



HAL
open science

Modélisation interactive sous contraintes à partir d'images non-calibrées : Application à la reconstruction tridimensionnelle de bâtiments

Sébastien Cornou

► **To cite this version:**

Sébastien Cornou. Modélisation interactive sous contraintes à partir d'images non-calibrées : Application à la reconstruction tridimensionnelle de bâtiments. Interface homme-machine [cs.HC]. Université Blaise Pascal - Clermont-Ferrand II, 2004. Français. NNT : . tel-00008292

HAL Id: tel-00008292

<https://theses.hal.science/tel-00008292>

Submitted on 28 Jan 2005

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

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

N° d'Ordre : ****

THÈSE

Présentée

DEVANT L'UNIVERSITÉ BLAISE PASCAL

pour obtenir

le grade de DOCTEUR DE L'UNIVERSITÉ BLAISE PASCAL

Électronique et Systèmes

par

Sébastien CORNOU

TITRE DE LA THÈSE :

**Modélisation interactive sous contraintes
à partir d'images non-calibrées :
Application à la reconstruction tridimensionnelle de bâtiments**

COMPOSITION DU JURY :

Table des matières

I	Algorithmes de Reconstruction	17
1	État de l'art : reconstruire un nuage de points libres	19
1.1	Introduction	19
1.2	Formulation du problème	21
1.3	Méthodes linéaires : matrice fondamentale et tenseur trifocal	25
1.3.1	Duo de caméras : estimation de la matrice fondamentale	25
1.3.2	Triplet de caméras : le tenseur trifocal	28
1.3.3	Bilan sur la matrice fondamentale et le tenseur trifocal	30
1.4	Méthode de factorisation	31
1.5	Méthodes d'intersection-résection	33
1.6	Ajustement de faisceaux par minimisation	34
1.6.1	Principe de base	35
1.6.2	Description de la méthode de minimisation	35
1.6.3	Mise en oeuvre et exploitation des particularités de l'ajustement de faisceaux	39
1.6.4	La question du choix de jauge	42
1.6.5	Méthode par alternance	47
1.7	Récapitulatif et comparaison entre ces différentes méthodes	47
2	CACHE_POS : Un nouvel algorithme d'ajustement de faisceaux	51
2.1	Introduction	51
2.2	Des algorithmes dissimulant des paramètres à l'estimation non-linéaire	53
2.2.1	Principe	53
2.2.2	Choix de l'approche	55
2.3	Présentation détaillée de CACHE_POS	56
2.3.1	Codage de CACHE_POS	57
2.3.2	Performance des algorithmes d'estimation de poses	60
2.3.3	Choix implicite du repère de reconstruction	67
2.3.4	Exploitation des structures creuses dans CACHE_POS	67
2.4	Étude expérimentale CACHE_POS	73
2.4.1	Conditions de test	73
2.4.2	Etude de la convergence	75
2.4.3	La qualité de la reconstruction	82
2.4.4	Temps de calcul	85

2.5	Bilan	87
2.6	Applications sur des données réelles	87
2.7	Conclusion	91
II	La reconstruction d'objets structurés	93
3	Etat de l'art sur la reconstruction d'objets structurés	95
3.1	Introduction	95
3.2	Partir d'une reconstruction 3D de primitives élémentaires	97
3.3	Les approches par modèles complexes (type CAO)	99
3.3.1	Une approche par blocs paramétriques	100
3.3.2	Utiliser les parallélépipèdes	100
3.3.3	Une approche probabiliste	100
3.4	La modélisation par contraintes : une solution mixte	101
3.4.1	La reconstruction sous contraintes comme un problème linéaire	102
3.4.2	Description et simplification des contraintes par graphes	104
3.5	Conclusion	104
4	Modélisation et reconstruction d'objets contraints	105
4.1	Introduction	105
4.2	Modélisation d'un objet au moyen de contraintes	105
4.2.1	Choix du type de contraintes	106
4.2.2	Modélisation par contraintes dures	110
4.2.3	Étapes de construction d'un modèle contraint	112
4.2.4	Fusion d'objets contraints	118
4.3	Reconstruction d'objets contraints à partir d'images	119
4.3.1	Les étapes de reconstruction	120
4.3.2	Évaluation et analyse de la méthode proposée	121
4.4	Conclusion	127
III	Applications pratiques	129
5	Application à la reconstruction de bâtiments	131
5.1	Introduction	131
5.2	Spécificités des scènes architecturales	132
5.2.1	Problème du point de vue	132
5.2.2	Problème de localisation des primitives	132
5.3	Exemple sur un bâtiment complexe : le château de Sceaux	136
5.3.1	Présentation de la séquence	136
5.3.2	Processus de reconstruction	138
5.3.3	Résultats	142
5.4	Autres exemples	149
5.5	Conclusion	152

6 Perspectives	153
6.1 Introduction	153
6.2 Utiliser les informations géométriques	154
6.2.1 Le problème est-il suffisamment contraint?	154
6.2.2 Initialisation pertinente du modèle 3D.	156
6.2.3 Synthèse sur l'utilisation de la géométrie	157
6.3 Utiliser des appariements de points homologues	157
6.3.1 Apparier des points homologues entre les images	157
6.3.2 Amélioration simultanée du calibrage et du modèle 3D	163
6.3.3 Faciliter l'opération de fusion de modèles	165
6.4 Exploiter la texture présente dans les images	165
6.4.1 Détecter les incohérences et les occultations	166
6.4.2 Automatiser le passage du modèle ponctuel au modèle surfacique .	166
6.4.3 Extraire efficacement la texture du modèle	167
6.5 Conclusion	168
Conclusion	169
Bibliographie	171
6.5.1 Hypothèses initiales	179
6.5.2 But	179
6.5.3 Sous l'hypothèse de la projection perspective	180
6.5.4 Sous l'hypothèse de la projection orthographique à l'échelle	181
6.5.5 Déroulement de l'algorithme	182
6.5.6 Cas particulier d'un objet planaire	182
6.6 Localisation d'un objet volumique par vision monoculaire	185
6.6.1 Introduction	185
6.6.2 Les mises en correspondance	185
6.6.3 Critère à minimiser	186
6.6.4 Résolution du problème par la méthode de Newton-Raphson . . .	187
6.6.5 Calcul des dérivées partielles	188

Table des figures

0.1	Part de marché par type d'appareils photographiques.	14
1.1	Modèle sténopé.	21
1.2	Paramètres caractéristiques des pixels pris en compte dans le modèle sténopé.	23
1.3	Relations géométriques entre deux vues.	26
1.4	Relations géométriques entre trois vues.	29
1.5	Structure creuse de la matrice A	40
1.6	Représentation de l'ensemble de Newton pour $n=3$	42
1.7	Comparaison des différentes méthodes de reconstruction.	49
2.1	Exemple de scène de test	56
2.2	Evolution du critère au cours des itérations.	57
2.3	Schéma de l'algorithme de Levenberg-Marquardt	61
2.4	Erreur sur la position du centre optique en fonction du bruit sur la mire 3D.	62
2.5	Erreur sur l'orientation de la caméra en fonction du bruit sur la mire 3D.	62
2.6	Erreur sur la position du centre optique en fonction du bruit sur les points 2D.	64
2.7	Erreur sur l'orientation de la caméra en fonction du bruit sur les points 2D.	64
2.8	Erreur sur la position du centre optique en fonction de l'erreur sur la longueur focale.	66
2.9	Erreur sur l'orientation de la caméra en fonction de l'erreur sur la longueur focale.	66
2.10	Structure creuse de la matrice quasi-hessienne pour la méthode classique et la méthode CACHE_POS.	72
2.11	Evolution du critère en fonction du nombre d'itérations pour les différentes approches.	73
2.12	Taux de succès en fonction de l'erreur 3D initiale	76
2.13	Taux de convergence en fonction de la longueur focale initiale.	79
2.14	critère en fonction des valeurs de la longueur focale	80
2.15	Erreurs 3D finales en fonction du bruit 2D.	82
2.16	Erreur 2D à la convergence en fonction du bruit 2D	83
2.17	Distribution des écarts entre la mesure moyenne et chacune des mesures.	84
2.18	Temps de calcul en fonction du bruit 2D et de l'erreur 3D initiale.	85
2.19	Les six images disponibles.	89
2.20	Les longueurs mesurées et la distance de référence.	89

2.21	Projections initiales des segments dans une image.	90
2.22	Projection finale des segments après convergence.	90
4.1	Entrée/Sortie au niveau du modèle 3D.	110
4.2	Deux stratégies : fournir les contraintes sans les ordonner ou imposer une procédure arborescente?	112
4.3	Relations entre les sommets contraints	113
4.4	Ensemble des contraintes géométriques définies.	116
4.5	Modélisation d'un parallélépipède rectangle	117
4.6	Exemple de contrainte de fusion incompatible avec les contraintes internes au modèle 2.	118
4.7	Diagramme présentant l'interaction entre les différents éléments impliqués dans la reconstruction.	120
4.8	Exemple de modèles.	122
4.9	Les différentes étapes d'une convergence	124
4.10	Reconstruction binoculaire : données	125
4.11	Reconstruction mono-image : données	125
4.12	Comparaison entre les reconstructions contraintes ou non	126
5.1	Contraintes sur les positions de prise de vue.	134
5.2	Exemple d'occultations	134
5.3	Façade du château de Sceaux.	135
5.4	La mairie de Londres.	135
5.5	Les images de la séquence du Château de Sceaux.	137
5.6	Zoom sur la texture du modèle 3D de la fenêtre.	139
5.7	Exemple de la fenêtre du rez-de-chaussée.	139
5.8	Projection d'un modèle 3D dans les images	140
5.9	Autres éléments de détail reconstruits.	140
5.10	Fusion des différents éléments de détail afin d'obtenir un modèle global.	141
5.11	Projection du modèle en fil de fer sur une image globale.	143
5.12	Modèle surfacé mais non texturé.	143
5.13	Détail du modèle filaire reprojété	144
5.14	Projection du modèle fil de fer de l'avancée dans une image.	145
5.15	Modèle représenté en fil de fer.	146
5.16	Modèle représenté en fil de fer.	146
5.17	Modèle texturé.	147
5.18	Modèle texturé.	147
5.19	Modèle texturé.	148
5.20	Modèle texturé.	148
5.21	Reconsruction de la maison bretonne	150
5.22	La reconstruction de l'église de Dirac	151
6.1	Test de visibilité	158
6.2	Relation homographique entre la zone de référence et la zone d'évaluation.	159
6.3	Relation entre l'aspect et la covariance.	161

6.4	Evolution de l'aspect en cours d'appariement	161
6.5	Nuage de points 3D extraits par triangulation	162
6.6	Etude de la cohérence de la texture	167

Liste des tableaux

2.1	Conditions de test de l'influence du modèle 3D initial et du bruit 2D sur les positions dans les images sur le taux de succès.	76
2.2	Influence du bruit 2D dans les images sur le taux de succès	76
2.3	Conditions de test de l'influence du nombre de points sur le taux de succès.	78
2.4	Influence du nombre de points sur le taux de succès	78
2.5	Conditions de test de l'influence du taux de visibilité sur le taux de succès.	78
2.6	Taux de convergence en fonction du taux de visibilité.	78
2.7	Conditions de test de l'influence de la focale sur le taux de succès.	79
2.8	Conditions de test de l'influence de l'initialisation du modèle 3D sur le taux de succès.	80
2.9	Conditions de test du temps de calcul en fonction du bruit 2D et de l'erreur 3D initiale.	85
2.10	Temps de calcul en fonction du nombre de caméras.	86
2.11	Résultats obtenus sur une séquence présentant de fortes variations de longueur focale entre les différentes prises de vue.	87
2.12	Distances estimées et erreurs.	88
4.1	Comparaison entre la modélisation par contraintes et la modélisation sans contrainte.	122
4.2	Nombre d'itérations requis pour obtenir une convergence correcte.	122
4.3	Définition d'un parallélépipède.	126
4.4	Ajout de points coplanaires.	126
5.1	Données caractéristiques de la reconstruction de la maison bretonne.	150
5.2	Données caractéristiques de la reconstruction d'une partie de l'église de Dirac.	151
6.1	Exemple de modèle non estimable.	155

Notations

Eléments généraux

ϵ : epsilon machine, plus petite valeur que peut manipuler l'ordinateur (dépend du type de la variable).

Eléments géométriques

P, P_{3D} : Point 3D.

P_i : Point 3D numéro i .

p, p_{2D}, p_{image} : point localisé dans une image.

$p_{i,j}$: projection du point 3D i dans l'image j .

L : droite 3D.

L_i : Droite 3D numéro i .

l, l_{2D}, l_{image} : droite localisée dans une image.

$l_{i,j}$: projection de la droite i dans l'image j .

Eléments liés à une caméra

Γ : Matrice de projection.

Γ_i : Matrice de projection associée à la caméra i .

E : Matrice extrinsèque (attitude du capteur).

E_i : Matrice extrinsèque associée à la caméra i .

I : Matrice intrinsèque (paramètres internes du capteur).

I_i : Matrice intrinsèque associée à la caméra i .

C_i : Centre optique de la caméra i .

Eléments liés à plusieurs caméras

$F_{i \rightarrow j}$: Matrice fondamentale exprimant la relation entre deux points homologues p^i et p^j telle que $\tilde{p}^j F_{i \rightarrow j} \tilde{p}^i = 0$.

e_{ij} : Epipole, projection du centre optique de la caméra j dans l'image i .

Eléments mathématiques

\tilde{v} : Élément v exprimé en coordonnées homogènes.

$v_{[l*c]}$: Matrice de l lignes et c colonnes.

$v_{[l]}$: Vecteur de l lignes.

v^T : Transposé du vecteur v .

R : Matrice de rotation.

T : Vecteur de translation.

Id : Matrice identité.

$\det(v)$: déterminant de v .

$v^{[i]}$: i -ième ligne de v .

$v^{[\cdot, i]}$: i -ième colonne de v .

\wedge : produit vectoriel.

Introduction

L'essor de la vision par ordinateur est directement lié à la formidable révolution technique que nous connaissons depuis la fin de la seconde guerre mondiale. A l'image des bouleversements considérables provoqués par l'avènement de la machine à vapeur au XIXème siècle, la physique quantique, l'électronique, l'informatique ont modifié nos sociétés en rendant non seulement possibles, mais courantes et presque communes des prouesses techniques incroyables. Qui aurait cru, il y a cinquante ans, que nous serions aujourd'hui capables de faire tenir sur quelques centimètres carrés une puissance de calcul supérieure à celle qui nécessitait une pièce entière à l'époque ? Les progrès techniques effectués ces dernières décennies sont désormais complètement intégrés dans notre mode de vie à un point tel qu'il est possible de se demander si la technologie n'est pas devenue le ferment de notre civilisation. Que ce soit le vol d'un avion, la gestion du trafic ferroviaire, la détection des tumeurs, le fonctionnement d'une automobile, et la plupart des autres activités humaines, quasiment tout intègre des éléments électroniques orchestrés au moyen de logiciels auxquels nous demandons toujours plus d'intelligence, de souplesse et de fiabilité. Nous sommes les acteurs, les promoteurs, les bénéficiaires, et lorsque surviennent les inévitables pannes, les victimes de cette dépendance aux hautes technologies.

Au milieu de ces grands bouleversements, la vision par ordinateur vit, à son échelle, sa révolution. Pour disposer d'ordinateurs plus conviviaux, plus performants, plus subtils dans leur interprétation d'une situation donnée, il faut les doter de sens. Le goût, le toucher et l'odorat en sont encore au développement des capteurs permettant de caractériser l'environnement. Au contraire, les ordinateurs peuvent d'ores et déjà entendre par le biais de microphones et voir au moyen de caméras. Cette capacité de voir est décuplée depuis quelques années du fait de l'invasion des foyers par les technologies numériques. D'un point de vue économique, la figure 0.1 montre la formidable conquête du marché effectuée par le numérique au dépend des appareils photographiques argentiques. Bon nombre de foyers possède désormais un ordinateur et un capteur d'images numérique (WebCam, appareils photographiques, caméscopes), cet ensemble constituant souvent pour le chercheur en vision le matériel complet nécessaire à la mise en oeuvre de ses algorithmes. Au delà de l'aspect pratique de ce nouveau type d'appareils, les images acquises par ces capteurs constituent des éléments facilement utilisables et contenant des informations très riches sur la scène photographiée. Toute la difficulté et tout l'enjeu des recherches actuelles en vision est d'exploiter ces images afin d'en extraire des informations pertinentes et d'offrir de nouveaux outils utilisables par les industriels ou par le grand public.

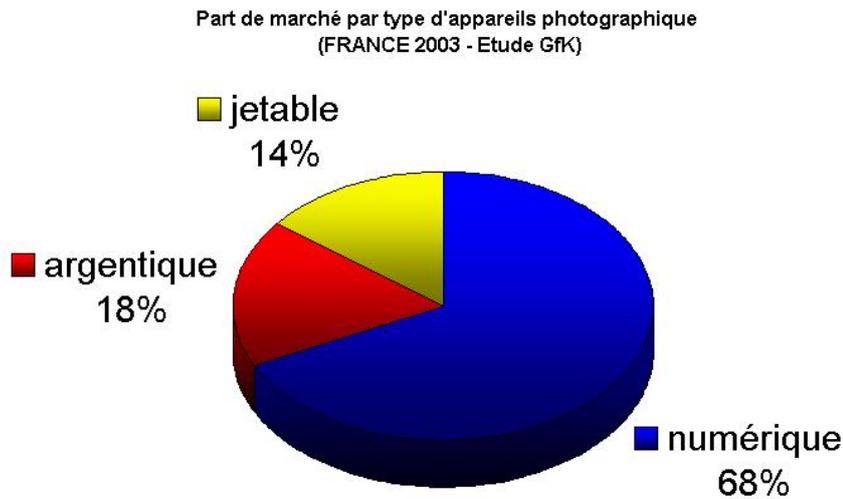


FIG. 0.1: Part de marché par type d'appareils photographiques.

Parmi les grandes diversités des informations que l'on peut souhaiter extraire d'une ou de plusieurs images, il y a l'agencement tridimensionnel de la scène observée. Les applications possibles pour les méthodes de reconstruction de modèles 3D à partir d'images sont nombreuses et, sans être exhaustif, nous pouvons en citer quelques unes :

- la reconstruction d'environnements industriels de sorte à connaître l'état réel d'un site à un instant donné,
- la reconstruction de monuments détruits du patrimoine, par exemple les Bouddhas géants de Bamiyan (cf. <http://www.photogrammetry.ethz.ch/research/bamiyan/>), en vue de leur reconstruction,
- la mise à disposition de modèles tridimensionnels d'objets culturels afin de faciliter leur étude (musées virtuels, ...) (cf. <http://www.arco-web.org>),
- la simulation d'éclairage de monument [Berger 96],
- la saisie du modèle 3D du visage d'un joueur de jeu vidéo afin d'accroître encore son immersion dans le jeu,
- la reconstruction métrologique intégrale de pièces manufacturées afin d'augmenter les performances et la fiabilité des produits,
- la reconstruction simple d'objets courants afin de favoriser le commerce électronique,
- et encore de nombreuses autres applications possibles...

Dans le cadre de cette thèse, nous avons étudié la modélisation d'objets structurés au moyen d'images non-calibrées. Un objet structuré est un objet présentant des particularités qui permettent de le décrire facilement à partir d'éléments géométriques ou

physiques (une boîte, un cylindre, la tour Eiffel,...). A l'opposé, un objet non structuré est, a priori, difficile à définir à partir de quelques règles simples (le ciel, la mer,...).

Notre objectif était de proposer une solution permettant à un utilisateur de modéliser un objet en utilisant à la fois ses connaissances sur la scène observée et des informations extraites d'une séquence d'images obtenues au moyen d'un appareil photographique. Afin de garantir la souplesse d'utilisation et de permettre l'usage d'images dont les caractéristiques de prise de vue sont totalement inconnues (images récupérées sur internet, archives,...), nous souhaitons inscrire notre approche dans le cadre de la reconstruction non-calibrée.

Le résultat de nos travaux est une méthode interactive complète de modélisation à partir d'images non-calibrées. La méthode développée dans cette thèse permet à l'utilisateur de modéliser tout d'abord l'objet au moyen de contraintes, puis de retrouver son modèle tridimensionnel à partir de la localisation des sommets 3D dans les différentes images disponibles. Pour parvenir à ce résultat, le système effectue un ajustement de faisceaux qui estime les dimensions de la scène observée sans requérir l'initialisation de la pose des images.

Ce document comporte trois parties :

1. **algorithmes de reconstruction** : cette première partie traite le cas canonique de la reconstruction d'un nuage de points libres de toutes contraintes. Nous introduisons, à la lumière de ce problème particulier, les méthodes classiques de reconstruction par vision, puis nous présentons l'algorithme d'ajustement de faisceaux, nommé `CACHE_POS`, que nous avons développé.
2. **reconstruction d'objets structurés** : nous introduisons les méthodes générales de reconstruction utilisant non seulement les informations obtenues au moyen des images, mais aussi les connaissances introduites par l'utilisateur sur la structure de la scène. Puis, après avoir présenté les différentes méthodes de reconstruction existantes, nous présentons la méthode de modélisation des objets 3D que nous avons développée, et nous montrons que ce type de représentation est compatible avec les techniques de reconstruction évoquées dans la partie précédente.
3. **application pratiques** : nous montrons, à travers un premier chapitre, les résultats que nous avons obtenus sur des séquences réelles d'images représentant des bâtiments photographiés avec un appareil photographique numérique. Puis, dans un second chapitre, nous présentons des perspectives d'améliorations qui semblent souhaitables afin d'améliorer l'efficacité, d'accroître la précision et simplifier l'utilisation de notre méthode...

Première partie

Algorithmes de Reconstruction

Chapitre 1

État de l'art : reconstruire un nuage de points libres

Sommaire

1.1	Introduction	19
1.2	Formulation du problème	21
1.3	Méthodes linéaires : matrice fondamentale et tenseur trifocal	25
1.3.1	Duo de caméras : estimation de la matrice fondamentale	25
1.3.2	Triplet de caméras : le tenseur trifocal	28
1.3.3	Bilan sur la matrice fondamentale et le tenseur trifocal	30
1.4	Méthode de factorisation	31
1.5	Méthodes d'intersection-résection	33
1.6	Ajustement de faisceaux par minimisation	34
1.6.1	Principe de base	35
1.6.2	Description de la méthode de minimisation	35
1.6.3	Mise en oeuvre et exploitation des particularités de l'ajustement de faisceaux	39
1.6.4	La question du choix de jauge	42
1.6.5	Méthode par alternance	47
1.7	Récapitulatif et comparaison entre ces différentes méthodes	47

1.1 Introduction

Au cours de ce chapitre, nous allons présenter différentes techniques permettant de retrouver la position de points 3D libres, observés dans plusieurs images. Les points 3D sont indépendants les uns des autres (absence de contraintes), et chacun d'entre eux est observé dans au moins deux images. Ce cas canonique a été très largement étudié et de nombreux algorithmes ont été proposés pour tenter de le résoudre. Lors de la présentation initiale de chaque méthode, les faux-appariements ne seront pas considérés, mais compte tenu de leur existence quasi-systématique, nous indiquerons leur influence

et les techniques disponibles afin de minimiser leur impact sur le résultat (voir la synthèse 1.7).

Le problème est traité du point de vue de l'auto-calibrage. Il s'agit de définir la position des points 3D ainsi que la position, l'orientation et les paramètres internes (longueurs focales, ...) des caméras à partir de la seule connaissance des projections des points dans les images. Afin d'obtenir un tel résultat, le système de mesure fait appel à une succession de procédures et d'algorithmes dont la mise en oeuvre définit le processus de mesure. Dans le cas de la reconstruction auto-calibrée à partir d'images, le processus de mesure est particulier, car il vise à estimer simultanément les dimensions de la scène observée et les propriétés des capteurs employés. Ce problème de reconstruction de nuage de points 3D à partir d'images est au coeur d'un grand nombre d'applications en vision artificielle (calibrage, métrologie industrielle, reconstruction temps réel à partir de vidéo, modélisation rapide pour internet...) et a donné lieu à de nombreuses publications. Ici, nous nous sommes principalement appuyés sur deux livres [Faugeras 01] et [Hartley 00] qui dressent un état de l'art récent sur les problèmes de reconstruction multi-vues. Le choix et la mise en oeuvre de ces techniques ne peuvent se faire qu'en réponse à un problème spécifique donné, néanmoins nous avons défini quatre critères afin de simplifier l'identification des qualités et des défauts de ces différentes techniques.

- **La précision** : C'est l'écart entre la scène réelle et le modèle 3D obtenu à partir des images. Dans la pratique cette écart se mesure souvent par la distance entre les primitives 2D localisées dans les images et les projections des primitives dans ces images. L'erreur estimée ainsi traduit simultanément les erreurs faites sur les modèles des capteurs et les erreurs sur la modélisation de l'objet 3D observé. Dans la pratique, cette mesure est la seule accessible mais, lors d'évaluation d'algorithmes sur des séquences de synthèse, nous effectuons l'évaluation de l'erreur sur la distance entre le modèle 3D parfait et le modèle reconstruit.
- **La rapidité** : C'est la capacité à fournir rapidement une évaluation de la scène 3D. La rapidité est parfois un critère important. La notion de rapidité a un sens différent selon l'application visée mais peut parfois être un élément critique
- **La généralité** : Cette propriété caractérise les conditions d'emploi de la méthode. Certains algorithmes requièrent, par exemple, que tout les points soient visibles dans toutes les images. Ce type de contraintes conditionne la classe de problèmes solubles ou non par une méthode donnée et, par conséquent, son caractère général.
- **La robustesse** : Comme nous l'avons déjà évoqué, les applications en vision sont fréquemment confrontées à des erreurs d'appariements. La résistance et la capacité à prendre en compte de telles erreurs sont des critères importants (notamment dans le cas de processus tout automatique), car elles conditionnent le succès de la reconstruction (précision, convergence,...).

1.2 Formulation du problème

Dans le cadre du problème de reconstruction d'un nuage de points 3D libres nous considérons une scène constituée d'un ensemble de N points 3D et de k vues. Les mesures disponibles correspondent à la position de la projection des points 3D dans chacune des images (chaque projection est associée à un point 3D particulier). L'objectif est de retrouver (à un facteur d'échelle près) l'ensemble de la structure tridimensionnelle du nuage de points à partir de ces observations.

La première difficulté à surmonter est l'appariement de points homologues entre les différentes images (projections d'un même point 3D). Cette opération d'appariement de points homologues présente deux écueils principaux :

- le premier est tout simplement l'erreur d'appariement provoquée, soit par une mauvaise identification du point lors de la saisie manuelle, soit par une erreur lors d'un processus d'appariement automatique (erreur de suivi d'un point d'intérêt dans un séquence vidéo, plusieurs points présentant des signatures visuelles proches entraînant des confusions...),
- le second écueil est dû aux imprécisions de localisation des projections. En règle générale, les méthodes de sélection manuelle dépassent rarement des précisions de l'ordre du pixel alors que les processus d'appariement automatique peuvent parfois atteindre des précisions de l'ordre du centième de pixel lorsque des primitives spécifiques sont utilisées (cibles réfléchissantes, ...).

La prise en compte de ces deux phénomènes conditionne la qualité du résultat final, mais ce n'est pas suffisant. En effet, disposer de mesures de qualité est fondamental, mais reconstruire une scène 3D implique aussi de définir un modèle représentant la scène 3D observée et les capteurs utilisés. Le type de modélisation choisi conditionne profondément les propriétés et la qualité du résultat final. Dans notre problème, deux types de modèles sont définis : les modèles de caméras et le modèle du nuage de points 3D.

Pour les caméras, un modèle fréquemment employé est le modèle sténopé (modèle dit en "trou d'épingle") (cf. figure 1.1).

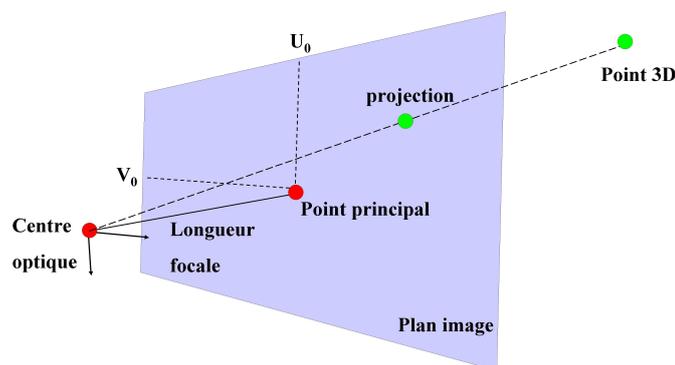


FIG. 1.1: Modèle sténopé.

Dans ce modèle, la projection d'un point 3D dans l'image est l'intersection entre la droite passant par le point 3D et le centre optique de la caméra, et le plan image. Ce modèle a l'avantage de décrire la caméra par une matrice (appelée matrice de projection) qui traduit une relation directe entre les points 3D, les paramètres de la caméra et la projection 2D. Cette matrice de projection Γ peut être décomposée en deux matrices :

- **la matrice intrinsèque** : cette matrice traduit la projection d'un point 3D exprimée dans le repère caméra sur le plan image. Cette matrice est dite "intrinsèque" car ses paramètres décrivent les propriétés du capteur considéré. Une caméra est constituée de deux éléments principaux, des lentilles formant l'objectif et un plan constituant le capteur qui va enregistrer le signal lumineux qu'il reçoit. Le rôle de l'objectif est de focaliser les rayons lumineux afin d'obtenir une image de la scène au niveau du capteur. Ce capteur est constitué d'éléments sensibles nommés pixels. Nous considérons ici le cas des capteurs réalisés à partir d'une rétine CCD dotée de l lignes et c colonnes de pixels. Cette matrice CCD est caractérisée par la largeur dx et la hauteur dy de chaque pixel ainsi que par l'angle d'inclinaison θ correspondant à l'angle entre les lignes et les colonnes de la matrice CCD. Ces différents éléments sont décrits sur la figure 1.2. Le modèle sténopé représente le processus de construction de l'image par une projection perspective. Dans ce modèle, une longueur caractéristique est la distance entre le centre de projection (centre optique) et le plan image (capteur CCD), cette distance est nommée longueur focale f . Le projeté orthogonal du centre optique sur le plan image a pour coordonnées (u_0, v_0) et est nommé point principal. Lorsque le point principal n'est pas connu, il est supposé au centre de l'image $(l/2, c/2)$

La matrice intrinsèque s'écrit ainsi :

$$I = \begin{pmatrix} \frac{f}{dx} & \tan(\frac{\pi}{2} - \theta) \frac{f}{dy} & u_0 \\ 0 & \frac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.1)$$

I traduit la projection d'un point 3D P_{camera} exprimé en coordonnées homogènes dans le repère caméra en un point p_{image} du plan image (également exprimé en coordonnées homogènes).

$$p_{image} = IP_{camera} \quad (1.2)$$

Calibrer le capteur signifie calculer les paramètres f , dx , dy , θ , u_0 et v_0 de la matrice intrinsèque. En pratique, l'angle θ est proche de 90° et par la suite nous considérerons $\theta = 90^\circ$ ce qui implique $\tan(\frac{\pi}{2} - \theta) = 0$. La matrice intrinsèque est définie à un facteur d'échelle près, par conséquent tout les paramètres restant f , dx , dy ne peuvent être calculés. Afin de pouvoir calibrer le capteur à un facteur d'échelle près, nous introduisons les variables $f_x = \frac{f}{dx}$ et $f_y = \frac{f}{dy}$ qui traduisent la longueur focale en fonction de la taille des pixels dans les directions verticale et horizontale. La matrice intrinsèque que nous utiliserons par la suite s'écrit donc :

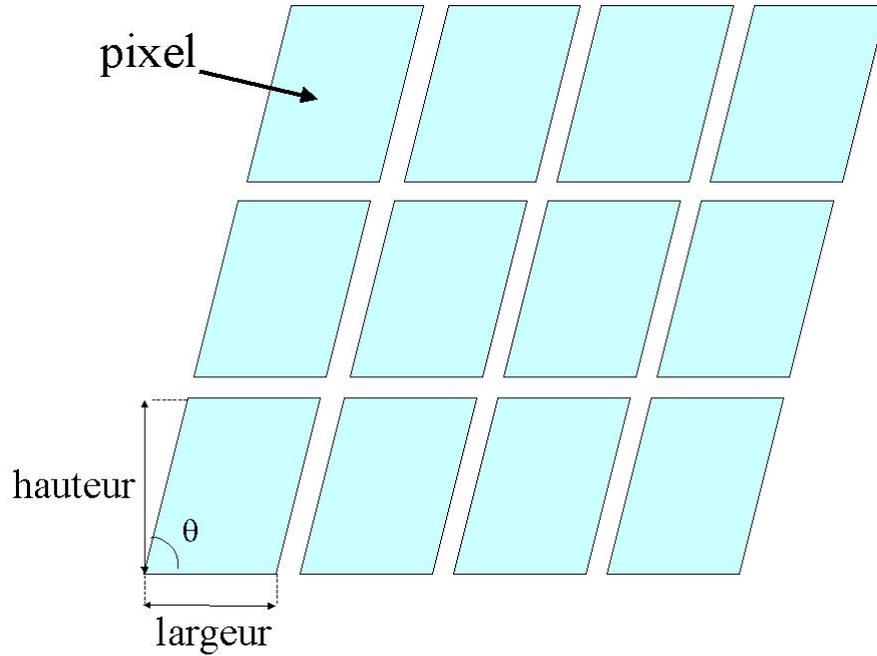


FIG. 1.2: Paramètres caractéristiques des pixels pris en compte dans le modèle sténopé.

$$I = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.3)$$

- **la matrice extrinsèque** : cette matrice traduit la conversion des coordonnées des points 3D du repère associé au nuage de points à un repère local associé à la caméra. Le repère caméra est défini avec pour centre le centre optique associé au capteur et par trois vecteurs directeurs :
 - \vec{i} : Vecteur unitaire colinéaire aux lignes du plan de pixels.
 - \vec{j} : Vecteur unitaire colinéaire aux colonnes du plan de pixels.
 - \vec{k} : Vecteur unitaire colinéaire aux vecteur centre optique - point principale.

Six paramètres sont employés : trois réels pour le vecteur de translation \vec{T} décrivant la position du centre optique de la caméra dans le repère associé au nuage de points et trois réels α, β, γ traduisant l'orientation de la caméra selon les notations d'*Euler*. La matrice extrinsèque E s'écrit ainsi ($R_{\alpha, \beta, \gamma}$ est la matrice de rotation associée à α, β, γ) :

$$E = \begin{pmatrix} R_{\alpha, \beta, \gamma}^T & -R_{\alpha, \beta, \gamma}^T T \\ 0_{[3]}^T & 1 \end{pmatrix} \quad (1.4)$$

La matrice de projection Γ s'écrit en combinant la matrice intrinsèque et la matrice extrinsèque :

$$\Gamma = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} R_{\alpha,\beta,\gamma}^T & -R_{\alpha,\beta,\gamma}^T T \\ 0_{[3]}^T & 1 \end{pmatrix} \quad (1.5)$$

La projection p_{image} d'un point 3D P_{3D} dans l'image est donné par la relation suivante :

$$p_{image} = \Gamma P_{3D} \quad (1.6)$$

Ce modèle de caméra est très souvent employé du fait de sa facilité d'usage. Toutefois, une représentation plus fine est parfois nécessaire afin d'accroître la qualité des reconstructions. Un complément fréquemment apporté au modèle sténopé consiste à modéliser les distorsions introduites par les lentilles et les problèmes de positionnement des éléments de la caméra (parallélisme entre le "plan" des lentilles et la matrice CCD...). Les distorsions peuvent être modélisées au moyen de polynômes indiquant la nature et l'importance des déformations. L'intégration de ces nouveaux paramètres brise la simplicité de la relation entre la position du point 3D et la projection de ce point dans l'image. La prise en compte des distorsions dépend, par conséquent, de la précision des informations 2D disponibles, de la nature, de la qualité des objectifs utilisés, des besoins, ...

La modélisation des éléments 3D présents dans la scène est également une partie délicate. Dans le cas d'un nuage de points 3D libres de toutes contraintes, la modélisation intuitive (et pratique !) consiste à décrire chaque point par ses coordonnées exprimées dans un repère cartésien. Dans la suite de cette thèse, lors de l'introduction de contraintes dans le nuage de points, la modélisation du modèle deviendra un point majeur. Quoiqu'il en soit, le choix du modèle 3D n'est jamais anodin et a des conséquences importantes sur la qualité de la reconstruction.

En résumé, ce problème de reconstruction de points libres est symptomatique des difficultés rencontrées dans les applications de reconstruction par vision. D'une part, les données disponibles sous la forme de primitives détectées dans les images sont bruitées et parfois erronées. D'autre part, la modélisation des capteurs et des éléments 3D de la scène implique des approximations de la réalité qui se traduisent par des erreurs plus ou moins acceptables sur la solution finale. Enfin, les processus informatiques de reconstruction ainsi que les méthodes mathématiques sous-jacentes peuvent intrinsèquement générer des difficultés liées aux arrondis, aux problèmes de convergence, aux résultats numériquement exactes mais physiquement faux (par exemple, on peut parfois converger vers des solutions plaçant l'objet derrière la caméra...).

Dans les paragraphes suivants, nous présentons quelques méthodes classiques de reconstruction. Aucune d'entre elles ne constitue une réponse universelle à notre problème, mais elles apportent des solutions aux difficultés rencontrées dans certaines applications. Nous allons présenter les différents algorithmes de reconstruction. La première classe d'algorithmes abordée repose sur le calcul des paramètres du modèle 3D et des caméras au moyen des matrices fondamentales (cas à deux images) ou des tenseurs trifocaux (cas à trois images). La seconde classe recouvre les méthodes qui utilisent une organisation particulière des données permettant de résoudre linéairement le problème en faisant appel

à des techniques de factorisation. Enfin, nous présentons une troisième classe d'algorithmes basée sur la minimisation d'un critère non-linéaire. Ces méthodes, généralement appelées "ajustement de faisceaux" du fait de l'interprétation géométrique qui peut en être donnée, sont fréquemment employées en fin de reconstruction pour améliorer la précision du résultat.

1.3 Méthodes linéaires : matrice fondamentale et tenseur trifocal

La reconstruction de la position de points 3D à partir de deux images, lorsque les conditions de prise de vues sont connues, correspond à la technique des parallaxes. Cette technique est utilisée depuis très longtemps et a permis, par exemple, à Aristarque de Samos d'estimer la distance Terre-Lune en 250 avant Jésus Christ. Elle est toujours au coeur d'un très grand nombre d'applications (relevés topographiques, calcul de la distance des étoiles par le satellite Hipparcos...) et, depuis les années 80, de nombreuses propriétés mathématiques ont été mises en évidence ouvrant la voie à des reconstructions ne nécessitant pas la connaissance préalable du capteur employé. Les très nombreux travaux réalisés dans le passé ont montré l'importance de la géométrie épipolaire, de la matrice fondamentale et du tenseur trifocal ainsi que l'existence de cas dégénérés dans le traitement des problèmes de reconstruction. De nombreuses techniques d'évaluation de ces éléments [Boufama 95] [Deriche 94] [Sturm 97] ont été proposées en fonction du problème considéré et du modèle de caméra choisi. La connaissance de la matrice fondamentale ou du tenseur trifocal suffit alors à effectuer la modélisation complète des capteurs et la reconstruction du modèle 3D à un niveau projectif. L'apport d'informations complémentaires connues, a priori (distance, angle...), permet par la suite de remonter jusqu'à une reconstruction euclidienne ou métrique.

1.3.1 Duo de caméras : estimation de la matrice fondamentale

Nous allons définir pour commencer la matrice fondamentale en mettant en évidence sa relation très étroite avec la géométrie épipolaire. Nous traitons le cas de deux images (i et j) observant un nuage de points. La figure 1.3 présente cette scène avec P le point 3D observé et p_i la projection de P dans l'image i . C_i correspond au centre optique (centre de projection) de la caméra i .

On peut remarquer que les points C_i, C_j, p_i, P, p_j (les indices correspondent au numéro de l'image) forment un plan. Considérons par exemple l'image j , l'intersection de ce plan avec l'image j est une droite passant par p_j et par la projection, appelée épipole, de C_i dans l'image j . Cette droite est nommée droite épipolaire et, sans rentrer dans les détails, on constate immédiatement l'existence de cas particuliers (dégénérés!) lorsque par exemple C_i est sur la droite $(C_j p_i)$.

La matrice fondamentale permet d'exprimer cette structure géométrique par le biais d'une relation mathématique entre \tilde{p}_i et \tilde{p}_j ($F_{j \rightarrow i}$ représente la matrice fondamentale entre une caméra j et une caméra i , la notation $\tilde{\cdot}$ indique l'utilisation de coordonnées

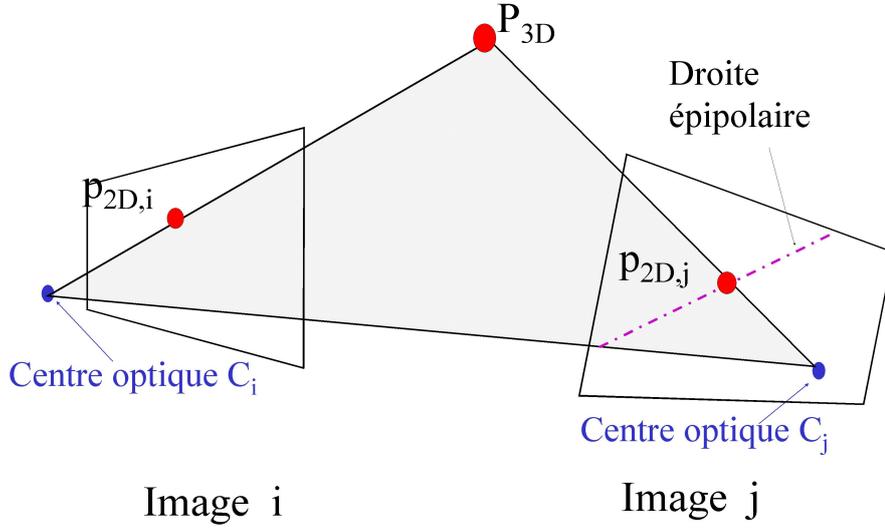


FIG. 1.3: Relations géométriques entre deux vues.

homogènes) :

$$\tilde{p}_i F_{j \rightarrow i} \tilde{p}_j = 0 \quad (1.7)$$

La matrice fondamentale se décompose en utilisant les matrices de projection associées à chacune des images (Γ_i et Γ_j définie par 1.5). La matrice Γ_j^+ est la pseudo-inverse de Γ_j et 1.7 devient donc :

$$\tilde{p}_i \Gamma_i \Gamma_j^+ \tilde{p}_j = 0 \quad (1.8)$$

Cette équation s'interprète géométriquement, la partie droite $\Gamma_j^+ \tilde{p}_j$ représente l'équation projective de la droite $\tilde{L} \cong (C_j p_j)$, le produit $\tilde{l} \cong \Gamma_i \tilde{L}$ est la droite épipolaire correspondant à la projection de la droite $(C_j p_j)$ dans l'image i . Le point \tilde{p}_i appartient à la droite épipolaire \tilde{l} , le produit scalaire $\tilde{p}_i \tilde{l}$ est donc nul.

La relation 1.7 doit être respectée pour toutes les paires de points. Cet ensemble d'équations est l'information majeure permettant d'estimer la matrice fondamentale entre deux caméras, mais les épipoles jouent un rôle particulier qui permettent d'ajouter une nouvelle contrainte sur F . En effet, les épipoles constituent deux noyaux de F (à droite et à gauche) ce qui implique que $\det(F) = 0$. Par conséquent, pour estimer F il faut résoudre le système suivant :

$$\begin{cases} \forall k \in [1 : N], \tilde{p}_i^k F_{j \rightarrow i} \tilde{p}_j^k = 0 \\ \det(F) = 0 \end{cases} \quad (1.9)$$

La présence du déterminant de F brise la linéarité du système par rapport aux éléments de la matrice fondamentale et, par conséquent, complique l'estimation de F . Deux stratégies sont alors possibles, on peut choisir, dans un premier temps, de résoudre

la partie linéaire puis contraindre le déterminant ou de tout résoudre simultanément. Les méthodes décrites dans la littérature apportent désormais des solutions à ce problème.

1.3.1.1 Le cas minimal : sept paires disponibles⁰

Il est possible d'écrire la matrice F sous la forme d'un vecteur colonne \vec{f} . Dès lors la relation 1.7 peut s'écrire sous la forme $A\vec{f} = \vec{0}$.

En notant $\tilde{p}_{k,i} = (x_{k,i}, y_{k,i}, 1)^T$ et $\tilde{p}_{k,j} = (x_{k,j}, y_{k,j}, 1)^T$ les composantes de la projection d'un même point 3D vu dans les images i et j , nous pouvons écrire pour chaque paire de projection :

$$A = \begin{pmatrix} x_{1,i}x_{1,j} & y_{1,i}x_{1,j} & x_{1,j} & x_{1,i}y_{1,j} & y_{1,i}y_{1,j} & y_{1,j} & x_{1,i} & y_{1,i} & 1 \\ \vdots & \vdots \\ x_{k,i}x_{k,j} & y_{k,i}x_{k,j} & x_{k,j} & x_{k,i}y_{k,j} & y_{k,i}y_{k,j} & y_{k,j} & x_{k,i} & y_{k,i} & 1 \end{pmatrix} \quad (1.10)$$

Deux cas peuvent être rencontrés. Soit A est de rang 8 auquel cas le vecteur f peut être estimé à un facteur d'échelle près, soit A est de rang 7 et il faut tirer partie de la contrainte de rang existant sur F . Ce cas est fréquemment rencontré lorsque seulement sept paires de points sont disponibles, la solution à l'équation $A\vec{f} = \vec{0}$ est alors de degré 2 et peut s'écrire sous la forme $\alpha F_1 + (1 - \alpha)F_2$ avec ($\alpha \in \mathfrak{R}$). F_1 et F_2 correspondent évidemment aux deux vecteurs formant une base de l'espace solution de $A\vec{f} = \vec{0}$. Une fois que cette base a été déterminée, il reste à calculer un réel α tel que $\det(F) = \det(\alpha F_1 + (1 - \alpha)F_2)$. Le problème se ramène au calcul des racines d'un polynôme de degré 3. Chaque racine réelle implique une matrice fondamentale candidate, les racines complexes ne sont pas retenues [Hartley 94].

1.3.1.2 L'algorithme à huit points

Si huit paires de points sont disponibles, la solution à un facteur d'échelle près est directement donnée en résolvant $A\vec{f} = \vec{0}$ (une décomposition par SVD peut être utilisée, la solution \vec{f} fournie vérifie alors la contrainte $\|\vec{f}\| = 1$) et cette méthode permet de traiter les cas où plus de huit couples de points sont disponibles. Hartley a montré dans [Hartley 97] une amélioration des performances lorsqu'une normalisation des données est effectuée au préalable. Une méthode de normalisation consiste à normaliser les données dans chaque image (indépendamment les unes des autres) en translatant l'ensemble des points 2D afin de centrer le barycentre et en appliquant un facteur d'échelle pour borner les coordonnées entre $-\sqrt{2}$ et $\sqrt{2}$.

Quoiqu'il en soit l'algorithme à huit points ne contraint pas le déterminant de F à être nul et cette contrainte doit être imposée a posteriori afin d'avoir une estimation de la matrice fondamentale cohérente avec la géométrie de la scène et d'accroître ainsi la qualité du résultat.

⁰Méthode décrite dans [Hartley 00] pages 264.

1.3.1.3 Méthode de minimisation : un critère algébrique

À partir de l'estimation de la matrice fondamentale obtenue par la méthode précédente (par exemple), il est possible d'utiliser une méthode de minimisation afin de déterminer la matrice fondamentale vérifiant au mieux l'équation 1.9. Comme pour toute méthode de minimisation, le choix des paramètres à optimiser conditionne le résultat obtenu. Dans le cas de l'estimation de la matrice fondamentale, l'estimation de chaque coefficient (moins un afin de fixer l'échelle) ne permet pas de contraindre le déterminant de F à être nul. Au contraire, une méthode décomposant F en un produit de deux matrices (une matrice singulière et une matrice antisymétrique dépendant d'un épipole) permet d'estimer F en respectant les contraintes indiquées (cf. [Hartley 00] pages 266-267).

1.3.1.4 Méthode de minimisation : un critère géométrique

Le critère algébrique que nous venons de présenter ne correspond pas à la minimisation d'une distance physique. D'autres méthodes de minimisation existent. Elles reposent sur la minimisation de la distance des points appariés dans une image avec les droites épipolaires leur correspondant. Plusieurs méthodes existent pour fixer le déterminant à 0 :

1. Décomposer F : La première solution est la même que celle présentée dans le paragraphe précédent, il s'agit de décomposer F en un produit d'une matrice singulière et d'une autre matrice pour assurer la singularité de la matrice F estimée. Le problème de cette méthode vient de la sur-paramétrisation qu'elle entraîne qui peut se traduire par des instabilités lors de la minimisation.
2. Créer une relation linéaire entre les colonnes F : une autre solution consiste tout simplement à exprimer une colonne comme une combinaison linéaire des deux autres. Le déterminant de la matrice devient nécessairement nul et huit paramètres suffisent à décrire F (six éléments de la matrice et deux coefficients reliant une colonne aux deux autres). Un problème survient toutefois lorsque les deux colonnes paramétrées sont elles-mêmes linéairement dépendantes. Pour gérer ce problème, il est nécessaire de sélectionner la modélisation adéquate en fonction du cas rencontré. Dans le même ordre d'idée, il est possible d'intégrer un des épipoles ou les deux pour paramétrer la matrice (cf. [Hartley 00] pages 268-271).

1.3.2 Triplet de caméras : le tenseur trifocal

Après avoir traité le cas de deux images, nous nous intéressons maintenant au cas de trois images. Considérons trois images indicées 1, 2, 3 et notons p_i la projection d'un point 3D dans l'image i . Si l'on considère les trois points homologues p_1 , p_2 et p_3 , nous pouvons écrire les relations épipolaires associées à chacun de ces couples :

$$\begin{cases} \tilde{p}_1^T F_{2 \rightarrow 1} \tilde{p}_2 = 0 \\ \tilde{p}_1^T F_{3 \rightarrow 1} \tilde{p}_3 = 0 \\ \tilde{p}_3^T F_{2 \rightarrow 3} \tilde{p}_2 = 0 \end{cases} \quad (1.11)$$

Nous notons e_{ij} (épipole) la projection du centre optique de l'image i dans l'image j . Entre les différents épipoles, nous avons la relation $\tilde{e}_{31}^T F_{2 \rightarrow 1} \tilde{e}_{32} = \tilde{e}_{12}^T F_{3 \rightarrow 2} \tilde{e}_{13} = \tilde{e}_{23}^T F_{1 \rightarrow 3} \tilde{e}_{21} = 0$ qui montrent que les trois contraintes exprimées dans l'équation 1.11 ne sont pas indépendantes. Si les matrices fondamentales sont connues, la connaissance de deux points homologues suffit à déterminer la position du troisième. La figure 1.4 présente la situation d'un point 3D observé dans trois images.

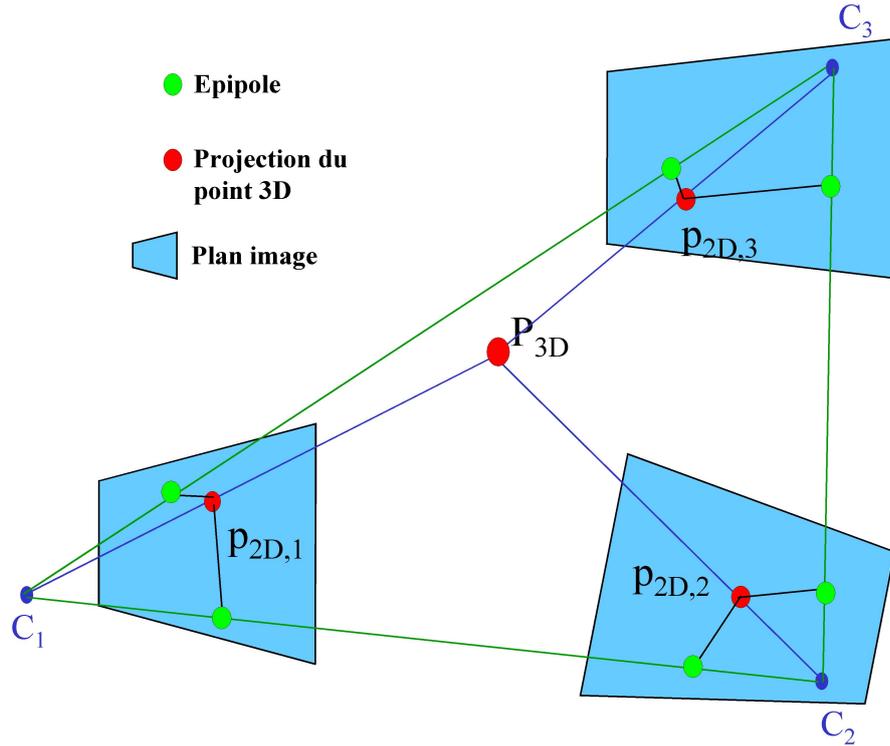


FIG. 1.4: Relations géométriques entre trois vues.

Considérons désormais une droite L observée par ces trois caméras. Soit l_i la projection de L dans l'image i à laquelle est associée la matrice de projection Γ_i . Le plan Π_i passant par le centre optique C_i de la caméra i et la droite l_i vaut $\Pi_i = \Gamma_i^T l_i$. Nous prenons les matrices de projection de la forme $\Gamma_1 = [Id_{[3*3]} 0_{[3]}]$, $\Gamma_2 = [A_{[3*3]} a]$ et $\Gamma_3 = [B_{[3*3]} b]$. Dès lors, nous pouvons écrire l'équation des trois plans Π_1 , Π_2 et Π_3 :

$$\Pi_1 = \Gamma_1^T l_1 = \begin{pmatrix} l_1 \\ 0 \end{pmatrix} \quad (1.12)$$

$$\Pi_2 = \Gamma_2^T l_2 = \begin{pmatrix} A^T l_2 \\ a^T l_2 \end{pmatrix} \quad (1.13)$$

$$\Pi_3 = \Gamma_3^T l_3 = \begin{pmatrix} B^T l_3 \\ b^T l_3 \end{pmatrix} \quad (1.14)$$

En fait, ces trois plans s'intersectent en L la droite 3D correspondant à l_1 , l_2 et l_3 , ce qui implique que leurs équations ne sont pas indépendantes. Cette intersection s'exprime sous la forme de la matrice $M = [\Pi_1\Pi_2\Pi_3]$ qui doit être de rang 2.

$$M = [\Pi_1\Pi_2\Pi_3] = \begin{pmatrix} l_1 & A^T l_2 & B^T l_3 \\ 0 & a^T l_2 & b^T l_3 \end{pmatrix} \quad (1.15)$$

Par conséquent, les colonnes de M sont linéairement dépendantes et l'on peut écrire $M^{[\cdot,1]} = \alpha M^{[\cdot,2]} + \beta M^{[\cdot,3]}$. En remarquant le zéro de la dernière composante de $M^{[\cdot,1]}$ on en déduit immédiatement $\alpha = kb^T l_3$ et $\beta = ka^T l_2$. Le paramètre k est un scalaire quelconque que nous fixons à $k = 1$. Dès lors, connaissant α et β , nous pouvons calculer toutes les composantes de $M^{[\cdot,1]}$ qui sont en fait les coordonnées de l_1 :

$$l_1^{[i]} = l_3^T (bA^{[\cdot,i],T})l_2 - l_2^T (aB^{[\cdot,i],T})l_3 = l_2^T (A^{[\cdot,i]}b^T)l_3 - l_3^T (aB^{[\cdot,i],T})l_2 \quad (1.16)$$

en introduisant la notation $T_i = A^{[\cdot,i]}b^T - aB^{[\cdot,i],T}$, la relation entre les projections de L dans chacune des images devient :

$$l_1^{[i]} = l_2 T_i l_3 \quad (1.17)$$

Les trois matrices $[T_{[3*3],1} T_{[3*3],2} T_{[3*3],3}]$ constituent le tenseur trifocal. Elles définissent les relations existant entre les trois images. Le tenseur trifocal est défini par vingt-six éléments car ce tenseur est défini à un facteur d'échelle près ($3 \times 3 \times 3 - 1$). Cependant, ce tenseur est contraint par la situation qu'il représente (tout ensemble de trois matrices ne constitue pas nécessairement un tenseur). Si l'on considère les degrés de liberté de la scène 3D, chaque matrice de projection apporte 11 paramètres libres ($3 \times 11 = 33$) auquel il faut supprimer 15 paramètres traduisant le fait que le système est vérifié à une transformation projective près (l'échelle est déjà compter $16 - 1 = 15$). Au total, cela fait 18 ($33 - 15$) degrés de liberté ce qui signifie que le tenseur trifocal doit satisfaire à 8 contraintes ($26 - 18$).

Nous avons introduit la notion de tenseur, le lecteur trouvera dans le chapitre 15 de [Hartley 00] et dans le chapitre 8 de [Faugeras 01] des informations sur le calcul pratique du tenseur trifocal ainsi qu'à la page 362 de [Hartley 00] les différentes formules permettant d'utiliser ce tenseur avec toutes les combinaisons possibles de droites et de points. Il apparaît que l'estimation de ce tenseur pour un triplet de vues fait appel à des méthodes et des des problèmes proches de ceux rencontrés lors de l'estimation de la matrice fondamentale (stabilité numérique, intégration des contraintes...). Au moins six points vus dans les trois images sont nécessaires à l'évaluation du tenseur. Ce faible nombre de points permet une évaluation robuste du tenseur en utilisant un algorithme de type RANSAC ([Fischler 81]).

1.3.3 Bilan sur la matrice fondamentale et le tenseur trifocal

Nous venons de dresser un rapide aperçu des techniques permettant d'estimer la matrice fondamentale et le tenseur trifocal. Dans le cas de la reconstruction d'une scène avec plusieurs images, l'estimation de ces matrices implique de créer des groupes de deux ou trois images tout en conservant une structure globale cohérente. La connaissance

de ces matrices traduit complètement la géométrie des caméras et permet d'estimer les matrices de projection associées aux capteurs et, par conséquent, des points 3D à une transformation projective près.

Les avantages de ces approches sont, tout d'abord, la rapidité d'estimation de ces matrices du fait de leur estimation linéaire ou du faible nombre de degrés de liberté dans le cas d'une estimation non-linéaire. Le second avantage est l'existence d'algorithmes à faible coût (algorithme à sept ou huit points pour la matrice fondamentale et six points pour le tenseur trifocal) permettant d'utiliser des approches robustes du type RANSAC afin d'éliminer d'éventuelles faux-appariements. Malgré tout, l'estimation de ces structures est souvent instable et sensible au bruit, leur utilisation n'est pas évidente avec des primitives plus complexes (modèle CAO...) et l'intégration de nouvelles données dans le modèle des capteurs (distorsions...) brise leur linéarité. Enfin, les opérations supplémentaires nécessaires pour remonter à des informations 3D exploitables (positions, orientations des caméras, modèle 3D ...) sont parfois délicates et limitent de ce fait le champ d'action de ces méthodes.

1.4 Méthode de factorisation

Dans le cas où n points sont visibles dans m images, il est possible d'effectuer une reconstruction 3D de la scène par factorisation. Tomasi et Kanade [Tomasi 92] ont proposé une méthode de factorisation basée sur le modèle orthographique de caméra. Cette méthode permet une reconstruction de la scène et des caméras à une transformation affine 3D près. Suite à ces travaux, Weinshall et Tomasi [Weinshall 93] ont proposé une extension au cas orthographique, Poelman et Kanade [Poelman 94] ont traité le cas para-perspectif puis Christy et Horaud [Christy 96], Sturm et Triggs [Sturm 96], Heyden [Heyden 97], Ueshiba et Tomita [Ueshiba 98] ont proposé des solutions pour le cas perspectif (les différents modèles de caméras sont, par exemple, décrits dans [DeMenthon 92] et []).

Afin d'illustrer cette méthode, nous présentons la méthode de factorisation lorsque la caméra est modélisée comme une caméra affine. Considérons n points 3D vus dans m images, notre objectif est de calculer la matrice de projection associée à chaque image et la position de chacun des points 3D. Nous noterons P_k la position 3D du k -ième point, $p_{k,i}$ la position du point k dans l'image i , I_i la matrice intrinsèque de la caméra i et E_i sa matrice d'orientation. Id est une matrice identité. Une des spécificités de cette méthode est la liberté complète laissée vis-à-vis du choix du repère. Nous pouvons choisir ce repère judicieusement en prenant par exemple $\tilde{P}_1 = (0_{[3]}, 1)^T$. Les équations de projection associées à une caméra affine sont données par la relation suivante (le point P_1 sert de référence) :

$$p_{k,i} - p_{1,i} = \frac{1}{\lambda_1} \underbrace{[Id] - p_{1,i}}_{\Gamma_{A_i}^T} I_i E_i^T (P_k - P_1) \quad (1.18)$$

Pour chaque point dans chaque image, on obtient la relation $\Gamma_{Ai}^T P_k = p_{k,i} - p_{1,i}$. En écrivant la relation linéaire ainsi obtenue pour l'ensemble des images i et des points k disponibles, le système devient :

$$\begin{pmatrix} \Gamma_{A1}^T \\ \vdots \\ \Gamma_{Ai}^T \\ \vdots \\ \Gamma_{Am}^T \end{pmatrix} (P_1 \quad - \quad P_i \quad - \quad P_n) = \begin{pmatrix} p_{1,1} - p_{1,1} & - & p_{i,1} - p_{1,1} & - & p_{n,1} - p_{1,1} \\ | & & | & & | \\ p_{1,j} - p_{1,j} & - & p_{i,j} - p_{1,j} & - & p_{n,j} - p_{1,j} \\ | & & | & & | \\ p_{1,m} - p_{1,m} & - & p_{i,m} - p_{1,m} & - & p_{n,m} - p_{1,m} \end{pmatrix} \quad (1.19)$$

La matrice de droite est appelée matrice des mesures car elle ne dépend que des positions observées dans les images. Si l'on considère le produit de matrice à gauche, la matrice composée d'éléments X est de rang 3, car elle est composée de points exprimés dans \mathbb{R}^3 . La matrice composée d'éléments P_{Ai}^T représente l'attitude des caméras; elle est également de rang 3. La matrice de droite admet donc au plus trois valeurs singulières non nulles. Une décomposition de type SVD permet d'aboutir à l'équation suivante (\dots bloc de zéros) :

$$\begin{pmatrix} p_{1,1} - p_{1,1} & - & p_{i,1} - p_{1,1} & - & p_{n,1} - p_{1,1} \\ | & & | & & | \\ p_{1,j} - p_{1,j} & - & p_{i,j} - p_{1,j} & - & p_{n,j} - p_{1,j} \\ | & & | & & | \\ p_{1,m} - p_{1,m} & - & p_{i,m} - p_{1,m} & - & p_{n,m} - p_{1,m} \end{pmatrix} = S \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & \dots \\ 0 & \sigma_2 & 0 & 0 & \dots \\ 0 & 0 & \sigma_3 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} D \quad (1.20)$$

Compte tenu de la présence des zéros dans les matrices et en ne prenant en compte que la i^{eme} ligne de D et la j^{eme} colonne de S , on obtient :

$$\begin{pmatrix} \Gamma_{A1}^T \\ | \\ \Gamma_{Ai}^T \\ | \\ \Gamma_{Am}^T \end{pmatrix} (P_i) = S_{[3*j]}^{[\cdot,j]} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} D_{[3*n]}^{[i]} \quad (1.21)$$

À partir de là, par identification, on trouve directement une solution avec :

$$(P_i) = (D_{[3*n]}^{[i]}) \quad (1.22)$$

Les colonnes de la matrice $D_{[3*n]}^{[i]}$ correspondent à la position des points 3D dans l'espace à une transformation affine près. Cette méthode permet d'estimer la position des points 3D ainsi que les matrices de projection affines associées aux caméras. Le passage de cette reconstruction à une reconstruction métrique implique l'apport de nouvelles informations sur la scène ou sur les caméras (matrice intrinsèque). Un des risques majeurs d'échec de cette méthode est le choix d'un point P_1 mal apparié, une mauvaise qualité

sur cet appariement se répercutera également sur la qualité du résultat. D'un point de vue pratique, la décomposition par SVD conduit à plus de trois valeurs singulières non nulles (du fait de la présence de bruit), les trois valeurs les plus élevées sont généralement bien marquées et il est facile de considérer comme nulles les valeurs trop faibles. Enfin, afin d'améliorer le conditionnement du problème, il est préférable (cf. [Sturm 96]) de normaliser les données en appliquant à chaque image une transformation permettant de centrer le barycentre du nuage de points 2D et de borner les coordonnées entre $-\sqrt{2}$ et $+\sqrt{2}$ comme dans le cas de l'estimation des matrices fondamentales [Hartley 97].

Le cas projectif est plus complexe du fait de l'apparition de non-linéarité dans les équations qui impliquent l'utilisation de méthodes itératives (par exemple : cf. [Sturm 96]).

Pour conclure, l'approche par factorisation présente l'avantage (notamment dans le cas affine) d'offrir une solution très efficace au problème de reconstruction, toutefois elle repose sur l'appariement de points homologues dans toutes les images. Dans le cadre des méthodes évoquées ici, l'absence d'un point homologue rend caduque la méthode, toutefois, des travaux comme ceux de Aanaes et al. dans [Aanaes 02] propose des méthodes permettant d'utiliser les approches par factorisation avec des données caractérisées par une incertitude ou même des localisations manquantes.

1.5 Méthodes d'intersection-résection

Les méthodes d'intersection-résection sont des algorithmes itératifs alternant les estimations de sous parties de l'ensemble des paramètres inconnus (cf. [Krauss 93]). Ces algorithmes alternent itérativement les estimations des sous ensembles de paramètres (lors du calcul des paramètres d'un sous ensemble tous les autres paramètres sont fixés). Aucun calcul de dérivées n'est effectué, cela conduit à des itérations peu coûteuses en nombre d'opérations mais, malheureusement, cette technique présente en général une convergence lente associée à un très grand nombre d'itérations et à un domaine de convergence réduit.

Dans le cas de notre problème de reconstruction d'un nuage de points 3D, une division naturelle consiste à estimer d'une part les paramètres associés aux caméras et de l'autre les paramètres associés au nuage de points. Cette approche rudimentaire supporte difficilement l'estimation de paramètres intrinsèques des caméras (même lorsque ceux-ci sont communs à toutes les images). La principale cause permettant d'expliquer ces piètres qualités est l'absence de prise en compte des inévitables couplages existant entre les paramètres des caméras et les paramètres de la scène 3D.

Des méthodes plus subtiles existent, elles tentent d'affiner la méthode précédente en intégrant des informations sur les dérivées des fonctions servant à évaluer les paramètres des caméras et les paramètres du modèle (des détails sont disponibles dans [Triggs 00] pages 27-28). Néanmoins, quelques soient les méthodes, l'efficacité n'est pas systématique et, même si elles constituent parfois une réponse adaptée à certains problèmes, elles ne sont pas une réponse universelle aux problèmes d'ajustement de faisceaux. Dans le paragraphe suivant, nous allons présenter les techniques d'ajustement de faisceaux basées sur des méthodes de minimisation faisant appel à des dérivées et présentant, en général, de meilleures performances que les techniques d'intersection-résection.

1.6 Ajustement de faisceaux par minimisation

Dans ce paragraphe, nous allons présenter les techniques d'ajustement de faisceaux. Cette méthode de reconstruction est très fréquemment utilisée en vision par ordinateur car elle permet d'améliorer la qualité des reconstructions obtenues par les méthodes présentées précédemment (pour une synthèse récente se référer à [Triggs 00]).

Nous pouvons définir l'ajustement de faisceaux comme une méthode itérative d'estimation d'une scène 3D à partir d'images :

- minimisant une fonction mesurant l'adéquation entre un modèle 3D et les éléments observés dans les images,
- définissant l'état futur en fonction de la position présente et d'une prédiction (exprimée sous la forme du calcul des dérivées de la fonction mesure par rapport aux paramètres inconnus) dépendant de la modélisation du modèle 3D et des capteurs.

L'ajustement de faisceaux est basé sur une minimisation non-linéaire permettant de gérer des données éparses, de prendre en compte des primitives de différents types, et d'intégrer des contraintes. Toutefois, cette technique requiert une initialisation pertinente des paramètres à estimer et peut mener à des minima locaux voire même à des non convergences ! Afin d'introduire cette méthode de reconstruction avec un maximum de clarté, nous allons présenter dans un même mouvement la méthode d'ajustement de faisceaux en tant que problème de vision par ordinateur et les méthodes d'optimisation employées afin de parvenir à une solution.

Le succès d'une méthode d'ajustement de faisceaux est, en général, le fruit d'une prise en compte raisonnée de la réalité physique du problème posé (choix des modèles, choix judicieux des prises de vue...) et des propriétés mathématiques de l'ensemble des éléments inhérents à la méthode d'optimisation choisie. Le principe de l'ajustement de faisceaux tel qu'il est présenté dans le paragraphe suivant peut sembler simple de prime abord, mais dans la pratique de la reconstruction de scènes tridimensionnelles à partir d'images, cette technique prend parfois une physionomie dantesque ! En effet, les inter-corrélations entre les paramètres associés au modèle 3D et ceux associés aux caméras, la présence de bruit dans la détection des positions des primitives dans les images, les faux appariements, le mélange de primitives 3D de natures différentes, l'intégration de contraintes afin de fixer le repère ou de tirer partie de connaissances disponibles sur la scène et le besoin de fournir une valeur initiale aux paramètres multiplient les difficultés de mise en oeuvre de cette technique. De plus, le nombre élevé de paramètres à estimer rend difficile la visualisation des trajectoires suivies au cours de l'optimisation, rendant de ce fait délicate une analyse des causes de succès ou d'échec.

Notre exposé va débiter par une présentation du principe de l'ajustement de faisceaux. Les principes de base exposés montrent la relation étroite existant entre les caméras, la position des primitives dans l'espace et les images.

Par la suite, nous formalisons ce problème en le positionnant dans le cadre des méthodes de minimisation. Nous détaillons notamment la méthode dite de Levenberg-Marquardt que nous allons utiliser tout au long de cette thèse.

Dans un troisième temps, nous traitons des spécificités liées aux minimisations de type ajustement de faisceaux. Nous mettons en évidence l'aspect creux de certaines matrices et les gains de performances que cela peut engendrer.

Après cette présentation de la méthode, nous nous attardons sur un point délicat qui prendra une importance particulière dans le chapitre suivant, comment choisir (ou ne pas choisir !) le repère de reconstruction et quelle influence ce choix a-t-il sur le résultat final ?

Enfin, pour finir cette présentation, des méthodes permettant de traiter des séquences d'images et une méthode dissimulant l'estimation des paramètres intrinsèques sont présentés. Cette dernière approche ressemble dans l'idée à l'algorithme proposé dans le prochain chapitre.

1.6.1 Principe de base

Le principe de base de l'ajustement de faisceaux est d'estimer les paramètres des caméras et des points 3D en minimisant la distance entre les positions des points détectés dans les images et la projection des points 3D dans ces images.

Nous définissons une fonction f exprimant la projection de chacun des points 3D dans chacune des images en fonction d'un vecteur de paramètres \vec{x} caractérisant la scène. La position des points identifiés dans les images est stockée dans un vecteur \vec{y} . La solution au problème d'ajustement est obtenue lorsque $f(\vec{x}) = \vec{y}$. D'un point de vue géométrique, cette équation est au coeur du problème de reconstruction. De nombreux travaux ont permis de mettre en lumière certaines propriétés de ce problème, de définir des solutions possibles et de montrer l'existence de configuration admettant des solutions multiples ou, au contraire, aucune solution. La fonction f représente un modèle de la scène et le choix de f conditionne fortement le type de solution recherchée. Dans notre cas, le modèle retenu pour les caméras est le modèle sténopé classique avec, éventuellement, l'ajout de modèles polynomiaux pour corriger les distorsions. Du point de vue du modèle de l'objet 3D, deux possibilités sont envisagées dans cette thèse, d'une part, nous traitons, dans cette première partie, le cas des nuages de points libres où chaque point est paramétré par ses trois coordonnées, d'autre part, nous abordons par la suite le cas des modèles contraints pour lesquels le jeu de paramètres ne décrit pas directement les coordonnées des points 3D.

1.6.2 Description de la méthode de minimisation

Dans cette partie, nous allons présenter les méthodes de minimisation employées afin d'estimer les paramètres du modèle 3D et des caméras. D'un point de vue pratique, nous avons utilisé la méthode de Levenberg-Marquardt. Nous allons détailler, dans un premier temps, les principes mathématiques de cette technique, puis nous mettrons en évidence les améliorations rendues possibles par la prise en compte de la spécificité du problème d'ajustement de faisceaux. L'algorithme général de l'ajustement de faisceaux est le suivant :

ENTRÉE :

Mesures disponibles (\vec{y}) :

L'ensemble des projections de primitives 3D détectées dans les images.

Paramètres à estimer (\vec{x}) :

Dans le cadre de l'ajustement de faisceaux, ces paramètres à estimer peuvent être les orientations des caméras, la position de leurs centres optiques, l'ensemble des paramètres internes aux capteurs (longueurs focales, paramètres de distorsions...) et l'ensemble des paramètres permettant de décrire le modèle 3D observé. Une valeur initiale de ces paramètres doit être fournie (elle conditionne le résultat de la convergence!).

La fonction de calcul des projections 2D (f) :

C'est la fonction exprimant la projection des primitives 3D dans les images en fonction des paramètres recherchés. Dans le cas de l'utilisation d'une méthode de minimisation non-linéaire, il est nécessaire que cette fonction soit continue et dérivable lorsque la méthode choisie implique des calculs de dérivées.

ESTIMATION :

1- Recherche d'une meilleure solution pour \vec{x} :

Recherche d'une modification applicable aux vecteurs de paramètres actuels permettant de réduire la distance entre les primitives 2D détectées dans les images et la projection des primitives 3D dans ces images.

2- Mise à jour :

Mise à jour du vecteur de paramètres solution et prise de décision de poursuivre la recherche d'un minimum (refaire 1) ou fin du processus (allez à SORTIE) (nombre d'itération Max atteint, distance entre les mesures et le modèle suffisamment faible...).

SORTIE :

Une solution pour le vecteur de paramètres \vec{x} . Cette solution correspond à la position des primitives 3D et à l'ensemble des paramètres caractérisant les capteurs (positions et propriétés internes).

1.6.2.1 Formalisation du problème de minimisation

Afin de traduire l'ajustement de faisceaux comme un problème de minimisation, nous cherchons désormais l'ensemble de paramètres \vec{x} minimisant la fonction $\phi(\vec{x}) = f(\vec{x}) - \vec{y} = 0$ (distance entre les points 3D reprojétés et les points 2D détectés dans les images). La fonction ϕ est continue et dérivable (car les modèles employés le sont) et, par conséquent, le développement de Taylor de la fonction ϕ au voisinage de \vec{x}_0 est :

$$\phi(\vec{x}) = \phi(\vec{x}_0) + \sum_i \frac{\partial \phi}{\partial x_i}(\vec{x}_0)(x_i - x_{0,i}) + \frac{1}{2} \sum_{i,j} \frac{\partial^2 \phi}{\partial x_i \partial x_j}(\vec{x}_0)(x_i - x_{0,i})(x_j - x_{0,j}) + \epsilon \quad (1.23)$$

Cette équation peut s'écrire également ainsi :

$$\phi(\vec{x}) = \phi(\vec{x}_0) + \vec{\nabla} \phi(\vec{x}_0) \cdot \Delta \vec{x} + \frac{1}{2} \vec{x} H \vec{x} + \epsilon \quad (1.24)$$

avec H la matrice hessienne telle que $H_{ij} = \frac{\partial^2 \phi}{\partial x_i \partial x_j}$.

Dès lors, deux possibilités se distinguent selon que $\epsilon = 0$ ou que $\epsilon \neq 0$.

- **Si $\epsilon = 0$** : L'égalité $\phi(\vec{x}) = \phi(\vec{x}_0) + \vec{\nabla} \phi(\vec{x}_0) \cdot \vec{x} + \frac{1}{2} \vec{x} H \vec{x}$ est parfaitement vérifiée et, en dérivant par rapport à \vec{x} :

$$\vec{\nabla} \phi(\vec{x}) = H \vec{x} + \vec{\nabla} \phi(\vec{x}_0) = \vec{0} \quad (1.25)$$

On obtient immédiatement, ϕ est minimale lorsque $\Delta \vec{x} = -H^{-1} \vec{\nabla} \phi(\vec{x}_0)$.

- **Si $\epsilon \neq 0$** : Deux cas de figure sont possibles, soit la forme quadratique obtenue au moyen du développement de Taylor est une bonne approximation de la fonction ϕ , soit elle ne l'est pas. Dans le premier cas, on déduit de 1.25 que la solution \vec{x}_1 est fournie par :

$$\vec{x}_1 \simeq \vec{x}_0 - H^{-1} \vec{\nabla} \phi(\vec{x}_0) \simeq \vec{0} \quad (1.26)$$

Lorsqu'au contraire l'approximation s'avère grossière, la seule possibilité d'estimation est de se déplacer dans le sens opposé au gradient. Le déplacement se fait dans la direction du gradient avec un pas de descente arbitraire... Le lien entre deux itérations devient dès lors (A est le coefficient arbitraire fixant le pas) :

$$\vec{x}_1 \simeq \vec{x}_0 - A \vec{\nabla} \phi(\vec{x}_0) \quad (1.27)$$

La méthode 1.26 correspond en fait à la méthode de Newton-Raphson et 1.27 à une méthode de descente du gradient (steepest descent method). Le cas le plus favorable où $\epsilon = 0$ correspond à l'estimation du minimum d'une fonction véritablement quadratique, malheureusement, le critère estimé lors d'un ajustement de faisceaux n'est pas quadratique... Par conséquent, l'ajustement de faisceaux correspond au cas $\epsilon \neq 0$. Nous avons vu que dans ce cas les deux méthodes 1.26 et 1.27 peuvent être mises en jeu (en fonction de l'importance de ϵ). Dans la pratique, il semble délicat de choisir l'une ou l'autre des solutions. Nous allons voir dans le paragraphe suivant que la méthode de Levenberg-Marquard va apporter une pondération adaptative aux influences respectives de l'approche 1.26 et de l'approche 1.27.

⁰Méthode décrite dans [Press 93] pages 688 et suivantes.

1.6.2.2 La méthode de Levenberg-Marquard⁰

Sous l'hypothèse que la détection de projections 2D ait été perturbée par un bruit gaussien et que les incertitudes associées à la i^{eme} détection est σ_i , le critère minimisé lors d'un ajustement de faisceaux est la valeur du χ^2 (N : nombre de mesures disponibles) :

$$\chi^2(\vec{x}) = \sum_{i=1}^N \left[\frac{y_i - f_i(\vec{x})}{\sigma_i} \right]^2 \quad (1.28)$$

En vue de minimiser χ^2 en utilisant les deux méthodes précédemment évoquées, nous pouvons d'ores et déjà calculer le gradient et le hessien associés à cette fonction :

$$\frac{\partial \chi^2}{\partial \vec{x}_k} = -2 \sum_{i=1}^N \frac{[y_i - f_i(\vec{x})]}{\sigma_i^2} \frac{\partial f_i(\vec{x})}{\partial \vec{x}_k} \quad (1.29)$$

$$\frac{\partial^2 \chi^2}{\partial \vec{x}_k \partial \vec{x}_l} = 2 \sum_{i=1}^N \frac{1}{\sigma_i^2} \left[\frac{\partial f_i(\vec{x})}{\partial \vec{x}_k} \frac{\partial f_i(\vec{x})}{\partial \vec{x}_l} - [y_i - f_i(\vec{x})] \frac{\partial^2 f_i(\vec{x})}{\partial \vec{x}_k \partial \vec{x}_l} \right] \quad (1.30)$$

Afin de simplifier les notations, nous notons désormais : $\vec{\beta} = -\frac{1}{2} \vec{\nabla} \chi^2$ et la matrice de courbure $[\alpha] = \frac{1}{2} H_{\chi^2}$.

Notre objectif est de déterminer itérativement l'incrément $\partial \vec{x}$ à appliquer à la valeur initiale des paramètres \vec{x}_0 afin d'atteindre le jeu de paramètres \vec{x}_{min} minimisant la fonction χ^2 . Nous aurons $\partial \vec{x} = \vec{x}_{min} - \vec{x}_0$. L'équation 1.26 applicable lorsque χ^2 est proche d'une forme quadratique devient :

$$[\alpha] \partial \vec{x} = \vec{\beta} \quad (1.31)$$

et l'équation 1.27 applicable lorsque χ^2 est loin d'une forme quadratique devient (avec C_{ste} une constante arbitraire) :

$$\partial \vec{x} = C_{ste} \vec{\beta} \quad (1.32)$$

Si l'on souhaite effectuer un ajustement de faisceaux, il est nécessaire d'initialiser les valeurs des paramètres de \vec{x}_0 . Une initialisation peut être fournie soit par une connaissance a priori de la scène à reconstruire (environnement maîtrisé, contrôle industriel...), soit par l'exécution préalable d'un processus d'initialisation en utilisant, par exemple, les techniques précédemment détaillées dans ce document (matrice fondamentale, factorisation...). Les méthodes d'optimisation non-linéaires sont très fortement dépendantes des conditions initiales, leur convergence n'est pas assurée et lorsque la convergence est obtenue, il se peut que cela soit dans un minimum local. De plus, la méthode de Newton fonctionne bien lorsque la solution est proche mais, lorsque l'initialisation est de mauvaise qualité, l'hypothèse quadratique est tellement approximative que la convergence s'avère délicate voire même impossible. D'un autre côté, la méthode de gradient peut être particulièrement lente (exemple : si l'on imagine que l'on recherche un minimum à l'extrémité d'une vallée, les différentes itérations peuvent se traduire par une oscillation

entre chacune des deux pentes de la vallée!). La technique de Levenberg-Marquard permet de réduire la difficulté du choix de la méthode en combinant la méthode de la plus forte pente et la méthode d'inversion du hessien au fur et à mesure de la convergence.

Les deux idées permettant de marier continuellement ces deux techniques sont les suivantes :

- **Choix de la constante (C_{ste})** : La première difficulté est de fixer la constante arbitraire définie dans l'approche gradient. L'idée est de remplacer ce scalaire C_{ste} par un vecteur dont chacune des composantes est choisie judicieusement. Le choix de ces composantes constitue une mise à l'échelle du pas. Chaque composante est le produit d'une fonction du rayon de courbure qui lui est associé et d'un coefficient λ commun à toute les composantes permettant de faire varier le pas du gradient. L'équation 1.32 devient :

$$\partial x_i = \frac{1}{\lambda \alpha_{ii}} \beta_i \quad (1.33)$$

- **Remplacement de $[\alpha]$** : L'idée principale est de poser $[\alpha]' = [\alpha] + \lambda I$ avec I la matrice identité. Cette idée permet de remplacer les deux équations 1.31 et 1.33 par une seule :

$$[\alpha]' \partial \vec{x} = ([\alpha] + \lambda I) \partial \vec{x} = \vec{\beta} \Leftrightarrow [\alpha]' \partial \vec{x} = \left(\frac{1}{2} H_{\chi^2} + \lambda I \right) \partial \vec{x} = -\frac{1}{2} \vec{\nabla}_{\chi^2} \quad (1.34)$$

Lorsque λ tend vers 0, l'équation 1.34 tend vers 1.31 et lorsque λ est grand 1.34 tend vers 1.33.

La grande force de cette méthode est de permettre une prise en compte continue des deux méthodes d'optimisation. Le seul choix laissé à l'utilisateur est la définition de la valeur initiale de λ et de sa règle d'évolution (on peut prendre par exemple 10^{-3} comme valeur initiale, multiplier λ par 10 à chaque échec et diviser λ par 10 en cas de succès, cf. [Press 93]).

1.6.3 Mise en oeuvre et exploitation des particularités de l'ajustement de faisceaux

Nous venons de présenter la méthode de minimisation d'un point de vue général. Le coeur des performances de l'ajustement de faisceaux est très intimement lié à l'équation 1.34. En effet, à chaque itération, l'objectif est de déterminer le pas $\partial \vec{x}$ en résolvant le système linéaire $\left(\frac{1}{2} H_{\chi^2} + \lambda I \right) \partial \vec{x} = -\frac{1}{2} \vec{\nabla}_{\chi^2}$. Avant d'engager la résolution, il est nécessaire de connaître la matrice jacobienne ($\vec{\nabla}_{\chi^2}$) et la matrice hessienne (H_{χ^2}). Cela implique de calculer les dérivées premières et secondes de χ^2 et de stocker ces résultats en mémoire. Dans le cas de la reconstruction de nuage de points libres, une expression analytique peut être fournie pour ces dérivées, mais lorsqu'une telle expression n'est pas disponible, il faut recourir à un calcul numérique des dérivées qui implique de nombreuses évaluations de la fonction χ^2 . Ces évaluations sont coûteuses et les auteurs de [Press 93]

rapportent que le calcul des dérivées secondes déstabilisent la méthode et est particulièrement sensible au bruit dans les mesures. Les auteurs (et d'autres implémentations de cet algorithme font de même) remplacent les éléments de la matrice $[\alpha]$ par les éléments suivant moins coûteux à évaluer (on supprime les termes du second ordre dans 1.30) :

$$\alpha_{kl} = \sum_{i=1}^N \frac{1}{\sigma_i} \left[\frac{\partial \chi_i^2(\vec{x})}{\partial x_k} \frac{\partial \chi_i^2(\vec{x})}{\partial x_l} \right] \quad (1.35)$$

Cette substitution améliore la stabilité de l'algorithme et permet de n'estimer que des dérivées premières en limitant de ce fait le nombre d'appel au critère (le nombre d'appel dépend de l'utilisation de dérivées numériques ou analytiques).

Il reste à résoudre à chaque itération $([\alpha] + \lambda Id) \partial \vec{x} = -\frac{1}{2} \vec{\nabla} \chi^2$ qui est un système de la forme ¹ :

$$A \partial \vec{x} = B \quad (1.36)$$

Jusqu'ici nous n'avons pas exploité de spécificités particulières liées à l'ajustement de faisceaux appliqué à un nuage de points libres. C'est lors de la résolution de 1.36 que l'exploitation de ces spécificités permet de réduire les temps de calcul et les ressources mémoires nécessaires. Si l'on considère un nuage de point 3D libres, la structure de la matrice jacobienne est très creuse et présente une structure par blocs très marquée. Par conséquent, la matrice A présente également une structure par blocs très particulière constituée de trois ensembles (cf. 1.5) :

- U_i =matrice représentant la dépendance entre les mesures dans une image et les paramètres de la caméra associée.
- V_j = matrice traduisant les relations entre les paramètres du j^{eme} point libre et les mesures qui lui sont associées.
- W =matrice traduisant les inter-corrélations entre les paramètres du modèle 3D et ceux des caméras, c'est une matrice rectangulaire non creuse.

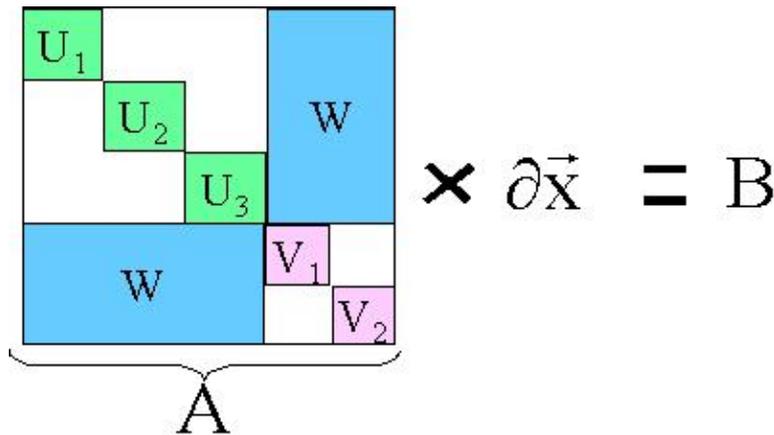


FIG. 1.5: Structure creuse de la matrice A (les valeurs des zones non grisées sont nulles).

¹si les $\sigma_i = 1$, on a $A = J^T J + \lambda Id$ et $B = -J^T [\vec{y} - f(\vec{x})]$ avec J la matrice jacobienne associées à χ .

Le système d'équation à résoudre s'écrit donc par blocs :

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \partial \vec{x}_{cameras} \\ \partial \vec{x}_{modele3D} \end{pmatrix} = \begin{pmatrix} B_{cameras} \\ B_{modele3D} \end{pmatrix} \quad (1.37)$$

Une solution pour résoudre efficacement ce système est de multiplier les deux membres de l'égalité par la matrice :

$$\begin{pmatrix} I & -WV^{-1} \\ 0 & I \end{pmatrix} \quad (1.38)$$

Le résultat est l'équation suivante :

$$\begin{pmatrix} U - WV^{-1}W^T & 0 \\ W^T & V \end{pmatrix} \begin{pmatrix} \partial \vec{x}_{cameras} \\ \partial \vec{x}_{modele3D} \end{pmatrix} = \begin{pmatrix} B_{cameras} - WV^{-1}B_{modele3D} \\ B_{modele3D} \end{pmatrix} \quad (1.39)$$

La nouvelle équation ainsi obtenue peut être résolue en deux temps. Dans un premier temps, il est possible d'estimer l'incrément à appliquer aux paramètres des caméras en résolvant l'équation :

$$(U - WV^{-1}W^T) \partial \vec{x}_{cameras} = B_{cameras} - WV^{-1}B_{modele3D} \quad (1.40)$$

Cette équation ne dépend que de $\partial \vec{x}_{cameras}$. Le calcul de l'ensemble des matrices est simple et la structure par bloc de V permet d'évaluer très efficacement son inverse. Le point le plus coûteux en terme de calcul est l'inversion de la matrice $U - WV^{-1}W^T$. Cette inversion peut bénéficier parfois de la structure particulière de cette matrice (symétrique, bande).

Un fois que $\partial \vec{x}_{cameras}$ est connu, il suffit de réinjecter sa valeur dans la seconde équation pour définir $\partial \vec{x}_{modele3D}$ avec :

$$\partial \vec{x}_{modele3D} = V^{-1}(B_{modele3D} - W^T \partial \vec{x}_{cameras}) \quad (1.41)$$

Dans le cas que nous venons de présenter, nous avons transformé notre système en multipliant l'équation 1.37 par la matrice 1.38 afin d'isoler l'évaluation de la variation à affecter aux paramètres des caméras. Ce choix est pertinent lorsque le nombre de paramètres associés aux caméras est inférieur au nombre de paramètres associés au modèle 3D. Dans le cas contraire, il peut être préférable d'isoler l'estimation des paramètres du modèle 3D par un procédé analogue.

Nous venons de présenter la méthode d'ajustement de faisceaux basée sur la méthode de Levenberg-Marquard et tirant partie de la structure du problème posé. Cette version n'est pas la seule et les algorithmes de type ajustement de faisceaux sont extrêmement nombreux et offrent souvent une réponse optimisée à un problème spécifique. Les méthodes d'ajustement de faisceaux sont intimement liées aux techniques de minimisation de fonction et, à l'image de ces méthodes, sont parfois délicates à employer du fait de problèmes de convergence et de l'absence de maîtrise des trajectoires parcourues dans un espace de paramètres pouvant atteindre de très grandes dimensions.

Afin d'illustrer les difficultés pouvant survenir lors de l'utilisation d'une méthode d'optimisation, nous faisons ici une petite digression. Nous prenons un exemple classique consistant à résoudre l'équation $x^n = 1$ dans l'espace complexe au moyen d'une méthode de minimisation (type newton ou Levenberg-Marquard). Cette équation admet n racines complexes. Si l'on fournit une valeur initiale x_0 et que l'on utilise une méthode de type Newton pour minimiser $|x^n - 1| = 0$, la convergence est obtenue à l'une des n racines. Afin de rendre apparente la ligne de partage des eaux, nous affectons au point (valeur initiale) de départ une couleur dépendant de la racine atteinte. L'image obtenue pour $n = 3$ est représentée à la figure 1.6 :

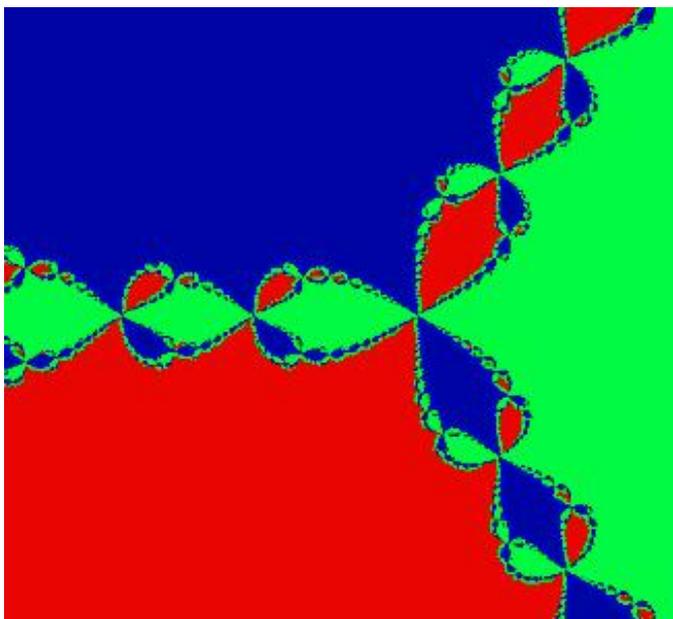


FIG. 1.6: Représentation de l'ensemble de Newton pour $n=3$.

Cette image montre la complexité du comportement de la méthode de Newton dans un cas simple. On comprend dès lors les difficultés pratiques auxquelles nous confrontent les techniques d'ajustement de faisceaux qui minimisent des fonctions plus complexes dans des espaces de dimension très nettement supérieure. Dans le chapitre suivant, nous présenterons la technique d'ajustement de faisceaux que nous avons mise au point. Nous verrons par le biais de diverses expériences les comportements particuliers et la difficulté à maîtriser intégralement les phénomènes mis en jeu. L'exemple de l'ensemble de Newton n'est pas ici une réponse au problème à venir, mais une illustration de la complexité des tâches de minimisation auxquelles nous sommes confrontés.

1.6.4 La question du choix de jauge

Jusqu'à présent nous avons présenté la méthode d'ajustement de faisceaux sans nous préoccuper particulièrement du repère dans lequel est effectué la reconstruction. Il est possible de choisir comme repère de reconstruction un repère associé à l'une des caméras,

un repère associé au modèle 3D, une autre référence... Existe-t-il une influence du choix du repère sur la reconstruction ? La précision de la reconstruction dépend-elle du choix du repère ? La solution en fin de convergence dépend-elle de ce choix ? Si oui, comment fixer le repère ?

Dans les paragraphes suivants, nous allons apporter quelques réponses à ces questions en nous basant sur les quelques travaux ayant traité ce problème. Dans un premier temps, nous aborderons l'influence du choix du repère sur la précision du modèle 3D final. Dans un second temps, nous aborderons le difficile problème de l'influence possible du choix du repère sur le comportement du processus d'estimation.

1.6.4.1 Choix du repère et mesure de longueurs

Parmi les questions précédentes, les deux premières, concernant la qualité de la reconstruction, ont été étudiées par Morris et al. dans différents articles [Morris 00a] [Morris 01a] [Morris 01b] [Kanatani 00b] [Kanatani 00a]. Ces travaux mettent en évidence la relation existant entre l'incertitude estimée sur les éléments du modèle 3D et le choix du repère. Le problème le plus patent est celui du choix du facteur d'échelle. En effet, les reconstructions sont généralement obtenues à un facteur d'échelle près et il est nécessaire d'apporter une information de longueur afin de dimensionner le modèle 3D comme dans la réalité. Par exemple, l'apport de cette métrique peut se faire en fournissant la distance entre deux points de la scène. Une telle stratégie fait reposer le choix du facteur d'échelle sur les deux points extrémités du segment choisi. Si ces deux points sont, par malheur, les deux points les plus mal estimés du modèle, les conséquences peuvent être désastreuses pour la qualité globale de la reconstruction. Dans [Morris 01a] et [Morris 01b] ce problème spécifique est étudié, diverses expériences mettent en évidence l'importance de l'orientation et de la longueur du segment choisi pour apporter la métrique. Sans surprise, il apparaît que les segments orthogonaux aux axes de visée et de grandes tailles sont, en général, les plus favorables. Les auteurs fournissent une méthode d'analyse de l'incertitude permettant de fixer le repère en minimisant l'incertitude sur la mesure de distance.

Vis-à-vis de l'ajustement de faisceaux, l'étude de l'influence du choix de repère vis-à-vis de l'incertitude sur les mesures permet de définir le plus habilement possible la stratégie de prise de vue et la (les) mesure(s) à effectuer sur le terrain afin d'apporter la métrique lors de la reconstruction. Cette étude fournit également un critère permettant de calculer en tenant compte des diverses incertitudes la longueur la plus probable d'un segment observé dans l'image. La partie suivante évoque la question du choix du repère non plus lors de l'interprétation des résultats mais lors du processus d'optimisation.

1.6.4.2 Choix du repère et processus d'estimation

Si l'on se place dans une configuration donnée, il existe une infinité de combinaisons de paramètres présentant les mêmes erreurs de reprojection. L'ensemble des paramètres vérifiant cette propriété traduit une même réalité physique exprimée dans des repères différents. Un exemple simple de changement de repère est tout simplement un changement d'échelle. Appliquer un même coefficient multiplicateur ($\neq 0$) à la position des

points 3D et des centres optiques des caméras ne change rien aux distances mesurées dans les images. L'ensemble des repères vérifiant cette propriété forme une structure appelée orbite de jauge. Tout le problème du choix du repère est de trouver la jauge appartenant à cette orbite et offrant les meilleures propriétés possibles vis à vis de la convergence et de la qualité d'estimation des paramètres. D'ores et déjà [Morris 00a] [Morris 01a] [Morris 01b] [Kanatani 00b] [Kanatani 00a] (cf. sous-partie précédente) ont montré que le choix d'un repère n'est pas anodin vis-à-vis de l'estimation des incertitudes associées aux éléments reconstruits et que des critères visant à minimiser l'erreur sur les mesures souhaitées peuvent être définis afin de sélectionner le repère le plus pertinent parmi tous les repères possibles.

Dans le cas des méthodes d'optimisation de type Newton, le choix du repère revient à imposer des contraintes sur le système. Ne pas imposer de telles contraintes implique de laisser au système des degrés de liberté supplémentaires qui ne peuvent pas être estimés. Diverses stratégies existent :

- **Repère fixé sur quelques points particuliers** : Une première stratégie consiste à sélectionner des points dont on connaît précisément les positions 3D. Par exemple, un point peut être pris comme origine du repère (3 ddl²), un second point peut contraindre l'axe des x en étant modélisé par un seul paramètre (X,0,0) (2 ddl) et un troisième peut être positionné dans le plan Oxy en le modélisant par (X,Y,0) (1 dll). Au total, six degrés de liberté sont ainsi fixés et définissent la position et l'orientation du repère de construction par rapport au modèle. Il est également possible de fixer le facteur d'échelle en fixant la coordonnée en X du second point. De tels choix sont arbitraires et n'offrent pas de garantie vis-à-vis de la reconstruction 3D du fait du rôle prépondérant de certains points par rapport aux autres. Comme dans le cas précédent du choix du facteur d'échelle, seul l'utilisateur peut assurer, par son expertise et des choix judicieux, que ces trois points ne sont pas les points les plus mal estimés du fait, par exemple, d'une détection de très mauvaise qualité dans les images...
- **Repère fixé sur les caméras** : Une autre méthode consiste à fixer le repère en contraignant les matrices de projection de certaines caméras à avoir des formes particulières. Par exemple, le repère peut être imposé en choisissant ($I_{3 \times 3} | 0$) comme matrice de projection de la première caméra et en imposant une ligne de la matrice de projection associée à la seconde caméra à (0, 0, 0, 1). Ici aussi, le côté arbitraire du choix du repère ne donne aucune assurance sur la qualité du résultat final de la reconstruction 3D.
- **Ne rien contraindre** : Dans ce cas, aucune contrainte n'est introduite dans le système et le repère peut évoluer librement. Cette liberté se traduit par sept degrés de liberté superflus dans le lot de paramètres estimés. L'influence de ce choix sur la trajectoire suivie par le jeu de paramètres au cours de l'estimation est absolument incontrôlée de même que les conséquences éventuelles sur l'évaluation

²ddl : degré de liberté

des dérivées (problèmes des dérivées numériques avec les arrondis machines...).

- **Contraindre le repère à rester dans une certaine zone de l'espace :** Une méthode de ce type est proposée par Krauss dans [Krauss 93], l'idée exposée est de fournir sept équations de contraintes ne générant pas de disparité dans la prise en compte des différents points 3D, mais permettant de limiter le plus possible les dérives du repère. Trois classes de contraintes sont définies :

1. *Contraindre la translation :* afin de laisser inchangé le centre de gravité du nuage de points. La variation en X, Y et Z de sa position doit être nulle au cours des itérations. Cette contrainte peut s'écrire (coordonnées du nouveau point i (X_i, Y_i, Z_i)) :

$$\begin{aligned}\sum_i dX_i &= 0 \\ \sum_i dY_i &= 0 \\ \sum_i dZ_i &= 0\end{aligned}\tag{1.42}$$

2. *Contraindre la rotation :* sur le même principe et suite à une linéarisation de la matrice de rotation (coordonnées de l'ancien point i (X_i^0, Y_i^0, Z_i^0)) :

$$\begin{aligned}\sum_i -Z_i^0 dY_i + Y_i^0 dZ_i &= 0 \\ \sum_i Z_i^0 dX_i - X_i^0 dZ_i &= 0 \\ \sum_i -Y_i^0 dX_i + X_i^0 dY_i &= 0\end{aligned}\tag{1.43}$$

3. *Contraindre le facteur d'échelle :* l'équation de contrainte est ici :

$$\sum_i X_i^0 dX_i + Y_i^0 dY_i + Z_i^0 dZ_i = 0\tag{1.44}$$

Ces sept équations permettent de contraindre les sept degrés de liberté restants. Krauss présente deux méthodes afin d'intégrer ces contraintes à l'ajustement de faisceaux. La première solution proposée est d'ajouter ce système aux équations de projection. Cette approche n'est pas efficace car elle brise la structure creuse des équations normales et n'assure pas une vérification stricte de ces contraintes. La seconde solution utilise les multiplicateurs de Lagrange. Si l'on note $C\vec{x} = \vec{0}$ le système formé par les sept équations de contraintes que l'on vient de présenter et \vec{k} un vecteur dont les composantes sont les sept multiplicateurs de Lagrange, l'équation normale $A\partial\vec{x} = B$ (1.36) devient :

$$\begin{pmatrix} A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \vec{x} \\ \vec{k} \end{pmatrix} = \begin{pmatrix} B \\ \vec{0} \end{pmatrix}\tag{1.45}$$

La structure creuse de A est conservée et les équations de contraintes sont vérifiées. La résolution de ces équations permet d'obtenir une solution à l'ajustement de faisceaux n'ayant pas strictement fixé le repère mais ayant intégré des équations de contraintes équilibrées entre tous les points permettant d'empêcher sa dérive.

- **Normalisation** : D'autres approches visant à contraindre habilement le choix du repère existent. McLauchlan dans [McLauchlan 99a] et [McLauchlan 99b] propose une technique adaptée au cas des reconstructions projectives et euclidiennes basée sur une normalisation des matrices de projection et des vecteurs représentant la position des points 3D dans l'espace. Les résultats obtenus montrent une légère accélération du processus de minimisation. Lorsque l'initialisation est proche de la solution, les gains ne sont pas véritablement significatifs. Le gain le plus important est un gain en stabilité lorsque l'initialisation est de mauvaise qualité.

La question du choix du repère demeure une question relativement ouverte. Les conséquences expérimentales de ce choix restent encore difficiles à prévoir dans les cas réels. Un point remarquable est l'éventuelle dissociation existant entre, d'une part, la description mathématique de l'influence du choix de repère et notamment l'invariance des mesures lors du parcours de l'orbite des jauges et, d'autre part, les conséquences pratiques de tels changements. Prenons l'exemple du calcul de la matrice jacobienne effectué au sein de l'algorithme de Levenberg-Marquardt. Si l'on choisit T une transformation quelconque transformant le repère en un autre repère appartenant à la même orbite, c'est-à-dire que le jeu de paramètres \vec{x} mène aux mêmes mesures que le jeu de paramètres $\vec{x}' = T\vec{x}$. Ceci implique que la transformation du repère ne modifie pas la valeur des dérivées et ne doit pas modifier la trajectoire suivie par le vecteur des paramètres à estimer au cours de l'optimisation. En fait, si aucun problème ne semble surgir a priori de ce point en cas d'estimation basée sur des dérivées analytiques, la situation est différente lorsque des dérivées numériques sont employées. En règle générale, afin de minimiser le nombre d'appel à la fonction critère Φ , les dérivées numériques sont évaluées de la façon suivante (\vec{h} représente le vecteur variation appliqué à \vec{x} , c'est un vecteur dont toutes les composantes sont nulles sauf la i ème qui vaut h_i , \vec{x}_i est la i ème composante de \vec{x}) :

$$\frac{\partial \Phi}{\partial \vec{x}_i} = \frac{\Phi(\vec{x} + \vec{h}_i) - \Phi(\vec{x})}{h_i} \quad (1.46)$$

Si, désormais, nous appliquons la transformation T à cette dérivée numérique, elle devient (\vec{h}_i représente le vecteur variation appliqué à \vec{x} , c'est un vecteur dont toutes les composantes sont nulles sauf la i ème qui vaut h_i , \vec{x}_i représente la i ème composante de \vec{x}) :

$$\left. \frac{\partial \Phi}{\partial \vec{x}_i} \right|_T = \frac{\Phi(T\vec{x} + T\vec{h}_i) - \Phi(T\vec{x})}{Th_i} \quad (1.47)$$

Conserver le même parcours d'optimisation en utilisant des dérivées numériques implique de modifier le pas de calcul des dérivées numériques en fonction de la transformation T traduisant le changement de repère. La plupart des implémentations de Levenberg-Marquardt compense les changements d'échelle en pondérant le pas par la valeur du critère $\Phi_{\vec{x}}$. Ces compensations ne prennent néanmoins pas en compte les erreurs d'arrondies et le choix de repères exotiques (très grande translation du centre du repère par rapport au barycentre du modèle par exemple) peuvent mener à des trajectoires différentes pour le vecteur de paramètres au cours de l'optimisation (problème de condi-

tionnement). Les méthodes permettant de contraindre le choix du repère ont l'avantage de limiter ces effets en bornant les variations du repère.

Ce problème du choix de jauge est encore un point difficile dans la maîtrise des algorithmes d'ajustement de faisceaux. L'approche que nous présentons dans le chapitre suivant apporte une solution à ce problème.

1.6.5 Méthode par alternance

La méthode de Malis et Bartoli [Malis 01] que nous présentons introduit au sein d'un processus d'ajustement de faisceaux, basé sur une minimisation non-linéaire, un processus d'estimation a posteriori. Cette méthode n'est pas une méthode d'intersection-résection car elle maintient une inter-corrélation entre les différents paramètres. L'idée de ces travaux est proche de l'ajustement de faisceaux que nous présenterons dans le chapitre suivant.

Nous avons vu, lors de la description de la technique d'ajustement de faisceaux, que l'ensemble des paramètres associés aux caméras et aux modèles 3D était évalué simultanément par le biais d'un algorithme de minimisation non-linéaire (type Levenberg-Marquard). L'idée essentielle de Malis et Bartoli est que les paramètres intrinsèques peuvent être calculés à partir de la structure 3D et de la position des caméras. À partir de là, ils effectuent une minimisation classique des paramètres de pose et de la structure 3D puis mettent à jour les paramètres intrinsèques correspondant. Les motivations principales pour développer ce type d'algorithme sont de supprimer la nécessité d'initialiser certains paramètres (les paramètres intrinsèques dans le cas présent) et, surtout, de réduire le nombre de paramètres estimés non-linéairement et, par conséquent, les temps de calcul nécessaires. Dans [Malis 01], cette approche a été évaluée sur des données de synthèse ainsi que sur des données réelles. Les résultats obtenus montrent une réduction des temps de calcul et une légère amélioration de la précision de la reconstruction 3D obtenue.

1.7 Récapitulatif et comparaison entre ces différentes méthodes

Nous venons au cours de ce chapitre d'évoquer un certain nombre de techniques permettant d'estimer une scène 3D (caméras et structure 3D) à partir d'un ensemble d'appariements de points. Bien souvent, ces techniques sont complémentaires les unes des autres et le choix de la technique adéquate est conditionnée par le problème concret effectivement rencontré. Afin de classer ces différentes techniques, nous avons évoqué quatre critères qui nous semblaient fondamentaux afin de déterminer l'adéquation d'une méthode avec un problème. La rapidité, la précision, la généralité et la robustesse sont quatre critères essentiels mais parfois antinomiques. La figure 1.7 présente un récapitulatif des propriétés de chacune des approches. On voit très nettement se distinguer deux groupes.

Le premier groupe contient les approches visant à estimer les matrices fondamentales, les tenseurs trifocaux et les approches de type factorisation. Ces techniques sont

particulièrement adéquates pour obtenir une reconstruction, les seules informations disponibles sont les appariements entre points homologues à une même entité 3D. En effet, aucune initialisation pour les paramètres de prises de vue ou les paramètres de la structure 3D ne sont requis et, la rapidité d'exécution de ces algorithmes permet de les employer dans des schémas robustes de type RANSAC. L'utilisation de méthodes robustes de type RANSAC permet de trier les appariements et de supprimer les faux-appariements (à condition que ceux-ci ne soient ni nombreux et ni trop bien organisés!). Ces approches constituent donc les approches de prédilection pour l'initialisation des processus de reconstruction. Toutefois, ces techniques sont complexes à mettre en place et demandent des développements spécifiques lorsque les primitives 3D estimées ne sont pas des points ou des droites. Enfin, ces approches ne sont pas très résistantes au bruit présent dans la localisation des primitives dans les images.

Le second groupe est principalement constitué des méthodes itératives de type ajustement de faisceaux. Les performances de ce groupe sont le complémentaire de celles du premier groupe. D'un point de vue général, ces méthodes présentent des temps d'estimation plus élevés que les méthodes précédentes, car elles requièrent un volume de calcul plus important du fait, notamment, du calcul de dérivées première et seconde de fonctions non-linéaires dépendant de nombreux paramètres. Dans la plupart des cas, le coût de ces processus d'estimation rend impossible leur intégration dans un processus de type RANSAC. Par contre, il est généralement très facile d'intégrer des informations sur la qualité des appariements en pondérant le critère non-linéaire sur la base d'une distance de Mahalanobis. Il est également aisé de rendre robuste ces méthodes en utilisant des M-estimateurs à condition toutefois d'avoir une initialisation suffisamment bonne. Le problème majeur des méthodes non-linéaires est de converger parfois à la bonne solution, parfois à une solution fautive (minimum local qui peut être proche du minimum global) et parfois... de diverger complètement!

Algorithme	Précision	Rapidité	Généralité	Robustesse	
				Faux appariements	Localisation approximative
Méthodes linéaires	Sensible au bruit.	Rapide : Méthode linéaire ou estimation linéaire avec peu de paramètres.	Difficile à transposer à des objets complexes (modèle CAO)	Possibilité d'extraire des faux-appariements par des méthodes robustes (RANSAC)	Dans certains cas, possibilité de prendre en compte les incertitudes sur la détection des positions dans les images
Factorisation	Sensible au bruit 2D Risques liés au choix d'un point de référence	Rapide : dépend principalement de la décomposition par SVD.	Difficile à transposer à des objets complexes (modèle CAO)	Possibilité d'extraire des faux-appariements par des méthodes robustes (RANSAC)	Dans certains cas, possibilité de prendre en compte les incertitudes sur la détection des positions dans les images
Intersection - résection	Bonne précision lorsque cela converge. Fort Risque de minima locaux	Performance dépendant énormément du problème Beaucoup d'itérations	Utilisable pour tout modèle 3D paramétrique	Très sensible aux faux-appariements.	Possibilité de prendre en compte les incertitudes sur la détection des positions dans les images
Ajustement de faisceaux	Optimal en cas de bruit gaussien Risque de minima locaux	Assez lent du fait de l'estimation du hessien. Dépend de la dimension du problème	Utilisable pour tout modèle 3D paramétrique	Très sensible aux faux-appariements.	Possibilité de prendre en compte les incertitudes sur la détection des positions dans les images

FIG. 1.7: Comparaison des différentes méthodes de reconstruction.

Chapitre 2

CACHE_POS : Un nouvel algorithme d'ajustement de faisceaux

Sommaire

2.1	Introduction	51
2.2	Des algorithmes dissimulant des paramètres à l'estimation non-linéaire	53
2.2.1	Principe	53
2.2.2	Choix de l'approche	55
2.3	Présentation détaillée de CACHE_POS	56
2.3.1	Codage de CACHE_POS	57
2.3.2	Performance des algorithmes d'estimation de poses	60
2.3.3	Choix implicite du repère de reconstruction	67
2.3.4	Exploitation des structures creuses dans CACHE_POS	67
2.4	Étude expérimentale CACHE_POS	73
2.4.1	Conditions de test	73
2.4.2	Étude de la convergence	75
2.4.3	La qualité de la reconstruction	82
2.4.4	Temps de calcul	85
2.5	Bilan	87
2.6	Applications sur des données réelles	87
2.7	Conclusion	91

2.1 Introduction

Au cours du chapitre précédent, nous avons présenté plusieurs techniques de reconstruction parmi lesquelles les méthodes de type ajustement de faisceaux. Les ajustements de faisceaux présentent comme principal avantage une grande souplesse dans la modélisation du modèle 3D et des caméras (les distorsions peuvent être traitées sans

difficulté). Toutefois, l'usage de ces méthodes est limité par la nécessité de disposer d'une bonne estimée initiale tant pour le modèle 3D que pour les capteurs. Cette phase d'initialisation est plus ou moins difficile selon les situations. Lorsque le modèle 3D observé est connu (mire de calibrage) le calcul des attitudes de prises de vue et l'affectation initiale des paramètres intrinsèques des capteurs est assez simple. Par contre, lorsque le problème traité consiste à reconstruire un modèle, dont aucune reconstruction initiale n'est disponible, en utilisant des images non calibrées et non localisées, la tâche devient beaucoup plus ardue. Lorsque le modèle 3D est principalement représenté par des points ou des segments libres de toutes contraintes, les méthodes linéaires présentées dans le chapitre précédent apportent une solution possible à ce problème d'initialisation. Mais que faire lorsque le modèle est décrit avec des primitives complexes reliées entre elle par des contraintes et comment effectuer une initialisation lorsqu'un point n'est vu que dans une image et ne peut être reconstruit que grâce aux contraintes imposées par le reste de la structure 3D ? C'est pour apporter une réponse à ce problème que nous avons cherché une méthode d'ajustement de faisceaux ne dépendant plus autant que la précédente de la qualité de l'initialisation.

Afin d'atteindre cet objectif, nous avons cherché à réduire le plus possible le nombre de paramètres à initialiser. Pour y parvenir nous avons proposé de dissimuler une partie des paramètres au sein du processus d'optimisation non-linéaire (les paramètres intrinsèques ou extrinsèques liés au capteur ou bien les paramètres liés au modèle 3D). Les trois algorithmes envisagés, suite à cette idée, sont présentés dans la première partie de ce chapitre. Après avoir comparé les caractéristiques de chacune des propositions, nous avons privilégié une approche dissimulant les paramètres de pose au coeur du processus non-linéaire.

La seconde partie de ce chapitre consiste en une présentation complète de cette algorithme avec notamment une analyse du comportement de l'algorithme d'estimation de pose de Davis et Dementhon, lorsqu'il est employé loin de ses conditions nominales (modèle 3D connu, caméra calibrée).

Dans un troisième temps, nous effectuons une comparaison de la méthode proposée avec un ajustement de faisceaux classique. Le premier point abordé traite de l'existence implicite d'un repère généré par l'utilisation de notre méthode qui permet, au contraire de la méthode classique, de s'affranchir complètement de la délicate question du choix de jauge. Le second point traite des possibilités d'implémentation creuse du nouvel algorithme. Enfin, dans un troisième temps, nous présentons une comparaison des domaines de convergence, des temps de reconstruction et de la qualité de la solution obtenue entre la nouvelle approche et un ajustement de faisceaux classique.

Après avoir décrit notre approche et présenté ses caractéristiques en les comparant à celles obtenues par une approche classique, nous montrerons un exemple de reconstruction obtenue sur une scène simple permettant de vérifier l'efficacité de notre approche.

2.2 Des algorithmes dissimulant des paramètres à l'estimation non-linéaire

2.2.1 Principe

Nous avons initialement présenté les principes de cette méthode dans [Cornou 02a]. Par la suite, nous noterons ϕ_{3D} les paramètres associés au modèle 3D, ϕ_I les paramètres intrinsèques des caméras et ϕ_E les paramètres extrinsèques caractérisant la position et l'orientation des capteurs. Les paramètres associés à une image spécifique i seront discernés par un exposant et s'écriront donc sous la forme : ϕ_I^i et ϕ_E^i pour la i -ième image. Les paramètres associés à la j -ième primitive d'un modèle seront notés ϕ_{3D}^j . La scène est donc décrite par un ensemble de paramètres $\phi = (\phi_{3D}, \phi_E, \phi_I)$ que nous allons chercher à estimer. Comme nous l'avons vu dans le chapitre précédent, il existe une relation mathématique entre ces paramètres. Afin de pouvoir la manipuler simplement, nous introduisons quelques notations supplémentaires :

- $P(\phi_I^i, \phi_E^i)$ ou P_i : Matrice de projection associée à la i -ième image.
- $E(\phi_E^i)$ ou E_i : Matrice extrinsèque associée à la i -ième image.
- $I(\phi_I^i)$ ou I_i : Matrice intrinsèque associée à la i -ième image.
- $\tilde{p}_k^i = (x_k^i, y_k^i, 1)^T$: position du k -ième point 2D dans la i -ième image.

Pour l'instant, nous allons étudier le cas d'un nuage de points libres de toutes contraintes¹. La position d'un point dans un repère euclidien est directement donnée par ses coordonnées $\phi_{3D,i} = (X_i, Y_i, Z_i, 1)^T$. Lorsque l'ensemble des paramètres de la scène est exact et en l'absence de bruit, nous avons la relation suivante (pour tout point k observé dans la i -ième image) :

$$\left\| \tilde{p}_k^i - \Gamma_i \phi_{3D,k}^i \right\| = 0 \quad (2.1)$$

L'objectif de l'ajustement de faisceaux est de minimiser ces distances. En présence de bruit supposé gaussien, d'écart type σ , entachant les données détectées dans les images, le critère à minimiser est la vraisemblance qui s'écrit (P : probabilité) :

$$P(\tilde{p}|\phi) = \prod_{i,k} \frac{1}{2\pi\sigma^2} e^{-\|\tilde{p}_{k,i} - \Gamma_i \phi_{3D,k}^i\|^2} \quad (2.2)$$

En prenant $-\log(P(\tilde{p}|\phi))$, on obtient le critère à minimiser au sens du maximum de vraisemblance :

$$\Theta = -\log(P(\tilde{p}|\phi)) = \sum_{i,k} \frac{1}{\sigma^2} \|\tilde{p}_{k,i} - \Gamma_i \phi_{3D,k}^i\|^2 + Constante \quad (2.3)$$

La constante ne dépendant pas des paramètres peut être supprimée. Ce critère Θ est la fonction à minimiser dans l'ajustement de faisceaux classique. Notre méthode s'appuie sur les relations particulières existant entre ϕ_{3D} , ϕ_E , ϕ_I . En effet, si la scène

¹Le cas des modèles décrits au moyen de contraintes sera abordé dans la partie suivante.

observée ne correspond pas à un cas dégénéré, il est possible d'exprimer chaque classe de paramètres en fonction des autres :

1. $\phi_{3D} = REC(\phi_E, \phi_I)$: reconstruction d'un modèle 3D à partir de caméras calibrées et positionnées.
2. $\phi_I = CAL(\phi_E, \phi_{3D})$: calibrage interne des caméras déjà positionnées grâce à un modèle 3D connu.
3. $\phi_E = POS(\phi_I, \phi_{3D})$: calcul de la pose d'une caméra calibrée par rapport à un modèle 3D connu.

L'intégration de ces relations dans le critère Θ permet d'envisager trois nouveaux critères :

1. CACHE_REC :

$$\Theta = \sum_{i,k} \frac{1}{\sigma^2} \|\tilde{p}_{k,i} - \Gamma(\phi_{I,i}, \phi_{E|i}) REC(\phi_E, \phi_I)_k\|^2 \quad (2.4)$$

2. CACHE_CAL :

$$\Theta = \sum_{i,k} \frac{1}{\sigma^2} \|\tilde{p}_{k,i} - \Gamma(CAL(\phi_E, \phi_{3D})_i, \phi_{E,i}) \phi_{3D,k}\|^2 \quad (2.5)$$

3. CACHE_POS :

$$\Theta = \sum_{i,k} \frac{1}{\sigma^2} \|\tilde{p}_{k,i} - \Gamma(\phi_{I,i}, POS(\phi_I, \phi_{3D})_i) \phi_{3D,k}\|^2 \quad (2.6)$$

Chacun de ces critères correspond à un problème spécifique de la vision par ordinateur et supprime une des classes de paramètres de la phase de minimisation de Θ , la contrepartie étant d'estimer la classe supprimée au moyen d'une fonction spécifique (REC, CAL, POS). L'algorithme CAL est proche de l'algorithme de Malis et Bartoli [Malis 01].

Les critères de choix de la méthode sont de deux ordres. Tout d'abord, les fonctions REC, CAL, et POS correspondent à des problèmes déjà largement traités pour lesquels des solutions ont été proposées. Ces fonctions seront appelées un grand nombre de fois au cours du processus d'estimation et leur vitesse d'exécution est, par conséquent, un point important. D'autre part, il demeure nécessaire d'initialiser les paramètres qui ont été conservés dans le critère.

Dans le cas de CACHE_REC, le calcul de la position des points 3D connaissant la position et le calibrage des capteurs est une tâche simple pouvant s'effectuer très efficacement et indépendamment pour chaque point. L'initialisation des paramètres internes des capteurs impliqués par ce choix est tout à fait réaliste et le seul point délicat peut être la définition d'une position initiale pour les caméras. Dans le cas où les points 3D constituant le nuage sont reliés par des contraintes, la fonction REC devient beaucoup plus difficile à évaluer et peut impliquer l'usage d'une méthode d'optimisation non-linéaire.

Cette dernière éventualité pénalise grandement l'usage de cette méthode dans le cas contraint.

Dans le cas de CACHE_CAL, l'estimation des paramètres de calibrage à partir de la 3D et des attitudes de prise de vue implique l'utilisation d'une méthode non-linéaire dès que des paramètres du type distorsion (ou autres) vont être employés. Ces estimations peuvent être pénalisantes en terme de temps de reconstruction et le besoin de fournir une initialisation pour la 3D et les attitudes des caméras peut être problématique dans certains cas.

Dans le cas de CACHE_POS, il existe des algorithmes d'estimation de pose très efficace permettant d'envisager un très grand nombre de calculs d'attitude. Cette approche ne requiert pas l'initialisation des paramètres de pose associés aux images et présente l'avantage de supprimer les paramètres d'orientation des prises de vue de la partie non-linéaire de l'estimation (indétermination). Néanmoins, cette méthode requiert l'initialisation des paramètres du modèle 3D.

2.2.2 Choix de l'approche

Nous avons évoqué trois approches possibles (CACHE_REC, CACHE_CAL, CACHE_POS).

L'approche CACHE_CAL a été délaissée car l'initialisation des paramètres internes des capteurs est nettement moins difficile que l'apport initial d'informations sur la 3D ou sur la position des prises de vues. Cependant, les travaux de Malis et Bartoli [Malis 01] et [Bartoli 03] ont montré qu'un gain de performance était possible par cette voie.

Afin d'étudier les approches CACHE_REC et CACHE_POS, nous avons utilisé des données de synthèse constituées d'un nuage de 20 points libres et 5 prises de vue associées obtenues avec une caméra de longueur focale 2000 pixels (cf. figure 2.1). Le modèle 3D est initialisé avec l'ajout d'un bruit sur la position des points 3D et sur la position des centres optiques associés à chacune des vues. La longueur focale est initialisée à sa valeur exacte (2000 pixels). La figure 2.2 suivante montre le résultat obtenu.

Dans le cas de CACHE_REC, la fonction utilisée pour identifier la position des points 3D est une triangulation classique (au moment de la triangulation les prises de vue sont considérées calibrées et positionnées). A partir de cette initialisation, nous avons tracé la valeur du résidu à chaque itération. On observe qu'après un début de convergence rapide le critère décroît lentement en passant de plateau en plateau. Ce comportement est très proche de ceux observés dans les approches intersection-résection et rend peu séduisante cette méthode. De plus, si la fonction REC est simple dans le cas d'un nuage de points libres, elle devient très complexe lorsque des contraintes sont ajoutées au modèle. Compte tenu de ces piètres performances et de la difficulté à intégrer des contraintes dans le modèle, nous abandonnons cette voie afin de nous concentrer sur CACHE_POS.

Dans le cas de CACHE_POS, la fonction utilisée est l'algorithme de Davis et Dementhon [Davis 95](décrit en annexe). Les performances obtenues sont beaucoup plus encourageantes et la solution finale est atteinte après neuf itérations. La dissimulation des paramètres de poses associés aux prises de vue supprime le problème de leur initialisation et de la représentation de l'orientation. Compte tenu de ces éléments et des

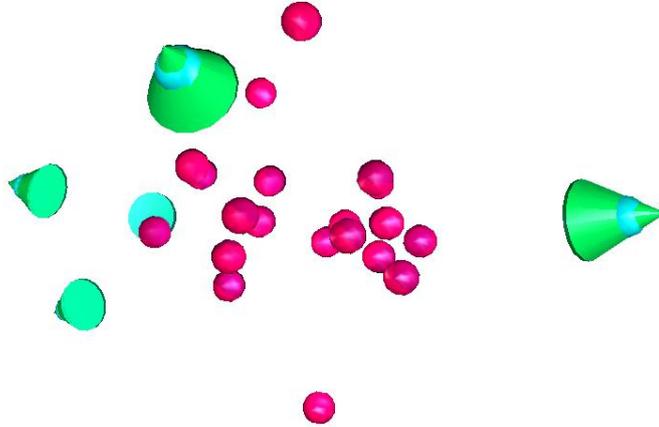


FIG. 2.1: Exemple de scène de test (En vert les positions de prise de vue et en rose les points 3D).

expériences présentées dans la suite de ce chapitre, l'algorithme CACHE_POS va être utilisé dans toute la suite de cette thèse.

2.3 Présentation détaillée de CACHE_POS

Après avoir introduit l'approche générale nous ayant mener à isoler certains paramètres au sein de l'estimation non-linéaire, nous allons détailler la méthode CACHE_POS. Cette méthode est un ajustement de faisceaux basé sur la minimisation d'un critère non-linéaire. **Cette approche n'est pas une méthode de type intersection-résection.** En effet, le modèle 3D et les paramètres de prises de vue ne sont pas traités séparément mais dans une même boucle d'optimisation. Le défaut majeur des approches intersection-résection tient à la négligence complète des inter-corrélations existant entre les paramètres du modèle 3D et ceux des caméras. Dans le cas de CACHE_POS, nous verrons que ces interactions sont prises en compte au sein du système linéaire résolu à chaque itération de l'algorithme de Levenberg-Marquardt (cf. chapitre 1).

De plus, l'intersection-résection ne calcule pas et, a fortiori, n'exploite jamais les dérivées de la fonction à minimiser. Au contraire, la force des méthodes d'ajustement de faisceaux, en général, et de CACHE_POS, en particulier, est d'utiliser les dérivées de la fonction minimisée afin de définir l'évolution du jeu de paramètres estimés. Cette stratégie est le coeur des différences de comportement entre des méthodes basées sur la seule valeur du critère ou sur ses dérivées par rapport aux paramètres.

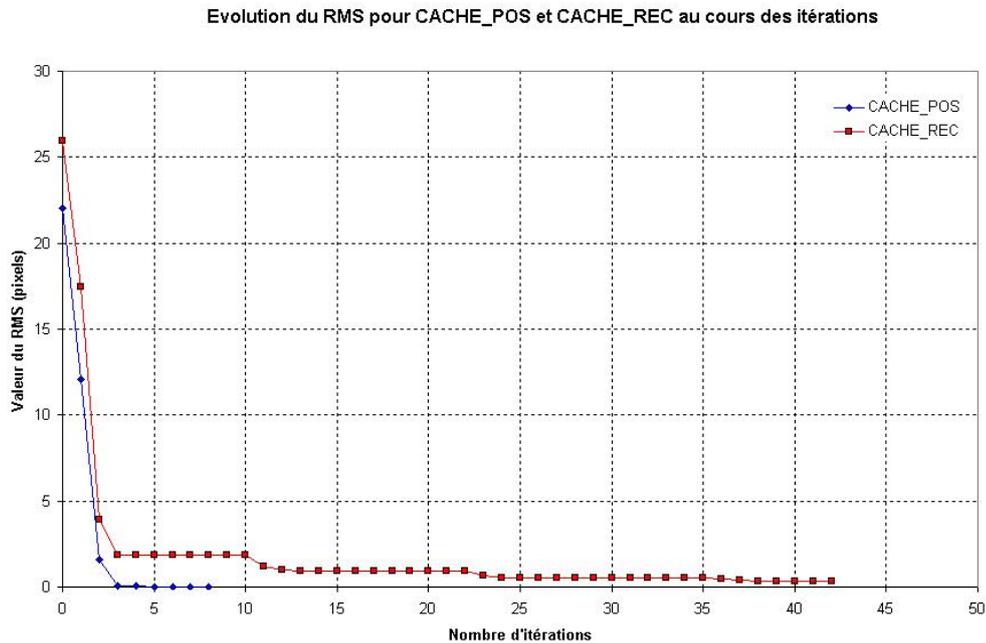


FIG. 2.2: Evolution du RMS en pixel généré par les méthodes CACHE_REC et CACHE_POS en fonction du nombre d'itérations (les croix correspondent aux itérations réussies.)

2.3.1 Codage de CACHE_POS

La méthode CACHE_POS est construit autour d'une minimisation effectuée au moyen de l'algorithme de Levenberg-Marquardt dont les principes ont été présentés dans le premier chapitre. Nous présentons à la figure 2.3 un organigramme décrivant les différentes étapes de cette méthode. CACHE_POS suit le même processus que celui présenté dans cette figure et les différences entre CACHE_POS et une méthode d'ajustement de faisceaux classique se trouvent aux niveaux des paramètres recherchés (initialisation), du calcul de la matrice Jacobienne (J), et au calcul du vecteur de mesure (ϵ). Afin de clarifier les différentes mise en oeuvre possible, nous allons décrire les trois points marquant des différences majeures avec la méthode classique. CACHE_POS cherche à minimiser le critère suivant :

$$\Theta = \sum_{i,k} \frac{1}{\sigma^2} \|\tilde{p}_{k,i} - \Gamma(\phi_{I,i}, POS(\phi_I, \phi_{3D})_i) \phi_{3D,k}\|^2 \quad (2.7)$$

Dans le cadre du modèle sténopé, la matrice de projection Γ est structurée ainsi $\Gamma = I(\phi_I)POS(\phi_I, \phi_{3D})$ et le critère devient :

$$\Theta = \sum_{i,k} \frac{1}{\sigma^2} \|\tilde{p}_{k,i} - I(\phi_I)POS(\phi_I, \phi_{3D})\phi_{3D,k}\|^2 \quad (2.8)$$

2.3.1.1 Initialisation

La phase d'initialisation de CACHE_POS consiste à constituer un vecteur \vec{x}_0 de paramètres à estimer. Les composantes de ce vecteur sont les paramètres intrinsèques (ϕ_I) des différentes caméras et les paramètres (ϕ_{3D}) associés au modèle 3D. La valeur initiale de chaque composante de \vec{x}_0 permet de fixer le point de départ du processus d'estimation. Ces valeurs ont pu être calculées au moyen d'autres méthodes (par exemple : exploitation de points de fuite pour initialiser les longueurs focales, reconstruction initiale en utilisant des méthodes linéaires, ...).

2.3.1.2 Calcul du vecteur de mesure : $\epsilon(\phi_I, \phi_{3D})$

En sortie de $\epsilon(\phi_I, \phi_{3D})$ nous obtenons un vecteur de mesure dont les composantes sont les distances entre les primitives 2D détectées dans les images et la projection des primitives 3D correspondantes dans ces mêmes images. L'algorithme de calcul du vecteur de mesure est le suivant :

ENTRÉE :

Un vecteur de paramètre $\vec{x} = [\phi_I, \phi_{3D}]$.

ESTIMATION :

1. **Calcul de la pose de chaque capteur :** pour tous les capteurs, calcul de leurs attitudes en calculant $RT_i = POS(\phi_{I,i}, \phi_{3D})$,
2. **Calcul de la mesure pour chaque primitive dans chaque image :**
 - projection de chaque primitive dans les images en calculant $\tilde{p}_{k,i} = I(\phi_{I,i})RT_i\phi_{3D,k}$,
 - stockage de la mesure dans un vecteur (k,i indicage de la mesure en fonction du numéro de la caméra et de la primitive 3D) $\epsilon(k, i) = \tilde{p}_{k,i} - \tilde{p}_{k,i}$.

SORTIE :

Le vecteur de $\epsilon(k, i)$.

Pour POS , nous avons utilisé deux algorithmes que nous décrivons en annexe :

- l'algorithme de Davis et Dementhon [Davis 95],
- l'algorithme de Lowe [Lowe 85].

L'algorithme de Davis et Dementhon est utilisé initialement car il ne requiert pas d'initialisation et permet donc d'effectuer l'ajustement de faisceaux sans connaître l'attitude des caméras au départ. Toutefois, cet algorithme n'offre pas une très bonne précision du fait de l'hypothèse para-perspective effectuée. Aussi, en fin d'estimation, cet algorithme est-il complété par celui de Lowe qui requiert une initialisation mais permet d'améliorer la précision au moyen d'une minimisation non-linéaire. D'autres méthodes d'estimation ont été proposées par Quan et Lan [Quan 99], Ameller et al. [Ameller 00] et Ansar et Daniilidis [Ansar 02], mais n'ont pas été évaluées dans le cadre de cette thèse.

2.3.1.3 Calcul de la matrice Jacobienne

Pour le calcul de la matrice Jacobienne, nous pouvons utiliser soit des dérivées analytiques soit des dérivées numériques. Nous présentons tout d'abord le cas des **dérivées numériques** qui sont les plus simples à mettre en oeuvre. La matrice Jacobienne est constituée des dérivées du vecteur de mesure ($\vec{\epsilon}$) par rapport aux paramètres recherchés au cours de l'estimation ((ϕ_I, ϕ_{3D})). Lorsque l'on fait appel à des **dérivées numériques**, on a ($e\vec{p}s_i$: vecteur de dimension égale à celle de $\phi_{I,i}$ dont toutes composantes sont nulles sauf la i-ème qui vaut eps , $e\vec{p}s_k$: vecteur de dimension égale à celle de $\phi_{3D,k}$ dont toutes composantes sont nulles sauf la k-ème qui vaut eps ,) :

$$\begin{aligned} \frac{\partial \epsilon^{[k]}(\phi_I, \phi_{3D})}{\partial \phi_{I,i}} &= \frac{\epsilon(\phi_{I,i} + e\vec{p}s_i * |\phi_{I,i}|, \phi_{3D}) - \epsilon(\phi_{I,i}, \phi_{3D})}{eps * |\phi_{I,i}|} \\ \frac{\partial \epsilon^{[k]}(\phi_I, \phi_{3D})}{\partial \phi_{3D,k}} &= \frac{\epsilon(\phi_I * |\phi_I|, \phi_{3D,k} + e\vec{p}s_k * |\phi_{3D,k}|) - \epsilon(\phi_I, \phi_{3D,k})}{eps * |\phi_{3D,k}|} \end{aligned} \quad (2.9)$$

Dans le cas des **dérivées analytiques**, nous considérons tout d'abord les dérivées par rapport aux paramètres intrinsèques des capteurs. On a :

$$\begin{aligned} \frac{\partial \epsilon^{[k]}(\phi_I, \phi_{3D})}{\partial \phi_{I,i}} &= \frac{\partial(\tilde{p}_{k,i} - I(\phi_{I,i})POS(\phi_{I,i}, \phi_{3D})\phi_{3D,k})}{\partial \phi_{I,i}} \\ &= - \frac{\partial(I(\phi_{I,i})POS(\phi_{I,i}, \phi_{3D})\phi_{3D,k})}{\partial \phi_{I,i}} \\ &= - \left(\frac{\partial(I(\phi_{I,i}))}{\partial \phi_{I,i}} POS(\phi_{I,i}, \phi_{3D})\phi_{3D,k} + I(\phi_{I,i}) \frac{\partial(POS(\phi_{I,i}, \phi_{3D}))}{\partial \phi_{I,i}} \phi_{3D,k} \right) \end{aligned} \quad (2.10)$$

Le calcul de $\frac{\partial(I(\phi_{I,i}))}{\partial \phi_{I,i}}$ est simple, il n'en va pas de même pour l'évaluation de $\frac{\partial(POS(\phi_{I,i}, \phi_{3D}))}{\partial \phi_{I,i}}$. En effet, les algorithmes d'estimation de pose utilisés sont des algorithmes itératifs. Il est difficile de calculer une expression analytique de $\frac{\partial(POS(\phi_{I,i}, \phi_{3D}))}{\partial \phi_{I,i}}$ aussi suggérons-nous d'avoir recours à des dérivées numériques. Cette utilisation de dérivées numérique n'est pas trop handicapante lorsque le coût d'un calcul de pose est faible. C'est le cas des méthodes de Dementhon et de Lowe lorsque peu de primitives 3D sont observées. Par conséquent, si l'on prend la relation suivante comme dérivée :

$$\frac{\partial(POS(\phi_{I,i}, \phi_{3D}))}{\partial \phi_{I,i}} \Big|_{num} = \frac{POS(\phi_{I,i} + e\vec{p}s_i * |\phi_{I,i}|, \phi_{3D}) - POS(\phi_{I,i}, \phi_{3D})}{eps * |\phi_{I,i}|} \quad (2.11)$$

A partir de là, $\frac{\partial \epsilon^{[k]}(\phi_I, \phi_{3D})}{\partial \phi_{I,i}}$ devient :

$$\begin{aligned} \frac{\partial \epsilon^{[k]}(\phi_I, \phi_{3D})}{\partial \phi_{I,i}} &= - \left(\left(\frac{\partial(I(\phi_{I,i}))}{\partial \phi_{I,i}} POS(\phi_{I,i}, \phi_{3D})\phi_{3D,k} + \right. \right. \\ &\quad \left. \left. I(\phi_{I,i}) \frac{POS(\phi_{I,i} + e\vec{p}s_i * |\phi_{I,i}|, \phi_{3D}) - POS(\phi_{I,i}, \phi_{3D})}{eps * |\phi_{I,i}|} \right) \right) \end{aligned} \quad (2.12)$$

De même, les dérivées par rapport aux paramètres associés au modèle 3D sont :

$$\frac{\partial \epsilon^{[k]}(\phi_I, \phi_{3D})}{\partial \phi_{3D,i}} = - \left(I(\phi_{I,i}) POS(\phi_{I,i}, \phi_{3D}) \frac{\partial \phi_{3D,k}}{\partial \phi_{3D,i}} + I(\phi_{I,i}) \frac{\partial(POS(\phi_{I,i}, \phi_{3D}))}{\partial \phi_{3D,i}} \phi_{3D,k} \right) \quad (2.13)$$

Dans la pratique et compte tenu de la seconde partie de nos travaux qui traite de l'estimation de modèles contraints, nous avons uniquement utilisé des dérivées numériques.

2.3.2 Performance des algorithmes d'estimation de poses

Les algorithmes d'estimation de poses supposent connus les paramètres intrinsèques de la caméra et le modèle 3D. Dans le cas de CACHE_POS, ces algorithmes sont employés en dehors de ces hypothèses. A partir de données de synthèse, nous avons étudié le comportement des méthodes de Dementhon et de Lowe en fonction des erreurs sur la mire 3D, du bruit sur la détection des positions des points dans les images et de l'erreur sur la valeur réelle de la longueur focale. Les deux méthodes évaluées sont les suivantes (elles sont toutes les deux présentées en annexe) :

- Dementhon : dans ce cas nous avons utilisé l'algorithme de Dementhon seul,
- Lowe : exécution de la méthode de Lowe après une initialisation par Dementhon.

Afin de caractériser la qualité de l'estimation de la pose des caméras nous avons défini deux mesures complémentaires. La première mesure évalue l'erreur sur le calcul de la position du centre optique de la caméra et la seconde mesure définit l'erreur sur l'orientation du capteur.

L'erreur en translation correspond à la distance moyenne entre le centre optique calculé au moyen des algorithmes de pose et la position parfaite. L'erreur de translation est par conséquent exprimée en unité de longueur de la scène. A titre indicatif, les points générés sont situés dans une sphère de rayon unité, et la focale de la caméra est de 2000 pixels.

L'erreur en orientation est toujours un élément difficile à définir. Nous avons choisi de définir celle-ci sous la forme d'un angle moyen d'erreur afin de faciliter la lecture des résultats. A chaque caméra (parfaite ou estimée) est associé un repère orthonormé direct dont l'orientation est décrite sous la forme d'une matrice de rotation dont les colonnes représentent l'orientation de chacun des axes du repère. L'angle moyen est la moyenne des angles existant entre les vecteurs des colonnes de la matrice de rotation associée à la caméra parfaite et ceux associés à la caméra estimée. La valeur moyenne obtenue est un angle exprimant l'écart angulaire moyen entre ces deux repères.

2.3.2.1 Influence des erreurs sur la mire 3D

Les graphiques 2.4 et 2.5 présentent les effets sur la position du centre optique et l'orientation du capteur provoqués par des erreurs sur la mire 3D. Dans cet essai, nous avons bruité la mire 3D en appliquant un bruit gaussien sur les positions des points 3D constituant la mire (le taux de bruit 3D correspond à l'écart type sur la distance par rapport au point parfait).

On observe que la position du centre optique est fortement influencée par la précision de la mire 3D. Les deux méthodes présentent une très forte erreur de localisation. On peut remarquer toutefois que la méthode de Lowe positionne la caméra moins loin de la solution que la méthode de Dementhon. En observant les erreurs d'orientation, il apparaît que, quelque soit la méthode, l'orientation du capteur est très vite fautive ce qui implique une attitude générale du capteur complètement fautive.

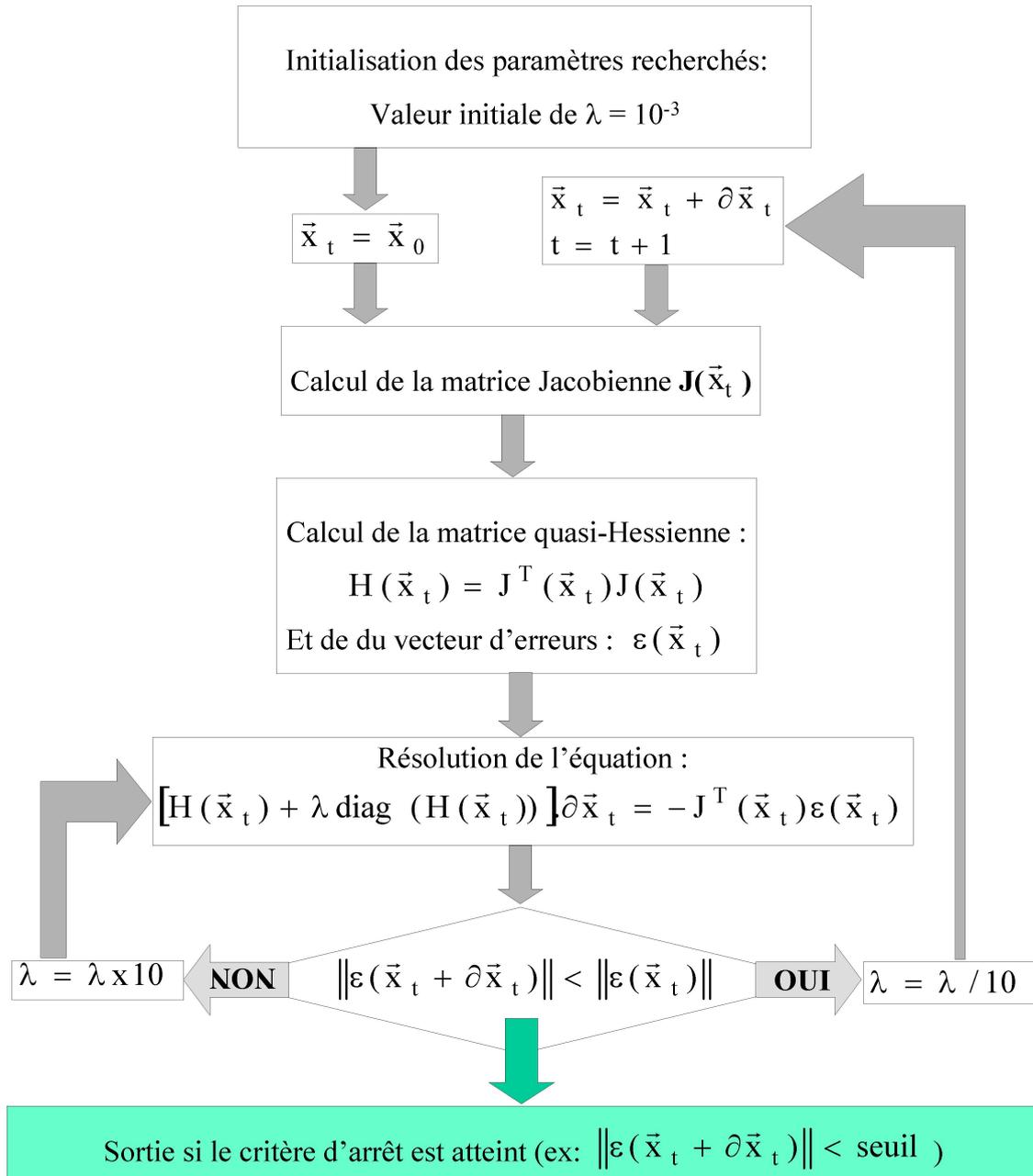


FIG. 2.3: Schéma représentant les différentes opérations effectuées lors de l'exécution de l'algorithme de Levenberg-Marquardt. \vec{x}_t représente le jeu de paramètres estimés à l'itération t (\vec{x}_0 : état initiale du vecteur de paramètres).

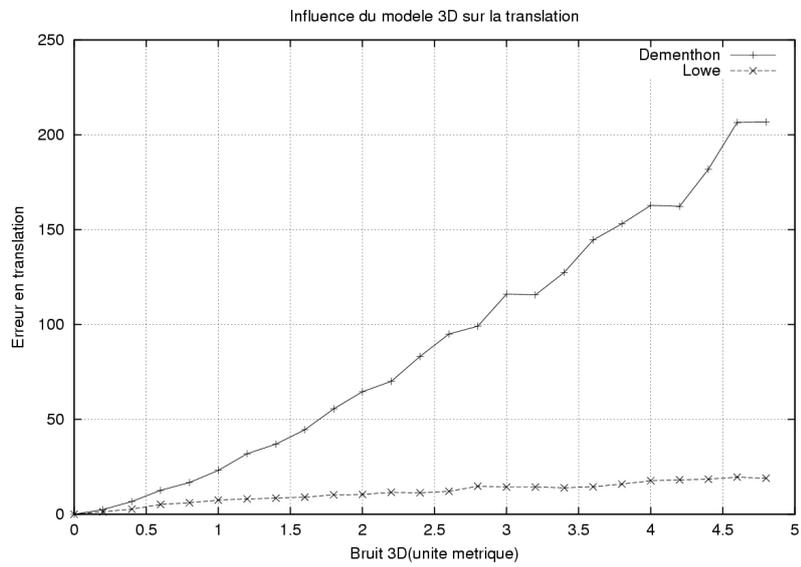


FIG. 2.4: Erreur sur la position du centre optique en fonction du bruit sur la mire 3D.

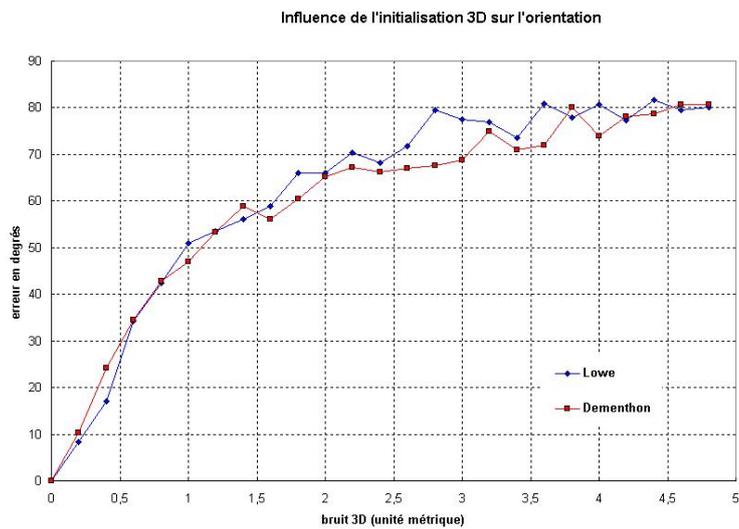


FIG. 2.5: Erreur sur l'orientation de la caméra en fonction du bruit sur la mire 3D.

2.3.2.2 Influence du bruit sur les données 2D

Dans ces essais, nous avons introduit un bruit sur les positions détectées dans les images. La première remarque, à la vue des figures 2.6 et 2.7, concerne l'influence assez faible de du bruit sur l'attitude globale de la caméra estimée. Si l'on analyse les courbes en détail, on observe que les performances de Lowe sont meilleures que celles de Dementhon. Ce résultat est naturel car la méthode de Lowe traite non-linéairement l'estimation de la pose et offre une solution au moindre carré qui prend en compte le modèle gaussien supposé sur les erreurs de localisation des primitives dans les images. Nous en concluons que la qualité de la reconstruction peut être améliorée en initialisant la pose par Dementhon puis en utilisant Lowe.

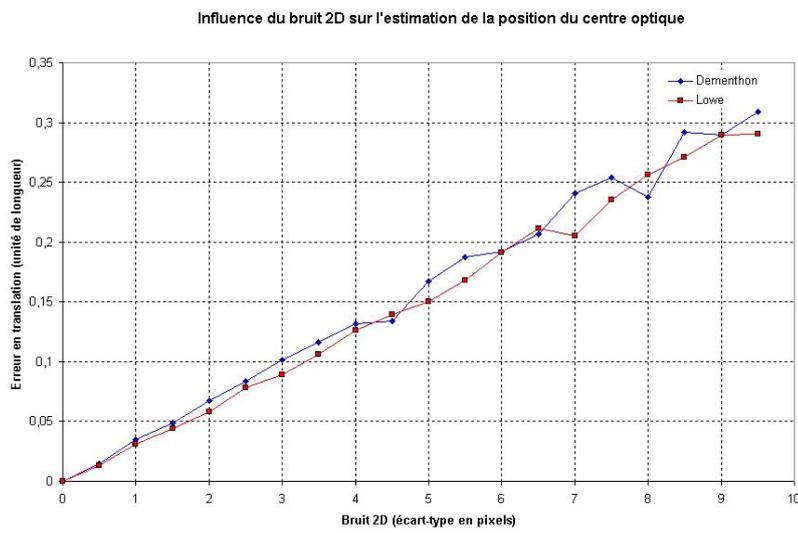


FIG. 2.6: Erreur sur la position du centre optique en fonction du bruit sur les points 2D.

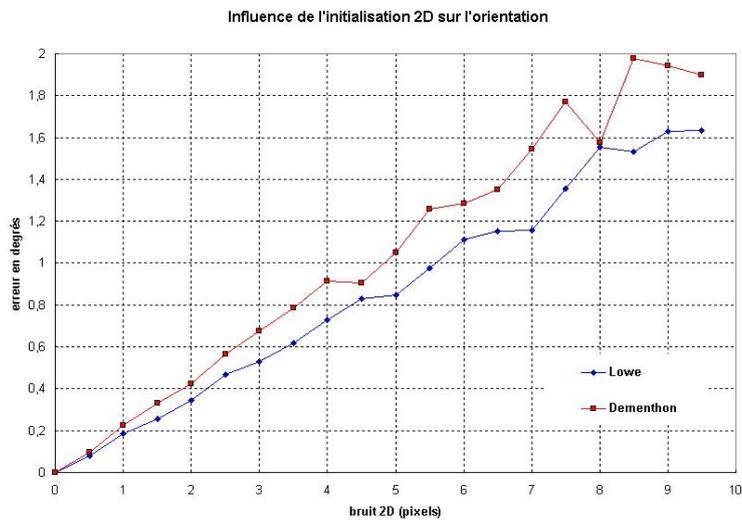


FIG. 2.7: Erreur sur l'orientation de la caméra en fonction du bruit sur les points 2D.

2.3.2.3 Influence des erreurs sur la longueur focale

Pour finir, nous avons évalué l'influence de la longueur focale sur l'estimation de la pose. La valeur parfaite de la focale était de 2000 pixels et nous avons introduit une valeur erronée comme longueur focale de la caméra afin d'évaluer l'influence de cette initialisation. On constate sur la figure 2.8 que l'erreur sur la position du centre optique est linéaire par rapport à l'erreur sur la focale. Il s'agit en fait pour les deux méthodes d'une compensation de l'erreur de la focale par un déplacement le long de l'axe optique. Vis-à-vis de l'orientation du capteur, on observe sur la figure 2.9 une erreur très importante sur l'orientation du capteur lorsque les longueurs focales diminuent. L'explication tient au fait que la focale diminuant, la caméra s'approche de l'objet et finit par atteindre une situation en contradiction avec l'hypothèse para-perspective de l'algorithme de Dementhon supposant que la profondeur de l'objet est faible par rapport à la distance entre l'objet et la caméra. A l'opposé, on observe une asymptote pour les grandes valeurs de longueur focale. Cette asymptote traduit le passage d'un modèle de type projectif à un modèle orthographique à l'échelle. Nous verrons, par la suite, que cette asymptote sera la cause d'un phénomène de non-convergence lors de l'utilisation de CACHE_POS.

Nous venons de voir par le biais d'expérience de synthèse que l'algorithme CACHE_POS va utiliser des méthodes d'estimation de poses très loin de leur condition nominale d'emploi. L'estimation des poses est très sensible aux erreurs sur la mire 3D supposée. Au contraire, ces méthodes, et plus particulièrement celle de Lowe, sont peu sensibles au bruit associé à la détection des primitives 2D dans les images. Enfin, la connaissance de la focale ne requiert pas une extrême précision même si de faibles valeurs sont à éviter compte tenu des divergences très prononcées qu'elle provoque sur l'attitude estimée du capteur. A la vue de ces différents éléments, la méthode de Lowe semble être la plus résistante et la plus précise. Cependant, la vitesse d'exécution et l'absence d'initialisation font de l'algorithme de Dementhon un candidat de choix pour le début de la phase de reconstruction.

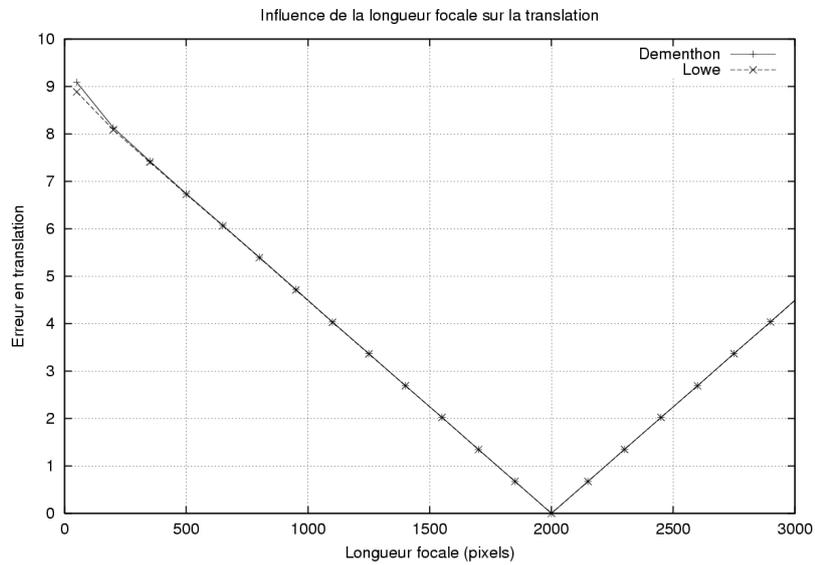


FIG. 2.8: Erreur sur la position du centre optique en fonction de l'erreur sur la longueur focale.

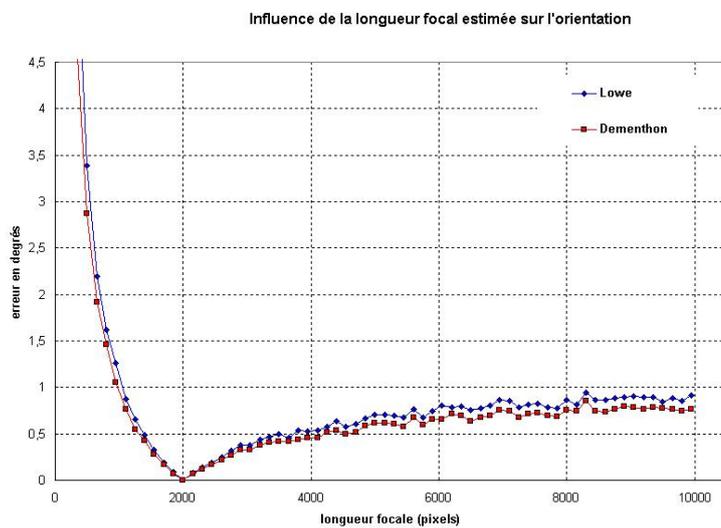


FIG. 2.9: Erreur sur l'orientation de la caméra en fonction de l'erreur sur la longueur focale.

2.3.3 Choix implicite du repère de reconstruction

Le choix du repère de reconstruction est une tâche délicate lors de l'utilisation d'une technique d'ajustement de faisceaux classique. Les possibilités offertes sont multiples et consistent pour la plupart à fixer des paramètres de la scène (paramètres associés aux prises de vue ou au modèle 3D) ou bien à introduire de nouvelles équations afin de supprimer les degrés de liberté superflus (rotation, translation et facteur d'échelle \Rightarrow sept degrés de liberté) (voir 1.6.4).

Plaçons-nous dans le cas où le repère n'a pas été fixé. Si l'on utilise un ajustement de faisceau classique, l'équation 1.36 du premier chapitre correspond à un système linéaire sous-déterminé puisqu'il manque sept équations pour obtenir une solution unique.

Dans le cas de l'algorithme CACHE_POS, au moment où l'équation 1.36 est résolue, les poses des prises de vue ont déjà été fixées par l'exécution de la fonction de calcul des poses. Au moment de la résolution de l'équation normale, les attitudes des prises de vues sont considérées comme connues, leurs positions et orientations imposant un repère totalement contraint au système. Par conséquent, l'équation normale résolue dans la partie non-linéaire de CACHE_POS est totalement déterminée.

Le choix d'une jauge reste encore aujourd'hui un élément délicat dans la définition d'un algorithme d'ajustement de faisceaux mais CACHE_POS, en dissimulant les paramètres de poses, apporte une solution originale à ce problème du choix de jauge.

2.3.4 Exploitation des structures creuses dans CACHE_POS

Nous avons vu précédemment que la matrice $J^T J$ présente dans le cas classique une structure particulière par blocs (cf. figure 2.10). L'exploitation de cette structure creuse permet d'améliorer grandement les performances de la méthode classique en réduisant l'espace mémoire requis à l'exécution et en accélérant le processus d'estimation. Ce gain de temps d'exécution est principalement dû au remplacement de l'inversion de la matrice quasi-hessienne de grande dimension par un ensemble d'inversions de matrices de petites dimensions.

Nous allons voir que la structure de CACHE_POS n'est pas aussi creuse que celle de la méthode classique. Nous avons vu, dans le premier chapitre, que la méthode de Levenberg-Marquardt calcule les modifications à appliquer aux paramètres en résolvant un système de la forme $A\delta\vec{x} = B$ qui devient $J^T J\delta\vec{x} = -J^T \epsilon(\vec{x})$ (J : matrice jacobienne, $\delta\vec{x}$: modification à apporté au vecteur de paramètre \vec{x} , $\epsilon(\vec{x})$: vecteurs de mesures associés à \vec{x}) dans le cas d'une minimisation par la méthode de Newton. Dans un premier temps, nous allons détailler la structure de la matrice jacobienne dans le cas de la méthode classique et dans le cas de la méthode CACHE_POS. Dans une seconde partie, nous allons comparer les structures des matrices quasi-hessiennes associées à ces deux méthodes et en déduire les champs d'applications privilégiés de chacune de ces méthodes.

2.3.4.1 Cas classique

Soient $(x_j^i, y_j^i, 1)^T$ la position du point i dans l'image j , m_j^i la mesure (le résidu) associée au point i dans l'image j , k l'indice d'une caméra quelconque, l l'indice d'un point 3D quelconque et P_j la matrice de projection associée à la caméra j dépendant des paramètres associés à cette caméra. On a :

$$m_j^i = \left\| \begin{pmatrix} x_j^i \\ y_j^i \\ 1 \end{pmatrix} - P_j \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \right\| \quad (2.14)$$

A partir de cette relation, nous pouvons calculer les dérivées partielles présentes dans la matrice jacobienne $J_{classique}$ (Nous ne tenons compte que d'un paramètre interne f_k).

– **Paramètres intrinsèques :** $\frac{\delta m_j^i}{\delta f_k} = -\frac{\delta P_j}{\delta f_k} \cdot P_{3D}^l$

– **Paramètres extrinsèques :**

$$\begin{array}{l|l} \frac{\delta m_j^i}{\delta T x_k} = -\frac{\delta P_j}{\delta T x_k} \cdot P_{3D}^l & \frac{\delta m_j^i}{\delta \alpha_k} = -\frac{\delta P_j}{\delta \alpha_k} \cdot P_{3D}^l \\ \frac{\delta m_j^i}{\delta T y_k} = -\frac{\delta P_j}{\delta T y_k} \cdot P_{3D}^l & \frac{\delta m_j^i}{\delta \beta_k} = -\frac{\delta P_j}{\delta \beta_k} \cdot P_{3D}^l \\ \frac{\delta m_j^i}{\delta T z_k} = -\frac{\delta P_j}{\delta T z_k} \cdot P_{3D}^l & \frac{\delta m_j^i}{\delta \gamma_k} = -\frac{\delta P_j}{\delta \gamma_k} \cdot P_{3D}^l \end{array} \quad (2.15)$$

– **Paramètres 3D :**

$$\begin{array}{l} \frac{\delta m_j^i}{\delta X_l} = -P_j \cdot \frac{P_{3D}^l}{\delta X_l} \\ \frac{\delta m_j^i}{\delta Y_l} = -P_j \cdot \frac{P_{3D}^l}{\delta Y_l} \\ \frac{\delta m_j^i}{\delta Z_l} = -P_j \cdot \frac{P_{3D}^l}{\delta Z_l} \end{array} \quad (2.16)$$

La plupart des dérivées présentées si dessus sont nulles. En effet, la variation d'un paramètre caractérisant la caméra k (tous les autres paramètres de la scène étant fixes) implique uniquement une modification des mesures effectuées dans l'image associée à cette caméra. Cela implique que la dérivée partielle par rapport à toute mesure effectuée dans une autre image est nulle. Par conséquent, dans le tableau 2.17, les dérivées partielles par rapport à un paramètre de la caméra k sont nulles si $j \neq k$ (par exemple : $\frac{\delta m_j^i}{\delta f_k} = 0$ pour $j \neq k$).

D'autre part, la variation d'un paramètre associé à un point 3D ne modifie que les mesures associées à ce point. Cela signifie que les dérivées partielles par rapport aux paramètres d'un point 3D i sont nulles si $i \neq l$ (par exemple : $\frac{\delta m_j^i}{\delta X_l} = 0$ pour $i \neq l$).

Prenons un cas simple, imaginons deux points 3D, observés dans deux images, paramétrés par leurs coordonnées (X,Y,Z) et deux caméras caractérisées chacune par une longueur focale. Les différents éléments sont paramétrés de la façon suivante :

– Caméra 1 : $f_1, T x_1, T y_1, T z_1, \alpha_1, \beta_1, \gamma_1$

- Caméra 2 : $f_2, Tx_2, Ty_2, Tz_2, \alpha_2, \beta_2, \gamma_2$
- Point 1 : X_1, Y_1, Z_1
- Point 2 : X_2, Y_2, Z_2

La matrice jacobienne $J_{classique}$ a la structure suivante :

$$J_{classique}^T = \begin{pmatrix} \frac{\delta mesure}{\delta parametre} & m_1^1 & m_1^2 & m_2^1 & m_2^2 \\ f_1 & \frac{\delta m_1^1}{\delta f_1} & \frac{\delta m_1^2}{\delta f_1} & 0 & 0 \\ Tx_1 & \frac{\delta m_1^1}{\delta Tx_1} & \frac{\delta m_1^2}{\delta Tx_1} & 0 & 0 \\ Ty_1 & \frac{\delta m_1^1}{\delta Ty_1} & \frac{\delta m_1^2}{\delta Ty_1} & 0 & 0 \\ Tz_1 & \frac{\delta m_1^1}{\delta Tz_1} & \frac{\delta m_1^2}{\delta Tz_1} & 0 & 0 \\ \alpha_1 & \frac{\delta m_1^1}{\delta \alpha_1} & \frac{\delta m_1^2}{\delta \alpha_1} & 0 & 0 \\ \beta_1 & \frac{\delta m_1^1}{\delta \beta_1} & \frac{\delta m_1^2}{\delta \beta_1} & 0 & 0 \\ \gamma_1 & \frac{\delta m_1^1}{\delta \gamma_1} & \frac{\delta m_1^2}{\delta \gamma_1} & 0 & 0 \\ f_2 & 0 & 0 & \frac{\delta m_2^1}{\delta f_2} & \frac{\delta m_2^2}{\delta f_2} \\ Tx_2 & 0 & 0 & \frac{\delta m_2^1}{\delta Tx_2} & \frac{\delta m_2^2}{\delta Tx_2} \\ Ty_2 & 0 & 0 & \frac{\delta m_2^1}{\delta Ty_2} & \frac{\delta m_2^2}{\delta Ty_2} \\ Tz_2 & 0 & 0 & \frac{\delta m_2^1}{\delta Tz_2} & \frac{\delta m_2^2}{\delta Tz_2} \\ \alpha_2 & 0 & 0 & \frac{\delta m_2^1}{\delta \alpha_2} & \frac{\delta m_2^2}{\delta \alpha_2} \\ \beta_2 & 0 & 0 & \frac{\delta m_2^1}{\delta \beta_2} & \frac{\delta m_2^2}{\delta \beta_2} \\ \gamma_2 & 0 & 0 & \frac{\delta m_2^1}{\delta \gamma_2} & \frac{\delta m_2^2}{\delta \gamma_2} \\ X_1 & \frac{\delta m_1^1}{\delta X_1} & \mathbf{0} & \frac{\delta m_2^1}{\delta X_1} & \mathbf{0} \\ Y_1 & \frac{\delta m_1^1}{\delta Y_1} & \mathbf{0} & \frac{\delta m_2^1}{\delta Y_1} & \mathbf{0} \\ Z_1 & \frac{\delta m_1^1}{\delta Z_1} & \mathbf{0} & \frac{\delta m_2^1}{\delta Z_1} & \mathbf{0} \\ X_2 & \mathbf{0} & \frac{\delta m_1^2}{\delta X_2} & \mathbf{0} & \frac{\delta m_2^2}{\delta X_2} \\ Y_2 & \mathbf{0} & \frac{\delta m_1^2}{\delta Y_2} & \mathbf{0} & \frac{\delta m_2^2}{\delta Y_2} \\ Z_2 & \mathbf{0} & \frac{\delta m_1^2}{\delta Z_2} & \mathbf{0} & \frac{\delta m_2^2}{\delta Z_2} \end{pmatrix} \quad (2.17)$$

On constate la présence de nombreux éléments nuls qui permettrons une résolution efficace par l'utilisation des approches creuses présentées dans le chapitre 1.

2.3.4.2 Cas CACHE_POS

Nous reprenons les mêmes notations que dans le paragraphe précédent. La principale différence est que la matrice de projection P_j qui dépendait auparavant des paramètres internes et des paramètres de pose de la caméra j dépend désormais des paramètres intrinsèques et des paramètres de la scène 3D (l'attitude de la caméra est déterminée par un algorithme d'estimation de pose). Par conséquent, les mesures sont toujours égales à :

$$m_j^i = \left\| \begin{pmatrix} x_j^i \\ y_j^i \\ 1 \end{pmatrix} - P_j \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \right\| \quad (2.18)$$

mais les dérivées partielles présentent dans la matrice jacobienne J_{CACHE_POS} valent désormais :

– **Paramètres intrinsèques :** $\frac{\delta m_j^i}{\delta f_k} = -\frac{\delta P_j}{\delta f_k} \cdot P_{3D}^l$

– **Paramètres extrinsèques :**

$$\begin{array}{l|l} \frac{\delta m_j^i}{\delta T x_k} = -\frac{\delta P_j}{\delta T x_k} \cdot P_{3D}^l & \frac{\delta m_j^i}{\delta \alpha_k} = -\frac{\delta P_j}{\delta \alpha_k} \cdot P_{3D}^l \\ \frac{\delta m_j^i}{\delta T y_k} = -\frac{\delta P_j}{\delta T y_k} \cdot P_{3D}^l & \frac{\delta m_j^i}{\delta \beta_k} = -\frac{\delta P_j}{\delta \beta_k} \cdot P_{3D}^l \\ \frac{\delta m_j^i}{\delta T z_k} = -\frac{\delta P_j}{\delta T z_k} \cdot P_{3D}^l & \frac{\delta m_j^i}{\delta \gamma_k} = -\frac{\delta P_j}{\delta \gamma_k} \cdot P_{3D}^l \end{array} \quad (2.19)$$

– **Paramètres 3D :**

$$\begin{array}{l} \frac{\delta m_j^i}{\delta X_l} = -\frac{P_j \cdot P_{3D}^l}{\delta X_l} \\ \frac{\delta m_j^i}{\delta Y_l} = -\frac{P_j \cdot P_{3D}^l}{\delta Y_l} \\ \frac{\delta m_j^i}{\delta Z_l} = -\frac{P_j \cdot P_{3D}^l}{\delta Z_l} \end{array} \quad (2.20)$$

La matrice de projection associée aux caméras dépend désormais des paramètres 3D. On constate que les dérivées par rapport aux paramètres des caméras sont inchangées par rapport à la méthode classique. Par conséquent, dans le tableau 2.19, les dérivées partielles par rapport à un paramètre de la caméra k sont nulles si $j \neq k$.

Par contre, la variation d'un paramètre associé à un point 3D modifie la pose de chacune des caméras observant ce point et, par conséquent, l'ensemble des mesures effectuées dans les images associées à ces caméras.

Reprenons l'exemple précédent avec deux points 3D et deux caméras :

La matrice jacobienne J_{CACHE_POS} a la structure suivante :

$$J_{CACHE_POS}^T = \begin{pmatrix} \frac{\delta mesure}{\delta parametre} & m_1^1 & m_1^2 & m_2^1 & m_2^2 \\ f_1 & \frac{\delta m_1^1}{\delta f_1} & \frac{\delta m_1^2}{\delta f_1} & 0 & 0 \\ Tx_1 & \frac{\delta m_1^1}{\delta Tx_1} & \frac{\delta m_1^2}{\delta Tx_1} & 0 & 0 \\ Ty_1 & \frac{\delta m_1^1}{\delta Ty_1} & \frac{\delta m_1^2}{\delta Ty_1} & 0 & 0 \\ Tz_1 & \frac{\delta m_1^1}{\delta Tz_1} & \frac{\delta m_1^2}{\delta Tz_1} & 0 & 0 \\ \alpha_1 & \frac{\delta m_1^1}{\delta \alpha_1} & \frac{\delta m_1^2}{\delta \alpha_1} & 0 & 0 \\ \beta_1 & \frac{\delta m_1^1}{\delta \beta_1} & \frac{\delta m_1^2}{\delta \beta_1} & 0 & 0 \\ \gamma_1 & \frac{\delta m_1^1}{\delta \gamma_1} & \frac{\delta m_1^2}{\delta \gamma_1} & 0 & 0 \\ f_2 & 0 & 0 & \frac{\delta m_2^1}{\delta f_2} & \frac{\delta m_2^2}{\delta f_2} \\ Tx_2 & 0 & 0 & \frac{\delta m_2^1}{\delta Tx_2} & \frac{\delta m_2^2}{\delta Tx_2} \\ Ty_2 & 0 & 0 & \frac{\delta m_2^1}{\delta Ty_2} & \frac{\delta m_2^2}{\delta Ty_2} \\ Tz_2 & 0 & 0 & \frac{\delta m_2^1}{\delta Tz_2} & \frac{\delta m_2^2}{\delta Tz_2} \\ \alpha_2 & 0 & 0 & \frac{\delta m_2^1}{\delta \alpha_2} & \frac{\delta m_2^2}{\delta \alpha_2} \\ \beta_2 & 0 & 0 & \frac{\delta m_2^1}{\delta \beta_2} & \frac{\delta m_2^2}{\delta \beta_2} \\ \gamma_2 & 0 & 0 & \frac{\delta m_2^1}{\delta \gamma_2} & \frac{\delta m_2^2}{\delta \gamma_2} \\ X_1 & \frac{\delta m_1^1}{\delta X_1} & \frac{\delta m_1^2}{\delta X_1} & \frac{\delta m_2^1}{\delta X_1} & \frac{\delta m_2^2}{\delta X_1} \\ Y_1 & \frac{\delta m_1^1}{\delta Y_1} & \frac{\delta m_1^2}{\delta Y_1} & \frac{\delta m_2^1}{\delta Y_1} & \frac{\delta m_2^2}{\delta Y_1} \\ Z_1 & \frac{\delta m_1^1}{\delta Z_1} & \frac{\delta m_1^2}{\delta Z_1} & \frac{\delta m_2^1}{\delta Z_1} & \frac{\delta m_2^2}{\delta Z_1} \\ X_2 & \frac{\delta m_1^1}{\delta X_2} & \frac{\delta m_1^2}{\delta X_2} & \frac{\delta m_2^1}{\delta X_2} & \frac{\delta m_2^2}{\delta X_2} \\ Y_2 & \frac{\delta m_1^1}{\delta Y_2} & \frac{\delta m_1^2}{\delta Y_2} & \frac{\delta m_2^1}{\delta Y_2} & \frac{\delta m_2^2}{\delta Y_2} \\ Z_2 & \frac{\delta m_1^1}{\delta Z_2} & \frac{\delta m_1^2}{\delta Z_2} & \frac{\delta m_2^1}{\delta Z_2} & \frac{\delta m_2^2}{\delta Z_2} \end{pmatrix} \quad (2.21)$$

On observe que la zone concernant les dérivées par rapport aux paramètres 3D a perdu sa structure creuse. Néanmoins, la partie supérieure de la matrice jacobienne présente toujours une structure creuse qui permet, la aussi, d'optimiser la résolution du système linéaire.

2.3.4.3 Comparaison des matrices quasi-hessiennes

A partir des matrices jacobienes $J_{classique}$ et J_{CACHE_POS} , nous pouvons calculer les matrices quasi-hessiennes associées à chacune de ces méthodes. L'inter-corrélation existant dans le cas de la méthode CACHE_POS entre les paramètres associés au modèle 3D et l'ensemble des mesures conduit à une matrice $H_{CACHE_POS} = J_{CACHE_POS}^T J_{CACHE_POS}$ moins creuse que la matrice $H_{classique} = J_{classique}^T J_{classique}$ associée à l'approche classique. La figure 2.10 montre la différence entre ces deux méthodes.

Afin de mettre en évidence l'importance de ces dérivées, nous avons tracé l'évolution du critère au cours des itérations, dans un cas (nommé CACHE_POS) en considérant la matrice quasi-hessienne complète, dans l'autre (nommé CACHE_POS creuse) en négligeant les éléments traduisant les inter-corrélations existants entre les paramètres

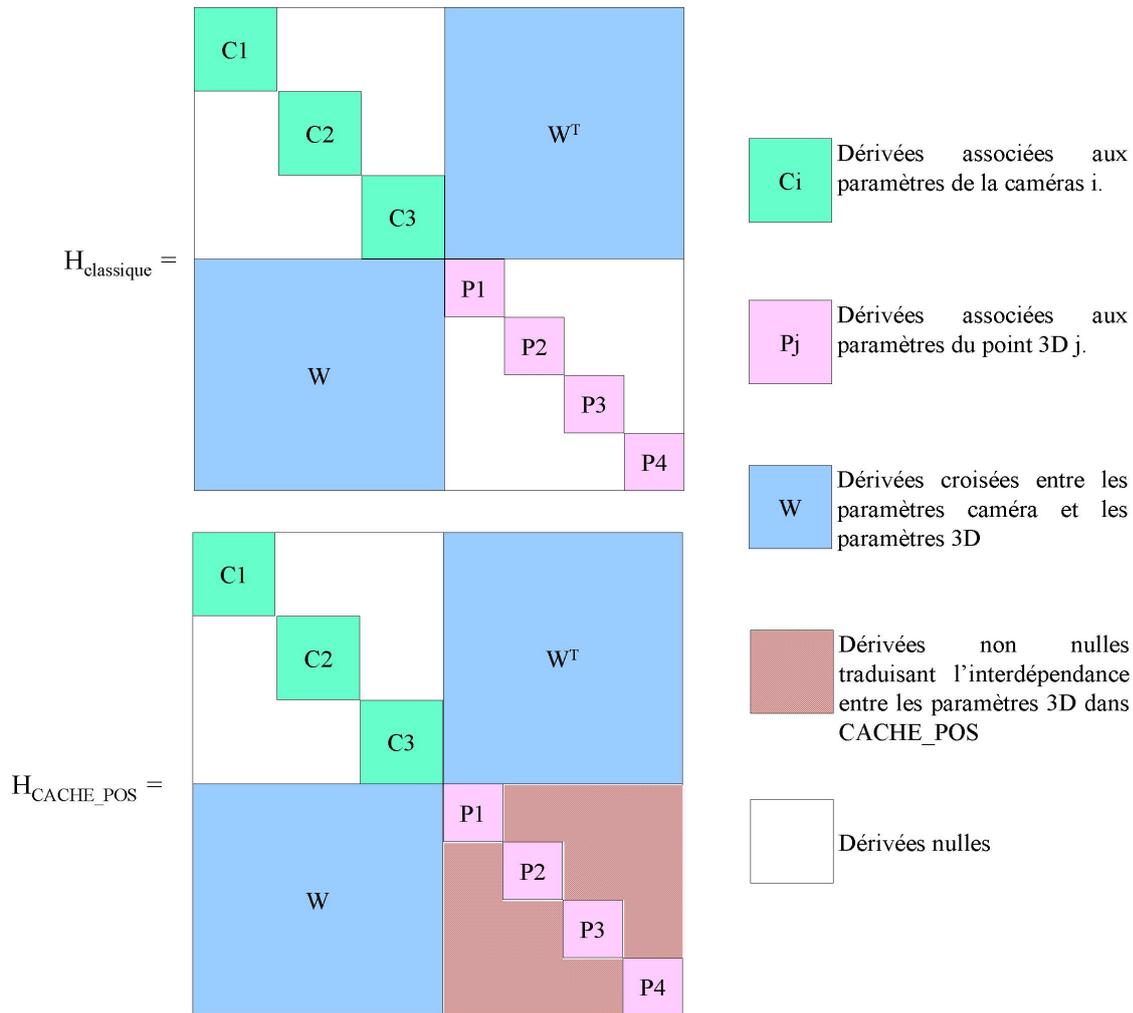


FIG. 2.10: Structure creuse de la matrice quasi-hessienne pour la méthode classique et la méthode CACHE_POS.

3D (nous les fixons arbitrairement à zéros). On constate sans équivoque que la convergence est beaucoup plus lente lorsque les inter-corrélations ne sont pas prises en compte. Ce comportement ressemble à celui observé lors de l'utilisation de méthodes du type intersection-résection. La prise en compte des inter-corrélations est un élément décisif dans la convergence de la méthode.

En conclusion, nous avons montré que la matrice quasi-hessienne de la méthode CACHE_POS présente une structure creuse au niveau des paramètres associés aux caméras et une structure pleine pour les paramètres associés au modèle 3D. Par conséquent, la méthode proposée sera moins performante que la méthode classique dans le cas des reconstructions présentant un grand nombre de paramètres 3D. En contrepartie, la reconstruction d'un modèle 3D décrit par peu de paramètres tirera pleinement partie des caractéristiques de CACHE_POS. Par conséquent, la méthode CACHE_POS et

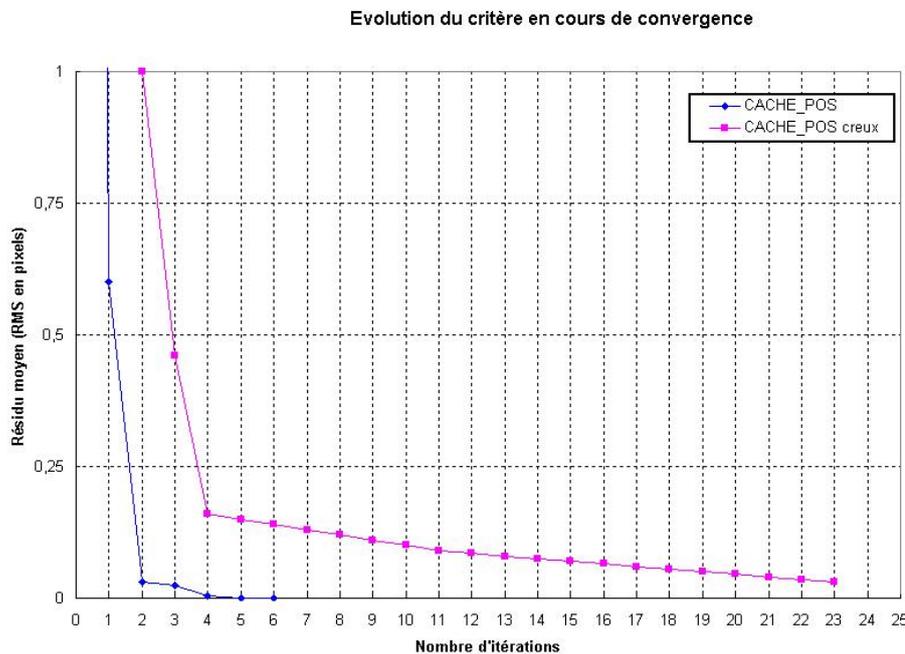


FIG. 2.11: Evolution du critère en fonction du nombre d'itérations pour les différentes approches.

la méthode classique apparaissent comme deux méthodes complémentaires, la première traitant les cas présentant un grand nombre d'images et la seconde traitant les modèles décrits par un grand nombre de paramètres 3D.

2.4 Étude expérimentale CACHE_POS

Afin d'étudier le comportement de notre algorithme, nous avons mené des expériences sur des données de synthèse. L'objectif de ces expériences est de permettre une meilleure compréhension du comportement de CACHE_POS. La difficulté de ce type d'évaluation tient principalement au caractère arbitraire de tels essais. Notre étude et ses résultats sont, par conséquent, à lire au regard des conditions d'expérimentation et des hypothèses de départ.

2.4.1 Conditions de test

Une première difficulté a résidé dans la volonté de comparer deux méthodes (la classique et CACHE_POS) dont les hypothèses de départ ne sont pas identiques. En effet, CACHE_POS ne requiert aucune initialisation des poses alors que la méthode classique implique une attitude initiale pour chaque image. La conservation de conditions de test objectives implique de définir des données initiales pour la méthode classique permettant

de ne pas trop biaiser les résultats. Compte tenu qu'à la première itération, CACHE_POS détermine la pose de chacune des images au moyen de l'algorithme de Dementhon, nous avons décidé d'employer cette méthode pour initialiser l'ajustement de faisceaux classique. Évidemment, une méthode complète initialiserait l'ajustement de faisceaux au moyen d'algorithmes (basés sur des méthodes de factorisation, l'estimation de matrices fondamentales, le calcul de tenseurs tri-focaux,...) tels que ceux décrits dans le chapitre 1, mais ce choix dissimulerait les différences de comportements existant entre les deux méthodes étudiées ici. La seconde difficulté a été de choisir une scène 3D suffisamment générale, afin d'éviter les conclusions trop spécifiques et, suffisamment simple pour pouvoir analyser les résultats. Nous avons choisi d'effectuer nos tests en synthétisant un nuage de points 3D libres de toutes contraintes. Nous considérons dans toute cette étude que les caméras sont dotées de pixels parfaitement carrés et que le point principal est localisé au centre de l'image. Les paramètres estimés sont une distance focale ($f_x = f_y$), la pose des caméras et les paramètres du modèles 3D. Ces éléments sont minimalistes mais correspondent à ce qu'il est raisonnable d'estimer sur des données fortement bruitées et à partir d'initialisations de mauvaise (voir même très mauvaise) qualité telles que nous les rencontrerons dans la suite de cette thèse lorsqu'il s'agira de traiter de la reconstruction de bâtiments. Ce problème est très différent des applications visant à estimer précisément les paramètres internes des capteurs pour lesquelles les méthodes classiques apportent une solution à compter du moment où une bonne initialisation est disponible. Les performances obtenues par la méthode classique dans les tests à venir sont, par conséquent, à analyser en tenant compte du fait que nous l'utilisons ici à l'extrême limite de son champ d'application.

La scène observée est un nuage de points 3D (cf. figure 2.1). Les points sont choisis aléatoirement (au moyen d'une loi uniforme) à l'intérieur d'une sphère de rayon 1. Chaque position de prise de vue est choisie aléatoirement autour de la sphère et l'axe de visée est orienté vers le centre du nuage de points. Le nombre de points vus dans chaque image est conditionné par un taux de visibilité (rapport entre le nombre de projections 2D définies et le nombre de points 3D existant) de sorte que les points 3D ne soient pas visibles dans toutes les images.

Afin de discerner l'influence de chacun des paramètres, nous avons utilisé divers scénarii, tous basés sur ce modèle du nuage de points libres. Trois propriétés principales ont été analysées :

- la convergence,
- la qualité de la reconstruction,
- le temps de calcul.

2.4.2 Etude de la convergence

Dans le cadre d'une optimisation non-linéaire, la convergence est obtenue lorsque les variations à appliquer au vecteur de paramètres, pour atteindre l'état à l'itération $t+1$ sont nulles. Cette définition correspond en fait à la découverte d'un minimum de la fonction minimisée au cours de l'ajustement de faisceaux. Cet état de stabilité du jeu de paramètres n'implique pas nécessairement qu'une solution pertinente, vis-à-vis du problème de reconstruction posé, ait été trouvée. En effet, il se peut fort bien qu'un minimum local ait été trouvé. Ce minimum remplit parfaitement le critère de stabilité de la solution, mais n'offre pas une solution acceptable au problème de reconstruction. Afin de ne pas comptabiliser ces solutions comme des succès, nous avons défini la convergence en y ajoutant des contraintes. Non seulement la solution doit être stationnaire, mais en plus, les seules solutions acceptables sont celles dont les valeurs des longueurs focales estimées sont comprises entre 300 et 100000 pixels et dont le résidu moyen final est au plus supérieur de 0.5 pixels à la précision de détection des primitives dans les images. Ces critères supplémentaires ont été choisis arbitrairement dans l'esprit de ce qui est détectable sans ambiguïté par un utilisateur.

Afin d'étudier les propriétés de CACHE_POS en terme de convergence, nous avons réalisé des tests en utilisant des scènes intégralement simulées. Le modèle 3D est constitué d'un ensemble de points disposés à l'intérieur d'une sphère unitaire. Lors des expériences nécessitant de perturber ce nuage de points, un bruit gaussien est ajouté à la position de chaque point 3D. L'écart type associé à ce bruit correspond à la distance entre le point initial et le point bruité. De même, les projections 2D des points dans les images sont également perturbées au moyen d'un bruit gaussien. Hormis pour les expériences analysant les effets de longueurs focales variables, la longueur focale lors des différentes prises a une valeur parfaite de 500 pixels. Un bruit suivant une loi normale d'écart type 20 pixels est systématiquement appliqué à cette valeur pour les essais ne concernant pas l'influence de l'initialisation de la focale. Sauf mention contraire, les expériences ont été menées sur 20 points 3D, 5 images et 50 essais par mesure. Des jeux de données rigoureusement identiques ont été employés avec les deux méthodes.

Une méthode d'optimisation non-linéaire est un processus itératif qui, dans notre cas, possède un caractère déterministe au sens où pour une condition initiale donnée l'état final obtenu en fin d'optimisation est toujours le même. A partir de ce constat, le taux de convergences réussies dépend, tout d'abord, de la nature de la scène et des informations disponibles, et dans un second temps, des paramètres initiaux choisis pour l'estimation d'une scène donnée.

2.4.2.1 Influence de l'initialisation 3D et du bruit 2D sur les positions dans les images sur le taux de succès

bruit 2D	Longueur focale	σ_{focale} initial	Bruit 3D	Nb. images	Nb. Points	Visibilité
[0..2]	500 px.	20 px.	[0..5]	5	20	0.8

TAB. 2.1: Conditions de test de l'influence du modèle 3D initial et du bruit 2D sur les positions dans les images sur le taux de succès.

Erreur 2D	Disparité du taux de succès CACHE_POS	Disparité du taux de succès Classique
0	-0.5%	+8%
1	+1%	+0%
2	-4%	-10%

TAB. 2.2: Influence du bruit 2D dans les images sur le taux de succès. Les valeurs représentent les écarts entre les taux de succès obtenus pour des bruits 2D donnés et les moyennes des taux de succès calculés pour un bruit 3D donné sous l'hypothèse d'une indépendance totale du taux de convergence par rapport au bruit 2D.

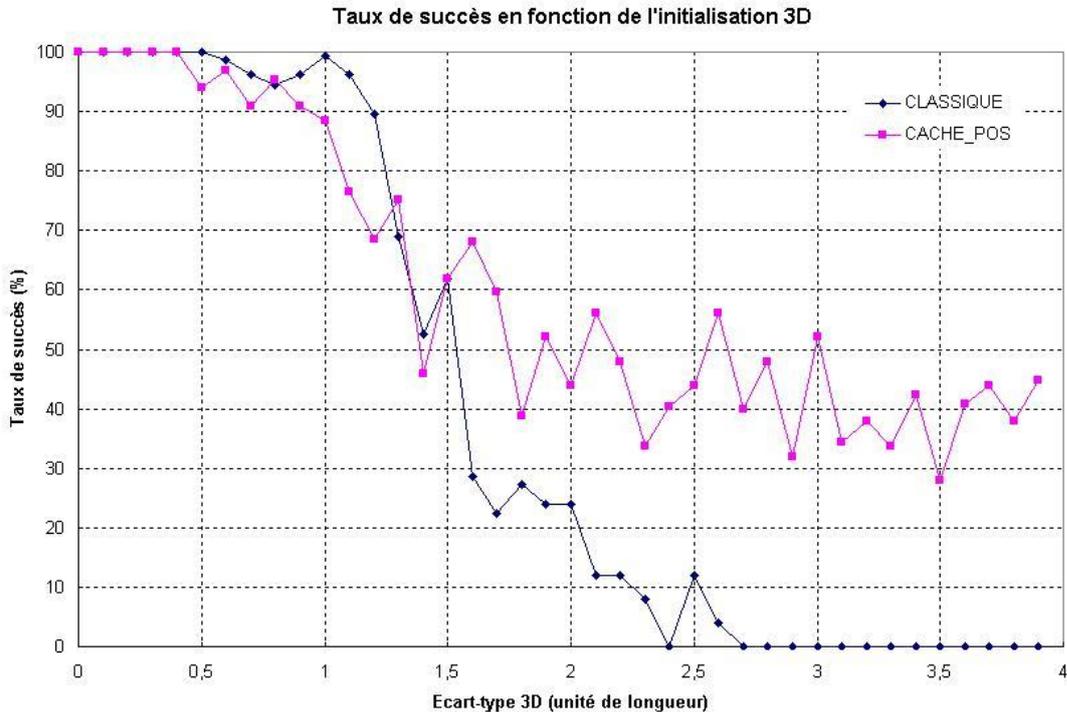


FIG. 2.12: Taux de convergence en fonction de l'erreur 3D initiale (le bruit 2D injecté au niveau des données 2D a un écart-type de 1 pixel).

Les résultats présentés dans le tableau 2.2 montrent l'influence du bruit 2D sur le taux de convergence. Pour évaluer cette influence, nous avons calculé, pour chacun des bruits 3D initiaux évalués, les taux moyens de succès pour chacune des deux méthodes (tous bruits 2D confondus). Si le taux de convergence est indépendant du bruit 2D, ces moyennes représentent le taux de convergence en fonction du bruit 3D initiale. Pour évaluer l'indépendance du taux de convergence de CACHE_POS et de la méthode classique par rapport au bruit 2D, nous avons calculé, pour chaque bruit 2D et chaque bruit 3D, la moyenne des écarts entre les taux de convergence supposés indépendants du bruit 2D et les taux de convergence observés pour chaque bruit 2D. En observant le tableau 2.2, on remarque que le taux de convergence de la méthode CACHE_POS est peu sensible au bruit ajouté à la localisation des primitives dans les images et est moins sensible à ce bruit que la méthode classique. Ceci montre que la qualité de détection des positions des points dans les images n'est pas le facteur prépondérant par rapport à la convergence.

La figure 2.12 montre l'évolution du taux de succès en fonction de l'erreur sur le modèle initial. Le bruit 2D est fixé à un écart type de 1 pixel. Le modèle 3D initial est obtenu en perturbant la position des points 3D parfaits positionnés à l'intérieur d'une sphère de rayon 1 par un bruit gaussien dont le rayon est indiqué en abscisse de la figure 2.12. Les deux courbes présentent trois zones principales :

1. Pour un bruit 3D entre 0 et 1, un plateau correspondant à des taux de succès élevés. La limite de 1 correspond au rayon de la sphère. C'est le moment où les points initiaux commencent à se mélanger et où l'organisation générale du modèle commence à être perturbée.
2. Pour un bruit 3D entre 1 et 2-2.5, on observe une décroissance régulière du taux de succès. Nous attribuons cette décroissance à une désorganisation de plus en plus prononcée de la structure 3D initiale. L'effet est d'autant plus violent pour la méthode classique que l'initialisation des positions des caméras est effectuée en exécutant une estimation de la pose à partir du nuage de points initial (cette situation initiale est la même pour l'algorithme CACHE_POS).
3. Pour un bruit supérieur à 2-2.5, on observe une stabilisation du comportement avec un taux de convergence nul pour la méthode classique et un taux de convergence constant pour la méthode CACHE_POS. Ce seuil de 2-2.5 correspond au diamètre de la sphère et traduit le moment à partir duquel les points initiaux sont complètement mélangés. Des essais supplémentaires effectués en tirant la position des points initiaux au moyen d'une loi uniforme dans un cube de côté 2 montrent le même taux de convergence.

Les expériences présentées ici montrent que le taux de succès dépend essentiellement de l'organisation initiale du modèle 3D. Cette notion d'organisation est mal définie et nous n'avons pas trouvé de critère pertinent permettant de dire, a priori, si un modèle 3D initial est favorable à la convergence ou non.

2.4.2.2 Influence du nombre de points sur le taux de succès

bruit 2D	Longueur focale	σ_{focale} initial	Bruit 3D	Nb. images	Nb. Points	Visibilité
2	500 px.	20 px.	3	5	[5..45]	0.8

TAB. 2.3: Conditions de test de l'influence du nombre de points sur le taux de succès.

Nombre de points	Taux de succès (%)
5	28
13	59
15	62
25	57
35	65
45	53

TAB. 2.4: Influence du nombre de points observés sur le taux de succès.

Le tableau 2.4.2.2 montre l'existence d'un nombre de points 3D critique au dessous duquel le taux de convergence chute considérablement. On constate qu'en présence de cinq points le taux de succès est seulement de 28 pour cent. Ce résultat traduit la grande instabilité l'estimation des poses dans ces conditions. Des essais complémentaires ont permis de montrer l'existence d'une valeur limite de treize points 3D au dessous de laquelle le taux de succès de CACHE_POS chute brutalement.

2.4.2.3 Influence du taux de visibilité sur le taux de convergence

bruit 2D	Longueur focale	σ_{focale} initial	Bruit 3D	Nb. images	Nb. Points	Visibilité
2	500 px.	20 px.	3	5	20	[40%..100%]

TAB. 2.5: Conditions de test de l'influence du taux de visibilité sur le taux de succès.

Taux de visibilité (%)	Taux de convergence (%)
40	21.3
60	31.3
75	46
100	56.7

TAB. 2.6: Taux de convergence en fonction du taux de visibilité.

Le taux de visibilité influence directement le nombre de mesures disponibles pour un nuage de points 3D donné. Les résultats présentés dans le tableau 2.6 montrent, pour une initialisation 3D très mauvaise, la relation très forte existant entre le taux de visibilité et le taux de succès. Ceci traduit l'importance de la redondance de l'information afin de parvenir à un positionnement cohérent des caméras les unes par rapport aux autres et, par conséquent, à une reconstruction globale cohérente.

2.4.2.4 Influence de la longueur focale initiale sur le taux de succès

bruit 2D	Longueur focale	σ_{focale} initial	Bruit 3D	Nb. images	Nb. Points	Visibilité
2	[100..5000] px.	0 px.	0.5	5	20	0.8

TAB. 2.7: Conditions de test de l'influence de la focale sur le taux de succès.

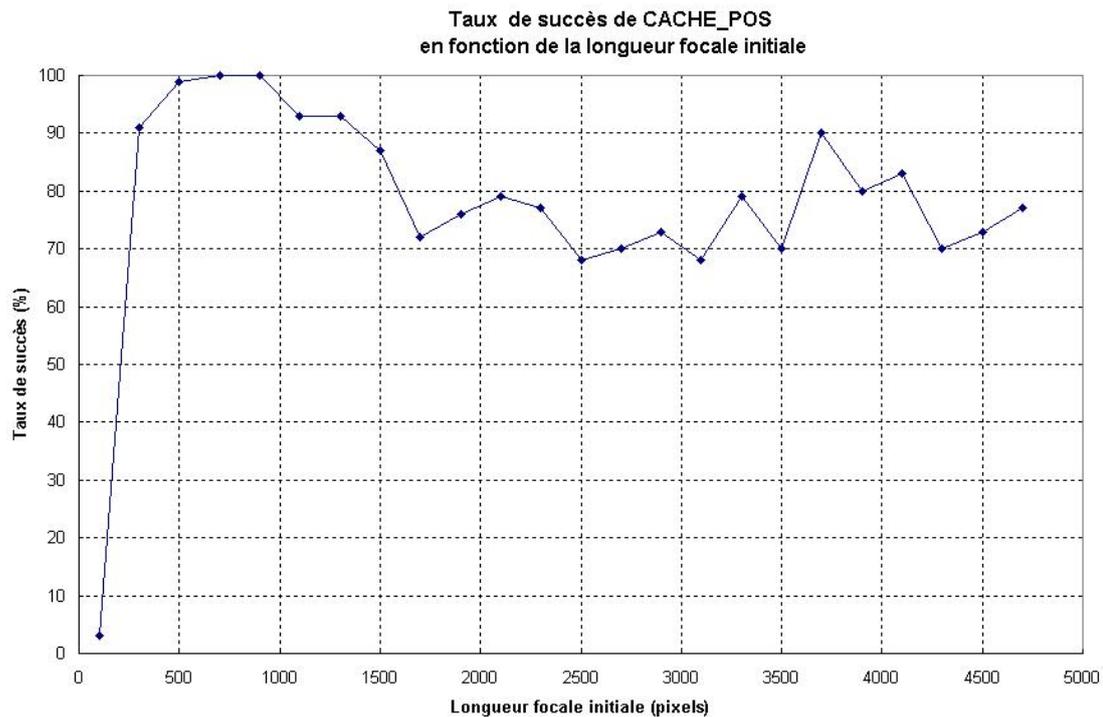


FIG. 2.13: Taux de convergence en fonction de la longueur focale initiale.

Pour finir, nous avons étudié l'influence de la valeur initiale de la longueur focale près de la solution (500 pixels). On observe que le taux de succès est de cent pour cent. Dans le cas d'une initialisation avec une valeur faible, le taux de convergence chute brutalement (une explication de ce phénomène est donné dans le paragraphe suivant). Pour les valeurs élevées, le taux de succès est de 80 pour cent. Nous avons également initialisé la valeur de la longueur focale à 30000 pixels, le taux de convergence obtenu est de 80 pour cent. Ces expériences montrent que la valeur initiale de la longueur focale influence relativement peu le taux de convergence lorsque l'initialisation n'est pas trop faible. Cette observation permet d'envisager l'initialisation arbitraire (à une valeur standard) des longueurs focales.

2.4.2.5 Bilan sur l'analyse du taux de succès

bruit 2D	Longueur focale	σ_{focale} initial	Bruit 3D	Nb. images	Nb. Points	Visibilité
2	500 px.	20 px.	0	5	20	0.8

TAB. 2.8: Conditions de test de l'influence de l'initialisation du modèle 3D sur le taux de succès.

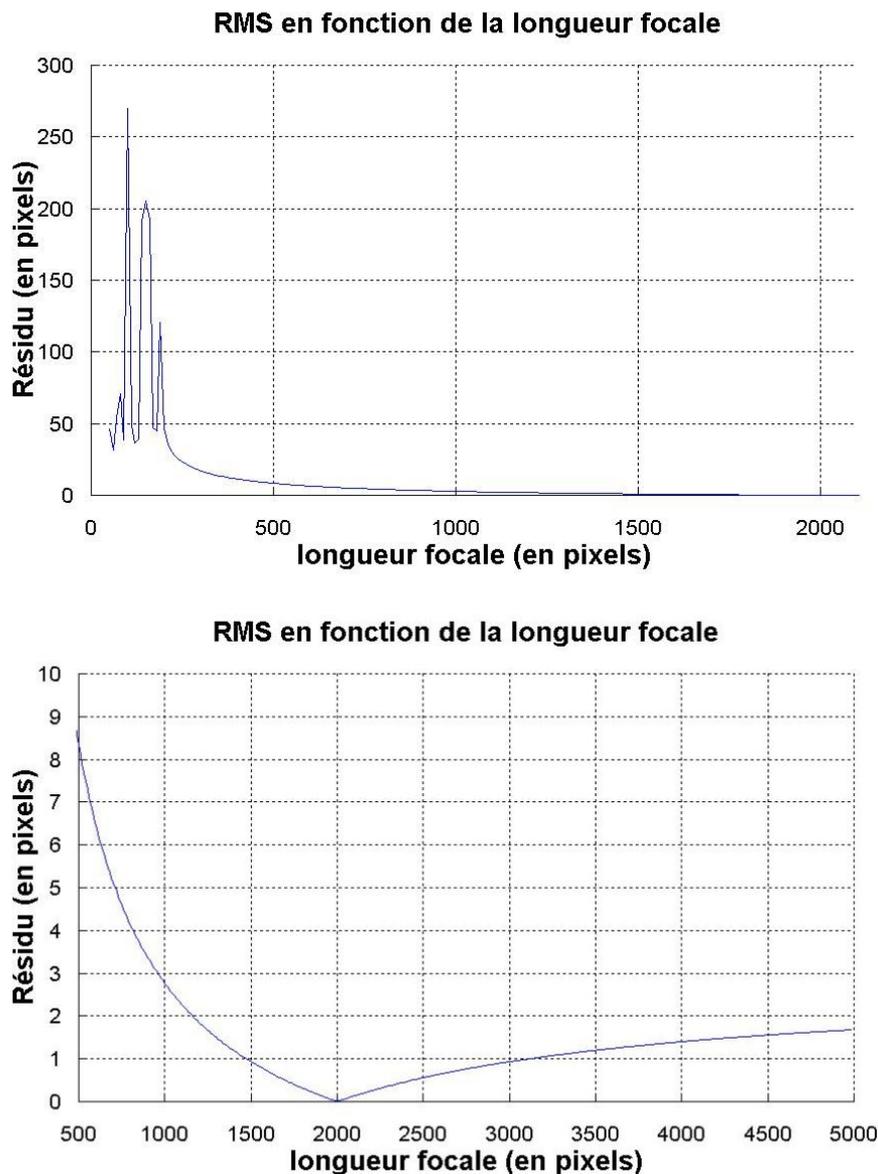


FIG. 2.14: Critère de CACHE_POS en fonction des valeurs de la longueur focale. **Les deux graphes représentent le même tracé**, ils ont été divisés afin de permettre d'observer clairement l'évolution du critère en fonction de la longueur focale.

Les quelques expériences que nous venons de présenter dressent un premier aperçu des propriétés de CACHE_POS. Il est évident que l'initialisation du modèle 3D et le nombre de mesures disponibles sont le coeur des causes de non-convergence à une solution acceptable. Afin de comprendre plus finement le rôle de l'initialisation 3D, nous avons effectué une expérience consistant à voir, pour une scène 3D donnée, le taux de succès de CACHE_POS en fonction des initialisations sur un nuage de points. Pour une scène donnée (conditions de test, cf. tableau 2.8), nous avons généré aléatoirement des situations initiales. La position initiale de chacun des points 3D a été définie aléatoirement par un tirage uniforme dans un cube de quatre unités de distance de coté (les points 3D correspondant effectivement au modèle ont été définis à l'intérieur d'une sphère de rayon 1). Les résultats montrent que le taux de succès est d'environ quarante pour cent. Ce taux est le même que celui trouvé précédemment. Nous en concluons que ce n'est pas telle ou telle scène 3D qui est favorable à la convergence, mais plutôt la disposition initiale des points du nuage qui mène soit à la solution soit à un minimum local.

L'observation au cas par cas des solutions rejetées nous a permis de constater que les cas non retenus étaient parfois caractérisés par une distance focale très grande ou très courte. Afin de comprendre ce phénomène nous avons tracé la valeur du critère en fonction de la longueur focale. Ces courbes présentées en figure 2.14 présentent une forme générale proche de celle observée sur la courbe d'influence de la longueur focale sur l'estimation de l'orientation du capteur par l'algorithme d'estimation de pose de Davis et Dementhon (cf. 2.9). On constate toutefois une différence au niveau des petites valeurs de focale. Les deux cas d'échec détectés par des longueurs focales finales incompatibles avec le problème posé deviennent interprétables. En fonction de la trajectoire suivie au cours de l'optimisation, deux cas d'échec se présentent. Soient les longueurs focales deviennent trop courtes, soient elles deviennent trop longues :

- **longueurs focales trop courtes** : l'estimation de la pose des images devient absolument instable car l'hypothèse para-perspective est brisée. En observant la partie gauche de la courbe 2.9, on voit que la prise en compte de plusieurs images génère un ensemble de minima locaux provoquant l'échec,
- **longueurs focales trop longues** : la pente devient extrêmement faible du fait du comportement asymptotique du critère ce qui provoque l'arrêt de l'estimation.

Du point de vue du nombre de convergences réussies, il apparaît que les trois principaux facteurs sont la situation initiale du modèle 3D, le taux de visibilité des points et le nombre de points observés dans la scène. La différence majeure existant entre la méthode CACHE_POS et la méthode classique est la capacité de la première méthode à converger parfois même lorsque le modèle initial est très erroné. Dans les différents essais que nous avons effectué, la méthode CACHE_POS présente donc un domaine de convergence plus large que la méthode classique.

2.4.3 La qualité de la reconstruction

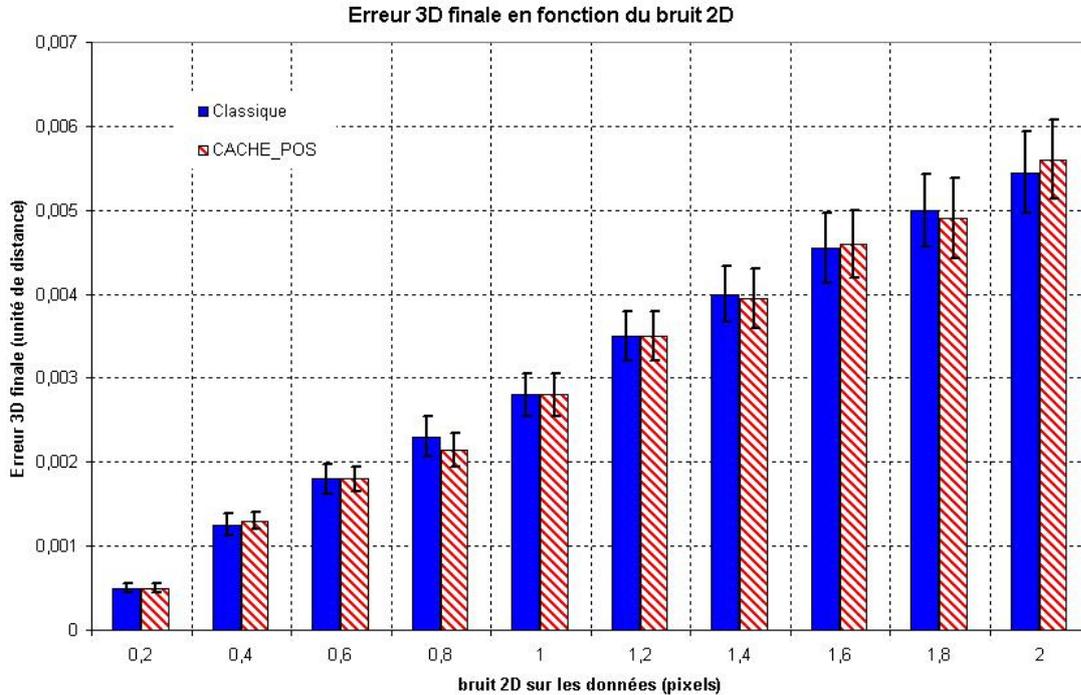


FIG. 2.15: Erreurs 3D finales en fonction du bruit 2D.

Afin d'évaluer la qualité de la reconstruction, nous avons analysé le résidu 2D moyen dans les images et l'erreur 3D finale par rapport au modèle 3D parfait.

La figure 2.15 montre que l'erreur 3D sur le modèle est linéaire par rapport au bruit 2D présent dans les images. L'erreur 3D a été définie en déterminant la distance entre les sommets du modèle parfait et ceux du modèle reconstruit suite à un recalage (en anglais "best fit") des nuages de points. Ce recalage est nécessaire car la méthode CACHE_POS ne fixe pas le repère lors de la reconstruction (cf. chapitre 2). On observe que la qualité des reconstructions obtenues par la méthode classique et par la méthode CACHE_POS est la même. Il n'apparaît pas de différence évidente entre les performances de la méthode classique et les performances de CACHE_POS. En fait, la seule différence que nous avons observée n'est pas représentée sur la figure car elle concerne la situation où le bruit injecté dans les données images est nul. Dans ce cas, la méthode classique converge et atteint un RMS final de l'ordre de 10^{-13} pixels (cette valeur dépend du critère d'arrêt choisi) alors que CACHE_POS ne parvient pas à descendre en dessous de 10^{-7} pixels en utilisant la méthode de Lowe pour estimer les poses (la méthode de Dementhon est moins précise).

La figure 2.16 montre le résidu final (RMS) obtenu en fin de reconstruction par la méthode CACHE_POS en fonction de l'influence de l'initialisation du modèle 3D et des erreurs présentes sur la détection des points dans les images. Chaque point représente le résultat moyen calculé sur 50 essais différents (scènes générées aléatoirement). A première vue, la structure en lignes horizontales de ces courbes indique l'indépendance existant

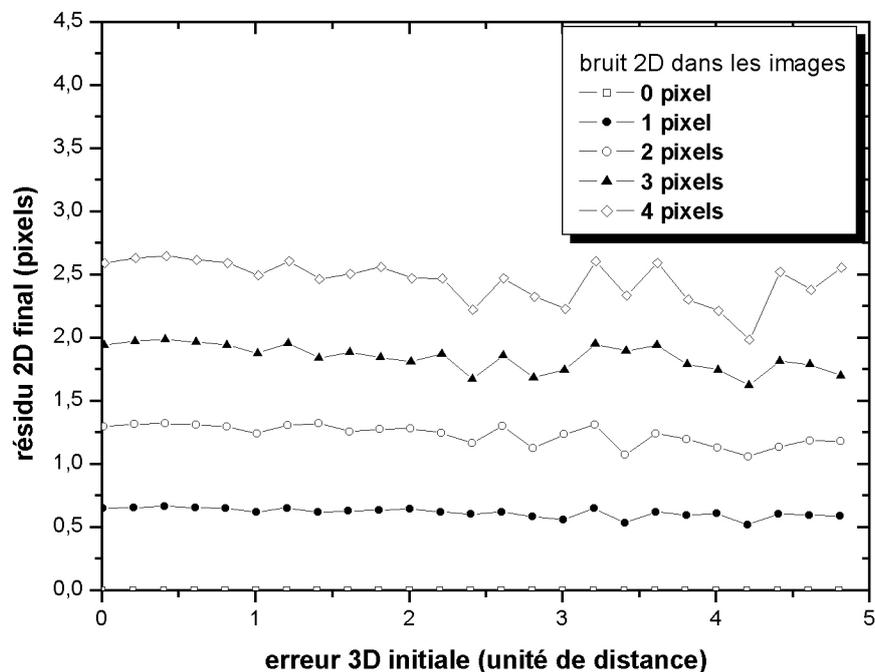


FIG. 2.16: Erreur 2D à la convergence en fonction du bruit 2D affectant la détection des primitives dans les images et de l'erreur 3D à l'initialisation. Chaque courbe correspond à un bruit 2D spécifique.

entre le résidu 2D final et l'initialisation 3D initiale. Pourtant, en observant plus finement la figure 2.16, on constate que plus l'initialisation 3D est erronée plus le tracé du RMS est perturbé. Ce phénomène est amplifié par le bruit 2D. Pour chacun des bruits 2D, nous avons analysé la disparité des données par rapport au RMS moyen pour un bruit 2D donné. Pour un ensemble de n mesures m_i , nous définissons la disparité δ_i du point i par :

$$\delta_i = m_i - \frac{\sum_{j=1}^n m_j}{n} \quad (2.22)$$

A partir ces résultats de la figure 2.16, nous avons, pour chaque bruit 2D, représenté l'histogramme des disparités par rapport au bruit dans les images. Sur ces histogrammes, présentés sur la figure 2.17, on observe que plus le bruit 2D augmente plus la distribution du nombre d'essais de part et d'autre de $\delta_i = 0$ est dissymétrique. Nous n'avons pas d'explications à ce phénomène mais ce peut être, par exemple, un indice de l'existence de minima locaux ou la traduction d'erreur de calcul lié à la liberté de jauge associée à CACHE_POS (plus le modèle initial est faux plus le repère peut dériver par rapport au repère du modèle parfait).

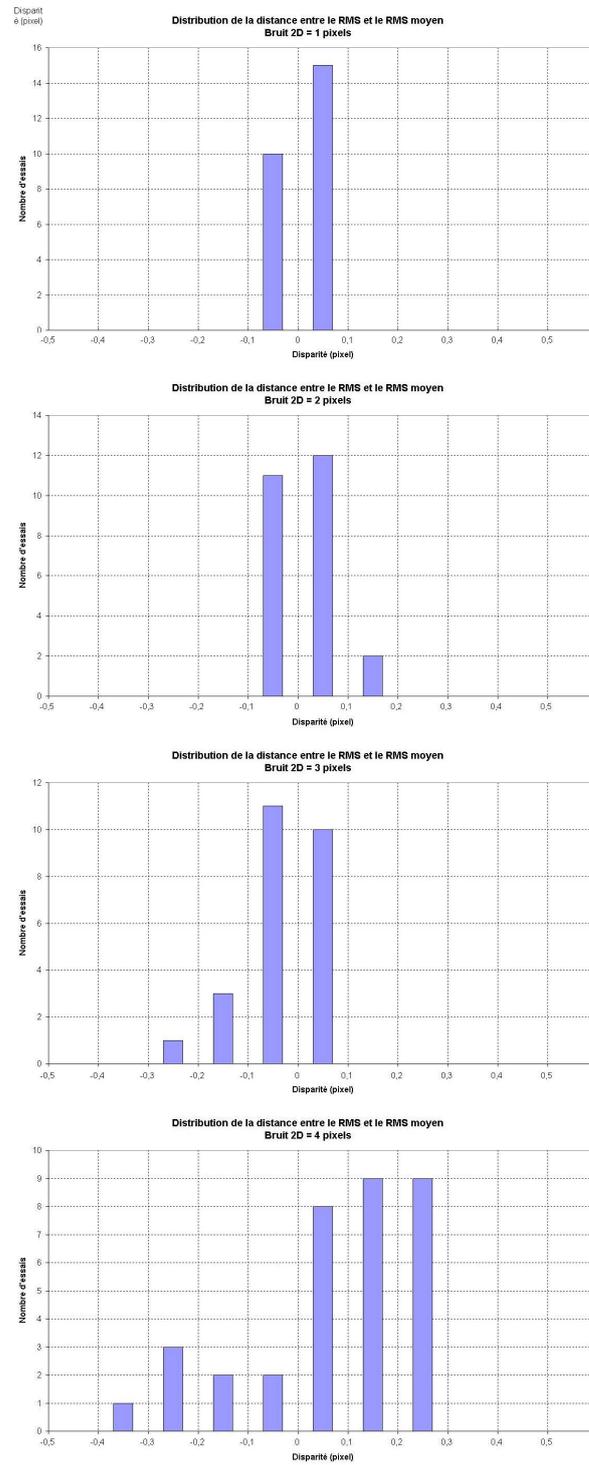


FIG. 2.17: Distribution des écarts entre la mesure moyenne et chacune des mesures.

2.4.4 Temps de calcul

bruit 2D	Longueur focale	σ_{focale} initial	Bruit 3D	Nb. images	Nb. Points	Visibilité
[1..4]	500 px.	20 px.	[0..5]	5	20	0.8

TAB. 2.9: Conditions de test du temps de calcul en fonction du bruit 2D et de l'erreur 3D initiale.

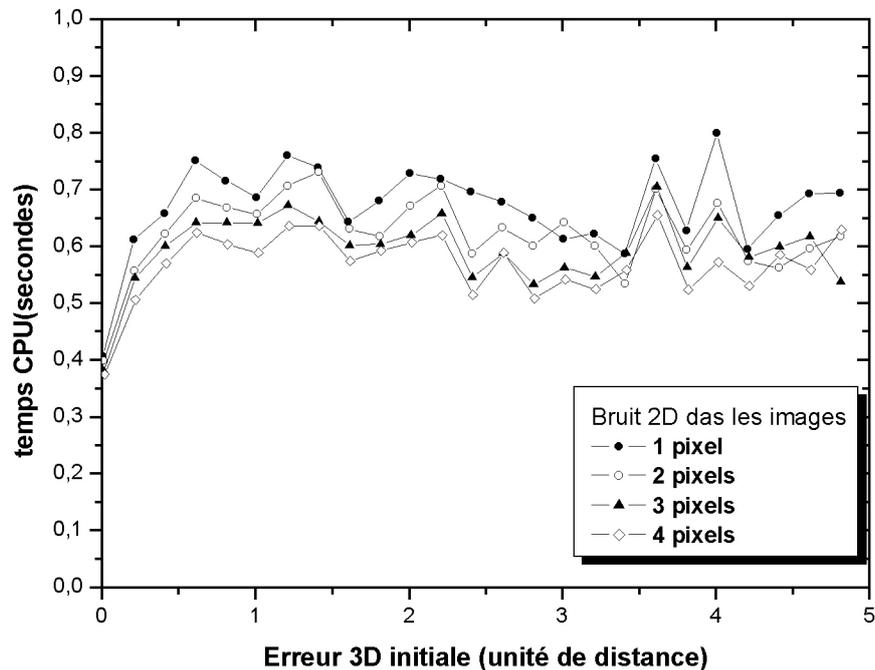


FIG. 2.18: Temps de calcul en fonction du bruit 2D et de l'erreur 3D initiale.

Nous avons évalué les performances en temps de calcul par rapport au modèle 3D initial, au bruit 2D lié à la détection des primitives dans les images et au nombre d'images. La figure 2.18, dont les conditions d'expérimentation sont décrites dans le tableau 2.9, montre la faible corrélation existant entre la durée de la reconstruction et le bruit 2D. Par contre, on observe que la forme initiale du modèle 3D conditionne le temps de reconstruction. Si le modèle est proche de la solution, l'optimisation s'effectue dans un temps plus court qu'en cas d'initialisation lointaine. On constate toutefois qu'au delà d'une erreur 3D initiale d'une unité, il n'y a plus d'allongement de la durée de reconstruction.

Dans le tableau 2.10 sont présentés les temps de calcul en fonction du nombre d'images. Les performances doivent être analysées en terme d'évolution plutôt qu'en différence absolue entre la méthode classique et CACHE_POS. En effet, la méthode classique utilisée ici n'exploitait pas les optimisations rendues possibles par la structure creuse de sa matrice quasi-hessienne. On observe que le temps de calcul de la méthode classique

s'accroît considérablement en fonction du nombre d'images, ceci s'explique par le fait que chaque image supplémentaire apporte six paramètres de pose supplémentaires à estimer. La méthode CACHE_POS ne conduit aucunement à un accroissement du nombre de paramètres lorsque le nombre d'images augmente. L'accroissement du temps de calcul est principalement lié à l'accroissement du nombre total de mesures. Évidemment, cet avantage observé en fonction du nombre d'images n'existe pas lorsque l'on augmente le nombre de points 3D.

Nombre de caméras	Temps en seconde	
	Classique	CACHE_POS
5	2.0	0.6
20	21	1.8
35	140	3.6

TAB. 2.10: Temps de calcul en fonction du nombre de caméras.

2.5 Bilan

Nous avons vu, à travers ces différents essais, les grandes lignes du comportement de CACHE_POS. Tout d'abord, en terme de convergence, il est apparu que CACHE_POS présente un taux de convergence bien supérieur à celui de la méthode classique lorsque le modèle 3D initial est très éloigné de la solution. De plus, CACHE_POS présente une grande souplesse quant à l'initialisation de la longueur focale initiale offrant ainsi la possibilité d'initialiser le processus d'optimisation avec une valeur standard. Nous avons montré par ailleurs que CACHE_POS parvient à une solution d'une qualité équivalente à la méthode classique (dans la gamme de précision étudiée ici). Enfin, cette méthode est non seulement adaptée à toutes les approches tirant avantage de l'absence d'initialisation des paramètres de pose et offre en plus, une accélération conséquente des processus de reconstruction utilisant un grand nombre d'images.

Ces tests réalisés sur des séquences simples (focale unique...) nous ont permis de mettre en évidence les propriétés de CACHE_POS. Afin de nous approcher de conditions plus réalistes, nous avons généré une séquence de cinq images prises chacune avec une longueur focale différente. Ces longueurs focales dont les valeurs véritables s'étalent de 500 à 5000 pixels ont toutes été initialisées à 1000 pixels. Les résultats obtenus sont présentés dans le tableau 2.12. On observe une chute du taux de convergence qui est seulement de 22 pourcent lorsque le modèle 3D initial est déterminé avec un écart type de quatre unités de distance (modèle très erroné). On retrouve par contre une asymptote de 4.2 secondes pour la durée de la reconstruction et un RMS de l'ordre de 1.8 pixels. L'estimation de scène photographiée en utilisant plusieurs longueurs focales différentes est possible même si le taux de convergence est plus faible, ce qui implique une plus grande sensibilité et donc une plus grande attention nécessaire à l'initialisation.

Erreur 3D initiale	RMS final 2D(pixels)	Taux de convergence(/100)	Temps de calcul (sec.)
0	1.77	100	1.74
2	1.79	40	4.29
4	1.80	22	4.72

TAB. 2.11: Résultats obtenus sur une séquence présentant de fortes variations de longueur focale entre les différentes prises de vue.

2.6 Applications sur des données réelles

Afin d'expérimenter l'algorithme CACHE_POS sur des données réelles, nous avons pris six images d'un bureau. Ces images ont été obtenues avec un appareil photographique équipé d'un objectif fixe. Par conséquent les paramètres intrinsèques sont identiques pour toutes les images. Ces images sont présentées figure 2.19. Elles ont une taille de 1204x801 pixels. Lors de l'estimation de la scène, nous avons supposé le point principal positionné au centre des images, il n'a pas été estimé. La longueur focale a été estimée au moyen de deux paramètres f_x et f_y permettant de prendre en compte les dimensions des pixels dans le sens de la largeur et de la hauteur.

Les points de vue sont convergents et tous pris du même côté du bureau. Vingt deux points ont été définis et localisés manuellement dans les images. Afin de comparer le modèle obtenu suite à la reconstruction avec la réalité, nous avons mesurées au moyen d'un réglet des longueurs spécifiques entre des points 3D facilement identifiables et nous avons positionné une référence métrique dans la scène. Les distances mesurées entre certains points 3D ainsi que la référence métrique (1 meter) sont indiquées dans la figure 2.20. Dans chaque image, il y a en moyenne soixante trois pour cent des points qui sont observés. Les paramètres caractérisant les longueurs focales (f_x et f_y) sont tous les deux initialisés à 2000 pixels. Le taux de visibilité sur cette séquence est de 63 %.

Avant la reconstruction, nous ne disposons pas de la position des points 3D. L'initialisation a été effectuée en générant aléatoirement leur position initiale au moyen d'un tirage suivant un loi uniforme dans un cube de coté 1. La figure 2.21 présente la projection des segments 3D dans une image juste après une initialisation de la pose des caméras au moyen de l'algorithme de Davis et Dementhon. On observe qu'ils sont très éloignés de la solution finale.

La figure 2.22 montre la projection des segments 3D dans la même image après une reconstruction réussie. Le taux de succès en fonction de l'initialisation des sommets 3D est de quarante quatre pour cent. Ce taux de succès est cohérent avec les résultats obtenus dans le tableau 2.6.

Le résidu moyen obtenu en fin d'estimation est 0.808 pixels et les paramètres caractérisant la longueur focale valent $f_x=1844,5$ pixels et $f_y=1847,2$ pixels. Afin de quantifier la qualité de la reconstruction, nous avons évalué les distances de référence en utilisant la métrique disponible. Les résultats sont présentés dans le tableau 2.12.

Numéro	Longueur réelle (cm) Incertitude = $\pm 0,1$ cm	Longueur estimée (cm)	Erreur relative (/100)
1	20.4	20.43	0.14
2	23.5	23.34	0.68
3	23.5	23.40	0.42
4	20.4	20.67	1.32
5	14.8	14.74	4.05
6	14.4	14.42	1.38
7	20.6	20.59	0.48
8	23.7	23.57	0.54
9	20.6	20.47	0.63
10	23.6	23.55	0.21
11	120.0	119.79	0.18
Moyenne	-	-	0.64

TAB. 2.12: Distances estimées et erreurs.

Les résultats sont cohérents avec l'incertitude des valeurs mesurées ce qui montre que la scène est correctement reconstruite. Cette reconstruction a été obtenue sans fixer de repère. Afin d'estimer les longueurs, le facteur d'échelle a été fixé a posteriori en utilisant la métrique observée, ce qui peut générer un biais.



FIG. 2.19: Les six images disponibles.



FIG. 2.20: Les longueurs mesurées et la distance de référence.

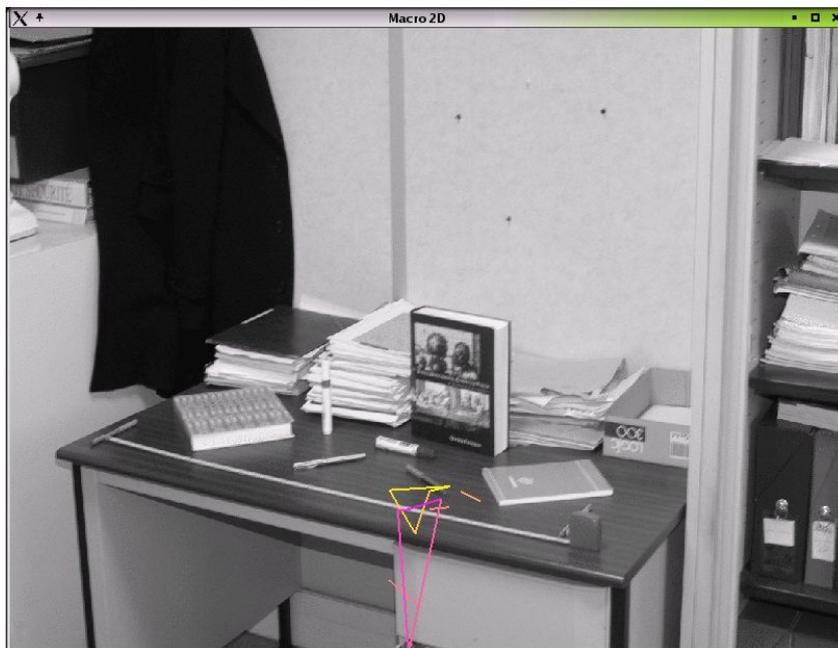


FIG. 2.21: Projections initiales des segments dans une image.

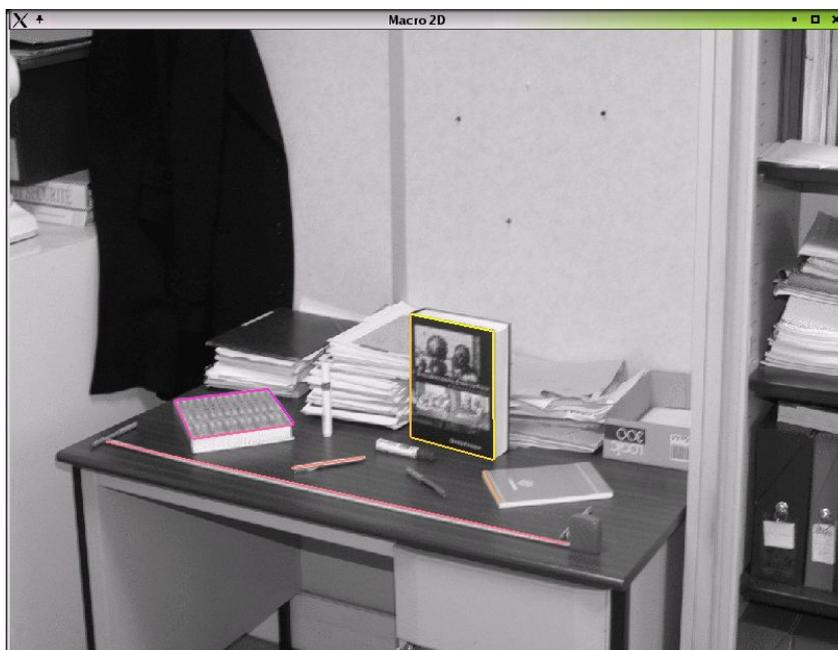


FIG. 2.22: Projection finale des segments après convergence.

2.7 Conclusion

Au cours de ce chapitre, nous avons présenté une technique d'ajustement de faisceaux nommée CACHE_POS consistant à estimer, par une méthode de type Levenberg-Marquardt, l'ensemble des paramètres caractérisant les capteurs et le modèle 3D. Les paramètres d'attitude des images sont traités spécifiquement et sont calculés en interne au moyen d'un algorithme d'estimation de pose. L'algorithme CACHE_POS possède plusieurs propriétés remarquables :

- **choix du repère** : CACHE_POS offre une solution originale à la question du choix de jauge en définissant implicitement un repère de reconstruction à chaque itération,
- **structure creuse** : Nous avons montré que la matrice hessienne possédait une structure creuse au niveau des blocs associés aux paramètres caméra. Cette structure permet d'envisager des implémentations creuses de cette technique permettant de traiter des problèmes d'ajustement de faisceaux basés sur un très grand nombre d'images prises au moyen de différentes caméras. Nous notons toutefois que cette méthode n'offre pas une structure creuse vis-à-vis des paramètres associés au modèle 3D. Par conséquent, elle n'est pas compétitive par rapport à la méthode classique pour traiter des problèmes où le nombre de paramètres 3D serait très élevé,
- **propriété de convergence** : Après les nombreux tests que nous avons effectués, il apparaît que cette méthode offre un domaine de convergence beaucoup plus large que la méthode classique. Nous avons montré que le taux de convergence dépendait essentiellement de la position initiale du modèle 3D et du taux de visibilité des points dans les images. Le bruit 2D et l'initialisation de la longueur focale ne jouent pas un rôle prépondérant,
- **qualité de la reconstruction** : Dans le cadre d'applications de reconstruction, la qualité des modèles 3D obtenue par CACHE_POS est équivalente à celle obtenue par la méthode classique. Toutefois, dans des applications requérant une plus grande précision (calibrage), l'algorithme de Dementhon et Davis, employé seul, devient une limite à la précision accessible. Suite à divers essais sur des données de synthèse non détaillées ici, il apparaît difficile d'effectuer une convergence en dessous de quelques dixièmes de pixel en utilisant uniquement l'algorithme de Davis et Dementhon. L'emploi en complément de l'algorithme de Lowe permet de réduire ce seuil ce qui permet d'envisager des reconstructions de précision,
- **temps de calcul** : La question du gain en temps de calcul dépend du problème posé. La première certitude est la réduction du nombre de paramètres et de la taille de la matrice jacobienne provoquée par l'usage de CACHE_POS. Toutefois, l'absence d'une structure creuse pour les paramètres 3D implique, pour les problèmes, où un très grand nombre de primitives 3D sont impliquées, une

efficacité bien plus grande pour la méthode classique que pour cette approche. Cette remarque est renforcée si l'on prend en compte l'accroissement considérable du temps d'exécution de l'algorithme d'estimation de pose lorsqu'un grand nombre de sommets est observé. Au contraire, lorsque le nombre de paramètres 3D est du même ordre ou inférieur au nombre de paramètres associés aux vues (attitude et paramètres intrinsèques) la méthode CACHE.POS devient plus performante car elle exploite pleinement la suppression de l'estimation des paramètres de pose. Cette dernière situation est celle que nous allons rencontrer dans le cadre de la reconstruction d'objets structurés.

Deuxième partie

La reconstruction d'objets
structurés

Chapitre 3

Etat de l'art sur la reconstruction d'objets structurés

Sommaire

3.1	Introduction	95
3.2	Partir d'une reconstruction 3D de primitives élémentaires	97
3.3	Les approches par modèles complexes (type CAO)	99
3.3.1	Une approche par blocs paramétriques	100
3.3.2	Utiliser les parallélépipèdes	100
3.3.3	Une approche probabiliste	100
3.4	La modélisation par contraintes : une solution mixte	101
3.4.1	La reconstruction sous contraintes comme un problème linéaire	102
3.4.2	Description et simplification des contraintes par graphes	104
3.5	Conclusion	104

3.1 Introduction

Lors de la partie précédente, nous avons abordé le problème de la reconstruction d'un nuage de points libres de toutes contraintes. Ce cas canonique nous a permis d'étudier les propriétés de l'algorithme d'ajustement de faisceaux que nous avons proposé. L'objectif de cette thèse est de définir un processus permettant d'obtenir le modèle d'un objet réel à partir d'images non-calibrées. Bien sur, nous pourrions appliquer directement les résultats précédents et décrire cet objet comme un ensemble de sommets remarquables constituant un nuage de points libres. Seulement, le nombre très élevé de degrés de liberté existant dans un tel modèle impliquerait de saisir une grande quantité d'information au niveau des images (localisation des sommets). Cette nécessité nuirait grandement à la convivialité et à la possibilité d'utiliser en pratique cette méthode.

Une solution possible est d'apporter de nouvelles informations en tirant partie des propriétés de l'objet observé. Les bâtiments, un stade de football, une piste de sprint, un centre ville, mon bureau, une boîte de sucre, un avion, possèdent des propriétés

fréquemment rencontrées dans les créations humaines. En effet, les éléments constituant notre environnement sont structurés au moyen d'axes de symétrie, de parallélismes, d'orthogonalités et d'autres règles encore constituant les fondements de la forme d'un grand nombre d'objets. Avant de présenter la méthode de reconstruction que nous proposons dans le chapitre suivant, nous allons dresser un état de l'art (nécessairement non exhaustif tant ce domaine est riche) des méthodes permettant de reconstruire des modèles 3D d'objets non déformables structurés en utilisant des images.

Si l'on considère un objet, comment faire pour obtenir un modèle tridimensionnel de cet objet à partir d'une séquence d'images non-calibrées ? L'objectif final est d'obtenir un modèle 3D le plus proche possible de la réalité. Pour parvenir à ce résultat, il est nécessaire de choisir un formalisme de représentation du modèle compatible avec son traitement informatique et les algorithmes de reconstruction employés. Une modélisation possible, utilisée dans la première partie de ce document, consiste à représenter l'objet comme un ensemble de points 3D décrits par leurs coordonnées exprimées dans un repère euclidien. A partir d'une telle représentation, comment faire pour imposer à ce nuage de points des contraintes tout en assurant la capacité à estimer la scène à partir des images ? Trois choix sont possibles :

1. définir le modèle comme un nuage de points libres et ajouter aux mesures fournies par la localisation des points dans les images, des mesures traduisant le respect des contraintes. Ces contraintes peuvent être qualifiées de contraintes souples car elles ne sont pas nécessairement vérifiées par le modèle final,
2. reconstruire un modèle constitué de primitives élémentaires (points, segments,...) puis, dans un second temps, rechercher le modèle, vérifiant les contraintes, le plus proche du nuage de primitives élémentaires,
3. définir le modèle 3D de sorte que les contraintes soient implicitement vérifiées.

La première et la deuxième approches présentent le défaut de ne pas tirer pleinement partie des contraintes pour réduire la dimension du problème de reconstruction. La troisième approche impose quant à elle de disposer d'une phase de modélisation susceptible de fournir un modèle contraint exploitable. Nous verrons par la suite que ces trois approches sont utilisées, la première plutôt dans les méthodes automatiques et les deux autres dans les méthodes supervisées.

Une question pratique se pose concernant la technique d'acquisition des images, faut-il travailler avec une (des) séquence(s) vidéo ou des images ? La séquence vidéo présente deux avantages. La grande ressemblance entre deux images successives de la séquence permet de mettre en oeuvre efficacement des algorithmes de suivi de primitives et, par conséquent, d'acquérir automatiquement un grand nombre d'informations. Le second avantage de la vidéo est la capacité à couvrir des scènes vastes en filmant le modèle par petite portion offrant ainsi une grande résolution sur l'ensemble de la scène au prix de séquences longues. Malgré ces avantages, l'utilisation de vidéos implique encore de manipuler un grand volume de données ce qui impose de sélectionner des images clefs afin de réduire le nombre d'images à traiter.

De plus, la photographie numérique présente l'avantage d'offrir une meilleure qualité d'images (la vidéo numérique offre toutefois des performances de plus en plus intéressantes

et des applications ([Pollefeys 00], [Lhuillier 03]) permettent déjà d'effectuer des reconstructions par ce moyen) et de ne conserver que les points de vue pertinents (en fonction de l'expertise du photographe) dans le cadre de la reconstruction, réduisant ainsi le volume des données à exploiter. La contre-partie de ce faible nombre d'images est, d'une part, l'absence de lien évident entre deux images successives de la séquence rendant difficile l'utilisation d'algorithmes d'appariement automatique et, d'autre part, la nécessité de sélectionner a priori les points de vue afin de rendre la reconstruction possible.

Dans le cadre de cette thèse, nous avons uniquement exploité des images fixes. Afin d'organiser la présentation des méthodes de reconstruction déjà existantes nous avons choisi de structurer ce chapitre en trois parties. Dans un premier temps, nous présenterons des approches basées sur la reconstruction initiale d'un nuage 3D de primitives élémentaires (points, segments...). Dans un deuxième temps, nous présenterons les approches s'appuyant sur des descriptions du modèle au moyen de blocs 3D complexes (modèles CAO). Enfin, dans une troisième partie, nous présenterons des méthodes intermédiaires utilisant des primitives élémentaires associées à des contraintes afin de réduire le nombre de degrés de liberté associés au modèle tout en évitant d'être dépendant d'une bibliothèque de formes 3D.

3.2 Partir d'une reconstruction 3D de primitives élémentaires

Les approches présentées ici utilisent une reconstruction de primitives de bas niveau (points, droites) avant d'introduire des modèles plus complexes au moyen de processus automatiques. Ce domaine a connu un essor certain ces dernières années et quelques techniques permettent désormais de reconstruire automatiquement des modèles à partir de plusieurs vues non calibrées. La grande force de ces approches est de pouvoir proposer des processus de reconstruction totalement automatiques. Ces applications sont particulièrement adaptées à des rendus réalistes mais dépendent toutefois fortement de la nature des informations disponibles dans les images et, particulièrement, de la richesse des textures et de la qualité des contours présents.

La première étape de ces méthodes consiste à reconstruire un nuage de primitives (souvent des points, parfois des segments, ...) à partir d'appariements automatiques. Cette phase d'appariement est cruciale pour le succès de la méthode car les appariements obtenus sont utilisés afin de déterminer les propriétés des prises de vues et la structure initiale du modèle 3D. Cette étape a été très largement étudiée et de nombreux algorithmes ont été proposés ces dernières années ([Pollefeys 99], [Mahamud 00]).

La tâche suivante consiste à donner du sens au modèle en y identifiant des éléments structurants. La première étape peut être, par exemple, l'extraction des plans principaux de la scène ou bien le placement par un utilisateur de formes complexes (cylindres, objets métiers...) dans ce nuage 3D (cf. [Bartoli 03]).

Cette problématique se rencontre également dans le cadre des relevés par télémétrie laser. En effet, les techniques laser permettent d'obtenir des nuages extrêmement denses (plusieurs milliers et même plusieurs millions de points 3D) de la surface des objets de la scène. Pour rendre cette information utilisable, il faut organiser ces informations en

apportant du sens au nuage de points. Ce sens est fourni sous la forme de modèle 3D comme des cylindres, des poutres ou toute autre forme que l'on positionne dans le nuage 3D de sorte à identifier les positions et les dimensions de ces blocs paramétriques par rapport au nuage de points. Dans le cadre des applications de vision, les informations disponibles sont non seulement le nuage 3D mais aussi les images, référencées par rapport au modèle, pouvant fournir des informations supplémentaires comme la texture ou la présence de contours.

Afin d'illustrer cette approche, nous détaillons les travaux effectués par Werner et Zisserman dans [Werner 02b] et [Werner 02a]. Leur approche est constituée de trois étapes principales. La première étape consiste à reconstruire automatiquement un nuage de points 3D à partir d'un ensemble d'appariements de points d'intérêt extrait dans les images. Afin de supprimer les faux appariements une sélection robuste de type RANSAC est exécutée en utilisant comme critère le respect du tenseur trifocal associé à chaque triplet d'images. Une fois les appariements triés, une reconstruction projective de la scène et des capteurs est effectuée.

La deuxième étape consiste à structurer ce nuage de points. L'objectif étant de reconstruire des scènes architecturales, les auteurs segmentent ce nuage de points en plans. Deux éléments supplémentaires sont introduits afin de faciliter cette opération :

- **une extraction de points de fuite** : Les scènes architecturales présentent souvent trois directions privilégiés (une verticale et deux horizontales). La détection des points de fuite dans les images permet de définir ces trois directions et de passer d'une reconstruction projective à une reconstruction euclidienne (ces trois points de fuite forment une base orthogonale de l'espace),
- **des appariements de segments** : Les segments extraits dans les différentes images sont appariés.

Dès lors, le problème consiste à déterminer un modèle polyédrique à partir de la connaissance des caméras, du nuage de points 3D issu des appariements automatiques, des appariements de lignes et des directions principales de la scène issues des points de fuite. Afin de tirer partie des directions extraites de la détection des points de fuites, la première étape consiste à rechercher les plans dont les vecteurs directeurs sont deux des directions principales. Cette recherche s'effectue au moyen d'un processus de type RANSAC mais, au lieu de sélectionner trois points quelconques du nuage de points, l'algorithme proposé sélectionne un seul point et deux directions principales. Cette stratégie réduit le nombre de candidats possibles et permet d'accélérer le processus d'extraction tout en améliorant les performances. A partir de cette première identification des plans principaux, les autres plans caractéristiques sont extraits. Ce problème consistant à rechercher des formes 3D particulières dans un nuage de points 3D se rencontre également dans le cas des reconstructions obtenues au moyen de scanner laser. En effet, les nuages de points obtenus en fin d'acquisition sont très denses et le recours à des techniques permettant de localiser des formes 3D dans ces données est essentielle pour rendre utilisable les informations recueillies (cf. [Moron 96] et [Ertl 01]).

La suite de la méthode concerne l'intégration d'éléments de détail dans le modèle 3D. L'idée est de raffiner chacun des plans identifiés lors de l'étape précédente en intégrant des formes 3D génériques à partir d'images rectifiées (images fronto-parallèle au plan sur lequel les détails sont recherchés). Ces formes génériques sont paramétrées par quelques valeurs caractéristiques qui permettent d'adapter chaque forme à différents types d'architecture. L'objectif de la méthode de recalage est de déterminer la forme et la position des éléments de détail en s'appuyant sur des mesures de cohérence effectuées dans plusieurs images. La détection de la position des éléments est basée sur les contours des modèles. Compte tenu de la difficulté à localiser les segments correspondant aux limites du bâtiment, les auteurs ont choisi de rechercher les bords un par un. Dans le cas d'un bord vertical, la première étape consiste à extraire une image de gradient horizontal puis à sommer cette image de gradient selon une direction verticale. Cette technique présente l'avantage de mieux résister au bruit présent dans l'image de gradient et la détection des minima permet de localiser les bords du détail recherché. Cette opération de détection des bords est effectuée dans toutes les images disponibles ce qui permet d'identifier les bords candidats quand bien même ils sont invisibles ou trop faiblement marqués dans certaines images. Une fois les bords verticaux identifiés, il est nécessaire d'identifier le type des primitives effectivement présentes. Pour ce faire, les auteurs déterminent la probabilité qu'à chaque modèle de convenir aux éléments observés dans les images. L'information utilisée est un critère de corrélation entre les différentes vues, la probabilité correspondant à un taux de corrélation déterminé au préalable par le biais d'un apprentissage.

3.3 Les approches par modèles complexes (type CAO)

Après avoir présenté des méthodes basées sur l'exploitation initiale de données bas-niveau obtenues à partir d'un appariement automatique de primitives élémentaires, nous présentons désormais des méthodes basées sur une description de type CAO des modèles. Cette modélisation CAO est effectuée par un utilisateur qui sélectionne et assemble des formes 3D définies dans une bibliothèque. Le modèle ainsi défini est localisé manuellement dans les images. L'avantage de ces représentations est de définir implicitement des contraintes, d'offrir immédiatement une modélisation exploitable par un ajustement de faisceaux et de définir le modèle avec ces surfaces. De plus, ces descriptions s'inscrivent naturellement dans la logique des outils de conception et de modélisation utilisés dans l'industrie (CATIA, AUTOCAD, Pro-Engineer, 3D studio, Maya, ...). Cette proximité permet une prise en main rapide de ces approches par les utilisateurs et positionne la photogrammétrie comme une assistance supplémentaire à la modélisation d'objets du monde réel.

Compte tenu de cette convergence de nombreuses recherches ont eu lieu et de nombreux produits commerciaux basés sur ce type d'approches sont apparus sur le marché. Sans être exhaustif, nous pouvons citer certains logiciels de ce type :

- application de relevé industriels : CyliCon de Siemens ([Navab 02]), AOMS ([Sayd 01]), Phidias ([PHI]), fotoG ([?]),

- application pour le rendu réaliste : Photomodeler ([PHO]), Image Modeler ([IMA]), CANOMA ([CAN]).

3.3.1 Une approche par blocs paramétriques

Du point de vue de la recherche, de nombreux travaux ont été réalisés. Debevec a proposé dans [Debevec 96] une méthode de reconstruction de bâtiments commercialisée sous la forme du logiciel FACADE. Le bâtiment est représenté par un ensemble de modèles paramétriques positionnés les uns par rapport aux autres au moyen de transformations rigides dont certains degrés de liberté peuvent être bloqués afin de contraindre les positions relatives de certains blocs par rapport aux autres. L'utilisateur indique la position des arêtes des modèles paramétriques dans les images, ces informations sont utilisées comme mesures dans un algorithme d'optimisation afin de déterminer les paramètres du modèle 3D et les positions des caméras (en général les caméras sont considérées calibrées, néanmoins la méthode fonctionne également avec des caméras non-calibrées).

3.3.2 Utiliser les parallélépipèdes

D'autres approches tentent de tirer partie des propriétés des formes 3D utilisées. Ainsi, parmi les modèles paramétriques possibles, les parallélépipèdes jouent un rôle particulier. Les parallélismes présents dans cet élément ont permis de développer des algorithmes originaux basés sur la détection de points de fuite dans les images. Ainsi, Caprile et Torre dans [Caprile 90] ont montré qu'il était possible d'estimer la longueur focale, le rapport hauteur sur largeur et l'inclinaison des pixels, la position du point principal du capteur et son orientation en supposant que les trois points de fuites détectés dans une image correspondent à trois directions orthogonales. Ce résultat est directement applicable au cas des parallélépipèdes rectangles puisqu'il présente trois directions orthogonales entre elles. Cependant, le calcul de la position des points de fuite est souvent une tâche difficile, aussi Chen et al. [Chen 99] ont-ils proposé d'utiliser directement le parallélépipède afin de calibrer la caméra à partir d'une seule vue. Une approche similaire mais plus générale a été proposée par Wilczkowiak et al. dans [Wilczkowiak 01]. Le choix entre une méthode basée sur des points de fuite et une méthode par parallélépipède dépend essentiellement de la scène observée. Dans certains cas, il n'existe pas de structure 3D suffisamment marquée pour être exploitée alors que la présence de portions d'arêtes permet aisément de calculer les points de fuite. Au contraire, dans d'autre cas, l'identification de formes 3D permet aisément de calibrer la caméra à partir d'une seule vue tout en dimensionnant l'objet.

3.3.3 Une approche probabiliste

Enfin, une autre méthode visant à générer des modèles 3D à partir de jeux d'images a été proposée par Dick et al. dans [Dick 01]. L'approche choisie consiste à décrire le modèle comme un ensemble de briques élémentaires paramétriques. Contrairement aux méthodes précédemment citées qui n'utilisent que la relation existante entre des primitives détectées dans des images et les primitives 3D associées, la méthode présentée ici

tente de reconstruire le modèle en s'appuyant sur des règles spécifiques au type d'objets reconstruits et en recherchant la cohérence entre la structure 3D et la texture présente dans les images.

Le modèle est représenté par un vecteur de paramètres de la forme (n, O_l, O_s, O_t) où n correspond au nombre de primitives du modèle, O_l identifie le type de chaque primitive, O_s correspond aux paramètres de structure de chaque primitive et O_t aux paramètres associés à la texture du modèle. Ce dernier vecteur de paramètres caractérisant la texture contient un ensemble de valeurs comprises entre 0 et 255 correspondant à la valeur de la luminance de chaque point d'une grille défini sur la surface du modèle. L'estimation de ces paramètres est effectuée en maximisant la probabilité du modèle reconstruit par rapport à la forme et à la texture. Dans le cas de la reconstruction de bâtiments, les distributions utilisées afin d'évaluer la forme ont été définies à partir de règles architecturales issues de livres d'architectures [Cruickshank 75] (dépendant par conséquent du type d'architecture observé) et d'observations de bon sens (verticalité,...). Les informations sont intégrées sous la forme de distribution de probabilités et décrivent, par exemple, le rapport entre la largeur et la hauteur d'une fenêtre. De la même manière, l'aspect est pris en compte par le biais d'une distribution décrivant la texture des éléments du bâtiment. Cette distribution a été construite expérimentalement en utilisant 30 images de façades d'architecture classique. Les textures correspondant à des primitives 3D ont été segmentées manuellement dans ces images, puis un processus complexe décrit dans [Dick 01] a permis de déterminer une distribution traduisant l'apparence des bâtiments. L'estimation du modèle le plus probable est un processus en plusieurs étapes. Tout d'abord, un appariement de points et de lignes permet d'obtenir une première reconstruction et d'auto-calibrer les caméras. Par la suite, une approche de type RANSAC prenant en compte les spécificités des bâtiments permet de structurer ce nuage de primitives 3D en plans. Enfin, la reconstruction ainsi obtenue sert de point de départ à des processus d'optimisation visant à maximiser la probabilité associée au vecteur de paramètres décrivant le modèle.

3.4 La modélisation par contraintes : une solution mixte

Jusqu'à présent, nous avons exposé deux approches de reconstruction d'objets structurés à partir d'images non calibrées. La première idée présentée consistait à reconstruire un modèle sans tenir compte des spécificités de l'objet, puis à utiliser les connaissances disponibles afin de simplifier le modèle libre en y définissant des objets structurants. L'avantage majeur de ces approches est d'offrir la possibilité d'une reconstruction totalement automatique à la condition d'être capable d'apparier efficacement des primitives simples (points, segments,...) entre les images disponibles et d'être capable d'identifier les éléments structurants de la scène. Le second type de méthodes consistait en une modélisation de type CAO permettant à l'utilisateur de paramétrer l'objet efficacement. Cette étape de modélisation est suivie d'une estimation des paramètres de la scène à partir de la localisation du modèle CAO dans les images. Cette méthode permet une prise en compte efficace d'un grand nombre de situations difficiles telles que les occultations

mais la finesse de la modélisation dépend de la richesse de la bibliothèque de formes 3D disponible.

L'idée des méthodes mixtes est de proposer des reconstructions plus structurées que dans le cas des reconstructions à partir de nuage de primitives élémentaires tout en conservant la généralité de ces approches, c'est-à-dire en évitant le recours systématique à des bibliothèques conçues en vue d'applications spécifiques. Les informations les plus naturelles à définir afin d'apporter de la structure au modèle sont d'ordre géométrique. La contrainte de co-planarité est assez fréquemment exploitée par les applications de vision car elle se traduit par des homographies liant les éléments coplanaires observés dans les différentes images de la séquence (exemple : exploitation du mouvement d'un ensemble de points coplanaires dans [Szeliski 98]). La même année, Sparr [Sparr 98] introduit des relations de co-planarité et de parallélisme alors que Bondyfalat et al. [Bondyfallat 98] utilisent une technique de réduction polynomiale afin de traiter des relations de parallélisme, d'orthogonalité et d'appartenance de points à des droites et à des plans. Plus récemment des travaux, de Grossman et Santos-Victor [Grossman 00] ainsi que Wilczkowiak et Al. [Wilczkowiak 03a] ont traité le cas de contraintes linéaires et bilinéaires par le biais de résolution de systèmes linéaires. Les travaux de Bazin [Bazin 01] et de Wilczkowiak et Al. [Wilczkowiak 03b] ont abordé la question de la définition de contraintes multiples et redondantes.

Deux stratégies possibles semblent se dessiner pour intégrer des contraintes aux modèles 3D. La première consiste à définir les contraintes sous la forme d'équations et à chercher la solution vérifiant au mieux les contraintes tout en étant cohérent avec les images. La seconde approche décrit les contraintes entre les sommets sous la forme d'un graphe et effectue des simplifications de sorte à obtenir une description minimale du modèle en terme du nombre de paramètres.

3.4.1 La reconstruction sous contraintes comme un problème linéaire

L'objectif de ces méthodes est de définir un système linéaire dont la solution vérifie au mieux les contraintes définies. Dans sa thèse [Grossman 01], Grossman propose une méthode de reconstruction traitant le problème comme un système linéaire. La première étape du processus de reconstruction est un calibrage intrinsèque des caméras à partir des points de fuite (cf. paragraphes précédents) et le calcul des directions principales de la scène (orientation des capteurs et directions des normales aux plans 3D remarquables). Une fois les prises de vue calibrées, des contraintes comme des coplanarités et des rapports de longueurs peuvent être exprimés linéairement en fonction des coordonnées des points 3D.

La contrainte de coplanarité est exprimée sous la forme d'un produit scalaire où v_i et v_j correspondent à deux vecteurs non colinéaires appartenant au plan et X_m , X_n correspondent à deux points appartenant à ce plan. Cette contrainte s'écrit :

$$(v_i \times v_j)^T (X_m - X_n) = 0 \quad (3.1)$$

La contrainte de distance est définie ainsi : Soient m et n appartenant respectivement à deux plans parallèles P et P' de vecteur normal v , et p et q appartenant respectivement

à deux plans parallèles Q et Q' de vecteur normal w . Le rapport de distance α entre PP' et QQ' s'exprime :

$$v^T(X_m - X_n) = \alpha w^T(X_p - X_q) \quad (3.2)$$

En groupant toutes les équations héritées de la définition de telles contraintes, il est possible d'écrire un système de la forme ($X = [X_1^T, \dots, X_N^T]$, la matrice B notée $B(v_1, \dots, v_D)$ ne dépend que des v_i) :

$$B(v_1, \dots, v_D)X = 0 \quad (3.3)$$

L'analyse des propriétés algébriques de cette équation permet à Grossman d'identifier l'existence de contraintes aberrantes et d'écrire $X = UV$ suite à une décomposition de B par SVD. U est une matrice dont les colonnes sont une base orthogonale de B et V est un vecteur. En plus de ces équations de contraintes, les observations sont ajoutées en définissant un nouveau système linéaire basé sur les équations de projection des points 3D dans les images. Après calcul, (cf. page 47 de [Grossman 01]), ce système prend la forme (F : le nombre d'images, X : idem que précédemment, A : des matrices $3N \times 3N$, L : des matrices $3N \times 3F$, T : un vecteur de dimension $3 \times F$ (position des caméras)) :

$$AX + LT = 0 \quad (3.4)$$

En introduisant la relation liant X à la matrice U , ce système devient $AUV + LT = 0$. Les inconnues sont V et T . Par rapport à ces inconnues, ce système admet-il une solution unique (à un facteur d'échelle et une translation près) ? Grossman apporte une réponse en montrant que la solution est unique si et seulement si le rang de la matrice $[AU|L]$ est de rang 4. Ce résultat permet très simplement de déterminer si le système est convenablement contraint... en l'absence de bruit. Toutefois, en présence de bruit dans les données 2D disponibles, l'auteur propose une solution permettant de conserver la validité du résultat précédent.

Dans la même veine, Wilczkowiak et al. propose dans [Wilczkowiak 03a] une méthode permettant de définir des contraintes linéaires et bilinéaires. Après une phase de calibration, les contraintes sont écrites sous la forme d'un système linéaire, les contraintes bilinéaires sont prises en compte si l'un des points intervenant dans leur définition est connu (cette connaissance transforme cette contrainte en contrainte linéaire). Compte tenu de l'incapacité à résoudre le système dans son intégralité en une seule passe, la solution est obtenue en plusieurs itérations. A la première itération, les sommets suffisamment contraints sont calculés, les variables non contraintes étant non traitées. A l'itération suivante, les points déjà estimés sont fixés ce qui permet de propager de nouvelles contraintes à des points non calculés jusqu'alors. Des itérations successives permettent de définir ainsi une solution complète au problème. L'une des tâches délicates de cette approche est de définir si un système linéaire est suffisamment contraint pour être résolu, Wilczkowiak et al. utilisent une analyse basée sur une décomposition de type SVD présenté dans [Bjork 90] [Golub 96] [Press 93]. Sans entrer dans les détails, la condition testée est la nullité des vecteurs du noyau. La présence de bruit dans les positions 2D détectées dans les images perturbe les composantes de ces vecteurs, et rend délicat l'identification des systèmes propres à la résolution. Les résultats obtenus sont toutefois très convaincants

et font de cette méthode une approche rapide et efficace pour les types de contraintes précédemment citées (linéaire et bilinéaire).

3.4.2 Description et simplification des contraintes par graphes

Une autre possibilité pour utiliser les contraintes est de laisser l'utilisateur saisir toutes les contraintes qu'il désire, puis de simplifier le modèle pour obtenir une représentation exploitable de la scène. Dans sa thèse [Bazin 01], Bazin utilise plusieurs types de primitives (2D : points, segments, rectangles, 3D : sommets, trièdres). Après la définition de contraintes géométriques entre ces divers éléments, la méthode proposée effectue une réduction de graphe permettant d'exprimer le modèle sous la forme d'un ensemble de contraintes cohérentes, sans cycle, permettant une utilisation directe du modèle qui est dorénavant défini par un nombre réduit de paramètres.

3.5 Conclusion

Dans le cadre de cette thèse nous avons été amené à choisir parmi ces trois approches celle qui nous semblait la mieux adaptée à notre objectif d'effectuer des reconstructions détaillées. Les techniques automatiques ne permettent pas de fusionner aisément des images avec un faible taux de recouvrement, c'est pourquoi nous avons privilégié les approches type CAO et type contraintes. Dans un premier temps, nous avons choisi d'utiliser une approche par blocs du même type que celle employée par Debevec dans FACADE. Cette approche s'est avérée blocante par la nécessité d'intégrer toujours plus de formes spécifiques dans la bibliothèque de formes et les difficultés de planification que cette limite engendrait pour l'utilisateur. La définition d'un modèle à partir de points liés par des contraintes a finalement été retenue car elle nous permet de conserver une démarche proche de la CAO tout en offrant une plus grande flexibilité et en permettant de traiter un grand nombre de cas.

Chapitre 4

Modélisation et reconstruction d'objets contraints

Sommaire

4.1	Introduction	105
4.2	Modélisation d'un objet au moyen de contraintes	105
4.2.1	Choix du type de contraintes	106
4.2.2	Modélisation par contraintes dures	110
4.2.3	Étapes de construction d'un modèle contraint	112
4.2.4	Fusion d'objets contraints	118
4.3	Reconstruction d'objets contraints à partir d'images	119
4.3.1	Les étapes de reconstruction	120
4.3.2	Evaluation et analyse de la méthode proposée	121
4.4	Conclusion	127

4.1 Introduction

Dans la première partie de ce document, nous avons décrit une méthode d'ajustement de faisceaux permettant d'estimer les paramètres d'un modèle 3D et les paramètres de prise de vues à partir des positions des sommets du modèle dans les images. Nous allons aborder ici le sujet de la modélisation 3D de l'objet observé. Cette méthode a été décrite dans [Cornou 03a] et dans [Cornou 04]. Une fois cette étape achevée, nous disposerons d'un outil de reconstruction à partir d'images et d'un outil de modélisation des objets 3D au moyen de contraintes. Dans la partie suivante, l'utilisation de ces deux techniques nous permettra d'effectuer des reconstructions d'objets structurés et notamment de bâtiments.

4.2 Modélisation d'un objet au moyen de contraintes

A présent, nous allons décrire la méthode de modélisation choisie afin de représenter les éléments 3D de la scène. Nous présenterons tout d'abord le type de contraintes retenu.

Par la suite, nous détaillerons la mise en oeuvre de ces contraintes et, pour finir, nous montrerons comment fusionner différents modèles afin d'obtenir des modèles 3D plus complexes en utilisant toujours la même approche.

4.2.1 Choix du type de contraintes

Une première approche consiste à décrire l'objet par un ensemble de primitives 3D (points, segments...) indépendantes les unes des autres. Cette approche permet de traiter des objets très variés, mais implique un nombre de degrés de liberté très élevé dans le modèle. L'estimation d'un tel modèle nécessite un grand nombre d'informations apportées par des positions de primitives détectées dans les images, et implique généralement la minimisation d'un grand nombre de paramètres par le biais d'une méthode d'optimisation non-linéaire.

Une autre approche consiste à décrire le modèle au moyen de blocs géométriques paramétriques (cube, pyramide, cylindre...) définis dans une bibliothèque de formes. L'avantage de ces blocs est de définir intrinsèquement des contraintes. En effet, l'apport d'un cube implique automatiquement la définition d'angles droits et de parallélismes... La définition implicite de ces contraintes permet de réduire considérablement les degrés de liberté du modèle 3D puisque les paramètres libres sont les paramètres de localisation des primitives 3D dans le repère de reconstruction (orientation et translation) et les paramètres internes au modèle (longueur d'un côté dans le cas d'un cube). Ces approches permettent de définir des modèles avec peu de degrés de liberté (grâce à la définition implicite de contraintes), mais sont limitées par les formes disponibles dans la bibliothèque et le type de liaison pouvant être défini entre les différents blocs (appui-plan, ...).

Nous avons décidé d'employer une troisième approche permettant de tirer partie des avantages des deux méthodes précédentes, c'est-à-dire :

- permettre de traiter des objets de forme assez générique,
- fournir un modèle décrit par le moins de paramètres possibles.

L'idée essentielle est d'utiliser des primitives 3D élémentaires liées entre elles par le biais de contraintes. Dans ce chapitre nous allons tout d'abord définir la notion de contraintes et les propriétés qu'elles doivent posséder afin de nous permettre d'estimer les paramètres de la scène au moyen d'algorithmes de type ajustement de faisceaux. Dans un deuxième temps, nous allons formaliser les contraintes, introduire la structure arborescente que nous employons et donner la liste des contraintes géométriques que nous avons implémentées afin d'effectuer les reconstructions présentées dans la dernière partie de ce document. Dans un troisième temps, nous montrerons comment nous fusionnons deux objets contraints. Enfin, nous expliquerons comment il est possible d'étendre cette description à des contraintes autres que des contraintes purement géométriques. La notion de contrainte est une notion vaste recouvrant tout un ensemble de problèmes et de nécessités physiques. Les contraintes sont employées fréquemment dans de nombreux domaines afin de sélectionner parmi une infinité de solutions possibles celle(s) qui répond(ent) au problème physiquement posé. Afin de structurer la notion de contraintes

nous avons défini deux grandes classes : les contraintes souples et les contraintes dures. Nous allons présenter les spécificités de chacune de ces classes.

4.2.1.1 Contraintes dures

Les contraintes dures sont les contraintes devant être strictement vérifiées. Ainsi, si l'on traite le problème de l'écoulement d'un fluide dans un tuyau, une contrainte naturelle est que l'eau est à l'intérieur du tuyau ce qui définit une frontière infranchissable pour le fluide. Pour toute molécule d'eau (traitée comme un point matériel), on peut donc écrire l'ensemble des positions (x,y,z) accessibles sous la forme d'une inégalité (tuyau de rayon R dont l'axe coïncide avec l'axe Oz du repère) :

$$x, y, z \in \mathfrak{R}^3, x^2 + y^2 < R^2 \quad (4.1)$$

Une telle contrainte définit une zone accessible de l'espace des paramètres et une zone inaccessible (son complémentaire). Dans le cas de la reconstruction par vision, nous rencontrons une telle contrainte pour la longueur focale. En effet, nous souhaitons que l'objet observé par une caméra soit situé devant le capteur, cela implique une longueur focale positive (alors que le modèle sténopé peut tout à fait, d'un point de vue mathématique, admettre des focales négatives). Un modèle acceptable de caméra est donc le modèle sténopé associé à la contrainte : "longueur focale supérieure à zéro". Dans le même ordre d'idée, nous pouvons trouver des contraintes d'égalité stricte. Imaginons, par exemple que l'un des capteurs soit localisé par le biais d'un GPS, il est possible d'imposer à son centre optique de vérifier ces coordonnées.

4.2.1.2 Contraintes souples

D'un autre côté, il peut être souhaitable que des contraintes soit vérifiées mais que cela ne soient pas une nécessité absolue. Par exemple, une entreprise de génie civile devant terminer les travaux à une date donnée est soumise à une contrainte de délai. Il est possible que ce délai ne soient pas respecté, cela ne remettra pas en cause la construction de l'ouvrage mais impliquera des pénalités qui viendront diminuer les revenus de l'entreprise. Cette contrainte s'écrit sous la forme d'un coût supplémentaire ϕ et non pas d'une limite dans l'espace des paramètres associés. Ainsi, le surcoût appliqué à la construction peut s'écrire en fonction du paramètre de durée de construction τ :

$$\begin{cases} \tau \leq \tau_0 \Rightarrow \phi = 0 \\ \tau > \tau_0 \Rightarrow \phi = cste(\tau - \tau_0) \end{cases} \quad (4.2)$$

En reconstruction à partir d'images, un exemple de contraintes souples est le critère de distance entre une primitive 3D projetée dans une image et sa position 2D détectée dans cette image. Il est préférable que cette distance soit la plus faible possible mais les contraintes physiques telles que le bruit ajouté aux positions 2D, les approximations dans les modèles de capteurs et les hypothèses faites sur la primitive 3D rendent impossible un respect strict de la contrainte d'alignement de l'image avec la primitive 3D.

Nous venons d'introduire deux types de contraintes : les contraintes dures correspondant à des contraintes nécessairement vérifiées et les contraintes souples pouvant être

approximativement respectées. Nous allons désormais voir en quoi ces deux types de contraintes interviennent différemment lors de leur intégration dans des algorithmes de minimisation.

4.2.1.3 Intégrations des contraintes dures ou souples dans les méthodes de minimisation

Un algorithme de minimisation (par exemple les méthodes d'ajustement de faisceaux) vise à déterminer l'ensemble des paramètres d'un vecteur \vec{x} tels que la valeur de f soit minimale. Si l'on ajoute **une contrainte dure**, elle peut prendre la forme d'une relation entre les paramètres et s'écrire $g(\vec{x}) = 0$. Dans ce cas, le système à résoudre devient :

$$\begin{cases} \vec{x}, \text{Inf}(f(\vec{x})) \\ g(\vec{x}) = 0 \end{cases} \quad (4.3)$$

Deux stratégies sont alors possibles.

- Soit il existe une relation h permettant d'exprimer explicitement certains paramètres en fonction des autres $\forall \vec{x}', g(h(\vec{x}')) = 0$, auquel cas le nouveau vecteur de paramètres à évaluer est \vec{x}' tel que :

$$\begin{cases} \vec{x}, \text{Inf}(f(h(\vec{x}'))) \\ g(h(\vec{x}')) = 0, \forall \vec{x}' \end{cases} \quad (4.4)$$

Ce changement de variable permet de minimiser la fonction dans un espace de dimension inférieure (dimension de \vec{x}'). La solution obtenue vérifiera nécessairement la contrainte. Nous avons choisi cette approche. Cependant, la réduction du nombre de paramètres n'est pas effectuée a posteriori mais au fur et à mesure de la saisie des contraintes de sorte que le modèle manipulé soit toujours décrit par un minimum de paramètres.

- Soit il n'existe pas un changement de variable adéquat. Dans ce cas, il n'est pas possible de résoudre directement l'équation dans un sous-espace vérifiant la contrainte. Une solution parfois employée consiste alors à résoudre le problème général, puis à chercher la solution la plus proche vérifiant la contrainte. Cette notion de proximité implique la définition d'une norme qui s'avère parfois très difficile à définir (voir même impossible). Cette approche ne conduit pas à une réduction de la dimension du problème à minimiser et n'intègre la contrainte qu'a posteriori.

Dans le cas **des contraintes souples**, une forme courante de prise en compte de la contrainte consiste à pondérer la valeur du critère à minimiser en fonction du respect de la contrainte. Si la contrainte est de la forme $g(\vec{x}) = 0$ (g toujours positive ou nulle) et la fonction $f(\vec{x})$ à minimiser, le critère à minimiser est alors ($f(\vec{x})$ et $g(\vec{x})$ sont des scalaires) :

$$\alpha.f(\vec{x}) + (1 - \alpha).g(\vec{x}) \quad (4.5)$$

α est un poids permettant de pondérer le respect de la contrainte. Si $f(\vec{x})$ et $g(\vec{x})$ sont des fonctions vectorielles, une autre approche consiste à ajouter des équations de mesures en minimisant la norme du vecteur :

$$\begin{pmatrix} f(\vec{x}) \\ g(\vec{x}) \end{pmatrix} \quad (4.6)$$

Dans tous les cas de contraintes souples, la dimension de l'espace dans lequel s'effectue la minimisation n'est pas changée et la contrainte est prise en compte lors de l'évaluation du critère. Rien n'assure que les contraintes seront respectées dans la solution finale, et seule une stratégie de pondération (souvent arbitraire et difficile à mettre en oeuvre) permet de rendre prépondérantes telles ou telles contraintes.

Après avoir présenté brièvement comment il était possible d'intégrer des contraintes dures ou souples dans un processus de minimisation de fonctions, nous présentons désormais les critères de choix que nous avons retenus en vue de la reconstruction d'objets structurés au moyen d'un ajustement de faisceaux. Les objets structurés sont caractérisés par un ensemble de propriétés géométriques communes (orthogonalités, parallélisme,...). Nous allons traduire ces propriétés sous forme de contraintes. Deux choix s'offrent à nous, les contraintes souples ou les contraintes dures.

L'avantage des contraintes souples est de permettre une description approximative de l'objet. Ainsi, un objet présentant un angle approximativement droit peut être reconstruit par ce moyen, le système définissant au mieux l'angle réel lors de l'estimation globale du modèle. Cette force est aussi la faiblesse de cette approche, en effet, ces degrés de liberté impliquent de chercher le minimum dans un espace de paramètres de très grande dimension et donc de disposer d'un grand nombre de mesures. De plus, les équations de contraintes ajoutées aux contraintes initiales de cohérence entre les images et le modèle 3D doivent fournir des équations supplémentaires à intégrer au système dont les pondérations sont difficiles à déterminer.

De part le respect strict des contraintes, les contraintes dures impliquent une absence totale de flexibilité par rapport aux informations fournies lors de la modélisation. Cette absence de flexibilité rend cette méthode sensible aux erreurs et aux approximations de modélisation, mais, d'un autre côté, elle permet de diminuer la dimension du problème à traiter en réduisant le nombre de paramètres libres décrivant le modèle. Cette réduction de la dimension du problème implique qu'une solution peut être obtenue avec moins de mesures que dans l'approche par contraintes souples.

Dans notre application de reconstruction d'objets structurés, et plus particulièrement, de bâtiments, nous souhaitons intégrer des contraintes afin de réduire la phase de saisie d'information dans les images et de faciliter l'estimation du modèle 3D en réduisant l'espace des solutions possibles. Ces deux considérations nous ont conduit à privilégier des contraintes dures. Comme nous l'avons déjà évoqué, ce choix implique un sacrifice quant à la souplesse de la modélisation puisque, par exemple, la définition d'un angle droit strict sera toujours vérifié parfaitement sur le modèle résultant même si dans la réalité cet angle possède une valeur différente. Nous acceptons toutefois ce défaut compte tenu de l'allègement du processus d'ajustement de faisceaux et de la convivialité d'utilisation que nous procurera ce choix.

4.2.2 Modélisation par contraintes dures

Nous allons désormais décrire la méthode utilisée pour décrire les modèles 3D au moyen de contraintes. Notre problème consiste à décrire un modèle 3D constitué d'un ensemble de points 3D liés entre eux par des contraintes. En l'absence de contrainte chaque point 3D est localisé dans l'espace au moyen de trois paramètres représentant les trois coordonnées du point. Un modèle 3D contenant n points est alors décrit par $3n$ paramètres. Si désormais nous introduisons des contraintes permettant de fixer p degrés de liberté, le modèle contraint est caractérisé par $3n - p$ paramètres. Admettons qu'un tel modèle soit disponible, l'apport d'un jeu de paramètres permet de calculer les positions de chacun des sommets 3D du modèle respectant les contraintes (cf. figure 4.1).

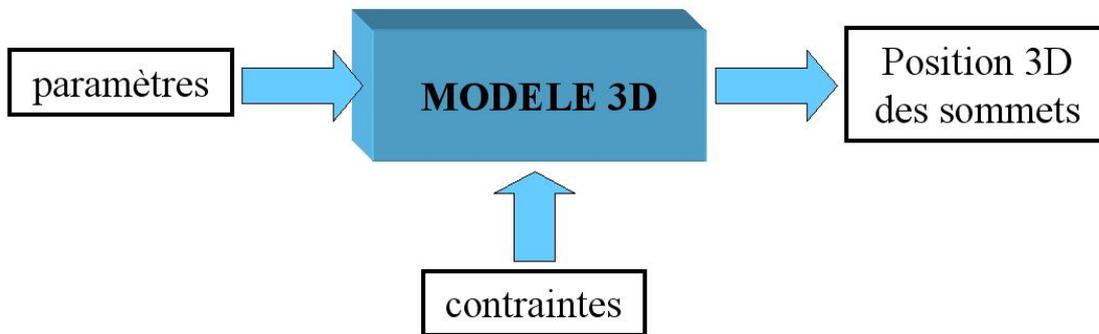


FIG. 4.1: Entrée/Sortie au niveau du modèle 3D.

Lors de la phase de reconstruction du modèle au moyen des images, nous allons utiliser une approche de type ajustement de faisceaux visant à estimer les paramètres du modèle 3D. Le modèle est alors une fonction ϕ telle que $([P_{3D}]$ coordonnées de l'ensemble des points 3D du modèle, x vecteur de paramètres de dimension p) :

$$[P_{3D}] = \phi(x) \quad (4.7)$$

Pour pouvoir estimer ces paramètres, il est essentiel que l'application ϕ représentant le modèle 3D et exprimant la position des points 3D en fonction des paramètres présentent certaines propriétés que nous détaillons désormais. Les propriétés nécessaires à l'application ϕ afin de permettre l'estimation du modèle par le biais d'une technique de minimisation de type Levenberg-Marquard sont :

- continuité,
- dérivabilité,
- unicité : chaque antécédent admet une image unique par ϕ .

La continuité et la dérivabilité sont nécessaires afin de permettre le calcul de la matrice jacobienne et d'approximer la fonction critère minimisée lors de l'ajustement de faisceaux par une forme quadratique. La nécessité que chaque vecteur de paramètres mène à un modèle unique est naturel, par contre, il est tout à fait acceptable qu'un même modèle 3D puisse être représenté par différents vecteurs de paramètres. Cette tolérance nous ramène quelques chapitres en arrière (chapitres 1 et 2), lorsque que nous avons abordé la question du choix de jauge dans l'ajustement de faisceaux. Nous avons montré que, dans certains cas, divers jeux de paramètres pouvaient mener à une même valeur de critère (par exemple, une modification globale du facteur d'échelle ne modifie pas la position des projections des points 3D dans les images). Le même phénomène peut se passer ici. Il est possible que des familles de paramètres conduisent à un même modèle 3D. Faut-il interdire cette éventualité? Non, car une telle interdiction risque de rendre difficile la définition de certaines contraintes (par exemple, la question se pose lors de la définition d'une contrainte d'orthogonalité simple).

Après avoir présenté les propriétés que doit respecter le modèle global obtenu à la fin de la modélisation, nous allons désormais entrer plus en détail dans la description de l'intégration des contraintes dans le modèle. Comme nous l'avons déjà mentionné l'objectif est d'obtenir un modèle final contraint dépendant d'un minimum de paramètres.

Nous disposons d'un ensemble de points 3D et nous souhaitons définir des contraintes entre ces sommets. Deux stratégies sont possibles, soit nous définissons des contraintes sans la moindre organisation, soit l'on impose une démarche dans la saisie des contraintes.

Pour l'utilisateur, l'avantage de définir librement des contraintes est de fournir l'ensemble des informations dont il dispose sur l'objet observé sans se soucier de l'ordre ou de la redondance de celles-ci. L'utilisateur déclare les sommets qu'il souhaite modéliser et définit les contraintes entre ceux-ci. Le résultat d'une telle saisie d'information est schématisé dans la partie gauche de la figure 4.2. Si cette saisie est simple, car totalement libre pour l'utilisateur, c'est au prix d'une complexité non négligeable du modèle contraint ainsi fourni. En l'état, c'est-à-dire juste après la saisie des contraintes, le modèle est inutilisable car il n'est pas possible de définir une paramétrisation minimale¹ du modèle. En effet, certaines des contraintes saisies sont redondantes. Ces redondances se traduisent sous la forme de cycles signifiant que le respect de certaines contraintes est conditionné par d'autres contraintes. La seule méthode pour obtenir un modèle utilisable est de simplifier ce modèle initial en supprimant les redondances. Cette opération implique la définition d'outils complexes permettant de détecter les cycles du graphe et de les réduire ([Bazin 01]).

Pour éviter cette étape de simplification, nous pouvons imposer à l'utilisateur de saisir les contraintes de telle sorte que chaque nouveau sommet créé est contraint uniquement par rapport aux sommets déjà existants (cf. partie droite de la figure 4.2). L'inconvénient d'une telle saisie est d'imposer à l'utilisateur une étape de planification et de rendre impossible l'ajout de contraintes sur des sommets déjà existant. Par contre, l'avantage est d'une telle approche est d'offrir immédiatement une modélisation utilisable. Compte

¹minimale au sens où, pour les contraintes fournies par l'utilisateur, le modèle est décrit par le moins paramètres possibles. Cela n'exclut pas qu'une description différente des contraintes puisse permettre de parvenir à une description de l'objet par un nombre de paramètres encore plus réduit.

tenu de la simplicité de cette dernière approche et de la difficulté à simplifier le graphe de contraintes lorsque les types de contraintes disponibles se complexifient, nous avons choisi d'utiliser la seconde méthode.

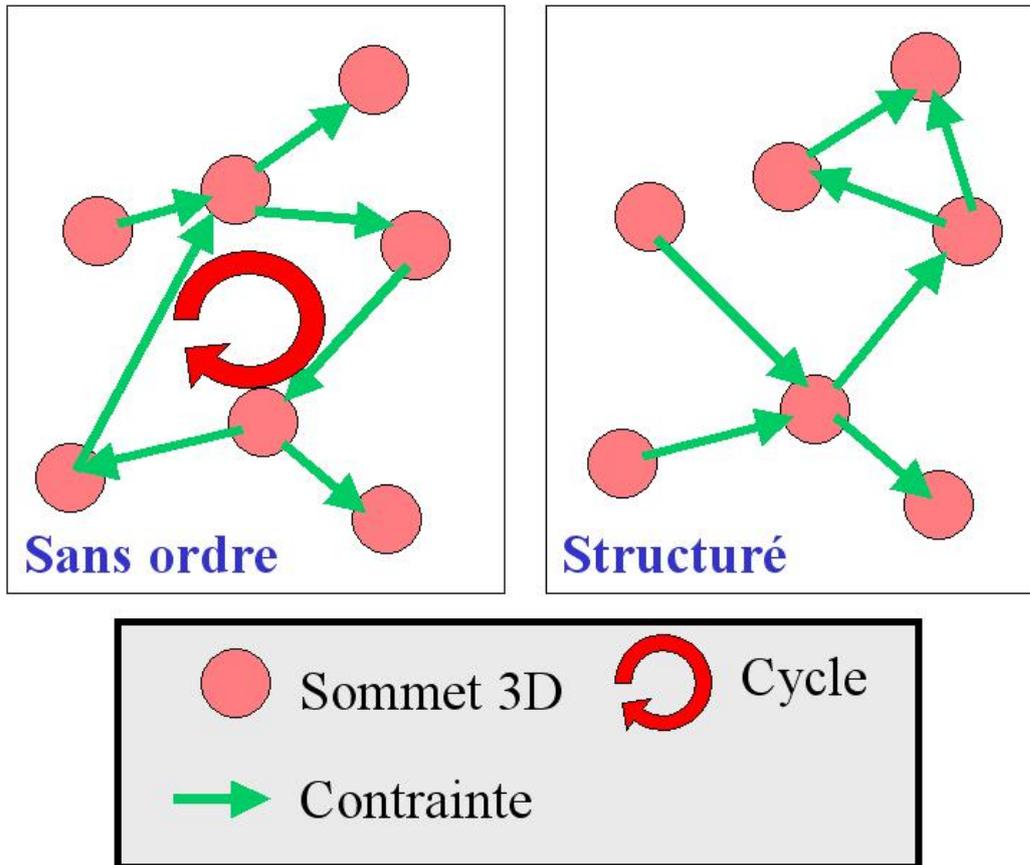


FIG. 4.2: Deux stratégies : fournir les contraintes sans les ordonner ou imposer une procédure arborescente ?

4.2.3 Étapes de construction d'un modèle contraint

Dans le cadre de notre méthode de modélisation, le n -ième sommet est défini par rapport aux $n-1$ précédents. Le premier sommet créé n'est pas contraint (puisque aucun antécédent n'existe) et possède autant de paramètres que de degrés de liberté possible pour ce sommet (trois si seul la position est considérée). Lorsqu'un ensemble de sommet a été défini, l'estimation des propriétés de chaque sommet se fait en chaîne, le sommet n étant évalué une fois que les $n-1$ premiers sommets l'ont été. Cette approche mène à une représentation du modèle 3D sous la forme d'un arbre où chaque noeud représente une primitive 3D. Chacune des branches entrant dans un noeud indique les noeuds antécédents nécessaires à l'évaluation de ses propriétés.

Si l'on considère un noeud particulier, ses propriétés sont définies par la connaissance de ses sommets antécédents et les valeurs des paramètres libres qui lui sont associés. Notons, $\phi_{sommets}$ la fonction traduisant la contrainte appliquée à ce sommet, P_{3D} la position 3D de ce sommet, la position du noeud est caractérisée par un vecteur de paramètres $x_{sommets}$ propre à ce point traduisant ses degrés de liberté et ses antécédents (noeud déjà estimé). La relation liant ces éléments est présentée dans la figure 4.3 et se traduit par la relation :

$$P_{3D} = \phi_{sommets}(x_{sommets}, antecedent) \quad (4.8)$$

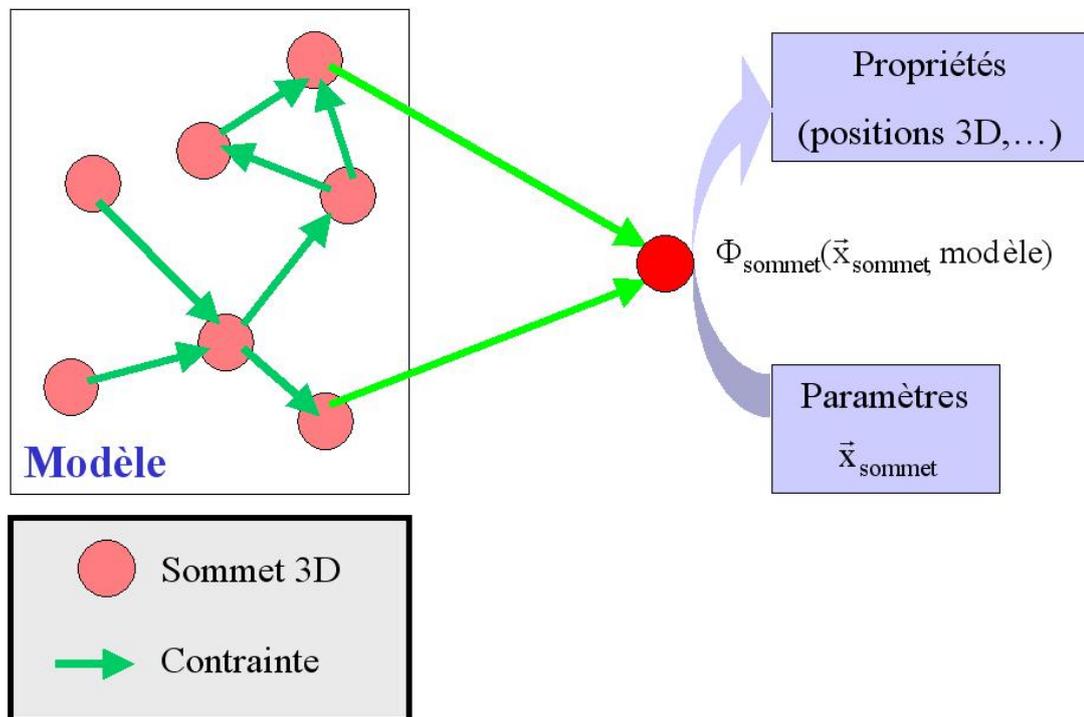


FIG. 4.3: Chaque nouveau sommet est lié au modèle existant par une contrainte. Les propriétés de ce sommet sont calculées en fonction des propriétés de chacun des sommets auxquels il est lié et des valeurs des paramètres qui lui sont associés.

Cette méthode de modélisation possède le désavantage que pour chaque modification d'un paramètre associé à un sommet de l'arbre, il est nécessaire de propager ce changement à tous les sommets en aval de celui-ci. Toutefois le modèle résultant de cette modélisation arborescente est directement exploitable dans un algorithme de type ajustement de faisceaux. En effet, ce modèle est décrit par un minimum de paramètres puisque chaque sommet est paramétré de façon minimale ; le nombre total de paramètres décrivant ce modèle est la somme des degrés de liberté associés aux sommets.

Nous allons désormais appliquer cette méthode de modélisation au cas des objets solides décrits au moyen de règles géométriques. Les primitives considérées ici sont

les sommets 3D d'un solide, et les contraintes sont des relations géométriques établies entre ces sommets. La liste des relations géométriques utilisées dans nos travaux est présentée dans la figure 4.4. Ces relations vérifient toutes les propriétés recherchées dans les contraintes (continuité, dérivabilité,...) et sont toutes relativement simples à implémenter. La connaissance de la position 3D de chacun des antécédents et des valeurs des paramètres associés à chacun des degrés de liberté permet de calculer la position du nouveau point 3D. A titre d'exemple, nous allons détailler l'implémentation des différentes règles disponibles dans notre application (d'autres règles peuvent être développées en vue de mieux répondre aux applications visées) :

- **point libre** : il n'y a aucun antécédent pour définir ce point qui est paramétré par ses trois coordonnées dans le repère monde (repère absolu),
- **égalité vectorielle** : trois points antécédents (A, B et C) sont requis pour calculer la position de D. La contrainte impose $\vec{AB} = \vec{CD}$. On obtient la position de D en calculant $D = C + \vec{AB}$,
- **parallélisme** : trois points antécédents (A, B et C) sont requis pour calculer la position de D. La contrainte traduit la relation $\lambda \vec{AB} / \|\vec{AB}\| = \vec{CD}$. Le seul paramètre associé à D est λ . Connaissant la position de A, B et C. On obtient immédiatement la position de D en calculant (Si A=B nous prenons D=C) :

$$D = C + \lambda \cdot \frac{\vec{AB}}{\|\vec{AB}\|} \quad (4.9)$$

- **orthogonalité simple** : deux antécédents sont requis A,B pour calculer C. La contrainte traduit la relation $(AC) \perp (AB)$. Deux paramètres caractérisent C : un angle θ et une longueur l . Le calcul de la position de C consiste tout d'abord à translater tous les points de -A afin de placer A à l'origine du repère puis à calculer la rotation R_{AB} permettant de placer B sur l'axe z sur repère. Dans ce repère particulier, C_{local} a pour position :

$$C_{local} = l \cdot \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix} \quad (4.10)$$

C s'obtient en calculant $C = R_{AB}^T \cdot C_{local} + A$.

- **orthogonalité double** : trois antécédents sont requis A,B,C pour calculer la position du point D. La contrainte se traduit les relations $\vec{AB} \cdot \vec{AD} = 0$ et $\vec{AC} \cdot \vec{AD} = 0$. Le seul paramètre libre associé à ce nouveau point est un longueur λ . La position de D est donnée par (Si A=B ou A=C nous prenons D=A) :

$$D = A + \lambda \cdot \frac{\vec{AB}}{\|\vec{AB}\|} \wedge \frac{\vec{AC}}{\|\vec{AC}\|} \quad (4.11)$$

- **planarité** : trois antécédents sont requis A,B,C pour calculer la position du point D. La contrainte de planarité associe deux degrés de liberté au nouveau point créé. Ces deux degrés de liberté λ_1 et λ_2 correspondent aux coordonnées de D dans un repère plan construit à partir de \vec{AB} et de \vec{AC} . En pratique D se calcule ainsi (Si A=B ou A=C nous prenons D=A) :

$$D = A + \lambda_1 \cdot \frac{\vec{AB}}{\|\vec{AB}\|} + \lambda_2 \cdot \frac{\vec{AC}}{\|\vec{AC}\|} \quad (4.12)$$

- **égalité de distance** : là encore, trois antécédents sont requis A,B,C pour calculer la position de D. La contrainte traduit la relation $\|\vec{CD}\| = \|\vec{AB}\|$. L'idée est d'utiliser un repère sphérique afin de traduire cette contrainte. Les paramètres associés au point D sont donc θ et ϕ . La position de D s'obtient en calculant :

$$D = C + \|\vec{AB}\| \cdot \begin{pmatrix} \cos(\theta)\cos(\phi) \\ \sin(\theta)\cos(\phi) \\ \sin(\phi) \end{pmatrix} \quad (4.13)$$

Les règles géométriques décrites, dans 4.4, permettent de décrire un grand nombre d'objets en conservant un minimum de paramètres à estimer. Dans la figure 4.5, nous montrons les différentes étapes permettant de définir un parallélépipède rectangle au moyen de ces contraintes. Le modèle final est totalement défini au moyen de neuf paramètres. Ce nombre de paramètres correspond effectivement aux degrés de liberté existants : trois pour la position, trois pour l'orientation et trois pour les dimensions de l'objet.

Nous venons de présenter la méthode de modélisation des solides au moyen de contraintes géométriques. Cette approche arborescente contraint l'utilisateur à définir chaque nouveau sommet par rapport aux précédents, et permet ainsi d'éviter les contraintes redondantes, le recours à des techniques de simplification de graphes et la définition de contraintes incohérentes entre elles. Les modèles obtenus par cette approche sont décrits par un minimum de paramètres et respectent toujours strictement les contraintes fournies. L'exemple du parallélépipède présenté figure 4.5 permet d'envisager la définition de modèles élémentaires proches des briques définies dans les approches type CAO. Par conséquent, nous allons maintenant aborder la possibilité de définir des blocs indépendants et de les fusionner afin d'obtenir un modèle constitué de blocs élémentaires.

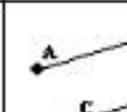
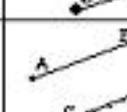
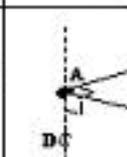
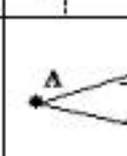
Nom	Description	Antecedents	Equations (contraintes)	Degrés de liberté
Point libre		Aucun	$A(x,y,z)$	3 (coordonnées)
Egalité vectorielle		A,B,C	$CD=AB$	0
Parallélisme		A,B,C	$CD=\lambda AB$	1 (1 longueur)
Orthogonalité simple		A,B	$(AC) \perp (AB)$	2 (1 longueur 1 angle)
Orthogonalité double		A,B,C	$(AD) \perp (AB)$ et $(AD) \perp (AC)$	1 (1 longueur)
Planarité		A,B,C	$AD=\alpha AB+\beta AC$	2 (2 coordonnées)
Egalité de distance		A,B,C	$\ CD\ =\ AB\ $	2 (2 angles)

FIG. 4.4: Ensemble des contraintes géométriques définies. Les degrés de liberté indiquent le nombre de paramètres à estimer par le biais de l'ajustement de faisceaux. Les lettres en caractères gras correspondent aux vecteurs et les lettres en italique indiquent le point contraint par la règle.

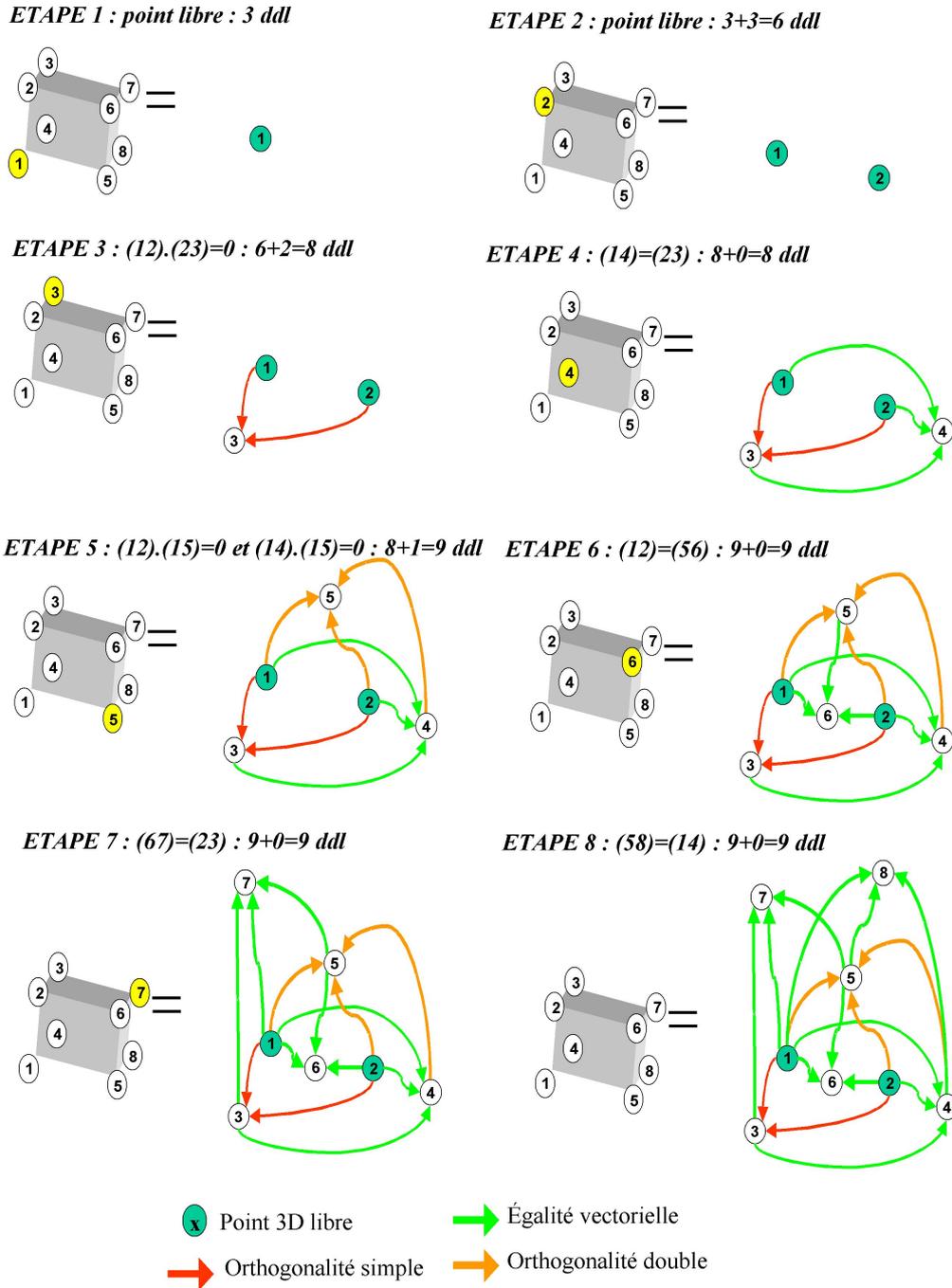


FIG. 4.5: Les différentes étapes de modélisation d'un parallélépipède rectangle. Au total, le parallélépipède rectangle est représenté par seulement 9 paramètres libres.

4.2.4 Fusion d'objets contraints

Lors de la modélisation, il peut être plus simple et plus efficace de définir indépendamment des briques élémentaires au moyen de contraintes que de chercher à décrire l'ensemble du modèle en un seul tenant. Par exemple, il est souvent plus efficace de reconstruire séparément un modèle du corps d'un bâtiment et un modèle de fenêtre, puis de les fusionner ensemble en dupliquant autant de fois que nécessaire la fenêtre, que de chercher à modéliser dans un même modèle complexe l'ensemble de ces éléments. Pour effectuer de telles opérations de fusion, il est impératif d'avoir une modélisation compatible avec cette opération. Prenons deux objets modélisés par deux arbres que l'on souhaite fusionner. Si fusionner signifie seulement placer ces deux objets dans un même repère, il suffit de définir six nouveaux degrés de liberté traduisant l'attitude de l'un des objets par rapport à l'autre. Cependant, cette solution ne permettra jamais d'assurer qu'une fenêtre repose effectivement sur la façade d'un bâtiment. Pour parvenir à contraindre un objet à se positionner par rapport à un autre élément 3D en respectant certaines propriétés, il est nécessaire d'introduire des contraintes.

Prenons un exemple, imaginons deux modèles et essayons d'imposer une contrainte du type "le sommet k du modèle 2 appartient à la droite support des points i et j du modèle 1". La situation est décrite dans la figure 4.6. Dans le cas présenté, la contrainte de liaison des deux modèles peut être en contradiction avec la contrainte liant le point k au modèle 2. Pour lever cette contradiction, différentes stratégies sont possibles.

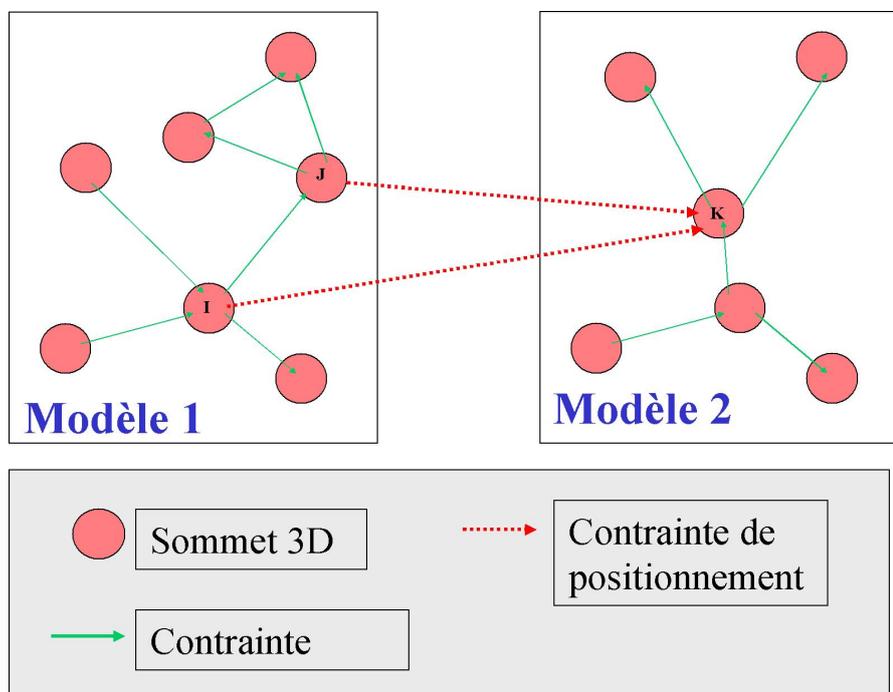


FIG. 4.6: Exemple de contrainte de fusion incompatible avec les contraintes internes au modèle 2.

- La première solution serait de supprimer une des deux contraintes, c'est à dire ré-injecter des degrés de liberté dans le système en négligeant certaines contraintes. Cela implique éventuellement l'apport de nouvelles informations afin de disposer de suffisamment de mesures pour estimer ces paramètres.
- Une autre possibilité est de combiner ces deux contraintes afin de les fusionner en une seule. Ceci revient à simplifier le graphe formé par les deux arbres fusionnés et cela va à l'encontre du choix fait initialement d'éviter ce type de manipulation.
- La dernière possibilité est de simplifier le problème en supposant que l'un des modèles (le modèle 2) a déjà été estimé et que ses paramètres internes sont désormais fixés. Dans ce cas, il est possible de convertir ce modèle en un modèle vectoriel ne dépendant plus d'aucun paramètre interne et d'appliquer à ce modèle une transformation euclidienne (une rotation, une translation et un facteur d'échelle qui modifie de la même façon les positions 3D de tous les sommets du modèle) l'orientant, le positionnant et le dimensionnant par rapport au modèle 1. Dans ce cas, pour définir l'attitude du modèle 2 par rapport au modèle 1, il faut connaître les positions de deux points et une autre direction du modèle 2 dans le repère du modèle 1. Pour établir cette relation entre les deux modèles, nous pouvons définir, en ayant recours aux contraintes désirées, trois nouveaux points dans le modèle 1. Ces trois nouveaux points, mis en relation avec leurs correspondants dans le modèle vectoriel 2, permettent de définir sans ambiguïté le changement de repère de 2 vers 1. On notera que le troisième point sert uniquement à donner une direction et n'intervient pas dans l'établissement du facteur d'échelle qui est calculé à partir de la distance entre les deux premiers points exprimés dans chacun des repères. La qualité du calage des deux modèles est donc sensible à la qualité de reconstruction des trois points choisis comme référence.

Cette dernière voie, bien que limitée et génératrice (potentielle) d'erreurs de placement permet de fusionner deux modèles en respectant strictement les contraintes choisies. Nous verrons que cette approche répond particulièrement bien aux problèmes posés lors de la reconstruction de bâtiments, et notamment, à la question de l'intégration d'éléments de détails précis dans des modèles de moins bonne qualité. Une approche similaire permet également d'envisager la définition d'une bibliothèque de modèles 3D génériques (parallélépipède, pyramide, ...) définis au moyen de ces contraintes afin de proposer à l'utilisateur des briques élémentaires à la manière des approches CAO évoquées dans le chapitre précédent.

4.3 Reconstruction d'objets contraints à partir d'images

Nous allons désormais mettre en oeuvre l'ensemble des éléments présentés précédemment pour retrouver le modèle 3D d'objets structurés à partir d'images non-calibrées. L'utilisateur interviendra pour modéliser le bâtiment en définissant des contraintes et définira la position des sommets du modèle dans certaines images. Dans

un premier temps, nous allons présenter l'ordonnancement des tâches de la reconstruction permettant à un utilisateur de passer des images à un modèle. Dans un second temps, nous présenterons une évaluation de la méthode proposée et une analyse de ses principales propriétés.

4.3.1 Les étapes de reconstruction

Le diagramme de la figure 4.7 présente les grandes fonctions de notre logiciel de reconstruction. Trois grandes classes d'interactions ont été définies : la saisie de données, l'ordre d'effectuer une estimation de la scène et l'exportation d'un modèle utilisable en dehors de ce logiciel afin d'exploiter le modèle 3D obtenu (ce qui est tout de même l'objectif ultime de toute reconstruction). Nous allons prendre chacune de ces trois fonctions principales afin de montrer leur propriété propre et les interactions qu'elles entretiennent les unes avec les autres.

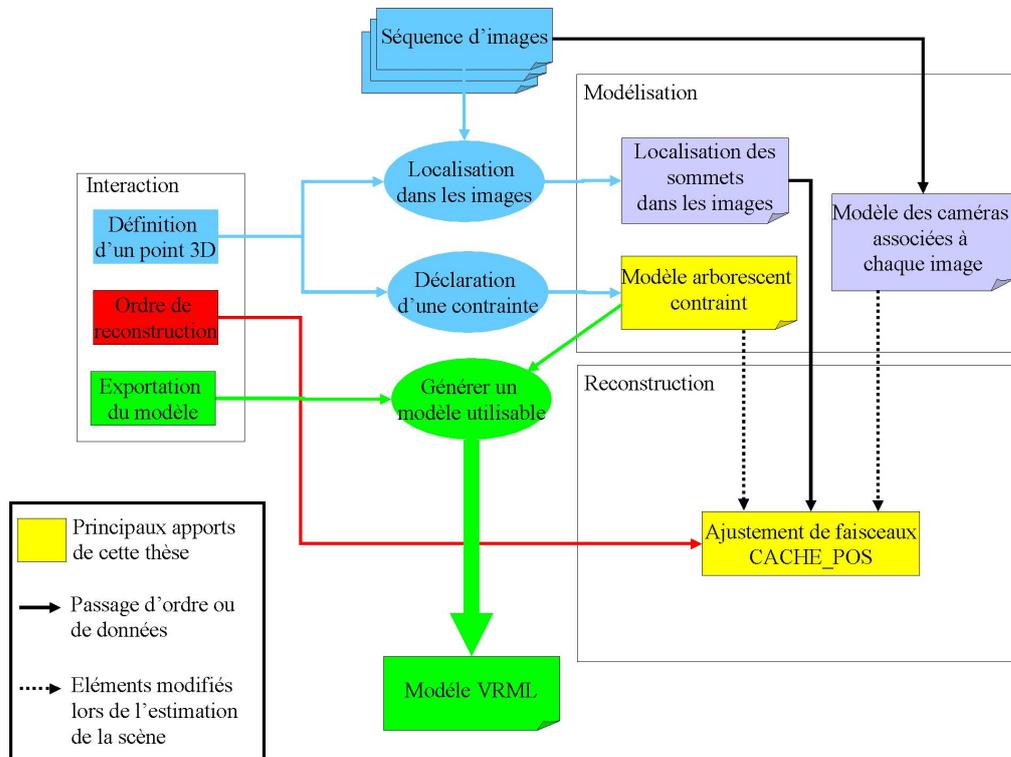


FIG. 4.7: Diagramme présentant l'interaction entre les différents éléments impliqués dans la reconstruction.

La saisie des informations est à la base du processus de reconstruction. Les informations saisies sont la position des primitives dans les images et la définition des contraintes intrinsèques aux modèles. La position 2D des sommets du modèle est donnée par le pointage de cette primitive dans les images concernées. La définition d'une contrainte consiste tout d'abord à sélectionner les sommets antécédents (le nombre de sommets requis dépend

de la contrainte désirée), puis à sélectionner la contrainte choisie dans une liste. Bien sûr, la saisie de contraintes erronées ou une fausse localisation d'une primitive dans une image peut entraîner un échec complet de la reconstruction, mais notre approche réduit le volume de la saisie et, de ce fait, les risques d'erreur.

La phase d'estimation de la scène (capteurs et modèle 3D) peut être exécutée à tout instant par l'utilisateur. Cet ordre lance immédiatement l'exécution de l'algorithme d'ajustement `CACHE_POS` (chapitre 2) qui prend alors, comme état initial, l'état actuel du modèle 3D et effectue une reconstruction. Nous avons vu que cette méthode peut ne pas converger lorsque nous traitons le cas des nuages de points libres, c'est également le cas lors de l'évaluation de modèles contraints. Nous avons montré que le modèle 3D initial est la cause la plus probable de la divergence. Par conséquent, si la solution obtenue ne semble pas convenable (longueurs focales irréalistes, RMS trop important) alors le système génère aléatoirement un nouveau modèle 3D respectant les contraintes définies par l'utilisateur. Les paramètres internes des caméras sont également réinitialisés à des valeurs standards : points principaux aux centres des images, longueurs focales à 2000 pixels. Cette nouvelle situation initiale sert de nouveau point de départ à une estimation. Cette démarche est effectuée jusqu'à obtention d'une solution acceptable. A la place de cette méthode d'initialisation empirique, il serait préférable d'effectuer un choix pertinent dans la position de chacun des sommets 3D du modèle afin de construire un modèle initial plus cohérent. Ce point particulier est la principale perspective d'amélioration à notre méthode de reconstruction mais le recours à des méthodes classiques décrites au chapitre 1 est souvent impossible à cause du peu de points présents dans certaines images et du manque de redondance entre les images.

Quoiqu'il en soit, l'utilisation de `CACHE_POS` associé avec des contraintes permet d'obtenir des reconstructions avec peu de saisies dans les images alors que les méthodes d'ajustement de faisceaux classiques ne parviennent pas à converger dans de telles conditions (initialisation encore plus difficile car il faut positionner les caméras, domaine de convergence plus restreint). Malgré ces difficultés d'initialisation, notre méthode reste utilisable sans initialisation préalable du modèle 3D et prend quelques secondes dans les cas simples, et jusqu'à deux minutes dans les cas complexes pour fournir une reconstruction.

4.3.2 Evaluation et analyse de la méthode proposée

4.3.2.1 Propriétés de convergence

Nous tentons ici de caractériser la méthode que nous avons développée. Il est très difficile d'effectuer une étude systématique de ce type d'algorithme, car les structures des modèles influencent tant les résultats qu'une analyse générale des propriétés est impossible. Cependant, afin d'avoir quelques informations quant à la pertinence de cette méthode, nous avons utilisé un ensemble de modèles reconstruits au cours de cette thèse (une liste exhaustive de ces exemples est fournie en annexe) afin de déterminer quelques grandes lignes du comportement de notre approche.

Nous avons analysé trois exemples (cf. figure 4.8) : un cube, une maison bretonne et un modèle complet du château de Sceaux. Le tableau 4.1 résume les propriétés de

ces trois modèles. Le premier apport de notre approche contrainte est une réduction du nombre de paramètres requis pour définir la position des sommets 3D du modèle.



FIG. 4.8: Exemple de modèles employés pour étudier le comportement de la méthode proposée. A gauche un cube, au centre une maison bretonne et, à droite, le château de Sceaux.

Modèle 3D	Nombre de sommets	Nombre de mesures	Nombre de paramètres		Rapport %
			Non contraint	Contraint	
Cube	8	14	24	9	37.5
Maison bretonne	47	204	141	35	25
Château de Sceaux	1078	496	3234	70	2

TAB. 4.1: Comparaison entre la modélisation par contraintes et la modélisation sans contrainte. Le rapport exprime le pourcentage entre le nombre de paramètres requis lors d'une description contrainte et le nombre de paramètres nécessaire lorsque tous les points sont libres (trois paramètres par point).

Modèle 3D	Nombre de paramètres	Nb d'initialisations avant convergence
Cube	9	3
Maison bretonne	35	3
Avancée du chateau de Sceaux	75	15
Château de Sceaux	70	>50

TAB. 4.2: Nombre d'itérations requis pour obtenir une convergence correcte.

On observe que l'accroissement de la complexité d'un modèle ne se traduit pas par un accroissement proportionnel du nombre de paramètres à estimer. L'exemple du château de Sceaux est le plus remarquable puisque le nombre de paramètres à estimer dans le cas contraint est égal à deux pour cent du nombre de degrés de liberté laissés au modèle dans le cas libre. Ce gain, considérable, justifie l'emploi de `CACHE.POS` comme algorithme d'optimisation. En effet, l'optimisation des 3234 paramètres laissés libre en l'absence de contraintes aurait pleinement tiré partie d'un ajustement de faisceaux classique creux. Cependant, l'utilisation de cette technique requiert une initialisation pertinente du modèle 3D qui semble difficile à obtenir du fait du peu d'information saisie dans les images. Par conséquent, l'utilisation d'une modélisation sans contrainte aurait nécessité un nombre beaucoup plus important de sélections manuelles dans les images.

Au contraire, lorsque l'on intègre les contraintes, la réduction à 70 du nombre de paramètres 3D est tout à fait compatible avec `CACHE_POS`. L'intégration de contraintes permet également de réduire considérablement l'apport d'information sur la localisation des primitives dans les images. Ainsi, dans le cas du château de Sceaux, 496 mesures (soit 248 cliques souris) ont suffi alors qu'il en aurait fallu un minimum de 3234 mesures (plus de 1500 cliques souris) en l'absence de contrainte.

Afin d'évaluer la convergence de notre algorithme `CACHE_POS` en présence de modèle contraint, nous avons pris chaque modèle que nous avons très fortement déformé aléatoirement afin de sélectionner des modèles initiaux très éloignés de la solution. Les déformations appliquées consistent à choisir pour chaque paramètre représentant un angle une valeur entre 0° et 360° au moyen d'un tirage uniforme dans cette intervalle, et à choisir aléatoirement $+1$ ou -1 pour chaque paramètre 3D correspondant à une longueur. En répétant cette opération à plusieurs reprises, nous avons déterminé un taux de convergence pour chaque modèle, les résultats sont présentés dans le tableau 4.2

Il faut être prudent sur les conclusions de ce test. L'idée assez naturelle consistant à penser que plus le modèle est complexe plus le taux de convergence est faible se vérifie, mais il est très difficile de généraliser une telle remarque. L'organisation du modèle et la nature des contraintes définissant le modèle jouent probablement un rôle prépondérant dans la capacité de `CACHE_POS` à converger. Toutefois, comme le montre la figure 4.9, la définition de règles permettant d'initialiser efficacement le modèle reste encore délicate à établir. Dans certains cas, lorsque les données s'y prêtent, il est possible de s'appuyer sur les méthodes classiques d'initialisation. L'apport de connaissances a priori sur le modèle doit permettre d'améliorer ces performances.

En définitif, notre méthode permet d'obtenir une réduction drastique du nombre de paramètres à estimer et de saisir beaucoup moins d'information au niveau de l'image. Toutefois, l'introduction de contraintes complique l'initialisation et en l'absence de connaissance a priori sur la forme 3D recherchée, il apparaît dans la pratique qu'une stratégie itérative consistant à alterner les phases de modélisation et de reconstruction est préférable. En effet, elle fournit une bonne initialisation à `CACHE_POS` puisqu'à chaque itération le modèle 3D est proche de la solution. De plus, elle permet à l'utilisateur de voir le modèle 3D se former au fur et à mesure des étapes de reconstruction. Au cours des différentes reconstructions que nous avons réalisées, nous avons toujours obtenu un résultat positif en utilisant cette méthode.

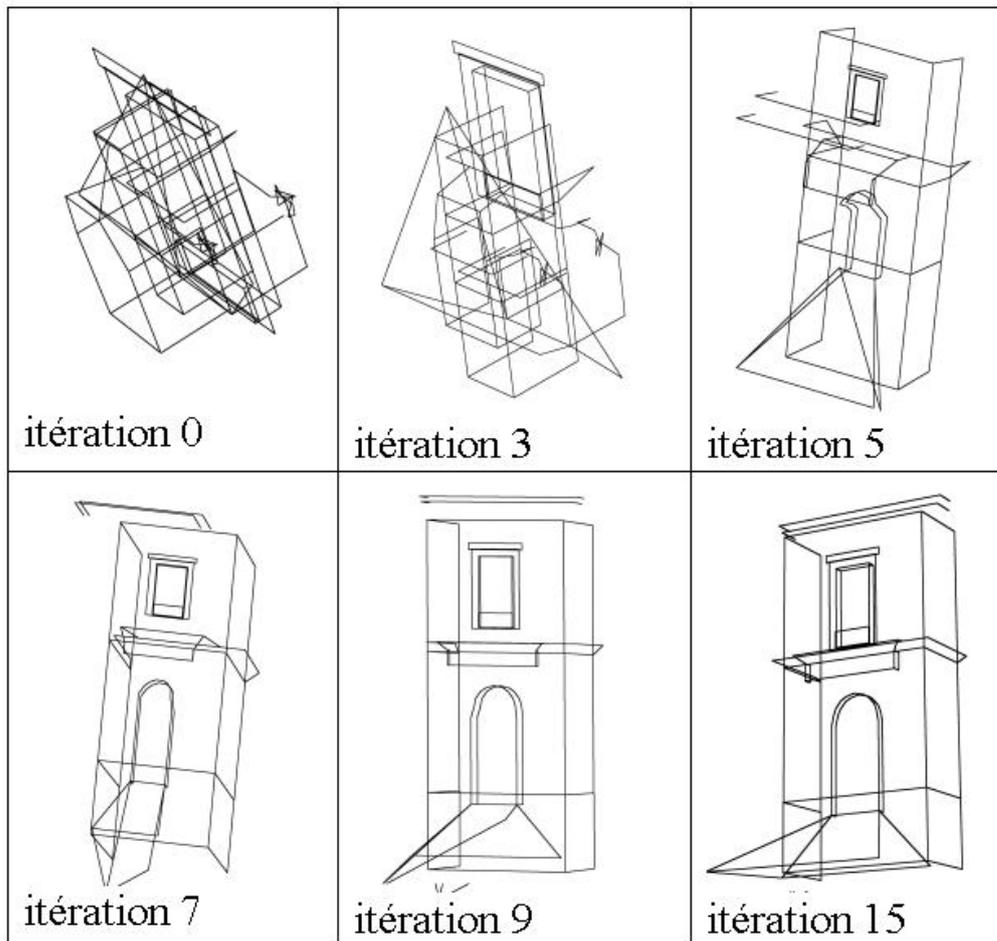


FIG. 4.9: Exemple de l'avancée du chateau de Sceaux : les différentes étapes d'une convergence réussie à partir d'un modèle très éloigné de la solution.

4.3.2.2 Mise en évidence de l'apport des contraintes

Afin de mettre en évidence l'intérêt d'utiliser des contraintes nous avons repris quelques images de la séquence de bureau vue au chapitre 2 et nous avons reconstruit la surface de certains éléments. Deux reconstructions ont été effectuée :

1. une reconstruction à partir de deux images (figure 4.10) sans utiliser de contraintes,
2. une reconstruction à partir d'une seule image (figure 4.11) en utilisant des contraintes.



FIG. 4.10: Les deux images utilisées pour reconstruire le modèle 3D sans contraintes.

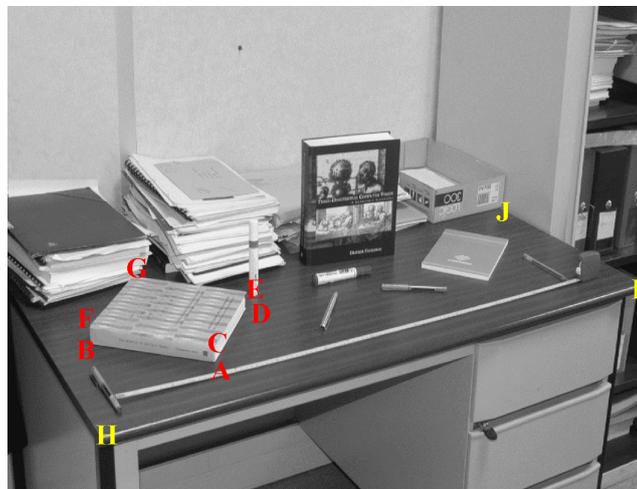


FIG. 4.11: L'image utilisée pour reconstruire le modèle 3D au moyen de contraintes.

Afin de montrer un exemple de modélisation à partir de la figure 4.11, nous allons présenter dans la définition des sommets d'un livre (sommets en rouge) et de quelques points du plan (sommets en jaune).

Le livre est un parallélépipède, le processus d'introduction des contraintes est donc le suivant (tableau 4.3) :

Etape	Nouveau sommet	Contraintes	Nb de paramètres libres
1	A	libre	3
2	B	libre	3
3	C	$\vec{AC} \cdot \vec{AB} = 0$	2
4	F	$\vec{AC} = \vec{BF}$	0
5	D	$\vec{AC} \cdot \vec{AD} = 0$ et $\vec{AB} \cdot \vec{AD} = 0$	1
6	E	$\vec{DE} = \vec{AC}$	0
7	G	$\vec{EG} = \vec{CF}$	0

TAB. 4.3: Définition d'un parallélépipède.

puis nous ajoutons les points H, I et J marquées en jaune qui sont coplanaires à A, B, D (tableau 4.4).

Etape	Nouveau sommet	Contraintes	Nb de paramètres libres
8	H	coplanaire à A, B, D	2
9	I	coplanaire à A, B, D	2
10	J	coplanaire à A, B, D	2

TAB. 4.4: Ajout de points coplanaires.

La figure 4.12 présente le résultat des reconstructions sans et avec contraintes. Les sommets permettant de définir deux nouveaux plans ont été ajouté à la scène afin de permettre une meilleure visualisation de la différence entre les deux approches. On observe que la reconstruction sans contraintes comporte de nombreuses imperfections et que les livres lévitent au dessus du plan du bureau malgré l'utilisation de deux images. A l'opposé, l'intégration des contraintes force les livres à reposer sur le bureau et permet de reconstruire la scène avec un meilleur rendu visuel malgré l'usage d'une unique image.

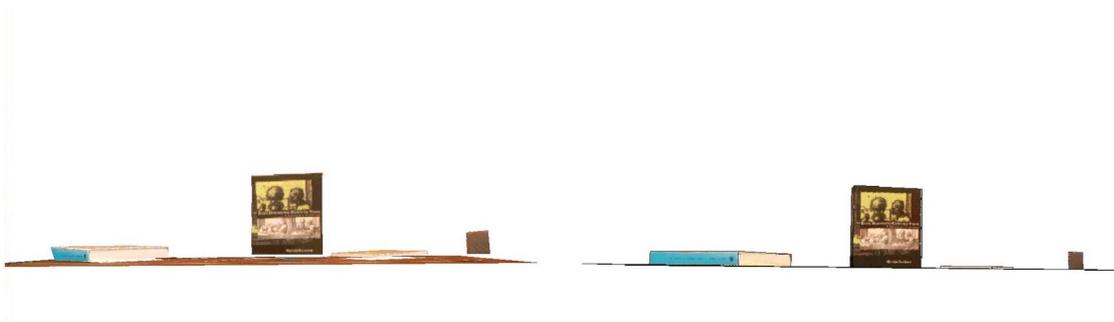


FIG. 4.12: Comparaison entre les reconstructions contraintes ou non. A gauche le résultat sans contrainte, à droite celui avec contraintes.

4.4 Conclusion

Nous avons vu dans ce chapitre une méthode de modélisation supervisée d'objets structurés par des contraintes. Nous avons tout d'abord introduit la notion de contraintes et défini les propriétés que doit posséder un modèle pour pouvoir être estimé en utilisant des algorithmes de minimisation de type Levenberg-Marquardt. Par la suite, nous avons proposé une structure arborescente permettant d'obtenir un modèle décrit par un minimum de paramètres et respectant toujours strictement les contraintes définies. Cette approche, plus exigeante qu'une approche par graphe, du point de vue de la saisie des contraintes, présente l'avantage de ne requérir aucune opération de simplification de graphes et d'offrir une représentation du solide, structuré par des contraintes, directement utilisable.

En vue de l'application de ce type de modélisation au cas des bâtiments, nous avons introduit la liste des contraintes qui seront employées par la suite. Nous avons également exposé une méthode simple permettant de fusionner des objets modélisés indépendamment afin d'obtenir un modèle contraint plus complexe. Cette étape de fusion, limitée au cas où les paramètres d'un des deux modèles à fusionner sont connus, va s'avérer essentielle dans l'intégration de certains éléments de détail dans des modèles 3D globaux de bâtiments.

Nous avons désormais l'ensemble des éléments nous permettant d'effectuer la reconstruction de modèles 3D. Nous disposons, en effet, d'un algorithme d'ajustement de faisceaux présenté au chapitre 2 qui permet d'estimer les paramètres d'un modèle en fonction de la position des sommets 3D dans des images, et nous avons défini une méthode de modélisation par contraintes dans ce chapitre. Par conséquent, nous allons présenter dans le chapitre suivant l'utilisation de ces outils dans le cadre de la reconstruction de bâtiments.

Troisième partie

Applications pratiques

Chapitre 5

Application à la reconstruction de bâtiments

Sommaire

5.1	Introduction	131
5.2	Spécificités des scènes architecturales	132
5.2.1	Problème du point de vue	132
5.2.2	Problème de localisation des primitives	132
5.3	Exemple sur un bâtiment complexe : le château de Sceaux	136
5.3.1	Présentation de la séquence	136
5.3.2	Processus de reconstruction	138
5.3.3	Résultats	142
5.4	Autres exemples	149
5.5	Conclusion	152

5.1 Introduction

Nous avons présenté une méthode d'ajustement de faisceaux et une technique permettant d'utiliser des contraintes pour modéliser un objet. A partir de ces deux éléments, nous disposons d'une méthode interactive complète permettant de modéliser un objet à partir d'images. Nous avons utilisé cette approche afin de modéliser des bâtiments, ces résultats ont été présentés dans [Cornou 03b] et [Cornou 02b]. Cette application a été choisie car elle présente à la fois un intérêt scientifique du fait des difficultés à reconstruire des objets observés dans des environnements non contrôlés et un intérêt pratique dans la perspective de l'obtention de modèles réalistes de bâtiments.

Dans ce chapitre, nous exposons tout d'abord les spécificités des scènes architecturales. Puis, dans un second temps, nous présentons des exemples de modélisation de bâtiments obtenus au moyen de notre approche. Le principal exemple présenté ici est la modélisation du château de Sceaux obtenue à partir de photographies numériques non-calibrées. Cette exemple illustre la possibilité de faire des reconstructions de bâtiments

complexes en vue de la préservation du patrimoine architectural et met en évidence l'avantage d'une approche stratifiée afin de permettre à l'utilisateur de concentrer le travail et la qualité de la reconstruction sur les éléments lui paraissant importants.

Les deux autres exemples présentés sont une maison bretonne et une église romane. Ces bâtiments sont reconstruits beaucoup plus sommairement que le château de Sceaux. Ceci démontre que cette méthode peut aussi permettre de reconstruire rapidement (environ 1h30) des modèles simples de bâtiments en ouvrant ainsi la voie à des applications grand public.

5.2 Spécificités des scènes architecturales

Avant d'aborder la description des cas pratiques, attachons nous tout d'abord à l'étude des spécificités liées aux scènes architecturales. Dans le cadre d'une application de reconstruction d'un modèle 3D à partir d'images, deux éléments vont intervenir successivement. Le premier d'entre eux traite de la capacité à prendre des photographies à partir de points de vue adaptés à la reconstruction. Le second intervient lors de l'exploitation des images et concerne la capacité à utiliser ces images.

5.2.1 Problème du point de vue

Le problème du choix des points de vue a été très étudié par les photogrammètres. Le chapitre 9 de [Atkinson 96]¹ traite des questions de choix de la disposition des prises de vue autour d'une scène ou d'un objet. Dans le contexte industriel, il est souvent possible de positionner les capteurs au mieux afin d'obtenir une configuration permettant d'atteindre la plus grande précision possible. Dans le cas de la reconstruction de bâtiments, on ne rencontre quasiment jamais cette opportunité. En effet, la plupart des bâtiments sont construits en ville, et les points de vue disponibles sont souvent limités par les murs des bâtiments environnants. De plus, les véhicules, la végétation et bien d'autres éléments urbains constituent autant de gênes limitant les possibilités de prises de vue. Dès lors, il va être nécessaire de composer avec tous ces éléments et d'effectuer des prises de vues partielles qui, jointes, doivent permettre d'obtenir une couverture suffisante de l'ensemble des primitives décrivant le bâtiment. A titre d'exemple, la figure 5.1 montre les limites ayant conditionnées les prises de vue autour d'une maison bretonne.

5.2.2 Problème de localisation des primitives

En plus du problème du choix des points de vue, se pose la question de la possibilité de localiser les primitives intéressantes dans les images. Différentes difficultés peuvent en effet se présenter. Le premier type de difficulté concerne les cas où le sommet recherché est invisible (cf. figure 5.2). L'invisibilité peut être due à la présence d'un objet statique (un arbre...) ou au passage d'un élément en mouvement (piéton, voiture...).

L'autre cas de figure concerne les situations où la localisation d'une primitive dans les images est difficile. Ce problème peut être provoqué par la présence de divers éléments

¹chapitre rédigé par C.S. Fraser

dans le voisinage proche de la primitive visée (gouttière, raccord de toit, ...). Nous avons rencontré ce problème à plusieurs reprises dans l'exemple du château de Sceaux notamment au niveau des coins des faces principales du bâtiment à cause des pierres d'angle et du sol ainsi au niveau du toit à cause des gouttières et de la multitude d'éléments décoratifs.

Une idée pour s'affranchir de ces difficultés peut être d'utiliser les contours extraits de l'image (par exemple, en utilisant un filtre de Canny-Deriche [Deriche 87]). Les contours sont moins sensibles aux occultations, car lorsque l'on recherche un segment, il est possible d'effectuer cette détection en ne voyant qu'une partie de celui-ci. Cependant, les contours ne sont pas toujours facilement détectables notamment du fait de la texture des murs qui présentent de nombreux joints rectilignes qui provoquent de nombreuses confusions. Nous n'avons pas approfondi cette voie qui a par ailleurs été déjà utilisée dans de nombreux travaux dont notamment ceux de Debevec et al. [Debevec 96].

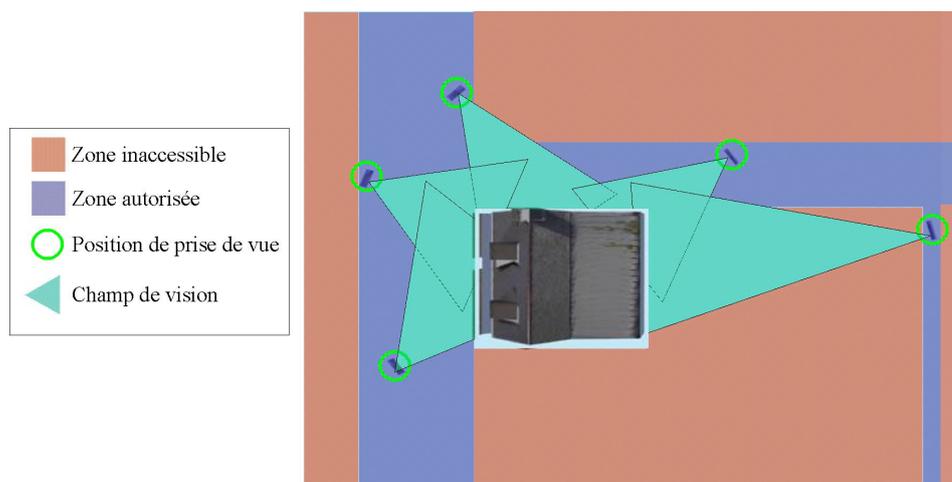


FIG. 5.1: Contraintes sur les positions de prise de vue dans le cas d'une maison bretonne (vue du dessus).



FIG. 5.2: Exemple d'occultations générées par un objet statique et une personne en mouvement.



FIG. 5.3: Façade du château de Sceaux.



FIG. 5.4: La mairie de Londres. (*Greater London Authority Building, Londres. Date : 24 novembre 2002 - Photo par Philipp Holzmann*)

5.3 Exemple sur un bâtiment complexe : le château de Sceaux

Nous allons présenter désormais l'ensemble du processus de reconstruction ayant permis d'obtenir un modèle tridimensionnel du château de Sceaux à partir d'images non calibrées. Afin d'évaluer notre outil sur une séquence difficile, nous avons choisi de reconstruire ce château (cf. figure 5.3) à partir d'une séquence d'images obtenues au moyen d'un appareil photographique numérique². Comparé à certains bâtiments modernes comme la mairie de Londres (figure 5.4), le château de Sceaux possède une forme a priori simple puisqu'il est globalement polyédrique. Cependant, nous avons été confronté à de nombreuses difficultés liées aux nombreux détails présents sur ce monument et à la limitation du nombre de points de vue. Malgré tout, nous avons été en mesure d'obtenir un modèle détaillé de ce bâtiment. Nous allons, dans un premier temps, présenter la séquence d'images qui nous a servi à effectuer cette reconstruction. Dans un deuxième temps, nous présenterons les différentes étapes du processus de reconstruction. Pour finir, nous présenterons les résultats obtenus.

5.3.1 Présentation de la séquence

En moins de deux heures, dix-neuf photographies (cf. figure 5.5) ont été prises à partir du sol avec un appareil photographique numérique et deux objectifs de longueur focale 28 mm et 135 mm. La dimension des images est de 2000x3008 pixels. Les distorsions n'ont pas été corrigées, les focales et les points de prise de vue sont inconnus.

Le château de Sceaux est constitué d'un ensemble complexe d'éléments architecturaux dont la reconstruction nécessite une organisation préalable du processus de modélisation. Cette organisation a consisté à diviser le travail de modélisation en plusieurs parties correspondant à différents éléments du château dont des images sont présentées en figure 5.5. Dans le cas présent, nous avons tout d'abord modélisé le corps du bâtiment au moyen huit images prises autour du château. Puis, afin de reconstruire ce bâtiment avec plusieurs niveaux de détail (en fonction des centres d'intérêt de l'utilisateur final), nous avons défini quelques éléments dont nous souhaitions une reconstruction fine :

- l'avancée au centre de la façade (3 images disponibles, objectif 28mm),
- une fenêtre du rez-de-chaussée (3 images disponibles, objectif 28mm),
- une fenêtre du premier étage (2 images disponibles, objectif 28mm),
- un chien-assis (3 images disponibles, télé-objectif environ 135mm).

Le choix de certains éléments du modèle permet de concentrer l'effort de reconstruction sur les éléments importants pour l'utilisateur, de réduire le temps de reconstruction et de définir des éléments génériques (objets métier) que l'on peut répliquer (ce sera le cas des fenêtres et du chien-assis dans notre exemple). Le choix de l'utilisateur n'est pas définitif et il sera toujours possible d'apporter de nouvelles images afin de définir de nouveaux éléments de détail à fusionner avec le modèle.

²appareil D100 de chez Nikon



FIG. 5.5: Les images de la séquence du Château de Sceaux.

5.3.2 Processus de reconstruction

Tous les modèles définis (cf. figure 5.9) ont été reconstruits en suivant le même processus. Prenons l'exemple d'une fenêtre du rez-de-chaussée présentée sur la figure 5.7. Nous avons défini un modèle contraint en utilisant la description par graphe présentée au chapitre 4. Le modèle obtenu en fin de modélisation par des contraintes est caractérisée par 14 degrés de liberté suffisant à décrire la position de chacun des 20 sommets 3D représentant la fenêtre.

L'utilisation de l'ajustement de faisceaux CACHE_POS décrit au chapitre 2 a permis de définir la valeur des 14 paramètres inconnus du modèle 3D de la fenêtre et les deux paramètres f_x et f_y et les attitudes caractérisant le capteur utilisé pour acquérir les images. La figure 5.8 montre le modèle tridimensionnel de la fenêtre projeté dans les images, le RMS final était de 2.5 pixels.

La fin du processus a consisté en la définition manuelle des surfaces du modèle puis à l'extraction des textures depuis les images. Les problèmes liés à ces deux dernières étapes sont présentés dans le dernier chapitre de ce document.

Le même processus de reconstruction que celui qui vient d'être décrit pour la fenêtre du rez-de-chaussée a été appliqué à tous les éléments du bâtiment.

Par la suite, tous les éléments ont été fusionnés afin d'obtenir un modèle global du château de Sceaux (cf figure 5.10). Par exemple, afin de localiser les fenêtres du rez-de-chaussée, nous avons ajouté trois sommets supplémentaires pour chaque fenêtre au modèle représentant le corps de bâtiment. Chacun de ces trois sommets représente un coin de fenêtre et est localisé par rapport au corps du bâtiment au moyen de contraintes identiques à celles utilisées pour modéliser les différents éléments. A partir de ces trois sommets, le changement de repère entre le repère associé au modèle générique de fenêtre et le modèle global du bâtiment est calculé (cf. chapitre 4). Le modèle générique est uniquement modifié par un facteur d'échelle appliqué à toute sa structure et conserve par conséquent les mêmes proportions et la même précision relative (les rapports de longueur sont conservés). Les textures des fenêtres ont été définies précédemment en utilisant des images de gros plan et ont donc une résolution relative à leurs images originales.

L'utilisation de cette modélisation stratifiés permet d'adapter la qualité de la reconstruction au besoin. Ainsi, si l'on observe le modèle final en détail, on peut remarquer que le rebord de la fenêtre a été reconstruit alors que ce détail aurait été inaccessible si nous n'avions pas utilisé des images de gros plans. De plus, on peut voir sur la figure 5.6 que cette texture est très détaillée et permet, par exemple, à l'utilisateur de voir très précisément l'état des boiseries de cette fenêtre en regardant le modèle 3D (les visseries sont observables).



FIG. 5.6: Zoom sur la texture du modèle 3D de la fenêtre.

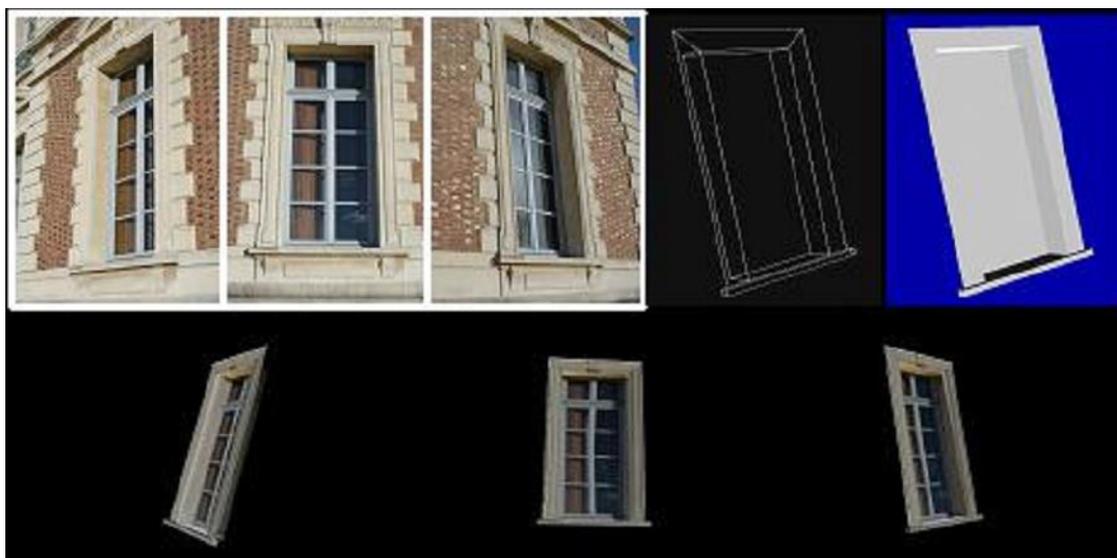


FIG. 5.7: Exemple de la fenêtre du rez-de-chaussée. Les trois images utilisées pour la reconstruction sont en haut à gauche. En haut à droite se trouve une image du modèle filaire obtenu et une image du modèle surfacique. Enfin, en dessous, il y a trois images du modèle texturé.



FIG. 5.8: Projection d'une partie du modèle 3D dans les images. Seul les segments pour lesquels les deux extrémités ont été localisées manuellement dans les images sont représentés.



FIG. 5.9: Autres éléments de détail reconstruits.

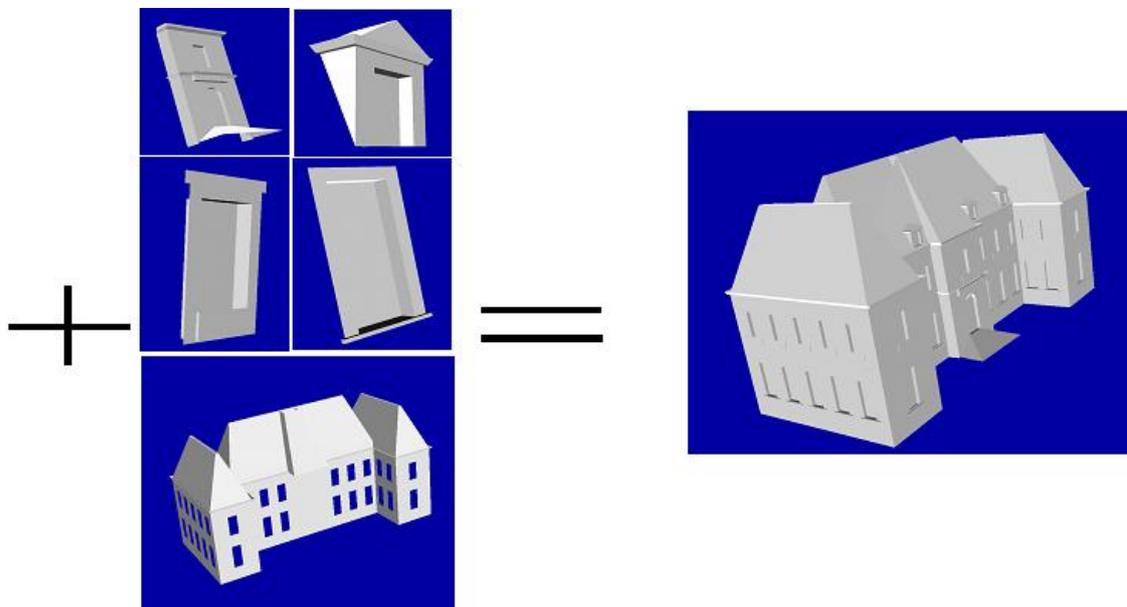


FIG. 5.10: Fusion des différents éléments de détail afin d'obtenir un modèle global.

5.3.3 Résultats

Le modèle final obtenu contient plus de 1000 sommets 3D et comporte différents sous-ensembles caractérisés par des précisions de reconstruction différentes. L'utilisation de contraintes a permis de décrire le modèle final avec seulement 70 paramètres. D'un point de vue pratique, l'alternance entre les étapes de modélisation (saisie de contraintes et de positions dans les images) et les étapes d'estimation de la scène s'est avérée pertinente en offrant à l'utilisateur une visualisation en cours de modélisation du modèle et en permettant de fournir une initialisation suffisante à l'algorithme d'estimation des paramètres de la scène à partir des images.

Deux jours de travail ont été nécessaires pour obtenir ces résultats et le RMS final est de 6.3 pixels. Ce résidu 2D relativement élevé s'explique par la difficulté à définir précisément la position des extrémités du bâtiment dans les images, par la présence de distorsions (ces distorsions n'ont pas été corrigées car aucun calibrage préalable n'a été effectué!) et par des approximations lors de la modélisation (cf. 5.13). Malgré tout, cette approche stratifiée nous a permis de reconstruire un modèle complet du château tout en conservant la précision et la texture acquises à la résolution maximale pour chacun des sous-modèles traités. Il est toujours possible de compléter le modèle et d'affiner sa précision en fonction du besoin, des images et du temps disponible.

La figure 5.13 et 5.14 montre la projection du modèle en fil de fer sur deux images. Ce décalage s'explique, d'une part par la définition inappropriée de certaines contraintes, et d'autre part par la difficulté à sélectionner la position des coins du toit qui sont difficilement visibles sur les images, aucune vue du dessus n'étant disponible. La figure 5.12 montre le modèle surfacé. Les figures 5.15, 5.16 montre le modèle en fil de fer, on notera le faible nombre d'éléments de surface définis (excepté la complexité introduite par la courbure de la porte). Les figures 5.17, 5.18, 5.19 et 5.20 présentent des images texturées de ce modèle. La texture est loin d'être parfaite et l'on peut remarquer notamment des disparités entre la texture des éléments de détail et la texture du modèle global. La prise en compte de ce phénomène et l'exploitation complète du modèle 3D (gestion des occultations) afin d'obtenir une texture de qualité est un problème difficile qui n'a pas été abordé dans cette thèse. Nous proposerons néanmoins quelques pistes de réflexion sur ce sujet dans le prochain chapitre.



FIG. 5.11: Projection du modèle en fil de fer sur une image globale.

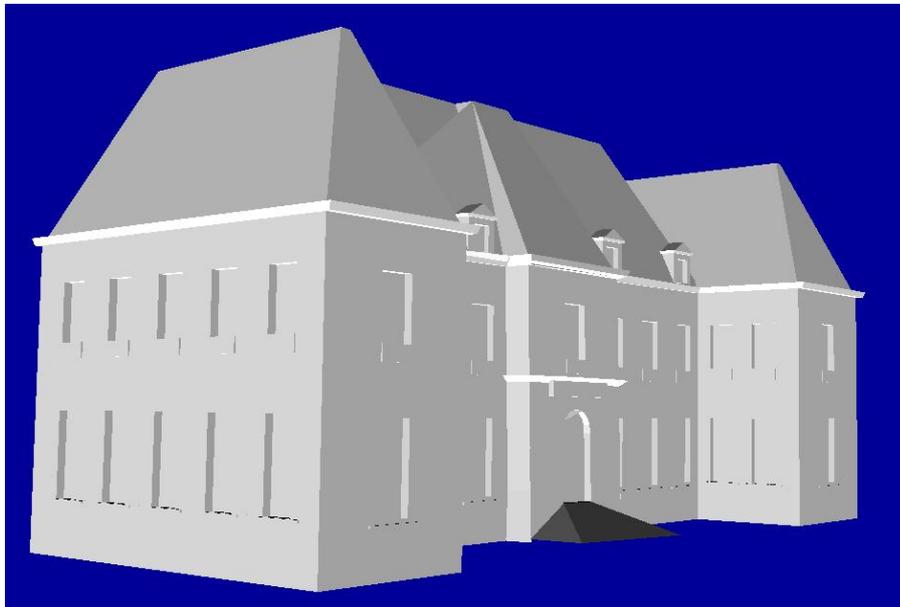


FIG. 5.12: Modèle surfacé mais non texturé.



FIG. 5.13: Détail du modèle filaire reprojété. On observe que l'arête du mur vertical à droite de l'image est mal positionnée. Cette erreur est due à la définition d'un arête complètement rectiligne ne tenant pas compte des décrochements présents.



FIG. 5.14: Projection du modèle fil de fer de l'avancée dans une image.

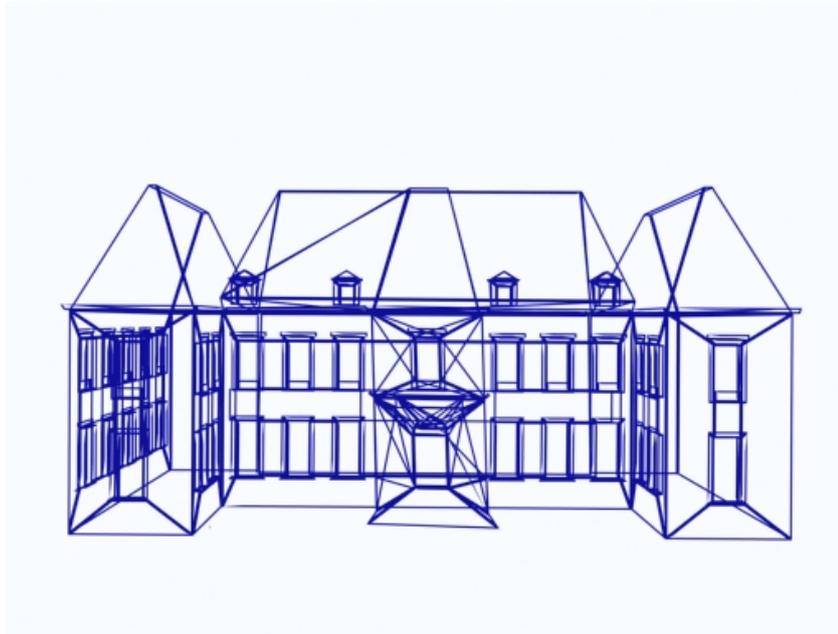


FIG. 5.15: Modèle représenté en fil de fer.

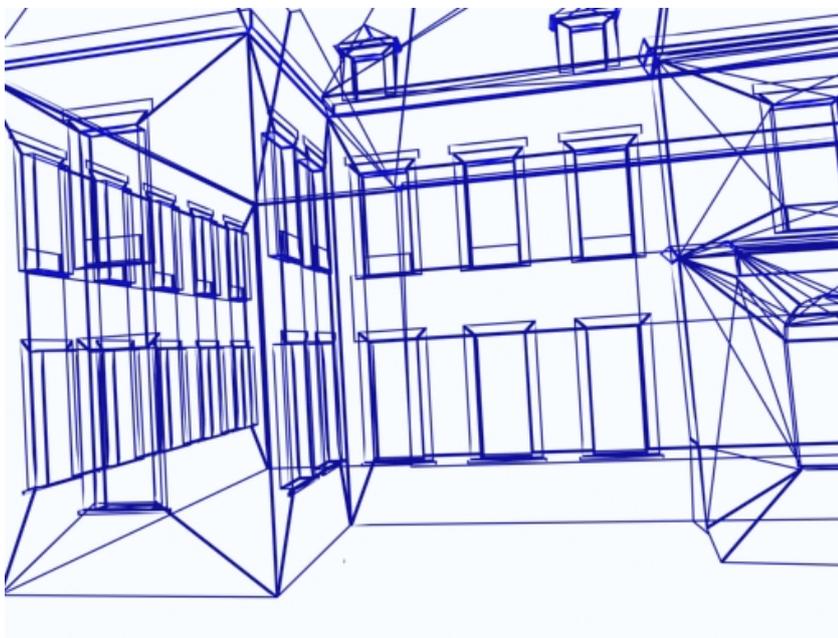


FIG. 5.16: Modèle représenté en fil de fer.



FIG. 5.17: Modèle texturé.

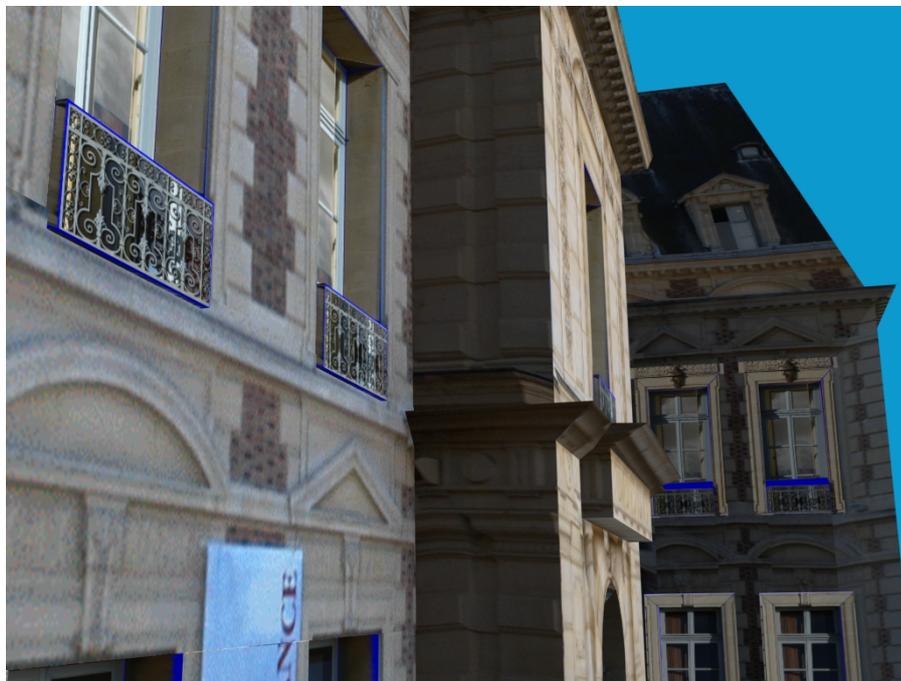


FIG. 5.18: Modèle texturé.



FIG. 5.19: Modèle texturé.



FIG. 5.20: Modèle texturé.

5.4 Autres exemples

Nous avons appliqué la même démarche afin de reconstruire deux bâtiments plus simples. Ces exemples montrent qu'il est possible, par cette approche, d'effectuer des reconstructions simples en un temps relativement court. Les détails concernant ces deux exemples sont présentés ci-dessous :

- **une maison bretonne** : le modèle final est montré dans la figure 5.21. Les positions de prise de vue sont décrites sur la figure 5.1. L'appareil photographie était un numérique grand public et les images ont une résolution de 1024×728 pixels. Comme le montre le modèle 3D représenté en fil de fer, le modèle est très rudimentaire. Moins d'une heure a été nécessaire pour obtenir ce résultat.
- **la chapelle de Dirac** : le modèle final est montré dans la figure 5.22. Toutes les photographies ont été prise du même côté du bâtiment. L'appareil utilisé est le même que celui employé lors de la reconstruction de la maison bretonne (numérique grand public). Ce modèle présente une structure complexe et le peu d'images disponibles rendait difficile la définition de point homologue. Malgré tout, notre méthode de reconstruction est parvenu à reconstruire de modèle à partir de cinq vues. La forme cylindrique à l'arrière de la chapelle n'a pas été reconstruite car une seule image permettait de l'observé le coté droit étant dissimulé par un bâtiment.

Ces deux exemples montrent qu'avec peu de vues et peu de sommets remarquables dans chaque image nous sommes capable d'obtenir des modèles 3D convenables en des temps de reconstruction faibles. Ceci démontre l'importance considérable des informations apportées par l'utilisateur sous forme de contraintes et la possibilité de fusionner dans une approche interactive des informations saisies en localisant des primitives dans les images et des informations a priori sur la structure 3D observée.

Exemple sur une maison bretonne

Nombre d'images	Nombre de paramètres	Nombre de mesures	Nombre de sommets	RMS final	Temps de reconstruction	Longueurs focales
5	53	102	47	4.72 pixels	1h10	[820 à 1289] pixels focales variables

TAB. 5.1: Données caractéristiques de la reconstruction de la maison bretonne.



FIG. 5.21: Reconsruction de la maison bretonne. En haut, les images utilisées, au centre, le modèle obtenu rendu en fil de fer et en bas, le modèle final texturé. Dans les images du haut, seul les segments pour lesquels les deux extrémités ont été localisées manuellement dans les images sont représentés.

Exemple sur une église romane

Nombre d'images	Nombre de paramètres	Nombre de mesure	Nombre de sommets	RMS final	Temps de reconstruction	Longueurs focales
7	100	154	67	4.74 pixels	1h30	975 pixels

TAB. 5.2: Données caractéristiques de la reconstruction d'une partie de l'église de Dirac.

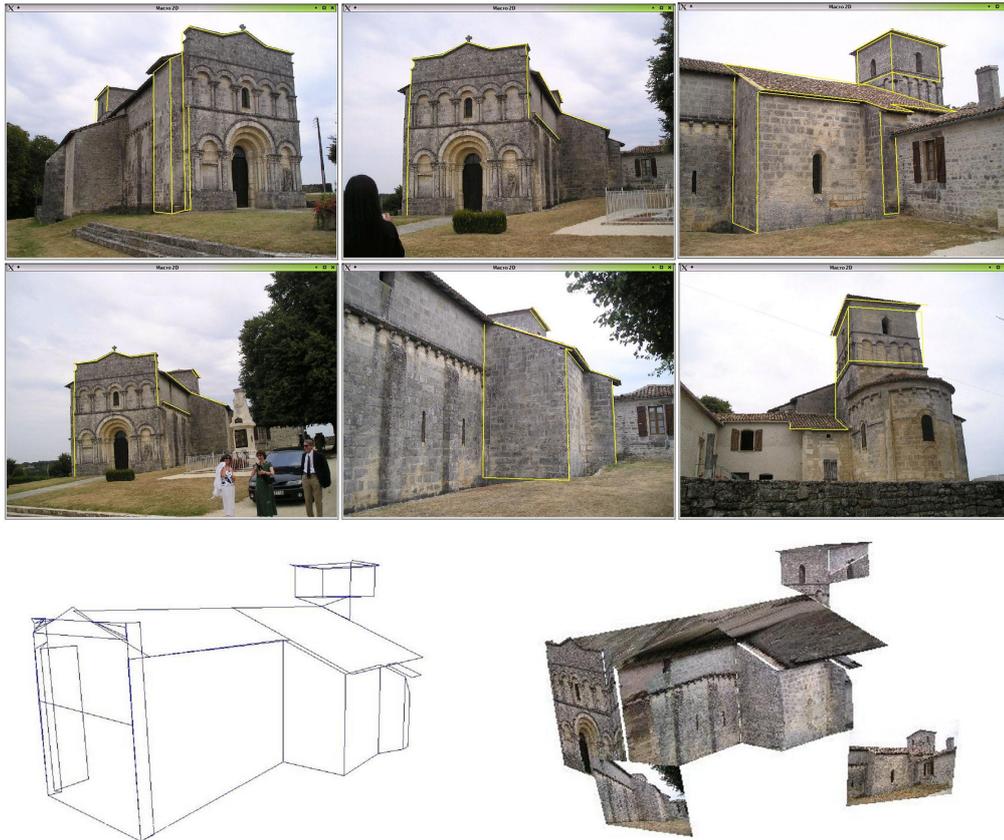


FIG. 5.22: La reconstruction de l'église de Dirac. En haut, toutes les images utilisées, en bas, le modèle obtenu rendu en fil de fer et le modèle final texturé. Dans les images du haut, seuls les segments pour lesquels les deux extrémités ont été localisées manuellement dans les images sont représentés. On remarque que le recalage du clocher est imparfait sur la dernière image, ceci traduit la saisie de contraintes aberrantes pour modéliser la partie de la toiture visible uniquement dans cette image.

5.5 Conclusion

Nous venons de présenter trois exemples de modélisation de bâtiments à partir d'images. Ces différents exemples nous ont permis de montrer que notre approche permettait de reconstruire des objets de diverses complexités sans calibrage préalable des caméras. Ces démonstrations nous ont également permis d'identifier des points difficiles et, par conséquent, des voies d'améliorations de notre méthode :

1. le premier point délicat est la localisation des primitives dans les images. Dans le cadre de nos exemples, ce problème vient tout d'abord de l'absence de vue aérienne qui rend difficile la localisation des extrémités des toits. D'autre part, la présence de nombreux détails et de nombreuses irrégularités dans les bâtiments complique l'identification de certains sommets par un enchevêtrement de structures superposées qui masque la position du coin recherchée (pierres d'angle,...),
2. la deuxième limitation tient à la nécessaire expertise de l'utilisateur pour déterminer si les informations saisies sont suffisantes pour effectuer la reconstruction,
3. le troisième élément délicat est la définition des surfaces du modèle. La définition manuelle des faces du modèle est une tâche fastidieuse et coûteuse en temps. La mise au point d'un processus automatique permettant de définir la surface et d'extraire la texture du modèle 3D à partir des images accroîtraient grandement l'efficacité et la convivialité du processus de reconstruction,
4. le dernier point à trait aux saisies de contraintes inadéquates. Comme nous l'avons vu, il arrive que l'utilisateur définisse des contraintes erronées (cf. figure 5.13 qui vont réduire la qualité de la reconstruction. Malheureusement, l'identification d'une contrainte erronée est difficile car l'erreur provoquée par cette contrainte est distribuée dans tous les éléments modélisés de la scène (modèle 3D et caméras). De plus, dans l'éventualité où cette erreur est identifiée, la suppression de cette contrainte aberrante peut conduire à un système non estimable (cf. deuxième limitation).

Malgré tous ces points à améliorer, la solution proposée répond à notre objectif initial qui était de définir une méthode interactive permettant de reconstruire des modèles structurés en intégrant simultanément des contraintes géométriques et des informations saisies à partir des images. Les exemples présentés dans ce chapitre ont montré qu'à partir de quelques photographies prises en conditions extérieures nous étions en mesure de reconstruire des modèles 3D de bâtiments. De plus, les exemples ont montré la cohérence de l'ensemble modélisation par contrainte et algorithme CACHE.POS et sa capacité à traiter des cas difficiles. Enfin, comme le montre les différents exemples traités dans ce chapitre, notre approche stratifiée permet de reconstruire des modèles complexes ou des modèles simples de bâtiments en présentant des temps de modélisation adaptés au niveau de détail recherché.

Chapitre 6

Perspectives

Sommaire

6.1	Introduction	153
6.2	Utiliser les informations géométriques	154
6.2.1	Le problème est-il suffisamment contraint ?	154
6.2.2	Initialisation pertinente du modèle 3D.	156
6.2.3	Synthèse sur l'utilisation de la géométrie	157
6.3	Utiliser des appariements de points homologues	157
6.3.1	Apparier des points homologues entre les images	157
6.3.2	Amélioration simultanée du calibrage et du modèle 3D	163
6.3.3	Faciliter l'opération de fusion de modèles	165
6.4	Exploiter la texture présente dans les images	165
6.4.1	Détecter les incohérences et les occultations	166
6.4.2	Automatiser le passage du modèle ponctuel au modèle surfacique	166
6.4.3	Extraire efficacement la texture du modèle	167
6.5	Conclusion	168

6.1 Introduction

Ce dernier chapitre dresse un bilan de toutes les difficultés restant à surmonter pour parvenir à un outil de reconstruction simplifiant au maximum le travail de l'utilisateur. Nous proposons différentes voies de recherche visant à améliorer différentes étapes du processus de reconstruction présenté dans ce document. La première partie concerne l'exploitation des informations géométriques issues de la localisation des sommets du modèle 3D dans les images. La deuxième partie aborde l'exploitation possible d'appariements denses de points homologues obtenus automatiquement. Enfin, la troisième partie aborde la question de l'exploitation de la texture présente dans les images en vue d'une détection des zones mal modélisées, d'une facétisation automatique et d'une extraction de texture de haute résolution.

6.2 Utiliser les informations géométriques

Nous avons proposé une méthode de reconstruction basée sur deux éléments : la localisation des points 3D à partir de leur projection dans les images et la définition d'un modèle contraint. L'exploitation de ces informations à caractère géométrique nous a permis de reconstruire un modèle 3D et d'estimer les positions, les orientations et les longueurs focales des capteurs. Deux problèmes subsistent toutefois. Tout d'abord, une fois que l'utilisateur a défini un modèle et a saisi des informations dans les images, le système est incapable de déterminer a priori si ces informations sont suffisantes pour permettre la reconstruction. D'autre part, l'initialisation du modèle 3D demeure un point problématique (cf. chapitre 2).

6.2.1 Le problème est-il suffisamment contraint ?

Considérons un objet, décrit par un modèle contraint, observé par plusieurs caméras. A quelles conditions peut-on exécuter l'algorithme d'estimation des paramètres associés à la scène en étant sûr d'avoir suffisamment de mesures pour suffisamment contraindre les paramètres. Nous divisons l'analyse de ce problème en deux parties correspondant respectivement au modèle et aux caméras.

Y-a-t-il assez de mesures pour évaluer le modèle 3D ?

Considérons que les caméras sont totalement calibrées (pose et paramètres intrinsèques connus), si le modèle 3D est constitué de points libres de toutes contraintes. Il suffit que chaque point soit vu dans deux images différentes pour permettre une reconstruction du modèle par triangulation. Au contraire, lorsque le modèle est décrit au moyen de contraintes, il est possible de reconstruire des points non observés dans les images mais positionnés dans l'espace grâce aux contraintes qui les lient au modèle.

Par exemple, imaginons que l'on souhaite reconstruire une sphère. Un sommet A doté de trois degrés de liberté (X, Y, Z) représentant le centre de la sphère peut être créé. Un second sommet B doté également de trois degrés de liberté (X, Y, Z) représentant un point sur la surface de cette sphère peut être ajouté. Les points C_i ajoutés par la suite sont contraints à être à une distance AB du sommet A et sont caractérisés par deux angles θ et φ (cf. chapitre 4). Si les points B et C_i sont observés dans deux images chacun, la sphère peut être reconstruite bien que le point A ne soit pas observable, sa position est estimable par le biais des informations saisies par rapport aux autres points. Le point A est un point virtuel constituant, en quelque sorte, une réserve non observable de degrés de liberté caractérisant le modèle. Comment déterminer si les informations globalement disponibles sont en nombre suffisant pour estimer ce modèle ?

La première idée est d'effectuer un comptage des degrés de liberté et des mesures disponibles. En pratique, un point observé dans une image fournit deux mesures indépendantes et un point observé dans deux images fournit trois mesures indépendantes. Une condition nécessaire mais pas suffisante est que la somme du nombre de mesures associées aux points soit supérieure ou égale au nombre de paramètres libres caractérisant le modèle. En pratique, ce critère n'est pas suffisant car certaines situations vérifient cette

propriétés sans être estimable. Par exemple, prenons le cas décrit dans le tableau 6.1, le point C n'est pas contraint et peut glisser librement selon l'axe AD alors que le nombre de mesure est égale au nombre de paramètres libres. Une analyse plus fine du rôle de chaque contrainte est nécessaire pour parvenir à lever cette difficulté.

Point	Contrainte	degrés de liberté	Nb. de sélection	Mesures
A	nulle	3	2	3
B	nulle	3	2	3
C	$\vec{AB} \cdot \vec{BC} = 0$	2	0	0
D	$\vec{CD} = \lambda \cdot \vec{BC}$	1	2	3
Total		9	6	9

TAB. 6.1: Exemple de modèle non estimable.

La seconde idée envisagée a été d'employer une approche identique à celle utilisé par Grossman et al. ainsi que Wilczkowiak et al. (cf. chapitre 3) consistant en l'étude des propriétés du noyau associé à un système linéaire. Dans notre cas, nous ne disposons pas de contraintes linéaires par contre, à chaque itération, CACHE_POS (enfin la méthode de Levenberg-Marquardt) résout un système linéaire de la forme $AX = B$ (cf. chapitre 1). Cette équation est une linéarisation par rapport aux paramètres 3D et aux paramètres caméras. Si l'on considère les caméras fixes, l'équation $AX = B$ traduit une approximation linéaire des relations entre le modèle 3D et les mesures disponibles dans les images. A l'image de ce que font Wilczkowiak et al. dans [Wilczkowiak 03a], nous décomposons A par SVD et nous identifions les valeurs singulières nulles ce qui permet d'identifier les vecteurs propres formant une base du noyau. Le système est contraint, c'est à dire qu'il existe une solution unique au système $AX = B$, si et seulement si tout les vecteurs du noyau sont nuls. Dans la pratique, le bruit rend difficile l'identification des valeurs propres nulles ce qui limite l'usage de cette approche. De plus, au cours des premières itérations les caméras et le modèle 3D sont complètement erronés, cela conduit à calculer des matrices jacobiennes et quasi-hessiennes pour des jeux de paramètres très éloignés de la solution rendant cette méthode inexploitable pour détecter a priori la présence de paramètres non contraints.

Déterminer a priori si les mesures et les contraintes fournies suffisent à estimer le modèle 3D n'est pas un problème simple et la solution ne nous semble pas évidente. Qu'en est-il pour les paramètres caractérisant les capteurs ?

Y-a-t-il assez de mesures pour évaluer les capteurs ?

Il est plus simple de déterminer si une caméra est estimable ou pas. Considérons un modèle 3D connu, l'estimation des paramètres de la caméra observant ce modèle est possible si suffisamment de points sont observés. Dans un cas non dégénéré (cf. [Sturm 97]), chaque point observé dans l'image fourni deux mesures (position en x et position en y) contraignant de ce fait deux paramètres du modèle du capteur. Par conséquent, le capteur étant modélisé par six paramètres d'attitude et k paramètres internes, l'estimation est généralement possible si au moins $(6 + k)/2$ points sont visibles.

En synthèse, nous n'avons pas trouvé une méthode fiable permettant de déterminer si les informations saisies sont suffisantes pour permettre l'estimation de la scène. En effet, nous pouvons déterminer quels sont les caméras insuffisamment contraintes mais il est toujours impossible de déterminer a priori si le modèle 3D l'est également. Ce point délicat est pourtant un élément essentiel en vue d'une amélioration du processus de reconstruction et de l'ergonomie d'utilisation de notre méthode. En effet, à partir du moment où nous serons en mesure de déterminer si le modèle est estimable (ou quel sous partie du modèle est estimable), nous pourrons estimer automatiquement les paramètres contraints au cours de la saisie des informations et il sera alors possible d'indiquer à l'utilisateur les éléments d'information manquants. Cette démarche allégera le rôle de l'utilisateur qui n'aura plus besoin de lancer manuellement l'ordre de reconstruction et permettra ainsi de réduire le niveau d'expertise requis pour utiliser le logiciel.

6.2.2 Initialisation pertinente du modèle 3D.

Le second problème évoqué concerne l'initialisation des paramètres internes des caméras et du modèle 3D avant l'exécution de `CACHE_POS`. La solution proposée jusqu'à présent consiste à tenter de converger à partir d'une situation initiale générée aléatoirement. Cette approche est envisageable grâce au taux de succès observé avec la méthode `CACHE_POS`, une telle approche est illusoire avec une méthode d'ajustement de faisceaux classique. Une méthode d'initialisation pertinente permettrait de converger immédiatement à la solution sans avoir à effectuer plusieurs tentatives.

La première difficulté est de caractériser les propriétés que doit posséder une situation pour permettre une convergence réussie. Lors des différents essais réalisés, nous avons constaté que pour une situation donnée, l'initialisation du modèle 3D conditionne le succès ou l'échec de la méthode. Notre conclusion est donc que l'organisation des sommets 3D les uns par rapport aux autres est le facteur prépondérant. A partir de ce constat, nous devons initialiser efficacement les paramètres du modèle 3D.

Une première idée est d'identifier des sous-ensemble de points homologues observés dans les mêmes images et de les utiliser afin d'effectuer une première reconstruction utilisant les approches classiques présentées dans le chapitre 1. La reconstruction ainsi obtenue servira de base à la reconstruction complète du modèle contraint. Dans la pratique, une telle approche est difficilement généralisable car il y a souvent peu de points observés dans plusieurs images. De plus, même si une reconstruction basée sur les seules positions de certains points dans plusieurs images est obtenue, le résultat se présentera sous la forme d'un nuage de points 3D libres. Déterminer le modèle 3D contraint le plus proche de ce nuage de points libres et ajouter de nouvelles images peut également mener à de nouveaux problèmes de convergence.

Une seconde idée consiste à conserver partiellement l'approche itérative. Si le problème précédent consistant à déterminer si une scène est suffisamment contrainte ou pas est résolu, il est possible de rechercher le plus petit modèle estimable (le modèle contenant le moins de paramètres possibles). Ce modèle minimal sera moins complexe à estimer par une approche basée sur une succession d'initialisations aléatoires. Une fois une solution acceptable obtenue, il sera possible de chercher un nouveau modèle estimable un peu plus grand et de s'appuyer sur cette première reconstruction pour n'estimer que

les nouveaux paramètres introduits. En itérant cette démarche, une initialisation globale du modèle complexe est accessible.

6.2.3 Synthèse sur l'utilisation de la géométrie

Nous avons soulevé deux points durs qui sont, d'une part, l'incapacité à déterminer a priori si une scène est estimable à partir des observations dans les images et des contraintes définies pour caractérisées le modèle 3D et, d'autre part, la nécessité de trouver une méthode d'initialisation plus systématique du modèle 3D afin d'obtenir une meilleure efficacité. La dernière idée exposée établit un lien entre ces deux difficultés. La résolution de ces problèmes permettrait d'obtenir une méthode plus rapide (plus de générations successives de situations initiales) et faciliterait l'utilisation de notre logiciel en permettant de guider l'utilisateur au cours de la reconstruction (en indiquant les mesures manquantes).

6.3 Utiliser des appariements de points homologues

Parmi les informations disponibles, nous n'avons exploité que la position des sommets du modèle dans les images mais nous n'avons pas utilisé l'information contenue dans les images. Après une reconstruction basée sur des informations purement géométriques (contraintes et positions des sommets dans les images), nous nous sommes appuyé sur ce premier résultat afin d'obtenir un ensemble d'appariements de points homologues entre les différentes images. Cet ensemble d'appariements peut servir de base à une amélioration de la précision de la reconstruction et à une (semi-)automatisation de l'opération de fusion de modèles. Dans un premier temps, nous allons présenter la technique d'appariements de points que nous avons utilisée. Par la suite, nous présenterons une méthode d'optimisation permettant d'améliorer la précision globale de la reconstruction (capteurs et modèle 3D) à partir des points appariés. Enfin, nous évoquerons comment ces informations peuvent permettre d'améliorer le processus de fusion de modèles contraints.

6.3.1 Apparier des points homologues entre les images

La première étape consiste à appairer des points d'intérêt entre les différentes images de la scène. Nous avons le choix d'extraire les points d'intérêt dans toutes les images puis d'établir une correspondance entre les points identifiés ou bien de sélectionner un point d'intérêt dans une image et de rechercher son correspondant dans les autres images. La première stratégie implique que les points d'intérêts soient les mêmes dans les différentes images. Cette éventualité est d'autant moins grande que les différences de points de vues sont importantes d'une image sur l'autre, c'est pourquoi nous avons choisi la seconde approche.

6.3.1.1 *Extraction de points d'intérêt*

L'objectif de l'algorithme que nous allons présenter est d'apparier des points homologues entre plusieurs images en connaissant au préalable un calibrage approximatif des caméras et le modèle 3D surfacé de l'objet observé. Considérons un ensemble de k

images. Nous sélectionnons l'image i dans laquelle nous effectuons une détection de points d'intérêt (détecteur SUSAN [Smith 95], détecteur de Harris et Stephen [Harris]). L'étape suivante consiste à rechercher dans chacune des autres images les points correspondant à ces points d'intérêt. Considérons un point d'intérêt particulier que nous noterons p_i^i (1-ième point dans la i -ième image). Nous recherchons sa position p_i^j dans l'image j . En s'appuyant sur la reconstruction préalablement effectuée, nous pouvons projeter p_i^i sur le modèle 3D et projeter le point 3D obtenu sur la caméra j . Cette étape nous permet de déterminer si ce point est visible dans l'image j et, le cas échéant, une position initiale de p_i^j . Ce processus est représenté dans la figure 6.1.

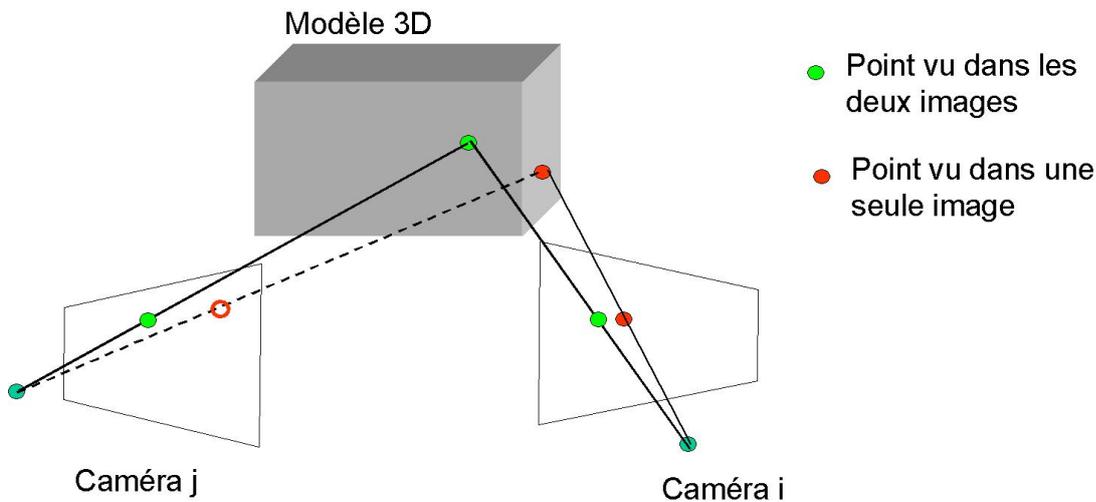


FIG. 6.1: Test de visibilité et initialisation de la position de la projection dans l'image j .

6.3.1.2 Initialisation de l'homographie

Une fois que l'on connaît un appariement initial (p_i^i, p_i^j). Nous cherchons à affiner le positionnement de p_i^j qui pour l'instant correspond rigoureusement à la reconstruction initiale. Afin d'améliorer ce positionnement, nous allons utiliser l'information contenue dans les images i et j . Nous considérons le point p_i^i comme le point de référence. Nous pouvons définir une fenêtre carré de côté c autour de ce point de référence et stocker la texture qu'il contient. Afin de pouvoir manipuler cette donnée, nous construisons un vecteur \vec{V}_{ref} de $c \times c$ composantes représentant le niveau de gris de chacun des pixels de la zone définie. Afin de mieux résister aux variations d'aspect provoquées par les changements de point de vue, ce vecteur est centré et normé. Nous faisons l'hypothèse que la surface de l'objet correspondant au carré entourant le point de référence est un plan. Par conséquent, la transformation passant du carré de l'image i à la zone correspondante dans l'image j est une homographie notée $H_{i \rightarrow j}$. Connaissant le modèle 3D, nous avons une estimation de $H_{i \rightarrow j}$. Pour une matrice $H_{i \rightarrow j}$ donnée, nous calculons un vecteur \vec{V}_{eval} (*eval* pour évaluation) de dimension $c \times c$ dont les composantes sont les niveaux de gris associés à la zone de l'image j correspondant au carré de l'image de référence. Cette relation entre les voisinages des points p_i^i et p_i^j est schématisée dans la figure 6.2.

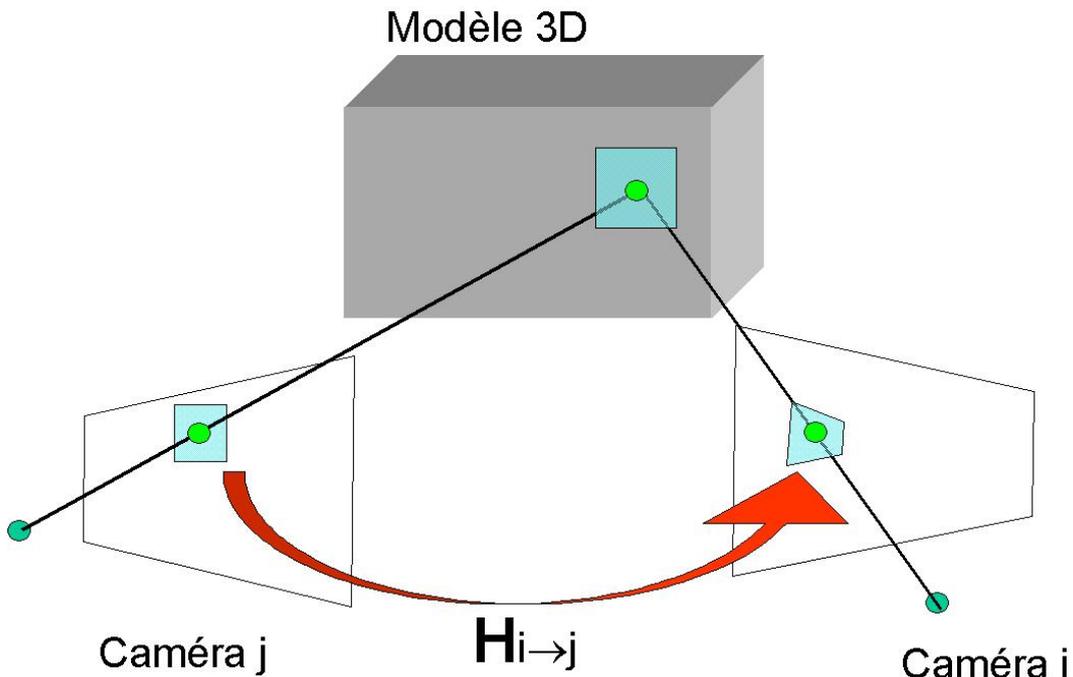


FIG. 6.2: Relation homographique entre la zone de référence et la zone d'évaluation.

Notre objectif est de calculer la position de p_i^j telle que les voisinages de p_i^i et p_i^j se ressemblent le plus possible. La ressemblance ϕ est évaluée par la distance existant entre V_{ref} et V_{eval} . En fait ϕ dépend de $H_{i \rightarrow j}$ et, l'objectif est de minimiser la fonction :

$$\phi(H_{i \rightarrow j}) = \|V_{ref}^{\vec{}} - V_{eval}^{\vec{}}\| \quad (6.1)$$

Compte tenu des erreurs existantes dans la reconstruction initiale ayant permis de calculer $H_{i \rightarrow j}$, il est probable que $\phi(H_{i \rightarrow j})$ soit relativement éloignée de son minimum (ceci est en particulier dû au caractère très local d'une corrélation). Par conséquent, avant d'effectuer une recherche fine du minimum de ϕ , nous effectuons une recherche grossière sur un voisinage assez large du minimum de ϕ . Cette recherche est effectuée au pixel près et consiste à balayer tout le voisinage de p_i^j en ajoutant une translation $\Delta\vec{T} = (T_x T_y 1)^T$ à la matrice $H_{i \rightarrow j}$. Une fois que $\Delta\vec{T}$ correspondant à $\phi(H_{i \rightarrow j} + \Delta\vec{T})$ minimum a été trouvée, la nouvelle valeur initiale pour l'homographie est $H_{i \rightarrow j}^0 = H_{i \rightarrow j} + [0_{3 \times 2} | \Delta\vec{T}]$.

6.3.1.3 Estimation non-linéaire de la position du point de mesure

Une fois $H_{i \rightarrow j}^0$ connu, nous faisons appel à une procédure de type Levenberg-Marquardt afin de rechercher plus précisément le minimum de ϕ . Nous approximons $H_{i \rightarrow j}$ par une transformation affine.

Nous approximons la transformation $H_{i \rightarrow j}$ par une affinité $A_{i \rightarrow j}$. Nous recherchons les six éléments de la matrice affine minimisant $\phi(A_{i \rightarrow j})$. Un exemple est donné à la figure

6.4. Nous disposons alors d'une matrice affine de la forme (le dernier élément est fixé à un) :

$$A_{i \rightarrow j} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \quad (6.2)$$

Une fois ϕ minimisée, nous pouvons calculer (sur la base de dérivées numériques) la matrice de covariance Ω associée à ce vecteur de paramètres (a,b,c,d,e,f). Les paramètres c et f correspondent à la translation entre l'image i et l'image j , nous extrayons la matrice de covariance Ω_{cf} qui leur est associée. Les valeurs propres de Ω_{cf} sont les variances associées à c et f , leur inverse traduit l'incertitude sur la localisation du point. Nous ordonnons ces variances et nous les notons désormais σ_{min}^2 et σ_{max}^2 .

En fin de processus, certains appariements sont erronés. Une technique fréquemment employée pour éliminer ces points aberrants est d'effectuer une sélection robuste au moyen d'un algorithme de type RANSAC utilisant la matrice fondamentale, le tenseur trifocal ou encore les contraintes épipolaires. Cette approche prend comme hypothèse le respect du modèle projectif du capteur. Dans notre cas, nous n'avons pas encore corrigé les distorsions présentes dans les images, l'utilisation d'une approche basée sur le modèle projectif risque donc d'introduire un biais dans notre sélection.

Pour éliminer cette difficulté nous avons utilisée les covariances σ_{min} et σ_{max} . Les appariements présentant des valeurs trop grandes pour σ_{max} ou σ_{min} sont éliminés (par exemple, toute valeur de σ supérieur à 1). L'étude du rapport $\frac{\sigma_{min}}{\sigma_{max}}$ nous renseigne sur la nature de l'appariement (cf. figure 6.3) :

- $\frac{\sigma_{min}}{\sigma_{max}} \approx 1$: appariement localisé de façon équivalente dans toutes les directions. C'est la situation la plus favorable a une bonne localisation car cela correspond à un coin net,
- $\frac{\sigma_{min}}{\sigma_{max}} \ll 1$: existence d'une isotropie impliquant le risque d'une détection décalé dans la direction présentant la plus grande incertitude. C'est typiquement le cas des contours rectilignes.

En fixant un seuil sur le rapport $\frac{\sigma_{min}}{\sigma_{max}}$, nous parvenons à sélectionner parmi tous les appariements possibles ceux qui semblent être les plus prometteurs. La figure 6.5 montre le résultat obtenu sur la séquence de l'avancée du château de Sceaux. Afin de visualiser le résultat nous avons triangulé la position des points 3D correspondants aux appariements à partir du calibrage obtenu à la fin de la reconstruction par contrainte.

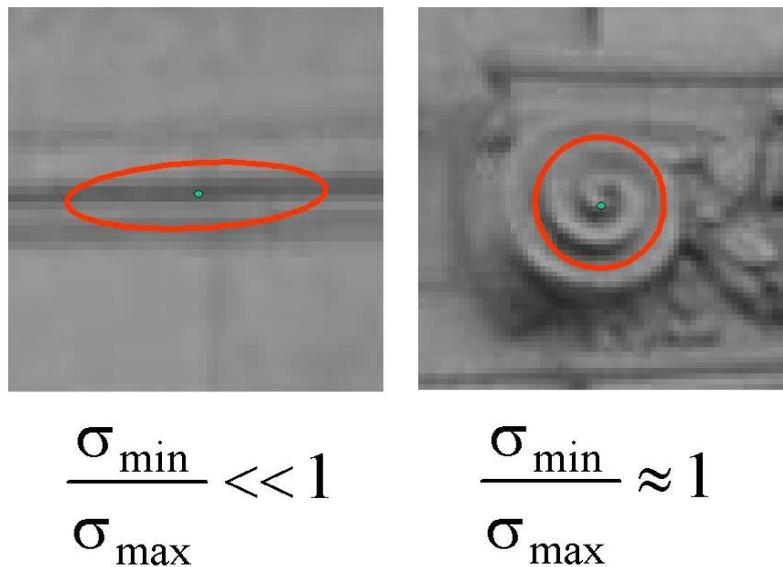


FIG. 6.3: Relation entre l'aspect et la covariance (Les ellipses en rouge représentent l'incertitude sur la localisation du point).

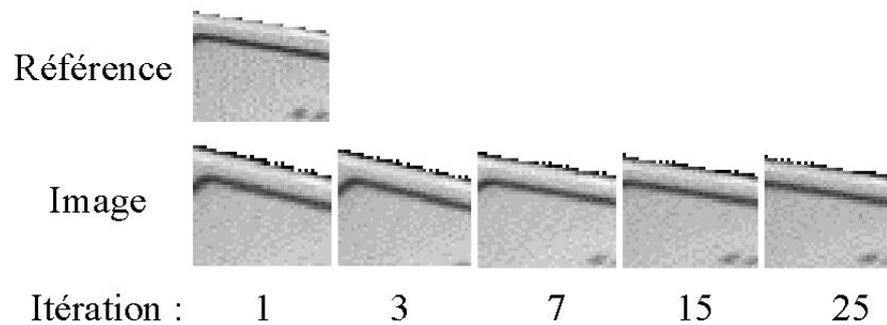


FIG. 6.4: Evolution de la texture du 'patch' de mesure en fonction du nombre d'itération.

La méthode d'appariement que nous avons proposée consiste en plusieurs étapes résumées dans l'algorithme suivant.

1. Extraction des points d'intérêt dans les images
2. Appariements grossiers des points entre les différentes images
3. Recherche fine des appariements par une méthode non-linéaire
4. Sélection des appariements en fonction des covariances sur la localisation

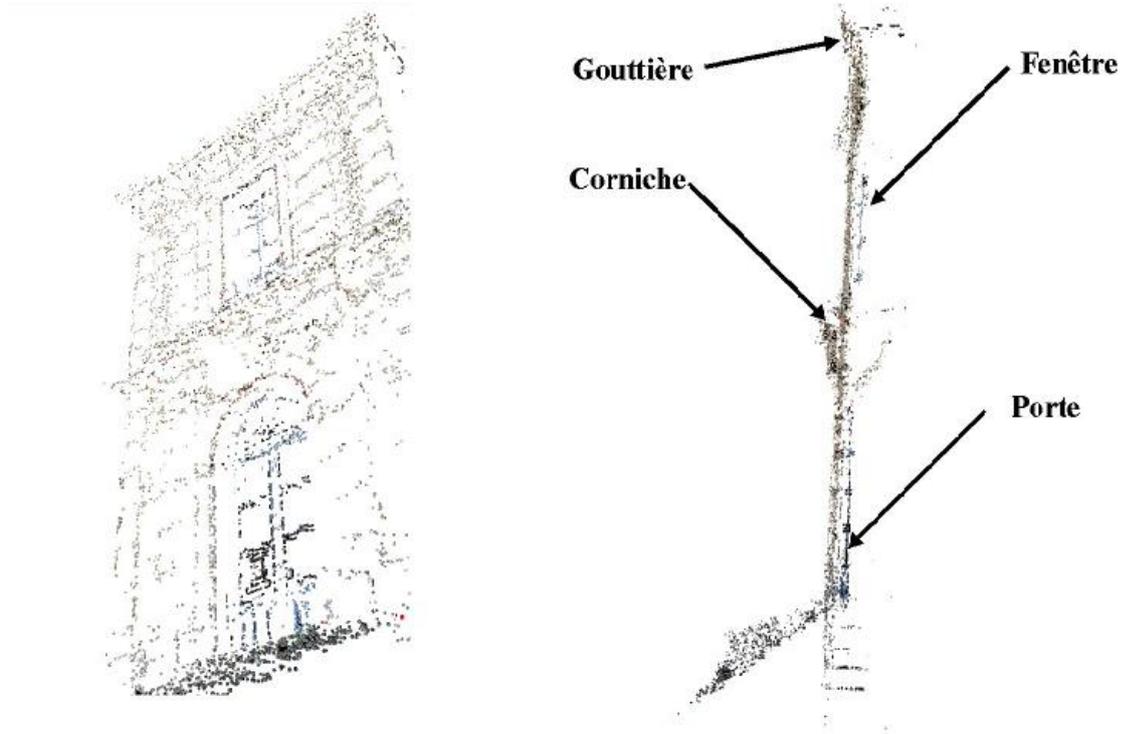


FIG. 6.5: Deux vues (face et profil) du nuage de points 3D extraits par triangulation à partir des appariements de points homologues dans les images. Les caméras n'ont pas été recalculés, la netteté des plans principaux de la façade montre que le calibrage fourni par la méthode exposée dans ce document est cohérent. Les appariements de points homologues dont le ratio $\frac{\sigma_{min}}{\sigma_{max}} < 0.3$ ont été éliminés.

Cette méthode d'appariements de points homologues propose un critère, uniquement basé sur les informations présentes dans les images, pour trier les données et supprimer les 'outlier' potentiels. Nous avons approximer la relation homographique par une affinité (l'extension à une homographie est simple seul le calcul des incertitudes est un peu plus compliqué du fait du facteur d'échelle à prendre en compte). Nous pensons qu'une intégration de l'incertitude calculées lors de l'appariement dans des algorithmes d'ajustement de faisceaux, de triangulation, de calcul de matrices fondamentales ou de tenseurs trifocaux (...) peut permettre de prendre en compte la qualité de la texture ayant guider l'appariement et de ne pas imposer un modèle pour rejeter les paires erronées (au contraire des approches de types RANSAC). Dans la suite de ce paragraphe, nous présentons une approche susceptible d'utiliser ces incertitudes afin d'améliorer la précision des reconstructions obtenues par notre méthode.

6.3.2 Amélioration simultanée du calibrage et du modèle 3D

Une fois que ces appariements entre points homologues sont disponibles, nous disposons de nouvelles informations pour améliorer la reconstruction de la scène précédemment obtenue. Deux éléments doivent être améliorés :

- le calibrage des capteurs (introduction de nouveaux paramètres : distorsions, ...),
- le modèle 3D.

A partir de l'ensemble des appariements de points homologues dont nous disposons, nous avons envisagé différentes approches possibles. Tout d'abord, le processus d'appariement que nous avons employé ne nous assure pas de l'absence d'appariements erronés. Deux possibilités s'offrent à nous, effectuer une sélection robuste au moyen d'une méthode de type RANSAC sur la base du respect des propriétés inférées par la matrice fondamentale existant entre chaque paire d'images ou le tenseur trifocal estimé sur les triplets d'images, ou bien utiliser un M-estimateur afin de supprimer en cours d'estimation les mauvais candidats. Au moment où nous écrivons ces lignes aucun choix définitif n'a été effectué. Toutefois, les approches basées sur des méthodes de types RANSAC sélectionnent les paires de points homologues en fonction de leur compatibilité avec un modèle projectif parfait. En présence de distorsions importantes, cela peut conduire par exemple à un rejet systématique des points homologues périphériques qui, du fait de leur éloignement du point principal de l'image, présenteront une forte erreur de localisation en comparaison à leur position estimée par un modèle projectif pur. Les approches par M-estimateur permettent une prise en compte plus souple de ces questions et peuvent plus facilement tenir compte de modèles plus complexes.

Pour améliorer l'estimation globale de la scène, nous pouvons utiliser deux approches. La première consiste à calibrer finement les capteurs au moyen de l'ensemble d'appariements, puis de définir le modèle 3D correspondant le mieux à ces conditions de prise de vue. Le principal défaut de cette approche est de séparer les variables associées aux modèles de celles associées aux prises de vue ce qui constitue une approche de type intersection-résection dont la convergence est difficile à maîtriser.

La seconde méthode consiste à estimer tous les paramètres simultanément. Nous proposons trois critères à minimiser qui s'appuient pour l'un d'entre eux sur des distances tridimensionnelles et pour les deux autres sur des distances évaluées dans les images. Les deux premiers critères font appel à une fonction de triangulation nommée *Trian* vérifiant (P_i un point vu dans n images, Ω : matrice d'incertitude sur la localisation d'un point homologue) :

$$P_i = \text{Trian}([\Gamma_1, \dots, \Gamma_n], [p_1, \Omega_1, \dots, p_n, \Omega_n]) \quad (6.3)$$

La fonction *Trian* effectue une triangulation mais prend en compte l'incertitude de localisation de chaque point homologue déterminée lors de la phase d'appariements et s'exprimant sous la forme d'une matrice d'incertitude sur la localisation.

1. Critère I :

Prenons une paire de points appariés et admettons qu'ils correspondent à un point 3D à la surface du modèle 3D reconstruit. Si la modélisation est parfaite, la distance entre ce point 3D et la surface la plus proche du modèle 3D est nulle. C'est ce critère de distance que nous proposons de minimiser. En notant $d(P_{3D}, \Pi(\phi_{3D}))$ la distance entre un point 3D P_{3D} et la surface Π du modèle 3, nous avons à minimiser le critère Θ_1 (M : M-estimateur) :

$$\Theta_1(\phi_{3D}, \phi_{cam}) = \sum_{app} M([d(Trian_{app}(\phi_{cam}), \Pi(\phi_{3D}))]) \quad (6.4)$$

2. Critère II :

Le deuxième critère consiste à trianguler la position du point 3D P_{3D} correspondant à un ensemble de points homologues puis à rechercher le point 3D $P_{3D \in \Pi}$ le plus proche de P_{3D} appartenant à la surface du modèle. Une fois ce point déterminé, la mesure à minimiser est alors la distance entre les points homologues et la projection de $P_{3D \in \Pi}$ dans chacune des images. En notant $Prox$ la fonction calculant la position du point 3D $P_{3D \in \Pi}$ pour un point 3D P_{3D} donné (M : M-estimateur, i : indice des images, k : indice des points) :

$$\Theta_2(\phi_{3D}, \phi_{cam}) = \sum_{app} \sum_{image} M(p_{i,k} - \Gamma_i [\underbrace{Prox(Trian_{app}(\phi_{cam}), \Pi(\phi_{3D}))}_{P_{3D \in \Pi}}]) \quad (6.5)$$

3. Critère III :

Le troisième critère consiste à choisir une projection de référence d'un point 3D dans un image puis de rechercher par lancer de rayon le point 3D $P_{3D \in \Pi}$ à la surface du modèle 3D lui correspondant. Une fois ce point 3D déterminé, la mesure à minimiser est alors la distance entre les points homologues et la projection de $P_{3D \in \Pi}$ dans chacune des images. En notant $Clip$ la fonction calculant la position du point 3D $P_{3D \in \Pi}$ à partir d'une projection 2D donnée, le critère à minimiser s'écrit (M : M-estimateur, i : indice des images, k : indice des points, app_{ref} : projection de référence) :

$$\Theta_3(\phi_{3D}, \phi_{cam}) = \sum_{app} \sum_{image} M(p_{i,k} - \Gamma_i [\underbrace{Clip(app_{ref}, \phi_{cam,i}, \Pi(\phi_{3D}))}_{P_{3D \in \Pi}}]) \quad (6.6)$$

Les fonctions Θ ne dépendent que de ϕ_{cam} et ϕ_{3D} . Les valeurs initiales sont directement issues de la reconstruction obtenue précédemment et la minimisation est effectuée au moyen de l'algorithme de Levenberg-Marquardt. Ces méthodes n'ont pas encore été évaluées mais présentent, sur le papier, l'avantage d'effectuer une estimation simultanée des paramètres des capteurs et du modèle 3D. Il reste à voir les problèmes de convergence et la qualité des résultats ainsi obtenus.

6.3.3 Faciliter l'opération de fusion de modèles

Une seconde application possible basée sur des appariements est de faciliter l'opération de fusion de modèles 3D. Jusqu'à présent la méthode utilisée consiste à définir trois points communs aux deux modèles et à calculer, à partir de ces points, la transformation positionnant un des modèles dans le repère de l'autre. Des travaux présentés dans [Dekeyser 03] par Dekeyser et al. propose une méthode pour détecter des éléments répétitifs dans les façades à partir d'une seule image. La méthode effectue tout d'abord une rectification de l'image en utilisant les principaux points de fuites détectés automatiquement. Après cette étape, la façade est fronto-parallèle au plan image avec un alignement des directions horizontale et verticale de la façade selon les directions O_x et O_y de l'image rectifiée. Par la suite, l'utilisateur localise manuellement un élément répétitif puis les motifs ressemblant à cet élément sont recherchés automatiquement dans la direction horizontale au motif de référence. Le critère de ressemblance utilisé est basé sur les contours et permet de supporter des changements d'aspect particulier comme un rideau tiré derrière une fenêtre. Cette approche peut améliorer la méthode de reconstruction que nous présentons dans ce document en permettant l'intégration automatique de modèles répétitifs à partir du moment où l'un de ces éléments a été saisi.

Une fois ces éléments répétitifs détectés et intégrés au modèle global, une autre idée est d'utiliser les appariements obtenus précédemment pour positionner les éléments de détail par rapport au modèle global. On peut observer sur la figure 6.5 que les plans supports de certains éléments de détails (fenêtre, porte) apparaissent clairement détachés du plan moyen de la façade. A partir de ce constat, nous envisageons d'utiliser un critère local basé sur la distance entre le nuage de points 3D issu des appariements et le modèle 3D du détail intégré pour affiner sa position. Cette problématique est proche du recalage de modèles CAO dans des nuages de points 3D très denses traité par Moron dans [Moron 96] et aussi par Chaperon et al. dans [Ertl 01], le logiciel commercial 3DIPSOS de la société Mensi effectue également ce type d'opération.

6.4 Exploiter la texture présente dans les images

Nous venons de présenter deux utilisations possibles des appariements de points homologues effectués entre les images. Ces appariements ont été obtenus en utilisant localement (au voisinage d'un point) l'information contenu dans les images. Nous allons maintenant présenter des perspectives s'appuyant sur une prise en compte plus large de l'aspect de la scène dans les images. La première application possible vise à identifier la présence d'incohérence entre la scène observée de sorte à détecter les occultations et à indiquer à l'utilisateur les zones mal modélisées. La deuxième partie aborde le problème du passage automatique d'un modèle ponctuel (la méthode de reconstruction proposée dans ce document reconstruit seulement les sommets du modèle) à un modèle surfacé. Enfin, le dernier point évoque la question de l'extraction automatique de la texture à partir des différentes images.

6.4.1 Détecter les incohérences et les occultations

Dans une situation idéale où la scène serait parfaitement modélisée, les informations présentes dans chacune des images seraient toutes cohérentes entre elles. Dans la réalité, les modèles obtenus ne sont que des approximations dont une des conséquences est l'existence d'incohérences entre les différentes images. Une fois notre reconstruction obtenue, si l'on considère la texture autour d'un point p_i dans l'image i , trois cas peuvent se présenter lorsque l'on compare cette texture aux textures correspondantes dans les autres images (par exemple j et k) (ces cas sont illustrés dans la figure 6.6) :

- **le cas cohérent** : Dans ce cas, le point 3D projeté correspondant au point p_i est véritablement un point appartenant au modèle 3D. Ce point 3D est réellement vu par les deux autres caméras j et k et les textures issues des trois images i, j, k sont cohérentes.
- **présence d'une occultation** : Dans ce cas, la texture au voisinage du point p_i ne correspond pas au modèle 3D mais à l'élément occultant. Le point 3D projeté est toutefois un point du modèle 3D (l'occultation n'est pas modélisée) et les textures associées aux voisinages des projections dans j et k ne seront pas toutes cohérentes avec le voisinage de p_i .
- **présence d'un élément de détail non reconstruit** : Lorsqu'un élément n'est pas reconstruit, on observe les mêmes phénomènes que dans le cas des occultations à la différence près que la proximité géographique entre le modèle déjà reconstruit et le détail. Cette proximité spatiale entre l'élément non reconstruit et le modèle 3D provoque une incohérence entre toutes les images observant cette zone qui peut permettre d'attirer l'attention de l'utilisateur sur cette zone.

6.4.2 Automatiser le passage du modèle ponctuel au modèle surfacique

La méthode présentée dans ce document reconstruit un nuage de point 3D à partir d'images et d'un ensemble de contraintes. Jusqu'à présent, les surfaces sont définies manuellement (cette méthode est également employée dans [Debevec 96] et dans [Faugeras 98]). Bien qu'il soit difficile d'évaluer l'importance de temps nécessaire à la définition des éléments de surface compte tenu de la diversité des modèles traités, cette durée ne peut pas être considérée comme négligeable dans le coût total de la reconstruction. L'automatisation de cette tâche est donc souhaitable. Une approche consistant à définir la surface en utilisant, par exemple, un maillage de Delaunay n'est pas pleinement satisfaisant car le maillage ainsi obtenu présente de nombreux artefacts et n'est pas cohérent avec les images car il ne tient compte ni de la texture ni des contours. Par conséquent, une étape de correction manuelle du maillage obtenu est nécessaire.

Morris et Kanade dans [Morris 00b] proposent une méthode permettant de définir un maillage pertinent du modèle 3D en utilisant l'information disponible dans les images. Le point de départ de cette méthode est un maillage valide mais erroné du modèle 3D ainsi

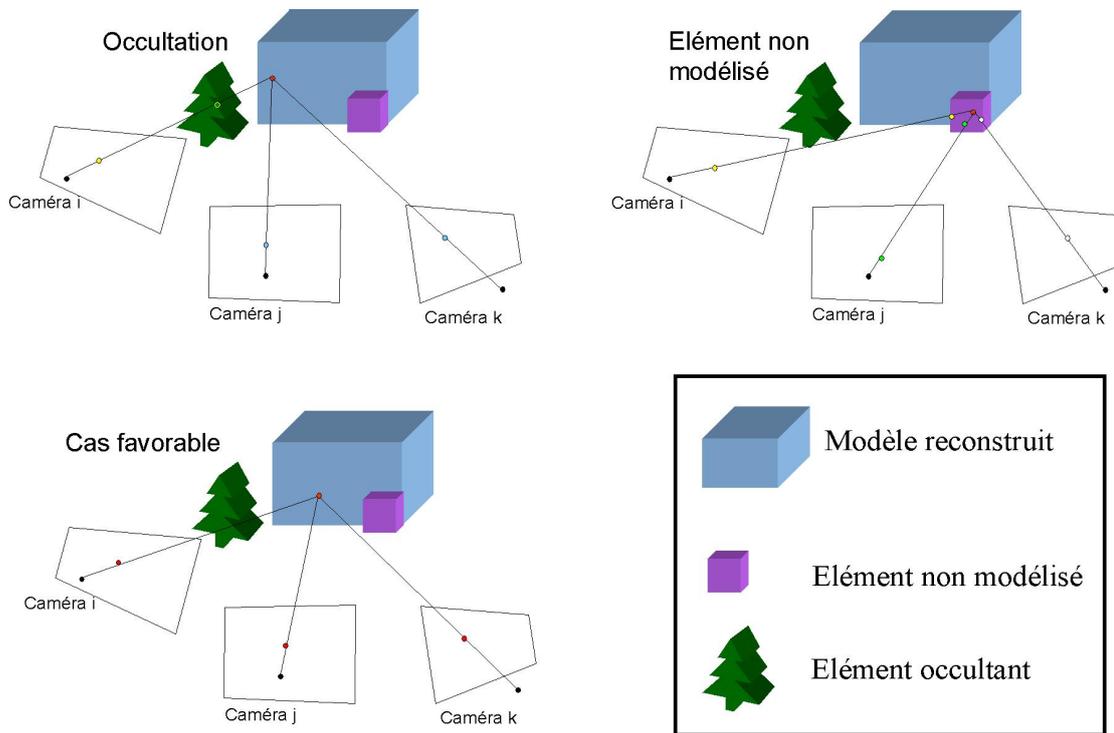


FIG. 6.6: Différents cas identifiables par une analyse de la cohérence entre les différentes images.

qu'un ensemble d'images calibrées. L'idée est de rechercher la meilleure triangulation en effectuant des permutations au niveau des arêtes du maillage jusqu'à maximiser un critère de vraisemblance basée sur l'apparence du modèle dans les images. Les résultats obtenus sont prometteurs car ils permettent de prendre en compte des points ou des segments anguleux au niveau de la surface tout en réduisant très efficacement le nombre d'artefacts. De plus, cette approche est suffisamment robuste pour supporter quelques réflexions spéculaires et permet d'identifier des zones de faibles cohérences pouvant correspondre à des éléments occultants ou à des structures mal modélisées.

La mise au point d'un algorithme automatique permettant de définir la surface du modèle 3D en partant uniquement du nuage de points reconstruits et des images calibrées est une perspective difficile dont les conséquences seraient, en cas de succès, une réduction conséquente du temps de modélisation. Toutefois, même sans parvenir à traiter intégralement tout le modèle, les travaux de Morris et Kanade permettent d'envisager une solution partielle permettant de surfacer automatiquement certaines parties du modèle.

6.4.3 Extraire efficacement la texture du modèle

Pour finir, les modèles obtenus par notre méthode souffrent de la qualité médiocre des textures extraites à partir des images. En effet, nous n'avons pas développé les outils nécessaires à une extraction fine de la texture dans les images. Obtenir un modèle réaliste implique d'extraire la texture en compensant les changements d'apparence intro-

duit par les changements de points de vues et requiert une prise en compte efficace des occultations.

Afin de limiter les temps de calcul, les solutions basées sur l'exploitation des possibilités offertes par les cartes graphiques et certains standards logiciels (OpenGL, Direct3D) semblent les plus pertinentes. Deux approches paraissent envisageables (en fonction des applications visées), la plus simple consiste à remplacer chaque prise de vue par un projecteur virtuel de texture (la texture étant l'image correspondant à cette prise de vue). La seconde approche consiste à rechercher pour chaque "élément de surface" du modèle 3D la couleur la plus probable...

6.5 Conclusion

Ce dernier chapitre a été l'occasion de présenter quelques voies d'amélioration possibles de notre approche en vue d'accroître la convivialité, l'efficacité et la qualité des résultats obtenus. La diversité des sujets (géométrie, appariement, texture,...) montre la richesse des questions soulevées par la reconstruction d'objets à partir d'images et l'ampleur des problèmes qui restent à résoudre.

Parmi les perspectives ouvertes dans ce chapitre, nous pensons que les verrous les plus difficiles à lever sont les problèmes évoqués dans la partie 'informations géométriques'. En effet, déterminer si le système est suffisamment contraint et comment initialiser le modèle 3D afin d'avoir, à coups sûrs, une convergence réussie sont des problèmes ardues.

Les outils visant à utiliser des appariements de points ou bien la texture présente dans les images sont actuellement l'objet de nombreux travaux. Cependant, leur utilisation demeure encore délicate et peut parfois se heurter à des échecs dû à l'aspect des images disponibles. De plus, une utilisation intensive de ces méthodes requiert un accroissement des performances en terme de temps de calcul pour pouvoir intégrer ces outils, non plus comme méthode finale de raffinement du modèle, mais comme des méthodes interactives au même titre que les outils de reconstruction basés sur les propriétés géométriques de la scène.

Conclusion

Notre objectif dans cette thèse était de développer une méthode interactive de modélisation d'objet permettant d'intégrer à la fois des connaissances a priori et des informations extraites d'images non-calibrées. Pour parvenir à ce résultat, nous avons développé une technique d'ajustement de faisceaux originale et une méthode de modélisation permettant d'intégrer des contraintes dans le modèle de l'objet observé.

Dans la première partie de ce document, nous avons exposé une méthode d'ajustement de faisceaux nommée `CACHE_POS` qui ne requière plus l'initialisation des attitudes des prises de vue et présente de nombreuses propriétés faisant de cette approche une solution complémentaire à la méthode d'ajustement de faisceaux traditionnelle. Nous avons montré que la structure de la matrice quasi-hessienne de `CACHE_POS` possède, comme la méthode classique, des éléments creux rendant particulièrement efficace l'emploi de cette approche lorsque le nombre de paramètres recherchés associés aux prises de vue est important. De plus, cette méthode présente un domaine de convergence plus large que la méthode classique, offrant de ce fait une plus grande souplesse d'emploi tout en offrant, dans la gamme de bruit qui concerne l'application visée, des précisions équivalentes à celles de la méthode classique. Pour finir, `CACHE_POS` apporte une réponse à la question du choix de jauge en apportant une solution originale consistant à fixer implicitement le repère de reconstruction à chaque itération du processus d'estimation.

Dans la deuxième partie, nous avons proposé une méthode de modélisation qui permet de caractériser le modèle observé au moyen de contraintes extrêmement diverses (pas nécessairement linéaires) tout en limitant le nombre de degrés de liberté à estimer en utilisant des informations saisies au moyen des images. La méthode de modélisation caractérisant chaque nouveau point par rapport à ceux déjà existants nous a évité d'avoir recours à des systèmes complexes d'identification de contraintes redondantes ou contradictoires. De plus, la décomposition de l'objet reconstruit en sous-éléments nous a permis de limiter la taille et la complexité des modèles reconstruits.

Après le développement de la méthode de modélisation et de l'algorithme d'ajustement de faisceaux, nous avons validé notre approche sur divers exemples. Ces essais nous ont permis de montrer que la méthode complète proposée offre une solution souple et fonctionnelle au problème de modélisation d'objets structurés à partir d'images. De plus, ils nous ont permis d'identifier les voies d'amélioration possible.

Enfin, les expériences présentées dans ce document montrent que notre approche peut s'appliquer à divers contextes. En effet, avec un même outil il est possible de modéliser des scènes très diverses à partir d'une ou de plusieurs images. Parmi les nombreuses

situations possibles, nous avons rencontrée les suivantes au cours de cette thèse :

- **dans le cas mono-image** : dans le chapitre 2, nous avons reconstruit un bureau à partir d'une seule image en utilisant des contraintes géométriques caractérisant la scène,
- **des modèles simples** : dans le chapitre 5, les exemples de la maison bretonne et de la chapelle romane montrent qu'il est possible de reconstruire des modèles simples de bâtiments en utilisant peu d'images,
- **des modèles complexes** : l'exemple du château de Sceaux montre qu'une stratégie de reconstruction stratifiée, consistant à reconstruire chaque élément particulier du bâtiment en utilisant des images dédiées, permet de reconstruire des modèles complexes avec des résolutions adaptées au besoin de l'utilisateur.

Le prototype développé au cours de ces travaux offre une solution complète à notre problème de reconstruction, mais il reste encore beaucoup de points à traiter pour accroître la convivialité et l'automatisation de l'outil. Les voies d'améliorations présentées dans le dernier chapitre de cette thèse ne sont qu'un extrait des idées bonnes ou mauvaises qu'il reste à tester, rejeter, valider, intégrer, publier,...

Bibliographie

- [Aanaes 02] H. Aanaes, R. Fisker, K. Aström, J. M. Cartensen. – Factorization with erroneous data. *ISPRS Commission III Photogrammetric Computer Vision*, pp. A :15–23, 2002.
- [Ameller 00] M.A. Ameller, B. Triggs, L. Quan. – Camera pose revisited - new linear algorithms, 2000.
- [Ansar 02] A. Ansar, K. Daniilidis. – Linear pose estimation from points and lines. *European Conference on Computer Vision (ECCV 2002)*, 2002.
- [Atkinson 96] K.B. Atkinson. – Close range photogrammetry and machine vision. *Whittles Publishing*, 1996.
- [Bartoli 03] A. Bartoli. – *Reconstruction et alignement en vision 3D : points, droites, plans, caméras*. – Phd thesis - Institut national polytechnique de grenoble, septembre 2003.
- [Bazin 01] P.L. Bazin. – *Modélisation et estimation du mouvement et de la structure dans une séquence d'images à partir d'indices géométriques épars, avec applications à la post-production audio-visuelle*. – Phd thesis - Université de Paris-Sud - Orsay, juin 2001.
- [Berger 96] M.O. Berger, G. Simon, S. Petitjean, B. Wrobel-Dautcour. – Mixing synthesis and video images of outdoor environments : Application to the bridges of paris. *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 1 :90–94, august 1996.
- [Bjork 90] A. Bjork. – Numerical method for least square problems. – *SIAM*, 1990.
- [Bondyfallat 98] D. Bondyfallat, S. Bougnoux. – Imposing euclidean constraints during self-calibration process. – *SMILE workshop*, 1998.
- [Boufama 95] B. Boufama, R. Mohr. – Epipole and fundamental matrix estimation using the virtual parallax property. *4th International Conference on Computer Vision (ICCV 1995)*, pp. 1030–1036, june 1995.
- [CAN] <http://www.canoma.com/>.
- [Caprile 90] B. Caprile, V. Torre. – Using vanishing points for camera calibration. *IJCV*, pp. 4 :127–140, 1990.
- [Chen 99] C. Chen, C. Yu, Y. Hung. – New calibration-free approach for augmented reality based on parameterized cuboid structure. *ICCV*, pp. 1 :30–37, 1999.

- [Christy 96] S. Christy, R. Horaud. – Euclidean shape and motion from multiple perspective views by affine iteration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18 :1098–1104, 1996.
- [Cornou 02a] S. Cornou, M. Dhome, P. Sayd. – Bundle adjustment : a fast method with weak initialisation. – *British Machine Vision Conference (BMVC'02)*, pp. 223–232, 2002.
- [Cornou 02b] S. Cornou, M. Dhome, P. Sayd. – Reconstruction de bâtiment à partir de n-vues non-calibrées. – *ORASIS*, 2002.
- [Cornou 03a] S. Cornou, M. Dhome, P. Sayd. – Architectural reconstruction with multiple views and geometric constraints. – *British Machine Vision Conference (BMVC'03)*, 2003.
- [Cornou 03b] S. Cornou, M. Dhome, P. Sayd. – Building reconstruction from n uncalibrated views. – *international ISPRS workshop "Vision Techniques for Digital Architectural and Archaeological Archives"*, 2003.
- [Cornou 04] S. Cornou, M. Dhome, P. Sayd. – Modélisation par contraintes et reconstruction 3d d'objets à partir de plusieurs vues non calibrées. – *Congrès francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2004)*, 2004.
- [Cruickshank 75] D. Cruickshank, P. Wyld. – London : the art of georgian building. – *Out of print*, 1975.
- [Davis 95] L. S. Davis, D.F. Dementhon. – Model-based object pose in 25 lignes of code. – *International Journal of Computer Vision*, pp. 15(2) :123–141, 1995.
- [Debevec 96] Paul E. Debevec, Camillo J. Taylor, Jitendra Malik. – Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach. *Computer Graphics*, 30(Annual Conference Series) :11–20, 1996.
- [Dekeyser 03] F. Dekeyser, F. Gaspard, L. De Luca, M. Florenzano, X. Chen, P. Leray. – Cultural heritage recording with laser scanning, computer vision and exploitation of architectural rules. – *international ISPRS workshop "Vision Techniques for Digital Architectural and Archaeological Archives"*, 2003.
- [DeMenthon 92] D. DeMenthon, L.S. Davis. – Exact and approximate solutions of the perspective-three-point problem. – *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 1100–1105, 1992.
- [Deriche 87] Rachid Deriche. – Using canny's criteria to derive a recursively implemented optimal edge detector. – *International Journal of Computer Vision (ICCV)*, 1987.
- [Deriche 94] Rachid Deriche, Zhengyou Zhang, Quang-Tuan Luong, Olivier D. Faugeras. – Robust recovery of the epipolar geometry for an uncalibrated stereo rig. – *ECCV*, pp. 567–576, 1994.

- [Dick 01] A. Dick, P. Torr, S. Ruffe, R. Cipolla. – Combining single view recognition and multiple view stereo for architectural scenes. – *8th International Conference on Computer Vision (ICCV)*, pp. 268–274, 2001.
- [Ertl 01] Thomas Ertl, Bernd Girod, Heinrich Niemann, Hans-Peter Seidel (édité par). – *Extracting Cylinders in Full 3D Data Using a Random Sampling Method and the Gaussian Image*. – Aka GmbH, 2001, 35–42p.
- [Faugeras 98] O. Faugeras, L. Robert, S. Laveau, G. Csurka, C. Zeller, C. Gauclin, I. Zoghlami. – 3-d reconstruction of urban scenes from images sequences. *CVIU*, 69(3) :292–309, 1998.
- [Faugeras 01] O. Faugeras, Q.T. Luong, T. Papadopoulos. – *The Geometry of Multiple Images*. – The MIT Press, 2001.
- [Fischler 81] M. Fischler, R. Bolles. – Random sample consensus : A paradigm for model fitting with application to image analysis and automated cartography. *Communication of the ACM*, pp. 24 :381–385, 1981.
- [Golub 96] G. Golub, C. van Loan. – Matrix computation. – *The Johns Hopkins University Press Ltd.*, 1996.
- [Grossman 00] E. Grossman, J. Santos-Victor. – Dual representation for vision-based 3d reconstruction. – *BMVC*, 2000.
- [Grossman 01] E. Grossman. – *Maximum likelihood 3D reconstruction from one or more uncalibrated views under geometric constraints*. – Phd thesis - Technical university of Lisbon, october 2001.
- [Harris] C. Harris, M. Stephens. – A combined corner and edge detector, journal = Proc. 4th Alvey Vision Conf., year = 1988, volume = 16, pages = 147-151, month = august.
- [Hartley 94] R.I. Hartley. – Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16 :1036–1041, october 1994.
- [Hartley 97] R.I. Hartley. – In defence of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6) :580–593, october 1997.
- [Hartley 00] R. Hartley, A. Zisserman. – *Multiple View Geometry in Computer Vision*. – Cambridge University Press, 2000.
- [Heyden 97] A. Heyden. – Projective structure and motion from image sequences using subspace methods. *10th Scandinavian Conference on Image Analysis (SCIA 1997)*, 1997.
- [IMA] <http://www.realviz.com/>.
- [Kanatani 00a] K. Kanatani. – Gauge-based reliability analysis of 3-d reconstruction from two uncalibrated perspective views. *15th Int. Conf. Pattern Recognition (ICPR 2000)*, 1 :76–79, september 2000 2000.

- [Kanatani 00b] K. Kanatani, D.D. Morris. – Gauges and gauge transformations in 3-d reconstruction from a sequence of images. *4th Asian Conference on Computer Vision (ACCV 2000)*, pp. 1046–1051, 8-11 January 2000.
- [Krauss 93] K. Krauss. – *Photogrammetry*. – UMMLEB buch, 1993.
- [Lhuillier 03] M. Lhuillier, L. Quan. – Image-based rendering by joint view triangulation. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 13(11) :1051–1063, 2003.
- [Lowe 85] G.G. Lowe. – Perceptual organization and visual recognition. *Kluwer, Automation* 6(5) :578–589, 1985.
- [Mahamud 00] S. Mahamud, M. Hebert. – Iterative projective reconstruction from multiple views. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.
- [Malis 01] E. Malis, A. Bartoli. – Euclidean bundle adjustment independent on camera intrinsic parameters. *report N° 4377*, december 2001.
- [McLauchlan 99a] P. McLauchlan. – Gauge independence in optimisation algorithms for 3d vision. *Proceedings of Vision Algorithms Workshop, in conjunction with ECCV'99*, 1999.
- [McLauchlan 99b] P. McLauchlan. – Gauge invariance in projective 3d reconstruction. *Proceedings of Multi-View Workshop, in conjunction with CVPR'99*, 1999.
- [Moron 96] V. Moron. – Mise en correspondance de données 3d avec un modèle cao. mai 1996.
- [Morris 00a] D.D. Morris, K. Kanatani, T. Kanade. – 3d model accuracy and gauge fixing. *report CMU-RI-TR-00-32*, december 2000.
- [Morris 00b] D.D. Morris, K. Kanatani, T. Kanade. – Image-consistent surface triangulation. *CVPR*, june 2000.
- [Morris 01a] D.D. Morris, K. Kanatani, T. Kanade. – Gauge fixing for accurate 3d estimation. *CVPR*, 2001.
- [Morris 01b] D.D. Morris, K. Kanatani, T. Kanade. – Gauge freedoms and uncertainty modeling for 3d computer vision. *PhD thesis*, march 2001.
- [Navab 02] N. Navab. – Canonical representation and three view geometry of cylinders. *ISPRS Commission III Photogrammetric Computer Vision*, pp. 218–224, 2002.
- [PHI] <http://www.phocad.de/produkte/phidias/english/english.html>.
- [PHO] <http://www.photomodeler.com/>.
- [Poelman 94] C.J. Poelman, T. Kanade. – A paraperspective factorization method for shape and motion recovery. *ECCV*, 2 :97–108, 1994.
- [Pollefeys 99] M. Pollefeys. – Self-calibration and metric 3d reconstruction from uncalibrated image sequences. *PhD thesis Katholieke Universiteit, Leuven*, 1999.

- [Pollefeys 00] M. Pollefeys, R. Koch, M. Vergauwen, L. Van Gool. – Automated reconstruction of 3d scenes from sequences of images. *ISPRS Journal Of Photogrammetry And Remote Sensing*, (55)4 :251–267, 2000.
- [Press 93] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. – *Numerical recipes in C++ - second edition*. – Cambridge University Press, 1993.
- [Quan 99] L. Quan, Z. Lan. – Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8) :774–780, 1999.
- [Sayd 01] P. Sayd, S. Naudet, M. Viala, L. Cohen, A. Dumont, F. Jallon. – Application : Aoms un outil de relevé 3d d’environnements industriels. *ORASIS 2001*, 2001.
- [Smith 95] S. M. Smith, J. M. Brady. – *SUSAN - A new approach to low level image processing*. – Rapport de Recherche n° TR95SMS1c, Chertsey, Surrey, UK, 1995.
- [Sparr 98] G. Sparr. – Euclidean and affine structure/motion for uncalibrated cameras from affine shape and subsidiary information. – *SMILE workshop*, 1998.
- [Sturm 96] P. F. Sturm, B. Triggs. – A factorization based algorithm for multi-image projective structure and motion. *4th European Conference on Computer Vision (ECCV 1996)*, april 1996.
- [Sturm 97] P. F. Sturm. – Vision 3d non calibrée : contribution à la reconstruction projective et étude des mouvements critiques pour l’auto-calibrage. *Thèse - Institut National Polytechnique de Grenoble*, décembre 1997.
- [Szeliski 98] R. Szeliski, P. H. S. Torr. – Geometrically constrained structure from motion : points on planes. – *SMILE workshop*, pp. 171–186, 1998.
- [Tomasi 92] C. Tomasi, T. Kanade. – Shape and motion from image streams under orthography : A factorization method. *Int. Journal of Computer Vision*, 9 :2 :137–154, 1992.
- [Triggs 00] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon. – Bundle adjustment - a modern synthesis. *Vision Algorithms : Theory et Practice*, 2000.
- [Ueshiba 98] T. Ueshiba, F. Tomita. – A factorisation method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. *5th European Conference on Computer Vision*, pp. 296–310, 1998.
- [Weinshall 93] D. Weinshall, C. Tomasi. – Linear and incremental acquisition of invariant shape models from image sequences. *ICCV*, pp. 675–682, 1993.
- [Werner 02a] T. Werner, A. Zisserman. – Model selection for automated architectural reconstruction from multiple views. – *Proceedings of the British Machine Vision Conference*, pp. 53–62, 2002.

- [Werner 02b] T. Werner, A. Zisserman. – New techniques for automated architecture reconstruction from photographs. – *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, vol. 2, pp. 541–555. Springer-Verlag, 2002.
- [Wilczkowiak 01] M. Wilczkowiak, E. Boyer, P. Sturm. – Camera calibration and 3d reconstruction from single images using parallelepipeds. *ICCV*, pp. 1 :142–148, 2001.
- [Wilczkowiak 03a] M. Wilczkowiak, P. Sturm, E. Boyer. – The analysis of ambiguous solutions in linear systems and its application to computer vision. – *BMVC*, pp. 53–62, 2003.
- [Wilczkowiak 03b] M. Wilczkowiak, G. Trombettoni, C. Jermann, P. Sturm, E. Boyer. – Scene modeling based on constraint system decomposition techniques. – *ICCV*, pp. 1004–1010, 2003.

Annexes

Algorithme d'estimation de pose : POSIT (Davis et Dementhon) [Davis 95]

6.5.1 Hypothèses initiales

Les hypothèses sous-jacentes retenues sont :

- connaissance du modèle de l'objet observé, défini dans un repère R_m ;
- acquisition d'une image de l'objet ;
- mise en correspondance entre des points détectés dans l'image et des points caractéristiques de l'objet.

6.5.2 But

Mettre en coïncidence les repères caméra (R_c) et modèle (R_m) et rechercher la rotation et la translation à appliquer au modèle pour le placer dans l'espace, dans une attitude conforme à l'image.

Les points p_i $i \in [1, n]$ sont les images des points P_i par projection perspective. Nous noterons (X_i, Y_i, Z_i) les coordonnées de P_i dans le repère (R_m) et (X'_i, Y'_i, Z'_i) les coordonnées de P_i après la rotation et la translation recherchées.

En coordonnées homogènes, nous avons :

$$\begin{pmatrix} X'_i \\ Y'_i \\ Z'_i \\ 1 \end{pmatrix} = \begin{pmatrix} R_{(3 \times 3)} & T_{(3 \times 1)} \\ 0_{(1 \times 3)} & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} i_x & i_y & i_z & u \\ j_x & j_y & j_z & v \\ k_x & k_y & k_z & w \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}$$

soit encore :

$$\begin{aligned}
 X'_i &= (i_x, i_y, i_z, u) \cdot \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = (I_{(1 \times 3)}, u) \cdot \begin{pmatrix} P_i(3 \times 1) \\ 1 \end{pmatrix} \\
 Y'_i &= (j_x, j_y, j_z, v) \cdot \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = (J_{(1 \times 3)}, v) \cdot \begin{pmatrix} P_i(3 \times 1) \\ 1 \end{pmatrix} \\
 Z'_i &= (k_x, k_y, k_z, w) \cdot \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = (K_{(1 \times 3)}, w) \cdot \begin{pmatrix} P_i(3 \times 1) \\ 1 \end{pmatrix}
 \end{aligned} \tag{6.7}$$

I, J, K sont les vecteurs directeurs du repère caméra (R_c) exprimés dans le repère du modèle (R_m) dans la position correspondant à la solution du problème.

6.5.3 Sous l'hypothèse de la projection perspective

Avec $p_i = (x_i, y_i, z_i (= f))$ mis en correspondance avec $P_i = (X_i, Y_i, Z_i)$, nous obtenons :

$$x_i = \frac{X'_i f}{Z'_i} = \frac{f(I, u) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix}}{(K, w) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix}} \Rightarrow x_i(K, w) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} = f(I, u) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} \tag{6.8}$$

$$y_i = \frac{Y'_i f}{Z'_i} = \frac{f(J, v) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix}}{(K, w) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix}} \Rightarrow y_i(K, w) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} = f(J, v) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} \tag{6.9}$$

$$x_i \left(\frac{K \cdot P_i}{w} + 1 \right) = \frac{f}{w}(I, u) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} \tag{6.10}$$

$$y_i \left(\frac{K \cdot P_i}{w} + 1 \right) = \frac{f}{w}(J, v) \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} \tag{6.11}$$

En posant :

$$\varepsilon_i = \frac{K \cdot P_i}{w} = \frac{k_x X_i + k_y Y_i + k_z Z_i}{w}$$

$$I^* = \frac{f}{w}(I, u) = \frac{f}{w}(i_x, i_y, i_z, u) = (I_1, I_2, I_3, I_4)$$

$$J^* = \frac{f}{w}(J, v) = \frac{f}{w}(j_x, j_y, j_z, v) = (J_1, J_2, J_3, J_4)$$

nous obtenons, en considérant toutes les correspondances, les deux systèmes linéaires suivants :

$$x_i(1 + \varepsilon_i) = I^* \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} \Rightarrow (P_i, 1) \cdot I^* = x_i(1 + \varepsilon_i) \quad (6.12)$$

$$i \in [1, n]$$

$$y_i(1 + \varepsilon_i) = J^* \cdot \begin{pmatrix} P_i \\ 1 \end{pmatrix} \Rightarrow (P_i, 1) \cdot J^* = y_i(1 + \varepsilon_i) \quad (6.13)$$

Ils peuvent être réécrits :

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_n & Y_n & Z_n & 1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} x_1(1 + \varepsilon_1) \\ \cdot \\ \cdot \\ x_n(1 + \varepsilon_n) \end{bmatrix} \quad (6.14)$$

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_n & Y_n & Z_n & 1 \end{bmatrix} \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \end{bmatrix} = \begin{bmatrix} y_1(1 + \varepsilon_1) \\ \cdot \\ \cdot \\ y_n(1 + \varepsilon_n) \end{bmatrix} \quad (6.15)$$

Seuls les vecteurs I^* , J^* et les coefficients ε_i sont inconnus dans ce système.

6.5.4 Sous l'hypothèse de la projection orthographique à l'échelle

On considère le plan Π passant par l'origine P_0 du repère du modèle (R_m) et parallèle au plan image.

En projection orthographique à l'échelle, on projette l'objet sur ce plan, puis on réalise une projection perspective sur le plan image (changement d'échelle en fonction de la distance).

A la solution recherchée, la distance entre le plan Π et l'origine du repère de la caméra est égale à la composante w de la translation.

Soit un point P_i du modèle et sa projection orthographique P_i'' sur le plan Π . Dans le repère caméra, P_i'' a pour coordonnées (X'_i, Y'_i, w) et a pour image $p_i''(x''_i, y''_i)$ avec :

$$x''_i = \frac{X'_i f}{w} \quad \text{et} \quad y''_i = \frac{Y'_i f}{w} \quad (6.16)$$

$$x''_i = \frac{f}{w}(I, u) \begin{pmatrix} P_i \\ 1 \end{pmatrix} \quad \text{et} \quad y''_i = \frac{f}{w}(J, v) \begin{pmatrix} P_i \\ 1 \end{pmatrix} \quad i \in [1, n]$$

Il apparaît que le terme négligé entre les deux modèles de projection n'est autre que $x_i \varepsilon_i$ ou $y_i \varepsilon_i$ avec $\varepsilon_i = (K \cdot P_i)/w$.

$K \cdot P_i$ est égal à la distance, selon l'axe optique, entre le point P_i et l'origine P_0 de (R_m). Si cette origine est choisie judicieusement (barycentre de l'objet), négliger ε_i revient à négliger la profondeur de l'objet (selon l'axe optique) par rapport à sa distance au centre optique. Cette hypothèse est souvent réaliste. Le fait que ε_i soit multiplié par x_i ou y_i

nous montre que l'on aura toujours intérêt à garder l'objet le plus centré possible dans l'image, pour minimiser l'erreur introduite par la projection orthographique à l'échelle.

L'originalité de la méthode de D. Dementhon réside dans un passage progressif du modèle de projection orthographique à l'échelle au modèle de la projection perspective.

6.5.5 Déroulement de l'algorithme

1. $t = 0$ et $\varepsilon_i(t) = \varepsilon_i(0) = 0$ pour $i \in [1, n]$ (projection orthographique à l'échelle) ;
2. calcul des vecteurs I^* et J^* à l'aide des 2 systèmes linéaires (6.14) et (6.15) (résolution au sens des moindres carrés) :

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_n & Y_n & Z_n & 1 \end{bmatrix}, \quad S_x = \begin{bmatrix} x_1(1 + \varepsilon_1(t)) \\ \cdot \\ \cdot \\ x_n(1 + \varepsilon_n(t)) \end{bmatrix}, \quad S_y = \begin{bmatrix} y_1(1 + \varepsilon_1(t)) \\ \cdot \\ \cdot \\ y_n(1 + \varepsilon_n(t)) \end{bmatrix} \quad (6.17)$$

$$AI^* = S_x \quad AJ^* = S_y \quad (6.18)$$

Si $n > 4$ les systèmes sont sur-déterminés et on utilise une minimisation au sens des moindres carrés :

$$\begin{aligned} {}^tAAI^* = {}^tAS_x &\Rightarrow I^* = ({}^tAA)^{-1}{}^tAS_x = A^+S_x \\ {}^tAAJ^* = {}^tAS_y &\Rightarrow J^* = ({}^tAA)^{-1}{}^tAS_y = A^+S_y \end{aligned} \quad (6.19)$$

A^+ est appelée la matrice *pseudo-inverse* de la matrice A . Cette matrice peut être précalculée car elle ne dépend que des points P_i du modèle appariés aux points image :

3.

$$N_I = \sqrt{I_1^2 + I_2^2 + I_3^2} \quad N_J = \sqrt{J_1^2 + J_2^2 + J_3^2}$$

$$I = (I_1, I_2, I_3)/(N_I) \quad J = (J_1, J_2, J_3)/(N_J) \quad K = I \wedge J \quad J = K \wedge I$$

$$w = \frac{f}{N_I} = \frac{f}{N_J} \quad u = I_4 \cdot \frac{w}{f} \quad v = J_4 \cdot \frac{w}{f}$$

4. $t = t + 1$, calcul des nouveaux $\varepsilon_i(t) = \frac{K \cdot P_i}{w}$

Si $|(\varepsilon_i(t) - \varepsilon_i(t-1))/n| < \text{seuil}$, alors I, J, K, u, v, w définissent la position recherchée, sinon retour à l'étape 2. Généralement la convergence nécessite 4 à 5 itérations sans nécessiter la recherche de conditions initiales particulières.

6.5.6 Cas particulier d'un objet planaire

Dans le cas des objets parfaitement plats, il est nécessaire d'apporter une modification à l'algorithme de Dementhon. En effet, la matrice tAA n'est que de rang 2. Si l'on s'arrange pour exprimer les coordonnées du modèle de telle façon que toutes les valeurs

en z soient nulles, on ne pourra calculer que les composantes I_1, I_2, I_4 et J_1, J_2, J_4 respectivement des vecteurs I^* et J^* .

Rappelons que :

$$I^* = (I_1, I_2, I_3, I_4) = \frac{f}{w}(i_x, i_y, i_z, u)$$

$$J^* = (J_1, J_2, J_3, J_4) = \frac{f}{w}(j_x, j_y, j_z, v)$$

Les vecteurs (i_x, i_y, i_z) et (j_x, j_y, j_z) étant de unitaires, on peut écrire la relation (6.20) :

$$\left(\frac{f}{w}i_x\right)^2 + \left(\frac{f}{w}i_y\right)^2 + \left(\frac{f}{w}i_z\right)^2 = \left(\frac{f}{w}j_x\right)^2 + \left(\frac{f}{w}j_y\right)^2 + \left(\frac{f}{w}j_z\right)^2 \quad (6.20)$$

En posant comme inconnues :

$$\begin{cases} x = \frac{f}{w}i_z \\ y = \frac{f}{w}j_z \end{cases} \quad (6.21)$$

nous obtenons une première expression :

$$x^2 - y^2 = \left(\frac{f}{w}j_x\right)^2 + \left(\frac{f}{w}j_y\right)^2 - \left(\frac{f}{w}i_x\right)^2 - \left(\frac{f}{w}i_y\right)^2 \quad (6.22)$$

De plus, les deux vecteurs (i_x, i_y, i_z) et (j_x, j_y, j_z) , sont orthogonaux et forment une base avec (k_x, k_y, k_z) , d'où la relation (6.23) qui découle du produit scalaire nul :

$$\left(\frac{f}{w}i_x\right)\left(\frac{f}{w}j_x\right) + \left(\frac{f}{w}i_y\right)\left(\frac{f}{w}j_y\right) + xy = 0 \quad (6.23)$$

Ce qui nous conduit à un système de deux équations à deux inconnues :

$$\begin{cases} x^2 - y^2 = A \\ xy = B \end{cases} \quad (6.24)$$

avec :

$$\begin{cases} A = \left(\frac{f}{w}j_x\right)^2 + \left(\frac{f}{w}j_y\right)^2 - \left(\frac{f}{w}i_x\right)^2 - \left(\frac{f}{w}i_y\right)^2 \\ B = -\left[\left(\frac{f}{w}i_x\right)\left(\frac{f}{w}j_x\right) + \left(\frac{f}{w}i_y\right)\left(\frac{f}{w}j_y\right)\right] \end{cases} \quad (6.25)$$

d'où une équation du second degré en x^2 (6.26), ayant pour solutions (6.27) :

$$x^4 - Ax^2 - B^2 = 0 \quad (6.26)$$

$$x^2 = \frac{A \pm \sqrt{A^2 + 4B^2}}{2} \quad (6.27)$$

Or, $\sqrt{A^2 + 4B^2}$ étant plus grand que A , leur différence est toujours négative et leur somme positive. Il n'existe donc qu'une seule solution positive en x^2 qui conduit à deux solutions réelles en x :

$$x = \pm \sqrt{\frac{A + \sqrt{A^2 + 4B^2}}{2}} \quad (6.28)$$

Etant donnée la relation (6.24) liant y et x , et pour une valeur de x non nulle :

$$y = \frac{B}{x} \quad (6.29)$$

Pour le cas particulier où $x = 0$, la valeur de y est extraite de l'expression (6.24) :

$$y = \pm\sqrt{-A} \quad (6.30)$$

L'algorithme de Dementhon appliqué au cas particulier des objets planaires conduit, pour chaque itération du système, à deux couples de solution (I_i^*, J_i^*) et donc à deux attitudes distinctes de l'objet dans l'espace.

Dans le cadre de cette étude, nous avons testé deux méthodes pour sélectionner à chaque itération une des deux attitudes. Ces deux approches diffèrent par le critère utilisé pour départager les deux solutions. Le premier mesure les résidus des deux systèmes linéaires traités. Le second calcule un critère lié à la reprojection des points 3D dans le plan image. Ces deux approches seront respectivement nommées (*critère-système*, *critère-image*) dans la suite de ce chapitre. Les critères correspondants sont :

$$\text{Arg} \min_{k \in [1,2]} (||AI_k^* - S_x|| + ||AJ_k^* - S_y||) \quad (6.31)$$

$$\text{Arg} \min_{k \in [1,2]} \sum_{i=1}^{i=n} \sqrt{\left(x_i - \frac{fX_i^{k'}}{Z_i^{k'}}\right)^2 + \left(y_i - \frac{fY_i^{k'}}{Z_i^{k'}}\right)^2} \quad (6.32)$$

Les expériences effectuées ont conduit à des résultats identiques pour les deux critères.

Il faut noter une double modification de la méthode due à Horaud et Christy [?], dans le sens :

- utilisation de segment à la place des points ;
- remplacement de la projection orthographique par la paraperspective.

Algorithme d'estimation de pose : méthode de Lowe [Lowe 85]

6.6 Localisation d'un objet volumique par vision monoculaire

6.6.1 Introduction

Dans ce paragraphe, nous allons décrire une technique initialement proposée par D.Lowe [?] pour calculer, à partir d'une image vidéo, la position relative d'un objet volumique par rapport au repère associé au capteur qui l'observe. Nous avons retenu cette approche pour son formalisme qui s'adapte, comme nous le verrons ultérieurement, à de très nombreuses situations. Il s'agit ici de résoudre un problème géométrique inverse. En effet, nous recherchons, à partir d'un ensemble de mise en correspondance entre des primitives 2D détectées dans une image et les éléments correspondant du modèle volumique de l'objet observé, l'attitude spatiale de ce dernier cohérente avec ces appariements. C'est-à-dire la position de l'objet dans l'espace qui réalise la coïncidence entre la projection perspective des primitives 3D avec leurs homologues dans l'image (voir figure ??).

6.6.2 Les mises en correspondance

L'algorithme initial proposé par D.Lowe [?] œuvre à partir d'appariements entre segment de droites, mais nous allons montrer que cette approche se transpose très facilement au cas où l'objet observé est caractérisé par un ensemble de points d'intérêt ; ces deux types d'appariements pouvant d'ailleurs être mixés au sein d'une même procédure de recherche d'attitude.

6.6.2.1 Appariement de droites

Soit une arête de l'objet délimitée par les points P_i^1 et P_i^2 dans le repère du modèle R_m . Supposons que cette arête crée dans l'image, par projection perspective, un segment caractérisé par un point quelconque $p_i = (x_i, y_i, f)$ et un vecteur $\vec{v}_i = (a_i, b_i, 0)$. Il est possible de définir un plan particulier, dénommé *plan d'interprétation*, passant par le centre optique de la caméra et contenant ce segment image. Ce plan est caractérisé par sa normale unitaire \vec{N}_i facilement estimable à partir des éléments image précédemment

mentionnés :

$$\vec{N}_i = \frac{\vec{op}_i \wedge \vec{v}_i}{\|\vec{op}_i \wedge \vec{v}_i\|} \quad (6.33)$$

Puisque les points P_i^1 et P_i^2 appartiennent à ce plan d'interprétation (voir figure ??), leurs coordonnées doivent vérifier les relations suivantes :

$$\begin{cases} \vec{N}_i \cdot P_i^{1'} = \vec{N}_i \cdot [\mathbf{R}_{\alpha\beta\gamma} P_i^1 + \mathbf{T}_{uvw}] = 0 \\ \vec{N}_i \cdot P_i^{2'} = \vec{N}_i \cdot [\mathbf{R}_{\alpha\beta\gamma} P_i^2 + \mathbf{T}_{uvw}] = 0 \end{cases} \quad (6.34)$$

REMARQUE 4.2.– Dans les équations (6.33) et (6.34), les symboles mathématiques $\vec{ij} \wedge \vec{ik}$ et $\vec{ij} \cdot \vec{ik}$ représentent respectivement le produit vectoriel et le produit scalaire. De plus, afin d'alléger certaines équations, nous omettrons le symbole vectoriel sur les normales aux plans d'interprétation.

6.6.2.2 Appariements de points

Lorsque l'objet est caractérisé par des points d'intérêt, l'étape de mise en correspondance définit des appariements entre un point P_i dont les coordonnées sont exprimées dans le repère du modèle R_m et la localisation de sa projection perspective dans le plan image p_i .

Il est alors possible de se ramener au cas précédent en définissant pour chaque point p_i deux plans d'interprétation qualifiés de *quasi horizontal* et de *quasi vertical*, de normales respectives N_i^1 et N_i^2 (voir figure ??) :

$$N_i^1 = \frac{\vec{op}_i \wedge \vec{Y}}{\|\vec{op}_i \wedge \vec{Y}\|} \quad N_i^2 = \frac{\vec{op}_i \wedge \vec{X}}{\|\vec{op}_i \wedge \vec{X}\|}$$

où $\vec{X} = (1, 0, 0)$ et $\vec{Y} = (0, 1, 0)$ sont respectivement les vecteurs unitaires des axes X et Y du repère caméra. A la solution, les deux relations qui imposent l'alignement des points p_i et P_i' doivent être vérifiées :

$$\begin{cases} N_i^1 \cdot P_i' = N_i^1 \cdot [\mathbf{R}_{\alpha\beta\gamma} P_i + \mathbf{T}_{uvw}] = 0 \\ N_i^2 \cdot P_i' = N_i^2 \cdot [\mathbf{R}_{\alpha\beta\gamma} P_i + \mathbf{T}_{uvw}] = 0 \end{cases} \quad (6.35)$$

6.6.3 Critère à minimiser

Afin d'intégrer dans un même cadre les calculs pour les deux types d'appariement évoqués précédemment (voir les systèmes d'équations (6.34) et (6.35)), nous considérerons des couples (P_i^j, N_i^j) correspondant respectivement à un point 3D de l'objet et à la normale du (ou d'un des) plan(s) d'interprétation associé(s). Avec cette notation, $N_i^1 = N_i^2$ pour un appariement entre droites et $P_i^1 = P_i^2$ pour un appariement entre points.

A la solution de notre problème de localisation, les n points du modèle appariés devraient se retrouver dans leurs plans d'interprétation respectifs. Toutefois, suite aux imprécisions de détection de primitives dans les images et, parfois, suite aux imperfections du modèle de l'objet observé, cet alignement ne peut être parfaitement réalisé. Il nous

faut donc définir un critère à minimiser. Nous avons proposé dans [?], par soucis de simplicité, de considérer la distance orthogonale du point 3D au plan d'interprétation associé.

Comme, par définition, un tel plan passe par l'origine du repère caméra R_c et est paramétré par une normale unitaire, la distance évoquée est égale à un simple produit scalaire :

$$\mathcal{D}(N_i^j, P_i^{j'}) = N_i^j \cdot P_i^{j'} = N_i^j \cdot [\mathbf{R}_{\alpha\beta\gamma} P_i^j + \mathbf{T}_{uvw}] \quad (6.36)$$

Soit $\mathcal{F}(\alpha, \beta, \gamma, u, v, w, N_i^j, P_i^j)$ la fonctionnelle qui a tout couple (P_i^j, N_i^j) associe la valeur définie en (6.36). L'objectif de la méthode de localisation est donc de trouver le vecteur de paramètres $(\alpha, \beta, \gamma, u, v, w)$ qui minimisent le critère suivant :

$$\mathcal{E} = \sum_{i=1}^n \sum_{j=1}^2 [\mathcal{D}(N_i^j, P_i^{j'})]^2 = \sum_{i=1}^n \sum_{j=1}^2 \mathcal{F}(\alpha, \beta, \gamma, u, v, w, N_i^j, P_i^j)^2$$

Ce critère n'est pas linéaire. Sa minimisation nécessite donc la mise en œuvre d'un processus itératif du type Newton-Raphson ou Levenberg-Marquard (voir [?] pour les détails théoriques concernant ces techniques) afin d'estimer le vecteur d'état recherché.

6.6.4 Résolution du problème par la méthode de Newton-Raphson

Soit A , le vecteur d'état correspondant à une rotation et une translation quelconque et A_k le vecteur d'état relatif à la rotation et translation déterminée à l'étape k du processus itératif :

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_6 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \cdot \\ \cdot \\ w \end{pmatrix} \quad A_k = \begin{pmatrix} a_{1k} \\ a_{2k} \\ \cdot \\ \cdot \\ a_{6k} \end{pmatrix} = \begin{pmatrix} \alpha_k \\ \beta_k \\ \cdot \\ \cdot \\ w_k \end{pmatrix}$$

Ecrivons le développement limité à l'ordre 1 de la fonctionnelle \mathcal{F} au voisinage de A_k :

$$\begin{aligned} \mathcal{F}(A, N_i^j, P_i^j) &\simeq \mathcal{F}(A_k, N_i^j, P_i^j) + \left. \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial A} \right|_{A=A_k} (A - A_k) \\ &\simeq \mathcal{F}(A_k, N_i^j, P_i^j) + \sum_{l=1}^6 \left. \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial a_l} \right|_{A=A_k} (a_l - a_{lk}) \end{aligned}$$

En supposant que A soit la solution recherchée, alors $\mathcal{F}(A, N_i^j, P_i^j) = 0$ et nous pouvons écrire l'approximation suivante :

$$-\mathcal{F}(A_k, N_i^j, P_i^j) \simeq \left. \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial A} \right|_{A=A_k} (A - A_k)$$

en posant $\Delta a_l = a_l - a_{lk}$ pour $l \in [1, 6]$ et en considérant l'ensemble des n appariements, nous obtenons le système linéaire suivant :

$$\left\{ \begin{array}{l} -\mathcal{F}(A_k, N_1^1, P_1^1) = \frac{\partial \mathcal{F}(A, N_1^1, P_1^1)}{\partial \alpha} \Big|_{A=A_k} \Delta \alpha + \dots + \frac{\partial \mathcal{F}(A, N_1^1, P_1^1)}{\partial w} \Big|_{A=A_k} \Delta w \\ \vdots \\ -\mathcal{F}(A_k, N_n^2, P_n^2) = \frac{\partial \mathcal{F}(A, N_n^2, P_n^2)}{\partial \alpha} \Big|_{A=A_k} \Delta \alpha + \dots + \frac{\partial \mathcal{F}(A, N_n^2, P_n^2)}{\partial w} \Big|_{A=A_k} \Delta w \end{array} \right.$$

Ce dernier ayant six inconnues, il est nécessaire de traiter au minimum trois appariements fournissant deux équations chacun. Il peut être noté matriciellement :

$$[F] = [J][\Delta A]$$

Sa solution au sens des moindres carrés, pour ($n > 3$), est donnée par :

$$[\Delta A] = ([{}^t J][J])^{-1} [{}^t J][F]$$

Cela revient à calculer le vecteur correctif ΔA qui minimise la somme des distances au carré des points du modèle aux plans d'interprétation. Connaissant le vecteur correctif ΔA , il est possible de calculer un nouveau vecteur d'état A_{k+1} par la relation suivante :

$$A_{k+1} = A_k + \Delta A$$

Si les conditions initiales A_0 sont choisies avec soin, une dizaine d'itérations suffit pour réaliser la convergence, le test d'arrêt pouvant porter sur la norme du vecteur ΔA .

6.6.5 Calcul des dérivées partielles

L'obtention de la matrice jacobienne $[J]$ nécessite l'estimation de nombreuses dérivées partielles. Nous allons voir comment simplifier ces calculs, notamment ceux associés aux angles d'Euler. Rappelons que la fonctionnelle $\mathcal{F}(A, N_i^j, P_i^j)$ est égale au produit scalaire suivant :

$$\mathcal{F}(A, N_i^j, P_i^j) = N_i^j \cdot [\mathbf{R}_{\alpha\beta\gamma} P_i^j + \mathbf{T}_{uvw}]$$

où $\mathbf{R}_{\alpha\beta\gamma}$ est égale au produit de trois matrices de rotation élémentaires autour de chaque axe du repère.

$$\begin{aligned} \mathbf{R}_{\alpha\beta\gamma} &= \mathbf{R}_\gamma \mathbf{R}_\beta \mathbf{R}_\alpha = \\ &\begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \end{aligned}$$

Si nous considérons par exemple la dérivée partielle de la fonctionnelle par rapport à l'angle β , nous devons effectuer le calcul suivant :

$$\frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial \beta} = N_i^j \cdot \frac{\partial [\mathbf{R}_\gamma \mathbf{R}_\beta \mathbf{R}_\alpha P_i^j + \mathbf{T}_{uvw}]}{\partial \beta} = N_i^j \cdot \left[\mathbf{R}_\gamma \frac{\partial \mathbf{R}_\beta}{\partial \beta} \mathbf{R}_\alpha P_i^j \right]$$

Afin de simplifier ces expressions, à chaque étape du processus itératif, on modifie la position du modèle dans son repère R_m . Soit A_k le vecteur d'état trouvé à l'étape k . On calcule alors les points $P_i^j|_k = \mathbf{R}_{\alpha_k\beta_k\gamma_k}P_i^j + \mathbf{T}_{u_kv_kw_k}$ qui deviennent les nouveaux points caractéristiques du modèle.

Ceci revient simplement à chaque étape à choisir un repère relatif à la dernière position calculée, en sorte que le calcul de toutes les dérivées partielles se fassent pour un vecteur de paramètre nul. Le critère à minimiser à l'étape $k + 1$ est donc :

$$\mathcal{E}_{k+1} = \sum_{i=1}^n \sum_{j=1}^2 \left[\mathcal{F}(A_k = 0, N_i^j, P_i^j|_k) + \sum_{l=1}^6 \frac{\partial \mathcal{F}(A, N_i^j, P_i^j|_k)}{\partial a_l} \Big|_{A=0} \Delta a_l \right]^2$$

Les dérivées partielles sont alors triviales. En effet, en notant $(N_{ix}^j, N_{iy}^j, N_{iz}^j)$ les composantes de la normale N_i^j , nous obtenons :

$$\mathbf{R}_\alpha = \mathbf{R}_\beta = \mathbf{R}_\gamma = I_{3 \times 3}$$

$$\frac{\partial \mathbf{R}_\alpha}{\partial \alpha} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad \frac{\partial \mathbf{R}_\beta}{\partial \beta} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \frac{\partial \mathbf{R}_\gamma}{\partial \gamma} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

d'où :

$$\begin{aligned} \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial \alpha} \Big|_{A=0} &= N_{iz}^j Y_i^k - N_{iy}^j Z_i^k & \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial u} \Big|_{A=0} &= N_{ix}^j \\ \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial \beta} \Big|_{A=0} &= N_{ix}^j Z_i^k - N_{iz}^j X_i^k & \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial v} \Big|_{A=0} &= N_{iy}^j \\ \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial \gamma} \Big|_{A=0} &= N_{iy}^j X_i^k - N_{ix}^j Y_i^k & \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial w} \Big|_{A=0} &= N_{iw}^j \end{aligned}$$

soit encore, de manière condensée :

$$\frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial \mathbf{R}_{\alpha\beta\gamma}} \Big|_{A=0} = P_i^j|_k \wedge N_i^j \quad \frac{\partial \mathcal{F}(A, N_i^j, P_i^j)}{\partial \mathbf{T}_{uvw}} \Big|_{A=0} = N_i^j$$

L'estimation du vecteur d'état relatif à l'étape $(k + 1)$ est obtenue par :

$$\begin{cases} \mathbf{R}_{\alpha_{k+1}\beta_{k+1}\gamma_{k+1}} &= [\mathbf{R}_{\Delta\alpha\Delta\beta\Delta\gamma}] \cdot [\mathbf{R}_{\alpha_k\beta_k\gamma_k}] \\ \mathbf{T}_{u_{k+1}v_{k+1}w_{k+1}} &= [\mathbf{R}_{\Delta\alpha\Delta\beta\Delta\gamma}] \cdot \mathbf{T}_{u_kv_kw_k} + \mathbf{T}_{\Delta u\Delta v\Delta w} \end{cases}$$