proposé 93
4.7 Reconstruction dense basée sur une séquence d'images calibrées via une méthode de reconstruction incrémentale initialisée avec la reconstruction euclidienne fournie par la méthode proposée 94

5.1 Chaîne de traitement de l'ARToolkit 101

5.2 RUNE-Tag . . . 102
5.3 Codes-barres . . . 103
5.4 Un cercle et son centre encode naturellement les points cycliques . . . 104
5.5 Image de vote fournie par la motif $\mathcal{M}$ . . . 105
5.6 Motif $\mathcal{M}$ . . . 107
5.7 Champ du gradient du motif $\mathcal{I}(\mathcal{M})$. . . . 109
5.8 C^2Tag . . . 110
5.9 Vote le long des lignes de champ . . . 112
5.10 Diagramme de flux du système de détection des C^2Tags . . . 113
5.11 Pyramide d'images . . . 114
5.12 Procédure de vote pour l'ellipse interne . . . 115
5.13 Segmentation de contour convexe . . . 116
5.14 Loi de conservation . . . 118
5.15 Exemples du protocole de vote sur images réelles . . . 119
5.16 Un résultat de la procédure de vote . . . 120
5.17 Expansion d'ellipse . . . 121
5.18 Optimisation de l'image du centre . . . 125
5.19 Sélection des coupes . . . 126
5.20 Fonctions d'identifications . . . 128
5.21 Exemples d'identifications . . . 128
5.22 Étude de validité de l'approximation des lignes de champ . . . 129
5.23 Exemples d'images de synthèse pour l'évaluation de l'algorithme de détection . . 131
5.24 Résultats fournis par l'algorithme de détection appliqué à des images de synthèse 132
5.25 Calcul du périmètre d'une ellipse . . . 135
5.26 Quelques résultats de l'algorithme de détection sur images réelles (1) . . . 136
5.27 Quelques résultats de l'algorithme de détection sur images réelles (2) . . . 137

6.1 Diagramme de flux de la méthode de suivi de caméra proposée . . . 142
6.2 Paradigme du suivi de caméra à partir d'un dispositif de deux C^2Tags coplanaires 150
6.3 La relation pôle-polaire . . . 159
6.4 Un cercle 3D dans l'espace projectif dual . . . 159
6.5 Scène composée de deux C^2Tags coplanaires . . . 162
6.6 Erreur 3D issue de l'estimation linéaire de la pose de la caméra à partir de l'image de deux C^2Tags . . . 164
6.7 Résultats de simulation pour l'ajustement de faisceaux d'un dispositif de deux C^2Tags coplanaires . . . 164
6.8 Résultats de simulation pour l'ajustement de faisceaux d'un dispositif de deux C^2Tags coplanaires avec une initialisation peu précise . . . 165
6.9 Résultats sur une séquence vidéo réelle du suivi de caméra à partir de deux C^2Tags coplanaires . . . 166
6.10 Sélection de vues-clés pour un suivi de caméra à partir de deux C^2Tags coplanaires 167

6.11 Photographies utilisées comme données d'entrée de l'algorithme de reconstruction 3D des C^2Tags pour la constitution de la base de connaissances 3D 168
6.12 Résultats à partir d'images réelles d'un suivi de caméra reposant sur la factorisation intégrant des points cycliques 169

Liste des tableaux

3.1 Algorithme de mise à l'échelle des images des points cycliques 57
3.2 Calcul des images des points cycliques manquantes 58
3.3 Algorithme de mise à l'échelle des images d'une paire de points cycliques en présence de points naturels. 64
3.4 Algorithme du calcul de l'image d'une paire de points cycliques manquante en présence de points naturels. 66

4.1 Algorithme d'autocalibrage unifié basé sur les images des points cycliques 88
4.2 Médianes des erreurs sur la distance focale en faisant varier le nombre de vues . . 90
4.3 Informations relatives aux séquences vidéo réelles 92

5.1 Algorithme de vote 116
5.2 Algorithme d'expansion d'ellipse 121
5.3 Dimensions moyennes des demi-axes des ellipses externes des images synthétisées des marqueurs 131
5.4 Analyse du taux de faux positifs de l'algorithme de détection 132
5.5 *Least Median of Squares* 134

Chapitre 1

Introduction

La vision par ordinateur (ou vision par calculateur) est une discipline relativement récente qui trouve ses racines dans la photogrammétrie, dont un des objectifs était au début des années soixante « d'obtenir des mesures précises et fiables, à partir de photographies » [Slama 1980]. Il est communément admis de dire que la vision par ordinateur est la discipline qui étudie les bases théoriques et algorithmiques grâce auxquelles de l'information quantitative ou qualitative sur une scène peut être inférée à partir de photographies prises selon des points de vue différents. Au croisement de l'informatique et des mathématiques appliquées, l'ensemble des tâches qui en découlent est disparate, mais on peut le scinder au moins en deux catégories. On peut distinguer les problèmes géométriques des problèmes photométriques, ces derniers étant des problèmes énergétiques où l'énergie mise en jeu est la lumière. Ces tâches opèrent sur les trois entités dont la mise en présence permet de prendre une photographie, à savoir : la scène, les sources lumineuses et l'appareil photographique, appelé de façon générique « caméra », lui-même composé d'un objectif et d'un boîtier muni d'un récepteur photosensible. Ces problèmes sont soit à inconnues continues soit à inconnues discrètes et, généralement, concernent l'estimation de paramètres continus liés à un modèle.

Le problème traité dans cette thèse est celui connu sous le nom du « calcul de la structure et du mouvement » (*Structure-and-Motion*) qui consiste à « expliquer » des « trajectoires » de primitives appariées dans un ensemble de photographies par un certain mouvement de l'appareil (dont sa trajectoire) et des caractéristiques géométriques de la scène photographiée relativement à une certaine représentation tridimensionnelle de l'espace projectif associée à la scène.

Contexte et enjeux. Dans le cadre de cette thèse, les enjeux liés à la résolution du problème sont à considérer dans le cadre d'une application de suivi de caméra (les termes anglophones consacrés sont *camera tracking* ou *matchmoving*) dont l'objectif est de calculer le mouvement de l'appareil dans un certain référenciel de la scène à partir d'une séquence vidéo. Le calcul de la structure de la scène est dans ce cas secondaire, et peut se limiter au strict minimum permettant un repérage tridimensionnel. Cette application est en général utilisée dans les phases de post-production audiovisuelle pour la création d'effets spéciaux, dont le but est d'incruster de façon réaliste les images

d'objets 3D virtuels dans une photographie existante en respectant les paramètres géométriques et photométriques de l'appareil (cf. figure 1.1). On parle de *réalité augmentée* lorsque cette opération doit être effectuée en temps réel. Bien que les ordinateurs permettent aujourd'hui de produire des images dont la qualité et le réalisme rivalisent avec celles que produisent nos appareils photos et nos caméras, combiner les deux de manière cohérente et imperceptible est loin d'être simple et constitue une des tâches de la vision par ordinateur. Ce réalisme ne peut être obtenue que si le mouvement de l'appareil est précisément calculé afin de garantir que l'objet incrusté soit correctement positionné dans les images réelles. Les autres tâches de la vision par ordinateur pour ce problème sont l'augmentation du degré d'automatisme et du degré de flexibilité lié à la diversité des scènes filmées et/ou des mouvements de la caméra.

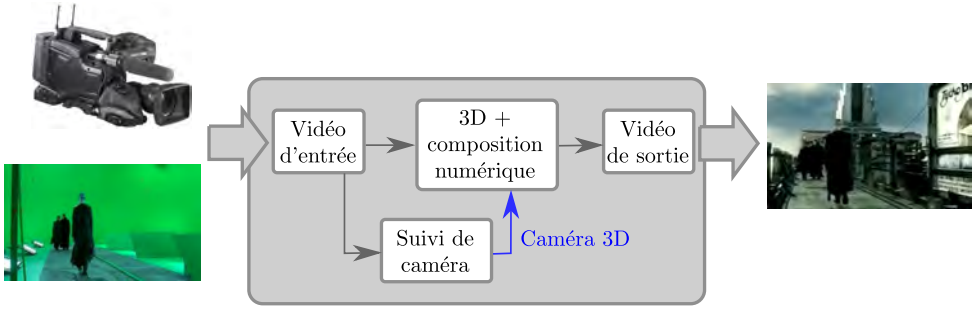


FIGURE 1.1: Exemple d'utilisation du suivi de caméra dans la chaîne de traitement pour la réalisation d'effets spéciaux.

Calcul de la structure et du mouvement. Nous supposons disposer d'une séquence de photographies, aussi appelées vues, produites par un appareil non calibré d'un point de vue géométrique, c.-à-d. dont les paramètres internes sont inconnus. Ce calibrage pourrait être obtenu à partir d'un traitement hors ligne nécessitant l'utilisation d'un équipement spécial, précédant le lancement de l'application proprement dite. Cependant, dans notre contexte, nous n'envisageons pas d'étape préalable et nous souhaitons proposer une solution automatique ne nécessitant pas d'avoir accès à la caméra. Les données d'entrée pour le calcul de la structure et du mouvement sont des correspondances de *primitives image*, c.-à-d. les images d'une même primitive 3D. Le paradigme général reposant sur l'analyse d'une collection d'images non calibrées se compose des trois étapes principales suivantes :

- extraction et mise en correspondance des primitives image ;
- calcul d'une reconstruction projective de la structure et du mouvement ;
- rectification de la reconstruction projective obtenue en une reconstruction euclidienne.

Bien qu'un très grand nombre de travaux aient été proposés sur le sujet, il n'en reste pas moins que ces méthodes présentent des limitations inhérentes au problème, telles que :

i. le possible manque de texture de l'environnement ou la présence de motifs répétitifs qui rendent très difficile les tâches d'extraction et de mise en correspondance des primitives images ;

ii. les singularités de la géométrie de la scène et/ou du mouvement de la caméra pour le calcul projectif de la structure et du mouvement et pour sa mise à jour euclidienne.

Dans ce manuscrit, nous tentons de répondre au mieux à ces limitations à travers la conception d'un *système de marqueurs*, dont le principe consiste à disposer dans la scène des objets artificiels de modèle 3D connus, ceci dans l'optique de proposer une solution de suivi de caméra présentant le meilleur compromis entre flexibilité et performances. Ainsi, lors de la conception de ce système, les questions qui ce sont naturellement posées sont les suivantes :

- De quelles caractéristiques géométriques doit disposer le marqueur pour rendre son image facilement détectable et identifiable afin de répondre au mieux aux problèmes (i) ?
- Quelle information géométrique de référence pour le calcul de la structure et du mouvement doit fournir l'image d'un marqueur afin de répondre au mieux aux problèmes (ii) ?

Lignes directrices. Nous souhaitons résoudre le problème du calcul de la structure et du mouvement en utilisant simultanément toutes les correspondances dans toutes les vues, c.-à-d. toutes les contraintes garantissant la rigidité de la scène. Ceci est exactement ce qui fait l'objet des méthodes de reconstruction par factorisation. Dans cette thèse, nous nous sommes ainsi intéressés aux méthodes de factorisation en proposant des extensions de certains paradigmes « classiques » du calcul de la structure et du mouvement à partir de données d'entrée intégrant points d'intérêt réels et *points d'intérêt virtuels*, c.-à-d. les images de points situés sur une conique virtuelle. Cette approche est motivée par l'utilisation de ces extensions aux *points cycliques*, points virtuels dont les images sont en général utilisées par les méthodes de « calibrage plan », c.-à-d. les méthodes de calibrage interne de la caméra à partir d'homographies induites par des primitives planes. En effet, les propriétés géométriques de ces primitives images présentent un grand intérêt pour le calcul de la structure et du mouvement reposant sur une factorisation projective à travers la notion de *marqueur cyclique*. Ce type de marqueur à la propriété d'être invariant par les déplacements parallèles à son plan de support. Ainsi, l'intérêt majeur d'une telle primitive est une mise en correspondance très flexible qui va nous permettre de pallier à un des problèmes critiques des méthodes de calcul de la structure et du mouvement par factorisation, à savoir le problème des données manquantes, c.-à-d. lorsque les primitives ne sont pas visibles dans toutes les vues.

Un des sous-problèmes critiques abordé dans le calcul de la structure et du mouvement est celui de l'autocalibrage de la caméra, problème consistant à transformer un calcul projectif de la structure et du mouvement en un calcul euclidien. Nous expliquons comment, dans le cas des points cycliques, utiliser l'information euclidienne fournie par les images de ces points d'intérêt virtuels, très « riches » en information euclidienne, dans l'algorithme d'autocalibrage opérant dans l'espace projectif dual et fondé sur des équations linéaires.

Dans notre cas, ces points d'intérêt virtuels sont en relation directe avec la notion de C^2Tags qui sont des marqueurs plans formés d'un ensemble de couronnes circulaires concentriques. Nous montrons en quoi ce type de marqueur comporte des propriétés intrinsèques « idéales » pour sa détection, son suivi et le calcul des images des points cycliques de son plan de support.

Organisation du manuscrit. Dans cette introduction, nous avons donné les idées générales sur le sujet étudié dans ce manuscrit.

Dans le chapitre 3, nous proposons des extensions du calcul de la structure et du mouvement

reposant sur un schéma de factorisation à partir de données d'entrée intégrant des points d'intérêt virtuels. Nous motivons notre approche via l'utilisation des images des points cycliques et fournissons l'intégralité du paradigme du calcul de la structure et du mouvement reposant sur de telles primitives image.

Dans le chapitre 4, nous traitons du problème de l'autocalibrage en proposant, d'une part, de nouvelles contraintes 3D linéaires reposant sur les images des points cycliques pour le problème de l'autocalibrage et, d'autre part, un algorithme d'autocalibrage unifié opérant dans l'espace 3D projectif dual.

Dans le chapitre 5, nous présentons le système de marqueurs C^2Tags ainsi que ses algorithmes de traitement, à savoir les algorithmes de détection et d'identification de leurs images.

Enfin, dans le chapitre 6, l'ensemble de ces contributions est mis au profit d'une méthode de suivi automatique de caméra.

Chapitre 2

Notations et rappels géométriques

Dans ce chapitre, nous présentons les notations employées dans cette thèse, à savoir les représentations des différentes entités géométriques mises en jeu ainsi que la plupart des opérateurs utilisés. Quelques rappels de géométrie projective sont également fournis.

2.1 Question de notations

Dans cette thèse, nous essaierons de distinguer le plus souvent possible les entités géométriques de leur représentation algébrique. Pour éviter une trop grande lourdeur d'écriture, nous nous autoriserons parfois à faire de telles confusions, si elles n'introduisent aucune ambiguïté dans le texte. Par exemple, dans le chapitre 3, aucune différence n'est faite entre un point et son vecteur de coordonnées ni entre une quadrique/conique et sa matrice de coefficients.

- Un vecteur est représenté par un caractère gras, par exemple $\mathbf{v}$, et une matrice par un caractère sans empattement (de typologie *sans serif*), par exemple M. La composante j du vecteur $\mathbf{v}$ sera notée v_j et l'élément (j,k) de la matrice M sera noté M_{jk}.
- Le vecteur nul de taille s est désigné par $\mathbf{0}_s$ et la matrice nulle de dimension $r \times c$ —aussi notée (r,c)— est désignée par $\mathsf{0}_{r\times c}$ ou $\mathsf{0}_s$ dans le cas d'une matrice carrée d'ordre s. La matrice identité d'ordre s est représentée par I_s.
- Les angles sont représentés par des caractères grecs minuscules, en général ϕ, θ, ψ, etc.
- Le caractère i est réservé à l'entité complexe telle que $i^2 = -1$.
- L'opérateur diag est utilisé pour désigner les matrices diagonales tel que $\mathrm{diag}(\mathbf{v})$ représente la matrice diagonale dont les éléments diagonaux sont les composantes du vecteur $\mathbf{v}$.
- Le signe $\wedge$ représente l'opérateur du produit vectoriel et $[\mathbf{e}]_\wedge$ la matrice anti-symétrique associée au vecteur $\mathbf{e} = (e_1 \;\; e_2 \;\; e_3)^\top$ tel que $\mathbf{e} \wedge \mathbf{x} = [\mathbf{e}]_\wedge \mathbf{x}$. La matrice $[\mathbf{e}]_\wedge$ s'écrit

$$[\mathbf{e}]_\wedge = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{2.1}$$

- La valeur $\det(\mathsf{A})$ représente le déterminant de la matrice A.
- L'opérateur $^\top$ désigne l'opérateur de transposition (matrice ou vecteur) et $^{-\top}$ l'opérateur de transposition de l'inverse d'une matrice (régulière).
- La notation $\mathbf{v}_{/p}$ représente le vecteur $\mathbf{v}$ privé de sa p-ème composante.
- L'opérateur $\otimes$ représente le produit de Kronecker.
- L'opération de *vectorisation* par colonne d'une matrice est notée vec. Par exemple, si l'on définit la matrice

$$\mathsf{A} = \begin{bmatrix} \mathbf{c}_1 & \dots & \mathbf{c}_c \end{bmatrix}$$

  alors

$$\mathrm{vec}(\mathsf{A}) = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_c \end{bmatrix}.$$

- Afin de procéder à la vectorisation des éléments uniques d'une matrice symétrique, nous utiliserons l'opérateur de *semi-vectorisation* par colonne, noté vech. Pour une matrice symétrique

A d'ordre n, on a :

$$\underbrace{\begin{bmatrix} A_{11} & & & & \\ \vdots & A_{21} & & & \\ \vdots & \vdots & \cdots & A_{(n-1)(n-1)} & \\ A_{n1} & A_{n2} & \cdots & A_{n(n-1)} & A_{nn} \end{bmatrix}}_{\mathsf{A}} \overset{\text{vech}}{\longmapsto} \underbrace{\begin{bmatrix} A_{11} \\ \vdots \\ A_{n1} \\ A_{21} \\ \vdots \\ A_{n2} \\ \vdots \\ A_{(n-1)(n-1)} \\ A_{n(n-1)} \\ A_{nn} \end{bmatrix}}_{\text{vech}(\mathsf{A})}$$

où $\text{vech}(\mathsf{A})$ désigne le vecteur colonne de taille $n(n+1)/2$ obtenu par vectorisation de la partie triangulaire inférieure de A.

– Nous désignons par *matrice de duplication* $\mathcal{P}_n$ l'unique matrice de dimension $n^2 \times n(n+1)/2$ qui associe $\text{vech}(\mathsf{A})$ à $\text{vec}(\mathsf{A})$ telle que

$$\text{vec}(\mathsf{A}) = \mathcal{P}_n \, \text{vech}(\mathsf{A}). \tag{2.2}$$

– L'évaluation de certains algorithmes présentés dans ce manuscrit est faite via l'expression de la racine carrée de l'erreur quadratique moyenne

$$REQM = \sqrt{\frac{1}{n} \sum_j (\hat{x}_j^{(k)} - x_j)^2} \tag{2.3}$$

avec n le nombre de paramètres estimés, $\hat{x}_j^{(k)}$ le $j^{\text{ème}}$ paramètre, calculé à l'itération k lorsqu'il s'agit d'algorithmes d'optimisation itératifs, et x_j le paramètre exact associé.

2.2 Rappels de géométrie projective

Il est indéniable que c'est grâce à l'apport de la géométrie projective que la vision par ordinateur a atteint une maturité qui a permis la conception d'algorithmes fiables et performants permettant de résoudre ses principales tâches, notamment celles de la reconstruction 3D. Le but des prochaines sections n'est pas de donner un cours complet sur la géométrie projective et la vision par ordinateur mais de définir certaines bases utilisées par la suite. De nombreux ouvrages très complets existent déjà et pour disposer d'informations complémentaires, nous invitons le lecteur à se reporter aux ouvrages [Semple 1952, Hartley 2004b].

2.2.1 L'espace projectif et son dual

L'espace projectif. Soit $\mathbb{K}_{n+1}$ un espace vectoriel de dimension $n+1$. Soit la *relation d'équivalence* $\sim$ définie sur $\mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\}$ par

$$\forall \mathbf{X}, \mathbf{Y} \in \mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\} \qquad \mathbf{X} \sim \mathbf{Y} \Leftrightarrow \exists \lambda \in \mathbb{K}\backslash\{0\} \mid \mathbf{X} = \lambda \mathbf{Y}.$$

Dans ce qui suit, la relation d'équivalence $\sim$ est appelée *égalité projective* et sera généralisée aux matrices.

On définit l'application $\mathcal{P}$ qui à tout vecteur de $\mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\}$ lui associe sa classe d'équivalence modulo l'égalité projective $\sim$. L'ensemble $\mathcal{P}(\mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\})$, c'est-à-dire l'ensemble quotient de $\mathbb{K}_{n+1}\backslash\{0\}$ suivant $\sim$, est appelé *espace projectif* déduit de $\mathbb{K}_{n+1}$ et est noté $P_n(\mathbb{K})$. Par définition,

$$dim(P_n(\mathbb{K})) = dim(\mathbb{K}^{n+1}) - 1 = n.$$

Pour tout $\mathbf{X} \in \mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\}$, l'élément $\mathcal{P}(\mathbf{X})$ est appelé *point projectif* de $P_n(\mathbb{K})$ et s'identifie à la droite vectorielle de $\mathbb{K}_{n+1}\backslash\{0\}$ de vecteur directeur $\mathbf{X}$. Ainsi il est usuel de dire que $P_n(\mathbb{K})$ est l'ensemble des droites vectorielles de $\mathbb{K}_{n+1}$ auquel on aurait soustrait le vecteur $\mathbf{0}_{n+1}$. Tout sous-espace vectoriel $F \subset \mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\}$ de dimension $r+1$ engendre un *sous-espace projectif* $\mathcal{P}(F) \subset P_n(\mathbb{K})$ de dimension r.

Dual d'un espace projectif. Le dual de l'espace projectif $P_n(\mathbb{K})$ est un espace projectif de même dimension, noté $P_n^*(\mathbb{K})$. La dualité vient de la relation bijective existant entre tout espace vectoriel et son dual : à tout point projectif de ce nouvel espace est associé un hyperplan de $P_n(\mathbb{K})$, de même qu'à tout hyperplan de $P_n^*(\mathbb{K})$ est associé un point projectif de $P_n(\mathbb{K})$. Un hyperplan de $P_n(\mathbb{K})$ est donc un « point » de $P_n^*(\mathbb{K})$ et peut être ainsi représenté par un vecteur $\mathbf{h} \in \mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\}$. Il s'ensuit que les points de $P_n(\mathbb{K})$, de vecteur $\mathbf{X} \in \mathbb{K}_{n+1}\backslash\{\mathbf{0}_{n+1}\}$, appartenant à cet hyperplan vérifient :

$$\mathbf{h}^\top \mathbf{X} = 0.$$

2.2.2 Quadriques projectives

Définition 1 (Quadrique projective) *Toute forme quadratique $\mathcal{Q}$ non nulle sur $\mathbb{K}^{n+1}$ définit une quadrique projective $\tilde{\mathcal{Q}}$ de $P_n(\mathbb{K})$ qui est l'ensemble des points de $P_n(\mathbb{K})$, appelé lieu de la quadrique, dont les vecteurs $\mathbf{X} \in \mathbb{K}^{n+1}$ satisfont l'équation $\mathcal{Q}(\mathbf{X}) = 0$.*

La matrice non nulle, symétrique et d'ordre $n+1$

$$\mathsf{Q} = \begin{pmatrix} Q_{1,1} & \frac{1}{2}Q_{1,2} & \cdots & \frac{1}{2}Q_{1,n+1} \\ \frac{1}{2}Q_{1,2} & Q_{2,2} & \cdots & \frac{1}{2}Q_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}Q_{1,n+1} & \frac{1}{2}Q_{2,n+1} & \cdots & Q_{n+1,n+1} \end{pmatrix}$$

associée à la forme quadratique

$$\mathcal{Q}(\mathbf{X}) = \sum_{i=1}^{n+1} \sum_{j=i}^{n+1} Q_{i,j} x_i x_j = \mathbf{X}^\top \mathsf{Q} \mathbf{X}$$

définit une quadrique projective $\tilde{\mathcal{Q}}$ dont le lieu a pour équation

$$\mathbf{X}^\top \mathsf{Q} \mathbf{X} = 0.$$

On appelle matrice de la quadrique $\tilde{\mathcal{Q}}$ toute matrice (non nulle, symétrique et d'ordre $n+1$) proportionnelle à Q.

2.2.2.1 Conjugaison et polarité relatives à une quadrique

Deux points $\tilde{X}$ et $\tilde{Y}$, de vecteurs $\mathbf{X}$ et $\mathbf{Y}$, sont ***conjugués relativement à une quadrique*** $\tilde{\mathcal{Q}}$ de matrice Q si et seulement si

$$\mathbf{X}^\top \mathsf{Q} \mathbf{Y} = 0.$$

L'ensemble des points conjugués à $\tilde{X}$, relativement à $\tilde{\mathcal{Q}}$ est un hyperplan $\tilde{U}$ dont le vecteur $\mathbf{U}$ est donné par

$$\mathbf{U} \sim \mathsf{Q}\mathbf{X}. \tag{2.4}$$

Définition 2 *L'hyperplan $\tilde{U}$, de vecteur (2.4), formé par l'ensemble des points conjugués à un point $\tilde{X}$, relativement à $\tilde{\mathcal{Q}}$, est appelé hyperplan polaire de $\tilde{X}$, et le point $\tilde{X}$ est appelé pôle de $\tilde{U}$.*

2.2.2.2 Tangence à une quadrique

Soit $\tilde{\mathcal{Q}}$ une quadrique et soient $\tilde{X}$ et $\tilde{Y}$ deux points, non situés sur $\tilde{\mathcal{Q}}$. Soient $\tilde{A}$ et $\tilde{B}$ les deux points où la droite passant par $\tilde{X}$ et $\tilde{Y}$ intersecte $\tilde{\mathcal{Q}}$. Si $\tilde{A}$ et $\tilde{B}$ coïncident en un même point, alors la droite est dite *droite tangente à la quadrique* $\tilde{\mathcal{Q}}$ en ce point, et celui-ci est dit *point de contact*.

On notera les points suivants :

- Le point de contact de la droite tangente à la quadrique $\tilde{\mathcal{Q}}$ passant par deux points $\tilde{X}$ et $\tilde{Y}$, non situés sur $\tilde{\mathcal{Q}}$, est conjugué à $\tilde{X}$ et $\tilde{Y}$, relativement à $\tilde{\mathcal{Q}}$.
- Pour $\tilde{X}$ (ou $\tilde{Y}$) fixé, l'ensemble des points de contact, relatif aux droites passant par $\tilde{X}$ (ou $\tilde{Y}$) qui sont tangentes à la quadrique $\tilde{\mathcal{Q}}$, est inclus dans le plan polaire de $\tilde{X}$ (ou $\tilde{Y}$).

Si $\tilde{A}$ appartient au lieu de la quadrique, alors l'ensemble des droites tangentes à $\tilde{\mathcal{Q}}$ au point $\tilde{A}$ forme un hyperplan. Ainsi, l'hyperplan formé par l'ensemble des droites tangentes à $\tilde{\mathcal{Q}}$ en un point de contact est appelé *hyperplan tangent à la quadrique* $\tilde{\mathcal{Q}}$ en ce point.

2.2.3 Quadriques projectives duales

Définition 3 *Toute forme quadratique $\mathcal{Q}^*$ non nulle sur $(\mathbb{K}^{n+1})^*$définit une quadrique projective $\tilde{\mathcal{Q}}^*$ de $P_n^*(\mathbb{K})$, dite quadrique projective duale, qui est l'ensemble des hyperplans de $P_n(\mathbb{K})$, appelé enveloppe de la quadrique, dont les vecteurs $\mathbf{U}$ satisfont l'équation $\mathcal{Q}^*(\mathbf{U}) = 0$.*

La matrice non nulle, symétrique et d'ordre $n+1$

$$\mathsf{Q}^* = \begin{pmatrix} Q^*_{1,1} & \frac{1}{2}Q^*_{1,2} & \cdots & \frac{1}{2}Q^*_{1,n+1} \\ \frac{1}{2}Q^*_{1,2} & Q^*_{2,2} & \cdots & \frac{1}{2}Q^*_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}Q^*_{1,n+1} & \frac{1}{2}Q^*_{2,n+1} & \cdots & Q^*_{n+1,n+1} \end{pmatrix},$$

associée à la forme quadratique sur $(\mathbb{K}^{n+1})^*$

$$\mathcal{Q}^*(\mathbf{U}) = \sum_{i=1}^{n+1}\sum_{j=i}^{n+1} Q^*_{i,j}u_iu_j = \mathbf{U}^\top \mathsf{Q}^*\mathbf{U}.$$

définit une quadrique projective $\tilde{\mathcal{Q}}$ *dont l'enveloppe a pour équation*

$$\mathbf{U}^\top \mathsf{Q}^*\mathbf{U} = 0.$$

On appelle ***matrice de la quadrique*** *de* $\tilde{\mathcal{Q}}^*$ *toute matrice (non nulle, symétrique et d'ordre* $n+1$*) proportionnelle à* Q^*.

Si $\tilde{\mathcal{Q}}$ est une quadrique de $P_n(\mathbb{K})$, dont la matrice Q est de rang plein, alors l'ensemble des hyperplans tangents à $\tilde{\mathcal{Q}}$ définit l'enveloppe d'une quadrique projective $\tilde{\mathcal{Q}}^*$ de $P_n^*(\mathbb{K})$ dont la matrice Q^* vérifie

$$\mathsf{Q}^* \sim \mathsf{Q}^{-1}. \tag{2.5}$$

Ainsi, puisqu'on identifie les hyperplans de $P_n(\mathbb{K})$ aux points de $P_n^*(\mathbb{K})$, la quadrique $\mathcal{Q}^*$ de $P_n^*(\mathbb{K})$ identifiée à $\mathcal{Q}$ est la quadrique projective de matrice (2.5) et, pour cette raison, est parfois appelée ***quadrique duale de*** $\mathcal{Q}$.

2.2.4 Quadriques dégénérées des espaces projectifs de dimensions deux et trois

Nous nous plaçons maintenant dans le cadre des espaces projectifs de dimensions deux et trois même si les définitions 4 et 5 restent valides pour les espaces projectifs de toute dimension. Nous restreindrons le cadre de notre travail aux formes quadratiques à coefficients dans $\mathbb{K} = \mathbb{R}$ et ainsi *nous ne considérerons que les quadriques projectives de matrices réelles*. Dans le cas général, une quadrique a $\frac{n(n+3)}{2}$ degrés de liberté correspondant aux $\frac{(n+1)(n+2)}{2}$ éléments de sa matrice moins le facteur constant non nul :

- si $n = 2$, les quadriques (qui sont alors appelées ***coniques***) ont *cinq degrés de liberté* ;
- si $n = 3$, les quadriques ont *neuf degrés de liberté*.

Définition 4 *On appelle rang d'une quadrique projective le rang de sa matrice associée. Une quadrique de rang plein est dite quadrique propre.*

Définition 5 *Une quadrique projective dégénérée est une quadrique projective qui n'est pas de rang plein, c.-à-d. une quadrique projective dont la matrice a un rang strictement inférieur à* $n+1$ *dans un espace projectif de dimension* n.

2.2.4.1 Enveloppes des quadriques dégénérées lorsque $n \in \{2, 3\}$

Nous nous restreignons à l'interprétation géométrique des quadriques dégénérées de $P_3^*(\mathbb{K})$ dans la proposition 6 ; celle-ci ne se déduit pas trivialement par dualité de l'interprétation géométrique des quadriques dégénérées de $P_3(\mathbb{K})$.

Proposition 6

$n = 3$ et $\operatorname{rang} \tilde{\mathcal{Q}}^* = 3$: l'enveloppe d'une quadrique projective $\tilde{\mathcal{Q}}^*$ de $P_3^*(\mathbb{K})$ de rang 3 est formée par l'ensemble des faisceaux de plans de $P_3(\mathbb{K})$ dont les axes sont des droites contenues dans un plan p de $P_3(\mathbb{K})$ et tangentes à une quadrique non dégénérée $\tilde{\mathcal{A}}$ de $P_3(\mathbb{K})$. On dira que $\tilde{\mathcal{Q}}^*$ est le « cône dual » d'une conique projective. Si $\mathbf{p}$ est le vecteur de p et si A^* est la matrice de la quadrique duale $\tilde{\mathcal{A}}^*$ alors la matrice Q de $\tilde{\mathcal{Q}}^*$, de rang 3, admet la décomposition suivante :

$$\mathsf{Q}^* \sim (\mathbf{p}^\top \mathsf{A}^* \mathbf{p})\mathsf{A}^* - (\mathsf{A}^* \mathbf{p})(\mathsf{A}^* \mathbf{p})^\top.$$

$n \in \{2, 3\}$ et $\operatorname{rang} \tilde{\mathcal{Q}}^* = 2$: l'enveloppe d'une quadrique projective $\tilde{\mathcal{Q}}^*$ de $P_n^*(\mathbb{K})$ de rang 2 est formée par l'ensemble des droites de $P_n^*(\mathbb{K})$ passant par au moins un point d'une paire de points distincts de $P_n(\mathbb{K})$. Si $\mathbf{u}$ et $\mathbf{v}$ sont les vecteurs de ces points, alors la matrice Q^* de $\tilde{\mathcal{Q}}^*$, de rang 2, admet la décomposition suivante :

$$\mathsf{Q}^* \sim \mathbf{u}\mathbf{v}^\top + \mathbf{v}\mathbf{u}^\top.$$

$n \in \{2, 3\}$ et $\operatorname{rang} \tilde{\mathcal{Q}}^* = 1$: l'enveloppe d'une quadrique projective $\tilde{\mathcal{Q}}^*$ de $P_n^*(\mathbb{K})$ de rang 1 est formée par un point de $P_n^*(\mathbb{K})$. Si $\mathbf{u}$ est le vecteur de ce point alors la matrice Q^* de $\tilde{\mathcal{Q}}^*$, de rang 1, admet la décomposition suivante :

$$\mathsf{Q}^* \sim \mathbf{u}\mathbf{u}^\top.$$

2.2.5 Transformation d'une quadrique projective

Soit H la matrice d'une homographie quelconque H de l'espace projectif. La transformation de toute quadrique projective de matrice Q par l'homographie H s'écrit

$$\mathsf{Q}' \sim \mathsf{H}^{-T} \mathsf{Q} \mathsf{H}^{-1}.$$

Dans l'espace projectif dual, la transformation par H de toute quadrique projective duale de matrice Q^* s'écrit

$$\mathsf{Q}^{*\prime} \sim \mathsf{H} \mathsf{Q}^* \mathsf{H}^\top.$$

2.2.6 Signature d'une quadrique projective

Nous appelons ***invariant projectif*** toute quantité qui reste inchangée quelle que soit la représentation projective considérée de l'espace. L'unique invariant projectif des quadriques projectives à

coefficients dans $\mathbb{C}$ est le rang de leur matrice. Puisque nous ne considérons que les quadriques à coefficients dans $\mathbb{R}$, il est possible de définir un autre invariant projectif appelé *signature de la quadrique*.

Définition 1 *La signature d'une quadrique projective de matrice* Q *(respectivement* Q^* *sous sa forme duale) est défini par* (ξ_1, ξ_2)*, où :*

$$\xi_1 = \max(\rho, \nu) \text{ et } \xi_2 = \min(\rho, \nu),$$

avec ρ *et* ν *qui dénombrent respectivement les valeurs propres positives et les valeurs propres négatives de* Q *(respectivement* Q^**).*

Proposition 7 *La signature d'une quadrique projective à coefficients dans* $\mathbb{R}$ *est projectivement invariante.*

La démonstration de cette proposition est immédiate à établir à partir de la signature d'une forme quadratique et de la loi d'inertie de Sylvester [Golub 1996, p. 403]. On notera que

$$\xi_1 + \xi_2 = \operatorname{rank} \mathsf{Q},$$

ce qui confirme que le rang de Q^* est aussi invariant projectivement.

Définition 8 *On appelle quadrique virtuelle toute quadrique projective de signature* $(\xi_1, \xi_2) = (R, 0)$ *où* R *désigne le rang de la quadrique, c'est-à-dire dont la matrice réelle de rang* R *à* R *valeurs propres de même signe.*

Il est facile de montrer qu'une quadrique virtuelle ne contient pas de points réels (représentés par des vecteurs réels) mais uniquement des points complexes conjugués (représentés par des vecteurs complexes conjugués).

Plus généralement, la signature définie précédemment permet d'établir la classification suivante des quadriques projectives. Nous dirons que deux quadriques sont projectivement équivalentes s'il existe une homographie transformant l'une en l'autre et vice-versa.

Pour les quadriques projectives $\mathcal{Q}$ et $\mathcal{Q}^*$ de matrices réelles respectives Q et $\mathsf{Q}^* = \mathsf{Q}^{-1}$, on a :

$$(\xi_1, \xi_2) = \begin{cases} (4,0) : \mathcal{Q}, \mathcal{Q}^* \text{ projectivement équivalentes à une sphère virtuelle} \\ (3,1) : \mathcal{Q}, \mathcal{Q}^* \text{ projectivement équivalentes à une sphère réelle} \\ (2,2) : \mathcal{Q}, \mathcal{Q}^* \text{ projectivement équivalentes à un hyperboloïde à une nappe} \end{cases} \tag{2.6}$$

On peut remarquer qu'il n'existe pas d'homographie qui transforme une sphère (virtuelle ou non) en un hyperboloïde à une nappe.

Pour les signatures d'une quadrique duale Q^* de rang 3 on a :

$$(\xi_1, \xi_2) = \begin{cases} (3,0) : \mathcal{Q}, \mathcal{Q}^* \text{ projectivement équivalentes à un cercle virtuel} \\ (2,1) : \mathcal{Q}, \mathcal{Q}^* \text{ projectivement équivalentes à un cercle réel} \end{cases} \tag{2.7}$$

2.2.7 Stratifications projectives

2.2.7.1 Structure affine d'un espace projectif. Hyperplan à l'infini.

Hyperplan à l'infini. Si F désigne un hyperplan vectoriel de $\mathbb{R}^{n+1}$, alors $P_n(\mathbb{R})\backslash\mathcal{P}(F)$ possède une structure d'espace affine de dimension n ; $\mathcal{P}(F)$ est alors appelé *hyperplan à l'infini* de $P_n(\mathbb{R})$ et est noté H_∞. On dit que la donnée de l'hyperplan à l'infini munit l'espace projectif d'une structure affine car on montre que $P_n(\mathbb{R})\backslash\mathcal{P}(F)$ a *de facto* une structure affine.

Représentation affine. Une *représentation affine* de l'espace projectif correspond au choix d'un système de coordonnées homogènes dans lequel l'hyperplan à l'infini est $H_\infty = \mathcal{P}(\mathbb{K}^n \times \{0\})$ c.-à-d. dont le vecteur s'écrit

$$\mathbf{H}_\infty = [0, ..., 0, 1]^\top .$$

Une représentation affine de $P_n(\mathbb{K})$ s'obtient par le plongement de l'espace affine $\mathbb{K}^n$ associé à lui-même dans l'espace projectif par la bijection de $\mathbb{K}^n$ sur $\mathcal{P}(\mathbb{K}^n \times \{1\})$

$$(x_1, \cdots, x_n)^\top \leftrightarrow \mathcal{P}(x_1, \cdots, x_n, 1)^\top.$$

De l'égalite $\mathcal{P}(\mathbb{K}^n \times \{1\}) = P_n(\mathbb{K}) \setminus \mathcal{P}(\mathbb{K}^n \times \{0\})$, on en déduit que $\mathcal{P}(\mathbb{K}^n \times \{1\})$ correspond aux points de $P_n(\mathbb{K})$ à l'exception de ceux contenus dans l'hyperplan projectif $\mathcal{P}(\mathbb{K}^n \times \{0\})$ c.-à-d. aux droites vectorielles de $\mathbb{K}_{n+1}$ à l'exception de celles contenues dans l'hyperplan vectoriel d'équation $x_{n+1} = 0$. Le plongement étant une bijection de $\mathbb{K}^n$ sur $\mathcal{P}(\mathbb{K}^n \times \{1\})$, nous concluons que $\mathcal{P}(\mathbb{K}^n \times \{1\})$ possède une structure d'espace affine de dimension n.

De plus,

- les points projectifs de $\mathcal{P}(\mathbb{K}^n \times \{0\})$ sont dits *points à l'infini* et s'identifient aux droites vectorielles de l'espace vectoriel $\mathbb{K}^n$, c.-à-d. aux directions des droites de l'espace affine $\mathbb{K}^n$;
- les points projectifs de $\mathcal{P}(\mathbb{K}^n \times \{1\})$ sont dits, par opposition, *points finis*, et s'identifient aux points de l'espace affine $\mathbb{K}^n$.

2.2.7.2 Structure affine euclidienne de l'espace projectif tridimensionnel. Conique absolue.

Soit $P_n(\mathbb{K})$ l'espace projectif de dimension $n = 3$ associé à une représentation affine, c.-à-d. tel que l'ensemble de ses points finis soit $\mathcal{P}(\mathbb{K}^n \times \{1\})$. La partie $\mathbb{K}^n \times \{1\}$ de $\mathbb{K}^{n+1}$ est un hyperplan affine de $\mathbb{K}^{n+1}$ que l'on peut munir d'une structure euclidienne en définissant le produit scalaire via la relation de conjugaison par rapport à un certain cône Λ_∞, ayant l'origine pour sommet et circonscrit à une conique virtuelle Ω_∞ de l'hyperplan à l'infini de signature $(3, 0)$. Dans toute représentation affine de $P_n(\mathbb{K})$, la matrice du cône s'écrit

$$\Lambda_\infty = \begin{bmatrix} \Omega_\infty & \mathbf{0}_n \\ \mathbf{0}_n^\top & 0 \end{bmatrix}$$

où Ω_∞ est une matrice réelle d'ordre n et définie positive.

Le produit scalaire sur $\mathbb{K}^n$ s'écrit alors :

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\Omega_\infty} = (\bar{\mathbf{x}}^\top, 1) \mathsf{\Lambda}_\infty \begin{pmatrix} \bar{\mathbf{y}} \\ 1 \end{pmatrix} = \bar{\mathbf{x}}^\top \Omega_\infty \mathbf{y}$$

où $\bar{\mathbf{x}}$ et $\bar{\mathbf{y}}$ sont les vecteurs de coordonnées cartésiennes de deux points finis de $P_n(\mathbb{K})$.

Conique absolue. La conique Ω_∞ est appelée *conique absolue*. Comme toute conique virtuelle, elle ne contient aucun point réel, uniquement des points conjugués complexes. Le cercle absolu est l'unique conique de l'espace projectif qui est invariant par les similitudes de l'espace affine euclidien $P_n(\mathbb{K}) \backslash \mathcal{P}(H_\infty)$. On dira que la donnée de la conique absolue munit l'espace projectif affine $P_n(\mathbb{K})$ d'une *structure euclidienne*.

Représentation affine euclidienne. Une *représentation affine euclidienne* de l'espace projectif est une représentation affine dans laquelle

$$\Omega_\infty = \mathsf{I}_n$$

c.-à-d. dans laquelle la conique absolue est un cercle absolu de rayon $\sqrt{-1}$.

Quadrique absolue duale. La conique absolue peut être définie dans l'espace projectif dual comme l'enveloppe d'une quadrique duale virtuelle, dégénérée de rang 3, notée $\mathcal{Q}^*_\infty$. Cette enveloppe est le cône dual à la conique absolue et est communément désignée par le terme *quadrique absolue duale*. C'est l'unique quadrique duale qui est invariante par les similitudes de l'espace projectif, c.-à-d. par des homographies dont les matrices sont de la forme

$$\mathsf{T} = \begin{bmatrix} s\mathsf{R} & \mathbf{t} \\ \mathbf{0}_n^\top & 1 \end{bmatrix}$$

où s est un scalaire non nul, R une matrice orthogonale (représentant une rotation si $\det \mathsf{R} = 1$) et $\mathbf{t}$ un vecteur de translation. La matrice de $\mathcal{Q}^*_\infty$ a la forme canonique suivante :

$$\mathsf{Q}^*_\infty = \begin{bmatrix} \mathsf{I}_n & \mathbf{0_n} \\ \mathbf{0}_n^\top & 0 \end{bmatrix}$$

pour toute représentation affine euclidienne de l'espace projectif et son rayon est $\sqrt{-1}$.

L'invariance du cercle absolu par toute similitude de matrice T se traduit ici simplement par l'égalité :

$$\mathsf{T}\mathsf{Q}^*_\infty\mathsf{T}^\top \sim \mathsf{Q}^*_\infty.$$

Le fait que $\mathbf{H}_\infty \in \ker \mathsf{Q}^*_\infty$ indique que Q^*_∞ « encode » simultanément les données de l'hyperplan à l'infini et du cercle absolu.

Chapitre 3

Calcul de la structure et du mouvement par factorisation projective incorporant des points cycliques

Dans ce chapitre, nous proposons d'utiliser les images de points cycliques, points virtuels situés sur la conique absolue, dans le calcul de la structure et du mouvement. Nous commençons par présenter les avantages liés à l'utilisation de telles primitives pour ce problème en introduisant la notion de marqueur cyclique. Nous montrons ensuite comment intégrer les images de points complexes dans un schéma de factorisation projective. Une méthode de factorisation projective intégrant les images des points cycliques est présentée, résolvant notamment les problèmes de mise à l'échelle des entrées de la matrice des données et de prédiction des données manquantes, et ceci, avec ou sans la présence d'images de points « naturels » réels.

3.1 Introduction

Le *calcul de la structure et du mouvement* fait ici référence au problème consistant à « expliquer » un ensemble « suffisamment grand » d'images de K points dans V vues, représentées par des vecteurs $\mathbf{x}_{vk} \in \mathbb{R}^3$, par une reconstruction tridimensionnelle des V caméras, représentées par des matrices de projection $\mathsf{P}_v \in \mathbb{R}^{3\times4}$, et des K points, représentés par des vecteurs $\mathbf{X}_k \in \mathbb{R}^4$, conformément à l'équation de projection

$$\mathbf{x}_{vk} \sim \mathsf{P}_v \mathbf{X}_k, \tag{3.1}$$

où $\sim$ désigne l'égalité projective.

La spécialisation euclidienne de ce calcul consiste à obtenir une représentation euclidienne de l'espace projectif tridimensionnel dans laquelle : (*i*) les matrices de projection sont de la forme [Hartley 2004b, p. 156]

$$\mathsf{P}_v = \mathsf{K}_v \mathsf{R}_v \left[\mathsf{I} \mid -\mathbf{c}_v \right] \tag{3.2}$$

où K_v désigne la matrice de calibrage [Hartley 2004b, p. 154], R_v désigne la matrice de rotation et $\mathbf{c}_v$ le vecteur de translation qui déterminent la « pose » de la caméra ; (*ii*) les vecteurs $\mathbf{X}_k$ ont pour composantes les « coordonnées cartésiennes augmentées » des points.

Nous différencions ici les méthodes qui résolvent le problème du *calcul de la structure et du mouvement* selon qu'elles considèrent comme données d'entrée des correspondances inter-vues de points et/ou des matrices fondamentales et/ou des matrices d'homographies associées à la restriction de la projection centrale à un plan de la scène. Le paradigme que nous proposons a pour ambition d'unifier les méthodes de calcul de la structure et du mouvement à partir de données contenant

- des images de primitives (planes) associées à des *marqueurs plans* [Bergamasco 2011, Kim 2005, Fiala 2005, Poupyrev 2000], cf. figure 3.1,
- et/ou des correspondances inter-vues d'images de points « naturels », c.-à-d. les images de primitives ponctuelles présentes naturellement dans la scène.

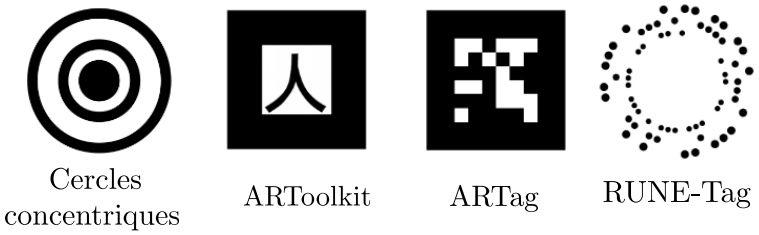


FIGURE 3.1: Exemples de marqueurs plans.

Ce paradigme repose sur un schéma de factorisation des images de points complexes.

Dans ce chapitre, nous appelons *technique de factorisation* toute technique de calcul fondée sur la réduction du rang de la matrice par blocs obtenue par concaténation des représentations matricielles de toutes les données, où la connaissance du rang théorique correspond à une contrainte de résolution, exprimant par exemple une contrainte de rigidité liée à la scène. Les techniques

de factorisation présentent un grand intérêt car elles considèrent de manière uniforme l'ensemble des données issues de toutes les vues. Le champ d'application de ces techniques est vaste et on trouvera dans [Wang 2011] un grand nombre de problèmes de vision par ordinateur résolus par factorisation. Les techniques de factorisation introduisent néanmoins des sous-problèmes délicats à résoudre : il faut « mettre à l'échelle » les données et traiter le cas (très courant en pratique) de celles manquantes, c.-à-d. lorsque les primitives ne sont pas visibles dans toutes les vues.

Calcul de la structure et du mouvement à partir de marqueurs plans. Les méthodes existantes de calcul de la structure et du mouvement à partir de marqueurs plans, comme ceux de [Fiala 2005, Poupyrev 2000], utilisent comme données principales les homographies du plan p de support d'un marqueur dans le plan pixélique d'une vue v, dont les matrices sont décomposables sous la forme

$$\mathsf{H}_{pv} = \mathsf{K}_v \mathsf{R}_v^\top \left[\mathbf{w}_p^1 \mid \mathbf{w}_p^2 \mid \mathbf{q}_p - \mathbf{c}_v \right] \tag{3.3}$$

où, relativement au repère orthonormé de la scène, le couple $(\mathsf{R}_v, \mathbf{c}_v)$ détermine la « pose » de la caméra, conformément à l'équation générale de projection (3.2), et où $\left\{ \mathbf{q}_p ; \mathbf{w}_p^1, \mathbf{w}_p^2 \right\}$ définit un repère orthonormé bidimensionnel du plan p ; $\mathbf{q}_p \in \mathbb{R}^3$ représente l'origine et $\left(\mathbf{w}_p^1, \mathbf{w}_p^2 \right) \in \mathbb{R}^3 \times \mathbb{R}^3$ est la paire de vecteurs orthonormés associée à la direction du plan. La restriction de la projection centrale au plan p est illustrée sur la figure 3.2.

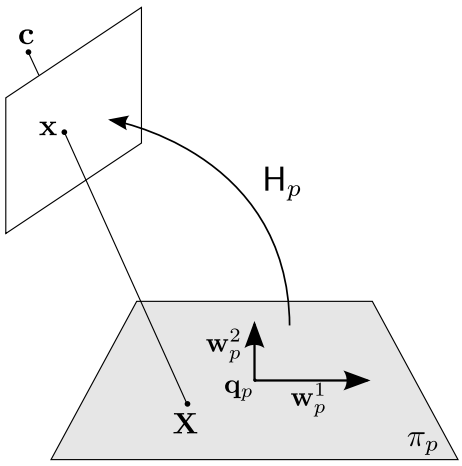


FIGURE 3.2: Restriction de la projection centrale à un plan de la scène.

Ces méthodes ont pour clef de voûte le « calibrage plan » [Hartley 2004b, p. 211], c.-à-d. le calibrage interne de la caméra à partir de ces homographies, dont les équations de base sont

$$(\mathbf{h}_{pv}^1)^\top \boldsymbol{\omega}_v \mathbf{h}_{pv}^2 = 0 \tag{3.4}$$

$$(\mathbf{h}_{pv}^1)^\top \boldsymbol{\omega}_v \mathbf{h}_{pv}^1 - (\mathbf{h}_{pv}^2)^\top \boldsymbol{\omega}_v \mathbf{h}_{pv}^2 = 0 \tag{3.5}$$

où $\mathbf{h}_{pv}^j$ représente la colonne numéro j de H_{pv}. Le prérequis d'avoir comme données d'entrée les matrices H_{pv} oblige à connaître *exactement* la géométrie du marqueur, par exemple à connaître les coordonnées cartésiennes (x, y) de tout point du marqueur dans un repère orthonormé bidimen-

sionnel de son plan de support.

Une fois les caméras calibrées, plusieurs approches ont été proposées pour le calcul de la structure et du mouvement : les premiers algorithmes ont été décrits conjointement dans [Sturm 1999] et [Zhang 1999] et des extensions de ceux-ci utilisant des schémas de factorisation ont été proposées dans [Sturm 2000, Ueshiba 2003]. La différence entre la méthode décrite dans [Sturm 2000] par rapport à celle proposée dans [Ueshiba 2003] réside dans le fait que les auteurs « calibrent » d'abord les homographies, ce qui leur permet d'extraire les matrices de rotation qu'ils factorisent afin d'obtenir les orientations des plans et des caméras. Par opposition, dans la méthode proposée par [Ueshiba 2003], les homographies sont directement factorisées. Le grand intérêt des schémas de factorisation dans ce cadre est de considérer de manière uniforme l'ensemble des matrices (des rotations ou des homographies), en les « concaténant » dans une matrice par blocs, et en imposant des contraintes de rigidité sur la structure géométrique de l'ensemble des marqueurs, via des contraintes de rang sur la matrice par blocs.

Calcul de la structure et du mouvement à partir de correspondances de points. Un des résultats majeurs de la vision par ordinateur [Hartley 2004b] est que le problème du calcul de la structure et du mouvement est bien posé pour deux vues, à condition de disposer d'un ensemble de correspondances de points entre ces deux vues qui soit suffisamment grand pour que l'on puisse calculer de façon unique la matrice fondamentale. Le point remarquable est qu'aucune connaissance *a priori* n'est nécessaire ni sur les caméras (non calibrées), ni sur la scène (si ce n'est qu'elle est rigide). Le « hic » est que la représentation associée à cette reconstruction est purement projective, dans laquelle aucun objet (comme par exemple le plan à l'infini) permettant le passage à une représentation affine n'est identifiable. Ce résultat se généralise au cas de correspondances de points entre plus de deux vues, où une reconstruction projective peut être obtenue via une technique de factorisation [Hartley 2004b, p. 444] [Wang 2011]. Le problème du passage d'une reconstruction projective à une reconstruction euclidienne est un problème à part entière, dont les tenants et aboutissants sont exposés dans [Hartley 2004b, p. 458-501]. Une spécialisation de ce problème appelée « autocalibrage » , où les seules contraintes supplémentaires dont on dispose découlent d'hypothèses sur les paramètres internes des caméras, est traitée dans le chapitre suivant.

3.2 Calcul de la structure et du mouvement à partir de marqueurs cycliques

Dans ce travail, nous utiliserons comme données principales, *non pas* les matrices des homographies d'un marqueur vers une vue (de la forme (3.3)) *mais* les sous-matrices formées par les deux premières colonnes de celles-ci, *à un facteur complexe commun près*. De façon équivalente, nous pouvons dire que nous utiliserons comme données principales les paires de vecteurs

complexes conjugués

$$\tilde{\mathbf{I}}_{vp}^{\pm} \overset{\mathbb{C}}{\sim} \mathsf{H}_{pv} \begin{bmatrix} 1 \\ \pm i \\ 0 \end{bmatrix} \triangleq \mathbf{h}_{pv}^{1} \pm i\, \mathbf{h}_{pv}^{2}, \qquad i^2 = -1, \tag{3.6}$$

où $\overset{\mathbb{C}}{\sim}$ désigne l'*égalité projective complexe*, c.-à-d. l'égalité à un facteur d'échelle complexe près, et où H_{pv} est la matrice d'ordre 3 donnée en (3.3). Les deux vecteurs $\tilde{\mathbf{I}}_{vp}^{\pm}$ représentent les images, dans la vue v, d'une paire de points complexes conjugués remarquables en géométrie projective, connue sous le nom de *paire de points cycliques du plan* p. Tout plan affine a sa propre paire de points cycliques situés sur sa droite à l'infini, qui est ainsi commune à toute la famille de plans parallèles au plan p, c.-à-d. partageant la même droite à l'infini. Ce qui est remarquable est que les vecteurs de cette paire s'écrivent $[1, \pm i, 0]^\top$ dans toute représentation euclidienne 2D du plan et que, dans l'espace 3D, le lieu géométrique de toutes les paires est la conique absolue Ω_∞. Le lecteur qui le souhaite trouvera plus loin une introduction plus approfondie à la section 3.4.2.

Pourquoi vouloir incorporer des points cycliques comme données principales dans un schéma de factorisation ? La réponse est motivée par plusieurs points :

- c'est l'information qui munit naturellement le plan projectif d'une structure euclidienne et c'est la seule qui est nécessaire pour calibrer intrinsèquement une caméra à partir des équations (3.4,3.5) ;
- c'est l'information euclidienne principale délivrée par certains marqueurs plans, par exemple comme ceux formés d'une paire de cercles concentriques ;
- c'est une information qui va nous permettre de concevoir une classe de marqueurs plans ayant la propriété d'être *invariants par les déplacements parallèles à leur plan de support* et dont les images peuvent être mises en correspondance avec une très grande flexibilité, nous permettant ainsi de pallier à un point faible des techniques de factorisation qui est celui lié aux données manquantes.

Nous reprenons maintenant ce dernier point plus en détail en introduisant la notion de marqueurs cycliques et de marqueurs cycliques appariés.

3.2.1 Marqueurs cycliques

Définition 9 (Marqueur cyclique) *Nous appelons « marqueur cyclique » toute figure plane associée à un plan de support telle que, à partir de son image dans une vue, l'information minimale que l'on puisse calculer (sans ambiguïté) est l'image de la paire de points cycliques du plan de support.*

Il est à noter que tous les marqueurs plans, tels que ceux décrits dans [Poupyrev 2000, Fiala 2005, Bergamasco 2011], à partir desquels la matrice (3.3) de l'homographie peut être calculée, sont des marqueurs cycliques. Nous savons qu'un tel marqueur contient l'information euclidienne nécessaire pour calibrer intrinsèquement une caméra. En effet, les équations de calibrage plan (3.4,3.5)

peuvent se réécrire sous la forme

$$(\tilde{\mathbf{I}}_{vp}^{\pm})^{\top}\boldsymbol{\omega}_v\tilde{\mathbf{I}}_{vp}^{\pm} = 0 \tag{3.7}$$

qui est une équation complexe homogène, où le facteur d'échelle complexe λ_{vp}, tel que $\lambda_{vp}\tilde{\mathbf{I}}_{vp}^{\pm} = \mathsf{H}_{pv}[1, \pm i, 0]^{\top}$, peut être éliminé. Qu'en est-il pour le calcul de la structure et du mouvement ? Dans les techniques utilisant des marqueurs plans, les données sont les matrices des homographies qui, une fois calibrées, peuvent être décomposées sous la forme (3.3) dont on peut extraire la structure tridimensionnelle du plan et le mouvement de la caméra, à une similitude près de l'espace [Sturm 1999, Zhang 1999, Sturm 2000, Ueshiba 2003]. Dans le cas d'un marqueur cyclique, l'intuition pourrait nous suggérer que, pour calculer puis décomposer la matrice H_{pv} de l'homographie d'un marqueur, nous pourrions utiliser, à partir de l'équation (3.6), le fait que $\mathbf{h}_{pv}^1$ et $\mathbf{h}_{pv}^2$, respectivement les parties réelle et imaginaire de $\tilde{\mathbf{I}}_{vp}^{\pm}$, coïncident avec les deux premières colonnes de H_{pv} et que donc, pour entièrement déterminer H_{pv}, il suffirait de déterminer sa troisième colonne [1]. Mais l'intuition nous induit en erreur ! En effet, comme dit plus haut, il existe un facteur d'échelle complexe non nul λ_{vp} tel que $\lambda_{vp}\tilde{\mathbf{I}}_{vp}^{\pm} = \mathsf{H}_{pv}[1, \pm i, 0]^{\top}$ et il en découle que si λ_{vp} est défini sous la forme trigonométrique

$$\lambda_{vp} \triangleq r_{vp}(\cos\theta_{vp} + i\sin\theta_{vp}), \tag{3.8}$$

où $(r_{vp}, \theta_{vp}) \in \mathbb{R}^+ \times \mathbb{R}$, alors

$$\begin{aligned}
\tilde{\mathbf{I}}_{vp} &= \frac{1}{\lambda_{vp}}\mathsf{H}_{pv}\begin{bmatrix}1\\ i\\ 0\end{bmatrix}\\
\Leftrightarrow \tilde{\mathbf{I}}_{vp} &= \left(\frac{\cos\theta_{vp} - i\sin\theta_{vp}}{r_{vp}}\right)\mathsf{H}_{pv}\begin{bmatrix}1\\ i\\ 0\end{bmatrix}\\
\Leftrightarrow \tilde{\mathbf{I}}_{vp} &= \frac{1}{r_{vp}}\mathsf{H}_{pv}\begin{bmatrix}\cos\theta_{vp} & -\sin\theta_{vp} & \cdot\\ \sin\theta_{vp} & \cos\theta_{vp} & \cdot\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\ i\\ 0\end{bmatrix},
\end{aligned} \tag{3.9}$$

où « $\cdot$ » représente n'importe quelle valeur scalaire. On voit alors qu'il existe une ambiguïté dans le plan de support correspondant à une similitude plane. Ceci illustre très clairement un résultat connu : les points cycliques sont invariants par les similitudes du plan de support [Hartley 2004b, p. 52]. La conséquence est la suivante : une fois « calibrées » , les parties réelle et imaginaire de $\tilde{\mathbf{I}}_{vp}$ correspondent aux colonnes de la matrice

$$\mathsf{K}_v^{-1}\left[\mathbf{h}_{pv}^1 \mid \mathbf{h}_{pv}^2\right] \sim \frac{1}{r_{vp}}\mathsf{R}_v\left[\mathbf{w}_p^1 \mid \mathbf{w}_p^2\right]\begin{bmatrix}\cos\theta_{vp} & -\sin\theta_{vp}\\ \sin\theta_{vp} & \cos\theta_{vp}\end{bmatrix} \tag{3.10}$$

1. Considérons le cas d'un marqueur cyclique constitué de deux cercles concentriques. Nous savons que l'image de deux cercles concentriques permet de calculer, en plus de l'image de la paire de points cycliques, l'image de leur centre (commun) « à la bonne échelle », voir [Kim 2005].

à partir desquelles *il est impossible de déterminer de façon unique* la base vectorielle $\left(\mathbf{w}_p^1, \mathbf{w}_p^2\right) \in \mathbb{R}^3 \times \mathbb{R}^3$, introduite en (3.3) et sur la figure 3.2, donnant la direction du plan p. Cette indétermination est due à une rotation autour du vecteur normal à ce plan (défini par $\mathbf{w}_p^1 \wedge \mathbf{w}_p^2$) dont l'angle θ_{vp} (inconnu) dépend de λ_{vp}. Pour calculer la structure de la scène et le mouvement des caméras, à partir des images d'une paire de points cycliques dans plusieurs vues, il est donc nécessaire de « recaler » ces vecteurs en déterminant, pour chaque vue, une matrice de similitude compatible avec (3.10), c.-à-d., de façon équivalente, en effectuant une « mise à l'échelle complexe » de $\tilde{\mathbf{I}}_{vp}$. À notre connaissance, ce *problème de « mise à l'échelle complexe » des images des points cycliques* n'a jamais été traité.

La question que l'on peut aussi se poser est la suivante : pourquoi se prive-t-on de toute l'information géométrique délivrée par les marqueurs « classiques » pour ne considérer que les points cycliques de leur plan de support ? Pour nous justifier, nous introduisons dans le paragraphe suivant la notion de « marqueurs cycliques appariés » afin d'illustrer l'intérêt d'utiliser la paire de points cycliques fournie par les marqueurs cycliques comme primitive dans un scénario de calcul de la structure et du mouvement basé sur un schéma de factorisation.

3.2.2 Marqueurs cycliques appariés

Définition 10 (Marqueurs cycliques appariés) *Deux marqueurs cycliques M_1 et M_2 sont appariés si et seulement s'ils vérifient la relation d'équivalence $\mathscr{R}$: M_1 a un plan de support parallèle à celui de M_2.*

Il est important de noter ici qu'une classe d'équivalence modulo $\mathscr{R}$ est l'ensemble de marqueurs cycliques qui produisent les images de la même paire de points cycliques et que cette paire est celle de la famille des plans parallèles au plan de support de M, où M désigne un représentant quelconque de la classe.

Définition 11 (Images appariables de marqueurs cycliques) *Les images de deux marqueurs cycliques M_1 et M_2 sont appariables si et seulement si M_1 et M_2 sont appariés.*

Dans certaines applications comme le « suivi d'une caméra », le problème consiste à calculer le mouvement de la caméra, le calcul de la structure étant secondaire et pouvant se limiter au strict minimum permettant un repérage tridimensionnel. Dans cette logique, nous pouvons tirer profit de l'invariance des points cycliques par les similitudes planes, qui donne à un marqueur cyclique la propriété d'être *invariant par les déplacements parallèles à son plan de support* ; on peut donc apparier son image avec l'image de n'importe quel marqueur cyclique appartenant à sa classe d'équivalence modulo $\mathscr{R}$, c.-à-d. dont le plan de support est parallèle à celui du premier. En effet, si l'on considère les images dans une vue v de deux marqueurs cycliques appariés, c.-à-d. situés sur des plans parallèles p et p', ces deux images sont appariables dans le sens où les vecteurs $\tilde{\mathbf{I}}_{vp}$, calculé à partir du premier marqueur, et $\tilde{\mathbf{I}}_{vp'}$ calculé à partir du second marqueur, sont égaux à un scalaire complexe près. Dans le cadre du suivi de caméra, notre idée est donc de positionner dans la scène des familles de marqueurs cycliques appariés et de mettre en correspondance les images des marqueurs cycliques associés à cette classe d'équivalence dans les différentes vues. Il

apparaît clairement que l'intérêt d'une telle approche est d'assurer qu'un même marqueur cyclique soit visible dans le plus grand nombre de vues, même si certains marqueurs cycliques de la classe d'équivalence ne sont visibles que dans très peu de vues. Nous pallions ainsi à un point faible des techniques de reconstruction par factorisation qui souffrent du problème des données manquantes, c.-à-d. lorsque certaines primitives ne sont pas visibles dans toutes les vues. Ceci est illustré sur la figure 3.3 via l'utilisation de marqueurs cycliques constitués de cercles concentriques disposés sur des plans parallèles.

Nous présentons dans ce chapitre deux cas d'usage des marqueurs cycliques pour le calcul de la structure et du mouvement. Le premier cas d'usage (illustré sur la figure 3.3) met en jeu plusieurs marqueurs cycliques appariés, en introduisant dans une scène des marqueurs cycliques sur des familles de plans parallèles. Nous supposons, dans ce premier cas, que le déplacement de la caméra et/ou la géométrie de la scène rendent impossible l'utilisation d'un schéma de factorisation « classique », c.-à-d. intégrant des points « naturels », de par la présence d'un nombre trop important de données manquantes. Il s'agit donc, dans ce premier cas, de factoriser ***seulement*** les vecteurs des images des points cycliques de plusieurs plans non parallèles dans plusieurs vues. Le deuxième cas d'usage reprend l'idée de marqueur cyclique mais se différencie du premier cas d'usage dans le sens où l'on suppose cette fois disposer d'un ensemble d'images de points naturels visibles dans toutes les vues. L'idée est alors d'incorporer les images des points cycliques dans le schéma de factorisation classique et ainsi de factoriser *simultanément* les images des points naturels et des points cycliques, c.-à-d. d'utiliser l'ensemble des contraintes disponibles simultanément. Dans ce deuxième cas, nous nous limitons au sous-cas d'*une seule classe de marqueurs appariés*, c.-à-d. d'une seule famille de plans parallèles, car c'est le seul cas qui ne se déduit pas simplement du premier cas d'usage et de la factorisation de points naturels.

La méthode proposée concernant le premier cas d'usage a été publiée dans [Calvet 2012a]. Le deuxième cas d'usage constitue, quant à lui, une partie des contributions du travail publié dans [Calvet 2013].

Organisation du chapitre. Nous présentons tout d'abord un état de l'art sur les méthodes de factorisation dédiées au calcul de la structure et du mouvement. Le problème de factorisation intégrant des points complexes est décrit dans la section 3.5. La section 3.6 est ensuite dédiée à la présention du premier cas d'usage, à savoir une factorisation de rang trois de la matrice des données composée des images de points cycliques. La section 3.7 présente le second cas d'usage qui repose sur une factorisation de rang quatre d'une matrice des données composée des images de points cycliques et des images de points naturels.

3.3 État de l'art

Parmi les méthodes de calcul de la structure et du mouvement, les méthodes basées sur un schéma de factorisation présentent un grand intérêt car celles-ci considèrent de manière uniforme l'ensemble des données issues de toutes les images [Poelman 1997, Quan 1996, Tomasi 1992, Triggs 1996]. Dans ce cadre là, l'algorithme de factorisation a tout d'abord été introduit par

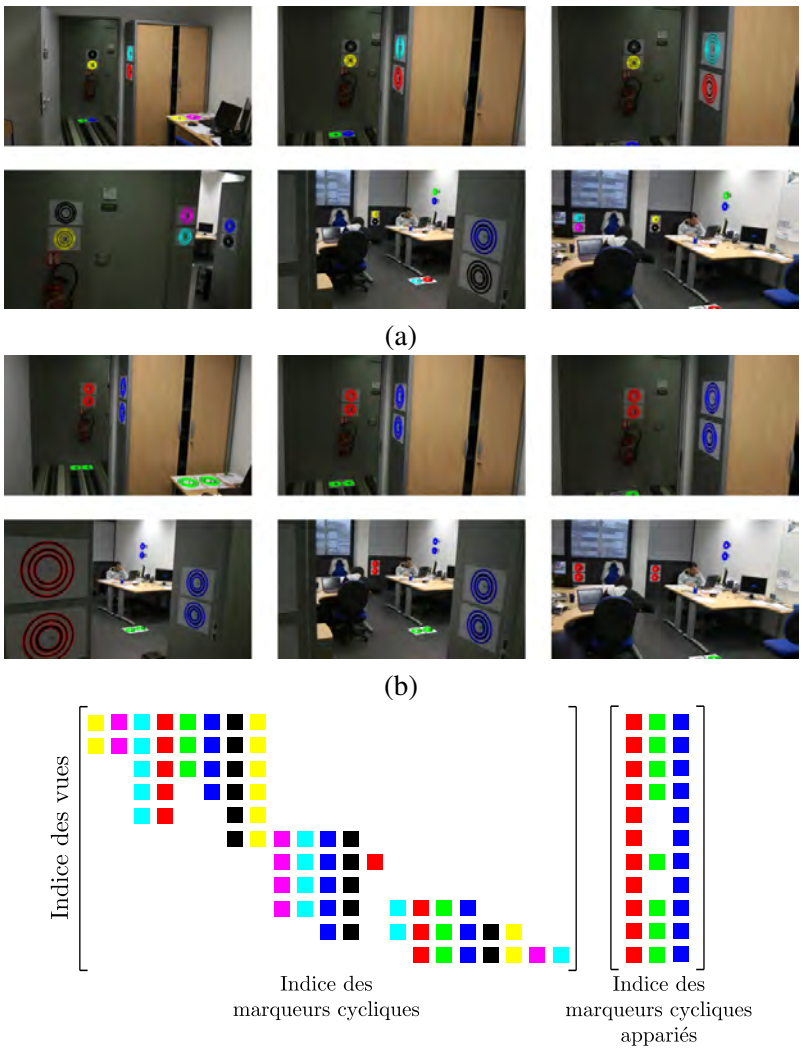


FIGURE 3.3: *Apport d'un suivi de marqueurs cycliques appariés (voir texte).* Les figures(a,b) montrent la même séquence de vues d'une scène contenant des marqueurs cycliques. Dans (a) ceux-ci sont suivis individuellement et donc visibles dans seulement quelques vues. Dans (b), en les appariant par familles de plans parallèles (1 couleur/famille) on renforce leur visibilité. Visualisation dans (c) des blocs manquants dans une matrice d'images de marqueurs « à factoriser » : à gauche, une colonne correspond à un marqueur cyclique et à droite elle correspond à une famille de marqueurs cycliques appariés.

[Tomasi 1992] au début des années 90. L'idée principale de cet algorithme est basée sur une approximation de rang réduit de la matrices des données. Celle-ci est factorisée en deux matrices, associées respectivement à la structure et au mouvement, via sa décomposition en valeurs singulières. La méthode suppose un modèle de caméra orthographique. Cet algorithme a par la suite été étendu au modèle de caméra « perspective faible » et « para-perspective » par [Poelman 1997].

[Christy 1996] ont étendu les méthodes citées ci-dessus au modèle de caméra perspective en réalisant la factorisation de manière incrémentale sous les hypothèses d'un modèle de caméra affine (comme approximation du modèle sténopé). [Sturm 1996] ont proposé une méthode de reconstruction projective via une factorisation de rang 4 de la matrice des données dont les éléments sont mis à l'échelle à partir des profondeurs projectives calculées grâce aux contraintes issues de la géométrie épipolaire. Ce type de méthode a ensuite été étudié dans [Han 2000, Heyden 1999, Mahamud 2000], où différents schémas itératifs sont proposés afin de retrouver les profondeurs projectives par minimisation des erreurs de reprojection. [Oliensis 2007] ont fourni une analyse théorique complète de la convergence de ces extensions itératives. Ils ont cependant montré qu'aucune ne présente la garantie de converger. Par ailleurs, ils proposent une extension simple, nommé CIESTA, permettant d'obtenir une initialisation fiable pour les autres algorithmes. Comparées aux approches précédentes, les méthodes décrites dans [Martinec 2002, Martinec 2005, Jia 2009] traitent du problème des données manquantes, c.-à-d. du calcul des entrées manquantes de la matrice des données liées au fait que les points à reconstruire ne sont généralement pas visibles dans toutes les vues. Dans [Martinec 2002], les profondeurs projectives sont calculées telles que décrit dans [Sturm 1996] et les données manquantes par une extension de [Jacobs 1997]. [Martinec 2005] proposent un calcul de mise à l'échelle des entrées de la matrice des données via la résolution d'un système linéaire basée sur les logarithmes des profondeurs projectives.

Les méthodes de factorisation citées ci-dessus supposent, en général, l'absence de données aberrantes. Le problème de factorisation robuste a plus récemment été étudié et des méthodes traitant de la résolution du problème en présence de données bruitées et aberrantes ont été proposées [Zaharescu 2009, Jia 2009, Anandan 2002, Gruber 2004, Tardif 2007]. Certaines de ces méthodes [Gruber 2004, Zaharescu 2009] se basent sur l'utilisation de l'algorithme *espérance-maximisation* [Dempster 1977]. [Buchanan 2005] ont proposé une comparaison détaillée d'un certain nombre de méthodes de factorisation.

Les méthodes basées sur un schéma de factorisation ont également été proposées pour la reconstruction simultanée de plans et de caméras. [Sturm 2000] a été le premier à proposer une méthode de reconstruction d'un ensemble de plans vus par plusieurs caméras via un schéma de factorisation. La méthode suppose que les caméras sont calibrées. La reconstruction est effectuée en deux étapes : (i) les orientations des plans et des caméras sont calculées par une factorisation de rang 3 de la matrice des données, dont les entrées sont les matrices de rotation 3D liées aux orientations relatives des plans par rapport aux caméras ; (ii) les positions des plans et des caméras sont ensuite calculées. [Rother 2002] ont proposé, sous l'hypothèse d'un ensemble de caméras non calibrées, une factorisation projective de rang 1 de la matrice des données dont les entrées sont les homographies liant le plan des pixels d'une caméra de référence aux plans des pixels des autres caméras. Ils proposent également une solution au problème des données manquantes,

c'est à dire lorsque certains plans ne sont pas visibles dans toutes les vues. Dans ce cas, et sous l'hypothèse qu'au moins un plan est visible dans toutes les vues, le problème est résolu en deux étapes : (i) toutes les positions des centres optiques sont calculées simultanément ; (ii) tous les plans sont ensuite reconstruits. [Ueshiba 2003] ont proposé, sous l'hypothèse d'un ensemble de caméras non calibrées, une factorisation projective de rang 4 de la matrice des données par blocs dont les entrées sont les matrices d'ordre 3 des homographies liant les plans composant la scène aux plans image des caméras. Une solution au problème de l'autocalibrage de la caméra est proposée afin d'obtenir la reconstruction euclidienne de la pose des plans et des caméras. Un état de l'art plus détaillé sur les méthodes de factorisation utilisées en vision par ordinateur est proposé dans [Wang 2011, p. 63-65].

3.4 Complexification de l'espace projectif réel

3.4.1 Pourquoi et comment complexifier un espace projectif réel ?

Si l'on part du postulat que toute matrice symétrique (non nulle) $\mathsf{M} \in \mathbb{R}^{(n+1)\times(n+1)}$ définit une quadrique de $\mathcal{P}_n(\mathbb{R})$ alors, dans le cas où sa signature est de la forme $(s, 0)$, avec $s > 0$, c.-à-d. où M est *définie (positive ou négative)*, le lieu de la quadrique est vide sur $\mathcal{P}_n(\mathbb{R})$ puisque $\mathbf{X}^\top \mathsf{M}\mathbf{X} \neq 0, \forall \mathbf{X} \in \mathbb{R}^{n+1}$. On dit qu'une telle matrice définit une *quadrique virtuelle* [Semple 1952].

Pour « donner corps » aux points d'une quadrique virtuelle, on définit la notion d'*espace projectif complexifié*, noté $\mathcal{P}_n(\mathbb{C})$, déduit de $\mathbb{C}^{n+1}$ qui, muni de la base canonique de $\mathbb{R}^{n+1}$, est un espace vectoriel de dimension $n+1$ autorisant les coordonnées homogènes à prendre des valeurs complexes. La matrice $\mathsf{M} \in \mathbb{R}^{(n+1)\times(n+1)}$ est ainsi définie positive sur $\mathbb{R}^{n+1}$ mais pas sur $\mathbb{C}^{n+1}$: elle définit alors une quadrique virtuelle de $\mathcal{P}_n(\mathbb{C})$ dont le lieu ne contient que des points appartenant à $\mathcal{P}_n(\mathbb{C}) \setminus \mathcal{P}_n(\mathbb{R})$, appelés *points virtuels* de l'espace projectif.

Les points complexes sont virtuels ; ils n'ont pas de réalité physique mais existent comme les intersections algébriques de deux coniques qui n'ont pas d'intersection géométrique réelle. Par exemple, une droite et une conique qui n'ont pas d'intersection géométrique réelle ont deux points d'intersection complexes.

Dans l'espace projectif complexifié, deux vecteurs complexes non nuls $\mathbf{X}_l$ et $\mathbf{X}_m$ de $\mathbb{C}^{n+1}$ représentent le même point projectif de $\mathcal{P}_n(\mathbb{C})$ si et seulement s'ils vérifient la relation

$$\exists \lambda \in \mathbb{C} \setminus \{0\} \mid \mathbf{X}_l = \lambda \mathbf{X}_m. \tag{3.11}$$

L'*égalité projective complexe* sera notée $\overset{\mathbb{C}}{\sim}$ de telle manière que $\mathbf{X}_l \overset{\mathbb{C}}{\sim} \mathbf{X}_m$ dans (3.11).

En fait, il existe plusieurs façons de complexifier un espace projectif réel. Nous proposons, dans ce qui suit, de représenter les points complexes de $\mathcal{P}_n(\mathbb{C})$ par des matrices réelles de dimension $(n+1, 2)$. L'intérêt de ces résultats est que l'on peut utiliser « à moindre coût » les outils de l'algèbre réelle pour traiter des vecteurs complexes.

Proposition 12 *L'ensemble des matrices de $\mathbb{R}^{(n+1)\times 2}$ a une structure d'espace vectoriel sur $\mathbb{C}$, muni de l'addition matricielle comme loi interne et de la loi externe $\circ : \mathbb{C} \times \mathbb{R}^{(n+1)\times 2} \to \mathbb{R}^{(n+1)\times 2}$*

définie par

$$\begin{aligned}(a+ib)\circ[\mathbf{X}\mid\mathbf{Y}] &= [\mathbf{X}\mid\mathbf{Y}]\,(a\mathsf{I}+b\mathsf{J})\\ &= [\mathbf{X}\mid\mathbf{Y}]\begin{bmatrix} a & b\\ -b & a\end{bmatrix}\\ &= [a\mathbf{X}-b\mathbf{Y}\mid b\mathbf{X}+a\mathbf{Y}]\end{aligned} \tag{3.12}$$

où $(a,b)\in\mathbb{R}\times\mathbb{R}$ *et*

$$\mathsf{I}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\qquad \mathsf{J}=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

(on remarquera que $\mathsf{J}^2=-\mathsf{I}$ *).*

- *On notera cet espace vectoriel* $\mathscr{C}(\mathbb{R}^{(n+1)\times 2})$.
- *On plonge* $\mathbb{R}^{n+1}$ *dans* $\mathscr{C}(\mathbb{R}^{n+1})$ *par* $\mathbf{X}\mapsto[\mathbf{X}\mid\mathbf{0}_{n+1}]$.

Proposition 13 *L'espace vectoriel* $\mathscr{C}(\mathbb{R}^{n+1})$ *est de dimension* $n+1$.

De ces propositions découle tout d'abord la réécriture de (3.11) que nous utiliserons systématiquement.

Proposition 14 *Deux vecteurs complexes non nuls* $\mathbf{X}+i\mathbf{Y}$ *et* $\mathbf{X}'+i\mathbf{Y}'$ *de* $\mathbb{C}^{n+1}$ *représentent le même point projectif de* $\mathcal{P}_n(\mathbb{C})$ *si et seulement si*

$$\exists(a,b)\in\mathbb{R}\times\mathbb{R},\ a^2+b^2\neq 0,\ \textit{tel que}\ [\mathbf{X}\mid\mathbf{Y}]=[\mathbf{X}'\mid\mathbf{Y}']\underbrace{\begin{bmatrix}a & b\\ -b & a\end{bmatrix}}_{a\mathsf{I}+b\mathsf{J}}$$

3.4.2 La paire de points cycliques d'un plan projectif

Par espace projectif, nous ferons désormais référence à l'espace projectif complexifié. Dans le plan projectif, on rappelle tout d'abord que toute droite réelle intersecte une conique (non dégénérée) en deux points.

Définition 15 (Paire de points cycliques) *Soit* P *un hyperplan réel de l'espace projectif. On appelle paire de points cycliques de* P *la paire de points conjugués complexes issue de l'intersection de la conique absolue* Ω_∞ *avec la droite à l'infini de* P.

Résultat 16 *Dans une représentation affine de l'espace projectif, toute famille de plans parallèles admet la même paire de points cycliques.*

Rappelons qu'un vecteur de $\mathbb{C}^{n+1}\setminus\{\mathbf{0}_{n+1}\}$ est dit isotrope si son carré scalaire est nul.

Résultat 17 *Soit une famille d'hyperplans parallèles dont les vecteurs, dans une représentation euclidienne de l'espace projectif, sont de la forme* $(\mathbf{n}^\top,d)^\top$*, où* $d\in\mathbb{R}$ *est une variable et* $\mathbf{n}\in\mathbb{R}^n$ *représente la normale (commune) aux plans.*

La paire de points cycliques associée à cette famille admet une paire de vecteurs isotropes de la forme :

$$\mathbf{C}_\infty^\pm = \begin{bmatrix} \bar{\mathbf{X}} \pm i\bar{\mathbf{Y}} \\ 0 \end{bmatrix} \tag{3.13}$$

où $(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \in \mathbb{R}^3 \times \mathbb{R}^3$ *est une paire de vecteurs orthonormés telle que* $\bar{\mathbf{X}} \wedge \bar{\mathbf{Y}}$ *est colinéaire à* $\mathbf{n}$.

Résultat 18 *Dans toute représentation euclidienne du plan projectif, la paire de points cycliques a pour vecteur :*

$$\mathbf{c}_\infty^\pm = \begin{bmatrix} 1 \\ \pm i \\ 0 \end{bmatrix}. \tag{3.14}$$

Résultat 19 *La paire de points cycliques est invariante par les simulitudes du plan projectif de telle sorte que, dans toute représentation euclidienne de celui-ci, on a*

$$\mathbf{c}_\infty^\pm \overset{\mathbb{C}}{\sim} \begin{bmatrix} s\mathsf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{c}_\infty^\pm, \tag{3.15}$$

où R *est une matrice réelle orthogonale d'ordre 2 et* $\mathbf{t}$ *un vecteur de dimension 2.*

Preuve. Ce résultat est un résultat très connu dont on trouvera une démonstration dans [Hartley 2004b]. Nous donnons ici une version différente. En supposant que

$$s\mathsf{R} = s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix},$$

on montre qu'il existe λ écrit sous la forme trigonométrique

$$\lambda = s(\cos\theta + i\sin\theta), \qquad (s, \theta) \in \mathbb{R}^+ \times \mathbb{R}, \tag{3.16}$$

tel que $\lambda \mathbf{c}_\infty^\pm = \begin{bmatrix} s\mathsf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{c}_\infty^\pm$. En effet :

$$\begin{aligned}
\frac{1}{\lambda} \begin{bmatrix} s\mathsf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{c}_\infty^\pm &= \frac{1}{s}(\cos\theta - i\sin\theta) \circ \begin{bmatrix} s\mathsf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathsf{R} & \frac{1}{s}\mathbf{t} \\ \mathbf{0}^\top & \frac{1}{s} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathsf{R} & \frac{1}{s}\mathbf{t} \\ \mathbf{0}^\top & \frac{1}{s} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & \cdot \\ \sin\theta & \cos\theta & \cdot \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathsf{I} & \frac{1}{s}\mathbf{t} \\ \mathbf{0}^\top & \frac{1}{s} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\
&= \mathbf{c}_\infty^\pm.
\end{aligned}$$

où « $\cdot$ » représente n'importe quelle valeur scalaire. ■

Définition 20 (Cercle) *Une conique projective est un cercle si et seulement si elle contient les points cycliques de son plan de support.*

Résultat 21 *Deux cercles du plan projectif s'intersectent en deux paires de points, l'une d'entre elles étant la paire de points cycliques du plan.*

Définition 22 (Droite isotrope) *Une droite est une droite isotrope si et seulement si elle contient un point cyclique.*

Résultat 23 *Par un point donné il passe deux droites isotropes. Dans le plan projectif, les deux droites isotropes passant par le centre d'un cercle sont tangentes au cercle.*

3.5 Le problème de la factorisation de points complexes

Nous abordons ici le problème de la factorisation projective reposant sur des images de points complexes.

Notations. Soit $\mathcal{P}_3(\mathbb{C})$ l'espace projectif associé à la scène, muni d'une certaine représentation euclidienne. Soit un ensemble de K points de $\mathcal{P}_3(\mathbb{C})$ représenté par l'ensemble d'indices

$$\mathrm{K} = \{1, \ldots, K\}, \tag{3.17}$$

et soit un ensemble de V vues, désigné par un ensemble d'indices

$$\mathrm{V} = \{1, \ldots, V\},$$

dans lesquelles les K points sont potentiellement visibles.

Nous désignons par

- $\mathbf{X}_k + i\mathbf{Y}_k$, où $[\mathbf{X}_k \mid \mathbf{Y}_k] \in \mathbb{R}^{4\times 2}$, le vecteur du point numéro $k \in \mathrm{K}$,
- $\mathbf{x}_{vk} + i\mathbf{y}_{vk}$, où $[\mathbf{x}_{vk} \mid \mathbf{y}_{vk}] \in \mathbb{R}^{3\times 2}$, le vecteur, dit *exact*, de son image dans la vue $v \in \mathrm{V}$, relativement à une représentation pixélique de celle-ci, c.-à-d. le vecteur vérifiant *exactement* l'équation de projection

$$\mathbf{x}_{vk} + i\mathbf{y}_{vk} = \mathsf{P}_v(\mathbf{X}_k + i\mathbf{Y}_k) \tag{3.18}$$

  où P_v désigne la matrice de projection définie par l'équation (3.2).

Nous représentons l'ensemble « théorique » des images des K points dans les V vues par la matrice par blocs

$$\mathcal{X} = \begin{bmatrix} [\mathbf{x}_{11} \mid \mathbf{y}_{11}] & \cdots & [\mathbf{x}_{1K} \mid \mathbf{y}_{1K}] \\ \vdots & \ddots & \vdots \\ [\mathbf{x}_{V1} \mid \mathbf{y}_{V1}] & \cdots & [\mathbf{x}_{VK} \mid \mathbf{y}_{VK}] \end{bmatrix} \in \mathbb{R}^{3V\times 2K} \tag{3.19}$$

dont le bloc (v, k) correspond au vecteur exact de l'image du point k dans la vue v.

On note $\tilde{\mathbf{x}}_{vk} + i\tilde{\mathbf{y}}_{vk}$ un vecteur du point k dans la vue v vérifiant

$$\lambda_{vk}(\tilde{\mathbf{x}}_{vk} + i\tilde{\mathbf{y}}_{vk}) = \mathbf{x}_{vk} + i\mathbf{y}_{vk},\ \lambda_{vk} \in \mathbb{C} \setminus \{\mathbf{0}\} \tag{3.20}$$

c.-à-d. un vecteur égal, à un facteur complexe (arbitraire) λ_{vk} près, au vecteur exact $\mathbf{x}_{vk} + i\mathbf{y}_{vk}$ vérifant (3.18).

3.5.1 Les données

L'ensemble des données dont on dispose est une partie de la totalité des vecteurs (3.20) (c.-à-d. représentant les images de *tous* les points dans *toutes* les vues) et est noté

$$\tilde{\mathscr{X}} \subseteq \{\tilde{\mathbf{x}}_{vk} + i\tilde{\mathbf{y}}_{vk}\}_{v\in\mathrm{V},k\in\mathrm{K}} \tag{3.21}$$

Les données de $\tilde{\mathscr{X}}$ sont concaténées dans la matrice par blocs, dite matrice des données,

$$\tilde{\mathcal{X}} = \begin{bmatrix} [\tilde{\mathbf{x}}_{11} \mid \tilde{\mathbf{y}}_{11}] & \cdots & [\tilde{\mathbf{x}}_{1K} \mid \tilde{\mathbf{y}}_{1K}] \\ \vdots & \ddots & \vdots \\ [\tilde{\mathbf{x}}_{V1} \mid \tilde{\mathbf{y}}_{V1}] & \cdots & [\tilde{\mathbf{x}}_{VK} \mid \tilde{\mathbf{y}}_{VK}] \end{bmatrix} \in \mathbb{R}^{3V\times 2K} \tag{3.22}$$

dont le bloc (v, k) est

- $[\tilde{\mathbf{x}}_{vk} \mid \tilde{\mathbf{y}}_{vk}]$ si $\tilde{\mathbf{x}}_{vk} + i\tilde{\mathbf{y}}_{vk} \in \tilde{\mathscr{X}}$ (c.-à-d. si le point k est visible dans la vue v)
- le bloc nul $\mathbf{0}_{3\times 2}$ sinon.

3.5.2 Le problème

Tout d'abord, deux faits importants sont à rappeler.

– $\mathcal{X}$ **est de rang maximum** 4**.**

$$\mathcal{X} = \underset{3V\times 4}{\begin{bmatrix} \mathsf{P}_1 \\ \vdots \\ \mathsf{P}_V \end{bmatrix}} \underbrace{\Big[[\mathbf{X}_1 \mid \mathbf{Y}_1] \quad \cdots \quad [\mathbf{X}_K \mid \mathbf{Y}_K] \Big]}_{4\times 2K}, \tag{3.23}$$

– Toute technique de factorisation de la matrice $\mathcal{X}$ permet d'obtenir une reconstruction projective de la structure et du mouvement.

Définition 24 (Donnée à l'échelle) *On appelle* vecteur à l'échelle *de l'image dans la vue v du point complexe k (ou simplement* donnée à l'échelle*), le vecteur*

$$\tilde{\mathbf{x}}'_{vk} + i\tilde{\mathbf{y}}'_{vk} = \mu_{vk}\,(\tilde{\mathbf{x}}_{vk} + i\tilde{\mathbf{y}}_{vk}) \tag{3.24}$$

où $\mu_{vk} \in \mathbb{C}$ est un scalaire non nul vérifiant la condition

$$\mu_{vk} = s_v \xi_k \lambda_{vk}, \qquad (s_v, \xi_k) \in \mathbb{R} \times \mathbb{C} \tag{3.25}$$

où $s_v \in \mathbb{R}$ et $\xi_k \in \mathbb{C}$ sont des scalaires non nuls, fixés respectivement pour toute vue v et tout point k ; λ_{vk} est le complexe non nul vérifiant (3.20).

Résultat 25 (Matrice des données mises à l'échelle) *En l'absence de bruit de mesure et dans le cas où l'inclusion au sens large de (3.21) est une égalité, la matrice par blocs des données mises à l'échelle, notée $\widetilde{\mathcal{X}'}$, dont le bloc (v, k) correspond à la matrice*

$$[\tilde{\mathbf{x}}'_{vk} \mid \tilde{\mathbf{y}}'_{vk}] \triangleq \mu_{vk} \circ [\tilde{\mathbf{x}}_{vk} \mid \tilde{\mathbf{y}}_{vk}] \tag{3.26}$$

où μ_{vk} est défini en (3.25), se factorise sous la forme

$$\widetilde{\mathcal{X}'} = \underset{3V\times 3V}{\begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_V \end{bmatrix}} \mathcal{X} \overbrace{\underbrace{\begin{bmatrix} \boldsymbol{\Xi}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Xi}_K \end{bmatrix}}_{2K\times 2K}}^{\boldsymbol{\Xi}_k = \Re(\xi_k)\mathsf{I} + \Im(\xi_k)\mathsf{J}} \tag{3.27}$$

qui est une matrice $3V \times 2K$ de rang maximum 4.

Preuve. Par (3.12), tous les blocs (v,k) de la matrice $\widetilde{\mathcal{X}'}$ se décomposent sous la forme :

$$\begin{aligned}[\tilde{\mathbf{x}}'_{vk} \mid \tilde{\mathbf{y}}'_{vk}] &\triangleq \mu_{vk} \circ [\tilde{\mathbf{x}}_{vk} \mid \tilde{\mathbf{y}}_{vk}] \\ &= s_{vk} \underbrace{\lambda_{vk} \circ [\tilde{\mathbf{x}}_{vk} \mid \tilde{\mathbf{y}}_{vk}]}_{[\mathbf{x}_{vk}|\mathbf{y}_{vk}]} \boldsymbol{\Xi}_k\end{aligned}$$

d'où découle directement (3.27). ■

Problème 26 (Les trois sous-problèmes à résoudre) *Le problème de factorisation consiste à résoudre les trois sous-problèmes suivants. :*

*SP1) « **Mise à l'échelle des blocs de la matrice des données** » au sens du résultat 25. Il s'agit ici de mettre à l'échelle chacun des blocs (v,k) de la matrice de données $\widetilde{\mathcal{X}}$ en le multipliant par un facteur non nul, noté $\mu_{vk} \in \mathbb{C}$, au sens de la définition 24, sachant que la solution pour μ_{vk} n'est pas unique mais est contrainte par (3.25).*

*SP2) « **Prédiction des blocs manquants de la matrice des données** ». Cela consiste à remplacer, dans le cas ou des données sont manquantes, les blocs nuls par des prédictions de données « à la bonne échelle ».*

*SP3) « **Réduction du rang de la matrice des données** ». Cela consiste à chercher une factorisation $\hat{\mathsf{U}}\hat{\mathsf{V}}$ approchant « au mieux » $\widetilde{\mathcal{X}'}$, où $\hat{\mathsf{U}}$ et $\hat{\mathsf{V}}$ sont des matrices réelles de dimension respective $(3V, r)$ et $(r, 2K)$, avec $r = \operatorname{rang}(\mathcal{X})$, solutions de*

$$\arg\min_{\mathsf{U},\mathsf{V}} \left\| \mathsf{UV} - \widetilde{\mathcal{X}'} \right\|_F^2 \text{ s.c. } \operatorname{rang}(\mathsf{UV}) = r \tag{3.28}$$

Dans le cas général, $r = 4$.

3.5.3 Mise à l'échelle des blocs de la matrice des données

Nous nous intéressons aux algorithmes de mise à l'échelle de la matrice des données qui résolvent le sous-problème SP1 énoncé dans le problème 26.

3.5.3.1 Les grandes lignes

Leur principe de base est le suivant. Au regard d'un traitement en séquence des vues, on opère de façon itérative en traitant, à l'étape courante, un couple de vues (v_p, v_c) : v_p est l'indice d'une vue, dite *précédente*, pour laquelle l'ensemble des facteurs d'échelle a été calculé à une étape précédente et v_c est l'indice de la vue non encore traitée, dite *courante*, pour laquelle l'ensemble des facteurs d'échelle doit être calculé.

- Initialement, la non-unicité de la solution permet de fixer à une valeur arbitraire tous les facteurs d'échelle $\{\mu_{v_0 k}\}_{k \in \mathrm{K}}$, associés à une certaine vue v_0, dite *vue-clé*.
- À la première étape, la vue précédente est $v_p = v_0$ et, à chaque étape, la vue courante v_c est choisie selon une heuristique laissée au choix du programmeur (par exemple comme proposé dans la section 3.6.1.4).

– L'algorithme s'arrête lorsque toutes les vues sont traitées.

Afin d'alléger les notations, lorsqu'il apparaît comme indice de vue, v_j est remplacé par j. Pour détailler une solution relative à la vue v_j, nous utiliserons la notation suivante pour le facteur d'échelle de l'image du point k

$$\mu_{jk} \triangleq a_{jk} + ib_{jk}, \qquad (a_{jk}, b_{jk}) \in \mathbb{R}^2. \tag{3.29}$$

3.5.3.2 Généralisation de la mise à l'échelle d'une donnée réelle à une donnée complexe

Dans [Sturm 1996], les auteurs ont décrit un algorithme de mise à l'échelle dans le cas où les données sont les vecteurs d'images de points réels. Leur algorithme traite chaque vecteur indépendamment et requiert la connaissance des matrices fondamentales entre les vues précédente v_p et courante v_c.

Relativement à l'itération associée au couple de vues (v_p, v_c), on peut généraliser la mise à l'échelle décrite dans [Sturm 1996] pour les points réels aux points complexes. En premier lieu, dans le cas de points complexes, la relation fondamentale de la vues v_p vers la vue v_c s'écrit

$$[\tilde{\mathbf{x}}_{ck} \mid \tilde{\mathbf{y}}_{ck}]^\top \widetilde{\mathsf{F}}_{pc} [\tilde{\mathbf{x}}_{pk} \mid \tilde{\mathbf{y}}_{pk}] = \mathsf{0}_{2\times 2} \Leftrightarrow \begin{cases} \tilde{\mathbf{x}}_{ck}^\top \widetilde{\mathsf{F}}_{pc} \tilde{\mathbf{x}}_{pk} - \tilde{\mathbf{y}}_{ck}^\top \widetilde{\mathsf{F}}_{pc} \tilde{\mathbf{y}}_{pk} = 0 \\ \tilde{\mathbf{x}}_{ck}^\top \widetilde{\mathsf{F}}_{pc} \tilde{\mathbf{y}}_{pk} - \tilde{\mathbf{y}}_{ck}^\top \widetilde{\mathsf{F}}_{pc} \tilde{\mathbf{x}}_{pk}^\top = 0 \end{cases} \tag{3.30}$$

où $\widetilde{\mathsf{F}}_{pc}$ désigne la matrice fondamentale entre la vue précédente v_p et la vue courante v_c. Chaque correspondance de points fournit deux équations de type (3.30). Le nombre minimum de correspondances pour estimer $\widetilde{\mathsf{F}}_{pc}$ est égal à 4. Ensuite, à partir de $\widetilde{\mathsf{F}}_{pc}$ et du vecteur $\tilde{\mathbf{e}}_{pc}$ de l'épipôle dans la vue v_c, image du centre de la caméra v_p, il est possible de calculer une solution linéaire pour la mise à l'échelle du vecteur $\tilde{\mathbf{x}}_{pk} + i\tilde{\mathbf{y}}_{pk}$ du point k dans la vue courante v_c , c.-à-d. une solution pour $\mu_{ck} = a_{ck} + ib_{ck}$, en supposant que le vecteur à l'échelle $\tilde{\mathbf{x}}'_{pk} + i\tilde{\mathbf{y}}'_{pk}$ (cf. définition 24) soit connu dans la vue précédente v_p. La solution repose sur l'équation

$$[\tilde{\mathbf{e}}_{pc}]_\wedge [\tilde{\mathbf{x}}_{ck} \mid \tilde{\mathbf{y}}_{ck}] \begin{bmatrix} a_{ck} & b_{ck} \\ -b_{ck} & a_{ck} \end{bmatrix} - \widetilde{\mathsf{F}}_{pc} [\tilde{\mathbf{x}}'_{pk} \mid \tilde{\mathbf{y}}'_{pk}] = \mathbf{0}_3 \tag{3.31}$$

Cette équation relie l'image par $\widetilde{\mathsf{F}}_{pc}$ du vecteur $\tilde{\mathbf{x}}'_{pk} + i\tilde{\mathbf{y}}'_{pk}$ d'un point de la vue v_p au vecteur $[\tilde{\mathbf{e}}_{pc}]_\wedge (\tilde{\mathbf{x}}'_{ck} + i\tilde{\mathbf{y}}'_{ck})$ de sa droite épipolaire dans la vue v_c.

Le système 3.31 comporte 2 équations linéaires indépendantes pour deux inconnues et une solution exacte existe à partir des images d'un point complexe dans deux vues.

3.5.4 Réduction du rang de la matrice des données

Le sous-problème SP3 de réduction du rang de la matrice des données *réelles* a une solution générique via la décomposition en valeurs singulières (SVD) de la matrice des données mises à l'échelle $\widetilde{\mathcal{X}'}$ (cf. [Hartley 2004b, p. 434-457]). Dans le cas de données complexes, l'utilisation de la proposition 12 nous permet de procéder de façon très similaire au cas d'une matrice des données *réelles*. Si certains points sont réels, alors les blocs de $\widetilde{\mathcal{X}'}$ correspondant à leurs images sont de

la forme $[\tilde{\mathbf{x}}'_{vk} \mid \tilde{\mathbf{y}}'_{vk}]$ avec $\mathbf{y}'_{vk} = \mathbf{0}_3$, pour tout $v \in \mathrm{V}$. Pour factoriser la matrice $\widetilde{\mathcal{X}'}$, on peut au préalable supprimer ses colonnes nulles, de la forme $\mathbf{0}_{3V}$, avant d'effectuer la décomposition SVD, sous les mêmes hypothèses que celles du sous-problème SP3.

3.6 Premier cas d'usage

Nous supposons ici disposer d'images de points « naturels » réels, mais avec un nombre trop important de données manquantes pour envisager un schéma de factorisation « classique » , et d'images de points cycliques encodées par des marqueurs cycliques appariés, c.-à-d. dont les plans de support sont distincts et non parallèles deux à deux. Il s'agit donc, dans ce premier cas, de factoriser *seulement* les vecteurs des images des points cycliques de plusieurs plans non parallèles dans plusieurs vues, en supposant un nombre raisonnable de données manquantes.

Si l'on suppose à présent le problème SP1 de « mise à l'échelle complexe » résolu, nous allons montrer que la factorisation de l'ensemble des images des points cycliques dans toutes les vues permet d'obtenir une reconstruction affine « partielle » des caméras, c.-à-d. de leurs « orientations » mais pas de leurs positions, et une reconstruction affine des points cycliques. Passer de cette reconstruction affine « partielle » à une reconstruction euclidienne « complète » ne pose aucun problème théorique. Pour obtenir une reconstruction euclidienne des orientations des caméras et des points cycliques, il suffit de déterminer l'image de la conique absolue dans une vue, soit en ajustant les images des points cycliques à l'image de la conique absolue (dont on extrait la matrice de calibrage), c.-à-d. basé sur un calibrage plan [Sturm 1999, Zhang 1999, Gurdjos 2003], soit en utilisant les homographies induites par le plan à l'infini comme décrit dans [Hartley 2004b, p. 476]. Une fois ce problème résolu, le dernier problème est celui de déterminer la position des plans et des caméras. Il est possible de retrouver conjointement les positions des centres des caméras et des marqueurs à partir des images « calibrées » d'un point caractéristique du marqueur (par exemple, l'image de son centre) et de la connaissance d'une longueur ou d'un angle, ou bien à partir d'autres points naturels mis en correspondance, ce qui permet de compléter la reconstruction.

3.6.1 Factorisation d'images de $K > 1$ points cycliques

Nous proposons de spécialiser la formulation du problème de factorisation des points complexes en faisant l'hypothèse que les indices de l'ensemble $\mathrm{K} = \{1, \ldots, K\}$ donnée en (3.17) correspondent à des *points cycliques* associés à des plans non parallèles deux à deux. Plus précisément, il s'agit de factoriser les images de $K \geq 2$ paires de points cycliques partiellement « visibles » dans $V \geq 2$ vues. Notre idée est de réaliser les mises à l'échelle *simultanées* de plusieurs images d'une même vue. À la différence du cas général, il s'agit ici d'images de paires de points complexes coplanaires, puisque les points cycliques sont tous situés sur le plan à l'infini.

Nous écrivons le vecteur du point cyclique associé à un plan k sous la forme (3.13), c.-à-d.

$$\mathbf{C}_k^\infty \triangleq \begin{bmatrix} \bar{\mathbf{X}}_k + i\bar{\mathbf{Y}}_k \\ 0 \end{bmatrix} \text{ avec } [\bar{\mathbf{X}}_k \mid \bar{\mathbf{Y}}_k] \in \mathbb{R}^{3\times 2},\ k \in \{1, \ldots, K\}$$

Nous désignons par $\mathbf{I}_{vk}$ le vecteur *exact* de son image dans la vue v et décomposons l'équation de projection d'un point cyclique sous la forme

$$\mathbf{I}_{vk} = \mathsf{P}_v \begin{bmatrix} \bar{\mathbf{X}}_k + i\bar{\mathbf{Y}}_k \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \mathbf{I}_{vk} = \mathsf{K}_v\mathsf{R}_v\mathsf{W}_k \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} \tag{3.32}$$

où la matrice orthogonale

$$\mathsf{W}_k \triangleq \left[\bar{\mathbf{X}}_k \mid \bar{\mathbf{Y}}_k \mid \bar{\mathbf{X}}_k \wedge \bar{\mathbf{Y}}_k\right]$$

représente une matrice de rotation ($\det \mathsf{W}_k = 1$) associée à l'orientation du plan k dans le repère de la caméra.

Nous reprenons les notations de la section 3.5, en particulier pour l'expression des images des points complexes et leur mise à l'échelle.

3.6.1.1 Équation de mise à l'échelle

Comme dans le cas général, l'algorithme opère de façon itérative. À chaque itération, il traite un couple de vues (v_p, v_c). Il s'agit de calculer *simultanément* des solutions pour $\{\mu_{cj}\}$ mettant à l'échelle *tous* les vecteurs $\{\tilde{\mathbf{x}}_{cj} + i\tilde{\mathbf{y}}_{cj}\}_{j\in \mathrm{J}}$ des images dans la vue courante v_c d'un certain sous-ensemble $\mathrm{J} \subseteq \mathrm{K}$ de points cycliques visibles, en supposant que les vecteurs à l'échelle $\{\tilde{\mathbf{x}}'_{pj} + i\tilde{\mathbf{y}}'_{pj}\}_{j\in \mathrm{J}}$ de leurs images dans la vue précédente v_p soient connus, c.-à-d. aient été précédemment calculés.

L'idée-clé est d'utiliser la mise en correspondance entre les images des points cycliques de la vue précédente v_p et celles de la vue courante v_c via l'homographie inter-vues induite par le plan à l'infini. On pourrait imaginer une solution qui serait d'estimer ces homographies en résolvant le problème décrit dans [Hartley 2004b, p. 91], puis transférer les vecteurs « à l'échelle » de la vue v_p vers la vue v_c. Le problème que l'on résoudrait serait alors différent du problème initial puisqu'on modifierait les vecteurs des images de la vue de destination alors que le problème initial est simplement de les mettre à l'échelle.

Pour tout point $j \in \mathrm{J}$, la solution repose sur l'équation

$$[\tilde{\mathbf{x}}_{cj} \mid \tilde{\mathbf{y}}_{cj}] \begin{bmatrix} a_{cj} & b_{cj} \\ -b_{cj} & a_{cj} \end{bmatrix} - \widetilde{\mathsf{H}}^{\infty}_{pc} \, [\tilde{\mathbf{x}}'_{pj} \mid \tilde{\mathbf{y}}'_{pj}] = \mathsf{0}_{3\times 2} \tag{3.33}$$

où la matrice $\widetilde{\mathsf{H}}^{\infty}_{pc}$ d'ordre 3 représente l'homographie inter-vues (de la vue v_p vers la vue v_c) induite par le plan à l'infini.

Montrons tout d'abord qu'il existe une solution de (3.33) pour μ_{cj} qui respecte la contrainte (3.25). Les vecteurs exacts des images d'un même point complexe dans les vues précédente et courante vérifient l'égalité

$$\mathbf{x}_{cj} + i\mathbf{y}_{cj} = \mathsf{H}^{\infty}_{pc}(\mathbf{x}_{pj} + i\mathbf{y}_{pj}) \tag{3.34}$$

où H_{pc}^{∞} désigne la matrice *exacte* de l'homographie induite par le plan à l'infini, c.-à-d. la matrice d'ordre 3 se décomposant sous la forme [Hartley 2004b, p. 339].

$$\mathsf{H}_{pc}^{\infty} = \mathsf{K}_c \mathsf{R}_c \mathsf{R}_p^{\top} \mathsf{K}_p^{-1} \tag{3.35}$$

Supposons que la solution pour une vue précédente v_p ait été correctement calculée, c.-à-d. qu'elle s'écrive sous la forme $\mu_{pj} = s_p \xi_j \lambda_{pj}$. En effectuant, dans l'équation (3.34), des substitutions découlant de (3.20,3.25) et de $\mathsf{H}_{pc}^{\infty} = z_{pc} \widetilde{\mathsf{H}}_{pc}^{\infty}$, $z_{pc} \in \mathbb{R}$, on obtient :

$$\begin{aligned}
\lambda_{ck}[\tilde{\mathbf{x}}_{cj} \mid \tilde{\mathbf{y}}_{cj}] &= \lambda_{pj}(z_{pc}\widetilde{\mathsf{H}}_{pc}^{\infty})[\tilde{\mathbf{x}}_{pj} \mid \tilde{\mathbf{y}}_{pj}] \\
\Leftrightarrow \lambda_{cj}[\tilde{\mathbf{x}}_{cj} \mid \tilde{\mathbf{y}}_{cj}] &= \frac{z_{pc}}{s_p \xi_j} \widetilde{\mathsf{H}}_{pc}^{\infty}[\tilde{\mathbf{x}}'_{pj} \mid \tilde{\mathbf{y}}'_{pj}] \\
\Leftrightarrow [\tilde{\mathbf{x}}_{cj} \mid \tilde{\mathbf{y}}_{cj}] \begin{bmatrix} a_{cj} & b_{cj} \\ -b_{cj} & a_{cj} \end{bmatrix} &= \widetilde{\mathsf{H}}_{pc}^{\infty}[\tilde{\mathbf{x}}'_{pj} \mid \tilde{\mathbf{y}}'_{pj}] \\
\text{où } \begin{cases} a_{cj} = \frac{s_p}{z_{pc}} \Re(\lambda_{cj}\xi_j) \\ b_{cj} = \frac{s_p}{z_{pc}} \Im(\lambda_{cj}\xi_j) \end{cases}
\end{aligned}$$

Il existe donc $\mu_{cj} = \frac{s_p}{z_{pc}} \lambda_{cj} \xi_j$ qui est bien de la forme (3.25) avec $s_c = s_p / z_{pc}$.

Quid de l'initialisation ? Dans la section 3.5.3.1, nous avons expliqué qu'il est possible de fixer à une valeur arbitraire tous les facteurs d'échelle $\{\mu_{0j}\}$ associés à la première vue traitée v_0, dite *vue-clé*. On peut associer un point cyclique à un plan de la famille de plans parallèles à laquelle il appartient. Considérons la vue-clé v_0 et utilisons la forme trigonométrique $\lambda_{0j} = r_{0j}(\cos\theta_{0j} + \sin\theta_{0j})$ où $(r_{0j}, \theta_{0j}) \in \mathbb{R}^+ \times \mathbb{R}$. Supposer que $\lambda_{0j} = 1$ est équivalent à attacher une nouvelle orientation au plan associé à chaque point cyclique $j \in \mathrm{J}$ en substituant $\mathbf{I}_{0j}$ par $\lambda_{0j}\tilde{\mathbf{I}}_{0j}$ dans (3.32) pour obtenir

$$\begin{aligned}
\tilde{\mathbf{I}}_{0j} &= \frac{1}{\lambda_{0j}} \mathsf{K}_0 \mathsf{R}_0 \mathsf{W}_j \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} \\
\Leftrightarrow \tilde{\mathbf{I}}_{0j} &= \left(\frac{\cos\theta_{0j} - i\sin\theta_{0j}}{r_{0j}} \right) \mathsf{K}_0 \mathsf{R}_0 \mathsf{W}_j \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} \\
\Leftrightarrow \tilde{\mathbf{I}}_{0j} &\sim \mathsf{K}_0 \mathsf{R}_0 \mathsf{W}_j \begin{bmatrix} \cos\theta_{0j} & -\sin\theta_{0j} & \cdot \\ \sin\theta_{0j} & \cos\theta_{0j} & \cdot \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix}
\end{aligned}$$

où « $\cdot$ » représente n'importe quelle valeur scalaire. À la première itération, la vue précédemment traitée est la vue-clé.

3.6.1.2 Résolution

Pour le sous-ensemble J $\subseteq$ K de points cycliques visibles dans les vues v_p et v_c, l'équation (3.33) peut se réécrire sous la forme

$$\mathsf{D}\mathbf{X} = \mathbf{0}_{6J} \tag{3.36}$$

où le vecteur

$$\mathbf{X} = \begin{bmatrix} \mathrm{vec}(\widetilde{\mathsf{H}}_{pc}^{\infty}) \\ \vdots \\ a_{cj} \\ b_{cj} \\ \vdots \end{bmatrix} \in \mathbb{R}^{9+2J} \tag{3.37}$$

a pour composantes les inconnues, à savoir les 9 éléments de $\widetilde{\mathsf{H}}_{pc}^{\infty}$ (traitée colonne par colonne) et les $2J$ facteurs d'échelle a_{cj}, b_{cj}, et où

$$\mathsf{D} = \begin{bmatrix} \vdots \\ \mathsf{B}_j \\ \vdots \end{bmatrix} \in \mathbb{R}^{6J\times(9+2J)} \tag{3.38}$$

désigne la matrice structurée par les blocs (de dimension $6 \times (9 + 2J)$) définis par

$$\mathsf{B}_j = \begin{bmatrix} \mathrm{diag}(\tilde{\mathbf{x}}'_{pj}, \tilde{\mathbf{x}}'_{pj}, \tilde{\mathbf{x}}'_{pj})^\top & \mathsf{0}_{3\times 2(j-1)} & -\tilde{\mathbf{x}}_{cj} & -\tilde{\mathbf{y}}_{cj} & \mathsf{0}_{3\times 2(J-j)} \\ \mathrm{diag}(\tilde{\mathbf{y}}'_{pj}, \tilde{\mathbf{y}}'_{pj}, \tilde{\mathbf{y}}'_{pj})^\top & \mathsf{0}_{3\times 2(j-1)} & -\tilde{\mathbf{y}}_{cj} & \tilde{\mathbf{x}}_{cj} & \mathsf{0}_{3\times 2(J-j)} \end{bmatrix}$$

où $j \in$ J.

Le système obtenu comporte $6J$ équations linéaires indépendantes pour $9 + 2J$ inconnues définies à un facteur d'échelle près. Chaque plan (via ses points cycliques) apportant un nombre d'équations égal à 6, une solution exacte (non nulle) existe si $6J = 9 + 2J - 1$, c.-à-d. si $J = 2$. En présence de bruit, une solution au sens des moindres carrés existe si l'on dispose d'un nombre minimal de $J = 2$ paires de points cycliques « visibles » dans deux vues.

L'algorithme itératif complet est donné dans la table 3.1.

3.6.1.3 Prédiction des blocs manquants de la matrice des données

Lorsque certains points cycliques ne sont pas « visibles » dans toutes les vues, certains blocs de la matrice des données sont manquants et la matrice à factoriser n'est plus entièrement définie. Heureusement, dans le cas des points cycliques, si l'on utilise la même approche que celle présentée dans le paragraphe précédent, ce problème admet une solution directe. En effet, à partir de l'homographie infinie inter-vues $\hat{\mathsf{H}}_{pc}^{\infty}$ solution de l'équation (3.33), nous pouvons transférer les images de tout point cyclique visible dans la vue v_p et manquant dans la vue v_c afin d'obtenir son image mise à l'échelle dans la vue v_c (cf. table 3.2).

PROBLÈME : Résoudre le sous-problème SP1 tel que formulé dans le probleme 26.

ALGORITHME :

1. Sélectionner une vue v_0 selon une heuristique H.
2. **Tant que** il reste des vues non traitées
   (a) Sélectionner un couple de vue (v_p, v_c) selon H.
   (b) Calculer $\mathrm{J} \leftarrow \mathrm{J}_p \cap \mathrm{J}_c$, ensemble des indices des points cycliques visibles dans les vues v_c et v_p, à condition que $|\mathrm{J}| \geq 2$.
   Contruire la matrice D définie en (3.38).
   (c) Résoudre $\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathrm{D}\mathbf{X}\|^2$ s.c. $\|\mathbf{X}\|^2 = 1$.
   (d) Déterminer $\{(\hat{a}_{cj}, \hat{b}_{cj})\}_{j \in \mathrm{J}}$ et $\hat{\mathsf{H}}_{pc}^{\infty}$ à partir de $\hat{\mathbf{X}}$.
   (e) **Pour tout** $j \in \mathrm{J}$,
   multiplier le bloc (v_c, j) de la matrice des données par $\begin{bmatrix} \hat{a}_{cj} & \hat{b}_{cj} \\ -\hat{b}_{cj} & \hat{a}_{cj} \end{bmatrix}$.
   Fin Pour
   (f) Calculer les blocs manquants (v_c, j_m) « à l'échelle » où $j_m \in \bar{\mathrm{J}}_c$ (cf. table 3.2).
Fin Tant que

TABLE 3.1: Algorithme de mise à l'échelle des blocs de la matrice des données (SP1).

3.6.1.4 Heuristique de sélection de vues

La vue-clé v_0 est sélectionnée telle que $v_0 = \arg\max_v |\mathrm{J}_v|$, c.-à-d. telle que le nombre de points cycliques « visibles » dans celle-ci soit maximal. Nous considérons ensuite deux ensembles de vues : l'ensemble des vues traitées, noté $\mathit{\Pi}$, et celui des vues non traitées, noté $\bar{\mathit{\Pi}}$. Ces deux ensembles sont initialisés à la première itération tels que $\mathit{\Pi} = \{v_0\}$ et $\bar{\mathit{\Pi}} = \{1 \cdots V\} \backslash v_0$. Ensuite, à chaque itération, un couple de vues $(v_p, v_c) \in \mathit{\Pi} \times \bar{\mathit{\Pi}}$ est sélectionné tel qu'il vérifie

$$\arg\max_{v_p, v_c} |\mathrm{J}_c \cap \mathrm{J}_p| \text{ s.c. } \bar{\mathrm{J}}_c \subset \mathrm{J}_p,$$

c.-à-d. tel qu'il maximise le nombre $|\mathrm{J}_p \cap \mathrm{J}_c|$ de points cycliques visibles à la fois dans les vues v_p et v_c et tel que tous les points cycliques manquants dans la vue v_c soient visibles dans la vue v_p.

3.6.2 Reconstruction euclidienne post-factorisation de la structure et du mouvement

La première question que nous nous posons ici est « que pouvons-nous obtenir à partir d'une factorisation intégrant plusieurs paires de points cycliques ? ». Deux faits importants peuvent être établis.

1. La matrice « théorique » $\mathcal{X}$, cf. équation (3.23), dont le bloc (v, k) correspond au vecteur exact de l'image du point k dans la vue v, est de **rang maximum** 3. En effet, les points cycliques sont des points à l'infini dont les vecteurs $\mathbf{C}_j^{\infty}$ sont de la forme (3.13) dans la

PROBLÈME : Calculer les blocs manquants (v_c, j_m) de la matrice des données « mises à l'échelle » $\widetilde{\mathcal{X}}'$ associés aux plans $j_m \in \{\mathrm{J}_p \cap \bar{\mathrm{J}}_c\}$ visibles dans la vue v_p et absents dans la vue v_c.

ALGORITHME :

1. Pour tout $j_m \in \{\mathrm{J}_p \cap \bar{\mathrm{J}}_c\}$, ensemble des points cycliques non visibles dans la vue v_c,
   (a) Calculer $[\tilde{\mathbf{x}}'_{cj_m}|\tilde{\mathbf{y}}'_{cj_m}] = \hat{\mathsf{H}}^{\infty}_{pc}[\tilde{\mathbf{x}}'_{pj_m}|\tilde{\mathbf{y}}'_{pj_m}]$ où $\hat{\mathsf{H}}^{\infty}_{pc}$ désigne la solution obtenue à l'étape (d) de l'algorithme présenté dans la table 3.1.
   (b) Assigner le bloc (v_p, j_m) mis à l'échelle de $\widetilde{\mathcal{X}}'$ à $[\tilde{\mathbf{x}}'_{cj_m}|\tilde{\mathbf{y}}'_{cj_m}]$.

TABLE 3.2: Étape optionnelle de l'algorithme présenté dans la table 3.1 : calcul des images des points cycliques manquantes.

représentation euclidienne de l'espace considérée. Il en découle de l'équation (3.32) que

$$\mathcal{X} = \underset{3V\times 3}{\begin{bmatrix} \mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ \mathsf{K}_V\mathsf{R}_V \end{bmatrix}} \underset{4\times 2K}{\begin{bmatrix} [\bar{\mathbf{X}}_1 \mid \bar{\mathbf{Y}}_1] & \cdots & [\bar{\mathbf{X}}_K \mid \bar{\mathbf{Y}}_K] \end{bmatrix}} \tag{3.39}$$

2. Toute technique de factorisation des images de points cycliques permet d'obtenir (au mieux) une **reconstruction affine 3D de la structure** (c.-à-d. des points cycliques) mais *seulement* une **reconstruction affine *partielle* du mouvement**. Nous rappelons qu'une reconstruction affine « complète » du mouvement correspond à la donnée des matrices de projection de la forme $\mathsf{P}'_v = \begin{bmatrix} \mathsf{H}^{\infty}_{v_0 v} \mid \mathbf{e}_{v_0 v} \end{bmatrix}$, où $\mathbf{e}_{v_0 v}$ est le vecteur de l'épipole correspondant à l'image du centre de la caméra v_0 dans la vue v [Hartley 2004b, p. 271]. Par « reconstruction affine partielle du mouvement » , nous exprimons le fait que H^{∞}_{pc} peut être calculée mais pas $\mathbf{e}_{pc}$. En effet, pour $v = 1$, en reprenant (3.39), on voit que

$$\begin{aligned} \mathcal{I} &= \begin{bmatrix} \mathsf{I}_3 \\ \vdots \\ \mathsf{K}_V\mathsf{R}_V\mathsf{R}_1^{\top}\mathsf{K}_1^{-1} \end{bmatrix} \begin{bmatrix} \mathsf{K}_1\mathsf{R}_1[\bar{\mathbf{X}}_1 \mid \bar{\mathbf{Y}}_1] & \cdots & \mathsf{K}_1\mathsf{R}_1[\bar{\mathbf{X}}_K \mid \bar{\mathbf{Y}}_K] \end{bmatrix} \\ &= \begin{bmatrix} \mathsf{I}_3 \mid \mathbf{0}_3 \\ \vdots \\ \mathsf{H}^{\infty}_{1V} \mid \mathbf{e}_{1V} \end{bmatrix} \begin{bmatrix} \mathsf{K}_1\mathsf{R}_1[\bar{\mathbf{X}}_1 \mid \bar{\mathbf{Y}}_1] & \cdots & \mathsf{K}_1\mathsf{R}_1[\bar{\mathbf{X}}_K \mid \bar{\mathbf{Y}}_K] \\ \mathbf{0}_2^{\top} & & \mathbf{0}_2^{\top} \end{bmatrix} \end{aligned} \tag{3.40}$$

Le problème consiste ici à calculer une reconstruction euclidienne « totale » de la structure et du mouvement. À l'issue de la résolution du problème (3.28) de réduction de rang, nous disposons des matrices $\hat{\mathsf{U}}$ et $\hat{\mathsf{V}}$ à une ambiguïté projective près représentée par la transformation T telles qu'en l'absence de bruit de mesure,

$$(\hat{\mathsf{U}}\mathsf{T}^{-1})^{\top} = [\ldots s_v(\mathsf{K}_v\mathsf{R}_v)^{\top}\ldots]^{\top} \tag{3.41}$$

$$\mathsf{T}\hat{\mathsf{V}} = [\ldots q_j[\bar{\mathbf{X}}_k \mid \bar{\mathbf{Y}}_k]\ldots]. \tag{3.42}$$

Dans notre cas, en supposant que $\mathsf{R}_1 = \mathsf{I}_3$, nous pouvons facilement montrer que $\mathsf{T} = s_1\mathsf{K}_1$.

Dans l'approche choisie, nous ne chercherons pas à déterminer les épipoles « manquants » pour compléter le mouvement, qu'il faudrait déterminer à partir des images des points naturels non utilisés dans la factorisation. En fait, à partir d'une factorisation des points cycliques, nous procédons différemment en calculant directement, c.-à-d. de façon non stratifiée, la structure et le mouvement dans une représentation euclidienne de l'espace. Les étapes de notre approche sont les suivantes :

1. Nous effectuons une reconstruction euclidienne 3D des points cycliques en calibrant la caméra numéro 1 (ou n'importe quelle vue-clé), c.-à-d. en estimant la matrice de calibrage K_1. L'estimation de cette matrice (triangulaire supérieure) se ramène au problème d'estimation linéaire de l'image de la conique absolue, dont la matrice est de la forme $\boldsymbol{\omega}_1 = \mathsf{K}_1^{-\top}\mathsf{K}_1^{-1}$ dans la vue 1. Cette estimation peut se formuler soit comme le problème d'ajustement des images des points cycliques à l'image de la conique absolue (problème dit du « calibrage plan » décrit dans [Hartley 2004b, p. 211]), soit comme le problème d'estimation de celle-ci (sous sa forme duale) à partir des homographies infinies inter-images tel que décrit dans [Hartley 2004b, p. 476] où les paramètres de calibrage sont supposés constants.
2. À partir des matrices de calibrage K_v, nous calculons les matrices de rotation à partir de la décomposition (3.40), décrivant les orientations des caméras, relativement au repère de la première caméra.
3. À partir d'un ensemble de correspondances d'images « calibrées » de points naturels et des matrices de rotation calculées dans l'étape 2, nous estimons les vecteurs de translation, décrivant les positions des caméras, comme décrit dans [Sturm 2000]. Il est à noter que des méthodes résolvant les calculs simultanés des positions des caméras et des positions des points 3D peut s'effectuer par des algorithmes tels que ceux décrits dans [Hartley 2004a, Kahl 2005, Ke 2007].

3.7 Deuxième cas d'usage

Dans le premier cas d'usage, nous avons décrit comment incorporer les images de plusieurs points cycliques dans un schéma de factorisation. Le deuxième cas d'usage se différencie du premier cas d'usage dans le sens où l'on suppose cette fois disposer d'un ensemble d'images de points naturels visibles dans toutes les vues. L'idée est ici d'utiliser comme données du problème de factorisation à la fois les images de $J \geq 1$ paires de points cycliques et de $N = K - J \geq 2$ points naturels, tous potentiellement « visibles » dans $V \geq 2$ vues. L'intérêt de cette approche est d'utiliser l'ensemble des contraintes disponibles simultanément.

Ceci nous permet d'obtenir une reconstruction projective de la structure et du mouvement. Il n'existe pas, à notre connaissance, d'approche similaire puisque dans les méthodes de calcul de la structure et du mouvement via un schéma de factorisation dont les entrées sont des homographies (telles que [Sturm 2000] ou encore [Ueshiba 2003]) il est nécessaire de disposer d'au moins deux homographies dans deux vues. D'autre part, ces solutions n'intègrent pas la possibilité d'introduire

les images de points naturels connues à un facteur d'échelle arbitraire. Le problème de « mise à l'échelle complexe » des images des points cycliques se pose d'une façon relativement différente et doit utiliser les images des points naturels pour être résolu. Notre contribution est d'expliquer comment, à partir de ce nouvel ensemble de données, effectuer la « mise à l'échelle » des images des points cycliques et comment calculer les images « non visibles » des points cycliques mises à l'échelle.

Les données du problème. L'ensemble des données est composé de deux sous-ensembles disjoints

$$\tilde{\mathscr{X}} = \tilde{\mathscr{I}} \cup \tilde{\mathscr{U}} \tag{3.43}$$

où

$$\tilde{\mathscr{I}} \subseteq \left\{ \tilde{\mathbf{I}}_{vp} \mid \lambda_{vp} \tilde{\mathbf{I}}_{vp} = \mathbf{I}_{vp}, \;\; \lambda_{vp} \in \mathbb{C} \setminus \{\mathbf{0}\} \right\}_{v \in \mathtt{V}, p=1..J} \tag{3.44}$$

est une partie (supposée suffisamment grande) de la totalité des images de la paire de points cycliques dans toutes les vues, et

$$\tilde{\mathscr{U}} = \{ \tilde{\mathbf{u}}_{vn} \;\mid\; z_{vn} \tilde{\mathbf{u}}_{vn} = \mathbf{u}_{vn}, \;\; z_{vn} \in \mathbb{R} \setminus \{0\} \}_{v \in \mathtt{V}, n=1..N} \tag{3.45}$$

est l'ensemble composé de la totalité des images de $N \geq 6$ points naturels dans toutes les vues.

- Le premier est représenté par la donnée élémentaire du vecteur $\tilde{\mathbf{I}}_{vp}$ qui correspond, en l'absence de bruit, au vecteur exact $\mathbf{I}_{vp}$, c.-à-d. vérifant (3.32), à un facteur complexe près (en général arbitraire).
- Le second est représenté par la donnée élémentaire du vecteur $\tilde{\mathbf{u}}_{vn}$ qui correspond, en l'absence de bruit, au vecteur exact $\mathbf{u}_{vn}$, c.-à-d. vérifant (3.1), à un facteur réel près (en général arbitraire).

3.7.1 Factorisation d'images de points naturels et de points cycliques

Nous nous restreignons au cas d'une seule classe de marqueurs cycliques appariés, c.-à-d. à une seule famille de plans de support parallèles avec $J = 1$, car il s'agit du seul cas qui pose problème dans le sens où il requiert, en théorie, les images d'un certain nombre de points naturels additionnels. La généralisation à $J > 1$ classes de marqueurs cycliques appariés est directe. Nous utiliserons l'indice $j \in \mathrm{K}$ pour référencer la (seule) paire de points cycliques considérée. Nous reprenons les notations utilisées dans le cadre général présenté dans la section 3.5.

L'algorithme que nous décrivons dans ce qui suit ne requiert que les vecteurs des épipôles inter-vues et non les matrices fondamentales inter-vues. L'algorithme fonctionne en deux passes. La première consiste à effectuer les mises à l'échelle *simultanées* des images de tous les points visibles dans la vue courante, et la seconde à effectuer les prédictions *simultanées* des images de tous les points non visibles dans la vue courante. D'un point de vue théorique, il aurait été possible de traiter simultanément ces deux sous-problèmes. En pratique, il s'est pourtant avéré que cette façon de procéder était la plus performante. Comme dans le cas précédent, l'algorithme opère de façon itérative. À chaque itération relative au couple de vues (v_p, v_c), il s'agit de calculer

simultanément des solutions pour $\{\mu_{ck}\}$ mettant à l'échelle *tous* les vecteurs $\{\tilde{\mathbf{x}}_{ck} + i\tilde{\mathbf{y}}_{ck}\}_{k\in \mathrm{K}}$ des images dans la vue courante v_c de l'ensemble de points K, qui contient $\{j\}$, en supposant que les vecteurs à l'échelle $\{\tilde{\mathbf{x}}'_{pk} + i\tilde{\mathbf{y}}'_{pk}\}_{k\in \mathrm{K}}$ de leurs images dans la vue précédente v_p soient connus, c.-à-d. aient été précédemment calculés.

3.7.1.1 Équations de mise à l'échelle

La solution repose sur trois équations. Pour le point cyclique $j \in \mathrm{K}$, la première équation est

$$[\tilde{\mathbf{x}}_{cj} \mid \tilde{\mathbf{y}}_{cj}] \begin{bmatrix} a_{cj} & b_{cj} \\ -b_{cj} & a_{cj} \end{bmatrix} - \widetilde{\mathsf{H}}_{pc} \, [\tilde{\mathbf{x}}'_{pj} \mid \tilde{\mathbf{y}}'_{pj}] = \mathsf{0}_{3\times 2} \tag{3.46}$$

où la matrice d'ordre 3 $\widetilde{\mathsf{H}}_{pc}$ représente l'homographie inter-vues (de la vue v_p vers la vue v_c) induite par un plan P qui contient la paire de points cycliques (ce plan appartient à une famille de plans parallèles à laquelle le plan à l'infini appartient aussi). La seconde équation exprime le fait que les épipôles sont en correspondance via l'homographie induite par n'importe quel plan, et donc en particulier par le plan P, tel que

$$\widetilde{\mathbf{e}}_{pc} - \widetilde{\mathsf{H}}_{pc}\widetilde{\mathbf{e}}_{cp} = \mathsf{0}_3 \tag{3.47}$$

Pour tout point naturel $k \in \mathrm{K} \setminus \{j\}$ la troisième équation est

$$[\tilde{\mathbf{e}}_{pc}]_\wedge \left([\tilde{\mathbf{x}}_{ck} \mid \tilde{\mathbf{y}}_{ck}] \begin{bmatrix} a_{ck} & b_{ck} \\ -b_{ck} & a_{ck} \end{bmatrix} - \widetilde{\mathsf{H}}_{pc} \, [\tilde{\mathbf{x}}'_{pk} \mid \tilde{\mathbf{y}}'_{pk}] \right) = \mathsf{0}_{3\times 2} \tag{3.48}$$

Il est facile de montrer que cette dernière équation est en fait une réécriture de la relation fondamentale (3.31) à partir de la décomposition de la matrice fondamentale [Hartley 2004b, p. 243] sous la forme $\widetilde{\mathsf{F}}_{pc} = [\tilde{\mathbf{e}}_{pc}]_\wedge \, \widetilde{\mathsf{H}}_{pc}$. Nous supposons disposer de l'ensemble $\tilde{\mathscr{U}}$ des images de points naturels ***réels***, cadre le plus courant et celui utilisé lors de nos expérimentations. L'équation (3.48), lorsqu'il s'agit des images de points naturels réels, se réécrit

$$[\tilde{\mathbf{e}}_{pc}]_\wedge \left(a_{ck}\tilde{\mathbf{x}}_{ck} - \widetilde{\mathsf{H}}_{pc} \, \tilde{\mathbf{x}}'_{pk} \right) = \mathbf{0}_3 \tag{3.49}$$

Paramétrage minimal. La première équation (3.46) représente sous forme matricielle un système de six équations linéaires par rapport aux inconnues $\{a_{cj}, b_{cj}\}$ et aux éléments de la matrice (inconnue) $\widetilde{\mathsf{H}}_{pc}$. La seconde équation (3.47) représente sous forme matricielle un système de trois équations linéaires par rapport à $\widetilde{\mathsf{H}}_{pc}$. La troisième équation (3.49) représente sous forme matricielle un système de trois équations linéaires par rapport à a_{ck} et à $\widetilde{\mathsf{H}}_{pc}$, dont deux sont indépendantes, la matrice $[\tilde{\mathbf{e}}_{pc}]_\wedge$ étant de rang 2.

Un résultat important est le suivant.

Proposition 27 *Il existe une (simple) infinité d'homographies dont les matrices vérifient les équations (3.46-3.49).*

Preuve. Soit Q un plan réel arbitraire, de vecteur $\mathbf{q}_p = (\mathbf{n}_p^\top, d_p)^\top$ relativement au repère de la caméra liée à la vue v_p, contenant la paire de points complexes j. Nous savons que la matrice de l'homographie induite par ce plan admet une décomposition [Hartley 2004b, p. 327] de la forme $\mathsf{H}_{pc}^{Q} = \mathsf{H}_{pc}^{\infty} - 1/d_p \mathbf{e}_{pc}\mathbf{v}_p^\top$ où H_{pc}^{∞} représente l'homographie induite par le plan à l'infini (3.35), $\mathbf{v}_p = \mathsf{K}_p^{-T}\mathbf{n}_p$ représente la ligne de fuite dans la vue v_p du plan et $\mathbf{e}_{pc}$ est le vecteur exact de l'épipole, image du centre de la caméra v_p dans la vue v_c. Il est facile de montrer que la matrice

$$\mathsf{H}_{pc}(x) = \mathsf{H}_{pc}^{Q} + x\mathbf{e}_{pc}\mathbf{v}_p^\top, \qquad \forall x \in \mathbb{R} \tag{3.50}$$

représente une homographie induite par un plan parallèle [2] à Q.

En substituant (3.50) dans (3.48) nous pouvons écrire

$$\begin{aligned}[\tilde{\mathbf{e}}_{pc}]_\wedge \left([\tilde{\mathbf{x}}'_{ck} \mid \tilde{\mathbf{y}}'_{ck}] - (\widetilde{\mathsf{H}}_{pc}^{Q} + x\mathbf{e}_{pc}\mathbf{v}_p^\top)[\tilde{\mathbf{x}}'_{ck} \mid \tilde{\mathbf{y}}'_{ck}]\right) &= \mathsf{0}_{3\times 2} \\ \Longrightarrow [\tilde{\mathbf{e}}_{pc}]_\wedge \left([\tilde{\mathbf{x}}'_{ck} \mid \tilde{\mathbf{y}}'_{ck}] - \widetilde{\mathsf{H}}_{pc}^{Q}[\tilde{\mathbf{x}}'_{ck} \mid \tilde{\mathbf{y}}'_{ck}]\right) &= \mathsf{0}_{3\times 2}\end{aligned}$$

ce qui veut dire que nous pouvons remplacer dans les équations (3.46-3.49) la matrice $\widetilde{\mathsf{H}}_{pc}$ par la matrice $\widetilde{\mathsf{H}}_{pc}^{Q}(x)$ de l'homographie induite par un plan *arbitraire* contenant la paire de points complexes, c.-à-d. avec une valeur arbitraire de x, sans affecter ces équations. ■

On lève l'ambiguïté sur $\widetilde{\mathsf{H}}_{pc}$ en fixant x dans (3.50) de telle façon que deux éléments de $\widetilde{\mathsf{H}}_{pc}(x)$ soient égaux (il est aisé de montrer l'existence d'une telle valeur de x). Par conséquent, dans les équations (3.46-3.49), la matrice $\widetilde{\mathsf{H}}_{pc}$ est paramétrée par 8 scalaires au lieu de 9 pour un paramétrage minimal.

Il est possible de définir un paramétrage minimal différent de $\widetilde{\mathsf{H}}_{pc}$ dans le cas où la matrice fondamentale est connue. Ce paramètrage est le suivant :

$$\widetilde{\mathsf{H}}_{pc}(\mathbf{y}) = [\mathbf{y}]_\wedge \widetilde{\mathsf{F}}_{pc} + \tilde{\mathbf{e}}_{pc}\tilde{\mathbf{e}}_{pc}^\top \tag{3.51}$$

où $\mathbf{y}$ est un vecteur non nul de $\mathbb{R}^3$ à deux degrés de liberté.

Ce paramètrage découle de la décomposition $\widetilde{\mathsf{H}}_{pc} = [\mathbf{l}_c]_\wedge \widetilde{\mathsf{F}}_{pc} + y_0\tilde{\mathbf{e}}_{pc}\mathbf{l}_p^\top$ [Hartley 2004b, p. 332] où $(\mathbf{l}_c \leftrightarrow \mathbf{l}_p)$ représente n'importe quelle correspondance de droites et y_0 un réel non nul. Pour un paramètrage minimal, nous pouvons considérer n'importe quelle droite de la vue v_c et donc, en particulier, la droite de vecteur $\mathbf{l}_p = \tilde{\mathbf{e}}_{pc}$. Ce choix se justifie par le fait que la droite considérée ne doit pas contenir l'épipôle.

2. En particulier, si $x = \frac{y-1}{yd_p}$ alors ce plan a pour vecteur $\mathbf{q}' = (\mathbf{n}_p^\top, yd_p)^\top$.

3.7.1.2 Résolution

Pour un ensemble de K points composé de $N = K - 1$ points naturels réels et de $J = 1$ paire de points cycliques, les équations (3.46), (3.47) et (3.49) peuvent se réécrire sous la forme

$$\mathtt{D}\mathbf{X} = \mathbf{0}_{9+3K} \tag{3.52}$$

où le vecteur

$$\mathbf{X} = \begin{bmatrix} \mathrm{vec}(\widetilde{\mathsf{H}}_{cp})_{/1} \\ a_{cj} \\ b_{cj} \\ \vdots \\ a_{ck_n} \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{10+K} \tag{3.53}$$

a pour composantes les inconnues, à savoir les 9 éléments de $\widetilde{\mathsf{H}}_{uv}$ (traitée colonne par colonne) privés du premier, les 2 facteurs d'échelle a_{vj}, b_{vj} et les facteurs d'échelle a_{ck_n}, $k_n \in \mathrm{K} \setminus \{j\}$, et où D est la matrice par blocs de dimension $((9 + 3(K-1)) \times 10 + K)$ tel que

$$\mathsf{D} = [(\mathbf{a}_1 + \mathbf{a}_2)\ \mathbf{a}_3\ \ldots\ \mathbf{a}_{11+K}] \tag{3.54}$$

où $\mathbf{a}_n$ représente la n-ème colonne de la matrice par blocs

$$\mathsf{A} = \begin{bmatrix} \mathsf{B} \\ \mathsf{E} \\ \vdots \\ \mathsf{C}_n \\ \vdots \end{bmatrix} \tag{3.55}$$

concaténant le bloc de dimension $6 \times (11 + K)$

$$\mathsf{B} = \begin{bmatrix} \mathrm{diag}(\tilde{\mathbf{x}}_{cj}, \tilde{\mathbf{x}}_{cj}, \tilde{\mathbf{x}}_{cj})^\top & -\tilde{\mathbf{x}}'_{pj} & -\tilde{\mathbf{y}}'_{pj} & \mathsf{0}_{3\times K} \\ \mathrm{diag}(\tilde{\mathbf{y}}_{cj}, \tilde{\mathbf{y}}_{cj}, \tilde{\mathbf{y}}_{cj})^\top & -\tilde{\mathbf{y}}'_{pj} & \tilde{\mathbf{x}}'_{pj} & \mathsf{0}_{3\times K} \end{bmatrix}, \tag{3.56}$$

le bloc de dimension $3 \times (11 + K)$

$$\mathsf{E} = \begin{bmatrix} \mathrm{diag}(\tilde{\mathbf{e}}_{cp}, \tilde{\mathbf{e}}_{cp}, \tilde{\mathbf{e}}_{cp})^\top & \mathsf{0}_{3\times 1+K} & -\tilde{\mathbf{e}}_{pc} \end{bmatrix} \tag{3.57}$$

et les blocs de dimension $3 \times (11 + K)$

$$\mathsf{C}_n = \begin{bmatrix} -({\tilde{\mathbf{x}}'_{pk_n}}^\top \otimes [\tilde{\mathbf{e}}_{pc}]_\wedge) & \mathsf{0}_{3\times 2+n} & ([\tilde{\mathbf{e}}_{pc}]_\wedge \tilde{\mathbf{x}}'_{pk_n}) & \mathsf{0}_{3\times K-n} \end{bmatrix}. \tag{3.58}$$

Le système obtenu comporte $9 + 3(K-1)$ équations linéaires dont $9 + 2(K-1)$ sont in-

dépendantes, pour $9 + K$ inconnues. Les points cycliques du plan apportent 6 équations, la mise en correspondance des épipoles un nombre de 3 équations et les images des $K - 1$ points naturels un nombre d'équations indépendantes égal à $2(K - 1)$. Ainsi, une solution exacte existe si $9 + 2(K - 1) = 9 + K$, c.-à-d. pour $K = 2$. En présence de bruit, une solution au sens des moindres carrés existe si l'on dispose des points cycliques de $J = 1$ plan et d'un nombre minimal de $N = 1$ points naturels « visibles » dans deux vues.

L'algorithme itératif complet est donné dans la table 3.3.

PROBLÈME :
Mettre à l'échelle les blocs de la matrice des données $\widetilde{\mathcal{X}}$ en calculant un ensemble $\{\mu_{vj}\}$ de scalaires complexes et un ensemble $\{a_{vk_n}\}$ de scalaires réels tels que la matrice des données « mises à l'échelle » $\widetilde{\mathcal{X}}'$ vérifie la factorisation 3.23 en l'absence de bruit de mesure.

ALGORITHME :

1. Sélectionner une vue v_0 selon une heuristique H.
2. ***Tant que*** il reste des vues non traitées
   (a) Sélectionner un couple de vue (v_p, v_c) selon H.
   (b) Contruire la matrice D définie en (3.54).
   (c) Résoudre $\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathsf{D}\mathbf{X}\|_2^2$ s.c. $\|\mathbf{X}\|^2 = 1$.
   (d) Déterminer $(\hat{a}_{cj}, \hat{b}_{cj})$ et $\{\hat{a}_{ck_n}\}$ à partir de $\hat{\mathbf{X}}$.
   (e) Multiplier le bloc (c, j) par $\begin{bmatrix} \hat{a}_{cj} & \hat{b}_{cj} \\ -\hat{b}_{cj} & \hat{a}_{cj} \end{bmatrix}$ et chacun des blocs (c, k_n) par $\hat{a}_{ck_n}$.

Fin Tant que

TABLE 3.3: Algorithme de mise à l'échelle des images d'une paire de points cycliques en présence de points naturels.

3.7.1.3 Algorithme de prédiction des blocs manquants de la matrice des données

Supposons que nous exécutions l'algorithme présenté dans la table 3.3 seulement pour les images visibles des points cycliques, alors certaines entrées (blocs) de la matrice des données $\widetilde{\mathcal{X}}'$ sont absentes. Le problème traité ici consiste à calculer ces données manquantes mises à l'échelle telles qu'en l'absence de bruit, la factorisation 3.23 soit vérifiée.

Une de nos contributions est de proposer un algorithme (présenté dans la table 3.4) qui résoud ce problème et de fournir sa preuve. Supposons que les points cycliques ne soient pas visibles dans une certaine vue v mais visibles dans un ensemble $L \subseteq \{1, \ldots, V\} \setminus v$ de $|L|$ vues, dont les éléments sont désignés par u_s où $s = 1..|L|$. Afin d'alléger la description de l'algorithme de prédiction, nous supposons ici que l'ensemble des images des points naturels sont mis à l'échelle, le calcul de leurs facteurs d'échelle se déduisant aisément de la méthode proposée. Le principe de l'algorithme repose sur l'existance d'une solution linéaire basée sur trois équations impliquant l'ensemble des images des points naturels mises à l'échelle $(\tilde{\mathbf{x}}'_{u_s k_n}, \tilde{\mathbf{x}}'_{v k_n})$ en correspondance dans

les vues u_s et v. La première consiste en la réécriture de l'équation (3.46) sous la forme

$$\mathtt{D}_v - \widetilde{\mathsf{H}}_{u_s v} \left[\tilde{\mathbf{x}}'_{u_s j} | \tilde{\mathbf{y}}'_{u_s j}\right] = \mathbf{0}_{3\times 2} \tag{3.59}$$

qui contraint linéairement les éléments de $\mathtt{D}_v$, matrice inconnue (de dimension 3×2) représentant les images mises à l'échelle des points cycliques manquants, c.-à-d. telle que $\mathtt{D}_v = \left[\tilde{\mathbf{x}}'_{vj} | \tilde{\mathbf{y}}'_{vj}\right]$, et les huit éléments de chacune des homographies $\widetilde{\mathsf{H}}_{u_s v}$. Les deuxième et troisième équations sont respectivement les équations (3.47) et (3.49) qui contraignent linéairement les huit éléments de chacune des homographies $\widetilde{\mathsf{H}}_{u_s v}$. Il s'agit d'un système d'équations linéaire non-homogène d'inconnues les éléments de $\mathtt{D}_v$ ainsi que les huit éléments de chacune des matrices $\widetilde{\mathsf{H}}_{u_s v}$.

Solution minimale. Pour un ensemble de K points composé de $N = K - 1$ points naturels réels et de $J = 1$ paire de points cycliques « visible » dans $|L|$ vues les équations (3.59), (3.47) et (3.49) peuvent se réécrire sous la forme

$$\mathtt{A}\mathbf{X} = \mathbf{b} \tag{3.60}$$

où le vecteur

$$\mathbf{X} = \begin{bmatrix} \tilde{\mathbf{x}}'_{vj} \\ \tilde{\mathbf{y}}'_{vj} \\ \mathrm{vec}(\mathsf{H}_{u_1 v})_{/1} \\ \vdots \\ \mathrm{vec}(\mathsf{H}_{u_{|L|} v})_{/1} \end{bmatrix} \in \mathbb{R}^{6+8(|L|)} \tag{3.61}$$

a pour composantes les inconnues, à savoir les 3 éléments de chacun des vecteurs $\tilde{\mathbf{x}}'_{vj}$ et $\tilde{\mathbf{y}}'_{vj}$, respectivement les parties réelle et imaginaire des images des points cycliques manquantes, et les $8|L|$ éléments des matrices $\mathsf{H}_{u_s v}$ (traitées colonne par colonne), et où

$$\mathsf{A} = \begin{bmatrix} \mathbf{d}_1 & \dots & (\mathbf{d}_{6+9(s-1)+1} + \mathbf{d}_{6+9(s-1)+2}) & \mathbf{d}_{6+9(s-1)+3} & \dots & \mathbf{d}_{6+9|L|} \end{bmatrix} \tag{3.62}$$

où $\mathbf{d}_n$ représente la $n$ème colonne de la matrice

$$\mathsf{D} = \begin{bmatrix} \mathsf{B}_{u_1} \\ \mathsf{E}_{u_1} \\ \mathsf{C}_{u_1} \\ \vdots \\ \mathsf{B}_{u_{|L|}} \\ \mathsf{E}_{u_{|L|}} \\ \mathsf{C}_{u_{|L|}} \end{bmatrix} \tag{3.63}$$

où

$$\mathsf{B}_{u_s} = \begin{bmatrix} \mathsf{I}_6 & \mathsf{0}_{6\times 9(s-1)} & -\mathrm{diag}(\tilde{\mathbf{x}}'_{u_s}, \tilde{\mathbf{x}}'_{u_s}, \tilde{\mathbf{x}}'_{u_s})^\top & \mathsf{0}_{6\times 9(|L|-s)} \\ & & -\mathrm{diag}(\tilde{\mathbf{y}}'_{u_s}, \tilde{\mathbf{y}}'_{u_s}, \tilde{\mathbf{y}}'_{u_s})^\top & \end{bmatrix}, \tag{3.64}$$

où

$$\mathsf{E}_{u_s} = \left[\begin{array}{ccc} \mathsf{0}_{3\times 6+9(s-1)} & \mathrm{diag}(\tilde{\mathbf{e}}'_{vu_s}, \tilde{\mathbf{e}}'_{vu_s}, \tilde{\mathbf{e}}'_{vu_s})^\top & \mathsf{0}_{3\times 9(|L|-s)} \end{array} \right], \tag{3.65}$$

et où

$$\mathsf{C}_{u_s} = \left[\begin{array}{ccc} & (\tilde{\mathbf{x}}'_{u_s k_1}{}^\top \otimes [\mathbf{e}_{u_s v}]_\wedge) & \\ \mathsf{0}_{3(J-1)\times(6+9(s-1))} & \vdots & \mathsf{0}_{3(J-1)\times 9(|L|-s)} \\ & (\tilde{\mathbf{x}}'_{u_s k_N}{}^\top \otimes [\mathbf{e}_{u_s v}]_\wedge) & \end{array} \right], \tag{3.66}$$

et où

$$\mathbf{b} = \left[\begin{array}{c} \mathbf{q}_{u_1} \\ \vdots \\ \mathbf{q}_{u_{|L|}} \end{array} \right] \tag{3.67}$$

où

$$\mathbf{q}_{u_s} = \left[\begin{array}{c} \mathbf{0}_6 \\ \tilde{\mathbf{e}}_{u_s v} \\ ([\mathbf{e}_{u_s v}]_\wedge \mathbf{x}'_{u_s k_1})^\top \\ \vdots \\ ([\mathbf{e}_{u_s v}]_\wedge \mathbf{x}'_{u_s k_N})^\top \end{array} \right]^\top. \tag{3.68}$$

L'ensemble des données consistant en les images (mises à l'échelle) dans $|L|$ vues des points cycliques de $J = 1$ plan et de $N = K - 1$ points naturels introduisent $8|L| + 6$ inconnues et fournissent $9|L| + 2N|L|$ équations. Ainsi, pour $|L| = 2$, $N = 1$ est le nombre minimal de points naturels nécessaire pour la résolution du système. L'algorithme complet est présenté dans la table 3.4.

PROBLÈME : Calculer un bloc manquant associé à une certaine vue v de la matrice des données « mises à l'échelle » $\widetilde{\mathcal{X}}'$ des points cycliques d'un plan « visibles » dans $|L| \geq 2$ vues u_s où $u_s \neq v$.

ALGORITHME :

1. Contruire la matrice A définie en (3.62) et le vecteur $\mathbf{b}$ défini en (3.67).
2. Résoudre $\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathsf{A}\mathbf{X} - \mathbf{b}\|_2^2$.
3. Déterminer les vecteurs $\tilde{\mathbf{x}}'_{vj}$ et $\tilde{\mathbf{y}}'_{vj}$ à partir de $\hat{\mathbf{X}}$.
4. Assigner le bloc (v, j) mis à l'échelle à $\left[\tilde{\mathbf{x}}'_{vj} | \tilde{\mathbf{y}}'_{vj}\right]$.

TABLE 3.4: Algorithme du calcul de l'image d'une paire de points cycliques manquante en présence de points naturels.

Notons qu'il est aussi possible de déterminer les images mises à l'échelle des points cycliques manquantes en les calculant tout d'abord à un facteur d'échelle près à partir des images disponibles et des matrices fondamentales [Hartley 2004b, p. 380] puis en les mettant à l'échelle comme décrit dans 3.7.1.1. Cependant, cette méthode a été testée en simulation et présente des résultats nettement moins bons que ceux fournis par la méthode proposée.

3.7.1.4 Heuristique de sélection de vues

Lorsque nous disposons d'une seule paire de points cyclique et d'un ensemble complet d'images de points naturels, les couples de vues sélectionnés sont simplement pris dans leur ordre chronologique. Dans le cas de $J > 1$ paires de points cycliques, il s'agit de la même heuristique que celle présentée dans la section 3.6.1.4.

3.7.2 Reconstruction euclidienne post-factorisation de la structure et du mouvement

Le sous-problème consiste à calculer une reconstruction euclidienne de la structure et du mouvement, c'est à dire à *calculer l'ambiguïté projective représentée par la transformation* T telle que, en l'absence de bruit,

$$\mathsf{T}^\top \hat{\mathsf{U}}^\top = [\dots s_v(\mathsf{K}_v \mathsf{R}_v[\,\mathsf{I} \mid -\mathbf{c}_v])^\top \dots]^\top \tag{3.69}$$

$$\mathsf{T}\hat{\mathsf{V}} = \begin{bmatrix}[\mathbf{X}_1|\mathbf{Y}_1] & \dots & [\mathbf{X}_K|\mathbf{Y}_K]\end{bmatrix}. \tag{3.70}$$

La résolution de ce problème fait l'objet du chapitre 4.

3.8 Résultats expérimentaux

Nous présentons dans cette section les résultats fournis par la méthode proposée pour les premier et deuxième cas d'usage.

3.8.1 Résultats du premier cas d'usage

Résultats sur données synthétiques. Nous avons comparé la technique de factorisation proposée avec celle décrite dans [Sturm 2000], désignée par STURM dans ce qui suit, que nous avons considérée comme étant le travail le plus proche de la méthode proposée. En outre, le lecteur doit garder à l'esprit que dans STURM les caméras sont supposées calibrées, (ce qui signifie que les matrices de calibrage K_v dans l'équation (3.32) sont connues). Ainsi, les données d'entrée consistent en un ensemble d'homographies « calibrées » liant chacun des plans au plan image qui doivent seulement être mises à l'échelle (de façon triviale) par un facteur d'échelle réel et non un facteur d'échelle complexe. Ceci offre un réel avantage à [Sturm 2000] en termes de précision mais notre intention a plutôt été de montrer que la méthode proposée présente des performances similaires.

Chaque simulation a été effectuée sur un ensemble de 20 vues de 4 plans non coplanaires. Les caméras ont été placées sur une trajectoire circulaire à une distance D du centre de la scène et en supposant que le facteur d'échelle est défini de telle sorte que le coté du marqueur plan carré soit égal à 1. Les paramètres internes des caméras sont fixés à des valeurs communes : $f = 1500$, $u_0 = 640$ et $v_0 = 360$ (les pixels sont supposés carrés). Les homographies liant chacun des plans de la scène vers le plan image sont calculées à partir de $10 \times 10 = 100$ points mis en correspondance par plan. Les parties réelle et imaginaire des images des points cycliques correspondent aux deux premières colonnes de chacune des homographies estimées, sur laquelle nous avons appliqué une similarité plane. Un bruit gaussien indépendant centré en zero a été ajouté aux points image en faisant varier son écart type sur différents niveaux. Pour chaque niveau de bruit, 100 tests ont été effectués et les moyennes des résultats sont affichées. L'ensemble des résultats est présenté sur la figure 3.4.

Résultats sur données réelles. Nous avons disposé dans une scène des marqueurs constitués de cercles concentriques, nommés C^{2}Tags (présentés dans le chapitre 5), fournissant les images des points cycliques. Les résultats obtenus sont présentés sur la figure 3.5. L'expérience a été reconduite avec les marqueurs de l'ARToolKit [Poupyrev 2000] dont les résultats sont présentés sur la figure 3.6. Dans cette deuxième expérience, les images des points cycliques sont données par $[\tilde{\mathbf{x}}_{vp}|\tilde{\mathbf{y}}_{vp}] = [\mathbf{h}^1_{vp}|\mathbf{h}^2_{vp}]$ où $\mathbf{h}^j_{vp}$ représente la colonne numéro j de l'homographie H_{vp} sous la forme de l'équation (3.3).

3.8.2 Résultats du deuxième cas d'usage

Résultats sur données synthétiques Les performances en simulation de la méthode proposée sont présentées à la fin du chapitre suivant, dans la section 4.4.1.

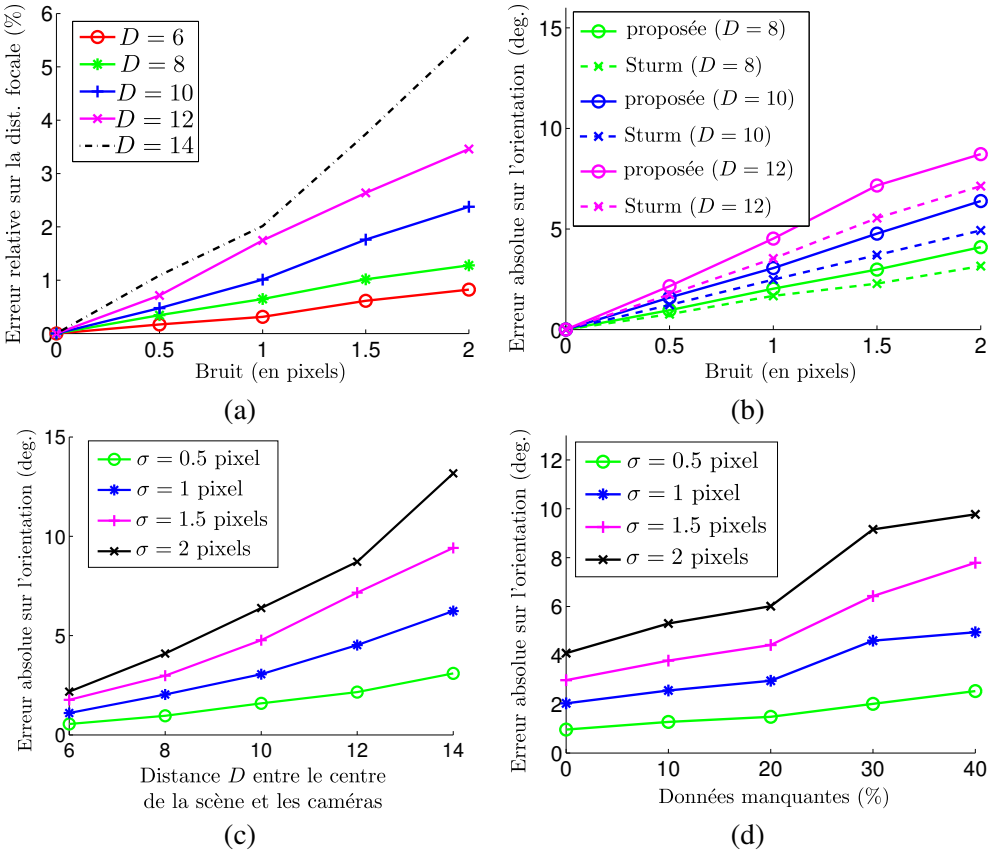
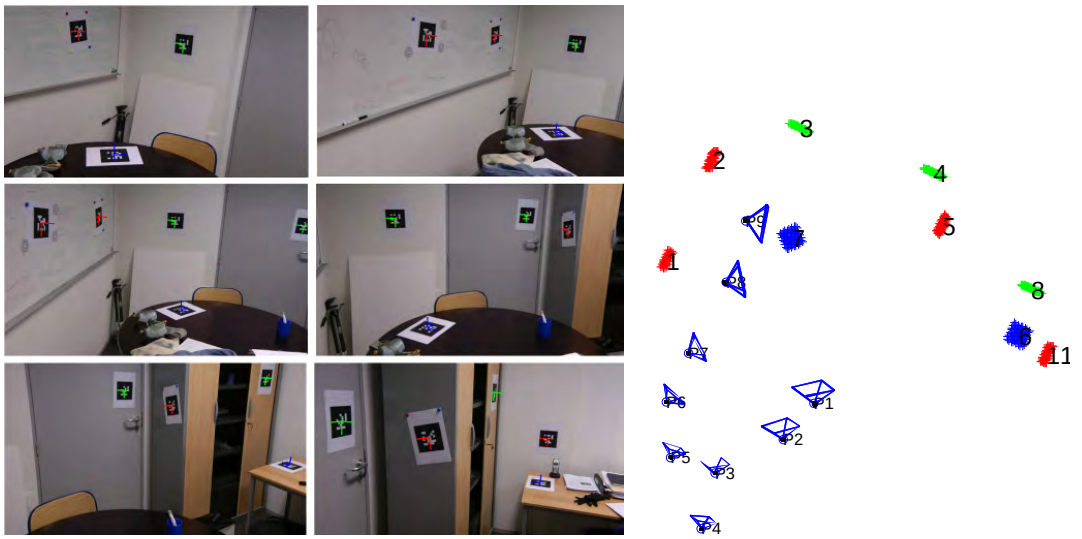


FIGURE 3.4: (a) Erreur sur la distance focale *vs.* le niveau de bruit. (b) Erreur sur les orientations. (c) Erreur sur les orientations *vs.* la distance des caméras au centre de la scène. (d) Erreur sur les orientations *vs.* la quantité de données manquantes.



FIGURE 3.5: (a) Vues-clés sélectionnées pour lesquelles chacun des C^2Tags a été augmenté par un repère dont la couleur est associée à l'indice de la classe de marqueurs cycliques appariés. (b) Reconstruction 3D euclidienne des marqueurs et des caméras.



(a) (b)

FIGURE 3.6: (a) Vues-clés sélectionnées pour lesquelles chacun des marqueurs du ARToolKit a été augmenté par un repère dont la couleur est associée à l'indice de la classe de marqueurs cycliques appariés (b) Reconstruction 3D euclidienne des marqueurs et des caméras.

Résultats sur données réelles. La méthode proposée a été testée sur des ensembles de vues extraites de plusieurs séquences vidéo. Nous présentons ici seulement une de ces expérimentations pour laquelle l'utilisation des images des points cycliques dans le schéma de factorisation s'est avéré être indispensable, quel que soit la méthode d'autocalibrage utilisée (présentée dans le chapitre 4). Les autres expérimentations sont présentées conjointement aux résultats fournis dans le chapitre 4 dédié au problème de l'autocalibrage. Il s'agit de la reconstruction d'un visage effectuée à partir de 17 points naturels, extraits via l'algorithme itératif KLT [Lucas 1985], dans 5 vues. Les images des points cycliques sont deux points d'intersection (complexes) des images des cercles parallèles formés par les contours des iris (mis en évidence sur la figure 3.7(a)). Ces dernières sont extraites dans seulement 3 vues sur les 5. À l'issue du calcul des matrices de projection euclidiennes, des reconstructions denses ont été calculées via l'utilisation du logiciel PMVS (décrit dans [Furukawa 2010]) afin d'évaluer visuellement les performances de la méthode proposée. La reconstruction dense obtenue à partir des matrices de projection euclidiennes calculées sans les points cycliques dans le schéma de factorisation est aberrante tandis que la méthode proposée présente de bonnes performances. Cette dernière est illustrée sur la figure 3.7(b).

3.9 Conclusion

L'objet de ce chapitre était tout d'abord de présenter l'intérêt d'utiliser les images de points cycliques pour le calcul de la structure et du mouvement reposant sur un schéma de factorisation. Nous avons vu que l'utilisation de telles primitives comme entrées de la matrice des données pouvait constituer une solution au problème des données manquantes. Les points cycliques ont des propriétés géométriques intéressantes nous permettant de résoudre les deux problèmes critiques

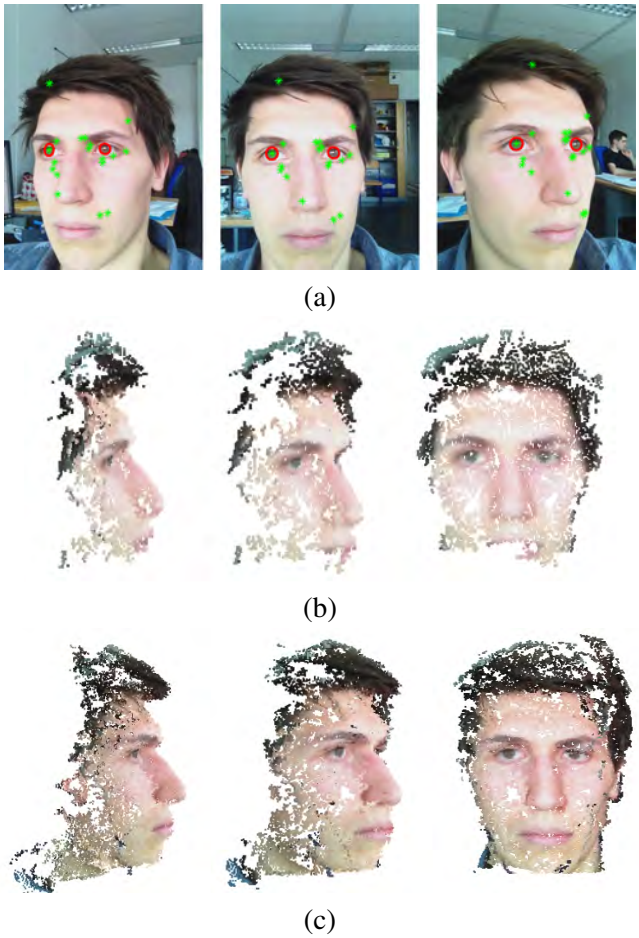


FIGURE 3.7: (a) 3 vues d'entrée. Les images des points naturels utilisés sont inscrites en vert. Les points cycliques considérés sont ici les points d'intersection des cercles parallèles, contours des iris marqués en rouge. Le nombre de vues total est de 5. Les images des points cycliques sont extraites dans seulement 3 vues (présentées en (a)), induisant ainsi la présence de blocs manquants dans la matrice des données. (b) Reconstruction dense obtenue à partir des matrices de projection euclidiennes associées aux 5 images. (c) Reconstruction dense obtenue à partir des matrices de projection euclidiennes pour un ensemble de 11 images calculées via un algorithme incrémental de calcul de la structure et du mouvement ayant comme initialisation la reconstruction euclidienne obtenue à partir des 5 vues (associée au résultat (b)).

des approches de reconstruction par factorisation, à savoir la mise à l'échelle des entrées de la matrice des données et le calcul des données manquantes. Les algorithmes traitant de ces deux problèmes, avec ou sans la présence de points naturels, ont été décrits et prouvés. Les expérimentations ont montré la validité des algorithmes proposés et leurs performances.

Perspectives. Il serait intéressant de proposer une méthode permettant de détecter les marqueurs cycliques dont les plans de support sont parallèles afin de déterminer automatiquement l'ensemble des classes de marqueurs cycliques appariés. Ces classes de marqueurs sont pour le moment renseignées par l'utilisateur où nécessitent une disposition spécifique des marqueurs dans la scène.

Un des axes de recherche que nous envisageons est l'unification des algorithmes de mise à l'échelle et de prédiction des blocs manquants à partir des équations linéaires proposées dans ce chapitre pour la réalisation de ces deux tâches, ceci y compris en présence d'images de points naturels manquantes.

Chapitre 4

Rectification euclidienne d'une reconstruction projective

Dans ce chapitre, nous rappelons tout d'abord en quoi consiste le problème d'*autocalibrage* d'une caméra. Un état de l'art sur le sujet est ensuite proposé. Nous rappelons les équations linéaires sur la conique absolue duale, décrites dans [Pollefeys 1999], permettant de résoudre le problème dans l'espace 3D projectif sous les hypothèses d'une caméra dont la forme des pixels et le point principal sont connus. Nous proposons une extension de cette méthode reposant sur les images des points cycliques. Le lien existant entre ces équations et les équations décrites dans [Pollefeys 1999] est établi. Nous proposons enfin un algorithme d'autocalibrage unifié reposant sur l'ensemble de ces équations.

4.1 Introduction

Dans le chapitre précédent, nous avons défini le problème du calcul de la structure et du mouvement comme celui qui consiste à « expliquer » un ensemble « suffisamment grand » d'images de M points dans V vues, de vecteurs homogènes $\mathbf{x}_{vm} \in \mathbb{R}^3$, par une reconstruction tridimensionnelle des V caméras, de matrices de projection $\mathsf{P}_v \in \mathbb{R}^{3\times4}$, et des M points, de vecteurs homogènes $\mathbf{X}_m \in \mathbb{R}^4$, conformément à l'équation de projection

$$\mathbf{x}_{vm} \overset{\mathbb{C}}{\sim} \mathsf{P}_v\mathbf{X}_m, \tag{4.1}$$

où $\overset{\mathbb{C}}{\sim}$ désigne l'égalité projective complexe définie par l'équation (3.6). Dans le deuxième cas d'usage du chapitre précédent, la spécificité de notre problème a été de traiter simultanément des images de points naturels et de $J \geq 1$ paires de points cycliques d'une scène rigide. Une solution a été proposée en utilisant une technique de factorisation à partir de la matrice par blocs associés aux images des points, après avoir décrit une procédure originale de mise à l'échelle complexe de ses blocs complexes. L'algorithme que nous avons proposé a permis d'obtenir un ensemble de solutions projectivement équivalentes tel que si $(\{\mathsf{P}_v\}, \{\mathbf{X}_m\})$ est une solution au problème de reconstruction, alors $(\{\mathsf{P}_v\mathsf{T}\}, \{\mathsf{T}^{-1}\mathbf{X}_m\})$ est aussi une solution, où $\mathsf{T} \in \mathbb{R}^{4\times4}$ désigne la matrice d'une homographie arbitraire de l'espace. Cette ambiguïté, exprimée par la matrice T dans l'équation de projection

$$\mathbf{x}_{vm} \overset{\mathbb{C}}{\sim} \mathsf{P}_v\mathbf{X}_m = \mathsf{P}_v\mathsf{T}\mathsf{T}^{-1}\mathbf{X}_m, \tag{4.2}$$

nous indique que la solution est une reconstruction tridimensionnelle de la scène mais dans une représentation projective arbitraire.

Le problème du passage d'une reconstruction projective à une reconstruction euclidienne, dit *problème de rectification euclidienne d'une reconstruction projective*, abordé dans ce chapitre, consiste à déterminer la matrice $\mathsf{H} \in \mathbb{R}^{4\times4}$ d'une certaine homographie telle que si $(\{\mathsf{P}_v\}, \{\mathbf{X}_m\})$ est une reconstruction projective alors $(\{\mathsf{P}_v\mathsf{H}\}, \{\mathsf{H}^{-1}\mathbf{X}_m\})$ est une reconstruction euclidienne, c.-à-d. une reconstruction coïncidant avec la « vérité terrain » à une similitude près de l'espace euclidien. En d'autres termes, il s'agit de déterminer une homographie qui permet le passage d'une représentation projective de la scène à une représentation euclidienne de celle-ci, dans laquelle le plan à l'infini et la conique absolue respectent leurs équations canoniques. Un résultat majeur [Hartley 2004b, p. 460] indique qu'une reconstruction projective $(\{\mathsf{P}_v\}, \{\mathbf{X}_m\})$ dans laquelle $\mathsf{P}_1 = [\mathsf{I}_3 \mid \mathbf{0}_3]$ peut être transformée en une reconstruction euclidienne $(\{\mathsf{P}_v\mathsf{H}\}, \{\mathsf{H}^{-1}\mathbf{X}_m\})$ par une homographie dont la matrice H est de la forme

$$\mathsf{H} = \begin{bmatrix} p\mathsf{K}_1 & \mathbf{0} \\ -\mathbf{p}^\top\mathsf{K}_1 & p \end{bmatrix} \tag{4.3}$$

où K_1 désigne la matrice de calibrage dans la vue 1 et $\boldsymbol{\pi}_\infty = [\mathbf{p}^\top, p]^\top$ est le vecteur du plan à l'infini dans la représentation projective considérée.

D'un point de vue géométrique, on peut dire qu'il existe deux classes d'algorithmes permettant de déterminer la matrice H. Une première approche, connue sous le nom « d'approche stratifiée », consiste à déterminer *séparément*, dans la représentation projective considérée,

- le vecteur $\boldsymbol{\pi}_\infty$ du plan à l'infini (cf. 2.2.7.1) et,
- la matrice de la conique absolue Ω_∞ relativement à ce plan ou, de façon équivalente, son image dans une certaine vue v représentée par la matrice $\boldsymbol{\omega}_v = \mathsf{K}_v^{-\top}\mathsf{K}_v^{-1}$, dont on extrait la matrice de calibrage K_v (cf. 2.2.7.2).

Une deuxième approche, appelée « approche directe » par opposition, consiste à estimer *simultanément* ces inconnues.

Dans ce chapitre, nous nous intéressons à la rectification euclidienne d'une reconstruction projective par une approche directe et en particulier au problème de l'autocalibrage des caméras. Ce problème consiste à déterminer la matrice H en n'utilisant que des contraintes découlant d'un certain nombre d'hypothèses sur les paramètres internes de la caméra. Triggs dans [Triggs 1997] a été le premier à énoncer le problème « direct » de l'autocalibrage d'une caméra comme celui de l'identification, dans l'espace projectif dual, du « cône dual » à la conique absolue (par « cône dual », nous désignons, comme dans [Sturm 1997], une quadrique duale dégénérée de rang 3) de signature $(3, 0)$, communément appelée « quadrique absolue duale » [Hartley 2004b]. Le résultat-clé est que la matrice (d'ordre 4 et de rang 3) de la quadrique absolue duale admet la décomposition suivante

$$\mathsf{Q}_\infty^* \sim \mathsf{H} \begin{bmatrix} \mathsf{I}_3 & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} \mathsf{H}^\top$$

où H est l'homographie de rectification euclidienne introduite précédemment. Les équations de base de cet « autocalibrage dual » sont dérivées de l'équation de projection de la conique absolue Ω_∞ dans la vue v sous sa forme duale

$$\boldsymbol{\omega}_v^* \sim \mathsf{P}_v \mathsf{Q}_\infty^* \mathsf{P}_v^\top \tag{4.4}$$

où $\boldsymbol{\omega}_v^*$, qui est une matrice symétrique d'ordre 3, représente l'image duale de Ω_∞. Pour résoudre ce problème,

- Triggs utilise une méthode dite quasi-linéaire, résolvant un système d'équations linéaire par rapport aux composantes de Q_∞^* et de $\boldsymbol{\omega}^*$, « sous-optimale » dans le sens où il y a un « sur-paramétrage » , c.-à-d. plus d'inconnues que de degrés de liberté.
- Sous l'hypothèse d'une caméra aux pixels carrés et de distance focale inconnue, Pollefeys *et al.* dans [Pollefeys 1999] ont proposé une formulation linéaire de l'autocalibrage dual, qui sera discutée plus loin.

4.1.1 Notre problème spécifique

Dans ce chapitre, nous décrivons une méthode de rectification euclidienne d'une reconstruction projective intégrant des points cycliques, qui réunit, à partir d'un formalisme unifié, les avantages de la formulation linéaire de l'autocalibrage telle que proposée par Pollefeys *et al.* dans

[Pollefeys 1999], [Seo 2004] avec ceux du calibrage plan [Sturm 1999, Zhang 1999, Gurdjos 2006], tout en réduisant leurs inconvénients, en particulier la nécessité d'un calibrage hors-ligne de la caméra.

Illustrons ces propos par les expériences suivantes. Considérons les deux collections d'images (a) et (c) de la figure 4.1 et leurs données associées, et analysons les résultats obtenus à partir des approches d'autocalibrage linéaires présentées dans [Pollefeys 1999, Seo 2004] et, d'autre part, ceux obtenus à partir d'une reconstruction euclidienne fondée sur un calibrage plan [Bouguet , Gurdjos 2006]. Nous constatons que, vis-à-vis des premières approches, l'autocalibrage échoue soit à cause d'un faible nombre de vues ou de correspondances de points naturels, soit à cause d'un mouvement critique de la caméra [Gurdjos 2009]. D'autre part, nous constatons également que la deuxième approche (calibrage plan) échoue sans doute à cause de la petite taille du motif plan utilisé et/ou à cause de son absence dans un certain nombre de vues. La méthode de rectification euclidienne proposée dans ce chapitre nous a ainsi permis d'obtenir les reconstructions denses, illustrées sur les figures 4.1(b)(d), telles qu'elles ont été calculées par le logiciel PMVS [Furukawa 2010] à partir des caméras euclidiennes associées à chacune des vues.

L'idée clé de ce travail est d'utiliser les images des points cycliques dans l'étape d'autocalibrage. Nous allons décrire en quoi les contraintes d'autocalibrage « classiques » proposées par Pollefeys peuvent être combinées avec des contraintes additionelles dérivées des images des points cycliques. Nous fournissons les fondements théoriques démontrant que les contraintes proposées par Pollefeys et celles proposées dans ce chapitre expriment le fait que la rétroprojection de l'image d'une paire de points cycliques est une paire de droites 3D intersectant la conique absolue, c.-à-d. isotropes conformément à la définition 22. L'assemblage des droites 3D isotropes est une variété quadratique de l'espace projectif de dimension 5, appelée *complexe quadratique absolu*, introduite dans [Ponce 2005], [Valdès 2005] (cf. état de l'art §4.2), pour autocalibrer une caméra. Cela signifie, aussi surprenant que cela puisse paraître, que les méthodes d'autocalibrage de Pollefeys *et al.* [Pollefeys 1999], celles proposées, mais aussi celles fondées sur la complexe quadratique absolue reposent en réalité sur le même paradigme.

Ce travail a été publié dans [Calvet 2013].

Organisation du chapitre. Un état de l'art sur les méthodes d'autocalibrage est tout d'abord présenté. Dans une deuxième partie, la formulation linéaire du problème d'autocalibrage proposée dans [Pollefeys 1999] est rappelée. Nous proposons ensuite une formulation linéaire étendue du problème intégrant les images de points cycliques. Nous établissons le lien existant entre les équations proposées et celles décrites dans Pollefeys *et al.*. Enfin, nous présentons les résultats expérimentaux validant la méthode proposée.

4.2 État de l'art

L'autocalibrage d'une caméra est le problème qui consiste à déterminer une homographie de rectification euclidienne de l'espace projectif associé à la scène, à partir de contraintes dérivées d'un certain nombre d'hypothèses sur la caméra. On peut classer les techniques d'autocalibrage

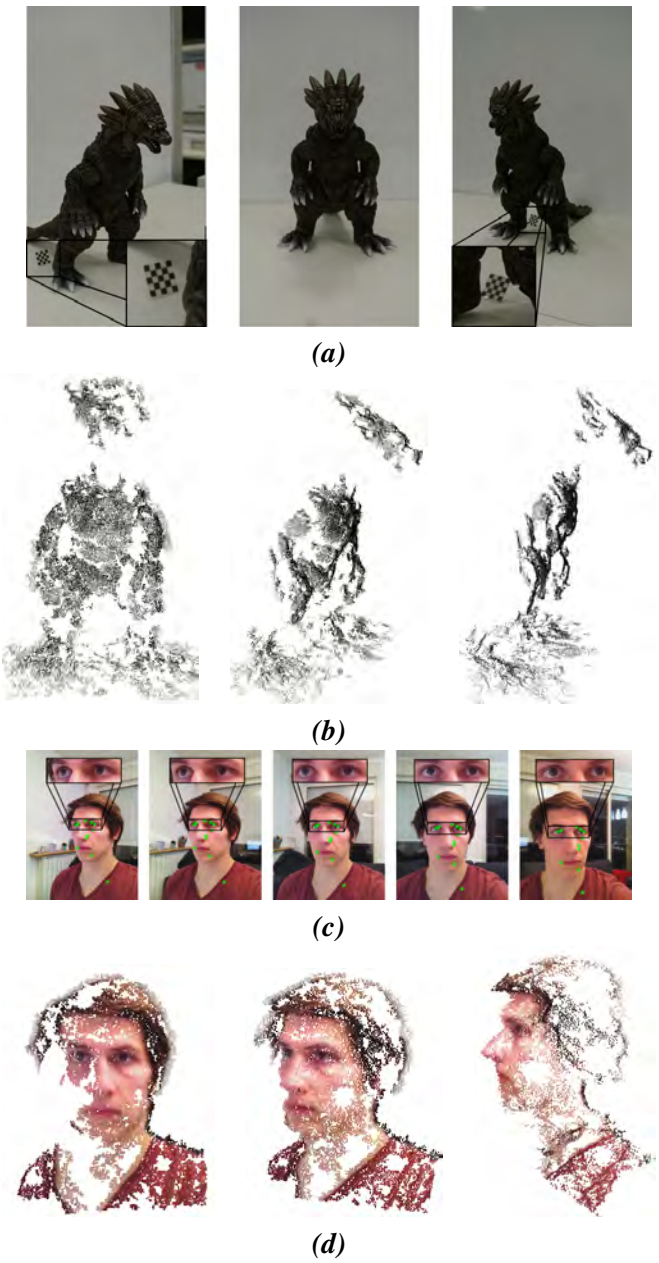


FIGURE 4.1: Mise en évidence des intérêts de la méthode d'autocalibrage proposée utilisant simultanément l'ensemble des contraintes issues des matrices de projection et des images des points cycliques au regard de la méthode d'autocalibrage dual « classique » ou celle reposant sur un calibrage plan. (a) Trois vues d'un dragon où 54 points peuvent être mis en correspondance et où une grille, sous la forme d'un damier de très faible dimension, est visible à l'exception de la deuxième vue. Les images des points cycliques sont issues du calcul de l'homographie liant le plan de support de la grille au plan image. (c) Cinq vues d'un visage où 11 points peuvent être mis en correspondance et les points cycliques considérés sont les deux points d'intersection (complexes) des deux cercles parallèles qui constituent les contours des iris. (b) Aucune reconstruction euclidienne cohérente n'a pu être obtenue à partir d'un calibrage plan, ni à partir de la méthode proposée par Pollefeys [Pollefeys 1999, Pollefeys 2004, Seo 2004]. (b)+(d) Seule la méthode d'autocalibrage proposée, basée sur les mêmes reconstructions projectives que pour les autres méthodes, a pu fournir à PMVS — même pour un motif plan de très faible dimension — des caméras euclidiennes précises. Ceci a abouti aux reconstructions denses visibles ici sous trois points de vues différents.

par la nature de ces hypothèses ; nous ne serons pas exhaustifs ici et nous nous intéresserons qu'aux techniques qui ne font aucune restriction sur les paramètres externes, c.-à-d. sur le mouvement, de la caméra et supposent ses paramètres internes variables au cours de la prise de vue. Nous rapportons quelques résultats marquants.

- D'un point de vue théorique, il a été établi simultanément dans [Heyden 1997] et dans [Pollefeys 1999] que la seule contrainte de pixels de forme rectangulaire constante inconnue (équivalente à celle d'un facteur d'obliquité nul) était une condition suffisante pour qu'une solution existe. Ces résultats majeurs ont néanmoins peu d'intéret pratique car il n'existe pas d'algorithme performant résolvant ce problème.
- L'hypothèse la plus faible pour qu'une solution analytique existe est celle de pixels de forme rectangulaire constante connue (équivalente à celle de pixels carrés). Le problème à résoudre s'énonce alors dans l'espace projectif $\mathcal{P}_5(\mathbb{C})$ de dimension 5, où l'objet recherché est la conique absolue représentée par la variété quadratique des droites isotropes 3D, appelé « *complexe quadratique absolu* » dans [Ponce 2005, Valdés 2004, Valdés 2006]. Le terme « complexe de droites » est synonyme d'assemblage des droites dans $\mathcal{P}_3(\mathbb{C})$; le terme « quadratique » refère au fait que cet assemblage s'exprime comme l'intersection de deux coniques de $\mathcal{P}_5(\mathbb{C})$ et le terme « absolu » refère au fait que ces droites intersectent Ω_∞. Un « point faible » de ce formalisme est une certaine complexité de l'algorithme résolvant le problème avec la nécessité de pouvoir disposer d'un nombre minimal important de vues pour que le problème soit bien posé.
- La seule hypothèse pour qu'une solution linéaire au sens des moindres carrés existe est celle d'une caméra aux pixels carrés de distance focale inconnue. Pollefeys *et al.* dans [Pollefeys 1999] ont formulé le problème dans l'espace projectif dual de dimension 3 où l'objet recherché est la conique absolue représentée par à une variété quadratique de plans (isotropes) de l'espace 3D, c.-à-d. à la quadrique absolue complexe, qui est la seule quadrique qui se projette en un cercle centré au point principal. Dans ce cas, le nombre minimal de vues est 3. Il est à noter que, dans la recherche de la solution, la contrainte de signature $(3, 0)$ sur les quadriques duales « candidates » est omise. Cet omission introduit des singularités, c.-à-d. des solutions ambiguës, généralement induites par certains mouvements de caméra, dits critiques. En particulier, tout mouvement de caméra sur la courbe focale d'une quadrique duale tel que l'axe optique est un axe focal de cette quadrique est critique [Gurdjos 2009].

Que les paramètres internes soient constants ou non, il est à noter que toute contrainte sur ceux-ci peut être imposée en utilisant des algorithmes sophistiqués d'optimisation globale, calculant une « solution garantie » pour le minimum global, à partir de contraintes non linéaires. Les auteurs de [Bocquillon 2007] ont décrit un algorithme d'optimisation fondé sur l'arithmétique d'intervalles qui résoud une spécialisation du problème abordé dans [Pollefeys 1999] dans le sens où la distance focale est supposée constante. Les auteurs de [Chandraker 2007a] proposent un cadre général de résolution fondé sur un critère qu'ils minimisent en utilisant des relaxations convexes et des contraintes de chiralité du plan à l'infini. Ces approches ont été aussi utilisées dans [Fusiello 2004] pour les *contraintes de Kruppa* [Hartley 2004b, p. 469], dans [Chandraker 2007b, Habed 2012]

pour la *contrainte du module* [Hartley 2004b, p. 473].

4.3 Autocalibrage dans l'espace projectif dual

4.3.1 Formulation linéaire de [Pollefeys 1999]

Nous rappelons ici la formulation initiale du problème tel que décrit dans [Pollefeys 1999], ainsi que les hypothèses et les données associées.

Les hypothèses ($\mathcal{H}$) et ($\mathcal{H}'$). Nous nous plaçons dans cette section sous l'hypothèse ($\mathcal{H}$) d'une caméra aux paramètres internes connus à l'exception d'une distance focale variable, c.-à-d. pouvant varier à chaque prise de vue. Sans perte de généralité, nous remplacerons ($\mathcal{H}$) par l'hypothèse équivalente ($\mathcal{H}'$) *d'une caméra à pixels carrés dont le point principal coïncide avec l'origine du repère pixélique.*

La matrice de calibrage K_v de toute caméra respectant l'hypothèse ($\mathcal{H}'$) a la forme diagonale

$$\mathsf{K}_v = \mathrm{diag}\,(f_v, f_v, 1) \tag{4.5}$$

où f_v est la distance focale de la caméra v exprimée en pixels.

Les données $\mathcal{D}$. Les données du problème tel qu'il a été initialement formulé par les auteurs de [Pollefeys 1999] consistent en une configuration de V caméras représentées par l'ensemble $\mathcal{D} = \{\mathsf{P}_v\}_{v=1..V}$ de matrices de projections relativement à une certaine représentation projective de la scène. Toute matrice de projection sera écrite sous la forme

$$\mathsf{P}_v = \begin{pmatrix} \mathbf{a}_v^\top \\ \mathbf{b}_v^\top \\ \mathbf{c}_v^\top \end{pmatrix} \qquad \text{où } v = 1..V, \tag{4.6}$$

où $\mathbf{a}_v^\top$, $\mathbf{b}_v^\top$ et $\mathbf{c}_v^\top$ désignent les lignes de la matrice P_v et où $\mathbf{c}_v$ représente en particulier le plan principal de la caméra v, c.-à-d. le plan parallèle au plan des pixels qui passe par le centre de la caméra.

Problème 28 (autocalibrage dual) *Dans la représentation projective de l'espace projectif dual associée aux données $\mathcal{D}$, il s'agit de rechercher, à partir de $\mathcal{D}$ et sous l'hypothèse ($\mathcal{H}'$), une quadrique duale de signature $(3, 0)$ c.-à-d. dont la matrice Q^* admet une décomposition de la forme*

$$\mathsf{Q}^* \sim \mathsf{H} \begin{bmatrix} \mathsf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3^\top & 0 \end{bmatrix} \mathsf{H}^\top \tag{4.7}$$

où H est la matrice réelle d'ordre 4 d'une certaine homographie de l'espace projectif telle que, dans le cas exact, $\{\mathsf{P}_v\mathsf{H}\}$ représente une reconstruction euclidienne des caméras.

Rappelons qu'il n'existe « en général » qu'une seule quadrique duale qui soit solution de ce problème : il s'agit de la quadrique duale à la conique absolue, c.-à-d. la quadrique absolue duale dont la matrice, dans la représentation projective considérée, sera notée Q_∞^*. Sous sa forme duale, la conique absolue se projette dans la vue v via l'équation

$$\boldsymbol{\omega}_v^* \sim \mathsf{P}_v \mathsf{Q}_\infty^* \mathsf{P}_v^\top \tag{4.8}$$

en une image duale représentée par $\boldsymbol{\omega}_v^*$ qui est une matrice symétrique d'ordre 3.

La matrice $\boldsymbol{\omega}_v^*$ a la propriété remarquable [Hartley 2004b] d'admettre la décomposition $\boldsymbol{\omega}_v^* = \mathsf{K}_v \mathsf{K}_v^\top$ et, sous l'hypothèse $(\mathcal{H}')$, de l'équation (4.5) il découle que $\boldsymbol{\omega}_v^*$ est diagonale de la forme :

$$\boldsymbol{\omega}_v^* = \operatorname{diag}\left((f_v)^2, (f_v)^2, 1\right). \tag{4.9}$$

4.3.1.1 Les équations de base

Les contraintes liées à l'hypothèse $(\mathcal{H}')$, cf. (4.9), peuvent s'écrire

$$\begin{aligned} \boldsymbol{\omega}_{v,12}^* &= 0 \qquad & \boldsymbol{\omega}_{v,11}^* &= \boldsymbol{\omega}_{v,22}^* \\ \boldsymbol{\omega}_{v,13}^* &= 0 \qquad & \boldsymbol{\omega}_{v,23}^* &= 0 \end{aligned}$$

où le scalaire $\boldsymbol{\omega}_{v,jk}^*$ représente l'élément (j,k) de la matrice $\boldsymbol{\omega}_v^*$, à partir de quoi, en considérant l'équation de projection (4.8), [Pollefeys 1999] ont dérivé les quatre équations indépendantes suivantes :

$$\mathbf{a}_v^\top \mathsf{Q}_\infty^* \mathbf{a}_v - \mathbf{b}_v^\top \mathsf{Q}_\infty^* \mathbf{b}_v = 0 \qquad \mathbf{a}_v^\top \mathsf{Q}_\infty^* \mathbf{b}_v = 0 \tag{4.10}$$

$$\mathbf{a}_v^\top \mathsf{Q}_\infty^* \mathbf{c}_v = 0 \qquad \mathbf{b}_v^\top \mathsf{Q}_\infty^* \mathbf{c}_v = 0 \tag{4.11}$$

qui contraignent linéairement les 10 éléments de la matrice Q_∞^*.

4.3.1.2 Résolution avec contrainte de signature $(3,0)$ *a posteriori*

Le problème est résolu en deux étapes. Dans la première étape, une solution $\hat{\mathsf{Q}}_\infty^*$, linéaire au sens des moindres carrés, est recherchée en omettant la contrainte de signature $(3,0)$. Chacune des matrices de projection apporte un nombre de quatre équations ; comme la matrice Q_∞^* comporte 10 coefficients, inconnus à un facteur multiplicatif près, un minimum de 3 caméras est requis afin d'obtenir une solution unique. Dans la seconde étape, la contrainte portant sur la signature de Q_∞^* est imposée *a posteriori* à partir de la résolution du problème suivant

$$\arg\min_{\mathsf{H}} \|\hat{\mathsf{Q}}_\infty^* - \mathsf{H}\operatorname{diag}(1,1,1,0)\mathsf{H}^\top\|_F^2 \tag{4.12}$$

Ce problème a une infinité de solutions ; une d'elles découle de la décomposition en valeurs singulières $\mathsf{U}\boldsymbol{\Sigma}\mathsf{V}^\top = \hat{\mathsf{Q}}_\infty^*$ à partir des trois plus grandes valeurs singulières $\sigma_j = \boldsymbol{\Sigma}_{jj}$, $j = 1..3$,

soit

$$\hat{\mathsf{H}} = \mathsf{U}\mathrm{diag}(\sigma_1^{1/2}, \sigma_2^{1/2}, \sigma_3^{1/2}, 1)$$

Néanmoins, l'omission des contraintes sur la signature de Q_∞^* dans la formulation du problème dans la première étape soulève un certain nombre de singularités liées à des mouvements critiques étudiés dans [Gurdjos 2009].

4.3.2 Formation linéaire étendue intégrant une paire de points cycliques

Les données $\mathcal{D}'$. Nous supposons disposer, en plus de la donnée $\mathcal{D} = \{\mathsf{P}_v\}_{v=1..V}$ décrivant une reconstruction projective de V caméras, d'une donnée supplémentaire sous la forme d'un ensemble $\{\tilde{\mathbf{I}}_w^\pm\}_{w\in W}$, où $W \subseteq \{1, \ldots, V\}$, représentant les images, dans certaines vues, d'une paire de points cycliques associée à une même famille de plans parallèles. Nous utiliserons la notation $\mathcal{D}' = \mathcal{D} \cup \{\tilde{\mathbf{I}}_w^\pm\}$ pour désigner ces données. Les figures 4.1(a)(c) illustrent deux exemples de ces données. L'extension à plusieurs paires est relativement directe et ne pose pas de problème théorique.

Le problème. Le problème a le même énoncé que celui du problème 28 en remplaçant les données $\mathcal{D}$ par les données $\mathcal{D}'$.

4.3.2.1 Points cycliques et contraintes d'autocalibrage

Il s'agit ici d'étudier les *contraintes d'autocalibrage* apportées par les images $\tilde{\mathbf{I}}_w^\pm$ d'une paire de points cycliques, en plus des équations de base (4.10) et (4.11) qui restent bien sûr vérifiées. Dans la suite, afin d'alléger les notations, l'indice w est omis. Comme dans le chapitre précédent, nous noterons $\tilde{\mathbf{I}}^\pm$ les images des points cycliques connues à un facteur d'échelle près, c.-à-d. les images *exactes* (cf. équation (3.32)) multipliées par un scalaire complexe arbitraire (inconnu). Si nous cherchons à étudier les contraintes sur $\boldsymbol{\omega}^*$ pour en dériver celles sur Q_∞^*, une idée simple est de partir de l'équation de base du « calibrage plan » (3.7)

$$(\tilde{\mathbf{I}}^\pm)^\top \boldsymbol{\omega} \tilde{\mathbf{I}}^\pm = 0 \tag{4.13}$$

[Hartley 2004b, p. 211] [Sturm 1999, Zhang 1999] où $\boldsymbol{\omega} \sim (\boldsymbol{\omega}^*)^{-1}$ représente l'image de la conique absolue Ω_∞ , et de considérer sa forme duale

$$(\tilde{\mathbf{T}}^\pm)^\top \boldsymbol{\omega}^* \tilde{\mathbf{T}}^\pm = 0 \tag{4.14}$$

où $\tilde{\mathbf{T}}^\pm$ représente la droite tangente à $\boldsymbol{\omega}$ en $\tilde{\mathbf{I}}^\pm$, définie par la relation pôle-polaire [Hartley 2004b, p. 58]

$$\tilde{\mathbf{T}}^\pm = \boldsymbol{\omega} \tilde{\mathbf{I}}^\pm.$$

Deux remarques qui limitent l'intérêt de raisonner dans le plan des pixels peuvent être faites.

1. $\tilde{\mathbf{T}}^\pm$ dépend de $\boldsymbol{\omega}$ qui est inconnue.

2. Si $\tilde{\mathbf{T}}^{\pm}$ était connu, il est clair qu'annuler les parties réelles et imaginaires de l'équation complexe (4.14) fournirait deux équations linéaires réelles par rapport aux éléments de $\boldsymbol{\omega}^*$ qui seraient *dépendantes* puisque $\boldsymbol{\omega}^*$ dépend du seul paramètre f.

À l'évidence, puisque l'inconnue est Q_∞^* liée à $\boldsymbol{\omega}^*$ par l'équation (4.8), il est nécessaire d'interpréter l'équation (4.14) dans l'espace 3D. Dans ce sens, nous nous intéressons à la paire de droites absolues $\mathsf{L}(\tilde{\mathbf{I}}^{\pm})$ issues de la rétroprojection des images $\tilde{\mathbf{I}}^{\pm}$ des points cycliques. Par construction, ces droites sont des droites conjuguées complexes qui intersectent la conique absolue (illustrées sur la figure 4.2). Il est à noter que la connaissance de ces droites est acquise à partir des données $\mathcal{D}'$ puisqu'un moyen simple de les représenter est d'utiliser une matrice de Plücker, qui est définie par la matrice antisymétrique

$$\mathsf{L}(\tilde{\mathbf{I}}^{\pm}) = \mathsf{P}^{\top}\left[\tilde{\mathbf{I}}^{\pm}\right]_{\times}\mathsf{P}.$$

Notre contribution est la mise en place des fondements théoriques permettant de répondre à la question suivante : *« existe-t-il de nouvelles équations linéaires indépendantes sur Q_∞^* amenées par une paire de droites absolues $\mathsf{L}(\tilde{\mathbf{I}}^{\pm})$?»*.

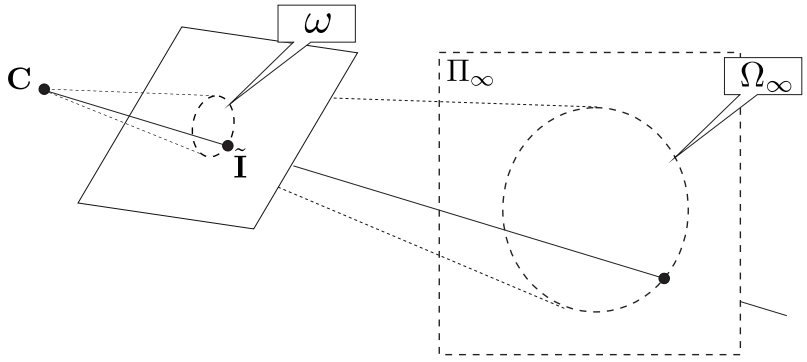


FIGURE 4.2: La droite passant par le centre optique $\mathbf{C}$ et l'image $\tilde{\mathbf{I}}$ d'un point cyclique intersecte la conique absolue Ω_∞.

4.3.2.2 Les équations du problème intégrant une paire de points cycliques

La proposition suivante met en évidence les contraintes linéaires sur Q_∞^* apportées par la droite $\mathsf{L}(\tilde{\mathbf{I}})$ (l'exposant « $\pm$ » est omis dans ce qui suit, $\tilde{\mathbf{I}}$ faisant référence indifféremment à $\tilde{\mathbf{I}}^{+}$ ou $\tilde{\mathbf{I}}^{-}$).

Proposition 29 *Une droite L intersecte la conique absolue Ω_∞ si et seulement si, par L il passe deux plans distincts de $\mathcal{P}_n^*(\mathbb{C})$, représentés par $\boldsymbol{\Pi}_0$ et $\boldsymbol{\Pi}_1$, vérifiant :*

$$\boldsymbol{\Pi}_0^{\top}\mathsf{Q}_\infty^*\boldsymbol{\Pi}_0 = 0, \tag{4.15}$$

$$\boldsymbol{\Pi}_0^{\top}\mathsf{Q}_\infty^*\boldsymbol{\Pi}_1 = 0. \tag{4.16}$$

D'un point de vue géométrique, l'équation (4.15) signifie qu'un des deux plans doit être tangent à Ω_∞, c.-à-d. un plan (complexe) de Q_∞^*, et l'équation (4.16) signifie que l'autre plan (complexe ou réel) doit être conjugué au premier par rapport à Q_∞^*.

Preuve. Soient $\boldsymbol{\Pi}_0$ et $\boldsymbol{\Pi}_1$ deux plans arbitraires distincts passant par L. L'ensemble des plans passant par L peut être représenté par les combinaisons linéaires

$$\boldsymbol{\Pi}(u) = \boldsymbol{\Pi}_0 + u\boldsymbol{\Pi}_1,$$

où le scalaire $u \in \mathbb{C} \cup \{\infty\}$ est une variable avec la convention $\boldsymbol{\Pi}(\infty) = \boldsymbol{\Pi}_1$. Par L il passe en général deux plans tangents à Ω_∞, c.-à-d. deux plans de Q_∞^*, dont les vecteurs s'écrivent sous la forme $\boldsymbol{\Pi}(u_j)_{j=1,2}$, où les scalaires u_1 et u_2 désignent les solutions pour u de l'équation quadratique

$$\boldsymbol{\Pi}(u)^\top \mathsf{Q}_\infty^* \boldsymbol{\Pi}(u) = 0. \tag{4.17}$$

La droite L intersecte Ω_∞ *si et seulement si* les deux plans tangents du cas général coïncident, c.-à-d. *si et seulement si* l'équation (4.17) a une solution double $u_1 = u_2$. Une condition nécessaire et suffisante pour l'existence d'une solution double est que le discriminant du membre de gauche de (4.17) soit nul, ce qui s'exprime par

$$(\boldsymbol{\Pi}_0^\top \mathsf{Q}_\infty^* \boldsymbol{\Pi}_1)^2 - (\boldsymbol{\Pi}_0^\top \mathsf{Q}_\infty^* \boldsymbol{\Pi}_0)(\boldsymbol{\Pi}_1^\top \mathsf{Q}_\infty^* \boldsymbol{\Pi}_1) = 0. \tag{4.18}$$

($\Rightarrow$) Supposons que L intersecte Ω_∞ ; l'équation (4.16) est vérifiée pour toute paire ($\boldsymbol{\Pi}_0$,$\boldsymbol{\Pi}_1$) de plans passant par L. Elle est donc vérifiée, en particulier si $\boldsymbol{\Pi}_0$ est le (double) plan tangent à Ω_∞ (dont le paramètre est $u_1 = u_2 = 0$). Dans (4.18), le fait que $\boldsymbol{\Pi}_0^\top \mathsf{Q}_\infty^* \boldsymbol{\Pi}_0 = 0$ implique que $\boldsymbol{\Pi}_0^\top \mathsf{Q}_\infty^* \boldsymbol{\Pi}_1 = 0$. ($\Leftarrow$) Soient deux plans distincts $\boldsymbol{\Pi}_0$ et $\boldsymbol{\Pi}_1$ passant par L et vérifiant (4.15,4.16). L'equation (4.18) étant alors aussi vérifiée, il s'ensuit que L est une droite qui intersecte Ω_∞. ■

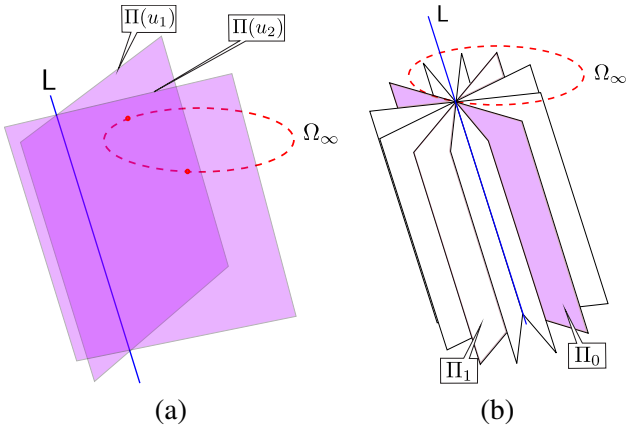


FIGURE 4.3: Preuve de la proposition 29. (a) Par une droite L, il passe deux plans $\Pi(u_1)$ et $\Pi(u_2)$ (combinaisons linéaires de Π_0 et Π_1), qui sont tangents à la conique absolue Ω_∞. (b) Lorsque L intersecte Ω_∞, $u_1 = u_2$ et $\Pi(u_1)$ et $\Pi(u_2)$ sont confondus.

La proposition 29 nous apprend la chose suivante : en intersectant la conique absolue, $\mathsf{L}(\tilde{\mathbf{I}})$ apporte les deux contraintes complexes (4.15,4.16) sur Q_∞^*, c.-à-d. quatre équations linéaires réelles obtenues en annulant les parties réelle et imaginaire des contraintes, à condition de pouvoir déterminer deux plans $\boldsymbol{\Pi}_0$ et $\boldsymbol{\Pi}_1$ passant par $\mathsf{L}(\tilde{\mathbf{I}})$ et conformes à la proposition 29.

Proposition 30 *Sous l'hypothèse ($\mathcal{H}'$), le plan de Q_∞^* passant par $\mathsf{L}(\tilde{\mathbf{I}})$, noté $\boldsymbol{\Pi}_0$ et vérifiant (4.15), est le plan complexe dont le vecteur*

$$\boldsymbol{\Pi}_0 = \mathsf{P}^\top \begin{bmatrix} \psi_1\psi_3 \\ \psi_2\psi_3 \\ -(\psi_1^2+\psi_2^2) \end{bmatrix} \tag{4.19}$$

ne dépend que de $\tilde{\mathbf{I}} = (\psi_1, \psi_2, \psi_3)^\top$.

Preuve. Le plan de Q_∞^* passant par $\mathsf{L}(\tilde{\mathbf{I}})$ est le plan tangent à Ω_∞ au point cyclique ayant pour image $\tilde{\mathbf{I}}$. Il peut être obtenu en rétroprojetant la droite $\mathbf{T}$ du plan des pixels, tangente à l'image $\boldsymbol{\omega}$ de Ω_∞ en $\tilde{\mathbf{I}}$, et est de la forme $\boldsymbol{\Pi}_0 = \mathsf{P}^\top\mathbf{T}$. Cela se vérifie facilement en substituant $\mathsf{P}\mathsf{Q}_\infty^*\mathsf{P}^\top$ par $\boldsymbol{\omega}^*$ dans (4.14). Nous avons mentionné le fait que $\mathbf{T}$ dépend de l'image $\boldsymbol{\omega}$ de la conique absolue, et que $\boldsymbol{\omega}$ est apparemment inconnue. Néanmoins, il existe une façon de contourner cette difficulté. D'une part, sous les hypothèses ($\mathcal{H}'$), le point à l'infini de la droite $\mathbf{T}$ tangente à $\boldsymbol{\omega}$ en $\tilde{\mathbf{I}}$ est donné par $\mathbf{p}_\infty = \mathbf{L}^\infty \times \mathbf{T}$, où $\mathbf{L}^\infty = (0,0,1)^\top$ représente la droite à l'infini du plan image. Il est ensuite aisé de vérifier que $\mathbf{p}_\infty \sim [\mathbf{L}^\infty]_\times \tilde{\mathbf{I}}$. En effet,

$$\begin{aligned}
\mathbf{p}_\infty &= [\mathbf{L}^\infty]_\times \mathbf{T} \\
&= [\mathbf{L}^\infty]_\times \boldsymbol{\omega}\tilde{\mathbf{I}} \\
&= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} (f)^2 & 0 & 0 \\ 0 & (f)^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{I}} \\
&\sim \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tilde{\mathbf{I}} \\
&= [\mathbf{L}^\infty]_\times \tilde{\mathbf{I}}.
\end{aligned}$$

D'autre part, puisque $\mathbf{T}$ contient les points $\mathbf{p}_\infty$ et $\tilde{\mathbf{I}}$, nous pouvons définir

$$\mathbf{T} = \left([\mathbf{L}^\infty]_\times \tilde{\mathbf{I}}\right) \times \tilde{\mathbf{I}} = \begin{pmatrix} \psi_1\psi_3 \\ \psi_2\psi_3 \\ -(\psi_1^2+\psi_2^2) \end{pmatrix}.$$

Enfin, le fait que $\boldsymbol{\Pi}_0 = \mathsf{P}^\top\mathbf{T}$ termine la preuve.

Un cas dégénéré se produit lorsque $\tilde{\mathbf{I}} = (1, \pm i, 0)^\top$, c.-à-d. lorsque le plan 3D du support est parallèle au plan image. ■

La proposition 29 nous renseigne également sur le fait qu'il existe une contrainte indépendante additionnelle sur Q_∞^* sous la forme de l'équation (4.16) où $\boldsymbol{\Pi}_1$ est un plan contenant $\mathsf{L}(\tilde{\mathbf{I}})$ et conjugué à $\boldsymbol{\Pi}_0 = \mathsf{P}^\top\mathbf{T}$ par rapport à Q_∞^*, c.-à-d. contenant la paire de points cycliques 3D.

Proposition 31 *Par* $\mathsf{L}(\tilde{\mathbf{I}})$*, il passe un plan réel* $\boldsymbol{\Pi}_1 \notin \mathsf{Q}^*_\infty$*, défini par*

$$\boldsymbol{\Pi}_1 = \mathsf{P}^\top(\tilde{\mathbf{I}}^+ \times \tilde{\mathbf{I}}^-). \tag{4.20}$$

qui est conjugué au plan $\boldsymbol{\Pi}_1$ *de la proposition 30 par rapport à* Q^*_∞*. Ce plan correspond à la rétroprojection de la ligne de fuite du plan contenant la paire de points cycliques.*

Preuve. Si le plan $\boldsymbol{\Pi}_0$ de la proposition 30 est un plan de Q^*_∞, alors son conjugué $\bar{\boldsymbol{\Pi}}_0$ par rapport à Q^*_∞ est aussi un plan de Q^*_∞. Si $\boldsymbol{\Pi}_1$ est un plan réel qui est conjugué à $\boldsymbol{\Pi}_0$ par rapport à Q^*_∞ alors il est aussi conjugué à $\bar{\boldsymbol{\Pi}}_0$. Le plan $\boldsymbol{\Pi}_1$ contient alors nécessairement les pôles de $\boldsymbol{\Pi}_0$ et $\bar{\boldsymbol{\Pi}}_0$ par rapport à Q^*_∞, qui sont les deux points cycliques conjugués formant la paire associée a une famille de plans parallèles. Le plan $\boldsymbol{\Pi}_1$ contient aussi nécessairement les images de ces deux points cycliques, notés $\tilde{\mathbf{I}}^+$ et $\tilde{\mathbf{I}}^-$ par cohérence avec les notations précédentes, et coïncide donc avec la rétroprojection de la droite réelle contenant ces images, c.-à-d. de la ligne de fuite de la famille de plans. Le plan $\boldsymbol{\Pi}_1$ passe par le centre de la caméra et la ligne de fuite, il est donc réel. ■

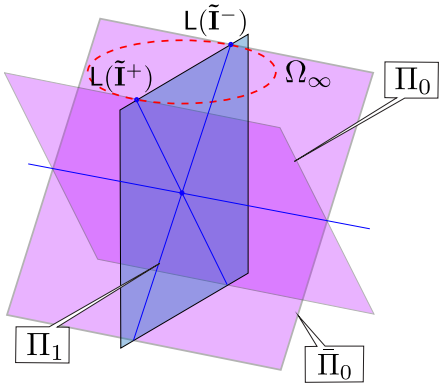


FIGURE 4.4: Interprétation géométrique des équations pour l'autocalibrage intégrant les points cycliques. Voir le texte pour plus de détails.

Au total, nous venons de montrer que, dans le cas général, les quatre équations apportées par les contraintes (4.15,4.16) sont linéairement indépendantes respectivement à Q^*_∞.

4.3.3 Équations de base revisitées

Nous pouvons légitimement nous demander s'il existe un lien entre les contraintes intégrant des points cycliques (4.15,4.16) et les contraintes « de base » (4.10,4.11) de l'autocalibrage. Afin d'établir ce lien, notre idée est de revisiter les équations (4.10,4.11) en les complexifiant dans un premier temps afin d'obtenir les équations algébriques équivalentes (où les indices sont omis)

$$(\mathbf{a} \pm i\mathbf{b})^\top \mathsf{Q}^*_\infty (\mathbf{a} \pm i\mathbf{b}) = 0 \tag{4.21}$$

$$\mathbf{c}^\top \mathsf{Q}^*_\infty (\mathbf{a}^j \pm i\mathbf{b}) = 0 \tag{4.22}$$

où $i^2 = -1$. Les équations obtenues (4.21,4.22) peuvent ainsi être, dans un second temps, réécrites de la manière suivante :

$$(\mathsf{P}^\top(1, \pm i, 0)^\top)^\top \mathsf{Q}^*_\infty (\mathsf{P}^\top(1, \pm i, 0)^\top) = 0 \tag{4.23}$$

$$(\mathsf{P}^\top(1, \pm i, 0)^\top)^\top \mathsf{Q}^*_\infty (\mathsf{P}^\top(0, 0, 1)^\top) = 0 \tag{4.24}$$

On voit alors que, dans le cas où $\tilde{\mathbf{I}}^\pm \sim (1, \pm i, 0)^\top$, la tangente à $\boldsymbol{\omega}$ en $\tilde{\mathbf{I}}^\pm$ dans le plan des pixels s'écrit

$$\tilde{\mathbf{T}}^\pm = \boldsymbol{\omega}\tilde{\mathbf{I}}^\pm \sim (1, \pm i, 0)^\top,$$

et la ligne de fuite s'écrit

$$\tilde{\mathbf{I}}^+ \times \tilde{\mathbf{I}}^- = (0, 0, 1)^\top.$$

Les équation de base (4.23,4.24) ainsi réécrites

$$(\mathsf{P}^\top \tilde{\mathbf{T}}^\pm)^\top \mathsf{Q}^*_\infty (\mathsf{P}^\top \tilde{\mathbf{T}}^\pm) = 0 \tag{4.25}$$

$$(\mathsf{P}^\top \tilde{\mathbf{T}}^\pm)^\top \mathsf{Q}^*_\infty (\mathsf{P}^\top(0, 0, 1)^\top) = 0 \tag{4.26}$$

sont des équations intégrant des points cycliques (4.15,4.16) puisque les plans $\boldsymbol{\Pi} = \mathsf{P}^\top \tilde{\mathbf{T}}^\pm$ et $\mathbf{q} = \mathsf{P}^\top(\tilde{\mathbf{I}}^+ \times \tilde{\mathbf{I}}^-)$ (qui est le plan $\mathbf{c}$ dont le vecteur est la troisième ligne de P dans l'équation (4.6)) sont deux plans qui vérifient la proposition 29 et se calculent selon les propositions 30 (cas dégénéré) et 31. La différence est que, dans les équations de base, le vecteur $\tilde{\mathbf{I}}^\pm$ ne représente pas l'image d'une paire de points cycliques *mais* la paire de points cycliques du plan projectif des pixels et le vecteur $(\tilde{\mathbf{I}}^+ \times \tilde{\mathbf{I}}^-)$ ne représente pas la ligne de *fuite* d'un plan mais la droite à l'infini du plan projectif des pixels, ceci à la condition que la représentation du plan projectif des pixels soit euclidienne, ce qui est le cas sous l'hypothèse ($\mathcal{H}$').

Les équations de base d'autocalibrage peuvent à présent être interprétées comme suit. Sous l'hypothèse ($\mathcal{H}$) ou son équivalent ($\mathcal{H}'$) , elles expriment le fait que les droites absolues, obtenues par rétroprojection des points cycliques du plan projectif des pixels intersectent la conique absolue. Ceci correspond exactement au même paradigme que celui des approches proposées par Ponce *et al.* [Ponce 2005, Ronda 2008] qui ont introduit le complexe quadratique (de droites) absolu pour l'autocalibrage de caméra, excepté que ce dernier peut également être utilisé dans le cas d'une caméra aux pixels carrés et de point principal inconnu. Le problème se formule naturellement dans $\mathcal{P}_5(\mathbb{R})$, c.-à-d. l'espace des droites aux coordonnées de Plücker, mais requiert des algorithmes d'optimisation sophistiqués.

4.3.3.1 Algorithme d'autocalibrage unifié

Nous proposons ici un algorithme unifié permettant de résoudre le problème 28 de l'autocalibrage dual. Il s'agit de chercher une matrice $\hat{\mathsf{Q}}^*_\infty$ solution des équations (4.15) et (4.16). Le terme *unifié* signifie que nous considérons à la fois les équations reposant sur les paires de points cycliques des plans des pixels et sur les images des points cycliques d'un plan. Lors de la résolution du problème, la contrainte de signature $(3, 0)$ est omise et sera traitée a posteriori, comme dans

[Pollefeys 1999], à partir de la solution $\hat{\mathsf{Q}}^*_\infty$ obtenue, tel que décrit par l'équation (4.12).

Les équations (4.15) et (4.16) reposant sur l'ensemble des images $\tilde{\mathbf{I}}^\pm_w$ d'une paire de points cycliques et sur l'ensemble des paires de points cycliques $(1, \pm i, 0)^\top$ des plans des pixels sont rassemblées pour former un système linéaire homogène, d'inconnues les 10 éléments de la matrice symétrique Q^*_∞. L'algorithme complet est décrit dans la table 4.1.

Solution minimale. Pour un ensemble de $\{\mathsf{P}_v\}_{v=1..V}$ de V matrices de projection et un ensemble $\{\tilde{\mathbf{I}}^\pm_w\}_{w\in W}$, où $W \subseteq \{1, \ldots, V\}$, de cardinal $|W|$, les équations (4.15) et (4.16) peuvent se réécrire sous la forme

$$\mathsf{D}\mathbf{X} = \mathbf{0}_{4(V+|W|)} \tag{4.27}$$

où le vecteur

$$\mathbf{X} = \mathrm{vech}(\mathsf{Q}^*_\infty)$$

a pour composantes les inconnues, à savoir les 10 éléments de Q^*_∞, où vech représente l'opérateur de semi-vectorisation, défini dans le chapitre 2, associé à la matrice de duplication $\mathcal{P}_{16}$ de dimension 16×10 (cf. équation 2.2), et où

$$\mathsf{D} = \begin{bmatrix} \mathsf{D}^c_1 \\ \vdots \\ \mathsf{D}^c_V \\ \mathsf{D}^p_1 \\ \vdots \\ \mathsf{D}^p_{|W|} \end{bmatrix} \mathcal{P}_{16}$$

où D^c_v est la matrice (de dimension 4×10) des données associées aux contraintes apportées par la paire de points cycliques du plan image v telle que

$$\mathsf{D}^c_v = \begin{bmatrix} (\mathbf{a}_v{}^\top \otimes \mathbf{a}_v{}^\top) - (\mathbf{b}_v{}^\top \otimes \mathbf{b}_v{}^\top) \\ \mathbf{b}_v{}^\top \otimes \mathbf{a}_v{}^\top \\ \mathbf{c}_v{}^\top \otimes \mathbf{a}_v{}^\top \\ \mathbf{c}_v{}^\top \otimes \mathbf{b}_v{}^\top \end{bmatrix}$$

où $\mathbf{a}_v{}^\top$, $\mathbf{b}_v{}^\top$ et $\mathbf{c}_v{}^\top$ représentent les 3 vecteurs lignes de la matrice de projection P_v (cf. équation (4.6)), et où D^p_w est la matrice (de dimension 4×10) des données associées aux contraintes apportées par les images de la paire de points cycliques du plan visible dans la vue w telle que

$$\mathsf{D}^p_w = \begin{bmatrix} (\boldsymbol{\pi}_{1,w}{}^\top \otimes \boldsymbol{\pi}_{1,w}{}^\top) - (\boldsymbol{\pi}_{2,w}{}^\top \otimes \boldsymbol{\pi}_{2,w}{}^\top) \\ \boldsymbol{\pi}_{2,w}{}^\top \otimes \boldsymbol{\pi}_{1,w}{}^\top \\ (\boldsymbol{\Pi}_{1,w}{}^\top \otimes \boldsymbol{\pi}_{1,w}{}^\top) \\ (\boldsymbol{\Pi}_{1,w}{}^\top \otimes \boldsymbol{\pi}_{2,w}{}^\top) \end{bmatrix}$$

où les vecteurs $\boldsymbol{\pi}_{1,w}$ et $\boldsymbol{\pi}_{2,w}$ correspondent respectivement aux parties réelle et imaginaire de $\boldsymbol{\Pi}_{0,w}$ tel que décrit par l'équation (4.19) et où $\boldsymbol{\Pi}_{1,w}$ est le vecteur tel que décrit par l'équation (4.20), tous deux associés à la matrice de projection P_w et aux vecteurs $\tilde{\mathbf{I}}_w^{\pm}$.

Le système obtenu comporte $4(V+|W|)$ équations linéaires indépendantes pour 10 inconnues. La paire de points cycliques d'un plan image apporte 4 équations ce qui est également le cas pour les images de la paire de points cycliques d'un plan dans une vue. Ainsi, une solution exacte existe si $4(V+|W|)=10$, c.-à-d. $V+|W|=\frac{5}{2}$. En présence de bruit, une solution au sens des moindres carrés existe si l'on dispose d'un nombre minimal de 2 vues, plus les images d'une paire de points cycliques dans une vue.

PROBLÈME :
Calculer Q_∞^* à partir de l'ensemble de matrices de projection $\{\mathsf{P}_v\}_{v=1..V}$ et de l'ensemble des images des points cycliques $\{\mathbf{I}_w^{\pm}\}_{w\in W}$ où $W \subseteq \{1,\ldots,V\}$.

ALGORITHME :

1. **Pour** toutes les vues v de 1 à V,

   / équations de base */*

   (a) Calculer $\boldsymbol{\Pi}_0^c \leftarrow \mathsf{P}_v^\top(1,i,0)^\top$ et $\boldsymbol{\Pi}_1^c \leftarrow \mathsf{P}_v^\top(0,0,1)^\top$

   (b) Ajouter la ligne $\mathsf{D}_v^c = \mathrm{blocDonnees}(\boldsymbol{\Pi}_0^c, \boldsymbol{\Pi}_1^c)$ à la matrice de données D

   (c) **Si** les points cycliques du plan sont visibles dans la vue **alors**

   / équations proposées */*

   i. Calculer $\tilde{\mathbf{I}}_v^+ = (\psi_1,\psi_2,\psi_3)^\top$ et son conjugué $\tilde{\mathbf{I}}_v^-$.

   ii. Calculer $\boldsymbol{\Pi}_0^p$ à partir de (4.19) et $\boldsymbol{\Pi}_1^p$ à partir de (4.20) et de $\tilde{\mathbf{I}}_v^+$.

   iii. Ajouter la ligne $\mathsf{D}_v^p = \mathrm{blocDonnees}(\boldsymbol{\Pi}_0^p, \boldsymbol{\Pi}_1^p)$ à la matrice de données D.

   Fin Si

   Fin Pour

2. Résoudre $\min_{\mathbf{X}} \|\mathsf{D}\mathbf{X}\|^2$ tel que $\|\mathbf{X}\|^2 = 1$.
3. Construire $\hat{\mathsf{Q}}_\infty^*$ à partir de la solution $\hat{\mathbf{X}}$.

FONCTION $\mathsf{D} = \mathrm{blocDonnees}(\boldsymbol{\Pi}_0, \boldsymbol{\Pi}_1)$:

$$\boldsymbol{\pi}_1 \leftarrow \Re(\boldsymbol{\Pi}_0)$$
$$\boldsymbol{\pi}_2 \leftarrow \Im(\boldsymbol{\Pi}_0)$$
$$\mathsf{D} \leftarrow \begin{bmatrix} (\boldsymbol{\pi}_1{}^\top \otimes \boldsymbol{\pi}_1{}^\top) - (\boldsymbol{\pi}_2{}^\top \otimes \boldsymbol{\pi}_2{}^\top) \\ \boldsymbol{\pi}_2{}^\top \otimes \boldsymbol{\pi}_1{}^\top \\ \boldsymbol{\Pi}_1{}^\top \otimes \boldsymbol{\pi}_1{}^\top \\ \boldsymbol{\Pi}_1{}^\top \otimes \boldsymbol{\pi}_2{}^\top \end{bmatrix} \mathcal{P}_{16}$$

TABLE 4.1: Algorithme d'autocalibrage unifié basé sur les images des points cycliques.

4.4 Résultats expérimentaux

4.4.1 Résultats sur données synthétiques

Les performances de la méthode proposée sont évaluées en analysant le gain en précision sur la reconstruction euclidienne finale, exprimé via l'erreur tridimensionnelle REQM (cf. équation (2.3)) calculée à partir d'un nuage de points « témoin » . Plusieurs paradigmes du calcul de la structure et du mouvement ont été évalués. Ces derniers sont mentionnés de façon générique par la notation PF$[n_1]$+SC$[n_2]$ qui signifie que nous effectuons la factorisation projective en incluant les images de n_1 paires de points cycliques suivi par l'autocalibrage incluant les contraintes fournies par les images de n_2 paires de points cycliques, avec $n_j = 0, 1$. L'idée était d'étudier le gain de précision issu des contraintes fournies par les images des points cycliques pour l'algorithme de factorisation présenté dans la section 3.7 et pour l'autocalibrage. Les expériences ont été conduites en faisant varier le nombre de vues, le nombre de points naturels, le bruit appliqué sur les mesures et la quantité de données manquantes (c.-à-d. le nombre de vues sur lesquelles les images des points cycliques du plan additionel ne sont pas visibles).

Nous considérons une scène constituée de N (≤ 20) points naturels et de 0 ou 1 paire de points cycliques. Les points cycliques sont calculés comme étant les points projectifs communs à deux cercles concentriques. Les points naturels sont distribués aléatoirement dans la sphère unité (de diamètre $d = 2$ unités) centrée à l'origine. Chacun des cercles est centré en $[\pm 0.2, \pm 0.2, z]$ (où z est pris aléatoirement dans $[-0.2, 0.2]$), de rayon $0.05d$ et $0.1d$ respectivement, et la normale du plan de support est aléatoirement prise dans le cône d'axe $[0, 0, 1]$ et d'angle $\pi/3$. Les caméras sont grossièrement orientées vers l'origine, placées aléatoirement à une distance prise dans $[2.85, 3.15]$ de l'origine. La résolution des images est de 512×512 pixels et les distances focales sont aléatoirement prises dans $[850, 1150]$. Le point principal est supposé coïncider (grossièrement) avec l'origine mais ses coordonnées sont aléatoirement prises dans $[-15, 15]^2$. Un bruit gaussien, centré en zero et d'écart type σ a été ajouté aux points image, y compris ceux à partir desquels les images des points cycliques sont calculées.

Les expériences suivantes comparent l'erreur tridimensionnelle REQM obtenue par les différentes séquences d'algorithmes (figurant dans les légendes de la figure 4.5) suivant plusieurs variables. Sur la figure 4.5(a), 6 vues ont été considérées et nous avons fait varier le niveau de bruit appliqué aux coordonnées des points image. Dans l'expérience suivante, les coordonnées des points image ont été affectées par un bruit d'écart type $\sigma = 1$ pixel. Sur la figure 4.5(b), nous avons fait varier le nombre de vues. Sur la figure 4.5(c), 6 vues sont considérées et le nombre de points naturels varie de 6 à 20. Sur la figure 4.5(d), nous considérons 6 vues tandis que le nombre de vues dans lesquelles les points cycliques sont visibles varie. Le résultat clé de ces expériences est que, comparée à l'approche « classique » du calcul de la structure et du mouvement associé à la séquence PF[0]+SC[0], la précision est améliorée que ce soit via PF[1]+SC[0] ou PF[0]+SC[1], les meilleurs résultats restant ceux fournis par PF[1]+SC[1]. Cela signifie que les algorithmes proposés pour la factorisation projective et l'autocalibrage contribuent de manière indépendante à l'amélioration des performances même si, de manière générale, le gain apporté par les contraintes basées sur les images des points cycliques pour l'autocalibrage est plus significatif que pour la

factorisation projective, particulièrement lorsque le nombre de vues diminue et que le bruit augmente. Lorsque le nombre de points naturels est très limité, l'ajout des images des points cycliques dans la factorisation projective réduit significativement l'erreur tridimensionnelle REQM et dans ce cas, les améliorations fournies par PF[1]+SC[1] proviennent principalement de la factorisation proposée. L'ensemble de ces observations peut être justifié en considérant le nombre de contraintes fournies par les images des points cycliques par rapport aux dimensions des systèmes d'équations impliqués respectivement dans la factorisation projective et dans l'autocalibrage.

Pour finir, la table 4.2 présente les erreurs médianes sur les distances focales calculées à partir de la méthode proposée en faisant varier le nombre de vues et en considérant 20 points naturels dont les images sont affectées par un bruit d'écart type $\sigma = 1$ pixel. La distance focale f_v de chaque caméra projective P_v a été calculée comme solution du problème aux moindres carrés linéaires :

$$\mathrm{diag}((f_v)^2, (f_v)^2, 1) \sim \mathsf{P}_v \mathsf{Q}^*_\infty \mathsf{P}_v^\top . \tag{4.28}$$

Nb. de vues	3	4	5	6	8	10	12
Erreur (%)	16.0	10.1	8.8	7.5	6.8	6.1	5.1

TABLE 4.2: Médianes des erreurs sur la distance focale en faisant varier le nombre de vues.

4.4.2 Résultats sur données réelles

La méthode proposée a été testée sur des ensembles d'images extraites de quatre séquences vidéo. Nous rappelons que la table 4.3 fournit les informations relatives à chacune des expérimentations. Les points image sélectionnés sont extraits et suivis via l'algorithme itératif KLT [Lucas 1985]. Les sous-ensembles de points respectant les contraintes épipolaires sont extraits via l'utilisation de l'algorithme RANSAC [Fischler 1981]. À l'issue du calcul des matrices de projection euclidiennes, des reconstructions denses ont été calculées via l'utilisation du logiciel PMVS (décrit dans [Furukawa 2010]) afin d'évaluer visuellement les performances des différents paradigmes.

Ringo. Dans cette première expérience, 12 points d'une figurine sont mis en correspondance dans 6 vues (illustrées sur la figure 4.6(a)). Les images des points cycliques sont calculées à partir des cercles concentriques situés sur un marqueur plan (présenté dans la deuxième partie de la thèse). La reconstruction dense basée sur les caméras euclidiennes calculées avec PF[1]+SC[0] est incohérente tandis que la méthode proposée utilisant les images des points cycliques lors de l'autocalibrage fonctionne correctement, fournissant les résultats présentés sur la figure 4.6(b).

Voiture. Dans cette expérience, 7 points d'une voiture sont mis en correspondance dans 8 vues (illustrées sur la figure 4.6(c)). Les points cycliques utilisés sont ici les points d'intersection (complexes) des deux cercles coplanaires formés par les roues. Les reconstructions denses basées sur

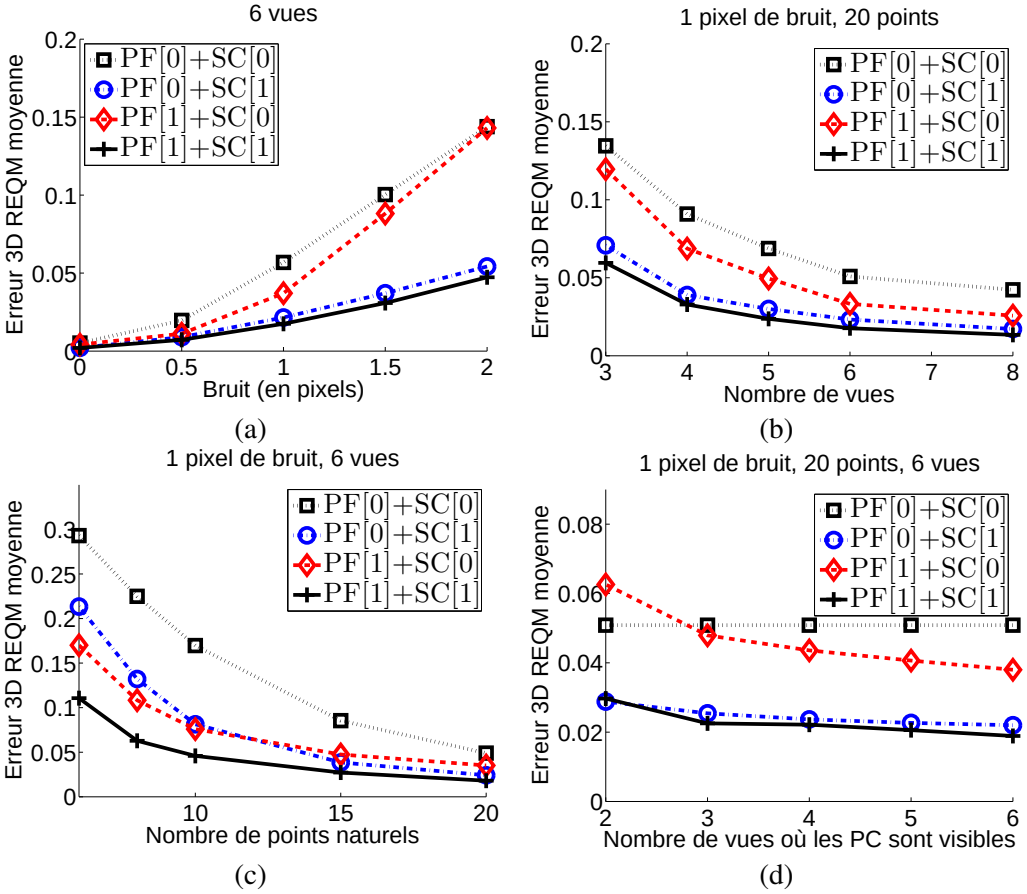


FIGURE 4.5: (a) Erreur suivant l'écart type du bruit appliqué sur les points mesurés. (b) Erreur suivant le nombre de vues utilisées. (c) Erreur suivant le nombre de points naturels considérés. (d) Erreur suivant le nombre de vues pour lesquelles les images des points cycliques sont visibles. Les résultats sont exprimés avec et sans l'introduction des images des points cycliques dans le schéma de factorisation. Le terme PC signifie points cycliques.

Exprérience	*Ringo*	*Voiture*	*Visage*	*Dragon*
Caméra	Iphone 4S (back)	Nikon J1	Iphone 4S (front)	Nikon J1
Résolution	1920×1080	1920×1080	640×480	1920×1080
Nb. de vues	6	8	5	3
Nb. de points	12	7	11	54
IPC manquantes	non	non	non	oui
Marqueur cyclique	cercles concentriques	cercles coplanaires	cercles parallèles	grille
AF euclidien	non	non	oui	oui

TABLE 4.3: Informations relatives aux quatres expériences effectuées sur des séquences vidéo réelles. Le terme ICP fait référence aux images des points cycliques. AF est l'abréviation d'ajustement de faisceaux et les *points* font référence aux points naturels.

les caméras euclidiennes calculées avec PF[0]+SC[0] et PF[1]+SC[0] sont incohérentes. La reconstruction fournie par PF[1]+SC[0] impliquant les images des points cycliques dans la factorisation projective présente cependant un léger gain de performance comparée à celle fournie par PF[0]+SC[0] seulement basée sur les images des points naturels. PF[1]+SC[1] fonctionne correctement, fournissant les résultats présentés sur la figure 4.6(d). Nous avons utilisé le résultat de la méthode proposée comme reconstruction euclidienne initiale dans une procédure incrémentale du calcul de la structure et du mouvement pour laquelle les résultats sont illustrés sur la figure 4.7.

Visage. Dans cette expérience, 11 points d'un visage sont mis en correspondance dans 5 vues, illustrées sur la figure 4.1(c) (de plus amples détails sont fournis dans la légende). La reconstruction dense basée sur les caméras euclidiennes calculées avec PF[1]+SC[0] n'a fourni aucun résultat cohérent tandis que PF[1]+SC[1] fonctionne correctement, fournissant les résultats illustrés sur la figure 4.1(d).

Dragon. Dans la dernière expérience, 54 points d'un dragon sont mis en correspondance dans 3 vues, illustrées sur la figure 4.1(a). Les homographies liant le plan de support d'une grille au plan image fournissent les images des points cycliques dans les vues 1 et 3. Seule la méthode proposée, basée sur la même reconstruction projective, fournit des résultats corrects via l'utilisation des images des points cycliques dans l'étape d'autocalibrage (illustrés sur la figure 4.1(b)). De plus amples détails sont fournis dans la légende de la figure 4.1. Nous avons pu constater que réaliser l'étape d'autocalibrage sans les images de points cycliques additionels pouvait échouer lorsque le nombre de points mis en correspondance était faible (cf. figure 4.1(d), figure 4.6(b) et figure 4.6(d)). Comme nous pouvons ici le constater, celle-ci peut également échouer lorsqu'un grand nombre de points est mis en correspondance dans un nombre de vues limité. Ces observations confirment en partie les résultats obtenus en simultation.

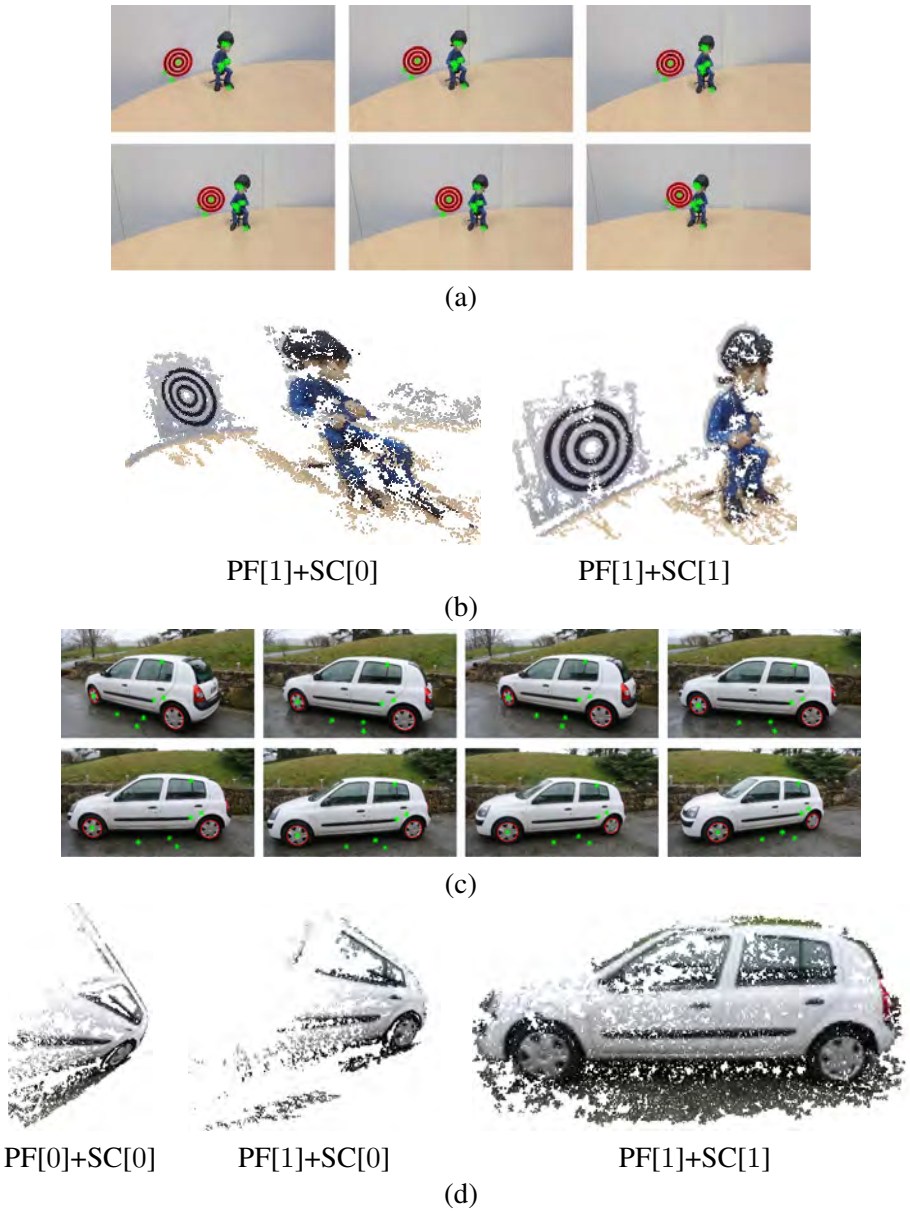


FIGURE 4.6: ***Ringo.*** (a) Vues d'entrée. (b) Reconstructions denses effectuées à partir des caméras euclidiennes calculées par PF[1]+SC[0] à gauche et par PF[1]+SC[1] à droite. ***Voiture.*** (c) Vues d'entrée. (d) Reconstructions denses effectuées à partir des caméras euclidiennes calculées par PF[0]+SC[0], PF[1]+SC[0] et PF[1]+SC[1] de gauche à droite. Dans (a) et (c), les images des points naturels utilisées sont inscrites en vert et les primitives image utilisées pour le calcul des images des points cycliques sont inscrites en rouge. Voir le texte pour plus de détails.

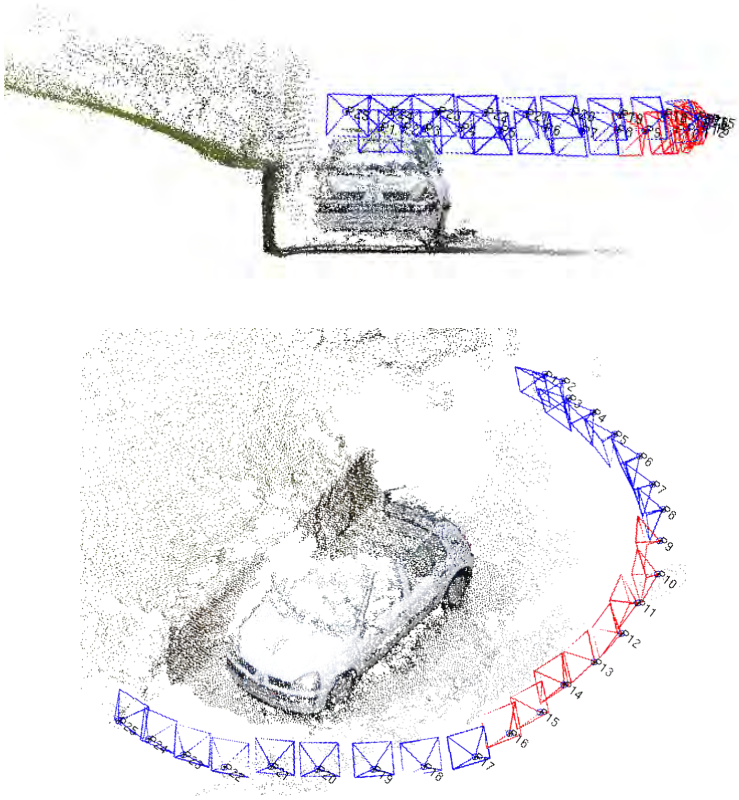


FIGURE 4.7: ***Voiture.*** Deux vues d'une reconstruction dense basée sur l'intégralité de la séquence ***Voiture*** pour laquelle un algorithme incrémental de calcul de la structure et du mouvement a été effectué à partir de la donnée initiale des caméras (en rouge) et de la structure euclidiennes obtenues à l'issue du calcul PF[1]+SC[1] basé sur les images présentées sur la figure 4.6(c). La reconstruction est ensuite optimisée via un ajustement de faisceaux.

4.5 Conclusion

Dans ce chapitre, la formulation linéaire de [Pollefeys 1999] pour l'autocalibrage opérant dans l'espace projectif dual a été rappelée. Nous avons fourni des équations additionelles indépendantes reposant sur les images des paires de points cycliques présentes dans la scène. D'un point de vue géométrique, nous avons montré que l'ensemble de ces équations constituaient des paires de droites absolues, rétroprojections des images des points cycliques. Un algorithme d'autocalibrage unifié, reposant sur l'ensemble de ces équations, a été proposé.

Dans de futurs travaux, il serait intéressant, d'une part, de déterminer les orientations des plans de support des points cycliques critiques au regard de la formulation proposée et, d'autre part, d'évaluer l'incidence de ces orientations sur les performances de la méthode proposée. Ceci pourrait permettre de contrôler les performances de l'algorithme d'autocalibrage via l'introduction de marqueurs plans dans la scène. Une telle approche pourrait ainsi offrir un bon compromis entre flexibilité et performances pour les méthodes de calcul de la structure et du mouvement.

Chapitre 5

Le système de marqueurs C^{2}Tags

Dans ce chapitre, nous présentons le système de marqueurs C^{2}Tags, marqueurs plans composés de couronnes circulaires concentriques. Nous fournissons tout d'abord un état de l'art sur les systèmes de marqueurs. Les propriétés ayant motivé la conception de notre système de marqueurs sont ensuite exposées. Enfin, nous proposons une description détaillée des algorithmes de détection et d'identification de l'image d'un C^{2}Tag.

5.1 Introduction

Définition 32 *Un marqueur est une entité artificielle, disposée dans la scène, associée à un ensemble de caractéristiques géométriques et/ou photométriques mesurables dans toute vue et destinées à encoder*

1. *une information géométrique de référence pour le calcul de la structure et du mouvement,*
2. *une information sémantique, en particulier identitaire.*

Nous appellerons « *primitives visuelles* » de telles caractéristiques et « *motif* » l'ensemble de celles-ci, c.-à-d. l'aspect visuel du marqueur. Par « *système de marqueurs* », nous faisons référence à un système constitué, d'une part, d'un ensemble de marqueurs et, d'autre part, d'une *ensemble d'algorithmes de traitement*, allant de la détection des images des marqueurs au décodage de l'information attachée. Idéalement, le système doit disposer d'une collection de marqueurs tous différents dont les images sont identifiables. Nous attribuons une dimension à un marqueur qui est la dimension du plus petit sous-espace projectif qui contient l'information géométrique de référence. Nous distinguons différents *types de marqueurs* : les marqueurs 0D, 1D, 2D et 3D correspondent à des primitives respectivement ponctuelle, linéaire, plane et tridimensionnelle.

Un des critères de performance d'un système de marqueurs est sa capacité à détecter le motif d'un marqueur dans une vue puis à identifier le marqueur à partir de celui-ci, par exemple en mettant en correspondance une rectification métrique du motif détecté avec le motif géométrique d'un marqueur de la collection. Ce critère est évalué par un certain nombre de métriques ou d'observations qualitatives. Nous listons ici une première partie de ces critères, considérés lors de la conception de notre système de marqueurs, dont la majorité sont couramment employés dans un tel contexte [Fiala 2010] :

1. le taux de *faux positifs*,
2. le taux de *confusion inter-marqueurs*,
3. le taux de *faux négatifs*,
4. la *taille minimale des marqueurs*,
5. la *taille de la bibliothèque des marqueurs*,
6. la *robustesse aux conditions d'éclairage*,
7. la *robustesse au calibrage photométrique*,
8. la *vitesse d'exécution*.

Le taux de *faux positifs* correspond au taux de détection de motifs dans une vue qui ne correspondent pas à des images de marqueurs. Le taux de *confusion inter-marqueurs* représente le taux de mauvaises mises en correspondances entre motifs détectés dans une vue et marqueurs, ce qui se produit lorsqu'un marqueur est confondu avec un autre. Le taux de *faux négatifs* correspond au taux de non détections. La *taille minimale des marqueurs* représente la taille requise (en pixels) dans l'image pour assurer une détection fiable. La *taille de la collection* est le nombre de marqueurs différents que le système peut supporter. La *robustesse aux conditions d'éclairage*

représente la capacité du système à détecter et identifier les marqueurs en présence de mauvaises conditions d'éclairage. La *robustesse au calibrage photométrique* fait référence aux propriétés optiques, inhérentes à l'utilisation conjointe du matériau constituant les marqueurs et de la caméra, nécessaires à la mesure. Enfin, le dernier critère concerne le temps de calcul lié à la complexité de l'algorithme de détection. Dans le cadre d'applications de réalité augmentée ou encore de la réalisation d'effets spéciaux, le système doit pouvoir fonctionner en temps réel ou dans de très courts délais relativement à la durée de la séquence vidéo à traiter.

En plus des critères précédents, nous souhaitons à présent insister sur trois autres critères de performance d'un système de marqueurs, ayant essentiellement guidé la solution proposée, à savoir :

9. la ***robustesse aux occultations***,
10. la ***robustesse au flou de bougé***,
11. l'***encodage de contraintes euclidiennes***.

La *robustesse aux occultations* et la *robustesse au flou de bougé* représentent la capacité du système à détecter et identifier les marqueurs lorsque leurs images sont occultées ou en présence de flou de bougé (dû en général à un mouvement rapide de la caméra). L'*encodage de contraintes euclidiennes* signifie que le marqueur doit encoder une information capable de fournir des contraintes euclidiennes sur la scène. Dans le cas d'un marqueur plan, cela doit permettre le calcul de la structure euclidienne du plan de support du marqueur (cf. §3.4.2).

Dans ce chapitre, nous souhaitons pouvoir proposer une solution au problème du calcul de la structure et du mouvement à travers la conception d'un système de marqueurs cycliques dont les avantages pour la résolution de ce problème ont été présentés dans le chapitre 3 de ce manuscrit.

Organisation du chapitre. Nous proposons dans ce qui suit un état de l'art sur les systèmes de marqueurs, suivi d'une description du problème spécifique abordé dans ce chapitre. Nous introduisons ensuite les éléments théoriques sur lesquels repose l'algorithme de détection de notre système de marqueurs. Une description détaillée des algorithmes de détection et d'identification des images de C^2Tags est ensuite fournie.

5.2 État de l'art

Dans cette partie, nous présentons un état de l'art des systèmes de marqueurs de type 0D et 2D, marqueurs les plus présents dans la littérature. Celui-ci est partiellement inspiré de celui fourni par Lepetit et Fua [Lepetit 2005] qui dressent une liste exhaustive des systèmes de marqueurs. Nous fournissons ensuite un bref état de l'art sur les codes-barres, en lien avec notre système de marqueurs.

5.2.1 Marqueurs 0D

Un marqueur 0D correspond à un point artificiel. Son motif est ainsi utilisé afin de disposer précisément dans une vue de l'information géométrique ponctuelle. Il est conçu de telle sorte qu'il

soit facilement détectable et identifiable par les algorithmes de traitement.

Les marqueurs présentés dans ce paragraphe sont tous associés à des motifs circulaires et seul leur centre constitue l'information géométrique de référence. Ce type de marqueurs a été utilisé pendant de nombreuses années en photogrammétrie rapprochée notamment car l'information ponctuelle peut être mesurée avec une précision bien plus élevée que pour des images de primitives naturelles. La plupart de ces systèmes de marqueurs utilisés dans l'industrie reposent sur des marqueurs circulaires composés de matériaux réfléchissants couplés à des caméras équipées de flashs. Les photographies sont exposées de telle sorte que l'arrière-plan disparaisse et que les marqueurs puissent être détectés automatiquement.

Des solutions à bas coût et de plus faible précision ont également été proposées. Nous pouvons par exemple mentionner les marqueurs appelés *Concentric Contrasting Circle* [Hoff 1996] constitués d'un anneau noir placé sur un fond blanc, ou *vice versa*. L'algorithme de détection consiste à seuiller l'image puis à appliquer un certain nombre d'opérations morphologiques afin d'identifier les régions noires et blanches des images des motifs. [State 1996] utilisent des marqueurs à codes de couleurs pour une identification plus fiable. Chaque marqueur est composé d'un point intérieur entouré d'un anneau. Quatre couleurs différentes sont utilisées et ainsi, le système dispose de douze marqueurs uniques. [Cho 1998] ont introduit un système de marqueurs multi-échelles afin d'augmenter leur *champ d'utilisation*. Ce dernier est défini par les distances minimale et maximale entre un marqueur et la caméra entre lesquelles le marqueur est détectable. Les marqueurs sont composés d'anneaux colorés d'épaisseurs proportionnelles à leurs rayons et dont le nombre varie selon la taille du marqueur. Ainsi, lorsque la caméra est proche des marqueurs, seuls les marqueurs de faibles dimensions sont détectés et, lorsque celle-ci est éloignée, ce sont les marqueurs de grandes dimensions qui sont détectés.

Les méthodes d'extraction précédentes nécessitent de définir des seuils pour la segmentation des images, seuils qui doivent généralement être ajustés aux conditions d'éclairage. Dans [Naimark 2002], un seuillage est appliqué non pas à l'image elle-même mais au gradient du logarithme de l'image, ce qui donne un algorithme de détection robuste aux éclairages hétérogènes. Dans le but d'augmenter le nombre de marqueurs différents, les auteurs ajoutent des « anneaux de données » entre un anneau intérieur et un anneau extérieur composant le marqueur. Ces anneaux supplémentaires sont composés de régions noires ou blanches pouvant être utilisées comme un code-barres afin d'encoder l'identifiant du marqueur. Une telle conception permet d'obtenir une capacité d'identification de 3×2^{15} marqueurs.

Tandis que les méthodes de détection précédentes sont fondées sur des algorithmes de détection *ad hoc*, [Claus 2004] utilisent une méthode d'apprentissage présentant d'importantes améliorations en termes de fiabilité. Les marqueurs sont constitués de disques noirs sur fond blanc. Des échantillons d'images de marqueurs sont collectés sous différents points de vue, en faisant varier l'échelle ainsi que les conditions d'éclairage. Une cascade de classifieurs est ensuite entraînée sur ces données : la première étape consiste en une classification rapide, basée sur une règle de décision de Bayes ; la deuxième, plus coûteuse mais plus robuste, est une classification au plus proche voisin. À l'exécution, toutes les sous-fenêtres de l'image sont classées par cette cascade de classifieurs. L'ensemble fournit une méthode de détection remarquablement fiable.

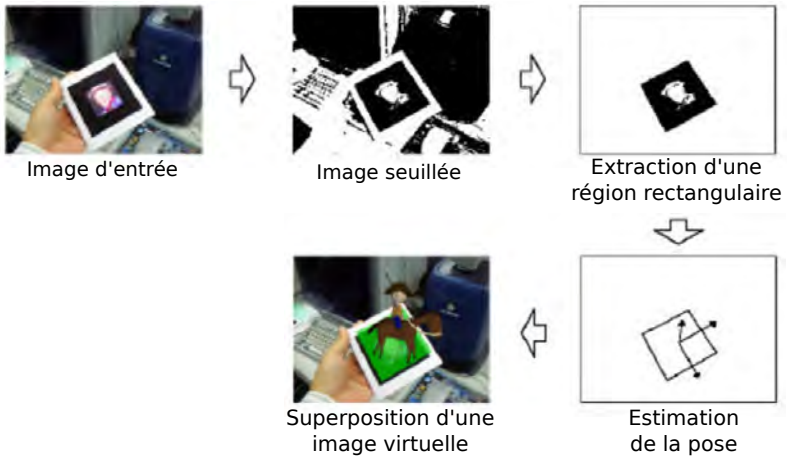


FIGURE 5.1: Chaîne de traitement de l'ARToolkit.

5.2.2 Marqueurs 2D

Les marqueurs 2D correspondent à des plans artificiels. Leur motif est situé sur un plan de support physique et se compose, en général, d'un nombre suffisant de primitives visuelles nécessaire au calcul de l'homographie représentant la restriction de la projection centrale au plan de support. Un calcul de pose, fonction de cette homographie (éventuellement l'homographie elle-même), constitue l'information géométrique de référence.

[Koller 1997] ont introduit des marqueurs à motifs carrés noirs, sur fond blanc, disposant de carrés rouges de faibles dimensions utilisés pour leur identification. Leurs coins sont estimés par l'ajustement de segments de droites aux points de gradient maximal sur les bordures du marqueur. Ils constituent un ensemble de quatre correspondances. La pose du marqueur est ensuite estimée en utilisant des filtres de Kalman étendus. [Rekimoto 1998, Kato 1999, Kato 2000] utilisent des marqueurs plans rectangulaires et montrent qu'un seul marqueur est suffisant pour le calcul de la pose de son plan de support. Leur approche a conduit à une solution de suivi 3D en temps réel, robuste et peu coûteuse, et à une implantation logicielle disponible publiquement, nommée ARToolKit, principale raison de sa popularité.

Les marqueurs du système ARToolKit ont une bordure noire sur fond blanc, comme la plupart des marqueurs précédemment décrits, qui facilite leur détection. L'identification est effectuée à partir du motif situé à l'intérieur de cette bordure. L'image est dans un premier temps seuillée et l'algorithme de détection localise les régions de pixels noirs connexes. Les régions dont les contours peuvent être approchés par quatre segments de droite sont conservées. Chacune de ces régions est ensuite rectifiée afin de supprimer la déformation liée à la perspective puis est comparée, par une mise en correspondance de modèles, aux motifs connus. La rectification repose sur l'utilisation de l'homographie calculée à partir de la mise en correspondance du marqueur. Le traitement complet, à savoir la détection des marqueurs et l'estimation de la pose, est réalisé à une cadence de 30 images par seconde et peut ainsi être exécuté en temps réel. Par ailleurs, sous de bonnes conditions d'éclairage, la pose estimée est suffisamment précise pour des applications de réalité augmentée. Le système requérant peu de ressources CPU, des applications basées sur

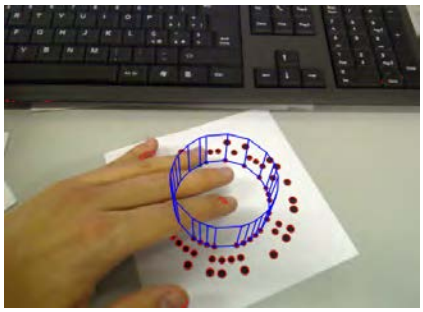


FIGURE 5.2: Exemple d'un marqueur RUNE-Tag. Le marqueur est détecté même en présence d'occultations. Une fois la pose de la caméra calculée, l'image est « augmentée ».

l'utilisation de ce type de marqueur sont implantées sur des appareils mobiles tels que les *smartphones*.
Le système de marqueurs ARTag proposé dans [Fiala 2005] utilise le même type de marqueur. Il a été conçu dans le but, d'une part, de répondre aux faiblesses du système ARToolkit, à savoir un taux de faux positifs et un taux de confusion inter-marqueurs élevés, et, d'autre part, de fonctionner sous des conditions d'éclairage dégradées et en présence d'occultations.

[Bergamasco 2011] proposent un système de marqueurs, nommés RUNE-Tags, constitués de disques noirs disposés sur des régions circulaires concentriques sur fond blanc (cf. figure 5.2). L'algorithme de détection effectue dans un premier temps une détection des ellipses présentes dans l'image (via OpenCV [OCV]). La localisation de l'image du marqueur repose ensuite sur un vote consistant à reprojeter le motif dont l'image est calculée à partir de chaque paire d'ellipses détectées. L'algorithme présente l'avantage d'avoir une faible complexité ($O(n^2)$ où n est le nombre d'ellipses détectées) et ne requiert pas de rectification de l'image. Le système présente de bonnes performances en termes de robustesse et de précision pour le calcul de la pose, et ce même en présence d'occultations.

5.2.3 Les codes-barres

L'encodage d'information sur des surfaces planes pour une détection par des techniques optiques a longtemps été effectué sous la forme de codes-barres standard, le plus souvent lus par des scanners lasers. Les codes-barres bidimensionnels ont été développés afin d'être lus par une caméra passive plutôt qu'un laser actif. Des exemples de codes-barres sont illustrés sur la figure 5.3. Ces motifs ont initialement été conçus pour porter de l'information sémantique et non de l'information géométrique. Le motif est typiquement bitonal et chacun de ses pixels représente une décision binaire pour l'algorithme de lecture. Les raisons pour lesquelles de tels dispositifs ne sont pas utilisés pour la conception de systèmes de marqueurs sont les suivantes : leur champ d'utilisation est trop restreint ; les déformations liées à la perspective les rendent difficilement détectables ; enfin, ils ne fournissent pas suffisamment de mises en correspondance pour le calcul de la pose. Cependant, le principe de bitonalité et les méthodes numériques sous-jacentes peuvent être utilisés dans la conception de systèmes de marqueurs.



FIGURE 5.3: Un code-barres standard unidimensionnel et des systèmes de codes-barres bidimensionnels. Les codes-barres standard et de type PDF417 sont prévus pour une lecture par scanner laser. Les codes-barres 2D de type Data Matrix, Maxicode et QR sont conçus pour porter de l'information et sont lus en étant positionnés face à la caméra. Ils sont mal adaptés à une utilisation dans des systèmes de marqueurs.

5.3 Le problème et ses motivations

Dans le chapitre 3, nous avons introduit la notion de « marqueur cyclique ». Nous rappelons qu'un marqueur cyclique est un entité géométrique décrite par un motif plan qui encode (sans ambiguïté) l'information donnant la « structure euclidienne » du plan de support, à savoir l'image de la paire de points cycliques de ce plan. L'utilisation du terme de « structure euclidienne » du plan de support se justifie par le fait que la paire de points cycliques du plan projectif est l'équivalent, en dimension deux, de la conique absolue de l'espace projectif tridimensionnel et qu'à partir de ceux-ci il est possible de « rectifier » l'image du motif du marqueur par une homographie du plan des pixels de façon à obtenir le motif original à une similitude plane près. Il est également nécessaire de pouvoir encoder une certaine information sémantique de telle sorte que les images des marqueurs puissent être identifier par les algorithmes de traitement.

Notre objectif est de concevoir un système de marqueurs cycliques répondant au mieux à l'ensemble des critères mentionnés dans l'introduction de ce chapitre. L'accent est mis sur la simplicité d'un encodage de la structure euclidienne puisque nous rappelons que celle-ci constitue la donnée d'entrée essentielle au calcul de la structure et du mouvement présenté dans la première partie de ce manuscrit. Nous allons voir qu'il est possible de répondre à cette exigence tout en proposant une solution de marqueurs présentant de bonnes propriétés photométriques nécessaires à une détection et une identification fiables.

Motivation 1 : encoder « naturellement » la structure euclidienne du plan de support. Il est usuel [Semple 1952, p.32] de définir la paire de points cycliques d'un plan comme le lieu d'intersection de *tout* cercle du plan projectif complexifié avec sa droite à l'infini. Du fait qu'elle est commune à tous les cercles, cette paire est invariante pas les similitudes planes. Sachant que la restriction de la projection centrale à un plan est une homographie et que le groupe des homographies préserve l'incidence, il s'ensuit que l'image de la paire de points cycliques peut être déterminée à partir de l'image de la droite à l'infini (appelée ligne de fuite) et de l'image d'un cercle du plan.

La seule donnée de l'image d'un unique cercle est donc insuffisante pour encoder la structure euclidienne du plan de support *sauf* si l'on dispose de la donnée supplémentaire de l'image du centre du cercle. En effet, par définition [Semple 1952, p. 119], le centre d'une conique est le

point du plan dont la droite polaire par rapport au cercle est la droite à l'infini. Par la préservation de l'incidence, la ligne de fuite du plan est la droite polaire de l'image du centre du cercle par rapport à l'image du cercle. Au total, nous pouvons mettre en évidence le fait que le *couple de primitives composé d'un cercle et de son centre* encode « naturellement » la structure euclidienne du plan de support (comme illustré dans la figure 5.4).

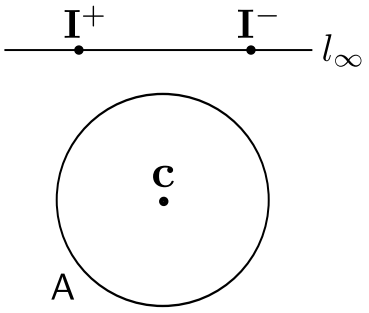


FIGURE 5.4: Les points cycliques $\mathbf{I}^+$ et $\mathbf{I}^-$ sont deux points imaginaires issus de l'intersection de tout cercle A du plan avec la droite à l'infini l_∞. La droite à l'infini l_∞ est la droite polaire du centre du cercle (désigné ici par **c**) par rapport au cercle.

Motivation 2 : être détectable en présence de fortes occultations. L'image d'un cercle, à savoir une ellipse sous les hypothèses de chiralité, constitue par nature une primitive dont la détection est robuste aux occultations [Kanatani 2004]. La détection simultanée de l'image du centre du cercle (ou, de façon équivalente, de la ligne de fuite du plan) ajoute une difficulté non négligeable. Il est possible d'approcher l'image du centre du cercle par le centre de l'image du cercle mais ces derniers ne sont confondus que sous l'hypothèse d'un modèle de caméra affine. Sous l'hypothèse du modèle sténopé, une idée simple est de matérialiser le centre du cercle comme le lieu d'intersection de plusieurs droites sécantes, comme dans [Ying 2007, Ying 2008]. La détection de l'image d'une paire (ou d'une famille) de cercles concentriques est une façon de contourner le problème initial en recherchant l'image du centre (commun) mais sans que le centre n'ait besoin d'être matérialisé dans le motif. Ce nouveau problème peut être résolu de façon élégante, comme par exemple dans [Jiang 2005]. La solution proposée par ces auteurs détecte l'image d'un couple de cercles concentriques en construisant une séquence de points convergant vers l'image du centre (commun). Cependant, un tel algorithme de détection n'est pas robuste aux occultations des images des cercles. La méthode proposée dans [Ying 2007] est, à notre connaissance, la seule méthode de détection robuste aux occultations. La méthode proposée consiste en une généralisation de l'idée proposée dans [Yuen 1989] pour la détection de l'image du centre, fondée sur la construction de droites passant par celui-ci. Cependant, cette méthode présente un certain nombre d'inconvénients : (*i*) la méthode ne garantit pas, en général, une précision suffisante, comme mentionné dans [Kanatani 2004] pour qualifier la plupart des extensions de la transformée de Hough ; (*ii*) en présence de fortes occultations, les droites construites sont en nombre limité (illustré sur la figure 5.5) ce qui rend difficile la localisation de l'image du centre ; (*iii*) la méthode s'appuie sur des segments de pixels de contour connexes calculés comme présenté dans [Kanatani 2004],

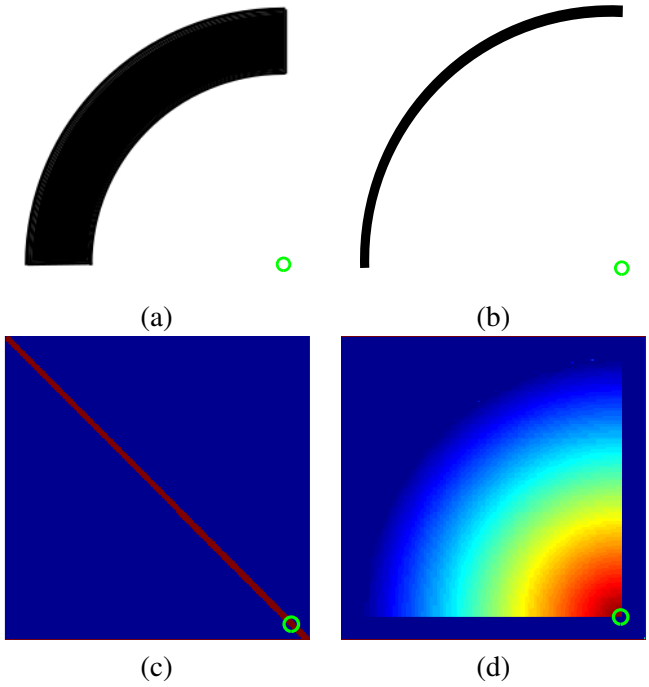


FIGURE 5.5: Comparaison des résultats du vote pour l'image du centre reposant sur une couronne circulaire (a) dans [Ying 2007] et sur le motif $\mathcal{M}$ (b) pour la méthode proposée, occultés aux trois-quarts. L'image du centre des motifs circulaires est mise en évidence par un cercle vert sur chacune des figures. La méthode de détection de l'image du centre proposée dans [Ying 2007] et celle proposée dans ce chapitre reposent sur un vote dans l'espace de paramètres 2D défini par l'ensemble des pixels de l'image. (c) Résultat du vote pour l'image du centre fourni par la méthode décrite dans [Ying 2007] associé au motif (a). (d) Résultat du vote fourni par la méthode proposée associé au motif (b). (c) La méthode proposée par [Ying 2007] présente un résultat ambigu car les droites accumulées dans l'espace de paramètres ne sont pas en nombre suffisant (à cause d'une trop forte occultation) pour fournir une solution unique. La méthode proposée est, quant à elle, mieux posée et présente une solution unique visible sur la figure (d).

méthode non robuste aux ruptures des segments de contour.

Dans ce chapitre, nous montrerons que le motif « idéal » est un motif circulaire dont la caractéristique photométrique est un albédo « en dégradé » dont la valeur est fonction de la distance au centre du motif (illustré sur la figure 5.6). Ce motif peut être vu comme l'assemblage infini de cercles concentriques déterminés par les courbes de niveau de l'albédo. Une de nos contributions est de montrer que ce motif est idéal pour au moins deux raisons.

- Le centre de ces cercles est le point de convergence des lignes de champ du gradient de la fonction d'albédo, qui sont les courbes du plan de support dont la tangente en tout point est colinéaire au gradient de cette fonction en ce point.
- L'image du centre est le point de convergence des lignes de champ du gradient de la fonction de niveau de gris associée à une image du motif avec l'idée que, pour parcourir une ligne de champ, il suffit de suivre la direction du gradient de cette fonction. La détection de l'image du centre du cercle peut alors être vue comme une procédure de vote, d'une façon un peu analogue aux approches fondées sur la transformée de Hough pour la détection de cercles [Duda 1972, Ballard 1981].

Motivation 3 : être détectable en présence de bruit et de flou. En pratique, les conditions réelles d'expérimentation ne permettent pas d'utiliser ce motif idéal. Nous le remplaçons par un motif plan, appellé C^{2}Tag, formé d'un ensemble de couronnes circulaires noires centrées sur un fond blanc (illustré sur la figure 5.8), dont les frontières sont des courbes de niveau d'un albédo « noir et blanc ». L'idée est d'insérer un nombre suffisant de couronnes afin de pouvoir construire une discrétisation « raisonnable » des lignes de champ de la fonction de niveau de gris d'une image du motif. Pour détecter l'image du centre, nous appliquons le même principe que pour le motif idéal : il s'agit de parcourir les lignes de champ de la fonction de niveau de gris d'une image, en suivant les gradients de cette fonction, pour détecter l'image du centre. Un des intérêts du C^{2}Tag est d'introduire un fort contraste ce qui garantit une mesure fiable de la direction des gradients.

Motivation 4 : être discriminant et identifiable via le support d'un codage. La substitution du motif $\mathcal{V}$ à un C^{2}Tag permet également, en plus de ses intérêts photométriques pour la mesure du gradient, de renforcer le caractère discriminant du motif à partir du nombre et/ou des épaisseurs relatives des couronnes. D'autre part, la présence de couronnes rend le motif assimilable à un code-barres circulaire, lisible « naturellement » indépendamment de la perspective géométrique [Gurdjos 2006], ce qui lui confère les capacités d'encodage de ce type de dispositif. Ceci est un avantage, par rapport à [Poupyrev 2000, Fiala 2005] où les régions du motif liées à la détection et à l'identification sont disjointes : une large bordure noire est utilisée pour la détection du motif et la région interne est destinée à l'identification. Dans la solution proposée, la surface totale du motif est conjointement utilisée pour les opérations de détection et d'identification, dans le but d'augmenter le champ d'utilisation des marqueurs.

5.4 La solution proposée : le motif C^{2}Tag

Dans cette section, nous introduisons tout d'abord ce qu'aurait pu être un motif idéal vis-à-vis de l'encodage de la structure euclidienne du plan, à savoir les données conjointes de l'image d'un cercle et de l'image de son centre. Nous disons « ce qu'aurait pu » car nous allons voir que ce motif n'est pas exploitable dans des conditions réelles d'expérimentation. Malgré cela, ce motif idéal va nous permettre de justifier l'emploi d'un motif formé de couronnes concentriques.

Soit un point du plan projectif dont le vecteur (homogène) est $\mathbf{v} \sim [x, y, 1]^\top$ dans une réprésentation affine du plan. Nous noterons $\bar{\mathbf{v}} = [x, y]^\top$ le vecteur des coordonnées cartésiennes du point de telle façon que $\mathbf{v} \sim [\bar{\mathbf{v}}^\top, 1]^\top$.

5.4.1 Un marqueur circulaire « idéal »

Soit le motif $\mathcal{M}$ formé d'un disque U centré de rayon unitaire, dont la caractéristique photométrique est un albédo « en dégradé » (du noir vers le blanc), et défini par la fonction suivante :

$$\alpha(\bar{\mathbf{x}}) = \begin{cases} 1 - \|\bar{\mathbf{x}}\|_2 & \text{si } \bar{\mathbf{x}} \in U \subset \mathbb{R}^2 \\ 0 & \text{sinon.} \end{cases} \tag{5.1}$$

Une illustration de ce motif est donnée dans la figure 5.6. Le motif $\mathcal{M}$ peut être vu comme l'assemblage « infini » de cercles concentriques déterminés par les équipotentielles du champ scalaire α, c.-à-d. par les courbes de niveau de l'albédo α. Tous ces cercles sont centrés en l'origine, appelée *centre du motif*.

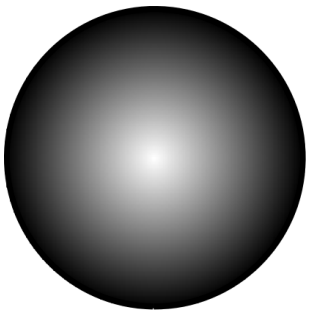


FIGURE 5.6: Motif $\mathcal{M}$

On considère maintenant $\mathcal{I}(\mathcal{M})$, une photographie « virtuelle » du motif $\mathcal{M}$. On se place sous les quatre hypothèses géométriques et photométriques suivantes liées à la formation de la photographie :

(*i*) l'appareil photographique (caméra) obéit au modèle du sténopé, où la « distortion perspective » de la photographie est modélisée par une projection centrale ;

(*ii*) les contraintes de chiralité [Hartley 2004b, p. 515] sont vérifiées, à savoir que la restriction de cette projection au plan du motif est quasi-affine par rapport au motif, c.-à-d. tous les points du motif sont situés « devant » l'appareil ;

(*iii*) la scène est éclairée par un faisceau lumineux parallèle et uniforme ;

(*iv*) la surface du motif est lambertienne.

Sous ces hypothèses, il existe une certaine homographie quasi-affine de matrice H associée à une fonction homographique $\mathbf{g}_\mathsf{H} : U \to \mathbb{R}^2$ définie par

$$\mathbf{g}_\mathsf{H} : \qquad (x, y) \mapsto \left(\frac{H_{11}x + H_{12}y + H_{13}}{H_{31}x + H_{32}y + H_{33}}, \frac{H_{21}x + H_{22}y + H_{23}}{H_{31}x + H_{32}y + H_{33}} \right) \tag{5.2}$$

telle que les niveaux de gris de $\mathcal{I}(\mathcal{M})$ en tout point $\bar{\mathbf{u}} \in \mathbb{R}^2$ du plan des pixels aient pour valeur :

$$\alpha_\mathsf{H}(\bar{\mathbf{u}}) = C\alpha\left(\mathbf{g}_\mathsf{H}^{-1}(\bar{\mathbf{u}})\right) \tag{5.3}$$

où $C > 0$ est une certaine constante.

D'un point de vue photométrique, si les cercles du motif $\mathcal{M}$ sont les équipotentielles du champ scalaire α, leurs images par $\mathbf{g}_\mathsf{H}$ sont les équipotentielles du champ scalaire α_H. D'un point de vue géométrique, l'image par $\mathbf{g}_\mathsf{H}$ d'une famille de cercles concentriques est une famille d'ellipses *non concentriques*.

Calcul de la structure euclidienne du plan. Comme expliqué dans la section 5.3, notre problème consiste à déterminer, dans la photographie, l'image du centre de ces cercles afin de calculer

la structure euclidienne du plan de support du motif. Pour le résoudre, nous nous intéressons au champ de gradient $\nabla\alpha_{\mathsf{H}}$, champ vectoriel dérivé du potentiel α_{H}, et aux courbes du plan de la photographie dont la tangente en tout point $\bar{\mathbf{u}}$ est colinéaire au vecteur $\nabla\alpha_{\mathsf{H}}(\bar{\mathbf{u}})$, c.-à-d. aux *lignes de champ* de α_{H}, comme illustré dans la figure 5.7(c).

Une ligne de champ de α_{H}, passant par $\bar{\mathbf{u}}_0$, est un arc paramétré $\boldsymbol{\phi} : T \subset \mathbb{R} \mapsto \mathbb{R}^2$, $t \mapsto \boldsymbol{\phi}(t)$, solution du système différentiel

$$\begin{cases} \boldsymbol{\phi}'(t) = \nabla\alpha_{\mathsf{H}}(\boldsymbol{\phi}(t)) \\ \boldsymbol{\phi}(t_0) = \bar{\mathbf{u}}_0 \end{cases} \tag{5.4}$$

où $(t_0, \bar{\mathbf{u}}_0) \in T \times \mathbb{R}^2$ définit une condition initiale.

Proposition 33 *Par tout point de la photographie à l'exception de l'image du centre, c.-à-d. par tout point de* $\mathbf{g}_{\mathsf{H}}(U \setminus \{\mathbf{0}\})$*, passe une et une seule ligne de champ de* α_{H}*.*

Preuve. Une ligne de champ est une courbe intégrale de l'équation différentielle (5.4). Sous les hypothèses du théorème de Cauchy-Lipschitz, par un point donné, il passe une courbe intégrale et une seule. ■

Proposition 34 *Si* ℓ *est une ligne de champ de* α_{H} *alors* ℓ *converge vers l'image du centre du motif.*

Preuve. La ligne de champ passant par $\bar{\mathbf{u}} \in \mathbf{g}_{\mathsf{H}}(U)$ est un chemin de $\mathbf{g}_{\mathsf{H}}(U)$ constitué d'une suite de déplacements infinitésimaux

$$\begin{cases} \bar{\mathbf{u}}_{i+1} = \lim_{d_i \to 0_+} \left(\bar{\mathbf{u}}_i + d_i \frac{\nabla\alpha_{\mathsf{H}}(\bar{\mathbf{u}}_i)}{\|\nabla\alpha_{\mathsf{H}}(\bar{\mathbf{u}}_i)\|_2} \right) \\ \bar{\mathbf{u}}_0 = \bar{\mathbf{u}} \end{cases}$$

Sachant que dans le voisinage d'un point $\bar{\mathbf{u}}_i \in \mathbf{g}_{\mathsf{H}}(U)$, la fonction de niveau de gris α_{H} croît le plus fortement dans la direction du gradient $\nabla\alpha_{\mathsf{H}}$, l'image $\bar{\mathbf{c}}$ du centre est le point de convergence de la suite puisque (*i*) $\bar{\mathbf{c}}$ est le point où $\alpha_{\mathsf{H}}(\mathbf{g}_{\mathsf{H}}(U))$ est maximal ; (*ii*) $\bar{\mathbf{c}}$ est l'unique point singulier de $\alpha_{\mathsf{H}}(\mathbf{g}_{\mathsf{H}}(U))$, c.-à-d. l'unique point tel que $\nabla\alpha_{\mathsf{H}}(\bar{\mathbf{c}}) = 0$, ce qui implique que $\alpha_{\mathsf{H}}(\bar{\mathbf{u}}_{i+1}) > \alpha_{\mathsf{H}}(\bar{\mathbf{u}}_i)$, $\forall \bar{\mathbf{u}}_i \neq \bar{\mathbf{c}}$. ■

La proposition 34 est illustrée sur la figure 5.7(c).

La solution à notre problème peut donc être énoncée ainsi, du moins sous forme théorique : à partir de deux lignes de champ de $\nabla\alpha_{\mathsf{H}}$, nous pouvons déterminer l'image du centre du motif $\mathcal{M}$, comme leur (unique) point commun. La direction du gradient $\nabla\alpha_{\mathsf{H}}$ est une quantité mesurable dans une photographie ; notre idée est que, pour parcourir une ligne de champ associée à α_{H}, il suffit de suivre la direction du gradient. Il est donc possible de construire les lignes de champ associées à la photographie $\mathcal{I}(\mathcal{M})$.

En pratique, les conditions réelles d'expérimentation ne permettent pas d'assurer une construction fiable des lignes de champ, notamment à cause de la violation des hypothèses susmentionnées.

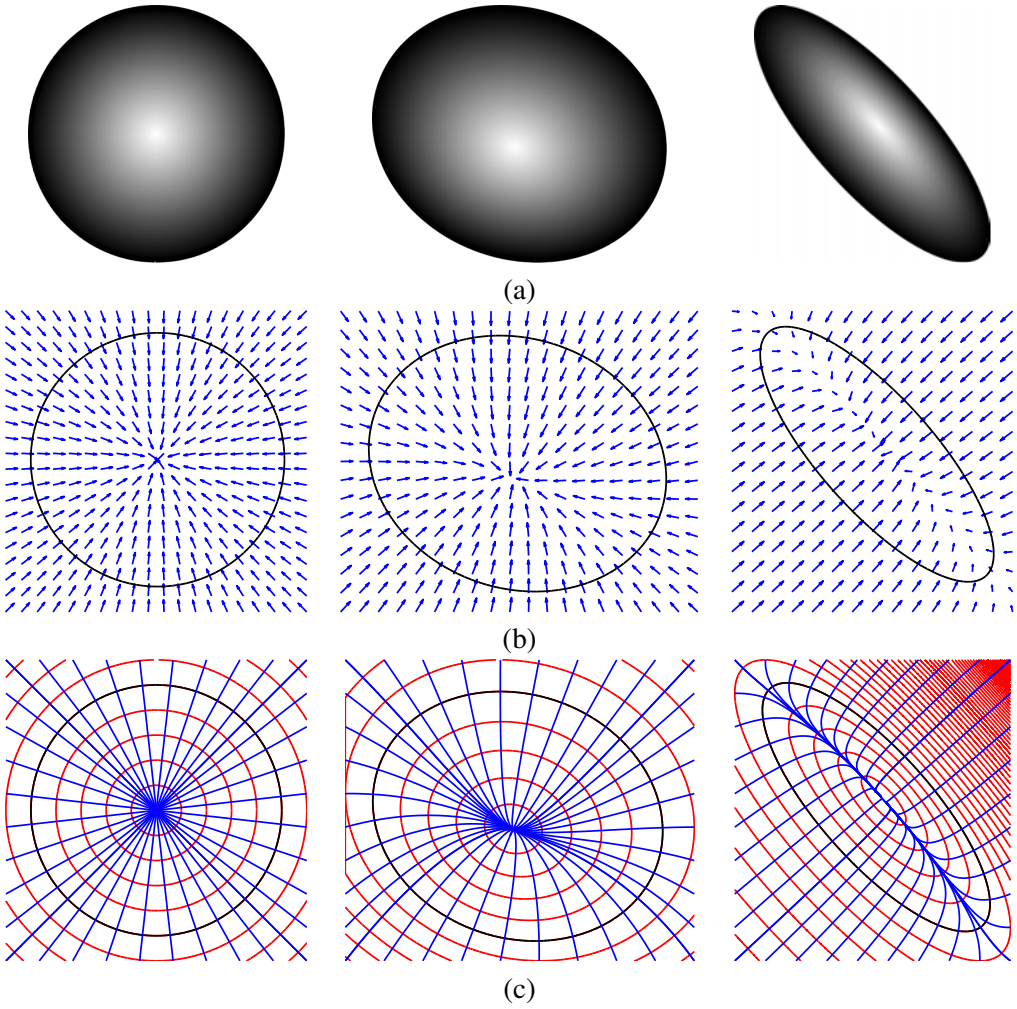


FIGURE 5.7: (a) Trois photographies « virtuelles » $\mathcal{I}(\mathcal{M})$ du motif $\mathcal{M}$, de niveau de gris α_{H} défini par l'équation (5.3). Elles correspondent à des prises de vues où les angles entre le plan du motif et le plan des pixels sont respectivement 0°, 35° et 70°. L'ellipse d'un « noir le plus pur » correspond au contour externe de $\mathcal{I}(\mathcal{M})$. (b) Champ de vecteurs $\nabla\alpha_{\mathrm{H}}$ défini par le gradient de α_{H}. (c) Lignes de champ (en bleu) et équipotentielles (en rouge) associées au champ de vecteurs $\nabla\alpha_{\mathrm{H}}$ (voir texte, propositions 33-34).

En particulier, les faibles variations de niveau de gris rendent la mesure du gradient sensible aux bruits et aux flous (de bougé et de mise au point). De plus, son absence de texture en fait un motif peu discriminant et le prive de moyen d'encoder de l'information. Pour ces raisons, $\mathcal{M}$ est remplacé par un C^{2}Tag (illustré sur la figure 5.8), motif composé d'un ensemble de couronnes dont les frontières circulaires sont les équipotentielles de la fonction d'albédo α du motif idéal $\mathcal{M}$.

5.4.2 Le C^{2}Tag, support des équipotentielles du motif « idéal »

Nous définissons un motif plan que nous appellerons C^{2}Tag : il s'agit d'un ensemble de *couronnes circulaires* noires centrées sur un fond blanc (illustré sur la figure 5.8). Par *couronne circulaire*, nous désignons une région du plan comprise entre deux cercles concentriques. Nous supposerons que le cercle externe a un rayon unitaire.



FIGURE 5.8: Un C^{2}Tag composé de 4 couronnes.

On considère une photographie « virtuelle » du C^{2}Tag en se plaçant sous les mêmes hypothèses géométriques et photométriques que pour le motif idéal. Nous désignons par I_{H} le champ scalaire associé à la fonction de niveau de gris de cette photographie, dépendant de la même homographie H que dans le cas du motif idéal, et par ∇I_{H} le champ de gradients dérivé de I_{H}. Il est à noter que les cercles d'un C^{2}Tag sont des équipotentielles du champ scalaire α et leurs images par H sont des équipotentielles du champ scalaire α_{H}. Il s'ensuit qu'en tout point $\bar{\mathbf{u}} \in \mathbb{R}^2$ les gradients des deux champs vectoriels $\nabla\alpha_{\mathsf{H}}$ et ∇I_{H} sont colinéaires c.-à-d. $\nabla\alpha_{\mathsf{H}}(\bar{\mathbf{u}}) \sim \nabla I_{\mathsf{H}}(\bar{\mathbf{u}})$. Pour déterminer l'image du centre des cercles du motif idéal, nous avons proposé de « suivre » les chemins donnés par les lignes de champ de $\nabla\alpha_{\mathsf{H}}$. L'idée est ici de construire des chemins déduits de ∇I_{H} qui soient des approximations « raisonnables » des lignes de champ de $\nabla\alpha_{\mathsf{H}}$.

Le C^{2}Tag n'utilise qu'un certain nombre de cercles du motif idéal à partir d'un albédo « binaire », c.-à-d. noir et blanc. Ceci permet d'introduire un fort contraste dans la photographie assurant une mesure fiable de la direction des gradients calculés aux points de contour, et d'apporter de la discriminance via le nombre et/ou les épaisseurs relatives des couronnes. D'autre part, la présence de couronnes rend le motif assimilable à un code-barres circulaire, ce qui lui confère les capacités d'encodage de ce type de dispositif.

Approximation des lignes de champ. Une ligne de champ de α_{H} s'écrit analytiquement :

$$\boldsymbol{\phi}(t) = \bar{\mathbf{u}}_0 + \int_{t_0}^{t} \mathbf{dt} = \bar{\mathbf{u}}_0 + \int_{t_0}^{t} \frac{\nabla\alpha_{\mathsf{H}}\left(\boldsymbol{\phi}(t)\right)}{\|\nabla\alpha_{\mathsf{H}}(\boldsymbol{\phi}(t))\|_2} dt \tag{5.5}$$

où l'on rappelle que $\nabla\alpha_{\mathsf{H}}\left(\boldsymbol{\phi}(t)\right)$ est parallèle à $\mathbf{dt}$ et où $dt = \|\mathbf{dt}\|_2$. Deux résultats importants en découlent qui sont les briques de base de notre approche. À partir d'une subdivision régulière de l'arc en n points $\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{n-1}$, l'intégrale 5.5 peut être calculée comme la somme de Riemann suivante

$$\boldsymbol{\phi}(t) = \bar{\mathbf{u}}_0 + \lim_{\Delta r \to 0} \sum_{i=0}^{n-1} \frac{\nabla I_{\mathsf{H}}(\bar{\mathbf{u}}_i)}{\|\nabla I_{\mathsf{H}}(\bar{\mathbf{u}}_i)\|_2} \Delta r \tag{5.6}$$

où $\Delta r = \frac{t-t_0}{n}$ et où $\mathbf{u}_i$ est le point de l'arc d'abscisse curviligne $t_0 + i\Delta r$. L'intérêt d'une telle expression est de définir la ligne de champ comme l'enchaînement de « courbes infinitésimales » , segments de droites de même longueur $\Delta r = \|\bar{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_i\|_2$ et colinéaires à $\nabla I_{\mathsf{H}}(\bar{\mathbf{u}}_i)$. Dans notre cas, les points $\bar{\mathbf{u}}_i$ ne sont pas régulièrement espacés le long de l'abscisse curviligne. Ils sont pris successivement sur les « ellipses de contour » (qui sont les équipotentielles de I_{H}), lieux des points sur lesquels l'information de direction du gradient est disponible et fiable grâce à un fort contraste. L'idée est d'introduire un nombre important de couronnes induisant une valeur Δr suffisamment faible pour garantir une bonne approximation des lignes de champ (illustrée sur la figure 5.9).

Encodage de la structure euclidienne Nous représentons par $\mathbf{c}$ (le vecteur de) l'*image du centre* et par $\bar{\mathbf{c}}$ le vecteur de coordonnées cartésiennes pixéliques de telle façon que $\mathbf{c} \sim [\bar{\mathbf{c}}^\top, 1]^\top$. Lors de la construction des lignes de champ à partir du motif C^{2}Tag, il est possible de s'approcher du point recherché $\mathbf{c}$ sans jamais l'atteindre. Nous allons voir qu'il est toutefois possible de contourner le problème. La structure euclidienne encodée par l'image d'un C^{2}Tag correspond à la donnée de l'image de la paire de points cycliques du plan de support du C^{2}Tag, ou de façon équivalente, au couple formé par l'image d'un cercle du plan et l'image de son centre. Or, le centre d'un cercle peut être considéré comme un cercle (dégénéré) de rayon nul, formé par la paire de droites isotropes passant par celui-ci, cf. définition 22 de la section 3.4.1. On peut donc définir la paire de points cycliques comme la (double) paire de points issue de l'intersection du cercle (de rayon unitaire par exemple) et du cercle de rayon nul. Notre idée est d'utiliser le fait que la paire de points cycliques est aussi la double paire de points issue de l'intersection du cercle et d'un autre cercle concentrique d'un certain rayon non nul [Kim 2005]. On peut donc remplacer l'image du cercle de rayon nul (c.-à-d. du centre) par l'image du cercle de rayon minimum. Cela va nous pemettre de faire converger les lignes de champ vers $\mathbf{c}$ en mesurant le nombre d'intersections de celles-ci avec l'image du cercle de rayon minimum, sachant que cette ellipse est intersectée par l'ensemble des lignes de champ [1] puisqu'elle constitue un contour fermé contenant $\mathbf{c}$.

Par la suite, nous appelerons *ellipse externe*, que nous noterons Q_1 ou simplement Q l'image du cercle de rayon maximum (c.-à-d. unitaire) et *ellipse interne*, que nous noterons Q_{2N} l'image du cercle de rayon minimum.

1. Cette propriété peut également être prouvée à partir de la loi de conservation du flux telle qu'énoncée par l'équation (5.8).

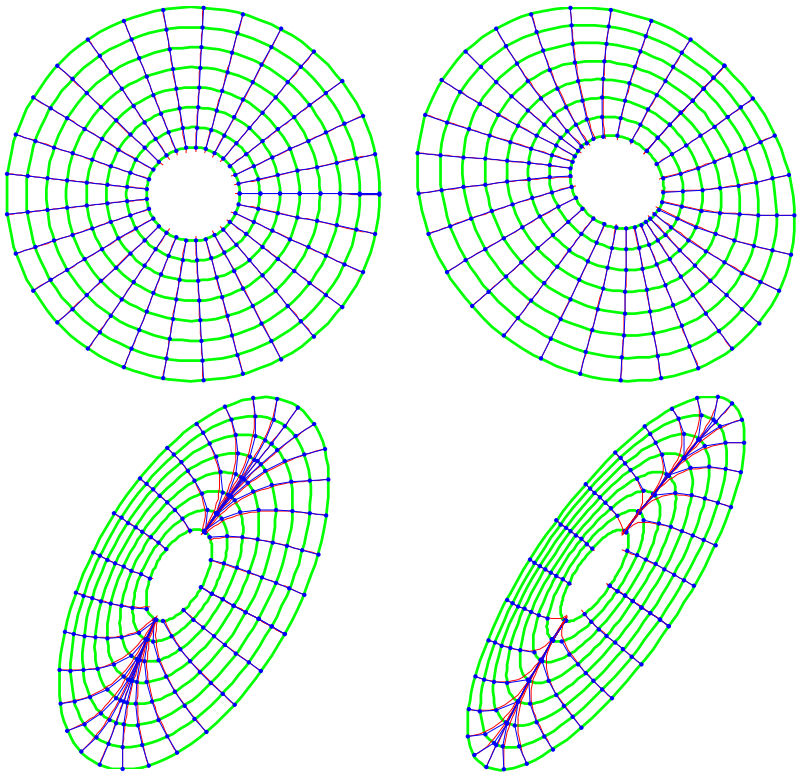


FIGURE 5.9: Évaluation qualitative des approximations des lignes de champ construites à partir des lignes de champ exactes. Il s'agit des images d'un motif constitué de 8 cercles concentriques (contours des 4 couronnes) de rayons régulièrement espacés entre 0.25 et 1. Les courbes vertes représentent les ellipses $\mathcal{C}_i$, $i = 1..8$, images des cercles de rayons décroissants. Les angles entre le plan de support et le plan image de la caméra valent 0°, 30°, 60° et 70° (de la gauche vers la droite et de haut en bas, respectivement). Les lignes rouges représentent les lignes de champ de $\nabla\alpha_{\mathsf{H}}$. Les lignes bleues sont les approximations des lignes de champ, constituées d'enchaînements de segments de droites. Chacun de ces segments a pour extrémités un couple de points $(\mathbf{u}_i, \mathbf{u}_{i+1}) \in \mathcal{C}_i \times \mathcal{C}_{i+1}$ et est dirigé orthogonalement à $\mathcal{C}_i$ au point $\mathbf{u}_i$.

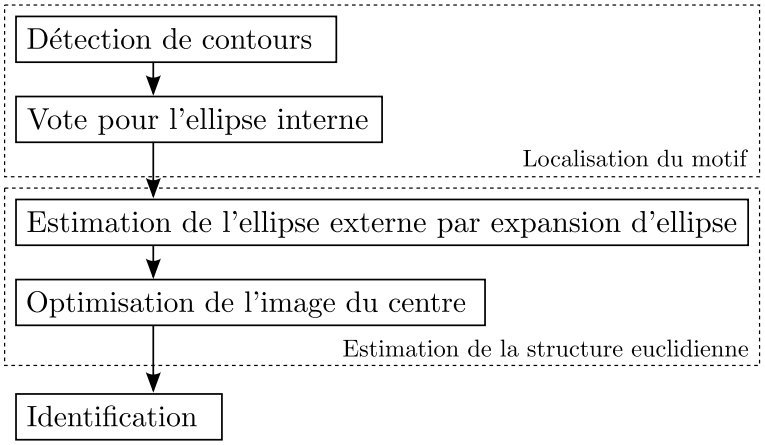


FIGURE 5.10: Diagramme de flux du système de détection.

5.4.3 Aperçu du système de détection

Bien que le système de détection soit robuste au changement d'échelle de l'image du motif, une analyse multi-résolution de la photographie est effectuée afin de réduire la complexité de certains des algorithmes qui le composent. Ainsi, parmi l'ensemble des étapes de l'algorithme de détection (présentées sur la figure 5.10), celles dédiées à la localisation du motif sont exécutées sur chacun des niveaux d'une pyramide d'images. Afin de garantir des performances optimales concernant les étapes d'*optimisation de l'image du centre* et d'*identification* (décrites plus loin), celles-ci sont effectuées sur l'image de base de la pyramide, c.-à-d. sur la photographie originale non sous-échantillonnée. Dans la suite et sauf mention contraire, le terme *image* désigne un niveau de la pyramide.

L'image que nous traitons est une image binaire de points de contour calculée à partir de l'algorithme décrit dans [Canny 1986]. On cherche à détecter des paires de points formés par deux points de contour, un pour l'ellipse externe et un pour l'ellipse interne. L'ensemble des points de contour est soumis à un vote pour l'élection de points de contour situés sur l'ellipse interne, selon un procédé qui « garantit » qu'un point de contour « élu » l'est nécessairement par un point de contour de l'ellipse externe. Les points de contour de l'ellipse interne « élus » sont segmentés en ensembles convexes de points de contour connexes. À partir des points de contour qui ont voté pour une ellipse interne, on effectue une estimation robuste de l'ellipse externe, à laquelle ils sont supposés appartenir. Cette étape constitue l'initialisation de l'algorithme d'*expansion d'ellipse* présenté dans [Kanatani 2004], qui vise à collecter la totalité des points de contour situés sur l'ellipse externe.

Chacun des motifs détectés est ensuite reprojeté dans l'image de base de la pyramide (comme illustré sur la figure 5.11). L'image de son centre est optimisée, fournissant conjointement la structure euclidienne du plan de support du C^{2}Tag. L'identification est alors possible en analysant les signaux 1D situés entre l'image du centre du motif et un certain nombre de points situés sur l'ellipse externe.

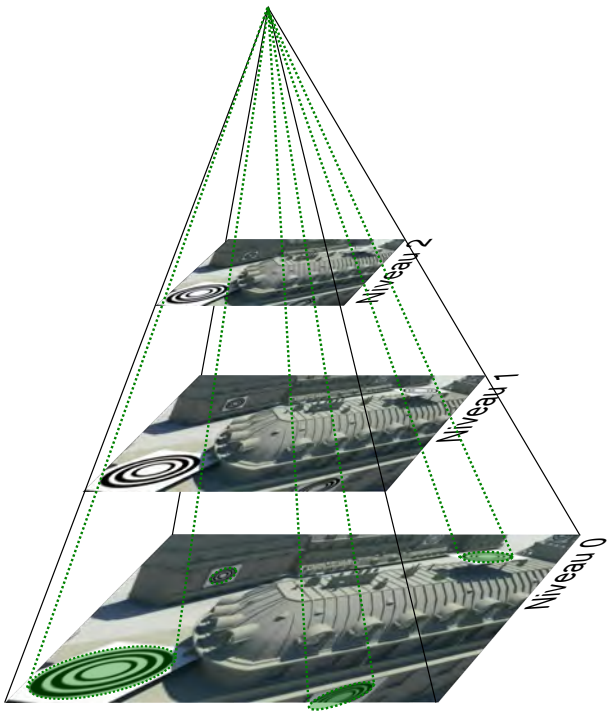


FIGURE 5.11: Pyramide d'images.

5.4.4 Détection

5.4.4.1 Système de vote pour l'ellipse interne

Les fondements théoriques de la procédure de vote pour la localisation de l'ellipse interne sont présentés dans la section 5.4. Le motif d'un C^{2}Tag est composé de N couronnes circulaires, chacune étant délimitée par deux cercles concentriques. L'image de ce marqueur est ainsi formée par $2N$ contours elliptiques. On cherche à détecter des couples de points de contour $(\mathbf{u}_1^j, \mathbf{u}_{2N}^j)$ tels que $\mathbf{u}_1^j$ est un point de contour de l'ellipse externe Q_1 et $\mathbf{u}_{2N}^j$ est un point de contour de l'ellipse interne Q_{2N}. La détection est formulée comme un problème de vote pour $\mathbf{u}_{2N}^j$, c.-à-d. le point de contour de l'ellipse interne. Dans le système de vote décrit, ce sont des points (et non des droites) qui sont accumulés dans l'espace de paramètres, à la différence des approches décrites dans [Yuen 1989, Ying 2007, Ying 2008].

Protocole de vote. Soit I la fonction de niveau de gris associée à la photographie du C^{2}Tag. On sélectionne initialement un point de contour (arbitraire) qui émet une *intention de vote*. Ce vote n'est un *vote effectif* que si l'on peut construire à partir de ce point une ligne polygonale $\mathcal{T}^j = \{\mathbf{u}_i^j\}_{i=1..2N}$, dont les sommets sont $2N$ points de contour $\mathbf{u}_1^j ... \mathbf{u}_{2N}^j$, qui vérifie certaines conditions d'approximation d'une ligne de champ du potentiel I. Si c'est le cas, le vote est émis et a pour objet le *seul* point $\mathbf{u}_{2N}^j$ pour lequel on fait l'hypothèse qu'il est un point de contour d'une ellipse interne Q_{2N}. Les conditions pour qu'une ligne polygonale $\mathcal{T}^j$ approche un arc d'une ligne de champ sont les suivantes. Pour tout $i \in \{1, ..., 2N-1\}$,

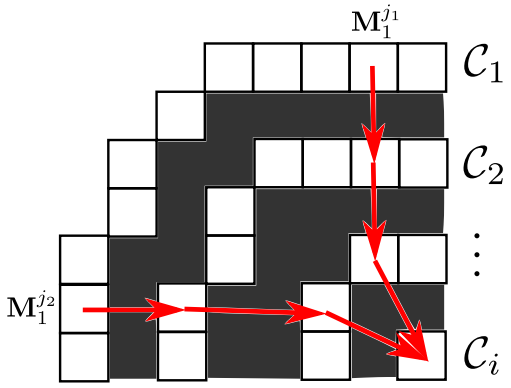


FIGURE 5.12: Illustration de l'algorithme de vote pour l'ellipse interne avec deux exemples d'intention de vote, c.-à-d. deux points $\mathbf{u}_1^{j_1}$ et $\mathbf{u}_1^{j_2}$ à partir desquels l'algorithme de vote (présenté dans la table 5.1) est exécuté. Les segments de droites d'extrémités $(\mathbf{u}_i^{j_1}, \mathbf{u}_{i+1}^{j_2})$ sont indiqués par les flèches rouges.

1. $\bar{\mathbf{u}}_{i+1} = \bar{\mathbf{u}}_i + a_i \nabla I(\bar{\mathbf{u}}_i)$, où $a_i < 0$ si i est impair et $a_i > 0$ sinon ;
2. les segments $\mathbf{u}_i^j \mathbf{u}_{i+1}^j$ ne contiennent pas des points de contour autres que les sommets ;
3. les segments $\mathbf{u}_i^j \mathbf{u}_{i+1}^j$ ne se coupent pas.

Pour chaque point de contour $\mathbf{u}_i^j$, la construction de la ligne polygonale $\mathcal{T}^j$ est ainsi régie par la même idée d'intention de vote : (*i*) une intention de vote correspond au tracé dans la vue d'une demi-droite d'origine $\mathbf{u}_i^j$ et de direction colinéaire au gradient en ce point ; (*ii*) dès que la demi-droite rencontre un point de contour $\mathbf{u}_{i+1}^j$, le segment de droite d'extrémités $(\mathbf{u}_i^j, \mathbf{u}_{i+1}^j)$ est conservé et une nouvelle demi-droite d'origine $\mathbf{u}_{i+1}^j$ et de direction colinéaire au gradient en $\mathbf{u}_{i+1}^j$ est tracée ; (*iii*) lorsqu'un $2N^{\text{ème}}$ point de contour $\mathbf{u}_{2N}^j$ est rencontré le vote est attribué à celui-ci. Notons que le tracé de la première demi-droite est effectué dans le sens opposé au gradient, les pixels de l'ellipse externe correspondant à des transitions du signal « blanc $\rightarrow$ noir ». Ensuite, en chaque point de rencontre, le sens de tracé est alterné, les transitions du signal le long des lignes de champ étant alternativement « blanc $\rightarrow$ noir » et « noir $\rightarrow$ blanc » , etc. L'algorithme de tracé d'une ligne polygonale candidate à être ligne de champ est décrit dans la table 5.1 et illustré sur la figure 5.12.

En pratique, il est possible d'introduire une hiérarchie de règles de décision (heuristiques ou issues d'algorithmes de classification supervisée) qui définissent la validité d'un ligne polygonale $\mathcal{T}^j$, portant par exemple sur les positions relatives des points $\{\mathbf{u}_i^j\}_{j=1..2N}$. Ainsi, une intention de vote violant l'une de ces règles, lors de la construction de $\mathcal{T}^j$, n'aboutira pas. L'utilisation de telles règles est intéressante afin de réduire la complexité de l'algorithme de vote, mais aussi le nombre de faux positifs. La procédure de vote appliquée à des images réelles est illustrée sur les figures 5.15 et 5.16.

5.4.4.2 Regroupement des candidats en segments de contour de l'ellipse interne

Chaque point de contour ayant reçu un nombre « suffisant » de votes est candidat à être un point de contour de l'ellipse interne. Les candidats sont regroupés par une procédure de chaînage afin de constituer des segments de contour de l'ellipse interne approchant des arcs de cette ellipse.

Pour chaque pixel de contour p de l'image de contour C
  $\bar{\mathbf{u}}_1^j \leftarrow p$
  $i \leftarrow 1$
  $sens \leftarrow -1$
  Tant que $i < 2N$
    $\bar{\mathbf{u}}_{i+1}^j \leftarrow \text{tracer}(\bar{\mathbf{u}}_i^j, sens * \mathbf{grad}(\bar{\mathbf{u}}_i^j), C)$
    $sens \leftarrow -sens$
    $i \leftarrow i + 1$
  Fin Tant que
Fin Pour

TABLE 5.1: Algorithme de traçé d'une ligne polygonale candidate à être ligne de champ. $\bar{\mathbf{u}}$' $= \text{tracer}(\bar{\mathbf{u}}, \bar{\mathbf{v}}, C)$ est une fonction de tracé de segment de droite, d'extrémité $\bar{\mathbf{u}}$ et de vecteur directeur $\bar{\mathbf{v}}$, dans un plan discret retournant le premier pixel de contour $\bar{\mathbf{u}}' \in C$ rencontré.

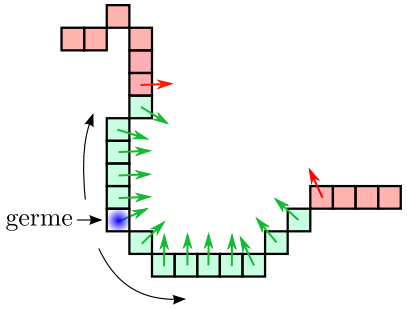


FIGURE 5.13: Segmentation convexe des segments de contour. La tache bleue correspond à la position du germe. Les flèches noires indiquent les directions prises par l'algorithme de chaînage de part et d'autre du germe. L'ensemble des pixels verts représente le segment de contour convexe obtenu. Les flèches rouges et vertes représentent les directions des gradients en chacun des pixels. Le chaînage est interrompu lorsque la direction du gradient est divergente (notée en rouge).

Cette segmentation est effectuée à partir d'une contrainte imposant que le segment de contour soit convexe. La raison pour laquelle le problème de validation des candidats est formulée comme un problème de segmentation en contours convexes et non pas en contours elliptiques est que, dans le cas d'une faible résolution du motif dans une vue, une imprécision importante de localisation des points de contour de l'ellipse interne, relativement à la taille de celle-ci, est envisagée.

Validation de segments de contour convexes. Cette étape a pour objectif de valider tout segment de contour convexe dont la moyenne des votes reçus pour les points le constituant est « suffisamment grande ». Une fois validé, ce segment est considéré comme approchant un arc de l'ellipse interne.

Pour chaque point de contour p, nommé *germe*, ayant reçu au moins σ_ϵ votes, un algorithme de chaînage est exécuté de part et d'autre de celui-ci (illustré sur la figure 5.13). Le chaînage est interrompu lorsque le gradient du point de contour courant est divergent par rapport au gradient du point précédent. Une fois les deux chaînages interrompus, on dispose alors d'une ligne polygonale, définissant un contour convexe et notée $\mathcal{C}_{2N}^p$, contenant le point p (illustré sur la figure 5.13). Le seuil σ_ϵ est défini dans la proposition qui suit comme le rapport entre le rayon du cercle externe

et celui du cercle interne. Cette proposition répond à la question « *quels candidats $\mathcal{C}_{2N}^p$ doit-on conserver ?* » L'objectif est ici de fournir un critère de validation des segments de contour convexes collectés pour approcher un arc de l'ellipse interne.

Proposition 35 *Un segment de contour convexe $\mathcal{C}_{2N}^p$ est conservé si*

$$\bar{V}_2^p > \sigma_\epsilon \tag{5.7}$$

où $\bar{V}_2^p$ représente la valeur moyenne du nombre de votes reçus par le segment de points de contour convexe $\mathcal{C}_{2N}^p$ et où $\sigma_\epsilon = 1/\epsilon$ représente le rapport de rayon entre le cercle externe (de rayon 1) et le cercle interne (de rayon ϵ).

Preuve. La démonstration repose sur la loi de conservation du flux du champ de vecteurs $\nabla\alpha_{\mathsf{H}}$ comme suit. Soit le contour fermé ∂S de la surface S délimitée par $\mathcal{A}_{\alpha_1}$, $\mathcal{A}_{\alpha_2}$, arcs des ellipses $\mathcal{C}_{i_1}^p$ et $\mathcal{C}_{i_2}^p$ respectivement, et deux lignes de champ $\mathcal{T}_1$ et $\mathcal{T}_2$ (cf. figure 5.14). Un résultat important concernant le champ de vecteurs $\nabla\alpha_{\mathsf{H}}$ est qu'il est conservatif. En effet, d'après le théorème du gradient, le champ de vecteur $\nabla\alpha_{\mathsf{H}}$ étant défini comme la dérivée d'un champ scalaire (ici α_{H}), alors celui-ci est conservatif :

$$0 = \oint_{\partial S} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} = \int_{\mathcal{A}_{\alpha_1}} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} + \int_{\mathcal{A}_{\alpha_2}} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} + \int_{\mathcal{T}_1} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} + \int_{\mathcal{T}_2} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} \tag{5.8}$$

où $\mathbf{dl}$ est le vecteur normal à la surface S, dirigé vers l'extérieur et de longueur égale à l'élément de courbe qu'il représente. Sur les arcs $\mathcal{T}_1$ et $\mathcal{T}_2$, le produit scalaire $\nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl}$ est nul puisque $\nabla\alpha_{\mathsf{H}}$ est, en tout point, tangent à la ligne de champ. Sur les arcs $\mathcal{A}_{\alpha_1}$ et $\mathcal{A}_{\alpha_2}$, les vecteurs $\nabla\alpha_{\mathsf{H}}$ et $\mathbf{dl}$ sont colinéaires, de même sens sur $\mathcal{A}_{\alpha_2}$ et de sens opposés sur $\mathcal{A}_{\alpha_1}$. Ainsi, l'équation (5.8) est équivalente à

$$\begin{aligned} -\int_{\mathcal{A}_{\alpha_1}} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} &= \int_{\mathcal{A}_{\alpha_2}} \nabla\alpha_{\mathsf{H}} \cdot \mathbf{dl} \\ \Rightarrow \int_{\mathcal{A}_{\alpha_1}} \|\nabla\alpha_{\mathsf{H}}\|_2 dl &= \int_{\mathcal{A}_{\alpha_2}} \|\nabla\alpha_{\mathsf{H}}\|_2 dl \\ \Rightarrow \bar{V}_1 L_{\alpha_1} &= \bar{V}_2 L_{\alpha_2} \\ \Leftrightarrow \bar{V}_2 &= \bar{V}_1 \frac{L_{\alpha_1}}{L_{\alpha_2}} \end{aligned} \tag{5.9}$$

où $\bar{V}_k$ désigne la moyenne de $\|\nabla\alpha_{\mathsf{H}}\|_2$ sur l'arc $\mathcal{A}_{\alpha_k}$ et L_{α_k} la longueur de celui-ci, pour $k = 1, 2$.

Si l'on considère à présent que les intentions de vote émises par l'ellipse externe $\mathcal{C}_1^p$ représentent le flux $\nabla\alpha_{\mathsf{H}}$ à travers l'arc $\mathcal{A}_1$, c.-à-d. à travers les L_1^p points de contour, éléments de courbes de $\mathcal{A}_1$, telles que la valeur moyenne du flux soit $\bar{V}_1^p = 1$, alors le nombre d'intentions de vote reçues, en moyenne sur les points de contour de $\mathcal{A}_\epsilon$, image de l'arc du cercle interne de rayon ϵ

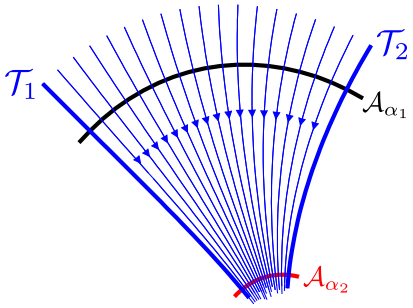


FIGURE 5.14: Mise en évidence de la loi de conservation du flux sur une surface S délimitée par les deux arcs $\mathcal{A}_{\alpha_1}$ et $\mathcal{A}_{\alpha_2}$ et les lignes de champ $\mathcal{T}_1$ et $\mathcal{T}_2$.

composé de L_ϵ^p points de contour, est égal à

$$\bar{V}_2^p = \frac{L_1^p}{L_\epsilon^p} \tag{5.10}$$

Ainsi, l'ensemble des segments de contour convexes $\mathcal{C}_{2N}^p$ conservés est

$$C_+^{(1)} = \left\{ \mathcal{C}_{2N}^p \mid \bar{V}_2^p > \sigma_\epsilon \right\} \tag{5.11}$$

où $\sigma_\epsilon = 1/\epsilon$ représente le rapport de rayon entre le cercle externe (de rayon 1) et le cercle interne (de rayon ϵ), et où $\bar{V}_2^p$ représente la valeur moyenne du nombre d'intentions de vote reçues par le segment de points de contour convexe $\mathcal{C}_{2N}^p$. ■

5.4.4.3 Estimation initiale de l'ellipse externe

Soit $C_+^{(1)}$ l'ensemble des segments de contour convexes conservés à l'étape précédente pour constituer l'ellipse interne. Pour chacun de ces segments, l'ensemble des points de contour ayant voté sur ce segment est collecté. Ceci est immédiat puisque, lors du vote, un point de contour $\mathbf{u}_{i+1}^j$ est indexé par le point de rencontre $\mathbf{u}_i^j$ précédent.

Soit E l'ensemble de points de contour ayant voté pour le segment de contour $\mathcal{C}^p \in C_+^{(1)}$ de l'ellipse interne et donc supposé contenir des points de contour de l'ellipse externe. Dans la suite, l'exposant p est omis pour alléger les notations et nous notons Q (la matrice de) l'ellipse externe. Une première estimation robuste de l'ellipse Q associée à l'ensemble des points de contour E est effectuée via l'algorithme LMedS (*Least Median of Squares*). L'objectif est d'éliminer les points de contour aberrants au sein de l'ensemble E et d'obtenir une première estimation de l'ellipse externe Q. L'algorithme LMedS est présenté dans la table 5.5, en annexe à ce chapitre.

Expansion de l'ellipse externe. L'algorithme d'expansion d'ellipse proposé dans [Kanatani 2004] est appliqué à l'ensemble E des points collectés sur l'ellipse externe dont l'estimation courante est notée Q. Le principe de l'algorithme est le suivant : (i) une enveloppe elliptique autour de Q est calculée ; celle-ci est définie par deux ellipses, expansion et contraction de Q, notées $\mathsf{Q}^{(+)}$ et $\mathsf{Q}^{(-)}$, telles que la valeur de chacun des demi-axes soit respectivement augmentée et réduite de δ pixels ;

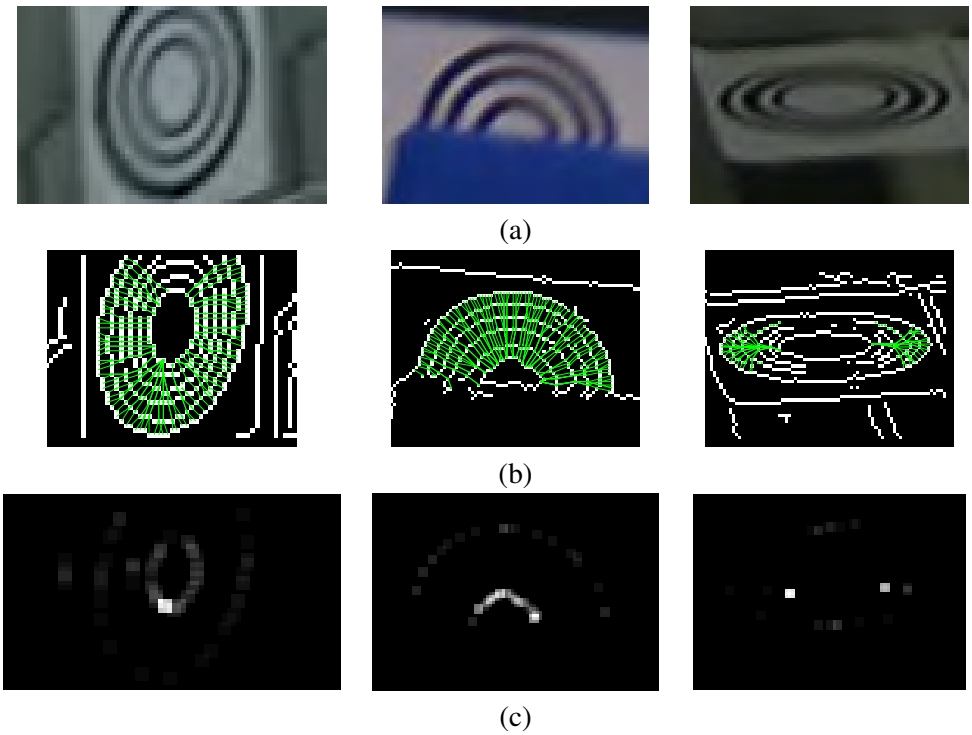
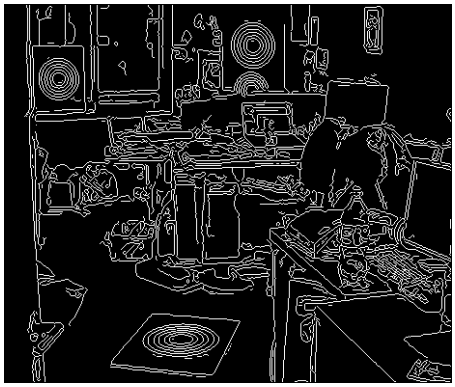


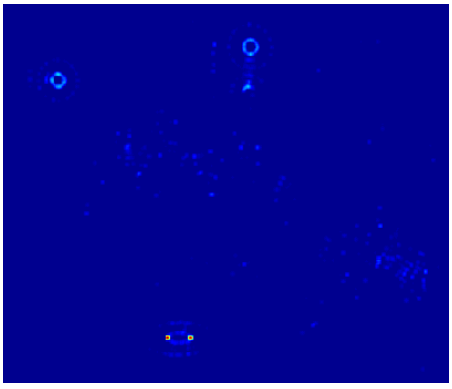
FIGURE 5.15: (a) Trois images de C^{2}Tags. (b) Les images binaires issues de la détection de contours sur lesquelles sont superposées les intentions de vote $\mathcal{T}^j$ des pixels de l'ellipse externe vers ceux de l'ellipse interne. (c) Images du résultat du vote sur lesquelles un filtre gaussien a été appliqué afin de pallier au manque de visibilité du résultat obtenu lié à l'échantillonnage.



(a)



(b)



(c)

FIGURE 5.16: Résultat du vote obtenu sur une image réelle. (a) Image originale en niveaux de gris. (b) Image binaire issue de la détection de contours. (c) Résultat du vote.

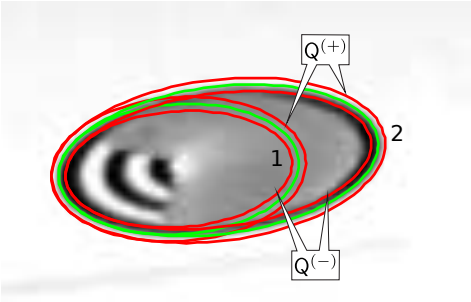


FIGURE 5.17: Illustration de l'algorithme d'expansion d'ellipse. Ce dernier est ici interrompu au bout de deux itérations. Les ellipses vertes représentent les ellipses Q estimées à chacune des itérations. Les ellipses rouges correspondent aux ellipses $\mathsf{Q}^{(+)}$ et $\mathsf{Q}^{(-)}$ délimitant l'enveloppe elliptique.

(ii) l'ensemble des points de contour connexes aux points de E et présents dans l'enveloppe sont collectés et l'ellipse Q est à nouveau estimée à partir de ce nouvel ensemble de points ; (iii) E et Q sont mis à jour. La procédure est répétée jusqu'à ce qu'il n'y ait plus aucun point ajouté à l'ensemble E. L'algorithme est présenté dans la table 5.2 et son principe général illustré sur la figure 5.17. Plus de détails sont fournis en annexe à ce chapitre et dans [Kanatani 2004].

Pour chaque ensemble de pixels candidats E
  $\mathsf{Q} \leftarrow \mathrm{ajusterEllipse}(E)$
  $(\mathsf{Q}^+, \mathsf{Q}^-) \leftarrow \mathrm{calculerEnveloppe}(\mathsf{Q}, \delta)$
  $n \leftarrow 0$
  Tant que $|E| > n$
    $n \leftarrow |E|$
    Pour chaque pixel p de E
      $E \leftarrow E \cup \mathrm{pixelsConnexes}(p, I, \mathsf{Q}^{(+)}, \mathsf{Q}^{(-)})$
    Fin Pour
  Fin Tant que
Fin Pour

TABLE 5.2: Algorithme d'expansion d'ellipse. La fonction $\mathrm{ajusterEllipse}(E)$ estime l'ellipse Q à partir de l'ensemble de points E tel que décrit dans [Fitzgibbon 1999]. La fonction $\mathrm{calculerEnveloppe}(\mathsf{Q}, \delta)$ calcule les ellipses $\mathsf{Q}^{(+)}$ et $\mathsf{Q}^{(-)}$ de l'enveloppe elliptique définie par Q et la marge δ. Enfin la fonction $\mathrm{pixelsConnexes}(p, I, \mathsf{Q}^{(+)}, \mathsf{Q}^{(-)})$ retourne l'ensemble des pixels connexes au pixel p et situés dans la région délimitée par les ellipses $\mathsf{Q}^{(+)}$ et $\mathsf{Q}^{(-)}$.

À l'issue de cette étape d'expansion d'ellipse, les couples (Q_i, E_i) conservés sont ceux qui vérifient le critère suivant

$$C_+^{(2)} \leftarrow \{\{\mathsf{Q}_i, E_i\} \mid |E_i|^2 > \alpha L_{\mathsf{Q}_i}\} \tag{5.12}$$

où L_{Q_i} désigne la circonférence (en pixels) de l'ellipse Q_i et où $\alpha = 60$ est choisi empiriquement. Le calcul de L_{Q_i} est décrit en annexe à ce chapitre. Il s'agit d'une heuristique qui vise à conserver seulement les arcs d'ellipses représentant une partie suffisamment importante de la circonférence totale de l'ellipse.

5.4.5 Optimisation de l'image du centre

Nous souhaitons à présent définir une *homographie de rectification métrique* transformant les ellipses de l'image d'un C^{2}Tag en des cercles centrés en l'origine. Nous dirons qu'une telle homographie est une *homographie de rectification métrique centrée*. Nous donnons les définitions suivantes.

Définition 36 *Soit $\mathcal{H}$ une homographie, de matrice H, du plan des pixels vers un plan euclidien.*

- *$\mathcal{H}$ est une homographie de rectification métrique si et seulement si elle « envoie » le vecteur $\tilde{\mathbf{I}}_\pm$ de l'image de la paire de points cycliques (du plan de support du C^{2}Tag) dans sa forme canonique, c.-à-d. si elle vérifie*

$$\mathsf{H}\tilde{\mathbf{I}}_\pm \sim (1, \pm i, 0)^\top \tag{5.13}$$

- *$\mathcal{H}$ est une homographie de rectification métrique centrée par rapport à une ellipse E, de matrice Q, si $\mathcal{H}$ est une homographie de rectification métrique qui transforme E en un cercle centré en l'origine, c.-à-d. si*

$$\mathsf{H}^{-\top}\mathsf{Q}\mathsf{H}^{-1} \sim \mathrm{diag}(1, 1, -R^2) \tag{5.14}$$

où $R > 0$ désigne le rayon du cercle.

Nous posons la question de trouver un paramétrage minimal de la matrice H d'une homographie de rectification centrée qui soit déterminé par une ellipse Q, image d'un cercle du C^{2}Tag. Afin de simplifier l'écriture de ce paramétrage, sans perte de généralité, nous supposons avoir effectué un changement de représentation affine du plan des pixels tel que $\mathsf{Q} = \mathrm{diag}(Q_{11}, Q_{22}, Q_{33})$.

Proposition 37 *Si $\mathsf{Q} = \mathrm{diag}(Q_{11}, Q_{22}, Q_{33})$ représente l'image d'un cercle et $\bar{\mathbf{c}} = [x_C, y_C]^\top$ représente l'image de son centre, alors toute homographie de rectification métrique centrée, fonction de Q et $\bar{\mathbf{c}}$, peut s'écrire sous la forme $\mathsf{H}(\mathsf{Q}, \bar{\mathbf{c}}) = \mathsf{H}_\mathcal{E}(\mathsf{Q}, \bar{\mathbf{c}})\mathsf{H}_\mathcal{A}(\mathsf{Q}, \bar{\mathbf{c}})\mathsf{H}_\mathcal{P}(\mathsf{Q}, \bar{\mathbf{c}})$, où*

$$\mathsf{H}_\mathcal{P}(\mathsf{Q}, \bar{\mathbf{c}}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Q_{11}x_C & Q_{22}y_C & Q_{33} \end{bmatrix} \tag{5.15}$$

$$\mathsf{H}_\mathcal{A}(\mathsf{Q}, \bar{\mathbf{c}}) = \begin{bmatrix} 1/b & -a/b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \textit{ avec } \begin{cases} a = \frac{Q_{22}x_C y_C}{Q_{11}x_C^2 + Q_{33}} \\ b = \left(\frac{Q_{22}Q_{33}}{Q_{11}} \left(\frac{Q_{11}x_C^2 + Q_{22}y_C^2 + Q_{33}}{(Q_{11}x_C^2 + Q_{33})^2} \right) - a^2 \right)^{1/2} \end{cases} \tag{5.16}$$

$$\mathsf{H}_\mathcal{E}(\mathsf{Q}, \bar{\mathbf{c}}) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \textit{ avec } [x, y] = \left[\frac{-x_C/b + y_C a/b}{Q_{11}x_C^2 + Q_{22}y_C^2 + Q_{33}}, -\frac{y_C}{Q_{11}x_C^2 + Q_{22}y_C^2 + Q_{33}} \right] \tag{5.17}$$

Si l'image Q du cercle n'est pas dégenérée, alors aucun des dénominateurs apparaissant dans (5.16-5.17) ne peut être nul :

- $Q_{11} \neq 0$ car aucune des valeurs propres de Q n'est nulle,
- $Q_{11}x_C^2 + Q_{22}y_C^2 + Q_{33} \neq 0$ car l'image $[x_C, y_C]$ du centre n'appartient pas à Q,
- $Q_{11}x_C^2 + Q_{33} \neq 0$ car le seul point du grand axe de Q qui appartienne à Q est, par définition, un sommet du grand axe et ce ne peut être le projeté « orthogonal » $[x_C, 0]$ de l'image du centre sur le grand axe.

Preuve. Toute homographie H de rectification métrique de l'image d'un motif plan se décompose [Hartley 2004b, p.42] en un produit de trois homographies $\mathsf{H} = \mathsf{H}_{\mathcal{E}}\mathsf{H}_{\mathcal{A}}\mathsf{H}_{\mathcal{P}}$ où $\mathsf{H}_{\mathcal{P}}$ est une matrice de rectification affine (« annulant » la distortion perspective) dont la dernière ligne correspond au transposé du vecteur de la ligne de fuite du plan de support du motif, $\mathsf{H}_{\mathcal{A}}$ est une matrice de rectification métrique (« annulant » la distortion affine) et $\mathsf{H}_{\mathcal{E}}$ est la matrice d'une similitude plane.

- La décomposition (5.15) se justifie par le fait que la ligne de fuite $\mathbf{v}$ du plan de support du C^{2}Tag est la droite polaire de l'image du centre $\mathbf{c} = (\bar{\mathbf{c}}^\top, 1)^\top$, par rapport à Q, c.-à-d. la droite $\mathbf{v} = \mathsf{Q}\mathbf{c}$.
- La décomposition (5.16) se justifie par les faits suivants. D'une part, nous pouvons représenter l'image de la paire de points cycliques sous forme duale par une conique de signature $(2, 0)$ duale à ceux-ci [Hartley 2004b, p. 53], dont la matrice de rang 2 s'écrit

$$\begin{aligned}\boldsymbol{\Sigma}^* &\sim \mathsf{H}^{-1}\mathrm{diag}(1,1,0)\mathsf{H}^{-\top} \\ &\sim \mathsf{H}_{\mathcal{P}}^{-1}\mathsf{H}_{\mathcal{A}}^{-1}\mathrm{diag}(1,1,0)\mathsf{H}_{\mathcal{A}}^{-\top}\mathsf{H}_{\mathcal{P}}^{-\top}\end{aligned}$$

  c.-à-d. indépendamment de $\mathsf{H}_{\mathcal{E}}$ du fait de l'invariance des points cycliques par les similitudes planes. D'autre part, la conique duale aux deux points d'intersection de toute droite $\mathbf{d}$ et toute conique C peut se décomposer sous la forme $[\mathbf{d}]_\wedge \, \mathsf{C} \, [\mathbf{d}]_\wedge$ [Hartley 2004b, p. 64]. Ceci donne, en particulier, pour la conique duale à l'image de la paire de points cycliques

$$\boldsymbol{\Sigma}^* \sim [\mathsf{Q}\mathbf{c}]_\wedge \, \mathsf{Q} \, [\mathsf{Q}\mathbf{c}]_\wedge \tag{5.18}$$

  puisque, par la définition 15 de la section 3.4.2, l'image de la paire de points cycliques est le lieu d'intersection de la ligne de fuite $\mathsf{Q}\mathbf{c}$ et de l'ellipse Q. On obtient alors l'égalité projective

$$\mathsf{H}_{\mathcal{P}}([\mathsf{Q}\mathbf{c}]_\wedge \, \mathsf{Q} \, [\mathsf{Q}\mathbf{c}]_\wedge)\mathsf{H}_{\mathcal{P}}^\top \sim \mathsf{H}_{\mathcal{A}}^{-1}\mathrm{diag}(1,1,0)\mathsf{H}_{\mathcal{A}}^{-\top} \tag{5.19}$$

$$\underbrace{\begin{bmatrix} ([\mathsf{Q}\mathbf{c}]_\wedge \, \mathsf{Q} \, [\mathsf{Q}\mathbf{c}]_\wedge)_{1:2,1:2} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathsf{X}} \sim \begin{bmatrix} a^2+b^2 & a & 0 \\ a & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.20}$$

  que l'on résoud exactement d'abord pour $a = X_{12}/X_{22}$ et puis pour $b = \sqrt{X_{11}/X_{22} - a^2}$.
- La décomposition (5.17) se justifie par le fait que $[x, y]^\top = -\mathbf{g}_{(\mathsf{H}_{\mathcal{A}}\mathsf{H}_{\mathcal{P}})}(\bar{\mathbf{c}})$ où $\mathbf{g}_{(\mathsf{H}_{\mathcal{A}}\mathsf{H}_{\mathcal{P}})}$ est la fonction homographique définie en (5.2) à partir de la matrice $\mathsf{H}_{\mathcal{A}}\mathsf{H}_{\mathcal{P}}$.

■

Connaissant Q, cette proposition nous permet de définir un *paramétrage minimal* $\mathsf{H}_{\mathsf{Q}}(\bar{\mathbf{c}})$ d'une homographie de rectification métrique centrée à partir de la donnée de Q, dont les deux seuls *deux degrés de liberté* sont les coordonnées (x_C, y_C) de $\bar{\mathbf{c}}$.

La fonction de coût (5.22) associée à l'optimisation de l'image du centre repose sur le fait que si $\mathbf{u}_1$ et $\mathbf{u}_2$ sont deux points distincts de E et que $\mathbf{g}_{\mathsf{H}}$ soit définie comme en (5.2) à partir de $\mathsf{H} = \mathsf{H}(\mathsf{Q}, \bar{\mathbf{c}})$, alors on a

$$\forall \alpha \in [0,1], \qquad I\left(\mathbf{g}_{\mathsf{H}}^{-1}\left(\alpha \mathbf{g}_{\mathsf{H}}(\bar{\mathbf{u}}_1)\right)\right) = I\left(\mathbf{g}_{\mathsf{H}}^{-1}\left(\alpha \mathbf{g}_{\mathsf{H}}(\bar{\mathbf{u}}_2)\right)\right) \tag{5.21}$$

où $I(\bar{\mathbf{u}})$ désigne le niveau de gris de la photographie au point $\bar{\mathbf{u}}$. Cette proposition exprime le fait que deux points, dont les rétroprojections sont à égale distance du centre du marqueur $\bar{\mathbf{o}} = [0,0]^\top$, ont même niveau de gris.

Fonction de coût. Soit $\mathbf{u}_j \in E$ un point situé sur l'ellipse externe. Dans ce qui suit, nous désignons par *coupe* en $\mathbf{u}_j$ le segment de droite de la vue dont l'extrémité initiale est l'image du centre $\mathbf{c}$ du marqueur et l'extrémité finale est le point $\mathbf{u}_j$. Soit

$$\left\{ \alpha_t \bar{\mathbf{m}}_j \mid \alpha_t = \frac{t-1}{T-1},\, t = 1..T \right\}$$

une discrétisation en T segments de la rectification métrique par $\mathsf{H}(\bar{\mathbf{c}})$ de la coupe en un point de l'ellipse externe $\mathbf{u}_j \in E$, c.-à-d. tel que $\bar{\mathbf{m}}_j = \mathbf{g}_{\mathsf{H}}(\bar{\mathbf{u}}_j)$.

L'optimisation de l'image du centre consiste à chercher $\hat{\mathbf{c}}$ solution du problème suivant :

$$\underset{\bar{\mathbf{c}}}{\operatorname{argmin}} \sum_{\substack{(j_1,j_2)\in J \\ j_1 \neq j_2}} \sum_{t=1}^{T} \left\{ I\left(\mathbf{g}_{\mathsf{H}(\bar{\mathbf{c}})}^{-1}(\alpha_t \bar{\mathbf{m}}_{j_1})\right) - I\left(\mathbf{g}_{\mathsf{H}(\bar{\mathbf{c}})}^{-1}(\alpha_t \bar{\mathbf{m}}_{j_2})\right) \right\}^2 \tag{5.22}$$

où $J = [1, .., |E|]^2$ désigne l'ensemble des couples (j_1, j_2) de points de contour de E et la valeur $\mathcal{I}\left(\mathbf{g}_{\mathsf{H}(\bar{\mathbf{c}})}^{-1}(\alpha_t \bar{\mathbf{m}}_{j_k})\right)$, $k = 1, 2$, représente le niveau de gris de l'image d'un point de la discrétisation du segment $\bar{\mathbf{o}}\bar{\mathbf{m}}_{j_k}$.

Il s'agit d'un problème d'estimation au sens des moindres carrés non linéaires. La valeur initiale de l'image du centre $\bar{\mathbf{c}}_{(0)}$ correspond aux coordonnées du centre de l'ellipse externe Q. L'optimisation est conduite via un algorithme de type quasi-Newton. Une représentation, sur une vue de synthèse, de l'ensemble des niveaux de gris collectés sur les coupes, avant et après l'optimisation, est fournie sur la figure 5.18. Notons au passage qu'une telle méthode d'ajustement de l'homographie H est indépendante de l'identifiant du marqueur.

L'étape de sélection de coupe présentée dans le paragraphe suivant vise à répondre aux trois sous-problèmes suivants, soulevés par la fonction de coût (5.22) :

1. Celle-ci permet de résoudre le problème d'estimation de l'image du centre en supposant que

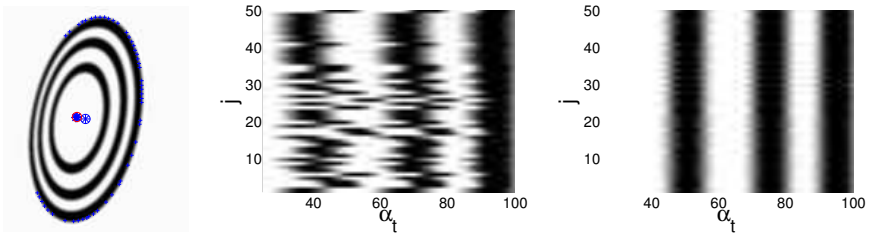


FIGURE 5.18: (a) Position de l'image du centre estimée à chaque itération de l'algorithme d'optimisation. Les cercles bleu et rouge représentent les positions avant et après l'optimisation respectivement. (b) Signaux collectés sur les coupes avant l'optimisation où le signal affiché au point de coordonnées (α_t, j) correspond au signal collecté sur l'échantillon α_t collecté sur la coupe j (cf. fonction de coût (5.22)). (c) Signaux collectés sur les coupes après l'optimisation.

l'image n'est pas dégradée par du flou de bougé. Or, comme nous l'avons mentionné dans l'introduction de ce chapitre, nous souhaitons que les algorithmes de traitement puissent détecter et identifier les images des C^{2}Tags, y compris en présence de flou de bougé.

2. Au regard de l'équation (5.22), la complexité de l'algorithme d'optimisation dépend essentiellement du nombre de points $\mathbf{u}_j \in E$ situés sur l'ellipse externe. Un cardinal $|E|$ trop élevé va induire un temps de calcul élevé ce qui s'oppose aux critères de performances attendus.

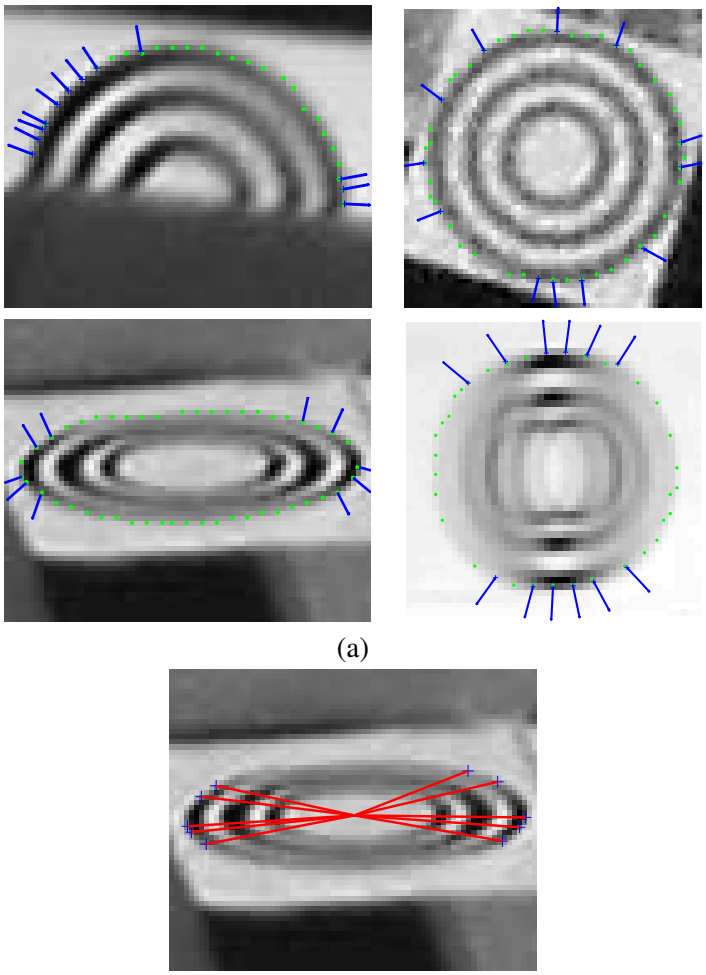
3. La répartition des points $\mathbf{u}_j \in E$ situés sur l'ellipse externe conditionne la convergence de l'optimisation (5.22).

Initialisation - Sélection des coupes. Afin de résoudre ces trois sous-problèmes, nous avons conçu un algorithme de sélection de coupes visant à collecter un sous-ensemble $S^E \subseteq E$ de points de contour sur l'ellipse externe, utilisé lors de l'optimisation (5.22), qui repose sur la minimisation conjointe de deux critères

1. $\| \sum_{j \in S^E} \frac{\nabla I(\bar{\mathbf{u}}_j)}{\|\nabla I(\bar{\mathbf{u}}_j)\|_2} \|_2 = \min_{S^E}$ où S^E désigne un sous-ensemble de E et $\nabla I(\bar{\mathbf{u}}_j)$ le gradient du signal au point $\bar{\mathbf{u}}_j$;

2. $\sum_{j \in S^E} \mathrm{var}(\{I(\bar{\mathbf{u}}^j_{\alpha_t})\}) = \min_{S^E}$ où $\{I(\bar{\mathbf{u}}^j_{\alpha_t})\}$ désigne l'ensemble des niveaux de gris aux points $\bar{\mathbf{u}}^j_{\alpha_t} = \alpha_t \hat{\mathbf{c}} + (1 - \alpha_t)\bar{\mathbf{u}}^j$ à partir de l'estimation initiale $\hat{\mathbf{c}}$ de l'image du centre, $\{\alpha_t\}$ désigne un ensemble de valeurs régulièrement espacées entre 0 et 1, et $\mathrm{var}(\cdot)$ l'opérateur de calcul de variance empirique.

Le premier critère représente la résultante des gradients (normalisés) sur les points de contour. Il vise à assurer une bonne répartition spatiale des pixels $\mathbf{u}_j$ sur l'ellipse externe. Le système est mal conditionné si les points de contour sont très proches les uns des autres. En effet, il est aisé de constater que si les points $\mathbf{u}_j$ sont confondus, alors le problème (5.22) présente une infinité de solutions. Le second critère représente la somme des variances des signaux collectés sur chacune des coupes. Il vise à pénaliser les coupes sur lesquelles le signal est homogène. Les zones non affectées par le flou (notamment le flou de bougé) sont donc privilégiées.



(a)

(b)

FIGURE 5.19: (a) Résultats obtenus lors de la sélection des coupes. Les points verts représentent l'ensemble des points de contour collectés (après un premier sous-échantillonnage). Les bases des flèches bleues situent les points de contour de l'ensemble E_{min} obtenus à l'issue de la sélection des coupes. La direction de chacune des flèches désigne la direction du gradient en ces points. (b) Les segments rouges représentent les coupes associées au résultat de l'image située en bas à gauche de (a).

La minimisation conjointe de ces deux critères est effectuée à partir de sous-ensembles $S_i^E \subset E$ de $N = 10$ points tirés aléatoirement (le nombre minimum est 3). La solution obtenue est considérée comme un sous-ensemble de points « convenables » pour l'optimisation de l'image du centre. Des résultats de sélections de coupes appliquées à des images réelles sont présentés sur la figure 5.19, afin d'illustrer l'intérêt de cette étape.

5.4.6 Identification

Comme nous l'avons mentionné plus haut, un C^{2}Tag peut-être assimilé à un code-barres circulaire. Nous disposons, à ce stade de l'algorithme de détection, d'un ensemble de points de contour sur l'ellipse externe et d'une estimation sous-pixélique de l'image du centre du marqueur. Nous désignons par $\mathcal{B}$ la bibliothèque des C^{2}Tags, c.-à-d. l'ensemble des indices des marqueurs

différents. Chaque modèle de marqueur, d'identifiant $k \in \mathcal{B}$, est associé à une fonction $\mathcal{M}_k$ constante par morceau prenant alternativement les valeurs -1 et 1, où -1 représente la couleur noire et 1 la couleur blanche. Des exemples de fonctions $\mathcal{M}_k$ sont illustrés sur la figure 5.20.

Étant donné un motif $\mathcal{M}_k$, nous définissons la distance entre l'image I du C^{2}Tag et $\mathcal{M}_k$ par

$$D(I, \mathcal{M}^k) = \int_0^1 \frac{1}{2\pi r} \int_0^{2\pi} d\left(I(\mathbf{g}_{\mathsf{H}}^{-1}(r\cos(\theta), r\sin(\theta)), \mathcal{M}_k(r))\right) dr d\theta \tag{5.23}$$

où H est une homographie de rectification métrique centrée, (r, θ) sont les coordonnées polaires d'un point dans le plan de rectification, et où d représente la mesure de dissimilarité décrite dans [Gallo 2011] telle que

$$d(I, -1) = \frac{[\max(I - \mu_b, 0)]^2}{2\sigma^2} \tag{5.24}$$

et

$$d(I, 1) = \frac{[\min(I - \mu_w, 0)]^2}{2\sigma^2} \tag{5.25}$$

où μ_w et μ_b représentent les moyennes des 50% plus grandes et des 50% plus petites valeurs du signal I, respectivement, et où σ^2 désigne la variance de I. Nous n'avons pu trouver de justifications théoriques pour l'utilisation d'une telle mesure de dissimilarité mais celle-ci a présenté dans notre cas de très bonnes performances.

En pratique, la mesure de dissimilarité est calculée de façon discrète. Elle est effectuée sur le sous-ensemble S^E_{min} des points de contour de l'ellipse externe, issus de l'étape de sélection des coupes. Le signal sur chacune des coupes est collecté sur un nombre fini de points associés aux valeurs α_t régulièrement espacées entre 0 et 1 et dont le nombre dépend du niveau d'échantillonnage souhaité. L'équation (5.23) devient alors

$$\bar{D}(\mathcal{I}, \mathcal{M}^k) = \frac{1}{N} \sum_{\alpha_t} \sum_{j \in S^E} d(I(\mathbf{g}_{\mathsf{H}}^{-1}(\alpha_t \mathbf{g}_{\mathsf{H}}(\bar{\mathbf{u}}_j))), \mathcal{M}_k(r)) \tag{5.26}$$

Une étape préliminaire peut consister à calculer une position sous-pixélique des points de contour $\mathbf{u}_j$ comme cela est décrit dans [Okumura 2006].

Nous définissons ainsi, par analogie avec les travaux présentés dans [Gallo 2011], la probabilité que l'image I du C^{2}Tag soit l'image du marqueur k par

$$p_k(I) \propto e^{-\bar{D}(I, \mathcal{M}_k)} \tag{5.27}$$

L'identifiant donné à une image $\mathcal{I}$ correspond ainsi à l'indice $k_{max} = \arg\max_k p_k(\mathcal{I})$ qui maximise cette probabilité.

Dans cette partie, les C^{2}Tags sont assimilés à des codes-barres circulaires pour lesquels les modèles sont définis de manière générique par un ensemble de fonctions $\mathcal{M}_k$. Les utilisateurs du système sont susceptibles de définir leurs propres fonctions $\mathcal{M}_k$. Il est ainsi possible d'envisager l'utilisation de modèles standard de type EAN ou UPC. L'élaboration d'un jeu de modèles de marqueurs permettant de maximiser la distance inter-marqueurs (par exemple la distance de Hamming [Hamming 1950]) constitue une des perspectives du travail réalisé.

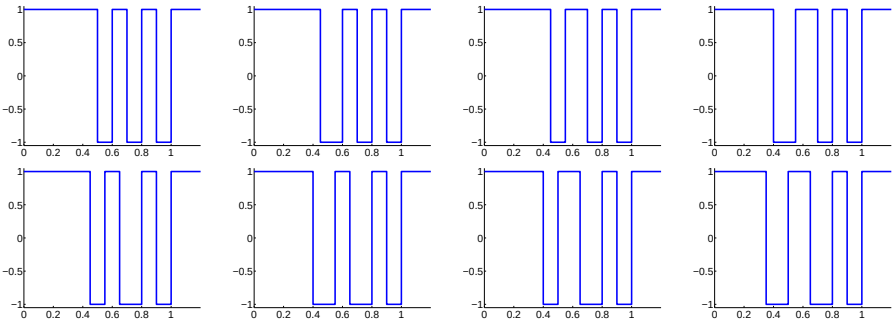


FIGURE 5.20: Fonctions $\mathcal{M}_k$ associées aux 8 premiers modèles de C^{2}Tags à 3 couronnes utilisés dans nos expérimentations.

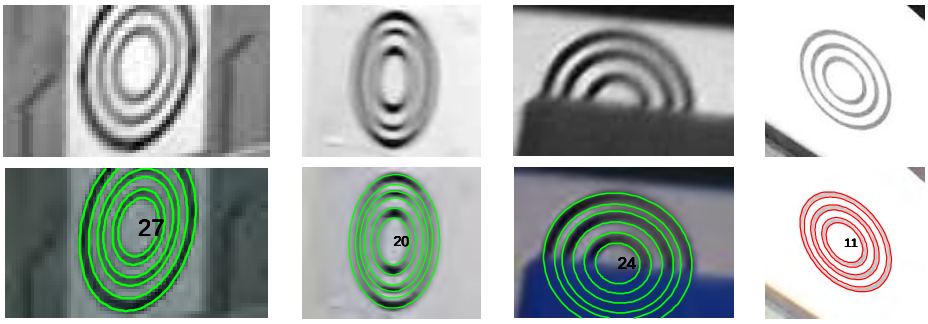


FIGURE 5.21: (a) (b) (c) Exemples d'identifications correctes. (d) Exemple d'une mauvaise identification (les couronnes sont «rognées» à cause de la surexposition).

5.5 Résultats expérimentaux

Nous décrivons dans cette partie l'ensemble des expérimentations ayant permis l'évaluation des performances de notre algorithme de détection sur images de synthèse et sur images réelles. L'algorithme de détection effectue ici une analyse des images sur trois échelles.

5.5.1 Résultats sur données de synthèse

5.5.1.1 Étude de validité de l'approximation des lignes de champ

Afin de valider l'approximation des lignes de champ définie dans la section 5.4.4, des simulations ont été effectuées sur 100 jeux de données pour lesquels nous avons exprimé le pourcentage d'intentions de vote ayant abouti, c.-à-d. le rapport entre le nombre d'intentions de vote émises par les points de contour de l'ellipse externe (image du cercle de rayon 1) et le nombre d'intentions de vote ayant abouti, c.-à-d. ayant intersecté l'ellipse interne (image du cercle de rayon 0.25). Les résultats sont exprimés en faisant varier le nombre de cercles (de rayons régulièrement espacés entre 1 et 0.25) constituant le marqueur, et l'angle entre le plan du marqueur et le plan image. L'ensemble des résultats est présenté sur la figure 5.22.

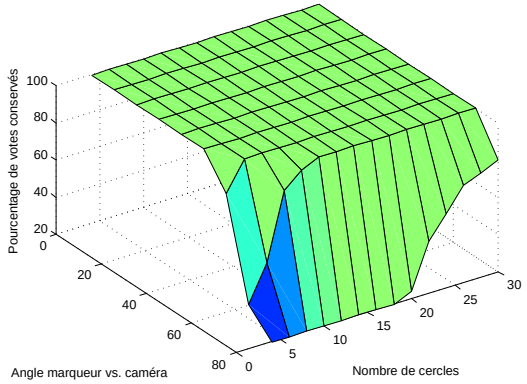


FIGURE 5.22: Étude de validité de l'approximation des lignes de champ selon le système de vote proposé. Pourcentage d'intentions de vote ayant abouti. Les intentions de vote sont émises par les points de contour de l'ellipse externe (image du cercle de rayon 1). L'ellipse interne est l'image du cercle de rayon 0.25.

La figure 5.22 nous permet de faire les deux constats suivants : (i) les approximations des lignes de champ obtenues à partir d'un nombre de cercles strictement inférieur à 6, c.-à-d. pour des marqueurs composés d'une ou deux couronnes, est inenvisageable puisqu'il apparaît clairement que la quantité d'intentions de vote atteignant l'ellipse interne diminue significativement au-delà d'un angle supérieur à 60° ; (ii) de manière générale, cette même approximation n'est pas valable au-delà d'un angle de 70° entre le plan image et le plan de support. Ceci est dû au fait que les ellipses présentent alors des zones de très fortes courbures, qui entraînent la divergence de certaines intentions de vote (cf. figure 5.9(d)).

5.5.1.2 Résultats de l'algorithme de détection

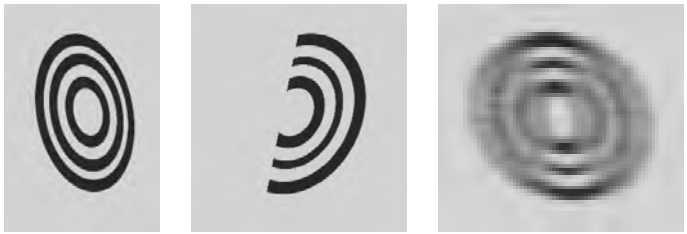
Nous avons utilisé pour nos expérimentations des marqueurs composés de 3 couronnes, la largeur de chacune des couronnes blanche et noire pouvant être égale à 0.1 ou 0.15 unité (et où le cercle externe est, par convention, de rayon unité). Nous avons donc évalué un jeu de 32 marqueurs. Les résultats présentés dans ce qui suit concernent le taux de faux négatifs suivant la distance entre la caméra et le marqueur, l'angle entre le plan des pixels et le plan de support du marqueur et enfin le flou de bougé. Nous présentons également le taux de confusion inter-marqueurs en faisant varier l'ensemble de ces paramètres. Les images des marqueurs sont synthétisées à partir de caméras dont les paramètres associés à leur position relativement à celle du marqueur sont mentionnés pour chacune des expérimentations. Nous désignons par D la distance entre la caméra et le marqueur, par θ l'angle (en degrés) formé entre le plan image et le plan de support du marqueur et par l la longueur (en pixels) du flou de bougé (unidimensionnel) appliqué à l'image de synthèse exacte dans une direction aléatoire. Si une ou plusieurs valeurs parmi ces trois paramètres ne sont pas mentionnées, cela signifie alors qu'elles sont prises aléatoirement dans les intervalles (par défaut) suivants : $D \in [10, 50]$, $\theta \in [0, 60]$ et $l \in [0, 10]$. Les caméras synthétisées possèdent des pixels carrés, leur point principal est centré à l'origine du repère pixellique. Le plan image a pour dimensions 1280×720 pixels et la distance focale de la caméra vaut 1000 pixels. Les images sont générées en niveau de gris pour des valeurs comprises entre 0 et 255 et un bruit gaussien d'écart-type 5 est appliqué à chaque pixel. Par ailleurs, un flou de mise au point est appliqué à l'ensemble des images via l'application d'un filtre gaussien d'écart type 0.5 pixel.

L'ensemble des résultats obtenus est présenté sur la figure 5.24. La lecture de cette dernière peut être effectuée conjointement à la celle de la table 5.24 qui renseigne le lecteur sur les dimensions de l'image du marqueur à détecter dans l'image synthétisée. Dans ce qui suit, nous évaluons le taux de faux négatifs, noté τ_n, et le taux de confusion inter-marqueurs, noté τ_c, en faisant varier les paramètres suivants :

La distance et l'orientation. La première expérience consiste à évaluer l'algorithme de détection en fonction de la distance entre la caméra et le marqueur. Cette évaluation est effectuée en faisant également varier l'angle entre le plan de support du marqueur et le plan image. Les résultats sont présentés sur la figure 5.24(a). Nous constatons une baisse de performances du détecteur au-delà d'une distance égale à 50 unités pour un angle allant jusqu'à 20°, puis au-delà d'une distance de 40 unités pour un angle supérieur à 40°. L'algorithme présente de mauvaises performances pour une inclinaison de 80°, ce qui se justifie essentiellement par les résultats obtenus dans la section 5.5.1.1 concernant la validité de l'approximation des lignes de champ pour le protocole de vote.

La superficie occultée. Le deuxième expérience consiste à évaluer l'algorithme de détection en fonction de la superficie occultée du marqueur. Le taux de faux négatifs est exprimé en fonction du pourcentage de superficie occultée par rapport à la superficie totale du marqueur. Le « type » d'occultation appliquée aux marqueurs est illustré sur la figure 5.23(b). Les résultats sont présentés sur la figure 5.24(b). L'algorithme de détection présente une importante baisse de performances

pour des occultations supérieures à 50% à une distance de 20 unités, et supérieures à 40% pour une distance supérieure à 30 unités. À une distance de 50 unités, seules les images de C^2Tags occultés à moins de 20% sont en général détectées.



(a) (b) (c)

FIGURE 5.23: Exemples d'images de synthèse pour l'évaluation de l'algorithme de détection. (a) Images synthétisées des C^2Tags respectivement (a) non occultée et non floue, (b) occultée (à 40%) et (c) floue (où $D = 40$ et $l = 10$ pixels).

Le flou de bougé. Cette dernière expérience vise à évaluer l'algorithme de détection et d'identification en présence de flou de bougé. Un exemple d'une image de synthèse d'un C^2Tag dégradée par du flou de bougé est illustrée sur la figure 5.23(c). À la différence des expériences précédentes, nous présentons également les performances liées à l'identification des marqueurs car il s'agit du cas présentant le plus de difficultés pour l'identification des marqueurs (par rapport aux deux premières expériences). Les résultats sont présentés sur la figure 5.24(c) pour le taux τ_n de faux négatifs et sur la figure 5.24(d) pour le taux τ_c de confusion inter-marqueurs. L'algorithme de détection présente de bonnes performances en présence de flou de bougé, tant pour la détection que pour l'identification. Pour une distance $D = 40$, on note une baisse de performances à partir de $l = 4$ pixels mais le taux de réussite reste supérieur à 60% jusqu'à $l = 10$ pixels. L'algorithme d'identification présente une légère baisse de performances au-delà d'une distance de 40 unités.

$\theta \backslash D$	10	20	30	40	50	60
0	100.0	50	33.3	25.0	20.0	16.7
20	97.2	48.5	32.3	24.3	19.4	16.2
40	88.4	44.2	29.5	22.1	17.6	14.7
60	75.3	37.6	25.0	18.7	15	12.5
80	59.1	29.5	19.6	14.7	11.7	9.8

TABLE 5.3: Dimensions moyennes (en pixels) des demi-axes des ellipses externes des images des marqueurs synthétisées. Les angles sont exprimés en degrés.

5.5.2 Résultats sur données réelles

Dans cette partie, nous présentons des résultats obtenus sur images réelles (photographies ou images extraites de séquences vidéo). Nous présentons dans un premier temps le taux de faux positifs obtenus sur quatre séquences vidéo. Enfin, nous illustrons quelques résultats fournis par l'algorithme de détection sur les figures 5.26 et 5.27.

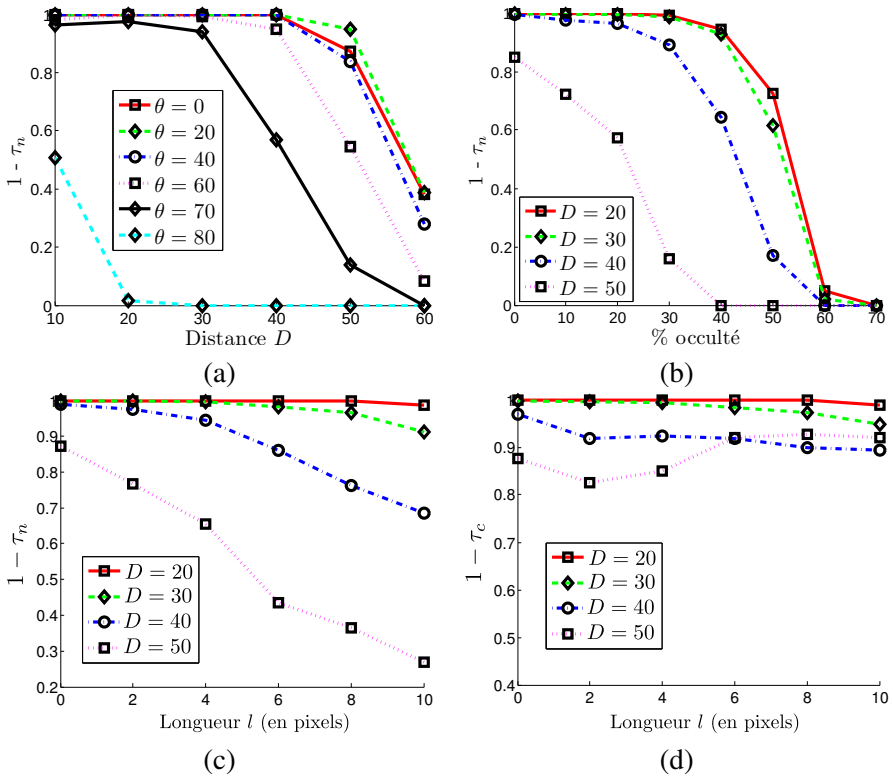


FIGURE 5.24: Résultats fournis par l'algorithme de détection exécuté sur des images de synthèse. Les résultats correspondent, pour chaque configuration, à une moyenne sur 500 images de synthèse. (a) Résultats obtenus en fonction de la distance D entre la caméra et le marqueur. L'expérience est renouvelée en faisant varier l'angle θ entre le plan image et le plan de support du marqueur. (b) Résultats obtenus en faisant varier la superficie du marqueur occultée (pourcentage occulté par rapport à la superficie totale). (c) et (d) représentent les résultats obtenus en faisant varier la longueur du flou de bougé (unidimensionnel) de 0 à 10 pixels. L'expérience est renouvelée en faisant varier la distance D. Les résultats sur le taux de faux négatifs et le taux de confusion inter-marqueurs sont illustrés en (c) et (d) respectivement.

Taux de faux positifs. L'évaluation du taux de faux positifs a été effectuée sur quatre séquences vidéo réelles désignées par les lettres A, B, C et D. L'ensemble des résultats est présenté dans la table 5.4. Bien qu'il soit relativement faible, le taux de faux positifs n'est pas nul. Il serait toutefois possible de définir un compromis entre le taux de faux positifs et le taux de faux négatifs via un jeu de paramètres pour l'algorithme de détection choisi suivant les performances attendues. Aucune étude sur le choix de ces paramètres n'a été menée et ceci constitue une des perspectives du travail réalisé.

Séquence	A	B	C	D	Total
Nb. images	1300	773	470	1287	3830
Nb. faux positifs	7	38	0	10	55
Taux faux positifs	0.005	0.049	0	0.008	0.014

TABLE 5.4: Résultats de l'algorithme de détection sur le taux de faux positifs. Quatre séquences vidéo ne contenant aucune image de C^{2}Tag ont été traitées par l'algorithme de détection. Le tableau illustre le nombre de fausses détections obtenues.

Temps de calcul. Le temps de calcul a été mesuré sur une séquence vidéo de 308 images de dimension 640×480 pixels pour laquelle deux marqueurs sont visibles dans toutes les images. En moyenne, le temps d'exécution total par image est de 132 *ms* soit environ un traitement de 7.6 images par seconde. Le temps moyen d'exécution de l'étape de localisation des motifs est de 39 *ms* et celui des étapes d'optimisation des images des centres et d'identification est de 93 *ms*. Il apparaît clairement que ces deux dernières étapes sont les plus coûteuses. Bien que des performances en temps réel ne soient pas atteintes, l'utilisateur peut, s'il le souhaite, réduire le niveau de robustesse de l'algorithme de détection à partir d'un jeu de paramètres adapté, au profit d'une réduction du temps de calcul. Dans nos expérimentations, nous avons en effet fait le choix de mettre l'accent sur les propriétés de robustesse du détecteur, au prix d'un temps de calcul élevé. Certaines optimisations du code implanté sont possibles.

5.6 Conclusion

Dans ce chapitre, nous avons présenté un système de marqueurs constitués de couronnes circulaires concentriques nommés C^2Tag. Les motivations quant à l'utilisation de ce type de marqueurs ont été exposées. Nous avons montré que le C^2Tag présentait de très bonnes propriétés tant pour la localisation de son image que pour son identification et ce, même en présence d'occultation et de flou de bougé. Des expérimentations sur images de synthèse et sur images réelles ont permis de valider la méthode proposée.

Perspectives. Parmi l'ensemble des étapes de l'algorithme de détection, l'étape de sélection des coupes pour l'optimisation de l'image du centre doit être améliorée. En effet, celle-ci présente des résultats décevants et sa complexité constitue une des entraves à des performances en temps réel.

Concernant la localisation du motif et plus précisément la construction des intentions de vote, approximations des lignes de champ, il serait intéressant, comme nous l'avons mentionné dans la section 5.4.4, d'utiliser des règles de décision issues d'une classification supervisée afin de réduire, d'une part, la complexité de l'algorithme de vote et, d'autre part, le nombre de faux positifs traités dans les étapes suivantes.

Une analyse des signaux le long des lignes de champ (convergentes) pourrait être utilisée afin de regrouper les points de contour sur l'ellipse externe potentiellement associées au même identifiant, et ainsi d'assurer une plus grande robustesse de l'estimation de l'ellipse externe.

Annexes

Algorithme « Least Median of Squares »

$P \leftarrow \mathrm{diag}(1, 1, 0)$
$\mathsf{Q}_m \leftarrow 0$
$S_m \leftarrow \infty$
$\{\mathbf{x}_\alpha\} \leftarrow$ 5 points pris aléatoirement dans E
$S \leftarrow \mathrm{med}_{\alpha=1}^{N} \frac{\mathbf{x}_\alpha^T \mathsf{Q} \mathbf{x}_\alpha}{\|\mathsf{P}\mathsf{Q}\mathbf{x}_\alpha\|}$
$n \leftarrow 0$
(i) **Tant que** $n < 10$
  $\mathsf{Q} \leftarrow \mathrm{calculerEllipse}(\{\mathbf{x}_\alpha\})$
  Si Q n'est pas une ellipse
    retourner à (i)
  Sinon
    $S \leftarrow \mathrm{med}_{\alpha=1}^{N} \frac{\mathbf{x}_\alpha^T \mathsf{Q} \mathbf{x}_\alpha}{\|\mathsf{P}\mathsf{Q}\mathbf{x}_\alpha\|}$
  Fin Si
  Si $S < S_m$
    $\mathsf{Q}_m \leftarrow \mathsf{Q}$
    $S_m \leftarrow S$
    $n \leftarrow 0$
  Sinon
    $n \leftarrow n + 1$
  Fin Si
Fin Tant que

TABLE 5.5: Algorithme « Least Median of Squares » pour l'estimation robuste d'ellipse présenté dans [Kanatani 2004]. La fonction $\mathrm{calculerEllipse}(\{\mathbf{x}_\alpha\})$ calcule l'ellipse passant par les 5 points de l'ensemble $\{\mathbf{x}_\alpha\}$.

Algorithme d'expansion d'ellipse

Nous rappelons ici comment sont calculées l'expansion $\mathsf{Q}^{(+)}$ et la contraction $\mathsf{Q}^{(-)}$, de δ pixels, de l'ellipse représentée par la matrice Q (décrit dans [Kanatani 2004]).

La matrice Q s'écrit

$$\mathsf{Q} = \begin{bmatrix} A & B & C \\ D & C & E \\ C & E & F \end{bmatrix}. \tag{5.28}$$

et

$$\mathsf{S} = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \quad \text{et} \quad \mathbf{c} = \begin{bmatrix} D \\ E \end{bmatrix} \tag{5.29}$$

Les deux ellipses $\mathsf{Q}^{(+)}$ et $\mathsf{Q}^{(-)}$ délimitant l'enveloppe elliptique s'écrivent sous la forme

$$\mathsf{Q}^{(\pm)} = \begin{bmatrix} \mathsf{S}^{(\pm)} & \mathsf{S}^{(\pm)}\mathsf{S}^{-1}\mathbf{c} \\ (\mathsf{S}^{(\pm)}\mathsf{S}^{-1}\mathbf{c})^\top & \mathbf{c}^\top \mathsf{S}^{-1}(\mathsf{S}^{(\pm)} - \mathsf{S})\mathsf{S}^{-1}\mathbf{c}) \end{bmatrix} \tag{5.30}$$

où

$$\mathsf{S}^{(\pm)} = \mathsf{U}\,\mathrm{diag}(\lambda_1^{(\pm)}, \lambda_2^{(\pm)})\mathsf{U}^\top$$

où U désigne la matrice orthogonale d'ordre 2 composée des vecteurs propres de S,

$$\lambda_1^{(\pm)} = \frac{c}{\sqrt{c/\lambda_1} \pm \delta^2} \quad \text{et} \quad \lambda_2^{(\pm)} = \frac{c}{\sqrt{c/\lambda_2} \pm \delta^2}.$$

Un point, représenté par le vecteur $\mathbf{x}$ de ses coordonnées cartésiennes augmentées, vérifiant l'inégalité

$$(\mathbf{x}^\top \mathsf{Q}^{(+)}\mathbf{x})(\mathbf{x}^\top \mathsf{Q}^{(-)}\mathbf{x}) < 0 \tag{5.31}$$

se situe à l'intérieur de la région délimitée par les ellipses $\mathsf{Q}^{(+)}$ et $\mathsf{Q}^{(-)}$.

5.6.1 Calcul de la circonférence de l'ellipse dans l'image de contours

L'algorithme d'extraction de contour fournit des contours d'épaisseur 1 pixel. La circonférence (en pixels) d'une ellipse définie par un ensemble de points de contour, c.-à-d. le nombre de pixels qui composent cet ensemble, peut ainsi être calculée comme le nombre de pixels « allumés » par un algorithme de tracé d'une ellipse, fondé sur le principe de [Foley 1990]. Il suffit simplement de calculer les trois points distincts A, B et C de l'ellipse Q pour lesquels

$$\frac{dx_A}{dy_A} = \frac{dx_B}{dy_B} = 1 \text{ et } \frac{dx_C}{dy_C} = -1.$$

Le nombre de pixels qui composent l'ellipse est donné par (cf. figure 5.25)

$$L_{\mathcal{C}} = 2\max(x_A - x_C, y_A - y_C) + 2\max(x_B - x_C, y_B - y_C).$$

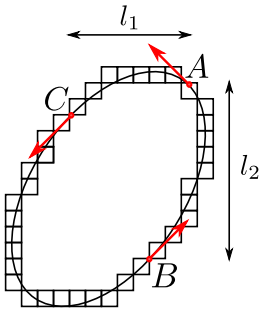


FIGURE 5.25: Illustration du calcul du périmètre d'une ellipse d'épaisseur 1 pixel. Les points rouges A, B et C représentent les points de l'ellipse tels que $\frac{dx_A}{dy_A} = \frac{dx_B}{dy_B} = 1$, $\frac{dx_C}{dy_C} = -1$. La longueur $2(l_1 + l_2)$ représente le périmètre de l'ellipse en pixels.

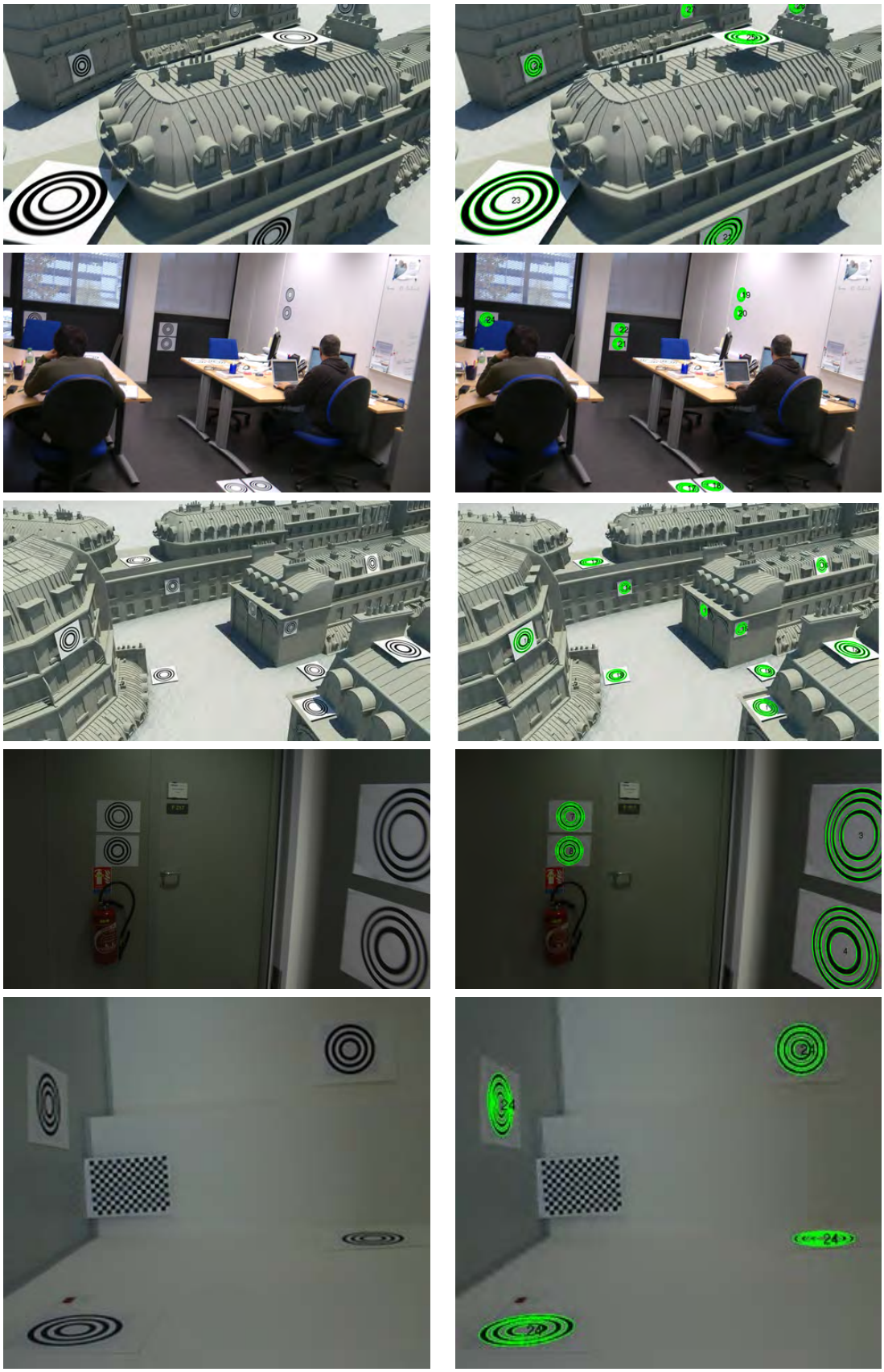


FIGURE 5.26: Quelques résultats de l'algorithme de détection sur images réelles.

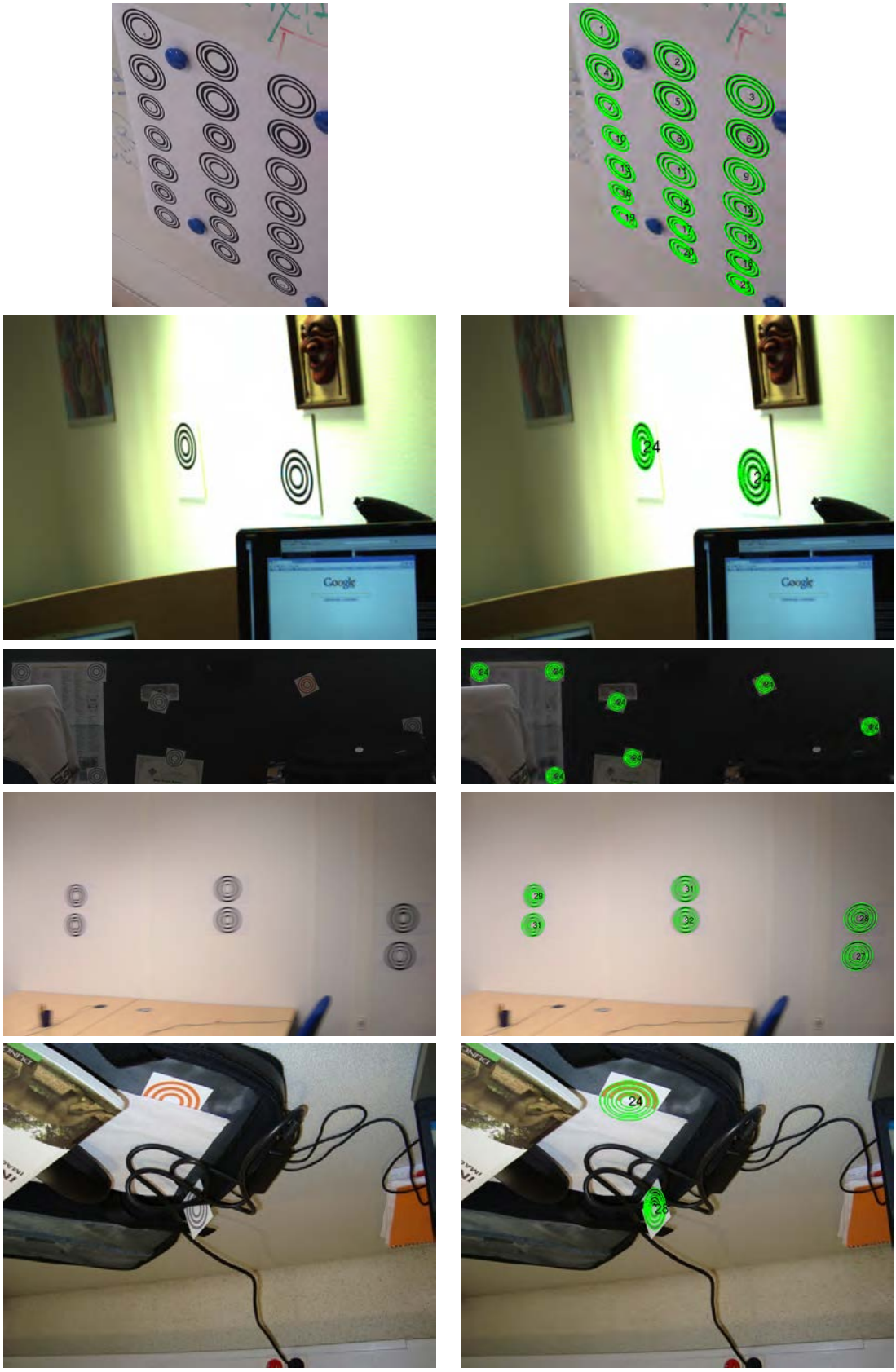


FIGURE 5.27: Quelques résultats de l'algorithme de détection sur images réelles.

Chapitre 6

Application au suivi de caméra

Nous présentons dans ce chapitre une méthode de suivi automatique de caméra reposant sur le système de marqueurs C^2Tags. Nous donnons le paradigme général du suivi de caméra hors ligne à partir des C^2Tags ainsi qu'une description détaillée des éléments qui le composent. Nous présentons ensuite une méthode de suivi de caméra en ligne reposant sur une base de connaissances 3D de C^2Tags. Nous fournissons par ailleurs un paramétrage 3D minimal d'une scène composée de C^2Tags pour l'ajustement de faisceaux.

6.1 Introduction

Le ***problème du suivi automatique*** d'une *caméra non calibrée*, c.-à-d. d'une caméra dont les paramètres internes sont inconnus, consiste à calculer à partir d'une séquence d'images, en plus de ses paramètres internes, les paramètres externes donnant la position et l'orientation de la caméra, pour *chacune* des vues. L'automatisation du suivi constitue une des principales difficultés du problème. Il existe deux grands types d'approches de suivi de caméra qui se distinguent par la nature de leurs données d'entrée.

Les approches *sans modèle* ou *basées points d'intérêt* [Fitzgibbon 2003] utilisent comme seules données les points image extraits et suivis le long de la séquence vidéo. Elles se décomposent généralement en quatre étapes principales :

- extraction des points d'intérêt dans chaque image,
- mise en correspondance robuste des points d'intérêt entre images voisines,
- reconstruction projective des points d'intérêt appariés,
- reconstruction euclidienne (autocalibrage).

Ces approches présentent l'avantage d'être totalement flexibles mais ne présentent cependant aucune garantie, en ce qui concerne les performances [Torr 1999, Gurdjos 2009].

Les approches *basées modèle* reposent sur la connaissance d'un modèle 3D partiel de la scène et sur la mise en correspondance d'un certain nombre de primitives de ce modèle avec leurs projections dans l'image. Nous pouvons citer, en particulier, l'ensemble des approches reposant sur des systèmes de marqueurs pour lesquelles le modèle 3D, supposé connu, correspond au modèle du marqueur introduit dans la scène et, éventuellement, lorsque plusieurs marqueurs sont utilisés, aux positions relatives de ces derniers. Ces méthodes présentent quant à elles de bonnes garanties de performances et sont, de par la faible complexité de leurs algorithmes sous-jacents, bien adaptées à des applications de suivi de caméra pour lesquelles les contraintes temporelles sont élevées (par exemple pour des applications de réalité augmentée) et/ou pour lesquelles des garanties de performances sont requises. Néanmoins, elles nécessitent toutefois l'introduction dans la scène d'un équipement particulier, ce qui peut présenter un inconvénient majeur pour certaines applications.

6.1.1 Contexte - le projet ANR ROM

Nous nous sommes intéressés à ce deuxième type d'approches dans le cadre d'un projet de recherche à visée industrielle, dont l'acronyme est *ROM* (*Real time On-set Matchmoving*), pour un certain nombre de raisons que nous présentons dans ce qui suit. Il s'agissait d'un projet de recherche concernant la création d'effets spéciaux et, plus spécifiquement, la prévisualisation de ces effets pendant le tournage. Les techniques de composition numérique (*compositing*), mélangeant images réelles et images de synthèse, ont bénéficié de l'apparition de logiciels dédiés très performants issus de la recherche en vision par ordinateur, en particulier pour le suivi de caméra. Cependant, de telles techniques restent réservées à la phase de post-production, c.-à-d. à une utilisation différée par rapport au tournage. Lors des tournages, il est fréquent que des séquences entières soient filmées sur un fond vert ou bleu. C'est souvent la compétence du superviseur d'effets spéciaux qui, seule, permet d'assurer que le tournage est effectué conformément

aux contraintes de la post-production. Ce procédé reste toutefois délicat et peut être sujet à de nombreuses imperfections. L'objectif du projet ROM était d'effacer les frontières entre la production et la post-production en concevant un système de prévisualisation des effets spéciaux qui, lors du tournage, fournira quasi instantanément au réalisateur une visualisation approchée du résultat final de la composition numérique.

Le verrou technologique majeur du projet était la résolution du problème du suivi de caméra, en donnant la priorité au respect des contraintes temporelles (c.-à-d. en temps réel). L'approche originale du projet a été fondée sur les trois éléments clés suivants :

- Diversité des scènes de tournage (en intérieur, en extérieur, en studio, etc.) : cela implique nécessairement une phase de préparation à l'issue de laquelle la scène pourra être équipée, c.-à-d. enrichie de marqueurs afin de disposer des meilleures garanties de performances de l'algorithme de suivi de caméra.
- Étape de préparation : elle doit permettre la constitution d'une *base de connaissances 3D* de marqueurs, c.-à-d. d'une base de connaissances contenant les positions 3D relatives des marqueurs disposés dans la scène, dont un sous-ensemble pourra être suivi correctement et en temps réel.
- Prévisualisation en temps réel : le jour du tournage, le fonctionnement attendu consiste à utiliser cette base de connaissances. Si les exigences du tournage impliquent la suppression de certains marqueurs introduits lors de l'étape de préparation, le mode de fonctionnement devra être aussi peu dégradé que possible.

La solution proposée, répondant à l'approche originale présentée ci-dessus, repose sur la conception d'une méthode de suivi de caméra *hors ligne* pour l'étape de préparation et sur celle d'un suivi de caméra *en ligne* pour l'étape de prévisualisation. Ces deux méthodes de suivi de caméra reposent sur l'utilisation du système de marqueurs C^2Tags.

Suivis « hors ligne » et « en ligne » d'une caméra. Nous désignons par « *suivi de caméra hors ligne* » un suivi de caméra reposant sur le traitement de l'intégralité de la séquence vidéo traitée, qui ne peut donc être effectué qu'une fois l'acquisition de celle-ci terminée. Il se différencie du *« suivi de caméra en ligne »* (par exemple effectué en temps réel) pour lequel le calcul de la pose de la caméra est effectué « à la volée », c.-à-d. à partir de la donnée d'une seule vue, en général couplée à des informations *a priori* sur la scène filmée telles qu'une connaissance partielle de sa structure tridimensionnelle (pour faire référence au suivi basé modèle). L'approche proposée est résumée sur la figure 6.1.

Partenaires. Le projet ROM était un projet de recherche à visée industrielle soutenu par l'Agence Nationale de la Recherche (ANR) impliquant les laboratoires INRIA Rhône-Alpes (Grenoble) et IRIT-ENSEEIHT (Toulouse), ainsi que la société de post-production DURAN DUBOI (Issy-les-Moulineaux). J'ai participé à ce projet tout au long de son déroulement, à savoir mes deux premières années de doctorat.

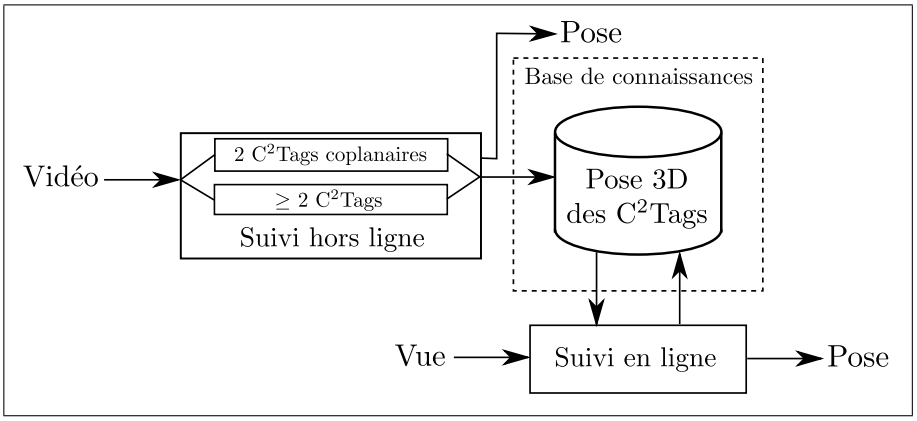


FIGURE 6.1: Diagramme de flux de la solution de suivi automatique de caméra proposée.

6.1.2 État de l'art

Nous présentons dans ce qui suit un état de l'art sur les méthodes de suivi de caméra en lien avec les enjeux du projet ROM. Il se décompose en quatre parties :

1. Algorithmes de suivi de caméra basés modèle.
2. Algorithmes sans modèle.
3. Algorithmes et logiciels utilisés en post-production.
4. Algorithmes utilisant d'autres capteurs.

1) Pour suivre, en temps réel, le mouvement d'une caméra se déplaçant dans une scène, les algorithmes les plus efficaces sont souvent basés sur de nombreuses connaissances *a priori*. Les connaissances introduites peuvent concerner la scène, le mouvement relatif de la scène par rapport au capteur, le modèle géométrique de la caméra, les contraintes d'éclairage, etc. Parmi les premiers algorithmes de suivi de caméra fonctionnant en temps réel, certains reposaient sur la connaissance d'un modèle CAO (Conception Assistée par Ordinateur) de l'objet à suivre [Lowe 1992]. Dix ans après, les travaux de Drummond forment un aboutissement du suivi basé modèle avec une approche basée sur les contours [Drummond 2002]. Des informations de texture (ou d'apparence) peuvent également être associées de différentes manières au modèle. Il est possible de mettre en correspondance les textures des images et du modèle [Pressigout 2007, Xiao 2004, Dehais 2010] ou de s'appuyer sur des correspondances de points d'intérêt contraintes par le modèle [Vacchetti 2004, Simon 2002]. L'utilisation d'une approche supervisée à travers des vues-clés pour lesquelles la solution du problème est connue est considérée comme une des techniques les plus efficaces [Vacchetti 2004]. Pour les approches précédentes, le modèle de l'objet considéré contient des informations topologiques (relations de connectivité entre les facettes par exemple). Il existe une autre catégorie de modèle sans topologie reposant typiquement sur des collections de *patches* texturés [Wiles 2001, Muñoz 2005, Rothganger 2006]. Plus récemment, des solutions de suivi de caméra en temps réel reposant sur un schéma de localisation global [Se 2005, Skrypnyk 2004] ont été proposées. Ces méthodes induisent une étape hors ligne de reconstruction 3D de l'environnement, effectuée à partir d'un ensemble de vues désignées par le

terme de *vues-clés*. De telles reconstructions sont obtenues via l'utilisation d'algorithmes de calcul de la structure et du mouvement tels que, par exemple, [Zhang 2007, Agarwal 2009]. Parmi ces méthodes de suivi de caméra, celles présentées dans [Konolige 2010, Dong 2009, Irschara 2009] reposent sur l'utilisation d'un *arbre de vocabulaire*, décrit dans [Nistér 2006], dont les éléments sont des descripteurs de points d'intérêt (en général SIFT [Lowe 2004]). Ce type d'approche consiste à rechercher, à partir d'une image *requête*, les images « les plus proches » au sein d'une base de données d'images ou de séquences vidéo. Dans le cadre du suivi de caméra, la base de données est composée de l'ensemble des vues-clés utilisées lors du suivi de caméra hors ligne. Lors du suivi de caméra en ligne, la vue courante est l'image requête et il s'agit de rechercher dans cette base de donnée la vue-clé « la plus proche » . Le calcul de la pose de la caméra est ensuite rendu possible par la mise en correspondance des points d'intérêt de la vue courante avec ceux de la vue-clé sélectionnée « la plus proche » , dont les positions 3D associés sont connues.

Dans les approches citées, l'extraction et la mise en correspondance des points d'intérêt peuvent constituer un obstacle à l'obtention de performances temps réel. D'autre part, celles-ci supposent de manière générale que l'on dispose de points d'intérêt identifiables de manière unique, une hypothèse pas toujours vérifiée comme par exemple pour des scènes constituées de motifs répétitifs. Afin de pallier à ces problèmes, il existe, parmi les méthodes de suivi basé modèle, des approches reposant sur des systèmes de marqueurs. Le lecteur trouvera un état de l'art détaillé sur les systèmes de marqueurs dans la section 5.2.

2) Nous livrons ici un passage en revue succint des algorithmes de suivi de caméra temps réel ou quasi temps réel sans modèle. Le suivi de caméra sans modèle *a priori* de la scène a toujours été un des enjeux de la vision par ordinateur. Il existe différents paradigmes, qui s'appuient tous sur les mêmes fondements, mais qui effectuent les calculs de différentes manières. Ces fondements sont la géométrie de la formation des images et la géométrie des images multiples. Les grands types d'approches sont la localisation et la cartographie simultanées (*Simultaneous Localization And Mapping*, ou SLAM) et le calcul de la structure à partir du mouvement (*Structure-from-Motion*, ou SFM). Les deux approches ont pour but de reconstruire le mouvement d'une caméra et la structure 3D de la scène observée. Tandis que le SLAM, à cause de son utilisation habituelle dans des applications en robotique mobile, vise explicitement le suivi de caméra en temps réel, la plupart des travaux en SFM ne prennent pas en compte la contrainte du temps réel. Par conséquent, les outils et les objectifs sont différents. Il s'agit d'une estimation incrémentale avec propagation de l'incertitude pour le SLAM [Thrun 2002], basée sur le filtre de Kalman, et d'une estimation hors ligne de la structure et du mouvement pour le SFM. En SFM, l'accent est souvent mis sur l'obtention d'estimations initiales avec un minimum de données ; la gestion des incertitudes est typiquement effectuée lors d'une étape d'optimisation finale (ajustement de faisceaux) prenant en compte toutes les données simultanément. Malgré ces différences conceptuelles, les frontières entre SLAM et SFM s'estompent actuellement. Les premières méthodes performantes de SLAM monoculaire en temps réel ont vu le jour il y a quelques années [Nistér 2004, Davison 2003]. Depuis, grâce à des puissances de calcul croissantes et aux améliorations de ces algorithmes, il est devenu possible de rajouter des fonctionnalités tout en gardant l'aspect temps réel. Ceci a notamment permis d'effectuer, en temps réel, des ajustements de faisceaux locaux sur un sous-ensemble

de vues de la séquence vidéo, ce qui permet une propagation moins approximative des incertitudes que dans le SLAM traditionnel, et accroît de manière significative la stabilité et la précision des résultats [Lhuillier 2008, Mouragnon 2009, Davison 2007, Klein 2007]. Plus récemment, les auteurs de [Newcombe 2011] ont proposé une solution très stable reposant sur une reconstruction dense effectuée en temps réel (utilisant des processeurs graphiques), y compris lors de mouvements rapides de la caméra.

3) Il existe de très nombreux produits, commerciaux ou non, de suivi de caméra plus ou moins automatiques et souvent appelés logiciels de *matchmoving*. Parmi eux, Voodoo [Voo] est une solution non commerciale, PixelFarm [Pix] est quant à elle la réincarnation commerciale d'Icarus. RealViz Match Mover [Rea] et Boujou [Bou] sont, à notre connaissance, les produits commerciaux les plus utilisées car ils sont conçus dans l'optique de proposer une solution automatique. Par contre, ces logiciels de post-production ne répondent pas aux exigences de temps réel ou du quasi temps réel posées par le problème de la prévisualisation. D'autres solutions existent telles que 3DEqualizer [3DE], Ssontech SynthEyes [Syn] ou encore Simi Reality Motion Systems [Sim]. Une solution logicielle académique de suivi de caméra, nommée Acts, est disponible gratuitement [Act] et repose sur la méthode de SFM présentée dans [Zhang 2007]. Enfin, nous pouvons mentionner le logiciel de modélisation Blender [Ble] qui, dans sa dernière version, intègre un module de suivi de caméra. Ce module repose sur la bibliothèque LibMV [LMV] qui est sur le point d'être intégrée dans OpenCV [OCV], une des bibliothèques de référence de la communauté de recherche en vision par ordinateur.

4) Les capteurs de position optiques fonctionnent avec des caméras et des marqueurs actifs ou passifs. Les marqueurs passifs (*retroreflexive markers*) sont conçus pour « briller » alors que les marqueurs actifs utilisent souvent des LED infrarouges. La position 3D des cibles relativement à la caméra peut être obtenue par triangulation. Il existe deux façons classiques de procéder. Soit les caméras qui observent les cibles sont fixes et ce sont alors les objets en mouvement que l'on veut suivre qui sont équipés avec des marqueurs [Brandner 2003]. Soit, à l'inverse, la scène est équipée de marqueurs fixes et c'est la caméra observant les cibles qui est suivie. Il existe des produits commerciaux issus des approches photogrammétriques tels que [Vic], Advanced Realtime Tracking [ART] et AICON3D [AIC]. D'autres approches de suivi de caméra utilisent des capteurs inertiels, le GPS, la Wiimote ou des Z-cam. Cependant, comme mentionné dans [Corke 2007], les capteurs seuls sont en général trop peu précis comme par exemple les capteurs inertiels qui fournissent des mesures bruitées lorsque le mouvement est lent ou encore les magnétomètres qui peuvent être affectés par la présence d'interférences magnétiques. L'utilisation du GPS ou d'une combinaison GPS/Centrale inertielle/Vision [Reitmayr 2007, Teller 2003] est souvent proposée en environnement extérieur. La caméra et les capteurs de mouvement sont alors couplés, afin de combiner les avantages des deux approches et de limiter leurs défauts. Dans ce cas, la caméra et les capteurs doivent être mutuellement calibrés [Hol 2010].

6.1.3 Notre problème spécifique

Dans ce chapitre, nous traitons du problème du suivi de caméra fondé sur le système de marqueurs cycliques C^{2}Tags. Comme nous l'avons vu dans le chapitre précédent, le système de marqueurs C^{2}Tags présente de nombreux intérêts pour cette application. Un des intérêts majeurs de ce type de marqueur est qu'il fournit directement les images des points cycliques de son plan de support. Ceci nous permet de bénéficier des avantages fournis par notre système de marqueurs cycliques pour la conception d'une méthode de suivi de caméra fondé sur le calcul de la structure et du mouvement présenté dans le chapitre 3. Nous présentons dans ce chapitre deux paradigmes de suivi de caméra. Le premier consiste en un suivi de caméra hors ligne et le second en un suivi de caméra en ligne.

Contributions. Nous proposons une solution automatique de suivi de caméra reposant sur l'utilisation du système de marqueurs C^{2}Tags. Le paradigme général d'un suivi de caméra hors ligne est présenté suivi d'une description détaillée de l'ensemble des éléments qui le composent, avec notamment un paramétrage minimal pour l'ajustement de faisceaux. Nous proposons également un suivi de caméra en ligne, reposant sur une base de connaissances 3D de C^{2}Tags.

6.1.4 Notations

Nous considérons dans ce qui suit une séquence vidéo *à traiter* décrite par un ensemble d'indices

$$\mathtt{V} = \{1, \ldots, V\} \tag{6.1}$$

des vues qui la composent. Nous notons $v \in \mathtt{V}$ l'indice de la vue à traiter. Nous supposons une scène constituée d'un ensemble

$$\mathtt{M} = \{1, \ldots, M\} \tag{6.2}$$

de C^{2}Tags de cardinal $M \geq 2$ dont les éléments sont désignés par l'indice m.

– Nous désignons par

$$\mathsf{P}_v = \mathsf{K}_v \left[\begin{array}{c|c} \mathsf{R}_v & \mathbf{t}_v \end{array} \right] \tag{6.3}$$

la matrice de projection liée à la vue v, dans une certaine représentation euclidienne, où K_v désigne la matrice de calibrage de la caméra, où R_v est la matrice de rotation représentant son orientation et où $\mathbf{t}_v$ désigne le vecteur de $\mathbb{R}^3$ des coordonnées cartésiennes de l'origine de référentiel objet dans le repère de la caméra.

– Les *points cycliques d'un C^{2}Tag* désignent les points cycliques de leur plan de support et leur vecteur *exact* sont définis tels que

$$\mathbf{C}^{\infty}_{\pm m} = \begin{bmatrix} \mathbf{w}^1_m \pm i\mathbf{w}^2_m \\ 0 \end{bmatrix} \tag{6.4}$$

où les vecteurs $\mathbf{w}^1_m$ et $\mathbf{w}^2_m$ sont deux vecteurs orthonormés de $\mathbb{R}^3$. Nous attachons ainsi une même orientation 2D, autour de la normale à leur plan de support, à tous les C^{2}Tags situés

sur des plans parallèles puisqu'ils ont en commun la même paire de points cycliques.

– Nous désignons par *pose du C²Tag* m, la matrice carrée d'ordre 4

$$\mathsf{D}_m = \begin{bmatrix} \mathsf{W}_m & \bar{\mathbf{C}}_m \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \tag{6.5}$$

où W_m est une matrice de rotation d'ordre 3 représentant l'orientation du marqueur m, telle que $\mathsf{W}_m = \left[\mathbf{w}_m^1 | \mathbf{w}_m^2 | \mathbf{w}_m^1 \wedge \mathbf{w}_m^2\right]$, et $\bar{\mathbf{C}}_m$ est le vecteur de $\mathbb{R}^3$ des coordonnées cartésiennes du centre du marqueur.

– Nous désignons par $\tilde{\mathbf{I}}_{vm}^{\pm} = \tilde{\mathbf{a}}_{vm}^1 \pm i\tilde{\mathbf{a}}_{vm}^2$ ***les images des points cycliques*** et par $\tilde{\mathbf{c}}_{vm}$ ***l'image du centre*** associés au C²Tag m dans la vue v tels que ***fournis par l'algorithme de détection*** présenté dans le chapitre 5. Nous représentons l'ensemble de ces données, conformément à l'équation (5.14), via la matrice, inverse d'une matrice de rectification métrique centrée, de l'homographie

$$\tilde{\mathsf{H}}_{vm} = \left[\tilde{\mathbf{a}}_{vm}^1 \mid \tilde{\mathbf{a}}_{vm}^2 \mid \tilde{\mathbf{c}}_{vm}\right] \tag{6.6}$$

solution du problème (5.22), en l'absence de bruit, et telle que

$$\left[\tilde{\mathbf{I}}_{vm}^{\pm} \mid \tilde{\mathbf{c}}_{vm}\right] = \begin{bmatrix} \tilde{\mathbf{a}}_{vm}^1 \pm i\tilde{\mathbf{a}}_{vm}^2 \mid \tilde{\mathbf{c}}_{vm} \end{bmatrix} = \tilde{\mathsf{H}}_{vm} \begin{bmatrix} 1 & 0 \\ \pm i & 0 \\ 0 & 1 \end{bmatrix}. \tag{6.7}$$

Les vecteurs $\tilde{\mathbf{I}}_{vm}^{\pm}$ et $\tilde{\mathbf{c}}_{vm}$ sont connus aux facteurs d'échelle arbitraires, respectivement $\lambda_{vm} = r_{vm}(cos\theta_{vm} + isin\theta_{vm}) \in \mathbb{C}$ et $r_{vm} \in \mathbb{R}$ près tels que

$$\begin{bmatrix} \mathbf{a}_{vm}^1 \mid \mathbf{a}_{vm}^2 \mid \mathbf{c}_{vm} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{a}}_{vm}^1 \mid \tilde{\mathbf{a}}_{vm}^2 \mid \tilde{\mathbf{c}}_{vm} \end{bmatrix} \begin{bmatrix} a_{vm} & b_{vm} & 0 \\ -b_{vm} & a_{vm} & 0 \\ 0 & 0 & r_{vm} \end{bmatrix} \tag{6.8}$$

où $\begin{cases} a_{vm} = r_{vm}\cos\theta_{vm} \\ b_{vm} = r_{vm}\sin\theta_{vm} \end{cases}$, où $\mathbf{a}_{vm}^1 \pm i\mathbf{a}_{vm}^2$ représentent les images *exactes*, par la matrice de projection P_v, des points cycliques exacts $\mathbf{C}_{\pm m}^{\infty}$ du marqueur m, et où $\mathbf{c}_{vm}$ désigne l'image *exacte* du centre $\bar{\mathbf{C}}_m$ de ce dernier, rassemblés dans la matrice de l'homographie *exacte*

$$\mathsf{H}_{vm} = \begin{bmatrix} \mathbf{a}_{vm}^1 \mid \mathbf{a}_{vm}^2 \mid \mathbf{c}_{vm} \end{bmatrix} = \mathsf{P}_v \begin{bmatrix} \mathbf{w}_m^1 & \mathbf{w}_m^2 & \bar{\mathbf{C}}_m \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.9}$$

– Enfin, nous désignons par $\mathbf{U}_k$, $k = 1..K$, les vecteurs représentant les points naturels présents dans la scène et par $\mathbf{u}_{vk}$ leurs images par la matrice de projection P_v telles que

$$\mathbf{u}_{vk} \sim \mathsf{P}_v \mathbf{U}_k. \tag{6.10}$$

6.1.5 Formulation du problème

Le problème consiste à calculer, pour chacune des vues $v \in \mathtt{V}$ de la séquence vidéo à traiter, la matrice de projection euclidienne P_v à partir d'un ensemble d'au moins deux images de C^{2}Tags données sous la forme des homographies (6.6). Plus précisément, il s'agit de trouver la matrice de calibrage interne K_v, une matrice de rotation R_v et un vecteur de translation $\mathbf{t}_v$ tels que la matrice P_v s'écrive sous la forme de l'équation (6.3).

Les données du problème. Nous supposons que nous disposons, pour chaque vue $v \in \mathtt{V}$, des deux ensembles suivants :

- Le premier est l'ensemble des homographies, sous la forme (6.6),

$$\bar{\mathscr{H}}_v = \left\{ \tilde{\mathsf{H}}_{vm_j} \right\}_{j=1..|\mathtt{M}_v|} \tag{6.11}$$

  où l'ensemble $\mathtt{M}_v = \{\cdots, m_j, \cdots\} \subseteq \mathtt{M}$ réprésente l'ensemble des indices des C^{2}Tags visibles dans la vue v dont les éléments sont désignés par m_j, $j = 1..|\mathtt{M}_v|$.
- Le second, ***optionnel***, est l'ensemble des vecteurs des images des points naturels visibles dans la vue v

$$\mathscr{U}_v = \{\mathbf{u}_{vk}\}_{k=1..|\mathscr{U}_v|}\,. \tag{6.12}$$

Les hypothèses. Pour chaque vue v, nous formulons les hypothèses suivantes :

- *Les paramètres internes de la caméra sont inconnus.*
- *Nous disposons d'un nombre $|\mathtt{M}_v|$ minimal de deux images de C^{2}Tags.*
- *L'ensemble $\bar{\mathscr{H}}_v$ ne contient aucune donnée aberrante pouvant être liée, soit à une mauvaise identification de l'image d'un C^{2}Tag, soit à un faux positif.*

6.2 Suivi de caméra hors ligne basé sur les C^{2}Tags

Un suivi de caméra hors ligne repose typiquement sur la calcul de la structure et du mouvement à partir d'un sous-ensemble de vues de la séquence vidéo à traiter, désignées par le terme de *vues-clés*.

Paradigme général. Un suivi automatique de caméra hors ligne à partir de marqueurs C^{2}Tags est composé des étapes suivantes :

1. détection et identification des images des C^{2}Tags,
2. calcul des images des points cycliques des C^{2}Tags,
3. sélection des vues-clés,
4. calcul du suivi pour les vues-clés,
   (a) reconstruction 3D des C^{2}Tags et des caméras,
   (b) ajustement de faisceaux,

5. calcul du suivi pour les vues intermédiaires.

L'intérêt d'une sélection de vues-clés, cf. étape (3), est de réduire significativement la complexité de l'*ajustement de faisceaux* (défini dans la section 6.4) par la réduction du nombre de vues traitées. En effet, pour les approches de suivi de caméra basées sur le calcul de la structure et du mouvement, le temps de calcul de l'ajustement de faisceaux représente une part significative du temps d'exécution total. Sa complexité est typiquement quadratique par rapport au nombre de vues à taiter [Fitzgibbon 2003]. Une des stratégies employées afin de réduire ce coût est de traiter seulement des vues-clés pour le calcul de la reconstruction 3D de la scène. Les algorithmes de sélection de vues-clés mettent généralement en jeu un certain nombre d'heuristiques à partir des primitives image dans le but d'assurer, autant que possible, une reconstruction 3D fiable.

Ces vues-clés constituent les données d'entrée de l'étape (4), à savoir l'étape de calcul de la stucture et du mouvement, lors de laquelle sont retrouvées les reconstructions 3D des C^{2}Tags (plus éventuellement des points naturels) et des caméras pour chacune des vues-clés. L'étape (5) consiste ensuite à calculer les reconstructions 3D des caméras pour l'ensemble des *vues intermédiaires*, c.-à-d. toutes les vues n'étant pas des vues-clés. Cette dernière étape est effectuée par *resection* [Hartley 2004b, p. 178-179]. Il s'agit, pour chaque vue intermédiaire, de calculer la reconstruction 3D de la caméra à partir de la mise en correspondance des images des C^{2}Tags avec leur reconstruction 3D obtenue à l'issue de l'étape (4).

Nous proposons dans ce qui suit deux méthodes de suivi de caméra hors ligne.

– La première repose sur une scène composée de $M = 2$ C^{2}Tags coplanaires, un dispositif qui constitue une configuration théorique minimale de marqueurs, pour une scène composée seulement de C^{2}Tags, nécessaire à la reconstruction 3D des caméras. En effet, il est aisé de montrer qu'à partir d'un tel dispositif, il est possible de calculer l'homographie de l'équation (6.9), liant le plan de support des marqueurs au plan des pixels. La méthode a été publiée dans [Calvet 2012b].
– La seconde méthode de suivi de caméra généralise la première à une configuration de $M \geq 2$ C^{2}Tags. La scène doit dans ce cas être composée d'un minimum de deux C^{2}Tags dont les plans de support ne sont pas parallèles. Son principe est basé sur le calcul de la structure et du mouvement présenté dans la section 3.6 qui tire profit des classes d'équivalence (cf. §3.1) formées par notre système de marqueurs cycliques, afin de reconstruire simultanément les C^{2}Tags et l'ensemble des caméras sans avoir recours, par exemple, à une approche de calcul de la structure et du mouvement incrémentale. La méthode a été publiée dans [Calvet 2012a].

Les deux méthodes de suivi de caméra présentées diffèrent essentiellement par l'étape du calcul du suivi pour les vues-clés.

Notations. Nous désignerons l'ensemble des vues-clés par $\mathtt{V}_C$ de cardinal $|\mathtt{V}_C| \geq 2$ et l'ensemble des vues intermédiaires par $\mathtt{V}_I = \mathtt{V} \setminus \mathtt{V}_C$ de cardinal $|\mathtt{V}_I|$, où $\mathtt{V}$ désigne l'ensemble des vues (6.1) de la séquence à traiter.

6.2.1 Suivi de caméra à partir de deux C^{2}Tags coplanaires

Nous proposons dans cette partie une solution de suivi de caméra hors ligne à partir d'un motif plan composé de deux C^{2}Tags coplanaires, numérotés $m \in \mathtt{M} = \{1, 2\}$, dont le plan de support est noté π. Nous supposons que nous disposons, pour chaque vue v, de l'ensemble composé des deux homographies

$$\tilde{\mathscr{H}}_v = \left\{ \tilde{\mathsf{H}}_{vm} \right\}_{m=1,2}. \tag{6.13}$$

Le référentiel objet est ici attaché à la pose du marqueur 1 pour laquelle l'orientation 2D autour de la normale au plan de support est choisie de telle sorte que le centre du marqueur 2 ait pour coordonnées cartésiennes $\bar{\mathbf{C}}_2 = (d, 0, 0)^\top$, où d représente la distance *inter-centres*, c.-à-d. la distance euclidienne entre les centres $\bar{\mathbf{C}}_1$ et $\bar{\mathbf{C}}_2$, en supposant les cercles externes des marqueurs de rayon unitaire. Ainsi, l'expression de la pose du C^{2}Tag m, $m = 1, 2$ est

$$\mathsf{D}_m = \begin{bmatrix} \mathsf{I}_3 & \bar{\mathbf{C}}_m \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \tag{6.14}$$

où $\bar{\mathbf{C}}_1 = (0, 0, 0)$ et $\bar{\mathbf{C}}_2 = (d, 0, 0)$, et, au regard de l'équation de projection (6.9), l'homographie liant le plan π, de pose D_1, au plan image v s'écrit sous la forme

$$\begin{aligned} \mathsf{H}_v &= \left[\begin{array}{c|c|c} \mathbf{a}_v^1 & \mathbf{a}_v^2 & \mathbf{c}_v \end{array} \right] = \mathsf{K}_v \left[\begin{array}{c|c} \mathsf{R}_v & \mathbf{t}_v \end{array} \right] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \mathsf{K}_v \left[\begin{array}{c|c|c} \mathbf{r}_{1,v} & \mathbf{r}_{2,v} & \mathbf{t}_v \end{array} \right] \end{aligned} \tag{6.15}$$

où $\mathbf{r}_{1,v}$ et $\mathbf{r}_{2,v}$ désignent les deux premières colonnes de R_v et où les vecteurs $\mathbf{a}_v^1$ et $\mathbf{a}_v^2$ représentent les parties réelle et imaginaire, respectivement, des images *exactes*, par la matrice de projection P_v, des points cycliques communs aux marqueurs et au plan π (pour lesquels l'indice m est omis). Notons que la distance inter-centres d peut être calculée avec la seule donnée des homographies $\tilde{\mathsf{H}}_{vm}$, $m = 1, 2$. En effet, celle-ci correspond à la distance euclidienne entre les rétroprojections, par $\tilde{\mathsf{H}}_{vm}$, des images des centres $\tilde{\mathbf{c}}_{v1}$ et $\tilde{\mathbf{c}}_{v2}$.

Paradigme. Les différentes étapes du suivi de caméra à partir de deux marqueurs C^{2}Tags coplanaires sont les suivantes :

1. détection et identification des C^{2}Tags,
2. calcul des images des points cycliques du plan de support π,
3. sélection des vues-clés,
4. calcul du suivi pour les vues-clés,
   (a) calibrage intrinsèque de la caméra,
   (b) reconstruction 3D des caméras,

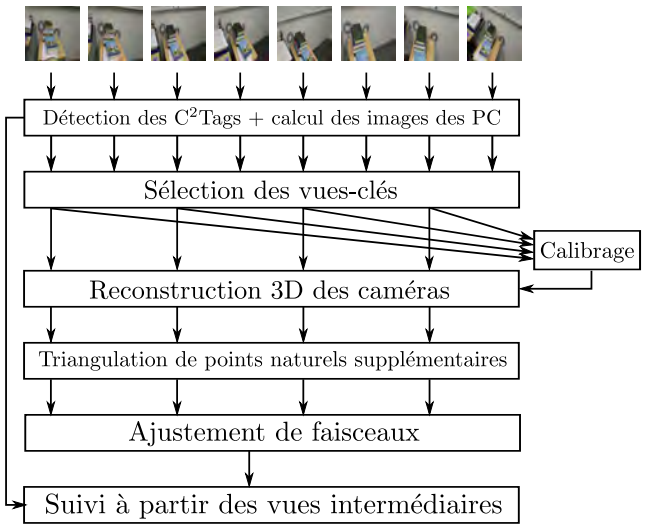


FIGURE 6.2: Paradigme du suivi de caméra à partir d'un dispositif de deux C^{2}Tags coplanaires.

(c) triangulation de points naturels supplémentaires (optionnel),

(d) ajustement de faisceaux,

5. calcul du suivi pour les vues intermédiaires.

Nous supposerons ici les paramètres internes de la caméra constants pour les raisons mentionnées dans le paragraphe dédié au calibrage interne de la caméra. L'ensemble du paradigme est illustré sur la figure 6.2.

Détection et identification des C^{2}Tags. Les solutions associées à ces deux sous-problèmes sont celles décrites dans le chapitre 5.

Calcul des points cycliques du plan de support. Ce problème consiste à calculer les images des points cycliques du plan de support π à partir « d'échantillons » que sont les images $\tilde{\mathbf{I}}_{vm}$ des points cycliques fournies dans la vue v par les deux C^{2}Tags sachant que, bien sûr, toutes ces images sont, en l'absence de bruit de mesure, égales à un facteur multiplicatif complexe près. Notre solution est de sélectionner la paire du marqueur $m_r \in \{1, 2\}$ ayant fourni le plus faible résidu vis-à-vis de la fonction de coût (5.22) dont l'objet est l'estimation conjointe des images des points cycliques et de l'image du centre.

Nous considérons dans la suite la donnée des images des points cycliques de π sous la forme d'une certaine homographie

$$\tilde{\mathsf{H}}_v = \begin{bmatrix} \tilde{\mathbf{a}}^1_{vm_r} & \tilde{\mathbf{a}}^2_{vm_r} & \cdot \end{bmatrix} \tag{6.16}$$

où $\cdot$ représente un vecteur de $\mathbb{R}^3$ arbitraire [1]. Un fait important est que, d'après les équations (6.8) et (6.15), la matrice $\tilde{\mathsf{H}}_v$ représente, en l'absence de bruit de mesure, l'homographie *exacte* $\tilde{\mathsf{H}}_v$

1. Choisi de telle sorte que la matrice $\tilde{\mathsf{H}}_v$ soit inversible, en prenant par exemple le vecteur $(\tilde{\mathbf{a}}^1_{vm_r} \wedge \tilde{\mathbf{a}}^2_{vm_r})$.

multipliée par une certaine similitude plane T telle que

$$\tilde{\mathsf{H}}_v \sim \mathsf{H}_v\mathsf{T}. \tag{6.17}$$

Sélection des vues-clés. Nous rappelons que l'objectif de l'étape de sélection des vues-clés est de réduire la complexité de l'ajustement de faisceaux. Sa conception prend ici également en compte les hypothèses sur lesquelles reposent le calibrage interne de la caméra fondé sur un calibrage plan, dont l'étape est décrite plus loin. Ce critère est construit de manière à assurer une bonne répartition des images des points cycliques sur l'image de la conique absolue et permet d'éviter les mouvements critiques [Sturm 1999] pour le calibrage plan, tel que par exemple une translation pure. À cette fin, nous proposons une heuristique reposant sur le critère suivant :

$$\rho_{uv} = |1 - (\mathbf{v}_u)^\top \mathbf{v}_v| \tag{6.18}$$

où $\mathbf{v}_u$ représente la ligne de fuite du plan π de la dernière vue-clé sélectionnée et $\mathbf{v}_v$ représente la ligne de fuite du plan de la vue courante, toutes deux normalisées pour avoir une norme unitaire. Dès que ρ est supérieur à $\epsilon = 0.03$ (déterminé de façon empirique) alors la vue v est sélectionnée comme une nouvelle vue-clé. Les lignes de fuite $\mathbf{v}_v$ sont calculées à partir des images des points cycliques telles que $\mathbf{v}_v \sim \tilde{\mathbf{I}}_v^+ \wedge \tilde{\mathbf{I}}_v^- \sim \tilde{\mathbf{a}}_v^1 \wedge \tilde{\mathbf{a}}_v^2$.

Calibrage interne de la caméra. Il s'agit d'un calibrage plan effectué à partir de l'ensemble $\{\tilde{\mathsf{H}}_v\}_{v\in\mathtt{V}_C}$ des images des points cycliques du plan de support, données sous la forme de l'équation (6.16), conduit conformément à l'algorithme présenté dans [Hartley 2004b, p. 211]. Les paramètres internes de la caméra sont supposés *constants*. Ceci se justifie par le fait que si nous ne disposons d'aucune connaissance *a priori* sur les paramètres internes de la caméra, il est nécessaire de disposer, en théorie, de $2 + \frac{1}{2}$ paires d'images de points cycliques, soit dans notre cas $2 + \frac{1}{2}$ vues, pour obtenir une solution au problème du calcul de l'image de la conique absolue. Les images des points cycliques (6.16) sont ainsi ajustées à l'image de la conique absolue dans chacune des vues-clés sélectionnées. Ceci nous fournit une solution $\hat{\mathsf{K}}$ pour la matrice de calibrage.

Reconstruction 3D des cercles. Le problème traité est le suivant : *étant données les v homographies obtenues sous la forme (6.17), c.-à-d. à une similitude arbitraire près du plan de support, et la matrice de calibrage $\hat{\mathsf{K}}$, estimer la position et l'orientation de la caméra dans le référentiel objet.* Plus précisément, il s'agit de trouver une matrice de rotation R_v et un vecteur de translation $\mathbf{t}_v$ tels que la matrice de projection P_v s'écrive sous la forme (6.3).

Pour chaque vue, cela nécessite deux étapes :

1. Conformément à l'approche [Sturm 2000], on cherche une matrice orthogonale S_v et un vecteur $\mathbf{w}_v$ solutions de

$$\begin{gathered}\underset{\mathbf{S}_v,\mathbf{w}_v,\lambda}{\arg\min} \parallel \lambda\hat{\mathsf{K}}^{-1}\tilde{\mathsf{H}}_v - \mathsf{S}_v \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{w}_v \parallel_F^2 \\ \text{s.c. } (\mathsf{S}_v)^\top \mathsf{S}_v = \mathsf{I}_3.\end{gathered} \tag{6.19}$$

Cela permet de « corriger au mieux » la matrice $\tilde{\mathsf{H}}_v$ en la remplaçant par le produit $\hat{\mathsf{H}}_v = \mathsf{K}\mathsf{S}_v \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{w}_v \Big)$ qui garantit d'avoir la « bonne forme » de matrice. À noter que l'ambiguïté sur les deux solutions possibles est levée en utilisant l'hypothèse suivante : la caméra se déplace en restant toujours du même côté du plan.

2. On calcule la matrice de projection P_v, c.-à-d. la matrice de rotation R_v et le vecteur $\mathbf{t}_v$ tels que P_v vérifie la décomposition (6.3). Ceci s'effectue en deux étapes :

   (a) On « aligne » les homographies $\hat{\mathsf{H}}_v$ obtenues à l'étape 1, c.-à-d. on cherche la similitude plane T à partir de $\hat{\mathsf{H}}_v$ telle que, en l'absence de bruit, la décomposition

   $$\hat{\mathsf{H}}_v \mathsf{T} = \mathsf{H}_v \tag{6.20}$$

   soit vérifiée. Cette étape ne comporte alors aucune difficulté si on utilise les hypothèses suivantes : (a) les deux marqueurs sont correctement identifiés ; (b) l'origine du référentiel objet est le centre du premier marqueur et le centre du second marqueur a pour coordonnées cartésiennes $(d, 0, 0)^\top$. Le calcul de la matrice T est décrit en annexe de ce chapitre.

   (b) Si on exprime la matrice de la similitude plane T sous la forme :

   $$\mathsf{T} = \begin{bmatrix} s_v \mathsf{A}_v & \mathbf{b}_v \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{6.21}$$

   où s_v est un facteur d'échelle, A_v une matrice de rotation d'ordre 2 et $\mathbf{b}_v$ un vecteur de dimension 2, la v-ème caméra est alors décrite par la matrice de projection suivante :

   $$\mathsf{P}_v \sim \mathsf{K}\mathsf{S}_v \begin{bmatrix} \mathsf{I} \mid \mathbf{w}_v \end{bmatrix} \begin{bmatrix} s_v \mathsf{A}_v & 0 & \mathbf{b}_v \\ \mathbf{0}^\top & s_v & 0 \\ \mathbf{0}^\top & 0 & 1 \end{bmatrix} \tag{6.22}$$

   où, pour faire le lien avec (6.3),

   $$\mathsf{R}_v = \mathsf{S}_v \begin{bmatrix} \mathsf{A}_v & 0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad \text{et} \quad \mathbf{t}_v = \frac{1}{s_v} \mathsf{S}_v \left[\begin{bmatrix} \mathbf{b}_v \\ 0 \end{bmatrix} + \mathbf{w}_v \right]. \tag{6.23}$$

Reconstruction de points naturels supplémentaires par triangulation. Il s'agit d'une étape optionnelle qui consiste à calculer les reconstructions 3D des points naturels, représentés par les vecteurs $\mathbf{U}_k$, $k = 1..K$, à partir de leurs images $\mathbf{u}_{vk}$ mises en correspondance dans les vues-clés, et de l'ensemble des caméras $\{\mathsf{P}_v\}_{v \in \mathtt{V}_C}$ calculé à l'étape précédente.

La solution repose sur l'équation de projection (6.10)

$$\begin{aligned} \mathbf{u}_{vk} &\sim \mathsf{P}_v \mathbf{U}_k \\ [\mathbf{u}_{vk}]_\wedge \mathsf{P}_v \mathbf{U}_k &= \mathbf{0}_3 \end{aligned} \tag{6.24}$$

qui contraint linéairement les éléments du vecteur $\mathbf{U}_k$. Si l'on considère le point $\mathbf{U}_k$ visible dans l'ensemble de vues $\mathtt{V}_k \subseteq \mathtt{V}_C$, alors le système composé de l'ensemble des équations (6.24) est résolu au sens des moindres carrés linéaires si l'on dispose d'un nombre minimal de $|\mathtt{V}_k| \geq 2$ vues. Plus de détails sont fournis dans [Hartley 2004b, p. 312-313].

Les images $\mathbf{u}_{vk}$ peuvent être extraites à partir d'algorithmes tels que [Lucas 1985] où [Lowe 2004] puis mises en correspondance de manière robuste via l'algorithme RANSAC [Fischler 1981].

Ajustement de faisceaux. L'ajustement de faisceaux dédié à ce type de dispositif est présenté dans la section 6.4. Lors de cette étape, sont optimisés à la fois l'ensemble des caméras, à savoir la matrice de calibrage interne K, les matrices de rotation R_v et les vecteurs $\mathbf{t}_v$, où $v \in \mathtt{V}_C$, et la structure, à savoir la distance inter-centres d et l'ensemble des points naturels $\mathbf{U}_k$.

Suivi à partir des vues intermédiaires. Il s'agit ici de calculer, pour chaque vue intermédiaire $v \in \mathtt{V}_I$, les matrices de projection P_v sous la forme de l'équation (6.3). La matrice de calibrage interne $\hat{\mathsf{K}}$ correspond à la solution obtenue à l'issue de l'ajustement de faisceaux. Le problème se ramène donc à calculer les matrices R_v et les vecteurs $\mathbf{t}_v$, pour lequel une solution est décrite dans le paragraphe dédié à la « reconstruction 3D des cercles ».

Si l'étape de triangulation de points naturels a été effectuée, alors le suivi à partir des vues intermédiaires est effectué conformément à la méthode proposée dans la section 6.2.2.2.

6.2.2 Suivi de caméra à partir de $N \geq 2$ C^{2}Tags

Nous présentons dans cette deuxième partie un suivi de caméra hors ligne qui constitue la généralisation du suivi de caméra précédent à une configuration générale de $M \geq 2$ C^{2}Tags. La scène se compose ici d'un minimum de deux C^{2}Tags non parallèles. Son principe s'appuie sur la notion de marqueur cyclique pour le calcul de la structure et du mouvement (présentée dans la section 3.6), qui tire profit de la propriété d'équivalence (cf. §3.1) des C^{2}Tags situés sur des plans parallèles afin de reconstruire simultanément les C^{2}Tags et l'ensemble des caméras. Les classes de marqueurs cycliques appariés sont supposées connues [2]. Les paramètres internes de la caméra sont supposés *variables*.

Paradigme. Un suivi de caméra hors ligne basé sur des marqueurs C^{2}Tags est composé des étapes suivantes :

1. détection et identification des C^{2}Tags.
2. calcul des images des points cycliques des C^{2}Tags.
3. sélection des vues-clés.
4. calcul du suivi pour les vues-clés.
   (a) reconstruction 3D affine des orientations des C^{2}Tags/des caméras par factorisation.

2. Cette information doit être fournie par l'utilisateur et/ou nécessite une disposition particulière de C^{2}Tags dans la scène liée à leur identifiant.

(b) calibrage interne de la caméra.

(c) reconstruction 3D euclidienne des orientations des C^{2}Tags/des caméras.

(d) reconstruction 3D euclidienne des positions des C^{2}Tags/des caméras.

(e) ajustement de faisceaux.

5. calcul du suivi pour les vues intermédiaires par *resection*.

Détection et identification des C^{2}Tags. Les solutions à ces deux sous-problèmes sont fournis dans le chapitre 5.

Calcul des images des points cycliques. Les images des points cycliques sont calculées lors de l'étape de détection des C^{2}Tags (cf. chapitre 5). Lorsque plusieurs marqueurs sont présents sur une même famille de plans parallèles, les images des points cycliques considérées correspondent à celles du C^{2}Tag ayant fourni le plus faible résidu par rapport à la fonction de coût (5.22) associée à l'optimisation conjointe de l'image du centre et des images des points cycliques du marqueur.

Sélection des vues-clés. Afin de garantir les hypothèses formulées par la méthode de calcul de la structure et du mouvement présentée dans la section 3.6, l'ensemble $\mathtt{V}_C$ des vues-clés sélectionnées doit respecter les conditions suivantes :

- $|\mathtt{V}_C| > 2$
- Quel que soit $v \in \mathtt{V}_C$, l'ensemble $\mathtt{M}_v$ des indices des C^{2}Tags visibles dans la vue v a pour cardinal $|\mathtt{M}_v| > 2$
- L'ensemble $\mathtt{M}_v$ doit être associé à un ensemble de C^{2}Tags non tous parallèles.

Ainsi, nous définissons l'heuristique de sélection de vues-clés suivante. L'ensemble des vues-clés est initialement assigné à $\mathtt{V}_C \leftarrow \{1\}$ et la dernière vue-clé sélectionnée, désignée par u, est assignée à $u \leftarrow 1$. Ensuite, à chaque itération, associée à la vue courante v, si un marqueur encore jamais rencontré durant les v premières vues apparaît dans la vue courante, ou que la moyenne sur l'ensemble des images des marqueurs visibles du critère (6.18) varie suffisamment (de plus de 0.03), alors la vue courante v est ajoutée à la liste des vues-clés et u est assigné à v. Ce critère permet, entre autres, d'assurer la reconstruction de l'ensemble des C^{2}Tags visibles dans la séquence vidéo.

6.2.2.1 Suivi à partir des vues-clés

Nous disposons, à ce stade, d'un ensemble de vues-clés pour lesquelles nous disposons des images des C^{2}Tags détectés sous la forme de l'équation (6.6). Le suivi à partir des vues-clés consiste alors à résoudre le problème du calcul de la structure et du mouvement tel que formulé dans la section 3.6 et pour lequel une solution pour les étapes 4-(a) à 4-(d) est proposée. L'étape d'ajustement de faisceaux est présentée dans la section 6.4.

À l'issue de cette étape, nous disposons des matrices de projection $\hat{\mathsf{P}}_v$, où $v \in \mathtt{V}_C$, sous la forme (6.3) et de l'ensemble des poses des marqueurs estimées sous la forme de l'équation (6.5)

$$\hat{\mathscr{B}} = \left\{\hat{\mathsf{D}}_m\right\}_{m\in\mathtt{M}} \tag{6.25}$$

Dans la suite, nous supposons les poses $\hat{\mathsf{D}}_m$ exactes et le symbole « ˆ » est omis.

6.2.2.2 Suivi à partir des vues intermédiaires par resection

Le problème traité ici est le suivant. *Pour chaque vue intermédiaire $v \in \mathtt{V}_I$, étant données les images des C^{2}Tags présentes dans la vue et l'ensemble (6.25) des poses des C^{2}Tags sous la forme de l'équation (6.5), estimer la matrice de projection* P_v. Ce problème est connu sous le nom de *resection* [Hartley 2004b, p. 178-179] [Fischler 1981]. Plus précisément, il s'agit de trouver la matrice de calibrage K_v, la matrice de rotation R_v et le vecteur de translation $\mathbf{t}_v$ associés à la caméra v tels que la matrice de projection P_v s'écrive sous la forme (6.3).

Le problème est résolu « vue par vue » et nous présentons dans ce qui suit la solution du problème associée à une certaine vue v.

Les données du problème. Les données correspondent à deux ensembles :

– Le premier est l'ensemble des homographies, sous la forme (6.6),

$$\bar{\mathscr{H}_v} = \left\{\tilde{\mathsf{H}}_{vm_j}\right\}_{j=1..|\mathtt{M}_v|} \tag{6.26}$$

– Le second est l'ensemble

$$\mathscr{B} = \{\mathsf{D}_m\}_{m\in\mathtt{M}} \tag{6.27}$$

des poses des C^{2}Tags sous la forme (6.5).

Méthode proposée. Le problème admet une solution linéaire [Hartley 2004b, p. 178-181] basée sur l'équation suivante :

$$\begin{aligned} \mathsf{P}_v\mathbf{X} &\sim \tilde{\mathbf{x}} \\ \Leftrightarrow [\tilde{\mathbf{x}}]_\wedge \mathsf{P}_v\mathbf{X} &= \mathbf{0}_3 \\ \Leftrightarrow (\mathbf{X}^\top \otimes [\tilde{\mathbf{x}}]_\wedge)\,\mathrm{vec}(\mathsf{P}_v) &= \mathbf{0}_3 \end{aligned} \tag{6.28}$$

où $\mathbf{X}$ représente dans notre cas, soit les points cycliques $\mathbf{C}_m^\infty$, soit les centres $\mathbf{C}_m$ des marqueurs, et où $\tilde{\mathbf{x}}$ représentent leurs images, désignées par $\tilde{\mathbf{I}}_{vm}$ pour les images des points cycliques et par $\tilde{\mathbf{c}}_m$ pour les images des centres (connues à un facteur d'échelle arbitraire près) et où $\otimes$ représente le produit de Kronecker. Dans le cas des images des points cycliques, l'équation (6.28) est complexe

et se réécrit sous la forme

$$\begin{aligned} (\mathbf{C}_m^{\infty T} \otimes [\tilde{\mathbf{I}}_{vm}]_\wedge) \operatorname{vec}(\mathsf{P}_v) &= \mathbf{0}_3 \\ \Leftrightarrow (\mathbf{w}_{1m}^\top + i\mathbf{w}_{2m}^\top) \otimes \left[(\tilde{\mathbf{a}}_{vm}^1 + i\tilde{\mathbf{a}}_{vm}^2)\right]_\wedge \operatorname{vec}(\bar{\mathsf{P}}_v) &= \mathbf{0}_3 \end{aligned} \tag{6.29}$$

ce qui donne les deux équations réelles suivantes :

$$(\mathbf{w}_{1m}^\top \otimes \left[\tilde{\mathbf{a}}_{vm}^1\right]_\wedge - \mathbf{w}_{2m}^\top \otimes \left[\tilde{\mathbf{a}}_{vm}^2\right]_\wedge) \operatorname{vec}(\bar{\mathsf{P}}_v) = \mathbf{0}_3 \tag{6.30}$$

$$(\mathbf{w}_{1m}^\top \otimes \left[\tilde{\mathbf{a}}_{vm}^2\right]_\wedge + \mathbf{w}_{2m}^\top \otimes \left[\tilde{\mathbf{a}}_{vm}^1\right]_\wedge) \operatorname{vec}(\bar{\mathsf{P}}_v) = \mathbf{0}_3 \tag{6.31}$$

où $\bar{\mathsf{P}}_v$ représente la matrice composée des trois premières colonnes de P_v.

Solution minimale. Pour un ensemble $\mathtt{M}_v \subseteq \mathtt{M} = \{1, \cdots, M\}$ de $|\mathtt{M}_v|$ images de marqueurs présentes dans la vue v, dont les éléments sont désignés par m_j où $j = 1..|\mathtt{M}_v|$, l'équation peut se réécrire sous la forme

$$\mathsf{D}\mathbf{X} = \mathbf{0}_{9|\mathtt{M}_v|} \tag{6.32}$$

où le vecteur

$$\mathbf{X} = \operatorname{vec}(\mathsf{P}_v) \in \mathbb{R}^{12} \tag{6.33}$$

a pour composante les inconnues, à savoir les 12 éléments de P_v (traitée colonne par colonne), et où

$$\mathsf{D} = \begin{bmatrix} \mathsf{B}_1 \\ \vdots \\ \mathsf{B}_{|\mathtt{M}_v|} \end{bmatrix} \tag{6.34}$$

désigne la matrice structurée par blocs (de dimension 9×12) définis par

$$\mathsf{B}_j = \begin{bmatrix} \mathbf{C}_{m_j}^\top \otimes [\tilde{\mathbf{c}}_{m_j}]_\wedge \\ (\mathbf{w}_{1m_j}^\top \otimes \begin{bmatrix}\tilde{\mathbf{a}}_{vm_j}^1\end{bmatrix}_\wedge - \mathbf{w}_{2m_j}^\top \otimes \begin{bmatrix}\tilde{\mathbf{a}}_{vm_j}^2\end{bmatrix}_\wedge) \;\; \mathsf{0}_{3\times 3} \\ (\mathbf{w}_{1m_j}^\top \otimes \begin{bmatrix}\tilde{\mathbf{a}}_{vm_j}^2\end{bmatrix}_\wedge + \mathbf{w}_{2m_j}^\top \otimes \begin{bmatrix}\tilde{\mathbf{a}}_{vm_j}^1\end{bmatrix}_\wedge) \;\; \mathsf{0}_{3\times 3} \end{bmatrix}. \tag{6.35}$$

Il s'agit d'un système linéaire homogène comportant $9|\mathtt{M}_v|$ équations, dont $6|\mathtt{M}_v|$ sont indépendantes. Chaque marqueur C^{2}Tag apporte 6 équations indépendantes. Un solution (non nulle) exacte existe si $6|\mathtt{M}_v| = 12 - 1$, c.-à-d. pour un nombre minimal de $|\mathtt{M}_v| = 1 + \frac{5}{6}$ marqueurs. $\frac{5}{6}$ signifie que seulement 5 équations parmi les 6 fournies par un marqueur sont utilisées. En présence de bruit, une solution au sens des moindres carrés existe si l'on dispose d'au moins $|\mathtt{M}_v| \geq 2$ C^{2}Tags.

Calcul de la pose de la caméra. Le problème est ici *d'estimer, à partir de la solution* $\hat{\mathsf{P}}_v$ *obtenue comme solution du problème précédent, la matrice de calibrage* K_v*, la matrice de rotation* R_v *et le vecteur* $\mathbf{t}_v$ *tels que, en l'absence de bruit de mesure,* $\mathsf{P}_v = \mathsf{K}_v\,[\,\mathsf{R}_v \mid \mathbf{t}_v\,]$.

Plusieurs méthodes peuvent être envisagées. Nous décrivons deux de ces méthodes qui présen-

tent, dans notre cas, les meilleurs résultats en pratique. Nous allons voir que le choix d'une méthode peut dépendre de la nature et de la quantité des primitives image disponibles dans la vue. Ces deux techniques estiment toutes en premier lieu les matrices $\mathsf{K}_v = \mathrm{diag}(f_v, f_v, 1)$ en supposant des pixels carrés et le point principal à l'origine, et diffèrent par la technique utilisée.

- Une première approche consiste à effectuer un calibrage plan [Hartley 2004b, p. 211] à partir de l'ensemble des images des points cycliques (éventuellement corrigées) des C^{2}Tags. Le nombre d'équations dépend du nombre d'images de C^{2}Tags dans la vue.
- La seconde consiste consiste à effectuer un calibrage plan à partir des images de points cycliques, de vecteurs $\mathbf{p}_v^1 \pm i\mathbf{p}_v^2$, $\mathbf{p}_v^2 \pm i\mathbf{p}_v^3$ et $\mathbf{p}_v^1 \pm i\mathbf{p}_v^3$ où $\mathbf{p}_v^c$ désigne la colonne c de P_v. Dans ce cas, toute vue donne le même nombre d'équations.

Une fois l'estimation de la matrice de calibrage $\hat{\mathsf{K}}_v$ obtenue, nous calculons la matrice $\hat{\mathsf{R}}_v$ et le vecteur $\hat{\mathbf{t}}_v$ solutions du problème

$$\begin{aligned} \underset{\mathbf{R}_v, \mathbf{t}_v, s_v}{\arg\min} \parallel s_v \hat{\mathsf{K}}_v^{-1} \hat{\mathsf{P}}_v - [\, \mathsf{R}_v \mid \mathbf{t}_v] \parallel_F^2 \\ \text{s.c. } (\mathsf{R}_v)^\top \mathsf{R}_v = \mathsf{I}_3. \end{aligned}$$

dont la solution est analogue à celle du problème (6.20).

6.3 Suivi de caméra en ligne basé sur une base de connaissances 3D

Dans cette section, nous présentons un suivi de caméra en ligne reposant sur une *base de connaissances 3D*. Celle-ci est constituée des reconstructions 3D des C^{2}Tags calculées à partir de la méthode de calcul de la structure et du mouvement présentée dans la section 3.6 et se présente sous la forme de l'équation (6.27). Un tel scénario implique donc une phase de « préparation du suivi » pendant laquelle l'utilisateur devra accéder à la scène afin de l'équiper, c.-à-d. d'introduire des C^{2}Tags en nombre suffisant, de telle sorte que lors du suivi en ligne, suffisamment de marqueurs soient visibles dans les images afin de garantir un calcul fiable et précis de la pose de la caméra. Ceci peut éventuellement nécessiter une connaissance « grossière » du mouvement de la caméra lors du suivi. Ainsi, la méthode proposée se compose des deux étapes principales suivantes :

1. Étape de préparation du suivi.
   (a) introduction dans la scène des C^{2}Tags,
   (b) acquisition d'une collection de vues de la scène sous la forme de photographies ou d'une séquence vidéo,
   (c) reconstruction 3D des C^{2}Tags à partir du calcul de la structure et du mouvement tel que présenté dans la section 3.6.
2. **Suivi de caméra en ligne.** Pour chaque vue du flux vidéo à traiter :
   (a) détection et identification des images des C^{2}Tags,
   (b) reconstruction 3D de la caméra par resection (§6.2.2.2).

L'étape (2), à savoir l'étape de suivi de caméra en ligne, correspond exactement au même problème que celui du suivi de caméra à partir des vues intermédiaires tel qu'il est formulé dans la section 6.2.2.2. Pour plus de détails sur la solution proposée, cf. section 6.2.2.2.

Comme nous l'avons mentionné plus haut, les avantages d'un tel suivi résident dans le fait qu'il présente de bonnes garanties de performances, et dans la faible complexité de ses algorithmes sous-jacents. Néanmoins, la principale difficulté de telles approches reste la mise en correspondance robuste entre les primitives image et les primitives 3D. Ce problème n'a pas été abordé dans cette thèse et constitue une des perspectives du travail réalisé.

6.4 Ajustement de faisceaux

L'ajustement de faisceaux est une technique utilisée pour calculer l'estimation au maximum de vraisemblance d'une reconstruction à partir de primitives images [Triggs 1999, Hadem 1981, Slama 1980]. Un estimateur au maximum de vraisemblance dans le cas des approches pour lesquelles les primitives à reconstruire sont des points naturels consiste en la minimisation d'une fonction de coût donnée par le *résidu de reprojection*, c.-à-d. la distance euclidienne entre les points mesurés dans les images et les points reprojetés, par rapport à tous les paramètres inconnus du modèle. Cette technique est optimale sous l'hypothèse d'un bruit additif gaussien centré pour chacune des positions 2D des points mesurés qui constituent des variables aléatoires indépendantes et identiquement distribuées. Les paramètres du modèle pouvant être inclus dans cette optimisation sont, en particulier, la structure 3D, les matrices de projection et le cas échéant, les données de calibrage interne de la caméra.

Ces méthodes d'ajustement de faisceaux ne sont pas adaptées à notre scène puisque nous ne disposons pas directement de mises en correspondance des points mesurés dans les images. Ainsi, dans notre cas, le résidu de reprojection est la distance euclidienne entre les points mesurés et les ellipses, images des cercles de l'espace 3D constituant les C^2Tags présents dans la scène. En pratique, ce résidu est une approximation de la distance euclidienne entre les points mesurés et les ellipses reprojetées. En effet, le calcul de la distance euclidienne entre un point et une ellipse induit la résolution d'une équation quartique, celle-ci est donc habituellement remplacée par une approximation plus simple et plus efficace.

Une de nos contributions est de proposer un paramétrage minimal des reconstructions 3D euclidiennes d'un C^2Tag et d'un dispositif composé de deux C^2Tags coplanaires.

6.4.1 Formulation générale du problème

L'objectif de l'ajustement de faisceaux est d'optimiser simultanément la structure de la scène et le mouvement de la caméra. Il s'agit d'un problème aux moindres carrés non linéaires. L'ensemble des paramètres ν pour le paramétrage 3D euclidien et projectif d'un C^2Tag et d'un dispositif composé de deux C^2Tags coplanaires est décrit dans la suite du chapitre. L'expression de la fonc-

tion de coût pour une scène composée de marqueurs C^{2}Tags est la suivante :

$$\mathcal{C}(\nu) = \sum_v \sum_m \sum_k d(x^m_{v,k}, \mathsf{E}^m_v) \tag{6.36}$$

où E^m_v est la projection du cercle 3D associé au marqueur d'indice m, $x^m_{v,k}$ un point mesuré sur l'ellipse externe [3] du marqueur m, où v désigne l'indice de la vue, et où $d(.,.)$ est l'approximation de la distance euclidienne entre un point et une ellipse proposée dans [Kanatani 1996], intéressante pour sa simplicité analytique. Nous l'interprétons géométriquement comme la distance du point $\mathbf{x}$ à la droite polaire de ce point par rapport à l'ellipse de matrice E [Hartley 2004b, p. 58] (cf. figure 6.3) :

$$d(\mathbf{x}, \mathsf{E}) = \frac{\mathbf{x}^\top \mathsf{E}\mathbf{x}}{\| \operatorname{diag}(1,1,0)\mathsf{E}\mathbf{x}\|_2}. \tag{6.37}$$

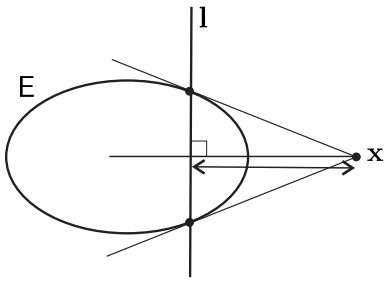
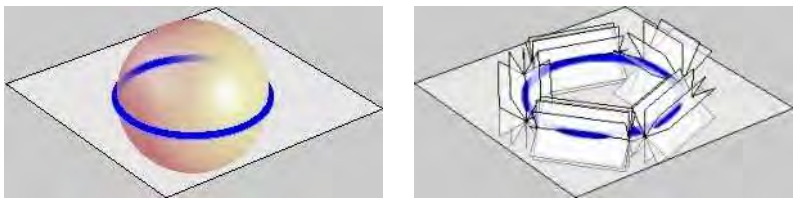


FIGURE 6.3: La relation pôle-polaire : la droite $\mathbf{l} = \mathsf{E}\mathbf{x}$ est la polaire du point $\mathbf{x}$ respectivement à la conique E [Hartley 2004b, p. 58].

6.4.2 Paramétrage 3D euclidien minimal

6.4.2.1 Paramétrage d'un C^{2}Tag

En premier lieu, nous utilisons une des « bizarreries » de l'espace projectif dual 3D afin de modéliser un cercle de l'espace 3D : une conique s'identifie à un cône dual, c.-à-d. à une quadrique duale de rang 3. Cette « dualité » est illustrée sur la figure 6.4.



(a) (b)

FIGURE 6.4: (a) Dans l'espace 3D projectif, un cercle correspond au lieu de points correspondant à l'intersection d'une sphère et d'un plan ; (b) dans l'espace dual, un cercle est une quadrique dégénérée, c.-à-d. une envelope de plans 3D (cf. §2.2.4).

3. L'ensemble des ellipses qui composent l'image du C^{2}Tag peut être considéré. L'indice associé à chacune des ellipses est ici omis.

L'intérêt d'un tel modèle réside dans le fait que l'équation de projection d'une conique s'écrit matriciellement, sous sa forme duale

$$\mathsf{E}_c^* \sim \mathsf{P}\mathsf{Q}_c^*\mathsf{P}^\top \tag{6.38}$$

où P est la matrice de projection, Q_c^* la matrice de la quadrique duale au cercle et E_c^* la matrice de l'ellipse dans l'espace projectif dual, image de Q_c^*, sachant que la matrice de l'image du cercle se déduit grâce à la relation

$$(\mathsf{E}_c^*)^{-1} \sim \mathsf{E}_c = \begin{pmatrix} (\mathbf{e}_2 \wedge \mathbf{e}_3)^\top \\ (\mathbf{e}_3 \wedge \mathbf{e}_1)^\top \\ (\mathbf{e}_1 \wedge \mathbf{e}_2)^\top \end{pmatrix} \tag{6.39}$$

où $\mathbf{e}_k$ est la colonne k de E_c^*.

On déduit facilement par dualité du résultat (8.10) de [Hartley 2004b, p. 202] qu'un cercle de rayon r fixé et de centre $\mathbf{C}$, obtenu par l'intersection d'une sphère S de rayon r et de centre $\mathbf{C}$ (3 ddl) et d'un plan $\boldsymbol{\pi}$ passant par $\mathbf{C}$ (2 ddl) admet la décomposition suivante :

$$\mathsf{Q}_\mathsf{c}^* = (\boldsymbol{\pi}^\top \mathsf{S}^* \boldsymbol{\pi})\mathsf{S}^* - (\mathsf{S}^*\boldsymbol{\pi})(\mathsf{S}^*\boldsymbol{\pi})^\top, \tag{6.40}$$

cf. proposition 6, où

$$\boldsymbol{\pi} = \left(\mathbf{n}^\top - \mathbf{n}^\top \bar{\mathbf{C}}\right)^\top \tag{6.41}$$

est l'expression du plan de support dans l'espace projectif dual et

$$\mathsf{S}^* = \mathbf{C}\mathbf{C}^\top - r^2 \operatorname{diag}(1,1,1,0) \tag{6.42}$$

est la matrice de la sphère de rayon r, où $\mathbf{C} = (\bar{\mathbf{C}}\ 1)^\top$ désignant le vecteur de coordonnées cartésiennes augmentées de son centre. La normale $\mathbf{n}$ au plan $\boldsymbol{\pi}$ est paramétrée par deux angles θ et ϕ associés à ses coordonnées sphériques

$$\mathbf{n} = \begin{pmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi \end{pmatrix} \tag{6.43}$$

On montre ainsi qu'un cercle 3D a un nombre minimal de 5 ddl. Un C^{2}Tag numéro m étant constitué d'un ensemble de cercles de rayons connus et fixés, de même centre et situés sur un même plan $\boldsymbol{\pi}$, il peut être représenté par le vecteur de paramètres

$$\boldsymbol{\nu}_m = \begin{bmatrix} \bar{\mathbf{C}}_m^\top & \theta_m & \phi_m \end{bmatrix}^\top. \tag{6.44}$$

Ensemble des paramètres. Le paramétrage de la structure et du mouvement est ainsi réuni dans le vecteur

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_{in} & \boldsymbol{\nu}_{ex,1} & \dots & \boldsymbol{\nu}_{ex,|\mathtt{V}_C|} & \boldsymbol{\nu}_1 & \dots & \boldsymbol{\nu}_M \end{bmatrix}^\top \tag{6.45}$$

où ν_{in} représente le vecteur des paramètres intrinsèques de la caméra, $\boldsymbol{\nu}_{ex,v}$ le vecteur de ses 6 paramètres externes associés à la vue-clé v et où $\mathtt{V}_C$ représente l'ensemble des vues-clés.

6.4.2.2 Paramétrage de deux C^{2}Tags coplanaires

Nous considérons ici deux C^{2}Tags coplanaires centrés respectivement en C_1 et C_2. Nous proposons dans ce qui suit un paramétrage minimal associé à un tel dispositif.

Centres des sphères. Soient deux sphères S_1 et S_2 centrées en C_1 et C_2 respectivement. Le centre C_1 (3 ddl) est paramétré par le vecteur homogène $\mathbf{C}_1 = (\bar{\mathbf{C}}_1^\top \ 1)^\top$ de ses trois coordonnées cartésiennes augmentées et le centre C_2 (3 ddl) est paramétré par le point à l'infini C_∞ et d tel que

$$\mathbf{C}_2 = \mathbf{C}_1 + d \begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix} \tag{6.46}$$

où $\mathbf{u} \in \mathbb{R}^3$ est le vecteur unitaire représentant la direction associée à C_∞, c.-à-d. la direction de la droite portée par C_1 et C_2 ; d est la distance entre les deux centres. Les deux ddl de $\mathbf{u}$ correspondent aux deux angles θ_u et ϕ_u associés à ses coordonnées sphériques :

$$\mathbf{u} = \begin{pmatrix} cos\theta_u cos\phi_u \\ sin\theta_u cos\phi_u \\ sin\phi_u \end{pmatrix} \tag{6.47}$$

À noter que, par construction, $\|\mathbf{u}\|_2 = 1$. Deux cercles coplanaires, de même rayon r fixé, ont 7 ddl dans l'espace 3D. Nous justifions cette affirmation dans ce qui suit en associant 7 paramètres à ces 7 ddl.

Plan de support. Soit l_C la droite passant par les centres C_1 et C_2 des deux sphères et soit $\mathsf{L_C}$ la matrice de Plücker associée, obtenue à partir des vecteurs $\mathbf{C}_1$ et $\mathbf{C}_2$ de C_1 et C_2, via la décomposition $\mathsf{L_C} = \mathbf{C}_1\mathbf{C}_2^\top - \mathbf{C}_2\mathbf{C}_1^\top$ [Hartley 2004b, p. 70]. Le faisceau de plans ayant pour axe cette droite contient $\boldsymbol{\pi}$, plan de support du cercle. Il contient aussi :

- le plan π_1 défini par l_C et le centre O de la caméra, dont le vecteur est $\boldsymbol{\pi}_1 = \mathsf{L}_\mathsf{C}^*\mathbf{O}$ avec L_C^* la représentation duale de la matrice de Plücker associée à $\mathsf{L_C}$,
- le plan π_2 défini par l_C et par un autre point P non situé sur π_1, dont le vecteur est $\boldsymbol{\pi}_2 = \mathsf{L}_\mathsf{C}^*\mathbf{P}$.

L'ensemble de ces entités géométriques est représenté sur la figure 6.5.

Un faisceau de plans est une famille linéaire de plans à 1 ddl. Le plan du support $\boldsymbol{\pi}$ (1 ddl) est ainsi paramétré par $\alpha \in \mathbb{R}$ tel que

$$\boldsymbol{\pi} = \alpha\boldsymbol{\pi}_1 + \boldsymbol{\pi}_2 \tag{6.48}$$

À noter que nous faisons implicitement l'hypothèse que le plan de support ne passe pas par le centre de la caméra car, dans (6.48), il n'existe pas de valeur de α telle que $\boldsymbol{\pi} = \alpha\boldsymbol{\pi}_1$.

En faisant la somme de tous les ddl mentionnés entre parenthèses, nous obtenons bien un total égal à 7. Notre dispositif peut ainsi être paramétré par le vecteur

$$\boldsymbol{\nu} = \begin{bmatrix} \bar{\mathbf{C}}_1^\top & \theta_u & \phi_u & d & \alpha \end{bmatrix}^\top. \tag{6.49}$$

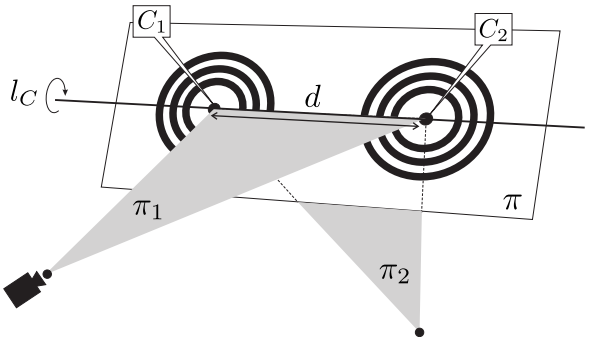


FIGURE 6.5: *Structure de la scène.* La droite l_C passe par les centres C_1 et C_2. Le plan π_1 est défini par l_C et le centre O de la caméra. Le plan π_2 est défini par l_C et par un point non situé sur π_1.

6.4.3 Paramétrage 3D projectif minimal

Un C^{2}Tag est paramétré par son cercle externe et le centre (commun) des cercles. Plaçons-nous dans une représentation projective arbitraire de $P_3(\mathbb{C})$. Si $\mathbf{C}$ est le vecteur du centre, alors ce cercle est dual au cône de $P_3^*(\mathbb{C})$ dont la matrice est

$$\mathsf{Q}_\mathsf{c}^* = \boldsymbol{\Sigma}_\infty^* - r\mathbf{C}\mathbf{C}^\top \tag{6.50}$$

où $r \in \mathbb{R}$ est le « rayon projectif » du cercle et où

$$\boldsymbol{\Sigma}_\infty^* = \mathbf{q}_1\mathbf{q}_1^\top + \mathbf{q}_2\mathbf{q}_2^\top$$

est la matrice de rang 2 de l'enveloppe formée par la paire de points cycliques du plan de support du C^{2}Tag, de vecteurs $\mathbf{q}_1 \pm i\mathbf{q}_2$ où $[\mathbf{q_1} \mid \mathbf{q_2}] \in \mathbb{R}^{3\times 2}$.

La démonstration de l'existence d'une telle décomposition peut se faire en réalité dans n'importe quelle représentation de l'espace. En particulier, nous choisissons une représentation euclidienne de $P_3(\mathbb{C})$ telle que le plan de support ait pour vecteur $\boldsymbol{\pi} = [0, 0, 1, 0]^\top$, que le centre du cercle soit à l'origine et que le rayon soit 1. Il en découle que le cercle admet la decomposition (6.40) qui est $\mathsf{Q}_\mathsf{c}^* = \mathrm{diag}(1, 1, 0, -1)$. Montrons qu'à partir de (6.50) on obtient le même résultat. Dans la représentation euclidienne considérée, $\boldsymbol{\Sigma}_\infty^* = \mathrm{diag}(1, 1, 0, 0)$ d'où

$$\mathsf{Q}_\mathsf{c}^* = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} - 1 \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & \\ & & & -1 \end{bmatrix}$$

ce qui termine la preuve.

Concernant les degrés de liberté (ddl) d'une représentation projective du cercle, les points cycliques en possèdent $8-2$, le centre $4-1$ et le rayon 1, donnant un total de 10 ddl. En considérant respectivement les parties réelle et imaginaire des points cycliques, on montre aisément que, dans

toute représentation projective, il existe (a, b) tel que

$$[\mathbf{q}_1 \mid \mathbf{q}_2] \begin{bmatrix} a & b \\ -b & a \end{bmatrix} = \begin{bmatrix} v_1 & v_5 \\ v_2 & v_6 \\ v_3 & v_7 \\ v_4 & 0 \end{bmatrix} \tag{6.51}$$

On obtient ainsi un paramétrage à 6 ddl des points cycliques.

L'équation du projection du cercle est donnée par (6.38).

6.5 Résultats expérimentaux

6.5.1 Résultats sur données de synthèse

Nous présentons ici les résultats obtenus en simulation pour le suivi de caméra hors ligne à partir de deux C^{2}Tags coplanaires. Les résultats sont exprimés à partir de la racine carrée de l'erreur quadratique moyenne (REQM). 100 jeux de données de synthèse ont été générés, chacun étant composé d'un ensemble de vues issues de 20 caméras.

Évaluation de la reconstruction 3D initiale de deux C^{2}Tags coplanaires. Nous évaluons ici la phase d'initialisation linéaire pour le calcul des paramètres internes, des positions et des orientations des caméras pour une scène composée de deux C^{2}Tags coplanaires. Soit $\mathbf{x}^{(0)}$ le vecteur de paramètres initial calculé comme présenté dans la section 6.2.1. Les données d'entrée sont les points 2D situés sur chacune des ellipses auxquels nous avons ajouté un bruit blanc gaussien dont les écarts-types sont successivement $\sigma_1 = 0.5$, $\sigma_2 = 1$ et $\sigma_3 = 2$ pixels. La figure 6.6 présente la REQM du vecteur $\mathbf{x}^{(0)}$ en fonction du nombre de points mesurés sur chacune des ellipses. Nous constatons une réduction importante de l'erreur REQM jusqu'à 50 points, qui se stabilise ensuite à 50, 100 et 150 points par ellipse pour un bruit de $\sigma_1 = 0.5$ pixel, $\sigma_2 = 1$ pixel et $\sigma_2 = 2$ pixels respectivement.

Évaluation de l'ajustement de faisceaux euclidien de deux C^{2}Tags coplanaires. L'ajustement de faisceaux euclidien a été évalué en simulation. Le dispositif étudié est celui composé de deux C^{2}Tags coplanaires. Nous présentons les résultats de l'algorithme d'optimisation non linéaire conduit via l'algorithme de Levenberg-Marquardt. Les deux expériences suivantes ont été réalisées :

- Dans un premier temps, l'ajustement de faisceaux a été effectué à partir du vecteur initial $\mathbf{x}^{(0)}$ issu de la procédure d'initialisation. L'écart-type du bruit appliqué sur les points mesurés est de 1 pixel. L'expérience est renouvelée en faisant varier le nombre de points mesurés sur chacune des ellipses. La figure 6.7 présente les résultats obtenus à chaque itération.
- Une deuxième expérience a été réalisée en effectuant l'ajustement de faisceaux basé sur un vecteur initial calculé à partir de seulement 5 points approximativement équidistants sur

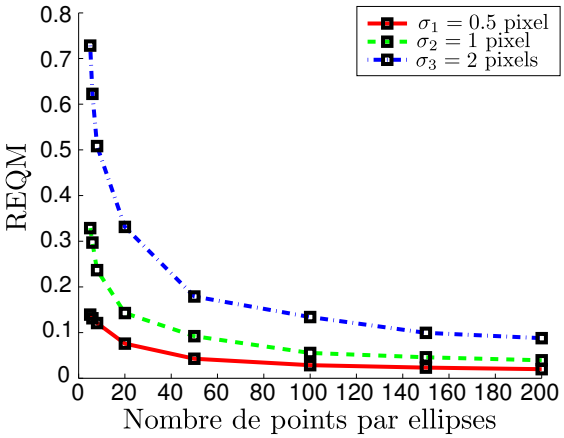


FIGURE 6.6: Erreurs moyennes du vecteur $\mathbf{x}^{(0)}$ suivant le nombre de points considérés sur chacune des ellipses. L'erreur est une moyenne estimée sur 100 simulations. L'expérience est renouvelée en faisant varier l'écart-type du bruit gaussien appliqué aux points mesurés.

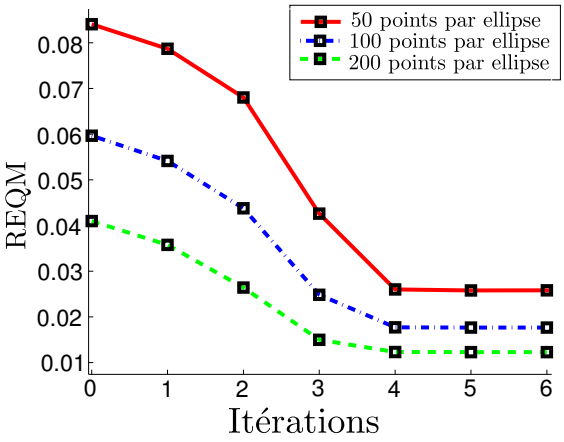


FIGURE 6.7: Convergence moyenne de l'algorithme de minimisation non linéaire lors de l'ajustement de faisceaux effectué sur 20 vues. L'erreur est une moyenne estimée sur 100 simulations. L'expérience est réalisée sur 20 vues en considérant 100 points par ellipse. Celle-ci est renouvelée en faisant varier l'écart-type du bruit gaussien appliqué sur les mesures.

chacune des ellipses. L'ajustement de faisceaux est ensuite conduit en considérant un ensemble de 200 points mesurés par ellipse. Le but de cette deuxième expérience est d'étudier le comportement de l'ajustement de faisceaux lorsque l'initialisation est peu précise. La figure 6.8 présente les résultats obtenus à chaque itération.

Nous pouvons noter, dans ces deux expériences, que le gain en précision obtenu par l'algorithme de minimisation est principalement atteint au bout de 4 itérations.

6.5.2 Résultats sur données réelles

Nous présentons dans cette section les résultats de chacun des algorithmes de suivi de caméra proposés à partir de séquences vidéo réelles.

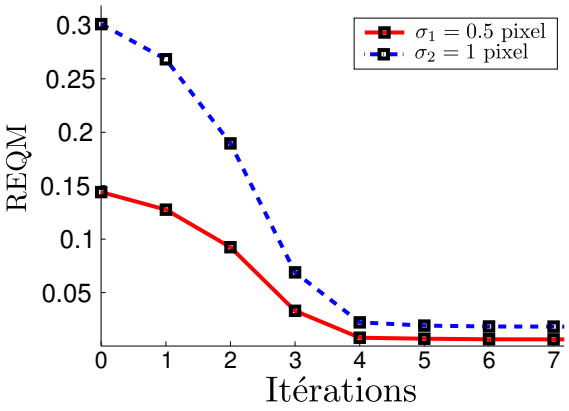


FIGURE 6.8: Convergence moyenne de l'algorithme de minimisation non linéaire lors de l'ajustement de faisceaux lorsque l'initialisation des paramètres est peu précise. L'erreur est une moyenne estimée sur 100 simulations. L'expérience est réalisée sur 20 vues et on considère 200 points par ellipse pour l'ajustement de faisceaux. Celle-ci est renouvelée en faisant varier l'écart-type du bruit gaussien appliqué aux mesures.

Suivi de caméra à partir de deux C^{2}Tags coplanaires. La figure 6.9 illustre les résultats obtenus pour la méthode de suivi de caméra présentée dans la section 6.2.1. La scène filmée comprend deux marqueurs C^{2}Tags coplanaires (fixés sur un mur) parfois partiellement occultés lors de l'acquisition de la séquence vidéo. Afin de permettre une évaluation visuelle des performances de la méthode proposée, une théïère synthétique a été ajoutée à la séquence vidéo, dont deux captures d'écran [4] sont montrées sur la figure 6.9(b). Le principe de sélection des vues-clés est illustré sur la figure 6.10. La figure 6.10(a) illustre l'évolution du critère défini par l'équation (6.18) lors de la séquence vidéo et la figure 6.10(b) présente, pour cette même expérience, l'ensemble des vues-clés sélectionnées.

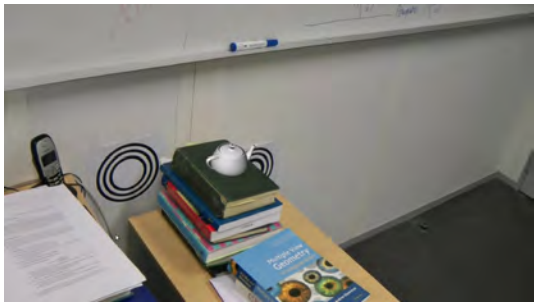
Suivi de caméra à partir de C^{2}Tags situés sur des plans parallèles. La figure 6.12 présente les résultats obtenus pour le suivi de caméra hors ligne présenté dans la section 6.2.2 et le suivi de caméra en ligne présenté dans la section 6.3 sur une même séquence vidéo. Il s'agit d'une scène d'intérieur dans laquelle ont été disposés des C^{2}Tags situés sur des plans parallèles, à savoir des plans parallèles aux murs et au sol. Deux expériences, associées au suivi de caméra hors ligne présenté dans la section 6.2.2 et au suivi de caméra en ligne présenté dans la section 6.3, ont été réalisées. Concernant la deuxième expérience, la base de connaissances 3D nécessaire au suivi de caméra en ligne a été calculée à partir d'une collection de photographies (illustrées sur la figure 6.11) conformément à la méthode de calcul de la structure et du mouvement présentée dans la section 3.6. Ces photographies constituent les vues-clés de l'algorithme de reconstruction et sont volontairement prises de points de vue différents afin de pouvoir comparer les performances du suivi de caméra hors ligne avec celles du suivi de caméra en ligne.

Les deux algorithmes de suivi ont présenté des performances similaires, ce qui peut essentiellement se justifier par des reconstructions 3D des C^{2}Tags très précises fournies par l'algorithme de

4. La vidéo « augmentée » est disponible à l'url suivante : `http://ubee.enseeiht.fr/vision/icip2012/demo.avi`.

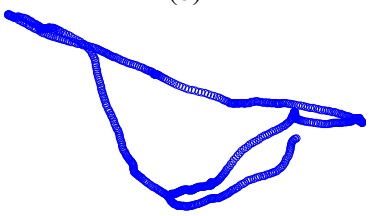


(a)





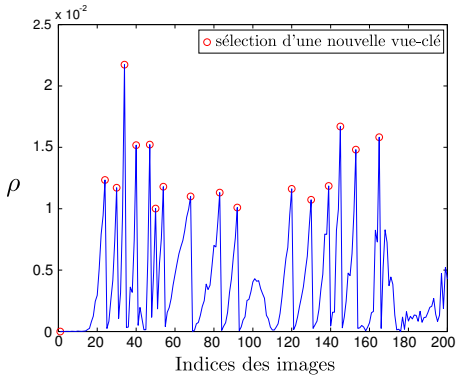
(b)





(c)

FIGURE 6.9: Résultats sur une séquence vidéo réelle d'un suivi de caméra à partir de deux C^2Tags coplanaires. (a) Image extraite de la séquence sur laquelle sont mises en évidence (en vert) les ellipses, images des cercles concentriques qui composent les C^2Tags. (b) Images extraites de la séquence « augmentée » par une théière de synthèse à l'issue du calcul du suivi de caméra. (c) Mouvement 3D de la caméra.



(a)



(b)

FIGURE 6.10: Sélection de vues-clés pour un suivi de caméra à partir de deux C^{2}Tags coplanaires. (a) Évolution du critère (6.18) au cours d'une séquence vidéo de 200 images. Chacune des vues-clés sélectionnées est mise en évidence par un cercle rouge. (b) Vues-clé sélectionnées dans (a), présentées dans l'ordre chronologique.

reconstruction. Les résultats fournis par les deux suivis sont illustrés sur une même figure, à savoir la figure 6.12. La figure 6.12(a) présente des vues extraites de la séquence vidéo « augmentée » [5] par des cônes de synthèse. La figure 6.12(b) illustre la reconstruction 3D des marqueurs et la trajectoire de la caméra obtenue.

Temps de calcul. Concernant le temps d'exécution de l'algorithme de suivi de caméra en ligne, pour lequel un des objectifs principaux est la réduction du temps de calcul, le goulet d'étranglement reste l'étape de détection des marqueurs, notamment lorsqu'un grand nombre de marqueurs est visible dans l'image. Plus de détails sur ce sujet sont fournis dans la section 5.5.2. Le calcul de la pose de la caméra (y compris son optimisation) présente quant à lui un temps négligeable par rapport au temps d'exécution total.

5. La séquence vidéo complète est disponible à l'url suivante : `http://ubee.enseeiht.fr/vision/icpr2012/demo.avi`.

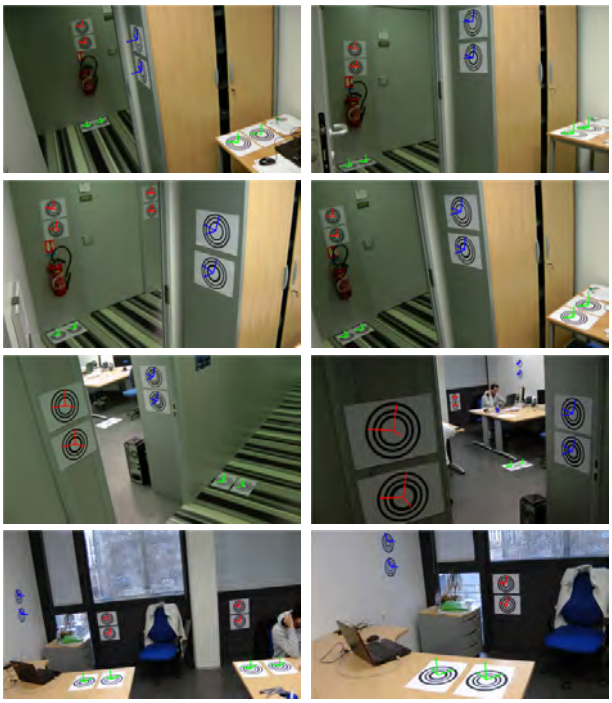


FIGURE 6.11: Photographies utilisées comme données d'entrée de l'algorithme de reconstruction 3D des C^2Tags pour la constitution de la base de connaissance 3D. Chaque image de C^2Tag a été augmentée par un repère de synthèse dont la couleur est associée à la classe d'équivalence des marqueurs cycliques, c.-à-d. associée à la même paire de points cycliques.

6.6 Conclusion

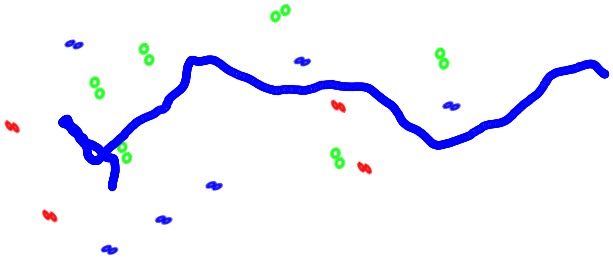
Dans ce chapitre, nous avons présenté plusieurs solutions de suivi de caméra reposant sur le système de marqueurs C^2Tags. Deux solutions de suivi de caméra hors ligne ont été proposées. La première repose sur un dispositif de deux C^2Tags coplanaires et la seconde sur une configuration générale de C^2Tags. À la différence d'un suivi de caméra hors ligne basé sur un calcul incrémental de la structure et du mouvement, cette seconde solution permet une reconstruction simultanée de l'ensemble des C^2Tags et des caméras grâce à la notion de marqueur cyclique et ce, même lorsqu'il s'agit d'une longue séquence vidéo. Un tel suivi requiert néanmoins une disposition de marqueurs particulière, à savoir des C^2Tags situés sur des familles de plans parallèles. Une telle disposition est certainement facile à mettre en œuvre dans des environnements intérieurs ou « créés par l'homme » . L'utilisation d'une telle stratégie peut par exemple être envisagée en studio (pour l'industrie du cinéma, de la télévison, etc.).

Pour des environnements où une telle disposition est difficile à mettre en œuvre, nous suggérons une approche incrémentale du calcul de la structure et du mouvement utilisant comme solution initiale une reconstruction 3D partielle obtenue, par exemple, à partir de la méthode présentée dans la section 3.7 utilisant conjointement des points d'intérêt naturels et des C^2Tags. Pour de tels cas d'usage, l'utilisation des C^2Tags peut être motivée par les trois points suivants :

– absence de texture dans certaines zones de l'environnement,
– meilleures garanties de performances pour le calcul d'une reconstruction projective de la



(a)



(b)

FIGURE 6.12: (a) Vues extraites de la séquence vidéo « augmentées » par des cônes synthétiques sur chacun des marqueurs. (b) Reconstruction 3D des marqueurs et de la trajectoire de la caméra retrouvée.

structure et du mouvement,

– meilleures garanties de performances pour l'autocalibrage.

Un axe de recherche que nous envisageons est celui du problème de la mise en correspondance robuste des images des C^{2}Tags avec leur modèle 3D utilisée lors du suivi de caméra en ligne.

Annexes

Calcul de $(\mathsf{T})^{-1}$

Le scalaire s_v, la matrice A_v et le vecteur $\mathbf{b}_v$ sont calculés à partir des équations suivantes.

$$s_v = \frac{\|g_{\hat{\mathsf{H}}}(\tilde{\mathbf{p}}_{v1}) - g_{\hat{\mathsf{H}}}(\tilde{\mathbf{p}}_{v2})\|_2}{d} \tag{6.52}$$

où $g_{\hat{\mathsf{H}}}(\cdot) \in \mathbb{R}^2$ est la fonction de rétroprojection par $\hat{\mathsf{H}}$ définie par l'équation (5.2). Si l'on note la droite

$$\mathbf{l} = (\hat{\mathsf{H}}_v^{-1}\tilde{\mathbf{p}}_{v1}) \wedge (\hat{\mathsf{H}}_v^{-1}\tilde{\mathbf{p}}_{v2}) \tag{6.53}$$

alors

$$\mathsf{A}_v = \begin{bmatrix} l_1 & -l_2 \\ l_2 & l_1 \end{bmatrix} \tag{6.54}$$

où l_j, $j = 1, 2$ désigne le $j$ème élément de $\mathbf{l}$ au facteur d'échelle défini tel que A_v soit unitaire. Enfin

$$\mathbf{b}_v = g_{\hat{\mathsf{H}}}(\tilde{\mathbf{p}}_{v1}), \tag{6.55}$$

c.-à-d. la rétroprojection par $\hat{\mathsf{H}}$ de l'image du centre $\tilde{\mathbf{p}}_{v1}$.

Chapitre 7

Conclusion

Dans ce manuscrit, nous avons présenté l'ensemble de nos contributions à la recherche en vision par ordinateur portant essentiellement sur le problème du calcul de la structure et du mouvement à partir d'une collection de vues non calibrées, et sur la conception d'un système de marqueurs dédié à une application de suivi de caméra.

Pour le calcul de la structure et du mouvement, nous avons proposé une méthode permettant d'intégrer les images de points virtuels dans un schéma de factorisation projective. La notion de marqueurs cycliques a motivé l'approche proposée qui intègre les images des points cycliques comme blocs d'entrée de la matrice des mesures. Nous avons décrit deux schémas de factorisation projective mettant en jeu les images des points cycliques, sans et avec l'utilisation conjointe des images de points naturels réels. Les algorithmes traitant des problèmes de mise à l'échelle des entrées de la matrice des mesures et de prédiction des données manquantes, inhérents aux approches par factorisation, ont été fournis et prouvés.

Nous avons également fourni des équations additionnelles indépendantes pour l'autocalibrage opérant dans l'espace projectif dual qui reposent sur les images de paires de points cycliques. Nous avons montré que les approches d'autocalibrage présentées dans [Pollefeys 1999, Pollefeys 2004, Seo 2004, Ponce 2005, Ronda 2008] suivaient en réalité le même paradigme consistant à identifier des droites intersectant la conique absolue. Dans la méthode proposée, ces droites sont les rétro-projections des images des points cycliques. Un algorithme d'autocalibrage unifié a été proposé.

La deuxième partie de ce manuscrit concerne la conception d'un système de marqueurs et son utilisation dans une application de suivi de caméra. Nous avons présenté le système de marqueurs C^2Tags, conçu dans l'optique de tirer profit des intérêts que présentent l'utilisation des marqueurs cycliques pour le calcul de la structure et du mouvement. Nous avons montré que le C^2Tag présente de très bonnes propriétés tant pour la localisation de son image que pour son identification et ce, même en présence d'occultation et de flou de bougé.

L'ensemble de ces contributions a finalement été mis au profit d'une application de suivi automatique de caméra reposant sur le système de marqueurs C^2Tags. Deux méthodes de suivi de caméra hors ligne ont été proposées ainsi qu'une méthode de suivi de caméra en ligne, conforme aux exigences du projet industriel ROM. Toutefois, la solution proposée doit encore être améliorée

afin d'atteindre des performances en temps réel en portant les efforts essentiellement sur les algorithmes de traitement du système de marqueurs C^2Tags.

Plus généralement, les questions que nous pouvons nous poser à ce stade sont les suivantes :

- Comment peut-on contrôler les performances du calcul de la structure et du mouvement proposé, c.-à-d. à la fois pour le calcul d'une reconstruction projective et pour l'autocalibrage, à partir des positions/orientations des marqueurs introduits dans la scène ?
- Par quels moyens peut-on proposer une configuration minimale de marqueurs qui garantisse les performances du calcul de la structure et du mouvement ?

Bibliographie

[3DE] *3DEqualizer*. http://www.3dequalizer.com.

[Act] *Acts*. http://www.zjucvg.net/acts/acts.html.

[Agarwal 2009] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz et R. Szeliski. *Building Rome in a day*. Dans International Conference on Computer Vision, 2009.

[AIC] *AICON3D*. http://www.aicon.de.

[Anandan 2002] P. Anandan et M. Irani. *Factorization with Uncertainty*. International Journal of Computer Vision, vol. 49, no. 2-3, pages 101–116, 2002.

[ART] *Advanced Realtime Tracking*. http://www.ar-tracking.de.

[Ballard 1981] D. H. Ballard. *Generalizing the Hough transform to detect arbitrary shapes*. PR, vol. 13, no. 2, pages 111–122, 1981.

[Bergamasco 2011] F. Bergamasco, A. Albarelli, E. Rodolà et A. Torsello. *RUNE-Tag : A high accuracy fiducial marker with strong occlusion resilience*. Dans Conference on Computer Vision and Pattern Recognition, 2011.

[Ble] *Blender*. http://www.blender.org.

[Bocquillon 2007] B. Bocquillon, A. Bartoli, P. Gurdjos et A. Crouzil. *On Constant Focal Length Self-Calibration From Multiple Views*. Dans Conference on Computer Vision and Pattern Recognition, 2007.

[Bou] *2D3 Boujou*. http://www.2d3.com.

[Bouguet] J. Y. Bouguet. *Camera Calibration Toolbox for Matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc/.

[Brandner 2003] M. Brandner, M. Ribo et A. Pinz. *State of the art of vision-based self-localisation*. Dans Robotic and Sensors Environments, 2003.

[Buchanan 2005] A. M. Buchanan et A. W. Fitzgibbon. *Damped Newton Algorithms for Matrix Factorization with Missing Data*. Dans Conference on Computer Vision and Pattern Recognition, 2005.

[Calvet 2012a] L. Calvet, P. Gurdjos et V. Charvillat. *Camera Tracking based on Circular Point factorization*. Dans International Conference on Pattern Recognition, 2012.

[Calvet 2012b] L. Calvet, P. Gurdjos et V. Charvillat. *Camera tracking using concentric circle markers : paradigms and algorithms*. Dans International Conference on Image Processing, 2012.

[Calvet 2013] L. Calvet et P. Gurdjos. *An Enhanced Structure-from-Motion Paradigm based on the Absolute Dual Quadric and Images of Circular Points*. Dans International Conference on Computer Vision, 2013.

[Canny 1986] J. Canny. *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pages 679–698, 1986.

[Chandraker 2007a] M. K. Chandraker, S. Agarwal, F. Kahl, D. Nistér et D. J. Kriegman. *Autocalibration via Rank-Constrained Estimation of the Absolute Quadric*. Dans Conference on Computer Vision and Pattern Recognition, 2007.

[Chandraker 2007b] M. K. Chandraker, S. Agarwal, D. J. Kriegman et S. Belongie. *Globally Optimal Affine and Metric Upgrades in Stratified Autocalibration*. Dans International Conference on Computer Vision, 2007.

[Cho 1998] Y. Cho, J. Lee et U. Neumann. *A Multi-ring Color Fiducial System and an Intensity-invariant Detection Method for Scalable Fiducial-Tracking Augmented Reality*. Dans International Workshop on Augmented Reality, 1998.

[Christy 1996] Stéphane Christy et Radu Horaud. *Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 11, pages 1098–1104, 1996.

[Claus 2004] D Claus et A. W. Fitzgibbon. *Reliable Fiducial Detection in Natural Scenes*. Dans European Conference on Computer Vision, 2004.

[Corke 2007] P. Corke, J. Lobo et J. Dias. *An Introduction to Inertial and Visual Sensing*. International Journal of Robotic Research, vol. 26, no. 6, pages 519–535, 2007.

[Davison 2003] Andrew J. Davison. *Real-Time Simultaneous Localisation and Mapping with a Single Camera*. Dans International Conference on Computer Vision, 2003.

[Davison 2007] A. J. Davison, I. D. Reid, N. Molton et O. Stasse. *MonoSLAM : Real-Time Single Camera SLAM*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pages 1052–1067, 2007.

[Dehais 2010] C. Dehais, G. Morin et V. Charvillat. *From rendering to tracking point-based 3D models*. Image Vision Comput., vol. 28, no. 9, pages 1386–1395, 2010.

[Dempster 1977] A. P. Dempster. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, vol. 39, no. 1, pages 1–38, 1977.

[Dong 2009] Z. Dong, G. Zhang, J. Jia et H. Bao. *Keyframe-based real-time camera tracking*. Dans International Conference on Computer Vision, 2009.

[Drummond 2002] T. Drummond et T Cipolla. *Real-Time Visual Tracking of Complex Structures*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pages 932–946, 2002.

[Duda 1972] R. O. Duda et P. E. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Communications of the ACM, vol. 15, no. 1, pages 11–15, 1972.

[Fiala 2005] M. Fiala. *ARTag, a Fiducial Marker System Using Digital Techniques*. Dans Conference on Computer Vision and Pattern Recognition, 2005.

[Fiala 2010] M. Fiala. *Designing Highly Reliable Fiducial Markers*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 7, pages 1317–1324, 2010.

[Fischler 1981] M. A. Fischler et R. C. Bolles. *Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Commun. ACM, vol. 24, no. 6, pages 381–395, 1981.

[Fitzgibbon 1999] A. W. Fitzgibbon, M. Pilu et R. B. Fisher. *Direct Least Square Fitting of Ellipses*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pages 476–480, 1999.

[Fitzgibbon 2003] A. W. Fitzgibbon et A. Zisserman. *Automatic Camera Tracking*. Dans Video Registration, chapitre 2, pages 18–35. Kluwer, 2003.

[Foley 1990] James Foley, Andries van Dam, Steven Feiner et John Hughes. Computer graphics : Principles and practice, second edition. Addison-Wesley Professional, 1990.

[Furukawa 2010] Y. Furukawa et J. Ponce. *Accurate, Dense, and Robust Multiview Stereopsis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 8, pages 1362–1376, 2010.

[Fusiello 2004] A. Fusiello, A. Benedetti, M. Farenzena et A. Busti. *Globally Convergent Autocalibration Using Interval Analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 12, pages 1633–1638, 2004.

[Gallo 2011] O. Gallo et R. Manduchi. *Reading 1-D Barcodes with Mobile Phones Using Deformable Templates*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 9, pages 1834–1843, 2011.

[Golub 1996] G. H. Golub et C. F. Van Loan. Matrix computations (3rd ed.). Johns Hopkins University Press, 1996.

[Gruber 2004] A. Gruber et Y. Weiss. *Multibody Factorization with Uncertainty and Missing Data Using the EM Algorithm*. Dans Conference on Computer Vision and Pattern Recognition, 2004.

[Gurdjos 2003] P. Gurdjos et P. F. Sturm. *Methods and Geometry for Plane-Based Self-Calibration*. Dans Conference on Computer Vision and Pattern Recognition, 2003.

[Gurdjos 2006] P. Gurdjos, P. F. Sturm et Y. Wu. *Euclidean Structure from $N \geq 2$ Parallel Circles : Theory and Algorithms*. Dans European Conference on Computer Vision, 2006.

[Gurdjos 2009] P. Gurdjos, A. Bartoli et P. F. Sturm. *Is Dual Linear Self-Calibration Artificially Ambiguous ?* Dans International Conference on Computer Vision, 2009.

[Habed 2012] A. Habed, K. A. Ismaeil et D. Fofi. *A New Set of Quartic Trivariate Polynomial Equations for Stratified Camera Self-calibration under Zero-Skew and Constant Parameters Assumptions*. Dans European Conference on Computer Vision, 2012.

[Hadem 1981] I. Hadem. *Bundle adjustment in industrial photogrammetry*. Photogrammetria, vol. 37, no. 2, pages 45–60, 1981.

[Hamming 1950] R. W. Hamming. *Error detecting and error correcting codes*. vol. 29, no. 2, pages 147–160, 1950.

[Han 2000] M. Han et T. Kanade. *Creating 3D Models with Uncalibrated Cameras*. Dans WACV, 2000.

[Hartley 2004a] R. I. Hartley et F. Schaffalitzky. *L_∞ Minimization in Geometric Reconstruction Problems*. Dans Conference on Computer Vision and Pattern Recognition, 2004.

[Hartley 2004b] R. I. Hartley et A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, second édition, 2004.

[Heyden 1997] A. Heyden et A. Åström. *Euclidean Reconstruction from Image Sequences with Varying and Unknown Focal Length and Principal Point*. Dans Conference on Computer Vision and Pattern Recognition, 1997.

[Heyden 1999] A. Heyden, R. Berthilsson et G. Sparr. *An iterative factorization method for projective structure and motion from image sequences*. Image and Vision Computing, vol. 17, no. 13, pages 981–991, 1999.

[Hoff 1996] W. A. Hoff, K. Nguyen et T. Lyon. *Computer-vision-based registration techniques for augmented reality*. Dans SPIE, 1996.

[Hol 2010] J. D. Hol, T. B. Schön et F. Gustafsson. *Modeling and Calibration of Inertial and Vision Sensors*. International Journal of Robotic Research, vol. 29, no. 2-3, pages 231–244, 2010.

[Irschara 2009] A. Irschara, C. Zach, J. Frahm et H. Bischof. *From structure-from-motion point clouds to fast location recognition*. Dans Conference on Computer Vision and Pattern Recognition, pages 2599–2606, 2009.

[Jacobs 1997] D. W. Jacobs. *Linear Fitting with Missing Data : Applications to Structure-from-Motion and to Characterizing Intensity Images*. Dans Conference on Computer Vision and Pattern Recognition, 1997.

[Jia 2009] H. Jia et A. M. Martínez. *Low-Rank Matrix Fitting Based on Subspace Perturbation Analysis with Applications to Structure from Motion*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pages 841–854, 2009.

[Jiang 2005] G. Jiang et L. Quan. *Detection of Concentric Circles for Camera Calibration*. Dans International Conference on Computer Vision, 2005.

[Kahl 2005] F. Kahl. *Multiple View Geometry and the L_∞-norm*. Dans International Conference on Computer Vision, 2005.

[Kanatani 1996] K. Kanatani. Statistical Optimization for Geometric Computation : Theory and Practice. Elsevier, Amsterdam, 1996.

[Kanatani 2004] K. Kanatani et N. Ohta. *Automatic Detection Of Circular Objects By Ellipse Growing*. IJIG, vol. 4, no. 1, pages 35–50, 2004.

[Kato 1999] H. Kato et M. Billinghurst. *Marker tracking and HMD calibration for a video-based augmented reality conferencing system*. Dans International Workshop on Augmented Reality, 1999.

[Kato 2000] H. Kato, M. Billinghurst et I. Poupyrev. *Virtual object manipulation on a table-top AR environment*. Dans International Symposium on Augmented Reality, 2000.

[Ke 2007] Q. Ke et T. Kanade. *Quasiconvex Optimization for Robust Geometric Reconstruction*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pages 1834–1847, 2007.

[Kim 2005] J. Kim, P. Gurdjos et I. Kweon. *Geometric and Algebraic Constraints of Projected Concentric Circles and Their Applications to Camera Calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 4, 2005.

[Klein 2007] G. Klein et D. Murray. *Parallel Tracking and Mapping for Small AR Workspaces*. Dans International Symposium on Mixed and Augmented Reality, 2007.

[Koller 1997] D. Koller, G. Klinker, E. Rose, D. E. Breen, R. T. Whitaker et M. Tuceryan. *Real-time vision-based camera tracking for augmented reality applications*. Dans Virtual Reality Software and Technology, 1997.

[Konolige 2010] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit et P. Fua. *View-based Maps*. International Journal of Robotic Research, vol. 29, no. 8, pages 941–957, 2010.

[Lepetit 2005] V. Lepetit et P. Fua. *Monocular Model-Based 3D Tracking of Rigid Objects : A Survey*. Foundations and Trends in Computer Graphics and Vision, vol. 1, no. 1, 2005.

[Lhuillier 2008] M. Lhuillier. *Automatic scene structure and camera motion using a catadioptric system*. Computer Vision and Image Understanding, vol. 109, no. 2, pages 186–203, 2008.

[LMV] *LibMV*. `http://code.google.com/p/libmv/`.

[Lowe 1992] D. G. Lowe. *Robust model-based motion tracking through the integration of search and estimation*. International Journal of Computer Vision, vol. 8, no. 2, pages 113–122, 1992.

[Lowe 2004] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, 2004.

[Lucas 1985] B. D. Lucas et T. Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision*. Dans International Joint Conference on Artificial Intelligence, 1985.

[Mahamud 2000] S. Mahamud et M. Hebert. *Iterative Projective Reconstruction from Multiple Views*. Dans Conference on Computer Vision and Pattern Recognition, 2000.

[Martinec 2002] D. Martinec et T. Pajdla. *Structure from Many Perspective Images with Occlusions*. Dans European Conference on Computer Vision, 2002.

[Martinec 2005] D. Martinec et T. Pajdla. *3D Reconstruction by Fitting Low-Rank Matrices with Missing Data*. Dans Conference on Computer Vision and Pattern Recognition, 2005.

[Mouragnon 2009] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser et P. Sayd. *Generic and real-time structure from motion using local bundle adjustment*. Image and Vision Computing, vol. 27, no. 8, pages 1178–1193, 2009.

[Muñoz 2005] E. Muñoz, J. M. Buenaposada et L. Baumela. *Efficient Model-Based 3D Tracking of Deformable Objects*. Dans International Conference on Computer Vision, 2005.

[Naimark 2002] L. Naimark et E. Foxlin. *Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker*. Dans International Symposium on Mixed and Augmented Reality, 2002.

[Newcombe 2011] R. A. Newcombe, S. Lovegrove et A. J. Davison. *DTAM : Dense tracking and mapping in real-time*. Dans International Conference on Computer Vision, 2011.

[Nistér 2004] D. Nistér, O. Naroditsky et J. R. Bergen. *Visual Odometry*. Dans Conference on Computer Vision and Pattern Recognition, 2004.

[Nistér 2006] D. Nistér et H. Stewénius. *Scalable Recognition with a Vocabulary Tree*. Dans Conference on Computer Vision and Pattern Recognition, 2006.

[OCV] *OpenCV*. `http://opencv.org`.

[Okumura 2006] B. Okumura, M. Kanbara et N. Yokoya. *Augmented reality based on estimation of defocusing and motion blurring from captured images*. Dans International Symposium on Mixed and Augmented Reality, 2006.

[Oliensis 2007] J. Oliensis et R. I. Hartley. *Iterative Extensions of the Sturm/Triggs Algorithm : Convergence and Nonconvergence*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pages 2217–2233, 2007.

[Pix] *PixelFarm*. `http://www.thepixelfarm.co.uk`.

[Poelman 1997] C. J. Poelman et T. Kanade. *A Paraperspective Factorization Method for Shape and Motion Recovery*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 3, pages 206–218, 1997.

[Pollefeys 1999] M. Pollefeys, R. Koch et L. J. Van Gool. *Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters*. International Journal of Computer Vision, vol. 32, no. 1, pages 7–25, 1999.

[Pollefeys 2004] M. Pollefeys, L. J. Van Gool, M.n Vergauwen, F. Verbiest, K. Cornelis, J. Tops et R. Koch. *Visual Modeling with a Hand-Held Camera*. International Journal of Computer Vision, vol. 59, no. 3, pages 207–232, 2004.

[Ponce 2005] J. Ponce, K. McHenry, T. Papadopoulo, M. Teillaud et B. Triggs. *On the Absolute Quadratic Complex and Its Application to Autocalibration*. Dans Conference on Computer Vision and Pattern Recognition, 2005.

[Poupyrev 2000] I. Poupyrev, H. Kato et M. Billinghurst. *ARToolKit*. `http://www.hitl.washington.edu/artoolkit`, 2000.

[Pressigout 2007] M. Pressigout et E. Marchand. *Real-time Hybrid Tracking using Edge and Texture Information*. International Journal of Robotic Research, vol. 26, no. 7, pages 689–713, 2007.

[Quan 1996] L. Quan. *Self-calibration of an affine camera from multiple views*. International Journal of Computer Vision, vol. 19, no. 1, pages 93–105, 1996.

[Rea] *RealViz Match Mover*. `http://www.realviz.com`.

[Reitmayr 2007] G. Reitmayr et T. Drummond. *Initialisation for Visual Tracking in Urban Environments*. Dans International Symposium on Mixed and Augmented Reality, 2007.

[Rekimoto 1998] J. Rekimoto. *Matrix : A Realtime Object Identification and Registration Method for Augmented Reality*. Dans Asia-Pacific Conference on Computer Human Interaction, 1998.

[Ronda 2008] J. I. Ronda, A. Valdès et G. Gallego. *Line Geometry and Camera Autocalibration*. Journal of Mathematical Imaging and Vision, vol. 32, no. 2, pages 193–214, 2008.

[Rother 2002] C. Rother, S. Carlsson et D. Tell. *Projective Factorization of Planes and Cameras in Multiple Views*. Dans International Conference on Pattern Recognition, 2002.

[Rothganger 2006] F. Rothganger, S. Lazebnik, S. Schmid et J. Ponce. *3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints*. International Journal of Computer Vision, vol. 66, no. 3, pages 231–259, 2006.

[Se 2005] S. Se, D. G. Lowe et J. J. Little. *Vision-based global localization and mapping for mobile robots*. IEEE Transactions on Robotics, vol. 21, no. 3, pages 364–375, 2005.

[Semple 1952] J. G. Semple et G. T. Kneebone. Algebraic Projective Geometry. Oxford University Press, 1952.

[Seo 2004] Y. Seo et A. Heyden. *Auto-calibration by linear iteration using the DAC equation*. Image and Vision Computing, vol. 22, no. 11, pages 919–926, 2004.

[Sim] *Reality Motion Systems*. `http://www.simi.com`.

[Simon 2002] G. Simon et M. Berger. *Pose Estimation for Planar Structures*. IEEE Computer Graphics and Applications, vol. 22, no. 6, pages 46–53, 2002.

[Skrypnyk 2004] I. Skrypnyk et D. G. Lowe. *Scene Modelling, Recognition and Tracking with Invariant Image Features*. Dans International Symposium on Mixed and Augmented Reality, 2004.

[Slama 1980] C. C. Slama, C. Theurer et S. W. Henriksen. Manual of photogrammetry. Falls Church, Va. American Society of Photogrammetry, 1980.

[State 1996] A. State, G. Hirota, D. T. Chen, W. F. Garrett et M. A. Livingston. *Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking*. Dans SIGGRAPH, 1996.

[Sturm 1996] P. F. Sturm et B. Triggs. *A Factorization Based Algorithm for Multi-Image Projective Structure and Motion*. Dans European Conference on Computer Vision, 1996.

[Sturm 1997] P. F. Sturm. Vision 3d non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage. 1997.

[Sturm 1999] P. F. Sturm et S. J. Maybank. *On Plane-Based Camera Calibration : A General Algorithm, Singularities, Applications*. Dans Conference on Computer Vision and Pattern Recognition, 1999.

[Sturm 2000] P. F. Sturm. *Algorithms for Plane-Based Pose Estimation*. Dans Conference on Computer Vision and Pattern Recognition, 2000.

[Syn] *Ssontech SynthEyes*. `http://www.ssontech.com`.

[Tardif 2007] J. Tardif, A. Bartoli, M. Trudeau, N. Guilbert et S. Roy. *Algorithms for Batch Matrix Factorization with Application to Structure-from-Motion*. Dans Conference on Computer Vision and Pattern Recognition, 2007.

[Teller 2003] S. J. Teller, M. E. Antone, Z. Bodnar, M. Bosse, S. R. Coorg, M. Jethwa et N. Master. *Calibrated, Registered Images of an Extended Urban Area*. International Journal of Computer Vision, vol. 53, no. 1, pages 93–107, 2003.

[Thrun 2002] S. Thrun. *Robotic Mapping : A Survey*. Dans G. Lakemeyer et B. Nebel, editeurs, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.

[Tomasi 1992] C. Tomasi et T. Kanade. *Shape and motion from image streams under orthography : a factorization method*. International Journal of Computer Vision, vol. 9, no. 2, pages 137–154, 1992.

[Torr 1999] P. H. S. Torr, A. W. Fitzgibbon et A. Zisserman. *The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences*. International Journal of Computer Vision, vol. 32, no. 1, pages 27–44, 1999.

[Triggs 1996] B. Triggs. *Factorization Methods for Projective Structure and Motion*. Dans Conference on Computer Vision and Pattern Recognition, 1996.

[Triggs 1997] B. Triggs. *Autocalibration and the absolute quadric*. Dans Conference on Computer Vision and Pattern Recognition, 1997.

[Triggs 1999] B. Triggs, P. F. McLauchlan, R. I. Hartley et A. W. Fitzgibbon. *Bundle Adjustment - A Modern Synthesis*. Dans ICCV Workshop on Vision Algorithms, 1999.

[Ueshiba 2003] T. Ueshiba et F. Tomita. *Plane-based Calibration Algorithm for Multi-camera Systems via Factorization of Homography Matrices*. Dans International Conference on Computer Vision, 2003.

[Vacchetti 2004] L. Vacchetti, V. Lepetit et P. Fua. *Stable Real-Time 3D Tracking Using Online and Offline Information*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 10, pages 1385–1391, 2004.

[Valdés 2004] A. Valdés, J. I. Ronda et G. Gallego. *Linear camera autocalibration with varying parameters*. Dans International Conference on Image Processing, 2004.

[Valdès 2005] A. Valdès et J. I. Ronda. *Camera Autocalibration and the Calibration Pencil*. Journal of Mathematical Imaging and Vision, vol. 23, no. 2, pages 167–174, 2005.

[Valdés 2006] A. Valdés, J. I. Ronda et G. Gallego. *The Absolute Line Quadric and Camera Autocalibration*. International Journal of Computer Vision, vol. 66, no. 3, pages 283–303, 2006.

[Vic] *ViconPeak*. `http://www.vicon.com`.

[Voo] *Voodoo*. `http://www.digilab.uni-hannover.de/docs/manual.html`.

[Wang 2011] G. Wang et Q.M.J. Wu. Guide to Three Dimensional Structure and Motion Factorization. Springer édition, 2011.

[Wiles 2001] C. Wiles, A. Maki et N. Matsuda. *Hyperpatches for 3D Model Acquisition and Tracking*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 12, pages 1391–1403, 2001.

[Xiao 2004] J. Xiao, S. Baker, I. Matthews et T. Kanade. *Real-Time Combined 2D+3D Active Appearance Models*. Dans Conference on Computer Vision and Pattern Recognition, 2004.

[Ying 2007] X. Ying et H. Zha. *An Efficient Method for the Detection of Projected Concentric Circles*. Dans International Conference on Image Processing, 2007.

[Ying 2008] X. Ying et H. Zha. *Efficient detection of projected concentric circles using four intersection points on a secant line*. Dans International Conference on Pattern Recognition, 2008.

[Yuen 1989] H. K. Yuen, J. Illingworth et J. Kittler. *Detecting partially occluded ellipses using the Hough transform*. Image and Vision Computing, vol. 7, no. 1, pages 31–37, 1989.

[Zaharescu 2009] A. Zaharescu et R. Horaud. *Robust Factorization Methods Using a Gaussian/Uniform Mixture Model*. International Journal of Computer Vision, vol. 81, no. 3, pages 240–258, 2009.

[Zhang 1999] Z. Zhang. *Flexible Camera Calibration by Viewing a Plane from Unknown Orientations*. Dans International Conference on Computer Vision, 1999.

[Zhang 2007] G. Zhang, X. Qin, W. Hua, T. Wong, P. Heng et H. Bao. *Robust Metric Reconstruction from Challenging Video Sequences*. Dans Conference on Computer Vision and Pattern Recognition, 2007.