**Département de formation doctorale en informatique** **École doctorale IAEM Lorraine**
**UFR STMIA**

# Gestion des occultations en Réalité Augmentée

## THÈSE

présentée et soutenue publiquement le 23 mai 2001

pour l'obtention du

**Doctorat de l'université Henri Poincaré – Nancy 1**
**(spécialité informatique)**

par

Vincent LEPETIT

**Composition du jury**

| | |
|---|---|
| *Président :* | Jean-Paul HATON, Professeur, Université Henri Poincaré, Nancy |
| *Rapporteurs :* | Michel DHOME, Directeur de recherche CNRS, Clermont-Ferrand<br>Claude LABIT, Directeur de recherche INRIA, Rennes |
| *Examinateurs :* | Marie-Odile BERGER, Chargée de recherche INRIA, Nancy (directrice de thèse)<br>Daniel THALMANN, Professeur, EPFL, Lausanne |
| *Rapporteur interne :* | Frédéric ALEXANDRE, Directeur de recherche INRIA, Nancy |

**Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503**

# Remerciements

Je remercierai tout d'abord Marie-Odile Berger, ma directrice de thèse qui m'a permis de travailler sur un sujet peu exploré, et m'a fait partager sa grande expérience avec patience.

Je remercie également Jean-Paul Haton, pour le grand honneur qu'il m'a fait en acceptant d'être président de mon jury de thèse. Chacun des membres du jury m'ont fait un non moins grand honneur: Michel Dhome et Claude Labit, rapporteurs d'une grande compétence, Frédéric Alexandre, qui a accepté d'être rapporteur interne sur un travail éloigné de son domaine de recherche, et Daniel Thalmann qui s'est déplacé jusqu'à Nancy pour la soutenance et qui m'a offert un post-doctorat dans son laboratoire.

Je me permettrai de remercier également ici Gilles, pour m'avoir supporté, dans les deux sens du terme, pendant presque 4 ans. Un ami à qui je dois beaucoup. Je remercie (vous avez un synonyme?) aussi un autre ami, Guillaume, qui m'a (entre autres) toujours accueilli avec beaucoup de gentillesse lors de mes missions au départ de Charles-de-Gaulle. J'adresse également un grand merci à Céline et Sophie pour s'être occupées de Miette pendant ces mêmes missions, malgré son ingratitude ! Je pense aussi à mes anciens camarades de promo (Bertrand, David, Frédéric...) pour qui il serait temps que j'arrête d'incruster des vaches et que j'aie enfin un vrai métier.

Je remercie également mes parents, qui ont toujours su me permettre de faire ce que je souhaitais, ainsi que mes grandes sœurs Véronique et Marie, qui m'ont toujours soutenu, mes beaux-frères Jean-Michel et Jean-Marc (sans qui la séquence de la voiture n'aurait pas été possible, merci encore), et toute la marmaille : mes filleuls Bastien et Julien, ainsi que Céline, Amélie et Jérôme. J'espère que ce manuscrit leur permettra de comprendre un peu mieux ce que fait leur oncle (il incruste des vaches).

*Souvenez-vous de ceci: il n'y a pas de traits, il n'y a que des volumes. [...] C'est le relief qui régit le contour.*

Auguste Rodin

# Table des matières

**Introduction générale** **1**

**Chapitre 1 La Réalité Augmentée: définitions, applications et problématiques** **3**

1.1 Définitions . . . 3

1.2 Applications . . . 3

1.2.1 Effets spéciaux . . . 4

1.2.2 Visualisation de projets . . . 4

1.2.3 Médecine . . . 5

1.2.4 Maintenance industrielle . . . 6

1.2.5 Jeu . . . 6

1.2.6 Télévision . . . 7

1.3 Problématique . . . 7

1.3.1 Point de vue . . . 8

1.3.2 Occultations . . . 8

1.3.3 Éclairage . . . 8

1.4 Avancées des travaux en RA . . . 9

1.4.1 Point de vue . . . 9

1.4.2 Occultations . . . 9

1.4.3 Éclairages directs et indirects . . . 11

1.5 Objectifs de la thèse . . . 12

1.6 Plan de la thèse . . . 13

**Chapitre 2 Calcul des points de vue** **15**

2.1 Géométrie d'une caméra : le modèle sténopé . . . 15

2.1.1 Les paramètres externes . . . 16

2.1.2 Les paramètres internes . . . 16

2.2 Techniques basées images . . . 17

2.2.1 Géométries projective et euclidienne . . . 18

2.2.2 La géométrie épipolaire et la matrice fondamentale . . . 18

2.2.3 Les paramètres internes . . . . . 20
2.2.4 Le tenseur trifocal . . . . . 22
2.2.5 Ajustement de faisceaux . . . . . 22
2.2.6 En conclusion sur les méthodes basées image . . . . . 24
2.3 Technique basée modèle développée dans l'équipe . . . . . 25
2.3.1 Introduction . . . . . 25
2.3.2 Utilisation de courbes 3D quelconques . . . . . 25
2.3.3 Estimation robuste . . . . . 27
2.3.4 Critère minimisé . . . . . 28
2.3.5 Résultats . . . . . 28
2.4 Méthode hybride: fusion de données 3D et 2D . . . . . 29
2.4.1 Principe . . . . . 30
2.4.2 Détection et appariement des points 2D . . . . . 30
2.4.3 Résultats . . . . . 32
2.5 Ajustement de faisceaux . . . . . 32
2.5.1 Implantation . . . . . 34
2.5.2 Expérimentations . . . . . 35
2.5.3 Remarques sur l'ajustement de faisceaux . . . . . 36
2.6 Conclusion . . . . . 36

**Chapitre 3 Estimation de l'erreur sur les points de vue** **41**

3.1 Causes de l'erreur . . . . . 41
3.2 Estimation de l'erreur . . . . . 42
3.2.1 Modélisation de l'erreur . . . . . 42
3.2.2 Cas général, méthode statistique . . . . . 43
3.2.3 $\mathbf{y}$ fonction de $\mathbf{x}$, méthode analytique . . . . . 43
3.2.4 $\mathbf{y}$ défini comme le minimum d'un critère $C(\mathbf{x},\mathbf{y})$ [Csurka et al.97] . . . . . 44
3.3 Méthode retenue . . . . . 45
3.3.1 Calcul de la matrice de covariance à un coefficient multiplicatif près . . . 45
3.3.2 Interprétation géométrique: les régions d'indifférence . . . . . 46
3.4 Cas des méthodes basée modèle et hybride . . . . . 47
3.4.1 Implantation . . . . . 47
3.4.2 Choix du paramètre $\epsilon$ . . . . . 47
3.4.3 Interprétation des résultats . . . . . 48
3.4.4 L'incertitude est plus importante en profondeur . . . . . 49
3.4.5 La méthode hybride réduit l'incertitude . . . . . 50

3.4.6 Évolution de l'incertitude en fonction de la distance aux primitives 3D . . 50
3.5 Cas de l'ajustement de faisceaux . . . . . 54
3.5.1 Forme du hessien . . . . . 54
3.5.2 Calcul des matrices de covariance associés aux $\mathbf{P_i}$ . . . . . 55
3.5.3 Calcul des ellipsoïdes d'indifférence . . . . . 56
3.5.4 Expérimentations . . . . . 56
3.6 Conclusion . . . . . 57

**Chapitre 4 Reconstruction 3D en vision par ordinateur** **59**
4.1 Occultations et contours . . . . . 59
4.1.1 Masque d'occultation et contours occultants . . . . . 59
4.1.2 Régions occultées . . . . . 60
4.1.3 Contours apparents . . . . . 60
4.1.4 Discontinuité de profondeur . . . . . 62
4.2 Points de vue utilisés pour la reconstruction . . . . . 62
4.2.1 Précision des points de vue . . . . . 62
4.2.2 Disposition des points de vue . . . . . 62
4.2.3 Nombre de points de vue . . . . . 63
4.3 Stéréoscopie binoculaire dense . . . . . 64
4.3.1 Corrélation d'intensités . . . . . 64
4.3.2 Réduction des ambiguïtés . . . . . 68
4.3.3 Gestion des occultations . . . . . 69
4.3.4 Discussion . . . . . 72
4.4 Extension de la stéréoscopie binoculaire à $n$ images . . . . . 73
4.4.1 Stéréoscopie « multi baseline » . . . . . 74
4.4.2 Fusion de cartes 3D denses . . . . . 74
4.4.3 Chaînage de mises en correspondance binoculaires . . . . . 74
4.4.4 Discussion . . . . . 75
4.5 Reconstruction par appariement de primitives 2D . . . . . 76
4.5.1 Densification par triangulation . . . . . 76
4.5.2 Discussion . . . . . 77
4.6 Reconstruction par « méthode d'ensemble de niveaux » (*level set method*) . . . . 78
4.6.1 Présentation . . . . . 78
4.6.2 Implantation . . . . . 78
4.6.3 Discussion . . . . . 78
4.7 Reconstruction par « découpage de l'espace » (*space carving*) . . . . . 79
4.7.1 Présentation . . . . . 79

4.7.2 Notion de reconstruction maximale . . . . . 80
4.7.3 Discussion . . . . . 81
4.8 Reconstruction avec intervention de l'utilisateur . . . . . 82
4.8.1 Façade . . . . . 82
4.8.2 Image Modeler . . . . . 83
4.9 Conclusion . . . . . 83

**Chapitre 5 État de l'Art sur la gestion des occultations en Réalité Augmentée 87**
5.1 Approche basée contours [Berger97] . . . . . 87
5.1.1 Description . . . . . 87
5.1.2 Discussion . . . . . 89
5.2 Approche par suivi temporel 2D des objets occultants . . . . . 90
5.2.1 Prédiction . . . . . 90
5.2.2 Ajustement local . . . . . 92
5.2.3 Discussion . . . . . 93
5.3 Approche interactive [Ong et al.98] . . . . . 94
5.3.1 Description . . . . . 94
5.3.2 Discussion . . . . . 95
5.4 Conclusion . . . . . 96

**Chapitre 6 Gestion semi-automatique des occultations 99**
6.1 Description générale . . . . . 99
6.1.1 Cas d'un objet ne présentant que des arêtes vives . . . . . 99
6.1.2 Cas d'un objet présentant des surfaces courbes . . . . . 102
6.2 Reconstruction des contours 3D . . . . . 102
6.2.1 Mise en correspondance . . . . . 102
6.2.2 Filtre médian . . . . . 103
6.2.3 Interpolation des parties non reconstruites . . . . . 105
6.2.4 Choix du contour à reprojeter . . . . . 106
6.3 Détermination des masques d'occultation . . . . . 107
6.3.1 Nécessité de l'étape de correction . . . . . 107
6.3.2 Correction par modèle de mouvement . . . . . 107
6.3.3 Choix du modèle de mouvement . . . . . 107
6.3.4 Critère à minimiser . . . . . 108
6.3.5 Minimisation du critère . . . . . 110
6.4 Estimation et prise en compte de l'erreur du contour reprojeté . . . . . 112
6.4.1 Erreur de reconstruction . . . . . 112

6.4.2 Erreur de reprojection . . . 112
6.4.3 Contraindre la recherche du minimum par les régions $\Lambda_i$ . . . 113
6.5 Entrée ou sortie de l'objet occultant . . . 114
6.6 Discussion sur l'affinement par un contour actif . . . 114
6.7 Choix des images-clé . . . 115
6.7.1 Graphe d'aspects . . . 115
6.7.2 Choix des images-clé . . . 115
6.8 Cas des contours apparents . . . 116
6.8.1 « Reconstruction » d'un contour apparent . . . 116
6.8.2 Erreur de reprojection . . . 117
6.8.3 Réduction de l'erreur en ajoutant une image intermédiaire . . . 119
6.8.4 Conclusion sur l'erreur due aux contours apparents . . . 120

**Chapitre 7 Gestion semi-automatique des occultations: expérimentations 123**
7.1 Outil semi-automatique de détourage dans une image . . . 123
7.1.1 Contours actifs . . . 123
7.1.2 *Intelligent Scissors* . . . 124
7.1.3 Calcul du chemin proposé . . . 124
7.1.4 Résultats . . . 124
7.2 Séquence Stanislas . . . 125
7.2.1 Statue seule . . . 125
7.2.2 Statue, piédestal et marches . . . 128
7.3 Première séquence du chalet : influence des contours apparents . . . 136
7.4 Séquence du chalet . . . 136
7.5 Séquence du Loria . . . 142
7.6 Séquence de la voiture : objet occultant mobile et rigide . . . 145
7.7 Séquence Saint-Epvre : cas d'un panoramique . . . 145
7.8 Un outil de segmentation temporelle d'objets dans des séquences vidéos . . . 147
7.8.1 Colorisation . . . 147
7.8.2 Composition vidéo . . . 147
7.8.3 Réalité diminuée . . . 147
7.9 Conclusion . . . 148

**Chapitre 8 Conclusion 157**
8.1 Apports de la thèse . . . 157
8.1.1 Estimation de l'incertitude des points de vue . . . 157
8.1.2 Une approche semi-automatique pour une grande précision . . . 157

8.1.3 Un outil de suivi dans des séquences vidéo . . . . . 158
8.2 Limites de la méthode . . . . . 158
8.2.1 Une étape d'affinement local? . . . . . 158
8.2.2 Objets occultants avec un graphe d'aspects complexe . . . . . 158
8.3 Application interactive . . . . . 158

**Bibliographie** **161**

# Table des figures


1.1 Incrustation d'un dinosaure virtuel pour le film Jurassic Park. . . . . 4
1.2 Exemple d'illumination artificielle du Pont Neuf. A gauche, l'image réelle; à droite, l'image augmentée. . . . . 5
1.3 Visualisation temps-réel d'un volume ultra-sonore pour la chirurgie (UNC Chapel Hill, Dept. of Computer Science). . . . . 5
1.4 Le jeu RV-Border Guards: à gauche, la scène réelle; à droite, la scène telle qu'elle est vue par un joueur portant un HMD (Mixed Reality Systems Laboratory Inc.). 6
1.5 Système CyberSport d'Orad: a. Indication de distance; b. ajout du blason d'une équipe sur le terrain. . . . . 7
1.6 Notre scène augmentée et différents rayons lumineux : 1. rayon occulté; 2. éclairage direct; 3. interactions lumineuses entre objets réel et virtuel. . . . . 8
1.7 a: Panneau publicitaire occulté; b: Prédiction de l'apparence du panneau; c: Détection de l'occultation par les couleurs; d: Après suppression spatio-temporelle des erreurs (images extraites de [Zoghlami et al.96]). . . . . 10
1.8 a: Scène réelle; b: Personnage virtuel et, en noir, le modèle de la table réelle; c: Composition finale (images extraites de [Torre et al.00]). . . . . 11
1.9 Illustration de la difficulté de la gestion des occultations à l'aide d'une reconstruction automatique. . . . . 12
1.10 a. Image réelle; b. Modèle de l'objet à incruster; c. Incrustation de l'objet virtuel; d. Après prise en compte de l'occultation; e. Après prise en compte de l'éclairage direct; f. Après prise en compte de l'éclairage indirect; g. Masque d'occultation; h. Occultation erronée montrant les besoins de précision. . . . . 14

2.1 Le modèle sténopé de caméra. . . . . 15
2.2 Les paramètres internes. . . . . 17
2.3 La contrainte épipolaire. . . . . 19
2.4 Le problème du facteur d'échelle sur 3 images. . . . . 22
2.5 La géométrie trifocale. . . . . 23
2.6 Utilisation du tenseur trifocal pour retrouver la trajectoire de la caméra. . . . . 23
2.7 La mire de calibration utilisée au LORIA. . . . . 25
2.8 Illustration du principe de la méthode développée dans l'équipe. . . . . 26
2.9 Exemple d'erreurs de suivi obtenues sur la séquence du Pont Neuf (une erreur locale est obtenue pour les primitives 1 et 4 et une erreur aberrante pour la primitive 5). Les lignes continues sont les courbes 2D suivies dans la séquence, les lignes en pointillé la reprojection des courbes 3D. . . . . 27
2.10 Incrustation d'un véhicule à distance croissante de l'opéra. . . . . 29

2.11 Une petite erreur de reprojection au niveau de l'objet réel suivi ($r \simeq r'$) peut conduire à une incrustation complètement fausse ($v \neq v'$) pour un objet virtuel éloigné de l'objet ayant servi au calcul du point de vue. . . . . 29
2.12 Exemple de points d'intérêt et d'appariements obtenus pour les deux premières images de la séquence (pour plus de visibilité, seule une partie de l'image est représentée). Les flèches relient les points d'intérêt de l'image 1, qui est affichée, aux points d'intérêt correspondants dans l'image 2. . . . . 33
2.13 a: Trajectoire de la caméra retrouvée avec la méthode hybride. b et c: Deux images de la séquence avec l'incrustation d'une voiture. . . . . 33
2.14 Images 15, 100 et 150 de la Séquence Stanislas. . . . . 35
2.15 **Séquence Stanislas (76 images, 2066 points 3D)** a: Trajectoire estimée par notre méthode hybride; b et c: Trajectoires et évolutions du résidu moyen en fonction du temps de calcul pour le jeu de points sans *outliers* b: en optimisant seulement les paramètres externes, c: tous les coefficients des matrices de projection. 38
2.16 **Séquence Stanislas (76 images, 2590 points 3D dont** *outliers***)** Trajectoires et évolutions du résidu moyen en fonction du temps de calcul pour le jeu de points avec *outliers*; a: sans détection des *outliers*; b: avec détection des *outliers*, en optimisant seulement les paramètres externes, c: avec détection des *outliers*, en optimisant tous les coefficients des matrices de projection. . . . . 39
2.17 Images 0, 60 et 120 de la séquence du chalet. . . . . 39
2.18 **Séquence du chalet (120 images, 218 points)** a. Estimation initiale; b. Trajectoire retrouvée en optimisant seulement les paramètres externes; c. Trajectoire retrouvée en optimisant tous les coefficients des matrices de projection; d. comparaison de l'évolution du résidu moyen entre la séparation points/projections et le gradient conjugué. . . . . 40

3.1 Attribution du point 3D $\mathbf{M}$ au point 2D $\mathbf{m}'$. . . . . 48
3.2 a. Une image de la séquence Stanislas et les primitives 3D; b. ellipsoïde d'incertitude pour cette image (on a représenté également l'orientation de la caméra aux sommets). . . . . 49
3.3 a. Appariement de points utilisé par la méthode hybride; b. l'ellipsoïde obtenu par la méthode basée modèle, et plus petit, celui obtenu par la méthode hybride ($\epsilon = 1.0$); c. à gauche, l'ellipsoïde obtenu par la méthode hybride, à droite, celui obtenu par la méthode basée modèle, séparés pour une meilleure comparaison. . . 50
3.4 **Séquence Stanislas:** a. Points de vue estimés à l'aide de la méthode basée modèle; b. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = 1.0$; c. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = \Phi^*$. . . . . 51
3.5 **Séquence Stanislas:** a. Points de vue estimés à l'aide de la méthode hybride; b. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = 1.0$; c. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = \Phi^*$. . . . . 52
3.6 **Séquence du chalet:** a. Points de vue estimés à l'aide de la méthode hybride; b. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = 1.0$; c. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = \Phi^*$ (voir partie 3.4.2 pour une discussion sur le choix de $\epsilon$). . . . . 53
3.7 **Séquence Stanislas:** a. Points de vue obtenus après ajustement de faisceaux; b. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = {\Phi^i_{\text{ajust}}}^*$. . . . . 57
3.8 **Séquence du chalet:** a. Points de vue obtenus après ajustement de faisceaux; b. ellipsoïdes d'incertitude pour ces points de vue et $\epsilon = {\Phi^i_{\text{ajust}}}^*$. . . . . 58

3.9 **Séquence du chalet:** Points d'intérêt utilisés par l'ajustement de faisceaux, dans a: l'image 0; b: l'image 60; c: l'image 120. . . . . 58

4.1 Contours et masque d'occultation. . . . . 60
4.2 Contours occultants et occultations : cas des arêtes vives et des contours apparents. 61
4.3 a. Discrimination des contours apparents : en pratique, les deux cas sont difficiles à différencier; b. Les contours apparents sont souvent mal retrouvés par les détecteurs de contours. . . . . 61
4.4 Précision d'une reconstruction par triangulation suivant la disposition des caméras. 63
4.5 A gauche : pour une surface parallèle aux plans images, l'utilisation de fenêtres de corrélation rectangulaires est fondée ; à droite : pour une disposition différente, les fenêtres ne devraient pas être rectangulaires. . . . . 65
4.6 Fenêtres de corrélation à proximité d'un contour occultant (Zones 1 et 2 : la fenêtre de corrélation dans la caméra 2 recouvre les objets 1 et 2; zones 3 et 4 : la fenêtre de corrélation dans la caméra 1 recouvre les objets 1 et 2). . . . . 66
4.7 a. Fenêtres de corrélation classiques; b. Fenêtres utilisées par [Devernay et al.94]; c. par [Kanade et al.94]; d. par [Geiger et al.95]; e. par [Intille et al.94]. . . . . 67
4.8 Espace de recherche des mises en correspondance. . . . . 68
4.9 Appariement multi-résolution, les flèches représentant les appariements obtenus aux différentes résolutions (figure extraite de [Oisel98]). . . . . 69
4.10 a. et b. Paire stéréoscopique utilisée par [Intille et al.94]; c. Carte de profondeur obtenue (les zones noires sont les occultations détectées). . . . . 71
4.11 a. et b. Paire stéréoscopique utilisée par [Birchfield et al.98]; c. Carte de profondeur obtenue. . . . . 71
4.12 Avantages $\oplus$ et inconvénients $\ominus$ de l'utilisation de points de vue proches et éloignés pour la stéréoscopie binoculaire . . . . . 72
4.13 a. Paire d'images stéréoscopiques; b. en trait plein, la mise en correspondance entre $E$ et $E'$, en trait pointillé des mises en correspondance possibles. . . . . 72
4.14 Absence de variation d'intensités à la frontière d'un objet. . . . . 73
4.15 Disparition d'un motif répétitif entre deux images stéréoscopiques. . . . . 73
4.16 a. Dispositif à 6 caméras de [Okutomi et al.93]; b. Scène vue par la caméra de référence; la carte de profondeur obtenue. . . . . 74
4.17 a. Dome 3D; b. un exemple de reconstruction obtenue (figures tirées de [Narayanan et al.98]). 75
4.18 a. et b. Première et dernière image de la séquence utilisée par [Koch et al.98]; c. carte de profondeur obtenue; d. modèle reconstruit texturé. . . . . 75
4.19 a. Scène à reconstruire; b. Segments détectés et reconstruits; c. Résultat d'une triangulation de Delaunay contrainte sur ces segments (figures tirées de [Bruzzone et al.92]). 76
4.20 a. Maillage initial et modèle résultant; c. et d. Maillage obtenu par [Morris et al.00] et modèle résultant (figures tirées de l'article). . . . . 77
4.21 a. 4 images sur les 20 utilisés pour reconstruire deux tores imbriqués; b. Plusieurs étapes de la convergence de la surface $\mathcal{S}$. . . . . 79
4.22 Schéma de principe du *space carving*. . . . . 80
4.23 a : Scène à reconstruire, vue par 4 caméras; b : Reconstruction maximale (figures tirées de [Kutulakos et al.98]); c : Une autre reconstruction possible. . . . . 81
4.24 a : Surface à reconstruire, vue par 4 caméras; b. à j : évolution de la reconstruction par space carving quand on augmente la fréquence de la texture de la surface (figure tirée de [Seitz et al.99]). . . . . 82

4.25 a. Une des images utilisées dans Façade et les segments ayant un correspondant dans le modèle construit par l'utilisateur (b); c. Reprojection de ce modèle dans l'image; d. Vue du modèle texturé (images extraites de [Debevec et al.96]). . . . . 83
4.26 a. Interface de ImageModeler 2.0; b. Modèle reconstruit (images extraites de [Goncalves00]). . . . . 84

5.1 Incrustation d'un rectangle virtuel (figures extraites de [Berger97]). a : image réelle considérée; b : région $m_I$; c : carte de contours; d : résultat du suivi de courbes; e : contours étiquetés *Devant*; f : points étiquetés *Douteux*; g : champ de gradient créé par les contours étiquetés *Devant* et le contour actif initial; h et i : masque après la convergence du contour actif; j : résultat de l'incrustation. . . . . 89
5.2 Suivi de la statue sur la séquence Stanislas. . . . . 93
5.3 Principe de construction du clone. . . . . 95
5.4 Voxels dynamiques utilisés par [Ong et al.98] (figure extraite de l'article). . . . . 96
5.5 Exemple d'incrustation extrait de [Ong et al.98]. . . . . 97

6.1 a : Séquence d'exemple; b : **Choix des images-clé** $I_{\text{clé-1}}$, $I_{\text{clé-2}}$ et $I_{\text{clé-3}}$, et **détourage manuel**. . . . . 101
6.2 **Reconstruction des contours 3D** a : $\mathbf{C}_{1,2}$ et b :$\mathbf{C}_{2,3}$. . . . . 101
6.3 **Projection des contours 3D dans les images intermédiaires** a. entre $I_{\text{clé-1}}$ et $I_{\text{clé-2}}$ et b. entre $I_{\text{clé-2}}$ et $I_{\text{clé-3}}$. . . . . 101
6.4 **Correction d'une projection**. . . . . 101
6.5 Erreur produite par les contours apparents. . . . . 102
6.6 Appariement des contours 2d. . . . . 103
6.7 a et b : Deux images-clé de la séquence Stanislas, avec en noir, les points non appariés. . . . . 104
6.8 Reconstruction du contour détouré dans les images-clé de la figure 6.7. . . . . 104
6.9 Contour de la figure 6.8 après filtrage. . . . . 105
6.10 Estimation des parties non reconstruites du contour 3D. . . . . 105
6.11 a : Contour final $\mathbf{C}_{1,2}$; b : contour final $\mathbf{C}_{2,1}$. . . . . 106
6.12 Choix du contour à reprojeter en fonction de l'image intermédiaire. . . . . 106
6.13 Illustration de l'intérêt des deux contours. a : Reprojection de $\mathbf{C}_{2,1}$ dans la première image-clé; b : reprojection de $\mathbf{C}_{2,1}$ dans la deuxième image-clé. Les flèches indiquent les défauts majeurs de la prédiction. . . . . 107
6.14 Image 118 : un exemple de reprojection; une étape de correction est nécessaire pour retrouver la position exacte. . . . . 108
6.15 Correction de la prédiction par corrélation entre l'image-clé et l'image intermédiaire.109
6.16 Image 118 : en pointillés, la prédiction par reprojection; en trait plein, la correction sans la contrainte des régions $\Lambda_i$. . . . . 110
6.17 Images 98 (a) et 99 (b) : en pointillés, la prédiction par reprojection; en trait plein, la correction sans la contrainte des régions $\Lambda_i$. . . . . 111
6.18 a. Estimation de l'erreur de reconstruction; b. estimation de l'erreur de reprojection. . . . . 112
6.19 a : Un point du contour dans une image-clé; b : les droites épipolaires associées à ce point dans l'autre image-clé; c : les reprojections associées à ce point dans une image intermédiaire. . . . . 113
6.20 Exemple de correction pour un objet partiellement visible. . . . . 114

6.21 a. Un exemple de contour corrigé, utilisé pour initialiser un contour actif; b. contour actif après convergence. . . . . 115
6.22 Les 26 régions maximales appartenant au graphe d'aspects d'un cube. . . . . 116
6.23 Construction du point $I$. . . . . 117
6.24 Erreur engendrée par un contour apparent. . . . . 118
6.25 La balise a un rayon de 12 cm et est à une distance de 5m. . . . . 119
6.26 Valeurs de $e_{\max}$ en pixels pour différentes valeurs de $\frac{R}{D}$ et de $\Delta$ ($f = 1000$). . . . 119
6.27 Comparaison de l'erreur de reconstruction d'un contour apparent pour 2 et 3 images-clé. . . . . 120
6.28 Image 98 : résultat de la correction a : sans et b : avec la contrainte des régions $\Lambda_i$. Le contour reprojeté est en pointillés, la correction en trait plein. Les points corrigés qui sont restés dans leur région $\Lambda_i$ sont représentés par un point noir, ceux qui sont en dehors par une croix noire. . . . . 121
6.29 Image 99 : résultat de la correction a : sans et b : avec la contrainte des régions $\Lambda_i$. Le contour reprojeté est en pointillés, la correction en trait plein. Les points corrigés qui sont restés dans leur région $\Lambda_i$ sont représentés par un point noir, ceux qui sont en dehors par une croix noire. . . . . 122

7.1 Deux résultats d'utilisation de l'outil *Intelligent scissors* : les points correspondent aux clics souris, les traits pleins aux contours retrouvés automatiquement et les traits pointillés aux segments de droite définis quand la méthode automatique a échoué. . . . . 125
7.2 Résultats sur la séquence Stanislas, statue seule, 2 images-clé.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 126
7.3 Résultats sur la séquence Stanislas, statue seule, 3 images-clé.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 127
7.4 Temps de réalisation de la séquence Stanislas pour la statue seule (2 images-clé, 151 images, 250 points environ par contour). . . . . 128
7.5 Temps de réalisation de la séquence Stanislas pour la statue seule (3 images-clé, 151 images, 250 points environ par contour). . . . . 129
7.6 Temps de réalisation de la séquence Stanislas pour la statue et le piédestal (3 images-clé, 151 images, 780 points par contour environ, points de vue obtenus par la méthode basée modèle). . . . . 129
7.7 Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus par la méthode basée modèle.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 130
7.8 Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus par la méthode basée modèle.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 131

7.9 Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus après ajustement de faisceaux.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 132
7.10 Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus après ajustement de faisceaux.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 133
7.11 Résultats sur la première séquence du chalet (2 images-clé).
Trait pointillé : contour reprojeté; trait plein : contour corrigé. . . . . 134
7.12 Résultats sur la première séquence du chalet (3 images-clé).
Trait pointillé : contour reprojeté; trait plein : contour corrigé. . . . . 135
7.13 Temps de réalisation de la première séquence du chalet (31 images, 2 images-clé, 370 points par contour environ). . . . . 136
7.14 Temps de réalisation de la première séquence du chalet (31 images, 3 images-clé, 370 points par contour environ). . . . . 137
7.15 Temps de réalisation de la séquence du chalet (120 images, 6 images-clé. 750 points environ). . . . . 137
7.16 Séquence du chalet : les 6 images-clé retenues. . . . . 138
7.17 Séquence du chalet. Contours 3D reconstruits à partir des images a : 0 et 30; b : 31 et 40 c : 41 et 120. . . . . 139
7.18 Séquence du chalet : résultats obtenus avec 6 images-clé.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 140
7.19 Séquence du chalet : résultats obtenus en ajoutant l'image 50 comme image-clé supplémentaire.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 141
7.20 Temps de réalisation de la séquence du Loria (2 images-clé, 483 images, 170 points environ). . . . . 142
7.21 Séquence du Loria : résultats obtenus en utilisant les points obtenus par la méthode 3d2d. . . . . 143
7.22 Séquence du Loria : résultats obtenus en utilisant les points obtenus après ajustement de faisceaux.
La projection est très éloignée de la position attendue, jusqu'à ne plus recouvrir l'objet occultant à partir de l'image 360.
Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction. . . . . 144
7.23 Mosaïque utilisée pour le rendu des reflets sur la voiture virtuelle de la séquence du Loria. . . . . 145
7.24 Séquence de la voiture : images-clé, primitives 3D utilisées pour l'estimation des points de vue, résultats (trait pointillé : contour reprojeté; trait plein : contour corrigé) et incrustation. . . . . 146

7.25 Incrustation d'un avion virtuel dans la séquence Stanislas. . . . . 149
7.26 Incrustation d'une vache virtuelle dans la première séquence du chalet. . . . . 150
7.27 Séquence du chalet : incrustation d'objets virtuels. . . . . 151
7.28 Séquence du Loria : incrustation d'une voiture virtuelle. . . . . 152
7.29 Résultats sur la séquence Saint-Epvre. . . . . 153
7.30 a. Une image originale; b-d : images après modification des couleurs de la statue. 154
7.31 Composition de la statue avec une séquence d'images de synthèse. . . . . 154
7.32 Séquence de la chaîne : a. les deux images-clé; b. résultat du détourage dans une image intermédiaire; c. suppression de la chaîne par interpolation. . . . . 155
7.33 a. Résultat du détourage dans une des images; b. reprojection de la scène reconstruite sans l'objet à reconstruire; c. masque de l'objet supprimé; d. résultat final. . . . . 155

# Introduction générale

De nos cinq sens, la vue est sans conteste celui que nous mettons le plus à contribution pour percevoir notre environnement. Dès le début des années 80, la Réalité Virtuelle a mis à profit cette constatation pour immerger l'utilisateur dans un environnement complètement virtuel, grâce à des images de synthèse de cet environnement. La Réalité Augmentée, quant à elle, vise également à nous faire percevoir des éléments virtuels, tels que des informations sur l'environnement, des objets tridimensionnels, des modifications de l'éclairage... mais en laissant l'utilisateur dans le monde réel. Ces nouveaux éléments sont synthétisés par ordinateur et ajoutés aux images du réel perçues par l'utilisateur. Il existe un grand nombre d'applications de ce concept, que nous présentons de façon non exhaustive dans le chapitre d'introduction.

Une composition réaliste du réel et du virtuel demande de résoudre trois catégories de problème : la détermination du point de vue, la prise en compte des occultations entre éléments réels et virtuels, ainsi que des interactions lumineuses entre ces éléments. La première catégorie a été l'objet de nombreux travaux, alors que les deux dernières sont, en comparaison, nettement moins considérées par la littérature, bien qu'importantes pour obtenir une incrustation réaliste. Cette thèse traite de la prise en compte des occultations.

Une reconstruction tridimensionnelle de la scène réelle permettrait *a priori* de résoudre ce problème. En 1976, Marr et Poggio ont émis l'hypothèse qu'une telle reconstruction relevait du traitement de l'information issue d'images bidimensionnelles de cette scène, et qu'elle pouvait donc être obtenue automatiquement par un ordinateur. Plus de 20 ans de recherche plus tard, reconstruire automatiquement une scène à partir d'une séquence vidéo quelconque est encore difficile, et une prise en compte de ces occultations entièrement automatique reste malheureusement utopique, du moins à moyen terme. En revanche, et cela va dans le sens de la philosophie actuelle de développement des logiciels de vision, l'ordinateur peut apporter une grande aide à partir d'une intervention humaine réduite.

La méthode de gestion des occultations développée dans cette thèse repose sur une idée très simple. L'utilisateur détoure à l'aide d'un outil semi-automatique le ou les objets occultants dans quelques images de la séquence. À partir de ces contours détourés, on peut reconstruire des contours 3D qui sont ensuite reprojetés dans les autres images pour obtenir un détourage automatique. Une étape supplémentaire est toutefois nécessaire pour affiner la prédiction fournie par la reprojection, pour plusieurs raisons que nous expliquerons.

Cette méthode permet d'obtenir, même dans le cas de séquences vidéo complexes, un détourage précis d'objets réels pour chacune des images de la séquence, à partir d'une intervention manuelle modérée. Ce détourage peut être utilisé non seulement pour la gestion des occultations, mais également pour d'autres applications utiles pour la post-production.

# Chapitre 1

# La Réalité Augmentée: définitions, applications et problématiques

Dans ce chapitre, nous présentons différents types d'application de la Réalité Augmentée. Nous décrivons également la problématique sous-jacente et définissons les objectifs de notre travail.

## 1.1 Définitions

Si, d'une façon générale, le terme de Réalité Augmentée (RA) désigne la fusion du Réel et du Virtuel, on peut diviser les applications de la RA en deux grandes familles, selon que cette fusion se fasse en temps réel ou non.

Dans les applications en temps réel telles qu'elles sont décrites par exemple dans [Azuma97] ou [Klinker et al.97], l'utilisateur perçoit des objets virtuels en même temps que l'environnement réel dans lequel il évolue. Cette perception se fait généralement par l'intermédiaire d'un casque de RA (en anglais Head Mounted Display ou HMD), semblable aux casques de Réalité Virtuelle, mais permettant de visualiser le monde réel en plus des images de synthèse des éléments virtuels.

De nombreux auteurs utilisent également le terme de Réalité Augmentée, dans un contexte légèrement différent, puisque dans leurs travaux, les images réelles et virtuelles sont composées sans la contrainte temps-réel [Thalmann et al.97, Faugeras98, Zisserman et al.99]. On parle également de *post-production*, puisque l'insertion des images virtuelles se fait lors d'une étape postérieure à l'acquisition de la séquence vidéo. Le temps consacré au traitement d'une image n'est alors plus contraint, et le fait de pouvoir considérer la séquence dans sa globalité permet d'obtenir de nombreuses informations sur cette séquence, comme nous le verrons dans cette thèse.

## 1.2 Applications

Les applications de la post-production sont d'ores et déjà nombreuses: ce sont principalement des effets spéciaux pour le cinéma ou les publicités, mais également des études d'impact ou d'éclairage en architecture. Elles réclament cependant encore une interactivité forte avec l'utilisateur. Les applications potentielles de la RA en temps-réel sont multiples, dans des domaines aussi variés que la médecine, la maintenance industrielle, le jeu... même si, à l'heure actuelle, peu de projets ont abouti sur des applications effectives. Nous présentons ici quelques applications, présentes ou futures.

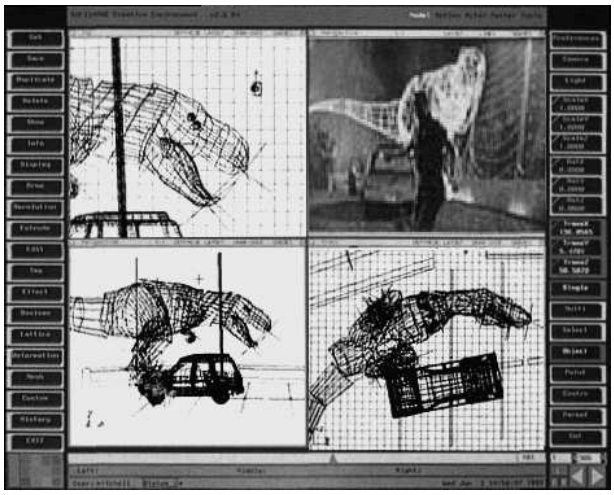


FIG. 1.1 – *Incrustation d'un dinosaure virtuel pour le film Jurassic Park.*

### 1.2.1 Effets spéciaux

Une utilisation directe de la post-production est évidemment les effets spéciaux pour le cinéma, la publicité ou les clips vidéo, et on ne compte plus les films intégrant des images de synthèse à des séquences vidéo.

La figure 1.1 illustre l'étape d'incrustation, pour le film *Jurassic Park*: les infographistes ont tout d'abord modélisé un dinosaure et son déplacement, puis connaissant la position de la caméra grâce à un bras articulé, ils utilisent un logiciel de synthèse d'images pour effectuer le rendu du dinosaure. Comme, lors de cette séquence, le dinosaure passe derrière une voiture (réelle), cette voiture a également été entièrement modélisée pour déterminer les parties non visibles du dinosaure. En revanche, l'acteur, qui cache également partiellement le dinosaure, a été ajouté après avoir été filmé séparément devant un fond bleu permettant la composition d'images (voir partie 1.4.2).

### 1.2.2 Visualisation de projets

La RA est aussi utilisée en architecture pour les études d'impact [Maver et al.85] ou l'illumination artificielle de monuments [Chevrier et al.95]. En particulier, le projet des ponts de Paris, qui a été confié en 1995 à l'équipe ISA dans le cadre d'une recherche soutenue par le PIR-Ville du CNRS, et en collaboration avec le CRAI à Nancy, vise à illuminer virtuellement un certain nombre de ponts situés autour de l'Île de la Cité. L'objectif est d'évaluer visuellement l'impact sur l'environnement de plusieurs projets d'illumination. Plutôt que d'effectuer des tests *in situ*, il est plus pratique et moins coûteux d'utiliser des images de synthèse. L'équipe a donc remplacé, dans des séquences vidéo, le pont réel par un pont virtuel, illuminé selon le système d'éclairage que les architectes souhaitent évaluer (figure 1.2).

Autre exemple: la société Renault a beaucoup utilisé la RA pour visualiser ses modèles de voitures dans des décors naturels, avant leur réalisation effective. Une des premières créations a été celui du projet Racoon [RD94] en 1994, où l'incrustation du modèle de synthèse a été effectuée en filmant une vraie voiture bardée de marqueurs, utilisés pour définir la position de la voiture virtuelle dans la séquence vidéo. Le dernier film créé par Renault Design, Escapade, met en scène le futur véhicule Koléos, et utilise (avec l'aide de la société HybridMC) des capteurs mécaniques au niveau de la caméra pour se passer de voiture réelle et de marqueurs visuels

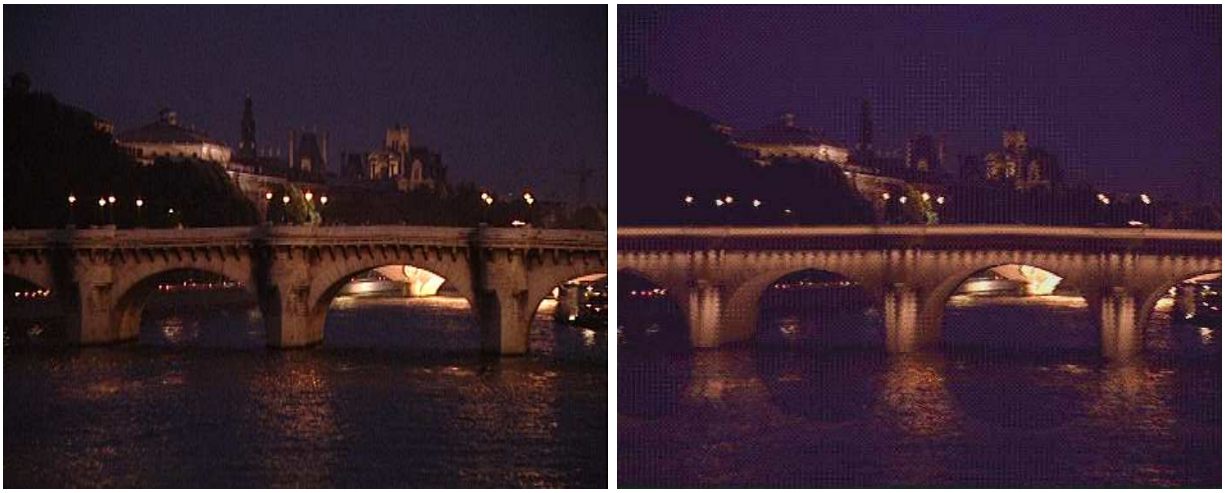


FIG. 1.2 – *Exemple d'illumination artificielle du Pont Neuf. A gauche, l'image réelle; à droite, l'image augmentée.*

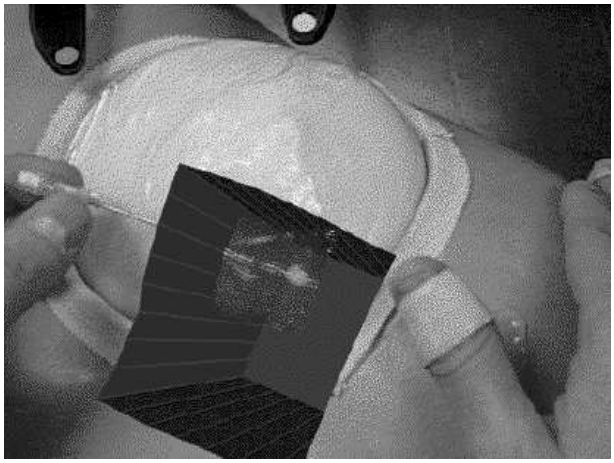


FIG. 1.3 – *Visualisation temps-réel d'un volume ultra-sonore pour la chirurgie (UNC Chapel Hill, Dept. of Computer Science).*

[Schmitt00]. Ces films sont utilisés à la fois par les ingénieurs pour évaluer le design du véhicule dans un environnement réel, et pour présenter les nouveaux modèles au grand public.

### 1.2.3 Médecine

De nombreux projets de RA en temps-réel visent à permettre aux chirurgiens de visualiser des données obtenues par des capteurs utilisés en médecine (comme des images ultra-sonores, la tomographie 3D, les images à résonance magnétique, etc...), superposées au corps du patient. Dans un projet exposé dans [State et al.96], par exemple, le chirurgien peut guider une aiguille vers une tumeur, en se basant sur des données ultra-sonores du patient, rendues visibles grâce à un HMD (figure 1.3).

Plus généralement, de nombreux travaux visent à fusionner automatiquement diverses sources d'informations 3D et 2D, comme le recalage d'une image tomographique ou à résonance magnétique dans une séquence d'images à rayons X [Feldmar et al.97, Roth et al.99] ou encore le


FIG. 1.4 – *Le jeu RV-Border Guards: à gauche, la scène réelle; à droite, la scène telle qu'elle est vue par un joueur portant un HMD (Mixed Reality Systems Laboratory Inc.).*

recalage de reconstructions 3D de la vascularisation cérébrale dans des images d'angiographie numérique soustraite [Kerrien et al.99]. Ainsi, en neuroradiologie interventionnelle par exemple, les travaux de Kerrien ont pour but de permettre au radiologue de savoir à tout instant où se trouve son cathéter dans le corps du patient.

### 1.2.4 Maintenance industrielle

En milieu industriel, les futures applications de la RA devraient permettre de simplifier considérablement certaines tâches d'assemblage, de maintenance ou de réparation. Ainsi, des chercheurs de Boeing ont développé un système permettant de guider les techniciens dans l'assemblage de réseaux électriques pour les avions [Curtis et al.98]. Avant que ce système ne soit mis en place, les techniciens plaçaient les fils selon des schémas gravés sur des panneaux. Un 747 comportant plus de 1000 réseaux électriques, et les réseaux étant différents d'un avion à l'autre, cela impliquait des coûts considérables pour le stockage, le transport et la construction des panneaux. Avec le nouveau système, tous les réseaux sont stockés en mémoire, et il suffit au technicien de choisir le réseau approprié, qu'il peut alors visualiser par-dessus un panneau vierge en s'équipant d'un HMD. Le même principe est mis en œuvre dans [Reiners et al.98] pour l'assemblage d'un mécanisme de fermeture de porte pour automobile. Des indications virtuelles peuvent aussi être ajoutées pour désigner certaines pièces d'objets manufacturés, comme des photocopieuses [Feiner et al.93] ou des moteurs [Rose et al.94, Ravela et al.96], et faciliter ainsi les opérations de maintenance ou de réparation sur ces objets.

### 1.2.5 Jeu

Les loisirs seront sans doute également l'un des domaines de développement futur de la RA. L'un des projets les plus spectaculaires, RV-Border Guards, a été présenté au *workshop* international de Réalité Augmentée [Ohshima et al.99]: il s'agit d'un jeu multi-joueurs, où chaque joueur porte un HMD, qui lui permet de voir les autres joueurs et la scène réelle, ainsi que les éléments virtuels du jeu, c'est-à-dire les armes, les casques, les ennemis... (figure 1.4). Cette application est prometteuse et permet d'envisager de nouveaux types de jeux.

a.  b. 

FIG. 1.5 – *Système CyberSport d'Orad: a. Indication de distance; b. ajout du blason d'une équipe sur le terrain.*

### 1.2.6 Télévision

Des sociétés comme Orad [Orad00] ou Symah Vision [Symahvision00] ont développé des systèmes permettant d'insérer des éléments virtuels en direct dans des diffusions télévisées notamment lors de la retransmission d'événements sportifs. Ces éléments virtuels peuvent être des indications de distance (d'un joueur au but, par exemple, figure 1.5.a) ou des images incrustées sur le terrain de jeu (figure 1.5.b).

Cette liste d'applications de la RA n'est pas exhaustive, et on pourrait encore citer de nombreux autres projets, comme la vidéo-conférence [Kato et al.99], ou le guidage d'un nouvel arrivant dans un bâtiment ou un lieu touristique [Azuma97].

## 1.3 Problématique

La RA est une discipline relativement récente, dont la majorité des avancées a eu lieu ces cinq dernières années. Des systèmes logiciels de composition d'images en phase de post-production (Maya, Lightwave...) sont aujourd'hui utilisés aussi bien par les producteurs d'effets spéciaux que par le grand public. Néanmoins, comme nous le verrons, une grande part de travail manuel subsiste, et les studios de cinéma utilisent encore des outils spécifiques et coûteux comme des bras mécaniques ou des « fonds bleus ». Les applications à base de HMD sont par contre plus expérimentales que réellement utilitaires. Si des compagnies aériennes telles que Boeing et McDonnell Douglas utilisent ou s'intéressent de très près à cette technologie, celle-ci n'est pas encore suffisamment bon marché, ni suffisamment flexible et confortable visuellement pour être appliquée dans l'industrie. Les applications médicales utilisant un HMD sont encore anecdotiques et de nombreux problèmes restent à résoudre avant qu'un acte chirurgical puisse être réalisé à l'aide d'un casque de RA, comme l'inconfort des casques HMD, encore trop lourds, et le manque de précision du positionnement entre les parties réelles et virtuelles.

Qu'il s'agisse d'un système temps-réel ou de post-production, un système de RA doit résoudre le problème suivant : étant donné un (ou plusieurs) objet(s) virtuel(s) dont on connaît le modèle 3D, la position, les propriétés photométriques, etc..., comment modifier l'image réelle pour créer une nouvelle image, dite image augmentée, représentant la scène réelle et les objets virtuels ?

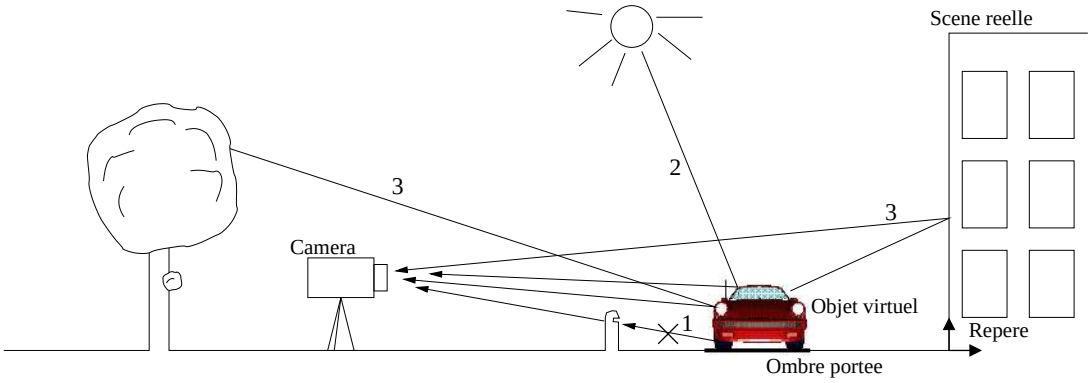



FIG. 1.6 – *Notre scène augmentée et différents rayons lumineux: 1. rayon occulté; 2. éclairage direct; 3. interactions lumineuses entre objets réel et virtuel.*

Nous allons voir que l'on peut décomposer les points à résoudre en trois catégories, que nous présentons ici sur un exemple consistant à incruster une voiture virtuelle près de notre centre de recherche (voir figure 1.6). Nous ferons ensuite le point sur l'avancée des travaux pour chacune de ces catégories.

### 1.3.1 Point de vue

Dans la pratique, les objets virtuels sont positionnés dans un repère associé à la scène, et il faut tout d'abord déterminer le point de vue de la caméra dans ce repère, c'est-à-dire la position et l'orientation, pour retrouver les rayons lumineux passant par cette caméra. En plus de ces paramètres externes, nous devons également connaître les caractéristiques internes de la caméra (comme la distance focale par exemple), afin d'établir la relation entre les rayons lumineux parvenant à la caméra et les points de l'image. Les paramètres internes et externes définissent une caméra virtuelle qui peut alors être utilisée pour synthétiser les images de l'objet virtuel (figure 1.10.c).

### 1.3.2 Occultations

Comme dans notre exemple, il se peut qu'un objet réel soit positionné entre la caméra et l'objet virtuel (figure 1.10.d). Si on néglige ce phénomène en superposant simplement l'image de l'objet virtuel à l'image réelle, cet objet semble devant l'objet occultant (figure 1.10.c). En général, il n'y a pas intersection entre l'objet virtuel et la scène réelle (dans le cas contraire, soit l'objet virtuel est mal placé, soit il s'agit d'un effet volontaire), et la partie visible de l'objet incrusté est sa projection privée de la silhouette bidimensionnelle de l'objet réel dans l'image, appelée *masque d'occultation* (cf. figure 1.10.g).

### 1.3.3 Éclairage

Une incrustation réaliste demande également de tenir compte de l'éclairage de l'objet virtuel par les sources lumineuses présentes dans la scène réelle, ce qui est en fait un problème de synthèse d'images, à condition de connaître les caractéristiques des sources lumineuses (position, couleur, puissance...). De plus, certains points de la scène réelle qui étaient éclairés ne le sont plus si l'objet virtuel est inséré entre eux et une source lumineuse. Ces points constituent l'ombre portée de l'objet virtuel, un élément important pour le réalisme puisqu'elle est utilisée par l'œil

humain afin de mieux évaluer la position spatiale de l'objet. Pour déterminer cette ombre, il faut disposer (en plus des caractéristiques des sources lumineuses) à la fois d'un modèle 3D et des propriétés des matériaux de la scène où est projetée cette ombre (voir figure 1.10.e).

En plus de l'éclairage de l'objet par les sources réelles (que l'on peut appeler *éclairage direct*), il existe également un *éclairage indirect* puisque la plupart des matériaux, même s'ils n'émettent pas de lumière propre, réfléchissent une partie de la lumière qu'ils reçoivent. Dans notre cas, cet éclairage se fait évidemment à la fois dans le sens scène réelle vers objet virtuel et le sens objet virtuel vers scène réelle (figure 1.10.f). Ici, il faut cette fois connaître la scène réelle dans sa globalité (géométrie et propriétés des matériaux) pour effectuer le rendu de l'objet virtuel et modifier l'image réelle là où l'objet virtuel ré-émet de la lumière.

On remarquera que connaître la géométrie de la scène réelle permettrait de résoudre, du moins en partie, plusieurs problèmes soulevés ici. Malheureusement, on ne dispose généralement pas d'un tel modèle, et l'acquérir (à l'aide d'un laser par exemple) se révèle très contraignant, voire impossible. Une autre solution, qui consisterait à dériver le modèle de la scène réelle à partir de la séquence vidéo, reste très délicate, comme nous le verrons au chapitre 4. C'est pourquoi on se contente généralement d'une reconstruction de la partie pertinente de la scène pour le problème considéré. Dans le cas des occultations, on peut également, si le contexte le permet, se limiter au problème bidimensionnel qu'est la recherche du masque d'occultation. Nous allons voir maintenant ce qui a été proposé dans la littérature pour résoudre les trois points que nous venons de soulever.

## 1.4 Avancées des travaux en RA

### 1.4.1 Point de vue

Une très large majorité des travaux en RA ont porté sur la détermination du point de vue, et de nombreuses approches ont été proposées pour la post-production et pour le temps-réel. Les méthodes les plus directes utilisent de capteurs mécaniques ou magnétiques, qui permettent de connaître directement la position de la caméra, ou alors des marqueurs visuels artificiels, facilement identifiables dans l'image et dont la position spatiale est connue, ce qui permet de déduire la position de la caméra.

D'autres méthodes (dites basées modèle) se servent de la connaissance *a priori* du modèle 3D d'un ou de plusieurs objets présent(s) dans la scène, pour retrouver la position de la caméra par rapport à cet objet. Il existe également des méthodes (dites *méthodes basées image* ou *auto-calibration*) qui suivent des éléments 2D (comme des points ou des segments) dans la séquence d'images pour en déduire le mouvement de la caméra.

Ces deux dernières approches sont les moins contraignantes puisqu'elles ne requièrent pas de matériel spécifique. Nous décrirons, dans le chapitre 2 les approches basées image, la méthode basée modèle développée dans notre équipe, et l'extension de cette méthode qui permet d'utiliser à la fois des données 3D et 2D.

### 1.4.2 Occultations

En dépit de son importance, relativement peu de travaux (par rapport au calcul du point de vue) ont porté explicitement sur les occultations en RA. Dans la pratique, la post-production se contente parfois d'effectuer un détourage manuel des objets occultants dans toutes les images de la séquence (ou éventuellement d'interpoler les masques 2D détourés dans certaines images-clé).

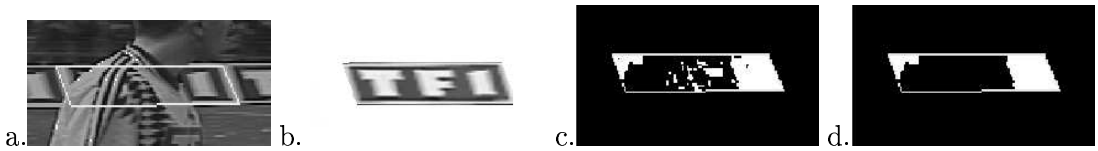



FIG. 1.7 – *a: Panneau publicitaire occulté; b: Prédiction de l'apparence du panneau; c: Détection de l'occultation par les couleurs; d: Après suppression spatio-temporelle des erreurs (images extraites de [Zoghlami et al.96]).*

Si on exclut cette solution simpliste et fastidieuse, on peut distinguer chez les systèmes existants quatre approches.

### Seuillage de couleurs (*chroma keying*)

Si l'objet occultant est devant un fond uni d'une couleur qui n'est pas présente sur cet objet, on peut retrouver sa silhouette par seuillage de couleurs: c'est le principe des « fonds bleus » (*blue screen* en anglais). Les studios virtuels permettent, en filmant un présentateur de télévision devant un tel fond bleu, d'intégrer ce présentateur en temps réel dans un environnement virtuel [Gibbs et al.96, Park et al.98], de façon cohérente, grâce à leur connaissance de la position de la caméra par des capteurs magnétiques ou mécaniques. C'est le principe également utilisé par Orad (cf. figure 1.5.b): les joueurs occultent systématiquement l'incrustation, et la couleur unie de la pelouse permet de déterminer leurs silhouettes.

La post-production peut également utiliser des fonds bleus d'une façon un peu différente, en filmant la séquence deux fois : la première prise constitue la séquence réelle, la deuxième est réalisée après avoir placé un fond bleu derrière les objets occultants permettant de les détourer automatiquement, le mouvement de la caméra étant évidemment le même entre les deux prises.

### Approche basée image

Zoghlami [Zoghlami et al.96] ont proposé une approche basée image pour ne plus se limiter à des objets de couleur uniforme, dans le cadre d'un projet consistant à remplacer un panneau publicitaire existant par le panneau virtuel d'une autre publicité, lors d'une retransmission télévisée d'événements sportifs. Connaissant l'image présente sur le panneau qu'on souhaite remplacer, la position de ce panneau et celle de la caméra, ils peuvent prédire l'apparence de ce panneau non occulté dans l'image considérée (figure 1.7.b). En comparant (de façon robuste) les couleurs de la prédiction et de l'apparence réelle, on peut retrouver la majeure partie des zones occultées du panneau par les joueurs (figure 1.7.c). Les zones où le joueur occultant le panneau et la prédiction sont de la même couleur ne sont donc pas directement détectées, mais elles peuvent être retrouvées par cohérence spatio-temporelle si la caméra ou le joueur bougent (figure 1.7.d). Ces travaux utilisent donc le fait qu'on peut, dans ce contexte, prévoir l'apparence de l'objet susceptible d'être occulté, ce qui peut être réalisé facilement et précisément puisque cet objet est plan.

### Approche basée modèle

[Balcisoy et al.00, Torre et al.00], dans le cadre d'un jeu de Dames en temps-réel contre un joueur virtuel, utilisent le modèle et la position des objets occultants, ici la table sur laquelle repose le plateau de jeu (voir figure 1.8). Il suffit alors de connaître la position de la caméra pour

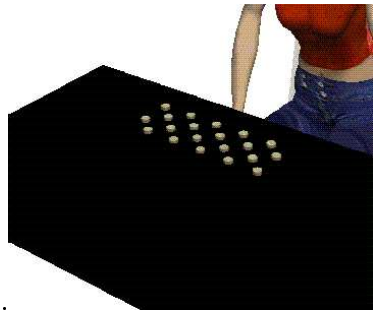
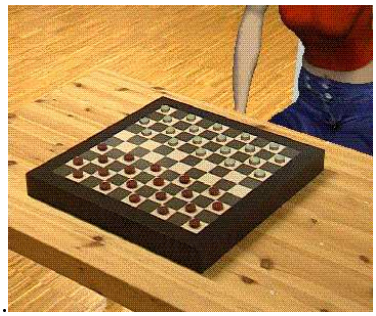
a.  b.  c. 

FIG. 1.8 – *a: Scène réelle; b: Personnage virtuel et, en noir, le modèle de la table réelle; c: Composition finale (images extraites de [Torre et al.00]).*

retrouver les silhouettes et les profondeurs des objets occultants. Le contexte contraint de cette application justifie ici la connaissance *a priori* de ces modèles. Pour d'autres applications plus générales, ces modèles ne sont pas forcément disponibles.

Pour la post-production, certains logiciels de synthèse d'images (Lightwave, 3D-Studio...) ou de reconstruction 3D (ImageModeler, PhotoModeler...) proposent à l'utilisateur de modéliser les objets réels en s'aidant des images de la séquence pour ensuite retrouver automatiquement les occultations (comme la voiture de la figure 1.1). Ce système reste contraignant pour l'utilisateur, et limité à des objets de formes relativement simples.

#### Approche basée reconstruction

Nous verrons que l'on peut envisager de reconstruire la scène réelle à partir d'images de cette scène prises de points de vue différents. Nous détaillerons dans le chapitre 4 les différentes techniques existantes, portant sur la reconstruction en vision par ordinateur d'une façon générale, et dans le chapitre 5, les travaux portant sur une gestion des occultations basée reconstruction. Cependant, de telles reconstructions ne permettent pas de retrouver les occultations de manière très précise. Nous présentons figure 1.9 une image extraite d'une vidéo de [Kanade et al.95]. Si le contexte de leur application est légèrement différent du notre puisqu'elle consiste à incruster des personnes réels dans un environnement virtuel (et non l'inverse), cette figure permet d'illustrer l'imprécision des reconstructions. La personne a été reconstruite grâce à un système de 5 caméras dont les positions sont connues très précisément. Malgré ce dispositif mis en œuvre, les occultations retrouvées manquent de précision; nous verrons pourquoi au chapitre 4.

### 1.4.3 Éclairages directs et indirects

Les premiers travaux sur l'illumination en RA ont sans doute été effectués par [Fournier et al.93]. Le modèle géométrique de la scène réelle était supposé connu et la position des sources lumineuses retrouvées manuellement. De plus, l'illumination globale était effectuée à partir des images de la séquence, ce qui suppose que cette séquence filme une large portion de la scène.

Pour éviter ces différentes limites, [Drettakis et al.97] ont proposé de retrouver la géométrie et la photométrie de la scène à partir de plusieurs mosaïques, moyennant une intervention raisonnable de l'utilisateur (une mosaïque est un ensemble d'images, prises d'une même position mais selon des directions différentes).

[Debevec98] a montré qu'on ne pouvait en fait pas se contenter d'images prises selon une seule exposition pour retrouver précisément les propriétés des matériaux de la scène réelle. Il décompose

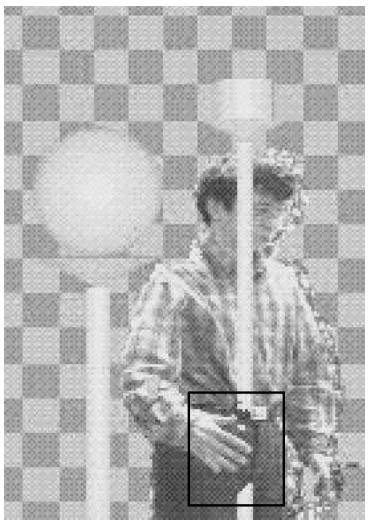


FIG. 1.9 – *Illustration de la difficulté de la gestion des occultations à l'aide d'une reconstruction automatique.*

d'abord la scène réelle en une scène locale et une scène distante. La scène locale est reconstruite manuellement, et les propriétés de ses matériaux sont calculées à partir de photographies d'une sphère réfléchissante selon différents temps d'expositions, ce qui permet de retrouver les ombres portées et les inter-réflexions près de l'objet virtuel. La scène distante, quant à elle, est supposée ne pas être affectée par l'objet virtuel, et les auteurs déterminent, toujours à l'aide de la même sphère, comment elle affecte l'objet.

[Sato et al.99] utilisent des objectifs *fisheye* (c'est-à-dire avec une ouverture de champ de 180 degrés) pour la reconstruction de la scène réelle par stéréoscopie à partir de plusieurs prises de vue avec cet objectif. En changeant également le temps d'exposition, ils retrouvent ainsi la géométrie et la photométrie de l'ensemble de la scène. Le rendu est ensuite effectué grâce à un algorithme de radiosité.

[Loscos et al.00] ont développé un système permettant de modifier l'éclairage d'une scène réelle, système qui n'est plus limité à l'insertion d'objets n'émettant pas de lumière propre, puisqu'il permet d'insérer également des sources lumineuses, ou de modifier les sources lumineuses déjà présentes dans la scène. Cette scène et la position des sources lumineuses réelles sont modélisées de manière semi-interactive comme dans [Drettakis et al.97]. Cette connaissance leur permet de retrouver, à l'aide d'un rendu en radiosité, les ombres projetées par les sources réelles, et de pouvoir ainsi simuler correctement la modification de l'intensité de ces sources. Grâce à un algorithme hiérarchique de radiosité permettant de prendre en compte des environnements dynamiques, les calculs requis sont suffisamment rapides pour que le système soit interactif.

## 1.5 Objectifs de la thèse

L'objectif de cette thèse est de définir un système permettant de prendre en compte les occultations. On l'a vu, les systèmes actuels sont encore très contraignants ou peu précis, et nous nous proposons, dans cette thèse, de diminuer la tâche de l'utilisateur tout en gardant la précision suffisante.

Nous nous sommes plus particulièrement intéressés à la gestion des occultations pour des applications de post-production, pour plusieurs raisons: les travaux de recherche portant sur la

détermination du point de vue ont montré récemment qu'en post-production, on pouvait retrouver précisément la position de la caméra à partir des seules images de la séquence à augmenter, en prenant en compte la globalité de la séquence. Nous nous proposons de ne travailler pareillement que sur cette séquence. Un second argument pour le choix d'un contexte post-production est que la reconstruction des objets occultants se fait à partir d'images de la scène prises de points de vue différents, ce qui serait délicat d'effectuer dans un contexte temps-réel.

Pour être utilisé en post-production, un système gérant les occultations doit être :

- **général :** Il doit pouvoir prendre en compte des objets occultants de forme quelconque, en particulier non polyédriques, et ne doit pas imposer de contraintes sur le mouvement de la caméra.
- **précis :** Le détourage de l'objet occultant doit être particulièrement précis. En effet, une légère erreur à ce niveau est très aisément perceptible par l'œil humain. Pour illustrer ce phénomène, nous avons légèrement déplacé le masque d'occultation (de l'ordre de quelques pixels) de notre exemple, le résultat est visible figure 1.10.h. De plus, ce système doit tenir compte des détails fins de l'objet occultant.
- **robuste :** Le système ne doit pas être gêné par la présence d'objets mobiles dans la scène. Un autre critère de robustesse est la prise en compte du fait que les points de vue ne sont pas connus très précisément en pratique. Nous verrons que ce point est très important et qu'il est souvent négligé par les algorithmes de reconstruction.
- **intuitif, avec une interaction réduite :** Nous verrons qu'une intervention de l'utilisateur reste inévitable. Néanmoins, il faut que celle-ci reste faible, et intuitive.

Le système développé dans cette thèse répond à la plupart de ces critères. La robustesse peut être mise en défaut si le mouvement de la caméra est trop rapide, ou si il y a trop d'objets mobiles, mais ceci est dû à la difficulté de retrouver alors la position de la caméra. De plus, les objets occultants doivent être rigides, mais éventuellement mobiles.

Dans certaines conditions que nous détaillerons, la précision peut être mise en défaut, mais ce problème peut être résolu, du moins en partie, au prix d'une interactivité plus forte.

## 1.6 Plan de la thèse

L'étape de détermination des points de vue est importante, non seulement pour permettre d'insérer l'objet virtuel au bon endroit, mais également parce que les points de vue sont utilisés par la méthode développée dans cette thèse. Nous commençons donc par décrire, dans le chapitre 2, le modèle de caméra utilisé, les principes sur lesquels sont bâtis les algorithmes d'estimation du point de vue, enfin les algorithmes développés dans notre équipe.

Le chapitre 3 montre ensuite comment estimer l'erreur de ces algorithmes: cette erreur est utilisée par notre méthode lors de l'étape de correction.

Puis, nous présentons, dans le chapitre 4, un état de l'art de la reconstruction tridimensionnelle en vision par ordinateur, dans l'optique de résoudre notre problème. Nous verrons également les méthodes plus spécifiques à la gestion des occultations au chapitre 5.

Enfin, le chapitre 6 décrira la méthode proposée après avoir tiré des conclusions des deux précédents chapitres, et les expérimentations de cette méthode seront présentées chapitre 7. Le chapitre 8 conclura ce manuscrit.

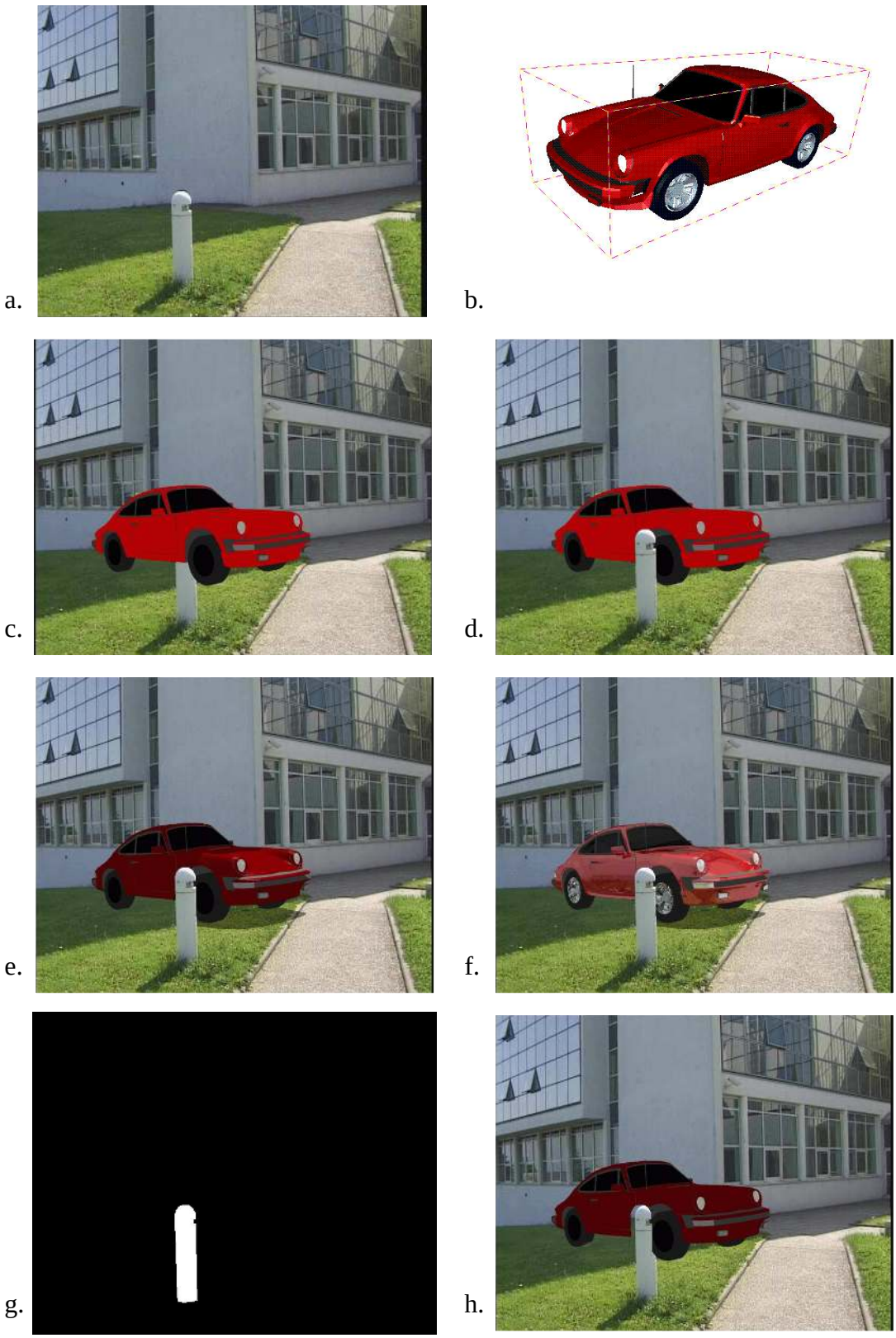



FIG. 1.10 – *a. Image réelle; b. Modèle de l'objet à incruster; c. Incrustation de l'objet virtuel; d. Après prise en compte de l'occultation; e. Après prise en compte de l'éclairage direct; f. Après prise en compte de l'éclairage indirect; g. Masque d'occultation; h. Occultation erronée montrant les besoins de précision.*

# Chapitre 2

# Calcul des points de vue

Si le calcul du point de vue n'est pas le sujet principal de cette thèse, il reste une étape déterminante pour obtenir une composition cohérente, et comme nous le verrons au chapitre 6, un point important sur lequel s'appuie notre système de résolution des occultations. Nous nous limiterons cependant ici à une présentation générale des principes de calcul basé image, et à la description de la méthode du calcul du point développée dans notre équipe, qui est essentiellement basée modèle, mais qui intègre également des éléments image.

Précisons nos contributions pour ce chapitre : le calcul du point de vue basé modèle de l'équipe a été développé initialement par Simon [Simon99], mais nous avons participé à l'élaboration de son amélioration, que nous appelons méthode hybride. Enfin, nous avons expérimenté des techniques d'ajustement de faisceaux.

## 2.1 Géométrie d'une caméra : le modèle sténopé

Pour incruster convenablement un objet virtuel dans une image réelle, le rendu de celui-ci doit être effectué selon une caméra virtuelle possédant les mêmes caractéristiques que la caméra réelle. Il faut pour cela adopter un modèle de caméra qui soit le plus réaliste possible : le modèle sténopé a l'avantage d'être simple tout en restant relativement fidèle à la réalité. Il est utilisé à la fois par la communauté vision et la communauté synthèse d'images, même si sa paramétrisation diffère chez ces deux communautés. Nous présentons dans la suite uniquement la paramétrisation utilisée en vision.

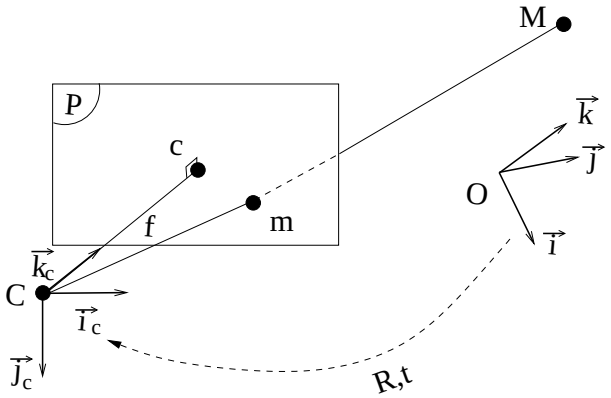



FIG. 2.1 – *Le modèle sténopé de caméra.*

Le modèle sténopé est représenté en figure 2.1 : un point $\mathbf{M}$ se projette en un point $\mathbf{m}$ sur le plan $\mathcal{P}$, selon une projection perspective de centre $\mathbf{C}$, situé à une distance $f$ non nulle du plan $\mathcal{P}$. Le plan $\mathcal{P}$ est appelé *plan image*, le point $C$ *centre optique* et la distance $f$ *distance focale*. Cette projection peut se décomposer en deux étapes : les coordonnées du point $\mathbf{M}$ sont exprimées dans le repère de la caméra en fonction des paramètres externes de la caméra, puis ce point est projeté dans le plan image en fonction des paramètres internes de la caméra.

### 2.1.1 Les paramètres externes

Si le point $\mathbf{M}$ a pour coordonnées $(X,Y,Z)$ dans le repère $(\mathbf{O}, \vec{i}, \vec{j}, \vec{k})$ de la scène, ses coordonnées $(X_c,Y_c,Z_c)$ dans le repère $(\mathbf{C}, \vec{i_c}, \vec{j_c}, \vec{k_c})$ de la caméra sont données par la relation :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t} = (\mathbf{R} \quad \mathbf{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.1}$$

où $(\mathbf{R} \quad \mathbf{t})$ exprime le déplacement rigide entre les deux repères (rotation et translation). La rotation $\mathbf{R}$ est souvent exprimée en fonction des angles de rotation $\gamma$, $\beta$, $\alpha$ autour respectivement des trois vecteurs de base $\vec{i}, \vec{j}, \vec{k}$ :

$$\mathbf{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix}.$$

Ces angles sont appelés *angles d'Euler*. Les paramètres du changement de repère sont donc au nombre de six : les trois angles d'Euler de $\mathbf{R}$ et les trois composantes du vecteur de translation $\mathbf{t}$. Ces paramètres, définissant l'orientation et la position de la caméra dans le repère de la scène, sont les paramètres externes (appelés également paramètres *extrinsèques*) de la caméra.

### 2.1.2 Les paramètres internes

Dans le repère de la caméra, la projection $\mathbf{m}$ du point $\mathbf{M}$ dans le plan image a pour coordonnées

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = f \begin{pmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{pmatrix}. \tag{2.2}$$

Il faut à présent exprimer le point $\mathbf{m}$ dans le repère 2D dans lequel on mesure effectivement les points images (en coordonnées pixel). Dans ce repère, les coordonnées pixel $(u,v)$ du point $\mathbf{m}$ sont données par l'équation (figure 2.2) :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} k_u & -k_u \cos\theta & u_0 \\ 0 & k_v \sin\theta & v_0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix}, \tag{2.3}$$

où $k_u$ et $k_v$ sont le nombre de pixels par unité de longueur suivant chacun des axes, $u_0$ et $v_0$ les coordonnées pixel du *point principal* $\mathbf{c}$, intersection de l'*axe optique* $(\mathbf{C}, \vec{k_c})$ avec le plan image, et

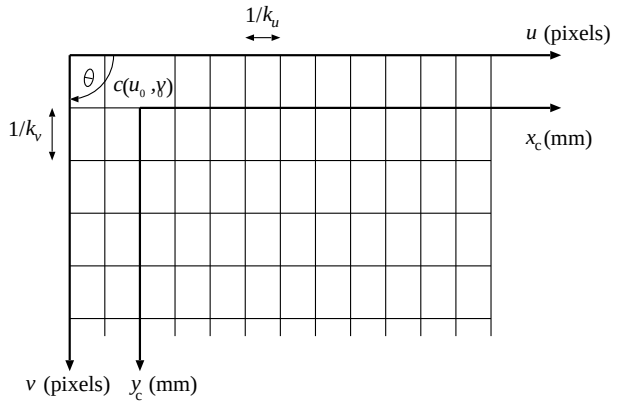



FIG. 2.2 – *Les paramètres internes.*

$\theta$ l'angle entre les deux axes du repère image. Les paramètres $k_u$,$k_v$,$f$,$u_0$,$v_0$,$\theta$ sont les paramètres *internes* (ou *intrinsèques*) de la caméra. En pratique, l'angle $\theta$ est très bien contrôlé et peut être considéré égal à $\frac{\pi}{2}$. D'autre part, il n'est pas possible de séparer les paramètres $k_u$ et $k_v$ de la distance focale $f$ : seules les valeurs $\alpha_u = k_u f$ et $\alpha_v = k_v f$ peuvent être calculées. Nous considérons donc le modèle simplifié à quatre paramètres $\alpha_u$, $\alpha_v$, $u_0$ et $v_0$. D'après les équations (2.1), (2.2) et (2.3), nous avons finalement :

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.4}$$

On note $\mathbf{A}$ la matrice des paramètres internes. La matrice $\mathbf{P} = \mathbf{A}(\mathbf{R} \quad \mathbf{t})$ est appelée *matrice de projection perspective* : elle permet d'exprimer directement la projection d'un point 3D de la scène en coordonnées pixel de l'image. Il s'agit d'une matrice $3 \times 4$ définie à un facteur d'échelle près, et possédant 11 paramètres indépendants.

Nous allons montrer comment on peut retrouver les paramètres internes et externes par des techniques de vision par ordinateur, tout d'abord celles basées uniquement sur des données image, puis celles basées sur la connaissance d'un modèle partiel de la scène, à travers la méthode développée dans notre équipe.

## 2.2 Techniques basées images

De nombreux travaux ont porté ces dix dernières années sur la calibration d'une caméra à partir de vues quelconques de l'environnement, et n'utilisant aucune connaissance *a priori*, ni sur la scène, ni sur les paramètres de la caméra. Il serait présomptueux de résumer l'ensemble des recherches sur ce thème en quelques pages, et un exposé plus long n'aurait pas sa place ici. Nous présentons donc ici les grandes lignes permettant de comprendre comment les paramètres de la caméra peuvent être retrouvés.

### 2.2.1 Géométries projective et euclidienne

Le principe général de ces techniques basées image, appelées également *autocalibration* (ou encore *structure from motion* quand les paramètres internes sont fixes) est de retrouver les projetés de primitives 3D (par exemple des points ou des segments) de la scène dans plusieurs images, et d'en déduire les matrices de projection pour chacune des images, en supposant que ces primitives aient la même position spatiale d'une image à l'autre. Mais supposons qu'on ait pu retrouver un ensemble de points $\mathbf{m}_{ij}$, $\mathbf{m}_{ij}$ étant la projection d'un point 3D $\mathbf{M}_j$ dans l'image $i$. Si l'on en déduit directement des matrices de projection $\mathbf{P}_i$ et des points $\mathbf{M}_j$ telles que :

$$\forall i \, \forall j \quad \mathbf{P}_i \mathbf{M}_j = \mathbf{m}_{ij}$$

les $\mathbf{P}_i$ ne sont alors connus qu'à une transformation homographique $H$ près. Il suffit en effet de remplacer $\mathbf{P}_i$ par $\mathbf{P}_i H$ et $\mathbf{M}_j$ par $H^{-1}\mathbf{M}_j$ pour retrouver les mêmes projetés [Faugeras et al.92]. L'espace ambiant est alors modélisé comme un espace *projectif*, noté $\mathbb{P}^3$.

La géométrie projective est très riche et joue un grand rôle en autocalibration; on en trouvera une très bonne description dans [Hartley et al.00] par exemple. Pour remonter à la structure euclidienne de la scène, [Faugeras et al.92] montrent qu'il faut déterminer la métrique du plan image, c'est-à-dire les paramètres internes de la caméra.

Nous allons voir qu'il existe des contraintes géométriques entre deux vues (ou contrainte épipolaire), que l'on peut exprimer à l'aide de la *matrice fondamentale* (introduite par [Luong92]), et des contraintes entre trois vues, exprimées par le *tenseur trifocal* ([Hartley97]). Une fois ces relations entre les images établies, et grâce à un système d'équations polynomiales dites *équations de Kruppa* qui permettent d'en déduire les paramètres internes, on peut retrouver la trajectoire de la caméra dans l'espace euclidien.

### 2.2.2 La géométrie épipolaire et la matrice fondamentale

#### La contrainte épipolaire

Considérons le cas de deux caméras (ou deux vues d'une même caméra qui s'est déplacée), de centres optiques respectifs $\mathbf{C}$ et $\mathbf{C}'$, non confondus. Nous pouvons voir sur la figure 2.3, qu'étant donné un point $\mathbf{m}$ dans le plan image $\mathcal{P}$, l'ensemble des points physiques $\mathbf{M}$ qui ont pu produire $\mathbf{m}$ se trouvent sur la demi-droite $[\mathbf{Cm})$. Ainsi, tous les correspondants possibles $\mathbf{m}'$ de $\mathbf{m}$ dans le plan image $\mathcal{P}'$ sont situés sur la projection $l'_{\mathbf{m}}$ de cette demi-droite 3D dans la seconde image. La droite $l'_{\mathbf{m}}$ est appelée *droite épipolaire* du point $\mathbf{m}$ dans le plan image $\mathcal{P}'$ de la seconde caméra, et passe par le point $\mathbf{e}'$, intersection de la droite $\mathbf{CC}'$ et du plan $\mathcal{P}'$, appelé *épipole* de la seconde caméra par rapport à la première caméra. De manière symétrique, on définit l'épipole $\mathbf{e}$ de la première caméra par rapport à la seconde caméra comme intersection de la droite $(\mathbf{CC}')$ et du plan $\mathcal{P}$.

Étant donné un point $\mathbf{m}$ dans le plan image $\mathcal{P}$, ses correspondants possibles dans le plan $\mathcal{P}'$ se situent donc sur la droite épipolaire $l'_{\mathbf{m}}$. Cette propriété est appelée *contrainte épipolaire*, et peut être exprimée à l'aide de la matrice fondamentale. Comme nous utilisons la contrainte épipolaire dans notre calcul du point de vue, nous détaillons ici les propriétés de cette matrice et sa détermination.

#### La matrice fondamentale

Soit $(\mathbf{\Delta R} \quad \mathbf{\Delta t})$ le déplacement relatif (ou *mouvement*) de la seconde caméra par rapport à la première caméra, exprimé dans le repère de la seconde caméra. Les trois vecteurs $\overrightarrow{\mathbf{CC}'} = \mathbf{\Delta t}$,

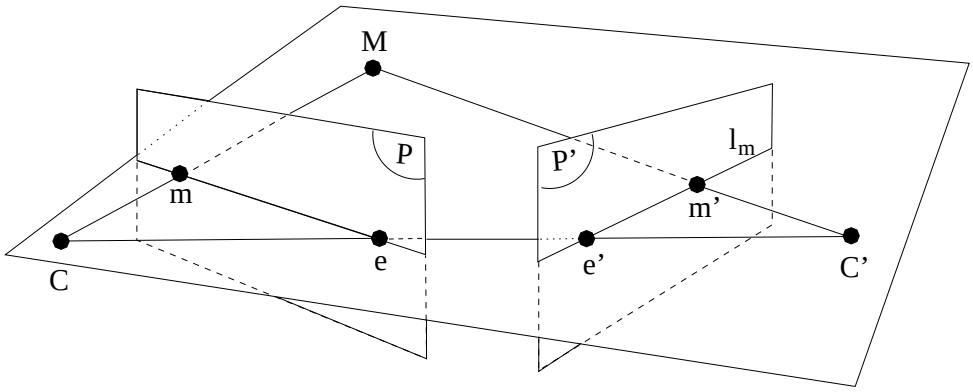



FIG. 2.3 – *La contrainte épipolaire.*

$\overrightarrow{\mathbf{CM}}$ et $\overrightarrow{\mathbf{C'M}}$ étant coplanaires, nous pouvons écrire la relation :

$$(\mathbf{\Delta t} \wedge \mathbf{\Delta R M}_c).\mathbf{M}_{c'} = 0, \tag{2.5}$$

où $\mathbf{M}_c$ est le vecteur $\overrightarrow{\mathbf{CM}}$ exprimé dans le repère de la première caméra, et $\mathbf{M}_{c'}$ le vecteur $\overrightarrow{\mathbf{C'M}}$ exprimé dans le repère de la seconde caméra (le vecteur $\overrightarrow{\mathbf{CM}}$ ayant pour coordonnées $\mathbf{\Delta R M}_c$ dans le repère de la seconde caméra).

Comme $\mathbf{\Delta t} \wedge \mathbf{x} = \mathbf{\Delta T x}$ avec $\mathbf{\Delta T} = \begin{pmatrix} 0 & -\Delta t_z & \Delta t_y \\ \Delta t_z & 0 & -\Delta t_x \\ -\Delta t_y & \Delta t_x & 0 \end{pmatrix}$, l'équation (2.5) peut s'écrire :

$$\mathbf{M}_{c'}^T \mathbf{\Delta T \Delta R M}_c = 0.$$

Si $\mathbf{q}$ et $\mathbf{q'}$ sont les coordonnées rétiniennes des points $\mathbf{m}$ et $\mathbf{m'}$ (respectivement), on a d'après l'équation (2.4), $\mathbf{q} = \mathbf{A M}_c$ et $\mathbf{q'} = \mathbf{A' M}_{c'}$, où $\mathbf{A}$ et $\mathbf{A'}$ sont les matrices des paramètres internes des deux caméras. Nous obtenons alors l'équation de Longuet-Higgins, qui relie un point $\mathbf{q}$ et son correspondant $\mathbf{q'}$ :

$$\mathbf{q'}^T \mathbf{F q} = 0, \tag{2.6}$$

où $\mathbf{F} = \mathbf{A'}^{-T} \mathbf{\Delta T \Delta R A}^{-1}$ est la matrice fondamentale ($(\mathbf{X}^{-1})^T$ est noté $\mathbf{X}^{-T}$). Le vecteur $\mathbf{l'_q}$ des coefficients de la droite épipolaire $l'_\mathbf{q}$ vaut $\mathbf{l'_q} = \mathbf{Fq}$. L'équation 2.6 exprime donc bien la contrainte épipolaire, qui est que le point $\mathbf{q'}$ correspondant au point $\mathbf{q}$ appartient à la droite $l'_\mathbf{q}$.

### Détermination de la matrice fondamentale

Les propriétés de la matrice fondamentale ont été étudiées par de nombreux auteurs (en particulier [Luong92]). Il s'agit d'une matrice $3 \times 3$ de rang 2, et donc de déterminant nul. L'équation (2.6) peut s'écrire :

$$\mathbf{U}^T \mathbf{f} = 0, \tag{2.7}$$

où $\mathbf{U} = [uu',vu',u',uv',vv',v',u,v,1]^T$ et $\mathbf{f} = [F_{11},F_{12},F_{13},F_{21},F_{22},F_{23},F_{31},F_{32},F_{33}]^T$, $(u,v)$ et $(u',v')$ étant les coordonnées pixel des points $\mathbf{q}$ et $\mathbf{q'}$. Si nous disposons de huit appariements de points, nous pouvons donc déterminer une solution unique pour $\mathbf{F}$, définie à un facteur d'échelle près (l'un des neuf coefficients est fixé à 1), en résolvant un système linéaire composé de huit

équations (2.7). En général, on dispose d'un nombre d'appariements $(\mathbf{q}_i,\mathbf{q}'_i)$ bien plus grand que huit, et l'équation est résolue aux moindres carrés en minimisant le critère linéaire :

$$\min_{\mathbf{F}} \sum_i (\mathbf{q}'^T_i \mathbf{F} \mathbf{q}_i)^2. \tag{2.8}$$

Malheureusement, le critère linéaire possède deux défauts qui le rendent très sensible au bruit : d'une part, il ne tient pas compte de la contrainte de rang ($\det(\mathbf{F}) = 0$), ce qui entraîne une incohérence de la géométrie épipolaire, qu'on peut remarquer facilement: les droites épipolaires doivent s'intersecter en un point unique (l'épipole) alors que ce n'est pas le cas des droites épipolaires obtenues avec une telle matrice. Luong propose donc de minimiser un critère non linéaire, qui est la distance euclidienne d'un point à la droite épipolaire de son correspondant. La distance euclidienne d'un point $\mathbf{q}'$ de la seconde image à la droite épipolaire $\mathbf{l}'_\mathbf{q} = (l'_1,l'_2,l'_3)^T = \mathbf{F}\mathbf{q}$ est donnée par :

$$d(\mathbf{q}',\mathbf{l}'_\mathbf{q}) = \frac{|\mathbf{q}'^T \mathbf{l}'_\mathbf{q}|}{\sqrt{(l'_1)^2 + (l'_2)^2}} \tag{2.9}$$

On peut envisager dans un premier temps de minimiser le critère suivant :

$$\min_{\mathbf{F}} \sum_i d^2(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i)$$

Cependant, contrairement au critère linéaire, ce critère n'est pas symétrique puisqu'il ne détermine que les droites épipolaires dans la seconde image : il s'ensuit une incohérence de la géométrie épipolaire obtenue entre les deux images. Pour obtenir une géométrie épipolaire cohérente, nous pouvons inverser le rôle des deux images en transposant la matrice fondamentale. Ceci conduit au critère suivant, qui opère simultanément sur les deux images :

$$\min_{\mathbf{F}} \sum_i (d^2(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i) + d^2(\mathbf{q}_i,\mathbf{F}^T\mathbf{q}'_i)) \tag{2.10}$$

Ce critère est normalisé, au sens où il ne dépend pas du facteur d'échelle choisi pour $\mathbf{F}$. Nous pouvons aussi prendre en compte le fait que $\mathbf{F}$ est de rang deux en paramétrant cette matrice par le nombre exact de variables indépendantes, qui est de sept, une fois que le facteur d'échelle a été pris en compte (la troisième ligne de la matrice est alors écrite comme une combinaison linéaire des deux premières). Luong montre que le critère non linéaire est beaucoup plus stable que le critère linéaire. Il met toutefois en évidence l'instabilité du calcul de la matrice fondamentale pour certains types de mouvements : les mouvements de faible amplitude, les mouvements dont la translation est parallèle au plan image et les translations pures.

### 2.2.3 Les paramètres internes

#### Les équations de Kruppa

L'introduction d'un invariant projectif important, la *conique absolue*, permet d'obtenir une contrainte géométrique exprimant le fait que le mouvement entre les repères des deux caméras est un déplacement rigide, et non une relation projective linéaire quelconque [Maybank et al.92].

Précisons tout d'abord que les points de l'espace projectif $\mathbb{P}^3$ sont définis à un facteur multiplicatif près par un vecteur à 4 coordonnées $(\mathcal{X},\mathcal{Y},\mathcal{Z},\mathcal{T})$, et le plan $\mathcal{T} = 0$ est appelé *plan à*

*l'infini* et noté $\Pi_\infty$. La conique absolue, notée $\Omega$, est située dans le plan à l'infini $\Pi_\infty$ et est défini par le système d'équations:

$$\begin{cases} \mathcal{X}^2 + \mathcal{Y}^2 + \mathcal{Z}^2 = 0 \\ \mathcal{T} = 0 \end{cases}$$

Notons qu'elle ne possède aucun point réel. Soit $\mathbf{M}$ un point de $\Omega$ : par définition, ses coordonnées projectives sont de la forme $\tilde{\mathbf{M}} = (\mathbf{M}^T \quad 0)^T$ avec $\mathbf{M}^T\mathbf{M} = 0$. On montre [Zeller et al.96], que l'image du point $\mathbf{M}$ dans la rétine appartient à la conique $\omega$ d'équation $\mathbf{q}^T\mathbf{K}\mathbf{q} = 0$, où $\mathbf{K} = \mathbf{A}^{-T}\mathbf{A}^{-1}$ ne dépend que des paramètres internes de la caméra. Cette propriété très intéressante va nous permettre d'établir les équations de Kruppa.

On sait que l'ensemble des tangentes à une conique forme aussi une conique, dite duale. Soit $\mathbf{B}$ la matrice de la conique duale de $\omega$ (c'est la matrice des cofacteurs de $\mathbf{K}$). Soit ($\mathbf{eq}$) une droite épipolaire de la première image. Cette droite est tangente à $\omega$ si et seulement si elle appartient à la conique duale de $\omega$, ce qui s'écrit :

$$(\mathbf{e} \wedge \mathbf{q})^T\mathbf{B}(\mathbf{e} \wedge \mathbf{q}) = 0. \tag{2.11}$$

D'autre part, les tangentes menées de chacun des épipoles $\mathbf{e}$ et $\mathbf{e}'$ à l'image de $\Omega$ se correspondent puisqu'elles sont les traces dans les deux plans images des deux plans tangents menés par la droite ($\mathbf{CC}'$) à $\Omega$. Si $\mathbf{q}$ appartient à l'une des deux tangentes menées de $\mathbf{e}$ à $\omega$ dans la première image, sa droite épipolaire représentée par $\mathbf{Fq}$ est donc tangente à la projection de $\Omega$ dans la deuxième image, qui est égale à $\omega$ si les paramètres internes de la caméra sont constants entre les deux vues (invariance de $\omega$ par déplacement rigide). Ceci s'écrit :

$$\mathbf{q}^T\mathbf{F}^T\mathbf{B}\mathbf{F}\mathbf{q} = 0. \tag{2.12}$$

Les équations (2.11) et (2.12) induisent deux équations quadratiques en les coefficients de $\mathbf{B}$, qui sont les équations de Kruppa. De $\mathbf{B}$, on peut déduire $\mathbf{K}$ puis $\mathbf{A}$. Puisque les paramètres internes sont au nombre de cinq, trois mouvements sont en théorie nécessaires pour les déterminer. Pour le modèle de caméra simplifié à quatre paramètres internes, deux mouvements sont suffisants.

### Autres approches

Quelques travaux ont porté sur l'obtention des paramètres internes à partir d'une image unique [Caprile et al.90, Liebowitz et al.99]. En particulier, Liebowitz et al. demandent à l'utilisateur de tracer dans l'image des droites qui sont parallèles dans le monde 3D, ce qui est possible en pratique pour des scènes urbaines ou d'intérieur. L'intersection des projections de ces droites est appelé *points à l'infini*. Liebowitz et al. montrent qu'à partir de plusieurs de ces points, on peut retrouver les paramètres internes.

Une fois connus les paramètres internes et la matrice fondamentale reliant deux caméras, on peut alors retrouver le mouvement entre ces deux caméras ($\mathbf{\Delta R} \quad \mathbf{\Delta t}$). Cependant, seule la direction de la translation peut être calculée, on ne peut donc pas se contenter de calculer le mouvement entre les paires d'images consécutives si on veut retrouver la trajectoire de la caméra sur plusieurs images. Ce problème est illustré en figure 2.4. Supposons que le centre optique de la caméra se déplace suivant les translations $\mathbf{\Delta t}_{12}$,$\mathbf{\Delta t}_{23}$. Nous savons que la translation calculée $\mathbf{\Delta t}_{1'2'}$ n'est définie qu'à un facteur d'échelle près. Si nous nous basons uniquement sur les images 2 et 3 pour calculer la translation $\mathbf{\Delta t}_{2'3''}$ entre ces deux images, comme cette translation n'est également définie qu'à un facteur d'échelle près, la direction de la translation $\mathbf{\Delta t}_{1'3''}$ ne sera pas

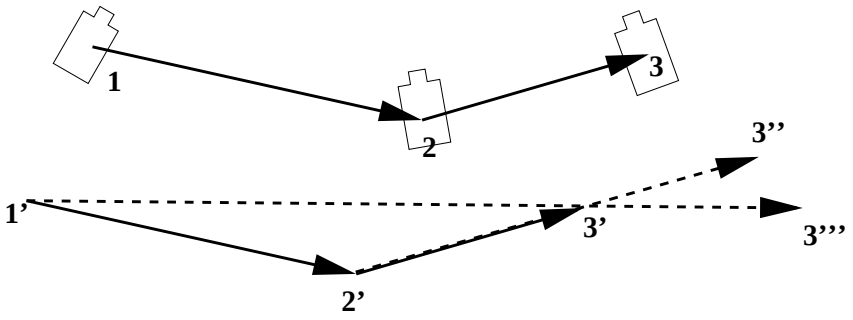



FIG. 2.4 – *Le problème du facteur d'échelle sur 3 images.*

la même que celle de la translation $\mathbf{\Delta t}_{13}$. Pour obtenir un facteur d'échelle cohérent, il faut en plus calculer la translation $\mathbf{\Delta t}_{1'3''}$ à partir des images 1 et 3 : la position $3'$ recherchée est alors l'intersection des deux droites portées par $\mathbf{\Delta t}_{1'3''}$ et $\mathbf{\Delta t}_{2'3''}$.

Une solution plus élégante est donnée par le tenseur trifocal, que nous présentons maintenant.

### 2.2.4 Le tenseur trifocal

Dans le cas de deux caméras, la matrice fondamentale permet de réduire la recherche d'un point $\mathbf{m}'$ correspondant à un point $\mathbf{m}$ en se limitant à la droite épipolaire de $\mathbf{m}$. Dans le cas de trois vues (figure 2.5), on a une redondance qui permet de réduire encore la recherche : étant donné un couple de points images ($\mathbf{m}_1$,$\mathbf{m}_2$) des deux premières vues satisfaisant la contrainte épipolaire, le point image $\mathbf{m}_3$ de la troisième vue correspondant au même point 3D peut être construit à l'aide d'un tenseur, dit *tenseur trifocal*. Si la matrice fondamentale est une matrice $3 \times 3$, le tenseur trifocal $\mathbf{T}$ est un tenseur $3 \times 3 \times 3$, et $\mathbf{m}_3$ peut être construit à l'aide des relations :

$$\mathbf{m}_3^l = \mathbf{m}_2^i \sum_{k=1}^{k=3} \mathbf{m}_1^k \mathbf{T}_{kjl} - \mathbf{m}_2^j \sum_{k=1}^{k=3} \mathbf{m}_1^k \mathbf{T}_{kil}$$

où $\mathbf{m}_1^2$ représente la deuxième coordonnée du point $\mathbf{m}_1$, par exemple. A l'aide d'une méthode statistique (RANSAC), [Torr et al.97] utilisent des correspondances de points sur 3 images pour estimer les coefficients de $\mathbf{T}$ et vérifier simultanément la mise en correspondance des primitives 3D.

Une fois estimés les tenseurs trifocaux entre chaque triplet d'images consécutives, on peut retrouver facilement la trajectoire de la caméra. Comme le montre la figure 2.6, le facteur d'échelle entre les tenseurs $T_{1\leftrightarrow 2\leftrightarrow 3}$ et $T_{2\leftrightarrow 3\leftrightarrow 4}$, par exemple, est calculé grâce aux caméras communes 2 et 3. [Fitzgibbon et al.98] utilisent ainsi le tenseur trifocal avec une approche hiérarchique pour déterminer une bonne estimation de la trajectoire.

### 2.2.5 Ajustement de faisceaux

L'utilisation de triplets d'images présente un défaut puisque les points 2D ne sont utilisés que localement, alors qu'ils peuvent être suivis sur un grand nombre d'images. Une quantité non négligeable d'information est donc perdue, et de fait, les trajectoires retrouvées souffrent d'imprécision. La dernière étape d'une recherche des points de vue consiste généralement, après avoir suivi suffisamment de points 2D sur la séquence et estimé les matrices de projections de chacune des images, à optimiser simultanément ces projections et la position des points 3D reconstruits

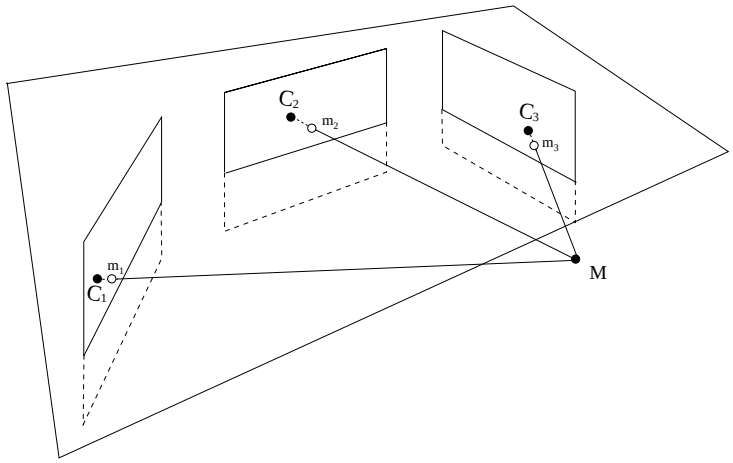



FIG. 2.5 – *La géométrie trifocale.*

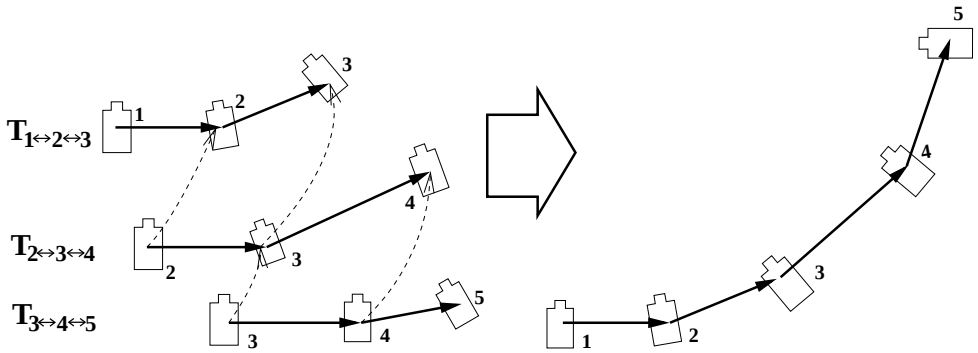



FIG. 2.6 – *Utilisation du tenseur trifocal pour retrouver la trajectoire de la caméra.*

à partir des points 2D. Pour cela, on cherche à minimiser l'erreur globale de reprojection, c'est-à-dire le résidu :

$$\Phi(\mathbf{P},\mathbf{M}) = \frac{1}{N_{\text{rep}}} \sum_{i,j} \sigma(i,j).\text{Dist}^2(\mathbf{P}_i\mathbf{M}_j,\mathbf{m}_{ij}) \tag{2.13}$$

avec $\sigma(i,j) = 1$ si $\mathbf{M}_j$ est suivi dans l'image $i$, et 0 sinon, $N_{\text{rep}}$ le nombre total de points 2D suivis (le facteur $\frac{1}{N_{\text{rep}}}$ permettant de normaliser $\Phi$), et Dist($\mathbf{m}$,$\mathbf{n}$) la distance euclidienne entre les points 2D $\mathbf{m}$ et $\mathbf{n}$.

Cette estimation est connue sous le nom d'« ajustement de faisceaux » (*bundle adjustement* en anglais), elle revient en effet à ajuster le faisceau de rayons entre chaque centre de caméra et l'ensemble des points 3D (ou, de façon équivalente, entre chaque point 3D et l'ensemble des centres de caméra). Son intérêt par rapport à ce qui vient d'être exposé est qu'elle prend en compte l'ensemble des points et des images de la séquence. Mais comme elle est effectuée à l'aide d'une minimisation numérique, elle requière une estimation initiale des projections qui doit être suffisamment proche du minimum, et qui peut être fournie par la méthode exposée plus haut.

Comme nous utilisons également l'ajustement de faisceaux pour affiner les points de vue obtenus à l'aide de notre méthode basée modèle, nous détaillons maintenant la minimisation de $\Phi$.

**Méthodes numériques directes** On peut évidemment utiliser une méthode numérique classique de minimisation: descente de gradient, gradient conjugué, méthode de Gauss-Newton... Néanmoins, outre l'estimation initiale, le problème posé par l'ajustement de faisceaux est essentiellement sa taille, puisqu'il consiste à rechercher un grand nombre d'inconnus: de 6 (si les paramètres internes sont connus *a priori*) à 11 paramètres pour chaque projection, et 3 paramètres pour chacun des points 3D. Utilisée directement, c'est-à-dire sans tenir compte de la structure particulière du problème à résoudre ici, les méthodes numériques ont une faible performance en temps de calcul (on trouvera une comparaison des méthodes numériques et de leurs efficacités pour l'ajustement de faisceaux dans [Triggs et al.00]).

**Algorithme de Hartley** [Hartley94] a proposé une méthode basée sur l'algorithme de Levenberg-Marquart, qui est une variante de l'algorithme de Newton. Cet algorithme nécessite d'inverser, à chaque itération, une matrice dont l'ordre est le nombre de paramètres à estimer. Hartley remarque que cette matrice, dans le cas de l'ajustement de faisceaux, a une structure en blocs particulière, et développe une méthode permettant d'optimiser l'inversion de cette matrice, pour réduire les temps de calcul.

**Séparation des paramètres** Une approche bien connue pour retrouver le minimum dans un espace de paramètres grand consiste à séparer les paramètres, estimer le minimum pour un sous-ensemble en fixant les autres paramètres, puis à itérer en changeant de sous-ensemble. L'ajustement de faisceaux se prête bien à cette approche puisqu'on peut estimer indépendamment chaque point 3D en fixant les caméras, puis chaque caméra en fixant les points 3D, et itérer. Cependant, cette méthode peut être dangereuse, puisqu'elle risque d'osciller voire de diverger. Nous verrons dans nos expérimentations que c'est effectivement parfois le cas. Cependant, en ne conservant que les points suivis sur suffisamment d'images, cette méthode converge alors vers une solution correcte.

**Réduction du problème** On peut également réduire le nombre de paramètres en séparant les inconnues en plusieurs sous-ensembles (par exemple les 10 premières images, les images 5 à 15, etc...), effectuer l'ajustement de faisceaux sur chaque sous-ensemble puis fusionner les résultats. La difficulté est alors de savoir comment effectuer cette fusion. De plus, on n'échappe pas à une étape finale de minimisation sur l'ensemble des données pour réellement tenir compte de l'ensemble de la séquence.

### 2.2.6 En conclusion sur les méthodes basées image

L'intérêt des méthodes basées image est de se passer de connaissances *a priori* sur la scène. Elles sont néanmoins très dépendantes de la qualité de la mise en correspondance des primitives 2D, c'est-à-dire d'une bonne distribution des primitives, de la présence éventuelle d'erreurs d'appariement ou d'un bruit trop important sur la position de ces primitives. Chaque étape (la détection des primitives, leur appariement, l'estimation de la matrice fondamentale et du tenseur trifocal, etc...) doit donc être réalisée très soigneusement. C'est pourquoi des logiciels commerciaux comme 3D-equalizer ou MatchMover de Realviz [Realviz00] mettent l'utilisateur à contribution pour superviser l'appariement des primitives 2D. Ce n'est que tout récemment que la société 2d3 [2d300] a sorti Boujou, qui est un produit entièrement automatique. Certaines séquences restent difficiles à traiter, comme celles à focale variable, ou les séquences présentant trop d'objets en mouvement (ce qui viole l'hypothèse de rigidité des points 3D suivis), et qui requièrent l'intervention de l'utilisateur.

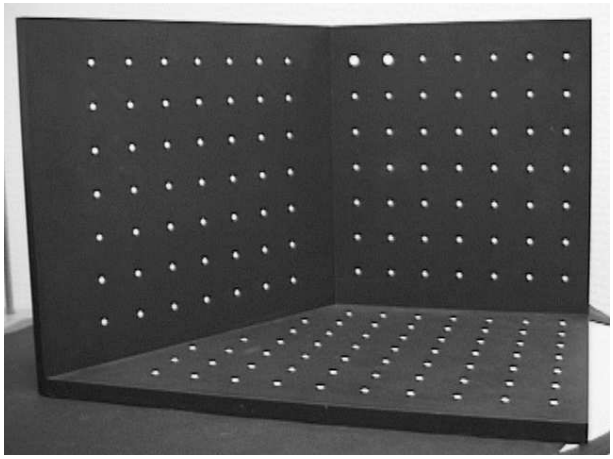


FIG. 2.7 – *La mire de calibration utilisée au LORIA.*

Enfin, les méthodes basées image retrouvent la trajectoire de la caméra à une transformation euclidienne près, c'est-à-dire que les points de vue ne sont pas exprimés dans un repère absolu (mais généralement dans le repère de la première caméra), et que l'échelle n'est pas connue. Cela peut rendre délicat l'intégration d'objets virtuels dans la scène.

## 2.3 Technique basée modèle développée dans l'équipe

### 2.3.1 Introduction

Les techniques basées modèle sont sans doute plus simples à mettre en œuvre et reposent sur la connaissance des coordonnées 3D d'un certain nombre de points de référence $\mathbf{M}_i$, et de leurs projections $\mathbf{m}_i$ dans le plan image, mesurées sous forme de coordonnées pixel $\mathbf{q_i}$. À partir de ces correspondances 3D/2D, il est possible de calculer les $N$ paramètres $\lambda_i$ de la matrice de projection perspective ($N = 10$ pour le modèle simplifié de caméra) à partir des $n$ équations $\mathbf{q}_i = \mathbf{P}(\lambda_1,...,\lambda_N)\mathbf{M}_i$, pourvu que $n$ soit suffisamment grand.

Néanmoins, la détermination simultanée des paramètres internes et externes de la caméra est généralement peu précise (voir [Bougnoux98]). Le calcul de ces paramètres est donc souvent séparé : les paramètres internes peuvent être obtenus en filmant une *mire de calibration* en début de session, formée de motifs répétitifs (cercles, ellipses ou rectangles), choisis pour définir des points d'intérêt qui peuvent être mesurés avec une très grande précision (voir par exemple la mire utilisée au LORIA en figure 2.7). Par la suite, seul le point de vue est recalculé, ce qui suppose que les paramètres internes ne varient pas au cours de la prise de vue.

### 2.3.2 Utilisation de courbes 3D quelconques

La plupart des algorithmes de calcul du point de vue utilisent des primitives simples : points [Dementhon et al.95], droites [Dhome et al.89, Shakunaga93, Kumar et al.94] ou cercles [Ferri et al.93]. Peu de travaux sont consacrés à l'utilisation de courbes 3D de forme libre. Kriegman et Ponce proposent une méthode algébrique pour calculer le point de vue à partir de l'observation des contours occultants d'un objet courbe [Kriegman et al.90]. Malheureusement, cette méthode ne fonctionne qu'avec des surfaces ou des courbes paramétriques (ou plus exactement exprimables sous forme de fractions de polynômes) et utilise la théorie de l'élimination qui est très lourde à

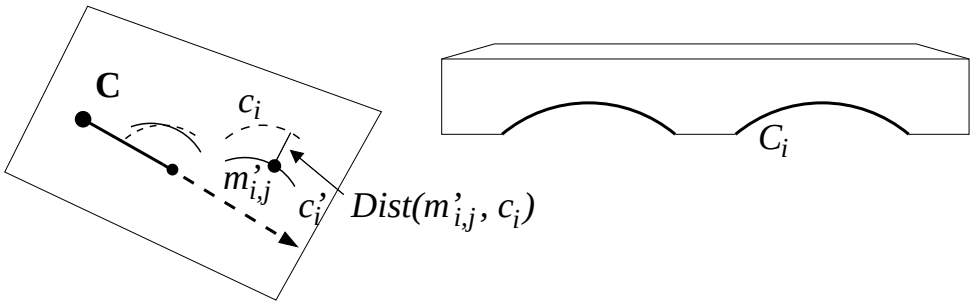



FIG. 2.8 – *Illustration du principe de la méthode développée dans l'équipe.*

mettre en œuvre [Simon95].

Le but de l'algorithme présenté ici est de calculer le point de vue à partir d'une estimée initiale $\mathbf{p_0}$ et des correspondances 3D/2D de $n$ courbes 3D quelconques décrites par des chaînes de points. Notons :

- $\mathcal{C}_i$ une courbe 3D, décrite par la chaîne de points 3D $\{\mathbf{M}_{i,j}\}_{1\leq j\leq l_i}$,
- $c_i$ la projection de $\mathcal{C}_i$ dans le plan image, décrite par la chaîne de points 2D $\{\mathbf{m}_{i,j}\}_{1\leq j\leq l_i}$, où $\mathbf{m}_{i,j} = Proj(\mathbf{R}\mathbf{M}_{i,j} + \mathbf{t})$,
- $c'_i$ le correspondant image de $\mathcal{C}_i$, décrit par la chaîne de points 2D $\{\mathbf{m}'_{i,j}\}_{1\leq j\leq l'_i}$.

Les courbes $c'_i$ sont suivies dans la séquence en utilisant la méthode de [Berger94] de contours actifs (introduits par [Kass et al.88]), qui utilise à la fois le flot optique et la distribution des intensités pour suivre des courbes d'une image à l'autre, même pour des déplacements importants (jusqu'à une vingtaine de pixels).

Une solution simple consisterait à minimiser l'erreur de reprojection entre les courbes $c_i$ et $c'_i$ (voir figure 2.8):

$$f(\mathbf{p}) = \sum_{i,j} \mathrm{Dist}(\mathbf{m}'_{i,j}, c_i) \tag{2.14}$$

où Dist est une fonction qui approxime la distance euclidienne d'un point à un contour.

Malheureusement, cette méthode n'est pas satisfaisante pour plusieurs raisons:

- comme nous l'avons déjà signalé, lorsque les courbes $c'_i$ sont suivies, des erreurs de localisation peuvent apparaître. Or, l'expérimentation montre que même une erreur faible dans le calcul de l'erreur de reprojection peut entraîner une grande erreur sur la localisation de la caméra.
- plus précisément, deux types d'erreur bien dissociés peuvent apparaître : certaines primitives peuvent être localement incorrectes lorsqu'elles sont localement mal suivies ou partiellement occultées par un objet de la scène. D'autres primitives peuvent être globalement aberrantes lorsqu'elles sont complètement occultées ou que l'algorithme de suivi fournit un mauvais résultat (un exemple obtenu dans la séquence du Pont Neuf où ces deux types d'erreur apparaissent est présenté en figure 2.9). La fonction (2.14) ne tient pas compte de cette distinction, alors que ces deux types d'erreur ne sont pas de même nature et ne pas les dissocier rend l'algorithme moins robuste et moins précis. Il faudrait, à l'inverse, pouvoir éliminer les primitives aberrantes et utiliser l'information efficace des primitives localement correctes.

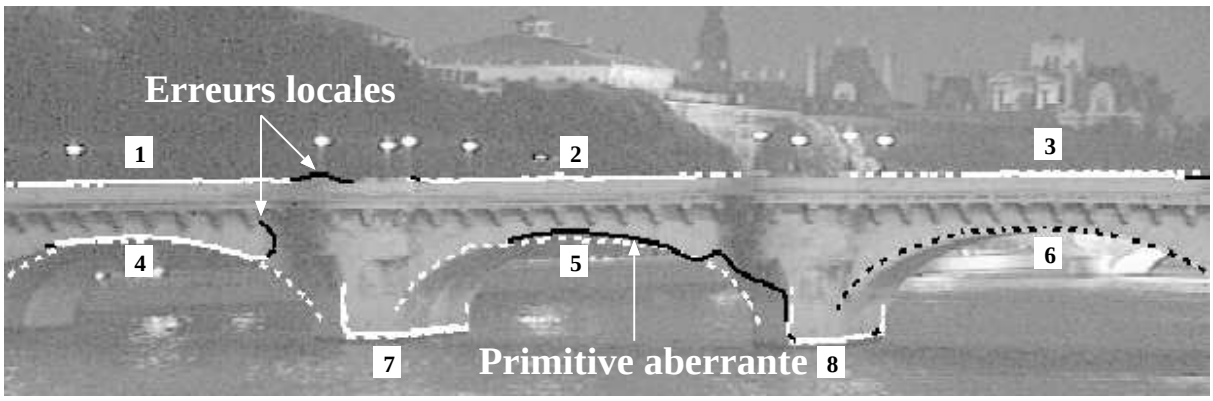



FIG. 2.9 – *Exemple d'erreurs de suivi obtenues sur la séquence du Pont Neuf (une erreur locale est obtenue pour les primitives 1 et 4 et une erreur aberrante pour la primitive 5). Les lignes continues sont les courbes 2D suivies dans la séquence, les lignes en pointillé la reprojection des courbes 3D.*

Pour remédier à cela, Simon a proposé dans sa thèse [Simon99] d'utiliser des *estimateurs robustes*, que nous présentons plus bas, permettant de tenir compte de la présence de données erronées. Lors de la minimisation de l'erreur de reprojection, ces estimateurs sont présents à deux niveaux: à un niveau local, où un résidu robuste est calculé pour chaque primitive, et à un niveau global, où une fonction robuste de ces résidus est minimisée. Le niveau local a pour but de réduire l'influence des erreurs locales tandis que le niveau global doit permettre d'éliminer les primitives aberrantes.

### 2.3.3 Estimation robuste

Les estimateurs robustes ont été introduits par les statisticiens [Rousseeuw et al.87] et largement utilisés en vision par ordinateur [Haralick et al.89, Kumar et al.94, Zhang et al.95]. Les estimateurs les plus couramment utilisés sont les M-estimateurs et les moindres carrés médians (*Least Median of Squares* - LMS). Les premiers minimisent la somme d'une fonction $\rho$ des résidus, de manière à ce que les résidus les plus élevés (points erronés) aient une influence moindre dans l'optimisation. Les moindres carrés médians minimisent non plus la somme d'une fonction des résidus, mais la médiane des résidus au carré, de sorte que les résidus les plus élevés n'ont plus aucune influence dans l'optimisation.

La technique de M-estimation, développée par Huber [Huber81], consiste à minimiser la somme d'une fonction des résidus $r_i$ :

$$f(\mathbf{p}) = \sum_{i=1}^{n} \rho(r_i), \tag{2.15}$$

où $\rho$ une fonction continue et symétrique, ayant un minimum en zéro. En différenciant $f$ par rapport aux six composantes de $\mathbf{p}$ et considérant que les dérivées sont nulles lorsque le minimum est atteint, on obtient les équations suivantes :

$$\sum_{i=1}^{n} \psi(r_i) \frac{\partial r_i}{\partial p_j} = 0, \text{ pour } j = 1,...,6, \tag{2.16}$$

où la dérivée $\psi(x) = \mathrm{d}\rho(x)/\mathrm{d}x$ est appelée *fonction d'influence*, car elle se comporte comme une fonction pondérante dans les équations (2.16).

Différents estimateurs ont été proposés, telles celui de Huber avec

$$\rho(x) = \begin{cases} x^2/2 & \text{si } |x| \leq c \\ c(|x| - c/2) & \text{sinon} \end{cases}$$

et celui de Tukey

$$\rho(x) = \begin{cases} \frac{c^2}{6}\left[1 - \left(1 - \left(\frac{x}{c}\right)^2\right)^3\right] & \text{si } |x| \leq c \\ c^2/6 & \text{sinon} \end{cases},$$

où $c$ est un seuil qui dépend généralement de l'écart-type des mesures.

### 2.3.4 Critère minimisé

Simon commence par calculer pour chaque courbe $C_i$, une erreur résiduelle $r_i$ qui traduit la distance entre la courbe projetée $c_i$ et la courbe mesurée dans l'image $c'_i$. Ce résidu est obtenu par une fonction robuste des distances $\{d_{i,j} = \text{Dist}(m'_{i,j}, c_i)\}_{1 \leq j \leq l'_i}$, afin de minimiser l'influence des erreurs locales.

$$r_i^2 = \frac{1}{l'_i} \sum_{j=1}^{l'_i} \rho_1(d_{i,j}) \tag{2.17}$$

L'emploi d'une fonction robuste des résidus $r_i$ permet ensuite de réduire l'influence des primitives aberrantes, c'est-à-dire les contours qui sont complètement faux, ou qui contiennent une trop grande proportion de points erronés. Finalement, le critère global minimisé pour estimer le point de vue $\mathbf{p}$ est:

$$f(\mathbf{p}) = \sum_{i=1}^{n} \rho_2(r_i), \tag{2.18}$$

où $r_i$ est donné par l'équation (2.17) Les fonctions $\rho_1$ et $\rho_2$ retenues sont la fonction de Huber pour $\rho_1$ et celle de Tukey pour $\rho_2$ (voir [Simon99] pour une justification du choix des estimateurs). Ce minimum est évalué grâce à la méthode de convergence quadratique proposée par Powell, décrite dans [Press et al.88].

### 2.3.5 Résultats

La méthode qui vient d'être décrite a été utilisée avec succès sur une séquence du Pont-Neuf, pour le projet des ponts de Paris présenté dans le chapitre 1. Malgré la faible précision du modèle 3D du pont utilisé, et les mauvaises conditions de luminosité (il s'agit d'une séquence de nuit) rendant difficile le suivi des primitives, elle a permis, de façon automatique, d'incruster un pont virtuel, semblable au pont réel, mais rendu selon un éclairage différent (voir figure 1.2).

Malheureusement, un problème survient quand l'objet virtuel est incrusté dans une zone éloignée des courbes 3D. Ce problème est illustré sur la séquence Stanislas: cette séquence montre une partie de la place Stanislas, sur laquelle apparaît l'opéra de Nancy, et nous souhaitons incruster un véhicule virtuel roulant sur la place. Pour calculer le point de vue, nous disposons d'une modélisation de la façade de l'opéra. La figure 2.10 montre le résultat de l'incrustation du véhicule dans une image de la séquence, à distance croissante de l'opéra, selon un point de vue

FIG. 2.10 – *Incrustation d'un véhicule à distance croissante de l'opéra.*

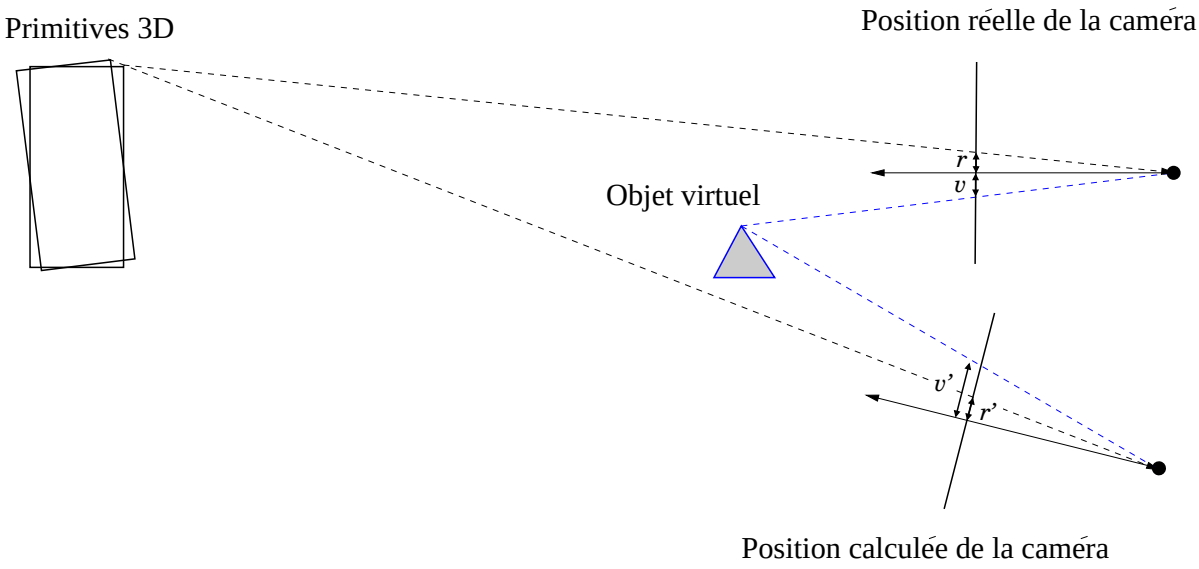



FIG. 2.11 – *Une petite erreur de reprojection au niveau de l'objet réel suivi ($r \simeq r'$) peut conduire à une incrustation complètement fausse ($v \neq v'$) pour un objet virtuel éloigné de l'objet ayant servi au calcul du point de vue.*

calculé en utilisant des courbes 3D situées sur cette façade. Nous voyons que plus le véhicule est éloigné de l'opéra, moins sa projection semble correcte. Pourtant, la projection de la façade est quant à elle correcte. Le schéma 2.11 explique ce phénomène: une petite erreur de reprojection au niveau des primitives 3D peut conduire à une incrustation fausse pour un objet virtuel éloigné de l'objet ayant servi au calcul du point de vue.

C'est pourquoi la méthode que nous venons de présenter a été étendue en une méthode hybride, qui prend en compte non seulement des connaissances 3D, mais également des appariements points 2D, susceptibles d'être détectés sur l'ensemble de la scène, et d'apporter ainsi l'information tridimensionnelle qui peut faire défaut.

## 2.4 Méthode hybride: fusion de données 3D et 2D

Le principe de la méthode hybride est de minimiser *simultanément* les résidus 3D/2D obtenus à partir de courbes suivies dans la séquence que nous venons de présenter, et des résidus 2D/2D obtenus à partir de points détectés et appariés entre deux images consécutives de la séquence, et

fondés sur la géométrie épipolaire.

### 2.4.1 Principe

Supposons que nous ayons établi $n$ appariements 3D/2D de courbes quelconques dans l'image courante $I$, et $m$ appariements de points $(\mathbf{q}_i,\mathbf{q}'_i)$ entre l'image $I$ et une autre image $I'$ de la séquence, dont on connaît le point de vue $(\mathbf{R}' \quad \mathbf{t}')$. Nous cherchons à calculer les paramètres $\mathbf{p}$ du point de vue $(\mathbf{R} \quad \mathbf{t})$ pour l'image $I$. Nous avons vu dans la partie 2.2.2, consacrée aux méthodes basées images, qu'à partir d'appariements 2-D/2-D entre les images $I$ et $I'$, la contrainte épipolaire permet de déterminer le mouvement (à un facteur d'échelle près) de la caméra $(\Delta\mathbf{R} \quad \Delta\mathbf{t})$ entre ces deux images. Nous pouvons donc ajouter une contrainte supplémentaire au calcul du point de vue basé sur le modèle de la scène en minimisant les résidus

$$v_i^2 = \frac{1}{2}(d^2(\mathbf{q}'_i,\mathbf{F}\mathbf{q}_i) + d^2(\mathbf{q}_i,\mathbf{F}^T\mathbf{q}'_i)). \tag{2.19}$$

$v_i$ peut être exprimé en fonction du point de vue recherché et du point de vue de l'image précédente: d'après 2.6, nous avons en effet $\mathbf{F} = \mathbf{A}^{-T}\mathbf{\Delta T}\mathbf{\Delta R}\mathbf{A}^{-1}$ (en considérant que les paramètres internes ne varient pas entre les deux images), où $\mathbf{\Delta T}\mathbf{x} = \mathbf{\Delta t} \wedge \mathbf{x}$. D'autre part, on montre facilement que

$$\begin{cases} \mathbf{\Delta R} = \mathbf{R}\mathbf{R}'^T, \\ \mathbf{\Delta t} = \mathbf{t} - \mathbf{R}\mathbf{R}'^T\mathbf{t}', \end{cases}$$

ce qui nous permet d'exprimer $\mathbf{F}$ en fonction de $\mathbf{R}$ et $\mathbf{t}$, et donc $\mathbf{p}$.

Le résidu $v_i$ est particulièrement commode à prendre en compte, puisqu'il s'agit d'une distance mesurable en pixels dans les images, que nous allons donc pouvoir combiner de façon cohérente avec les résidus 3D/2D $r_i$, qui sont des distances entre courbes de l'image, elles aussi mesurables en pixels. Ainsi, l'approche hybride consiste à minimiser la fonction:

$$h(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}\rho_2(r_i) + \frac{\lambda}{m}\sum_{i=1}^{m}\rho(v_i). \tag{2.20}$$

La somme de gauche est exactement la fonction robuste à deux niveaux $f(\mathbf{p})$ présentée dans la section précédente. La somme de droite, qui est donc le terme permettant de prendre en compte la contrainte épipolaire, utilise également un estimateur robuste, puisque certains des appariements entre points $\mathbf{q}$ et $\mathbf{q}'$ peuvent être erronés. $\lambda$ est un terme permettant de pondérer l'influence des appariements par rapport à celle des primitives 3D. Dans tous les résultats présentés dans la suite, nous avons fixé $\lambda = 1$.

La contrainte de reprojection nous permet de conserver un système stable et autonome tout en levant l'indétermination sur le facteur d'échelle de la translation. D'un autre côté, la contrainte épipolaire nous permet de prendre en compte des éléments de la scène qui ne sont pas modélisés *a priori*, et donc d'obtenir un point de vue plus précis que dans le cas du 3D/2D pur.

### 2.4.2 Détection et appariement des points 2D

L'obtention d'un nombre suffisant de couples de correspondants $(\mathbf{q},\mathbf{q}')$ est effectué de façon automatique, en deux étapes. Des points d'intérêt sont détectés par la méthode de Harris et Stephens dans l'image courante et l'image précédente de la séquence, puis ces points sont appariés par la méthode de Zhang et al.

### Détection des points d'intérêt

Un point d'intérêt correspond à un changement bidimensionnel d'une image. Comme l'image contient plus d'information en ces points qu'en des points correspondant à des changements unidimensionnels (comme les contours) ou à des régions homogènes, leur appariement est plus aisé. De nombreux détecteurs de points d'intérêt ont été proposés, mais le plus couramment utilisé est maintenant le détecteur de [Harris et al.88], qui prend en compte les dérivées premières de l'intensité de l'image (ou niveau de gris), en recherchant pour chaque pixel les valeurs propres de la matrice

$$e^{-\frac{x^2+y^2}{2\sigma^2}} \otimes \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix},$$

où $\otimes$ est le produit de convolution et $I_x = \partial I / \partial x$. Ces valeurs propres sont en fait les courbures principales de la fonction d'auto-corrélation du signal image: si ces deux courbures sont grandes, ceci indique une forte variation des niveaux de gris autour du pixel, et donc la présence d'un point d'intérêt. La première ligne de la figure 2.12 montre les points obtenus grâce au détecteur de Harris sur les deux premières images de la séquence de la place Stanislas.

### Appariement

Les points ainsi détectés sont appariés selon l'algorithme de [Zhang et al.94], qui utilise une technique de relaxation pour choisir les couples de correspondants parmi les candidats. Nous la résumons ici.

- **Identification des couples plausibles:** Autour de chaque point d'intérêt $\mathbf{q_i}$ de l'image 1, une zone de recherche (un rectangle) est définie, d'où sont extraits les points d'intérêt $\mathbf{q'_j}$ de l'image 2. La taille de cette zone de recherche est choisie en fonction de la disparité maximale estimée entre les images. Pour chaque couple de points $(\mathbf{q_i},\mathbf{q'_j})$, un *score de corrélation* normalisé $c_{ij}$ est calculé, qui évalue la similarité entre les fenêtres de corrélation autour des deux points. Ce score est compris entre -1 pour des fenêtres de corrélation complètement dissemblables et $+1$ pour des fenêtres identiques. Un seuil est alors utilisé (typiquement 0.8) pour sélectionner les couples plausibles, ou *appariements candidats*.
- **Définition d'un score d'appariement:** À l'issue de l'étape précédente, un point de la première image est susceptible d'être apparié avec plusieurs points de la seconde image, et vice versa. Considérons un appariement candidat $(\mathbf{q_i},\mathbf{q'_j})$. Soit $\mathcal{N}(\mathbf{q})$ les points voisins de $\mathbf{q}$ dans un disque de rayon $R$. Si $(\mathbf{q_i},\mathbf{q'_j})$ est un appariement correct, nous nous attendons à obtenir plusieurs appariements $(\mathbf{p_k},\mathbf{p'_l})$, où $\mathbf{p_k} \in \mathcal{N}(\mathbf{q_i})$ et $\mathbf{p'_l} \in \mathcal{N}(\mathbf{q'_j})$, tels que la position de $\mathbf{p_k}$ relative à $\mathbf{q_i}$ soit à peu près la même que celle de $\mathbf{q'_l}$ relative à $\mathbf{q'_j}$. À l'inverse, si $(\mathbf{q_i},\mathbf{q'_j})$ est un mauvais appariement, nous nous attendons à trouver peu d'appariements communs, voire pas du tout, dans leur voisinage. Le *score d'appariement* utilisé pour la relaxation, $SM(\mathbf{q_i},\mathbf{q'_j})$ ($SM$ pour *Strength of the Match*), repose sur ce constat. Son expression exacte, reportée dans [Zhang et al.94], est relativement complexe. Pour simplifier, $SM(\mathbf{q_i},\mathbf{q'_j})$ est égal au score de corrélation $c_{ij}$, multiplié par le nombre d'appariements voisins dont les positions relatives sont identiques, c'est-à-dire dont le rapport

$$r = \frac{|d(\mathbf{q_i},\mathbf{p_k}) - d(\mathbf{q'_j},\mathbf{p'_l})|}{[d(\mathbf{q_i},\mathbf{p_k}) + d(\mathbf{q'_j},\mathbf{p'_l})]/2}$$

est petit. $SM$ est symétrique, c'est-à-dire que $SM(\mathbf{q_i},\mathbf{q'_j}) = SM(\mathbf{q'_j},\mathbf{q_i})$.

- **Élimination des ambiguïtés par relaxation:** La technique de relaxation utilisée rompt avec les techniques classiques du "*vainqueur prend tout*", qui aboutit fréquemment à un minimum local, ou du "*perdant ne prend rien*", qui n'est pas symétrique. La technique du "*vainqueur prend tout*" consiste à considérer immédiatement comme corrects les appariements les plus plausibles, c'est-à-dire les couples $(\mathbf{q_i},\mathbf{q'_j})$ dont les points $\mathbf{q_i}$ ou $\mathbf{q'_j}$ n'obtiennent pas de score d'appariement plus élevé avec n'importe quel autre couple plausible qu'ils peuvent former. Tous les appariements associés aux points $\mathbf{q_i}$ et $\mathbf{q'_j}$ sont alors éliminés, et le processus est recommencé avec les appariements non éliminés. La technique du "*perdant ne prend rien*" revient à éliminer à chaque étape le point de l'image 1 qui obtient le score d'appariement le plus faible, jusqu'à ce qu'il ne reste plus que un et un seul candidat pour chaque point.
  La technique mise en œuvre peut être appelée "*certains vainqueurs prennent tout*". Tout comme pour la technique du "*vainqueur prend tout*", les $p$ appariements $\{\mathcal{P}_i\}$ obtenant les scores $SM$ les plus élevés sont qualifiés d'*appariements potentiels*, et rangés par ordre décroissant dans une table appelée $\mathcal{T}_{SM}$. Cependant, certains appariements peuvent obtenir un score d'appariement $SM$ élevé, tout en étant ambigus. Une deuxième table $\mathcal{T}_{NA}$ est donc créée, contenant des valeurs elles aussi classées par ordre décroissant, et indiquant dans quelle mesure chaque appariement $\mathcal{P}_i$ est non ambigu : $NA = 1 - SM^{(2)}/SM^{(1)}$, où $SM^{(1)}$ est le score $SM$ de $\mathcal{P}_i$ et $SM^{(2)}$ est le score $SM$ du deuxième meilleur appariement. Pour finir, les appariements potentiels $\mathcal{P}_i$ appartenant à la fois aux $qp$ premiers appariements de $\mathcal{T}_{SM}$, où $q \in ]0;1]$, et aux $qp$ premiers appariements de $\mathcal{T}_{NA}$ sont considérés comme corrects. Ainsi, les appariements potentiels ambigus ne sont pas sélectionnés même s'ils ont obtenu un score $SM$ élevé, et inversement les appariements ayant obtenu un score $SM$ petit ne sont pas sélectionnés, même s'ils ne sont pas ambigus.

La figure 2.12 montre le résultat d'un appariement entre deux images consécutives de la séquence.

### 2.4.3 Résultats

Un exemple de résultat pour la méthode hybride est présenté figure 2.13, qui montre la trajectoire retrouvée pour la caméra, et l'incrustation d'un véhicule dans deux images de la séquence, cette fois positionné correctement.

Cependant, on peut noter, dans le cas d'une animation où l'objet virtuel est éloigné des primitives 3D et proche de la caméra, un tremblement (*jittering* en anglais) de l'objet inséré. Pour éliminer ce problème, nous utilisons l'ajustement de faisceaux, décrit dans la partie 2.2.5, et dont nous détaillons l'implantation dans la suite.

Remarquons que la méthode hybride considère les images de façon séquentielle, elle peut donc être théoriquement utilisée dans un contexte temps réel. L'ajustement de faisceaux tel qu'il est présenté ici, en considérant globalement les images, supprime cet aspect séquentiel, et ne permet plus une utilisation en temps réel. Cependant, des approches de type ajustement de faisceaux pour le temps réel, qui prennent en compte toutes les images précédant l'image courante, commencent à apparaître [Mclauchlan00].

## 2.5 Ajustement de faisceaux

L'ajustement de faisceaux a été décrit partie 2.2.5. Nous présentons maintenant notre implantation qui nous a permis d'améliorer les points de vue obtenus par les méthodes précédentes.

FIG. 2.12 – *Exemple de points d'intérêt et d'appariements obtenus pour les deux premières images de la séquence (pour plus de visibilité, seule une partie de l'image est représentée). Les flèches relient les points d'intérêt de l'image 1, qui est affichée, aux points d'intérêt correspondants dans l'image 2.*


FIG. 2.13 – *a: Trajectoire de la caméra retrouvée avec la méthode hybride. b et c: Deux images de la séquence avec l'incrustation d'une voiture.*

.

### 2.5.1 Implantation

#### Méthode numérique

Nous avons tout d'abord implanté une résolution numérique appelée gradient conjugué, en utilisant la bibliothèque logicielle Numerical Recipes (voir [Press et al.88]). La méthode du gradient conjugué est une méthode itérative qui effectue un ensemble de minimisations à une dimension effectuées dans la direction courante du gradient de la fonction minimisée. Dans notre implantation, l'expression des dérivées a été déterminée grâce au logiciel de calcul formel Maple. D'après [Triggs et al.00], le gradient conjugué est la méthode du premier ordre la plus efficace pour l'ajustement de faisceaux.

#### Séparation points/projections

Nous avons également implanté une résolution basée sur la séparation des points 3D et des projections essentiellement pour sa simplicité. Elle nous a également permis d'éviter les erreurs d'estimation due aux points 3D *outliers*. Nous précisons ici son implantation:

**Initialisation:**

- Soit $(\alpha_i,\beta_i,\gamma_i,t_{xi},t_{y_i},t_{zi})$ les paramètres externes pour chaque image, estimés par notre méthode basée modèle ou hybride;
- Soit $\{\mathbf{m}_{ij}\}$, les points 2D suivis sur la séquence. Les points $\{\mathbf{m}_{i_1 j} \dots \mathbf{m}_{i_2 j}\}$ sont les reprojections du point 3D $\mathbf{M}_j$ dans les images $i_1$ à $i_2$.
- $\sigma(i,j) = 1$ si $\mathbf{M}_j$ est suivi dans l'image $i$, et 0 sinon;
- Il faut également disposer d'une estimation des points 3D. Pour cela, nous reconstruisons le point $\mathbf{M}_j$ par triangulation en utilisant les images $i_1$ et $i_2$.

**Minimisation sur les points:** Les paramètres $(\alpha_i,\beta_i,\gamma_i,t_{xi},t_{y_i},t_{zi})$ étant fixés, nous recherchons les points 3D qui minimisent l'erreur de reprojection:

$$\min_{\mathbf{M}_j} \sum_{i=i_1}^{i_2} \mathrm{dist}(\mathbf{P}_i\mathbf{M}_j,\mathbf{m}_{ij})^2$$

$\mathbf{M}_j$ est alors retrouvé à l'aide d'une minimisation numérique, initialisée grâce à l'estimation de $\mathbf{M}_j$ précédente. Une autre estimation de $\mathbf{M}_j$ consiste à rechercher la solution au sens des moindres carrés du système:

$$\begin{pmatrix} \mathbf{P}_{i11} - \mathbf{m}^u_{ij}\mathbf{P}_{i31} & \mathbf{P}_{i12} - \mathbf{m}^u_{ij}\mathbf{P}_{i32} & \mathbf{P}_{i13} - \mathbf{m}^u_{ij}\mathbf{P}_{i33} \\ \mathbf{P}_{i21} - \mathbf{m}^v_{ij}\mathbf{P}_{i31} & \mathbf{P}_{i22} - \mathbf{m}^v_{ij}\mathbf{P}_{i32} & \mathbf{P}_{i23} - \mathbf{m}^v_{ij}\mathbf{P}_{i33} \\ & \dots & \end{pmatrix} \mathbf{M}_j = \begin{pmatrix} \mathbf{m}^u_{ij}\mathbf{P}_{i34} - \mathbf{P}_{i14} \\ \mathbf{m}^u_{ij}\mathbf{P}_{i34} - \mathbf{P}_{i24} \\ \dots \end{pmatrix}$$

Le critère linéaire minimisé ici n'a pas de sens physique, contrairement à l'erreur de reprojection, et il est bien connu que de tels critères donnent généralement de moins bonnes estimations. En revanche, le temps de calcul consacré à cette estimation est beaucoup plus réduit que celui d'une minimisation numérique. Après expérimentations, nous avons constaté que l'estimation aux moindres carrés ne perturbait pas les résultats, c'est donc celle-ci que nous avons retenue. Ceci est sans doute dû à l'estimation initiale des points de vue utilisée, qui est très proche de la solution.

**Minimisation sur les projections:** Les points 3D $\{\mathbf{M}_j\}$ étant fixés, nous recherchons les projections qui minimisent l'erreur de reprojection des points visibles dans l'image considérée:

$$\min_{\mathbf{P}_i} \sum_j \sigma(i,j)\mathrm{d}(\mathbf{P}_i\mathbf{M}_j,\mathbf{m}_{ij})^2$$

FIG. 2.14 – *Images 15, 100 et 150 de la Séquence Stanislas.*

Si les paramètres internes sont connus, on peut paramétrer les $\mathbf{P}_i$ par les paramètres externes seuls $(\alpha_i,\beta_i,\gamma_i,t_{xi},t_{y_i},t_{zi})$, qui sont alors retrouvés à l'aide d'une méthode numérique.

Si les paramètres internes ne sont pas connus *a priori*, les 11 coefficients des matrices de projection peuvent être estimés à l'aide d'un moindres carrés:

$$A = \begin{pmatrix} \mathbf{M}x_i & \mathbf{M}y_i & \mathbf{M}z_i & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & -\mathbf{m}^u_{ij}\mathbf{M}x_i & -\mathbf{m}^u_{ij}\mathbf{M}y_i & -\mathbf{m}^u_{ij}\mathbf{M}z_i \\ 0.0 & 0.0 & 0.0 & 0.0 & \mathbf{M}x_i & \mathbf{M}y_i & \mathbf{M}z_i & 1.0 & -\mathbf{m}^v_{ij}\mathbf{M}x_i & -\mathbf{m}^v_{ij}\mathbf{M}y_i & -\mathbf{m}^v_{ij}\mathbf{M}z_i \\ & & & & & & & \cdots & & & \end{pmatrix}$$

$$A\left(\mathbf{P}_{i11}\ \mathbf{P}_{i12}\ \mathbf{P}_{i13}\ \mathbf{P}_{i14}\ \mathbf{P}_{i21}\ \mathbf{P}_{i22}\ \mathbf{P}_{i23}\ \mathbf{P}_{i24}\ \mathbf{P}_{i31}\ \mathbf{P}_{i32}\ \mathbf{P}_{i33}\right)^t = \begin{pmatrix} \mathbf{m}^u_{ij} \\ \mathbf{m}^v_{ij} \\ \cdots \end{pmatrix}$$

qui présente le même inconvénient que le critère linéaire au niveau des points, mais qui est également plus rapide qu'une minimisation numérique. De plus, laisser varier ainsi les coefficients de la projection peut lui faire perdre son caractère euclidien, et optimiser simultanément paramètres internes et externes peut créer des ambiguïtés de mouvement (entre une translation le long de l'axe optique et un changement de focale, par exemple; voir [Simon et al.99]). Nous verrons que c'est effectivement souvent le cas dans nos expérimentations.

**Itération:** Les deux minimisations sont itérées, jusqu'à ce que l'erreur de reprojection moyenne passe sous un seuil fixé manuellement.

**Prise en compte de la présence de points 2D mal suivis** Détecter les points 2D mal suivis (*outliers*) est souvent indispensable en pratique. Plusieurs approches sont possibles : [Luong92] a utilisé les estimateurs robustes pour l'estimation de la matrice fondamentale, [Torr et al.97] utilisent l'approche RANSAC pour le calcul du tenseur trifocal.

Nous avons retenue une solution très simple : soit $\sigma$, l'écart-type des erreurs de reprojection normalisées. Les points tels que leur erreur de reprojection est supérieure à $3\sigma$ sont supposés *outliers* et et ne seront pas pris en compte lors de l'itération suivante. Ils seront pris en compte pour les itérations ultérieures si leur erreur de reprojection devient inférieure à 3 fois l'écart-type courant.

### 2.5.2 Expérimentations

#### Séquence Stanislas

Cette séquence a été filmée d'une voiture en translation selon une direction approximativement orthogonale à l'axe de la caméra. La figure 2.14 présente quelques images de cette séquence.

Nous avons testé les deux méthodes décrites plus haut, sur la séquence Stanislas, avec comme estimation initiale, les 76 points de vue obtenus par notre méthode hybride (figure 2.15.a) et, comme points 2D, un jeu de 2590 points présentant de nombreux *outliers* ou un jeu de 2066 points où ces *outliers* ont été supprimés manuellement. Les figures 2.15 et 2.16 résument les résultats obtenus selon le jeu de points 2D utilisés, qu'on n'ait laissé varié que les paramètres externes ou l'ensemble des coefficients des matrices de projection, avec ou sans la gestion des *ouliers* présentée section 2.5.1 (étape 3).

Le graphique 2.15.b permet de comparer l'évolution du résidu moyen pour nos deux méthodes décrites plus haut: le gradient conjugué se révèle beaucoup plus lent. Tous les autres résultats présentés ont été obtenus à partir de la méthode de séparation points/projections.

On peut constater que, pour cette séquence, optimiser les 11 coefficients des projections permet d'obtenir un résultat satisfaisant, mais seulement en l'absence d'*ouliers*: dans le cas contraire, le processus diverge (voir figure 2.16.c). En présence d'*outliers*, leur détection est nécessaire et permet d'obtenir un résultat acceptable (voir figure 2.16.a et b).

#### Séquence du chalet

Cette séquence a été filmée à la main par un observateur en mouvement. La figure 2.17 présente quelques images de cette séquence. Les points de vue initiaux ont été obtenus par la méthode hybride.

Pour cette séquence, les deux méthodes n'ont pas convergé directement vers une solution correcte. Cela est sans doute dû à la nature de la trajectoire de la caméra, puisqu'elle commence par une translation arrière et finit par une translation avant, cas défavorables à l'estimation des points 3D. Nous avons donc conservé uniquement les points suivis sur suffisamment d'images (20, pour une séquence de 120 images), parmi lesquels restaient un faible nombre d'*outliers*. Les résultats sont présentés figure 2.18, uniquement pour la séparation points/projections. Encore une fois, se limiter à l'estimation des paramètres externes permet de retrouver une trajectoire correcte, alors que laisser varier les 11 coefficients des matrices de projection donne un mauvais résultat (voir notamment la confusion entre translation selon l'axe optique et changement de focale figure 2.18.c).

### 2.5.3 Remarques sur l'ajustement de faisceaux

La séparation points/projections, bien que très simple à implanter, nous a permis d'améliorer de façon significative l'estimation de la trajectoire des caméras obtenue par notre méthode hybride. On peut remarquer également qu'elle converge plus rapidement, ce qui est en contradiction avec les résultats présentés dans [Triggs et al.00]. De plus, l'optimisation sur l'ensemble des coefficients des matrices de projection, bien que souvent proposée dans la littérature, nous semble très aléatoire.

## 2.6 Conclusion

Nous venons de présenter différentes méthodes pour calculer la trajectoire d'une caméra grâce aux images de la séquence filmée. Les méthodes basée modèle et hybride développées dans l'équipe permettent d'estimer les points de vue, pour des scènes d'intérieur ou extérieures. Si les résultats de la méthode basée modèle sont assez dépendants de la distribution des primitives 3D utilisées, la méthode hybride permet de résoudre ce problème. Enfin, l'étape finale d'ajustement de faisceaux permet d'améliorer la précision des points de vue. Elle est assez sensible aux erreurs

de suivi de points 2D, et une supervision de l'utilisateur pour éviter de prendre en compte les points mal suivis reste parfois nécessaire.

De plus, l'erreur sur les points de vue estimés n'est pas toujours négligeable, même après ajustement de faisceaux. Pouvoir tenir compte de ces erreurs peut être très intéressant, notamment quand on utilise les points de vue pour reconstruire des primitives 3D à partir de leurs reprojections dans les images. Le chapitre suivant montre comment estimer cette erreur, et le chapitre 6 montre comment utiliser cette erreur pour estimer l'erreur sur la reconstruction de courbes 3D et leurs reprojections.

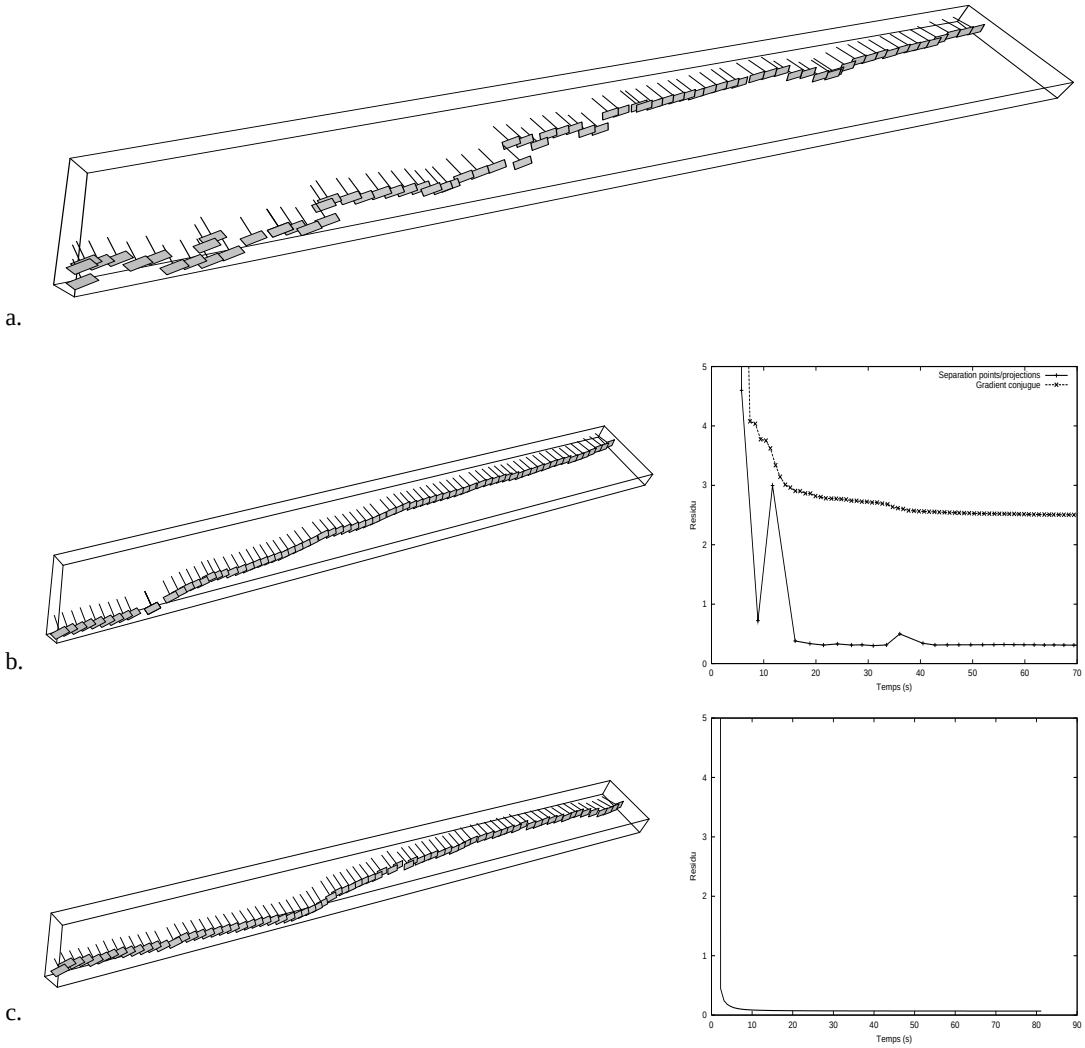



FIG. 2.15 – **Séquence Stanislas (76 images, 2066 points 3D)** *a: Trajectoire estimée par notre méthode hybride; b et c : Trajectoires et évolutions du résidu moyen en fonction du temps de calcul pour le jeu de points sans outliers b: en optimisant seulement les paramètres externes, c: tous les coefficients des matrices de projection.*

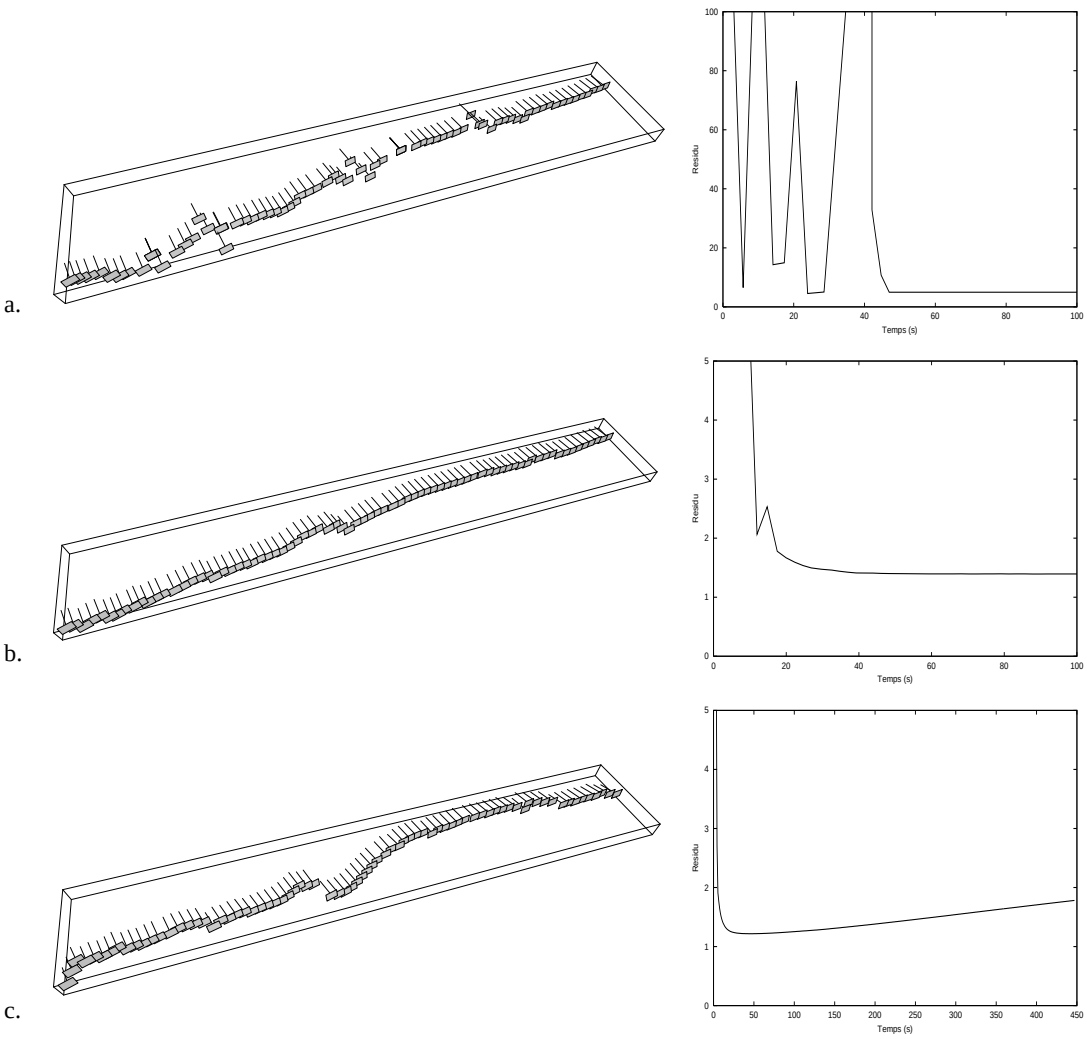



FIG. 2.16 – **Séquence Stanislas (76 images, 2590 points 3D dont** *outliers*) *Trajectoires et évolutions du résidu moyen en fonction du temps de calcul pour le jeu de points avec outliers; a: sans détection des outliers; b: avec détection des outliers, en optimisant seulement les paramètres externes, c: avec détection des outliers, en optimisant tous les coefficients des matrices de projection.*

FIG. 2.17 – *Images 0, 60 et 120 de la séquence du chalet.*

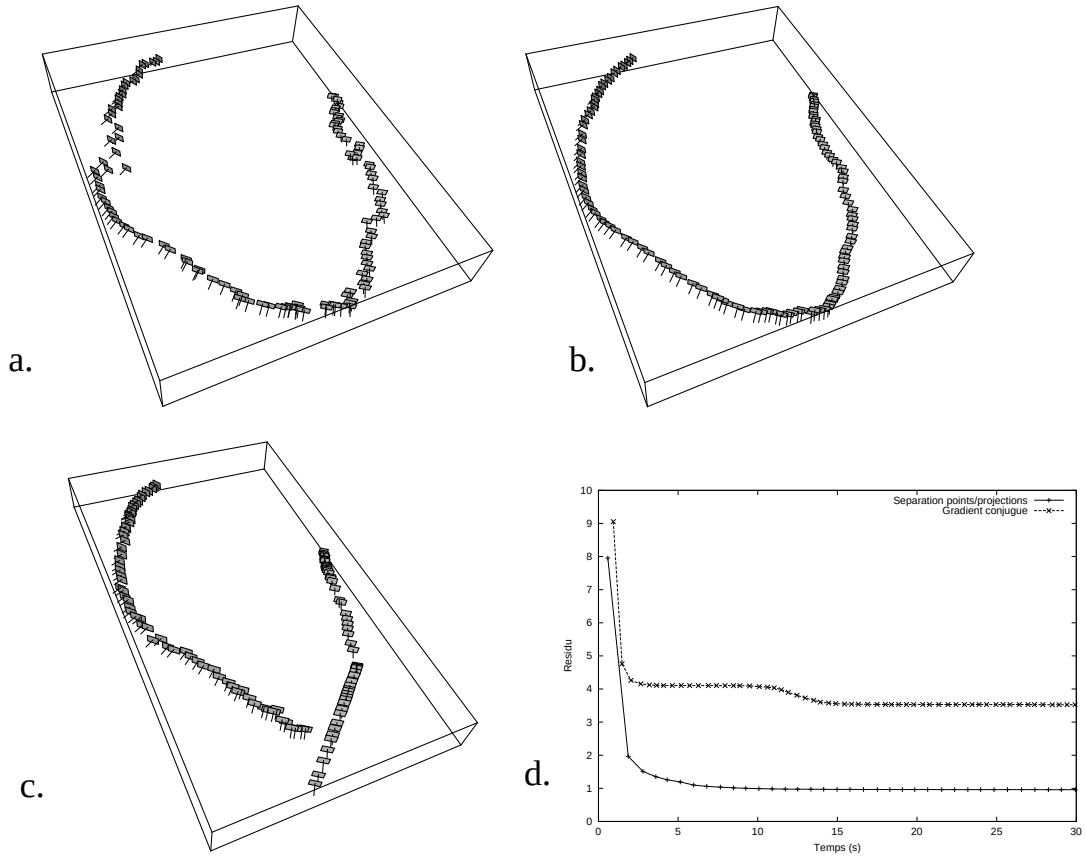



FIG. 2.18 – **Séquence du chalet (120 images, 218 points)** *a. Estimation initiale; b. Trajectoire retrouvée en optimisant seulement les paramètres externes; c. Trajectoire retrouvée en optimisant tous les coefficients des matrices de projection; d. comparaison de l'évolution du résidu moyen entre la séparation points/projections et le gradient conjugué.*

# Chapitre 3

# Estimation de l'erreur sur les points de vue

Nous venons de voir dans le chapitre précédent comment estimer les points de vue sur une séquence d'images. Cependant, l'incertitude sur cette estimation n'est pas négligeable, et il peut être intéressant d'estimer également cette erreur. Nous présentons donc dans ce chapitre la méthode que nous avons retenue pour réaliser cette estimation et les expérimentations effectuées. Nous verrons dans le chapitre 6 comment prendre en compte cette erreur pour évaluer ensuite l'erreur obtenue sur la reconstruction puis sur la reprojection de courbes tridimensionnelles effectuées suivant les points de vue estimés.

## 3.1 Causes de l'erreur

Un effort important a été réalisé dans la méthode présentée dans le chapitre précédent pour ne pas tenir compte des erreurs de mise en correspondance (*outliers*) lors de l'estimation des points de vue. Même ainsi, les résultats obtenus peuvent être imprécis, à cause d'une mauvaise distribution spatiale des courbes 3D (comme c'est le cas dans l'exemple de la figure 2.11) et du bruit des données utilisées. Ces données, dans le cas de la méthode basée modèle, sont:

- les courbes 3D, dont la position spatiale a pu être mal estimée;
- les courbes 2D, dont la position dans l'image n'est retrouvée qu'à une fraction de pixel près par les contours actifs;
- les paramètres internes, qui sont supposés connus par l'algorithme.

À celles-ci s'ajoutent dans le cas de la méthode hybride:

- les appariements de points 2D dont la position n'est connue qu'à une fraction de pixel près.

Les données utilisées par l'ajustement de faisceaux sont le suivi de ces points 2D sur la séquence, et les paramètres internes, même si les résultats dépendent fortement de l'initialisation.

Dans ce chapitre, nous montrons comment nous estimons l'erreur des points de vue, pour chacune des trois méthodes présentées précédemment : les méthodes basée modèle et hybride, ainsi que l'ajustement de faisceaux.

Afin de simplifier cette estimation, nous éliminons les appariements qui ont été déterminés comme *outliers* par les estimateurs robustes, ce qui permet de ne plus faire apparaître ces estimateurs robustes dans l'expression du résidu au voisinage du minimum. Après cette élimination, le résidu minimisé par la méthode basée modèle est donc:

$$\Phi(\mathbf{p}) = \Phi_{\text{3D2D}}(\mathbf{p}) = \frac{1}{N}\sum_i \frac{1}{l'_i}\sum_j Dist^2(\mathbf{m}'_{i,j},c_i),$$

avec $N$ le nombre de primitives restantes (les raisons de cette normalisation apparaîtront plus bas, elle ne change évidemment pas le minimum). $\mathbf{m}'_{i,j}$ est un point 2D, et $c_i$ est la projection (qui dépend de $\mathbf{p}$) d'une courbe 3D $C_i$ ($c_i = \text{Proj}_{\mathbf{p}}(C_i)$). Afin de simplifier l'écriture, nous noterons ce critère comme une seule somme de résidus:

$$\Phi_{\text{3D2D}}(\mathbf{p}) = \frac{1}{n}\sum_i r_i^2, \tag{3.1}$$

avec $n$ le nombre total de points $\mathbf{m}'_{i,j}$ et

$$r_i = \text{Dist}^2(\mathbf{m}'_{\sigma_1(i),\sigma_2(i)},c_{\sigma_1(i)}),$$

où $\sigma_1$ et $\sigma_2$ sont deux fonctions permettant cette renumérotation.

De même, le résidu minimisé par la méthode hybride sera noté:

$$\Phi(\mathbf{p}) = \frac{1}{2}\left(\Phi_{\text{3D2D}}(\mathbf{p}) + \Phi_{\text{2D2D}}(\mathbf{p})\right) = \frac{1}{2}\left(\frac{1}{n}\sum_i r_i^2 + \frac{\lambda}{m}\sum_i v_i^2\right). \tag{3.2}$$

Enfin, le critère minimisé par l'ajustement de faisceaux vaut:

$$\Phi_{\text{ajust}}(\mathbf{P},\mathbf{M}) = \sum_{i,j}\sigma(i,j).\text{Dist}^2(\text{Proj}_{\mathbf{P}_i}(\mathbf{M}_j),\mathbf{m}_{ij}) \tag{3.3}$$

avec $\sigma(i,j) = 1$ si $\mathbf{M}_j$ est suivi dans l'image $i$, et 0 sinon, $\mathbf{P}_i$ le point de vue associé à l'image $i$, et $\mathbf{m}_{ij}$ le point 2D associé à $\mathbf{M}_j$ dans l'image $i$.

## 3.2 Estimation de l'erreur

Dans cette partie, nous rappelons que l'erreur peut être modélisée par une matrice de covariance, ainsi que des méthodes classiques pour l'estimation de cette matrice.

### 3.2.1 Modélisation de l'erreur

Dans la suite, les données seront représentées par le vecteur $\mathbf{x}$, le résultat du calcul du point de vue par le vecteur $\mathbf{y}$. L'erreur sur $\mathbf{y}$, vue comme un vecteur aléatoire, peut être représentée par sa matrice de covariance $\Lambda_{\mathbf{y}}$:

$$\Lambda_{\mathbf{y}} = E[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^t]$$

où $E[\mathbf{y}]$ représente l'espérance de $\mathbf{y}$. La variable aléatoire

$$\chi^2 = (\mathbf{y} - E[\mathbf{y}])\Lambda_{\mathbf{y}}^{-1}(\mathbf{y} - E[\mathbf{y}])^t$$

suit une loi du $\chi^2$ à $r$ degrés de liberté, où $r$ est le rang de la matrice $\Lambda_{\mathbf{y}}$, en supposant que $\mathbf{y}$ suit une distribution Gaussienne. On note $P_{\chi^2}(k,r) = P(\chi < k)$, c'est-à-dire la probabilité que $\chi$ soit inférieur à une valeur $k$. En pratique, on choisit une probabilité $P_{\chi^2}(k,r)$ raisonnable (par exemple, $P_{\chi^2}(k,r) = 90\%$), et une table du $\chi^2$ fournit une valeur $k$ en fonction de cette probabilité et de $r$. $k$ permet alors de définir un hyper-ellipsoïde:

$$(\mathbf{y} - E[\mathbf{y}])\Lambda_{\mathbf{y}}^{-1}(\mathbf{y} - E[\mathbf{y}])^t \leq k^2 \tag{3.4}$$

La probabilité que $\mathbf{y}$ soit dans cet hyper-ellipsoïde est égale à $P_{\chi^2}(k,r)$.

Nous rappellerons comment calculer $\Lambda_{\mathbf{y}}$ tout d'abord statistiquement, puis analytiquement, en fonction de $\Lambda_{\mathbf{x}}$, dans le cas où $\mathbf{y} = \phi(\mathbf{x})$. Nous verrons ensuite le calcul analytique de $\Lambda_{\mathbf{y}}$ dans le cas qui nous intéresse plus particulièrement, à savoir quand la relation entre $\mathbf{x}$ et $\mathbf{y}$ est définie de façon implicite par la minimisation d'un critère positif $C(\mathbf{x},\mathbf{y})$.

### 3.2.2 Cas général, méthode statistique

La méthode statistique consiste à utiliser la loi des grands nombres pour approximer la moyenne. On considère alors un nombre important de sous-ensembles des données $\mathbf{x}$. Chaque sous-ensemble permet de calculer une réalisation $\mathbf{y}_i$ de $\mathbf{y}$, et si nous en avons un nombre $N$ suffisant, alors $E[\mathbf{y}]$ peut être approximée par:

$$E_N[\mathbf{y}] = \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i,$$

et $\Lambda_{\mathbf{y}}$ est alors approximée par :

$$\Lambda_{\mathbf{y}} = E_N\left[(\mathbf{y} - E_N[\mathbf{y}])(\mathbf{y} - E_N[\mathbf{y}])^t\right]$$

Cette méthode est évidemment coûteuse en temps de calcul, et suppose une bonne répartition des mesures. C'est pourquoi on a plutôt recours à un calcul analytique permettant d'approximer $\Lambda_{\mathbf{y}}$ lorsque $\mathbf{y}$ est une fonction analytique de $\mathbf{x}$.

### 3.2.3 y fonction de x, méthode analytique

Considérons le cas où $\mathbf{y}$ est calculée à partir des données $\mathbf{x}$ grâce à une fonction $\phi$:

$$\mathbf{y} = \phi(\mathbf{x})$$

$\Lambda_{\mathbf{y}}$ peut être alors approchée analytiquement. Si $\phi$ est $C^1$, elle peut être approximée par son développement de Taylor au voisinage de $E[\mathbf{x}]$ par:

$$\phi(\mathbf{x}) \simeq \phi(E[\mathbf{x}]) + J_\phi(E[\mathbf{x}]).(\mathbf{x} - E[\mathbf{x}]),$$

avec $J_\phi(E[\mathbf{x}])$ le Jacobien de $\phi$ en $E[\mathbf{x}]$. On a donc:

$$\begin{array}{rcll}\Lambda_{\mathbf{y}} & = & E[(\phi(\mathbf{x}) - E[\phi(\mathbf{x})])(\phi(\mathbf{x}) - E[\phi(\mathbf{x})])^t] & \simeq \quad E[J_\phi(E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^t J_\phi(E[\mathbf{x}])^t] \\ & & & = \quad J_\phi(E[\mathbf{x}])E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^t]J_\phi(E[\mathbf{x}])^t\end{array}$$

On peut donc approximer au premier ordre la matrice de covariance de $\mathbf{y}$ en fonction de la matrice de covariance de $\mathbf{x}$:

$$\Lambda_{\mathbf{y}} \simeq J_\phi(E[\mathbf{x}])\Lambda_{\mathbf{x}}J_\phi(E[\mathbf{x}])^t \tag{3.5}$$

Dans notre cas, nous n'avons pas d'expression analytique du point de vue en fonction des données car $\mathbf{y}$ est obtenu par minimisation d'un critère. Ce cas a déjà été étudié dans [Csurka et al.97] dans le cadre du calcul de l'incertitude de la matrice fondamentale. Nous rappelons maintenant les fondements théoriques de la méthode.

### 3.2.4 y défini comme le minimum d'un critère $C(\mathbf{x},\mathbf{y})$ [Csurka et al.97]

Dans le cadre de l'estimation de l'incertitude de la matrice fondamentale, [Csurka et al.97] ont utilisé le résultat suivant, qui découle du théorème des fonctions implicites et qui est donné dans [Faugeras93]:

Soit une fonction $\Phi : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ de classe $C^\infty$, $\mathbf{x}_0 \in \mathbb{R}^m$ une mesure et $\mathbf{y}^* \in \mathbb{R}^p$ un minimum local de $\Phi(\mathbf{x}_0,\mathbf{z})$. Si le hessien $\mathbf{H}$ de $\Phi(\mathbf{x}_0,\mathbf{z})$ par rapport à $\mathbf{z}$ est inversible en $(\mathbf{x},\mathbf{y}) = (\mathbf{x}_0,\mathbf{y}^*)$, alors il existe un ouvert $U'$ de $\mathbb{R}^m$ contenant $\mathbf{x}_0$, un ouvert $U''$ de $\mathbb{R}^p$ contenant $\mathbf{y}^*$ et une fonction de classe $C^\infty$ $\phi : \mathbb{R}^m \leftarrow \mathbb{R}^p$ tels que pour $(\mathbf{x},\mathbf{y})$ appartenant à $U' \times U''$, les deux relations « $\mathbf{y}$ est un minimum local de $\Phi(\mathbf{x}_0,\mathbf{z})$ » et « $\mathbf{y} = \phi(\mathbf{x})$ » sont équivalentes. En outre, nous avons le résultat suivant:

$$J_\phi(\mathbf{x}) = -\mathbf{H}^{-1}\frac{\partial \mathbf{\Psi}}{\partial \mathbf{x}}$$

où

$$\mathbf{\Psi} = \left(\frac{\partial \Phi}{\partial \mathbf{z}}\right)^t \quad \text{et} \quad \mathbf{H} = \frac{\partial \mathbf{\Psi}}{\partial \mathbf{z}}$$

En prenant $\mathbf{x}_0 = E[\mathbf{x}]$ et $\mathbf{y}^* = E[\mathbf{y}]$, l'équation 3.5 devient:

$$\Lambda_\mathbf{y} \simeq \mathbf{H}^{-1}\frac{\partial \mathbf{\Psi}}{\partial \mathbf{x}}\Lambda_\mathbf{x}\frac{\partial \mathbf{\Psi}}{\partial \mathbf{x}}^t \mathbf{H}^{-t} \tag{3.6}$$

dans le cas général. Csurka et al. considèrent ensuite le cas particulier d'une fonction implicite définie par une somme de carrés, c'est-à-dire où $\Phi$ est de la forme:

$$\Phi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} r_i^2,$$

on a alors :

$$\begin{aligned} \mathbf{\Psi} &= 2\sum_i r_i \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)^t \\ \mathbf{H} = \frac{\partial \mathbf{\Psi}}{\partial \mathbf{z}} &= 2\sum_i \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)^t \left(\frac{\partial r_i}{\partial \mathbf{z}}\right) \text{ au premier ordre} \end{aligned}$$

L'équation 3.6 devient :

$$\begin{aligned} \Lambda_\mathbf{y} &= 4\mathbf{H}^{-1}\sum_i \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)^t \left(\frac{\partial r_i}{\partial \mathbf{x}}\right)\Lambda_\mathbf{x}\left(\frac{\partial r_i}{\partial \mathbf{x}}\right)^t \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)\mathbf{H}^{-t} \\ &= 4\mathbf{H}^{-1}\sum_i \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)^t \Lambda_{r_i}\left(\frac{\partial r_i}{\partial \mathbf{z}}\right)\mathbf{H}^{-t} \end{aligned}$$

[Csurka et al.97] font ensuite plusieurs hypothèses simplificatrices :

- la moyenne des $r_i$ au minimum est nulle ;
- les $r_i$ sont indépendants ;
- l'erreur sur les $r_i$ est également distribuée,

ce qui leur permet d'approximer la variance $\Lambda_{r_i}$ de $r_i$ à partir de la valeur du critère au minimum $\Phi(\mathbf{y}^*)$ en utilisant la loi des grands nombres :

$$\Lambda_{r_i} = \frac{\Phi(\mathbf{y}^*)}{n-p}.$$

Comme $\mathbf{H} \simeq 2\sum \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)^t \left(\frac{\partial r_i}{\partial \mathbf{z}}\right)$, l'équation 3.6 devient finalement :

$$\Lambda_{\mathbf{y}} = \frac{2\Phi(\mathbf{y}^*)}{n-p}\mathbf{H}^{-1}\mathbf{H}\mathbf{H}^{-t} = \frac{2\Phi(\mathbf{y}^*)}{n-p}\mathbf{H}^{-t}, \tag{3.7}$$

ou, puisque $\mathbf{H}$ est symétrique et que $n$ est en pratique beaucoup plus grand que $p$:

$$\Lambda_{\mathbf{y}} = \frac{2\Phi(\mathbf{y}^*)}{n}\mathbf{H}^{-1}. \tag{3.8}$$

## 3.3 Méthode retenue

Malheureusement, les hypothèses simplificatrices utilisées par Csurka et al., si elles sont valables dans le cas où les $r_i$ sont des résidus liés à des appariements de points 2D, ne le sont plus dans notre cas.

Tout d'abord, les $r_i$ sont pour nous une distance d'un point à une courbe, ils sont donc positifs et leur moyenne n'est pas nulle au minimum. Mais surtout, l'hypothèse d'indépendance entre les $r_i$ est ici fausse, puisque les points d'une même courbe ne sont pas indépendants. Pour s'en convaincre, posons temporairement $r_i = \mathrm{Dist}^2(m'_{\sigma_1(i),\sigma_2(i)}, c_{\sigma_1(i)})$ (oublions le terme de normalisation pour simplifier), et considérons l'évolution de $\Lambda_{\mathbf{y}}$ quand on divise le pas de discrétisation des courbes 2D par $m$:

- la valeur de $n$ dans l'équation 3.8 est multipliée par $m$;
- la valeur de $\Phi(\mathbf{y}^*)$ est approximativement multipliée par $m$;
- la valeur de $\mathbf{H}^{-1}$ est approximativement divisée par $m$.

Si l'on se réfère à l'équation 3.8, $\Lambda_{\mathbf{y}}$ est alors approximativement divisée par $m$. En changeant le pas de discrétisation, on peut donc arbitrairement rendre l'incertitude aussi petite que l'on veut. En fait, le problème nous semble se poser également pour les appariements de points 2D, puisqu'on ne peut pas considérer les points appartenant à un même objet comme indépendants.

### 3.3.1 Calcul de la matrice de covariance à un coefficient multiplicatif près

Les hypothèses simplificatrices de [Csurka et al.97] permettent d'approximer la variance $\Lambda_{r_i}$ à partir de la valeur du critère au minimum $\Phi(\mathbf{y}^*)$ en utilisant la loi des grands nombres. On vient cependant de montrer que l'hypothèse d'indépendance conduit à un résultat biaisé pour le problème que nous considérons. Contentons-nous de supposer que l'erreur sur les $r_i$ est également distribuée. Posons donc :

$$\forall i \quad \Lambda_{r_i} = \Lambda_r$$

En suivant le même raisonnement que Csurka et al., on obtient la relation :

$$\Lambda_{\mathbf{y}} \simeq 2\Lambda_r\mathbf{H}^{-1} \tag{3.9}$$

La matrice $\Lambda_{\mathbf{y}}$ est donc approximativement proportionnelle à l'inverse du hessien, ce qu'on notera :

$$\Lambda_{\mathbf{y}} \sim \mathbf{H}^{-1} \tag{3.10}$$

Le fait de déterminer la covariance à un facteur multiplicatif près n'est pas forcément gênant. En effet, cette matrice est utilisée pour retrouver une zone de présence du point de vue sur la base d'un test du $\chi^2$. Or ce test dépend de la confiance souhaitée. Il suffit alors d'ajuster la zone de présence par une transformation homothétique.

### 3.3.2 Interprétation géométrique : les régions d'indifférence

Nous recherchons, au delà de la matrice de covariance une région, similaire à celle définie par l'équation 3.4, dans laquelle se trouve le point de vue réel. En utilisant la relation 3.9, cette équation peut être ré-écrite en :

$$(\mathbf{y} - E[\mathbf{y}])\mathbf{H}(\mathbf{y})(\mathbf{y} - E[\mathbf{y}])^t \leq K^2 \tag{3.11}$$

où $K^2 = 2k^2\Lambda_r$. Il reste alors à déterminer $K$ pour obtenir une région satisfaisante, c'est-à-dire contenant effectivement le point de vue réel.

[Bard74] décrit une approche dite des régions d'indifférence. Nous allons voir qu'elle permet d'interpréter ce facteur $K^2$ en fonction de notre problème, et ainsi de lui donner une valeur réaliste.

L'idée en est qu'il est raisonnable de supposer que les valeurs du critère presque aussi petites que celle au minimum nous satisfont presque autant que celle au minimum. En fixant une valeur $\epsilon$ de différence acceptable, cette constatation nous permet de définir un ensemble de résultats satisfaisants, appelé $\epsilon$-région d'indifférence. En notant $\Phi$ le critère minimisé, et $\mathbf{y}^*$ le minimum trouvé, la $\epsilon$-région d'indifférence est définie par :

$$\epsilon\text{-région} = \{\mathbf{y} \text{ tels que } |\Phi(\mathbf{y}) - \Phi(\mathbf{y}^*)| \leq \epsilon\} \tag{3.12}$$

Quand $\Phi$ est continue et $\mathbf{y}^*$ son unique minimum, la $\epsilon$-région est un ensemble connexe, et dans un voisinage suffisamment petit de $\mathbf{y}^*$, on peut approximer $\Phi$ par son développement de Taylor au second ordre:

$$\Phi(\mathbf{y}) \simeq \Phi(\mathbf{y}^*) + \nabla\Phi(\mathbf{y}^*)^t.\delta\mathbf{y}^* + \frac{1}{2}\delta\mathbf{y}.\mathbf{H}(\mathbf{y}^*).\delta\mathbf{y}^* \tag{3.13}$$

où $\nabla\Phi(\mathbf{y}^*)$ et $\mathbf{H}(\mathbf{y}^*)$ désignent respectivement le gradient et le hessien de $\Phi$ calculés en $\mathbf{y} = \mathbf{y}^*$. Comme $\mathbf{y}^*$ est le minimum de $\Phi$, le gradient $\nabla\Phi(\mathbf{y}^*)$ est nul, et l'équation 3.13 devient:

$$\Phi(\mathbf{y}) \simeq \Phi(\mathbf{y}^*) + \frac{1}{2}\delta\mathbf{y}^t.\mathbf{H}(\mathbf{y}^*).\delta\mathbf{y}$$

La $\epsilon$-région d'indifférence est alors définie par:

$$\epsilon\text{-région} = \left\{\mathbf{y} \text{ tels que } |\delta\mathbf{y}^t.\mathbf{H}(\mathbf{y}^*).\delta\mathbf{y}^*| \leq 2\epsilon\right\} \tag{3.14}$$

et est donc un hyper-ellipsoïde. Le rapprochement des relations 3.11 et 3.14 nous fournit une relation entre $\epsilon$ et $K$ :

$$K^2 = 2\epsilon.$$

Or, on peut interpréter assez simplement $\epsilon$ en fonction du problème traité. Nous discuterons dans la suite de différents choix pour la valeur de $\epsilon$.

Le principal intérêt de cette méthode par rapport à ce qui a été vu précédemment est qu'il n'y a pas d'hypothèse d'indépendance statistique des données. En outre, elle nécessite uniquement de calculer $\mathbf{H}(\mathbf{y}^*)$, matrice composée des dérivées secondes de $\Phi$ par rapport aux éléments du vecteur $\mathbf{y}$ dont l'expression exacte peut être déterminée grâce à un logiciel de calcul symbolique. Son seul paramètre est $\epsilon$, ce qui évite d'avoir à évaluer la covariance des données. Après cette description générale, nous allons voir maintenant la mise en œuvre de cette méthode pour l'évaluation de l'erreur des points de vue. Les méthodes basée modèle et hybride sont assez similaires, et nous les traiterons dans la même partie. L'ajustement de faisceaux sera ensuite considéré.

## 3.4 Cas des méthodes basée modèle et hybride

### 3.4.1 Implantation

Dans le cas qui nous intéresse, nous avons:

$$\mathbf{y} = \mathbf{p} = (\alpha,\beta,\gamma,t_x,t_y,t_z)^t;$$

Si l'on considère la méthode basée modèle:

$$\Phi(\mathbf{p}) = \Phi_{\text{3D2D}}(\mathbf{p})$$

s'il s'agit de la méthode hybride:

$$\Phi(\mathbf{p}) = \frac{1}{2}\left(\Phi_{\text{3D2D}}(\mathbf{p}) + \Phi_{\text{2D2D}}(\mathbf{p})\right).$$

Et finalement:

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{p}}\left(\frac{\partial \Phi}{\partial \mathbf{p}}\right)^t.$$

Dans le cas de la méthode hybride:

$$\mathbf{H} = \frac{1}{2}\left(\frac{\partial}{\partial \mathbf{p}}\left(\frac{\partial \Phi_{\text{3D2D}}}{\partial \mathbf{p}}\right)^t + \frac{\partial}{\partial \mathbf{p}}\left(\frac{\partial \Phi_{\text{2D2D}}}{\partial \mathbf{p}}\right)^t\right) = \left(\frac{1}{n}\sum_i \frac{\partial}{\partial \mathbf{p}}\left(\frac{\partial r_i}{\partial \mathbf{p}}\right)^t + \frac{1}{m}\sum_i \frac{\partial}{\partial \mathbf{p}}\left(\frac{\partial v_i}{\partial \mathbf{p}}\right)^t\right)$$

Sous forme matricielle:

$$\mathbf{H} = \frac{1}{2n}\sum_i \begin{pmatrix} \frac{\partial^2 r_i}{\partial \alpha^2} & \frac{\partial^2 r_i}{\partial \alpha \partial \beta} & & \\ \frac{\partial^2 r_i}{\partial \beta \partial \alpha} & \frac{\partial^2 r_i}{\partial \beta^2} & \cdots & \\ & \vdots & \ddots & \\ & & & \frac{\partial^2 r_i}{\partial t_z^2} \end{pmatrix} + \frac{1}{2m}\sum_i \begin{pmatrix} \frac{\partial^2 v_i}{\partial \alpha^2} & \frac{\partial^2 v_i}{\partial \alpha \partial \beta} & & \\ \frac{\partial^2 v_i}{\partial \beta \partial \alpha} & \frac{\partial^2 v_i}{\partial \beta^2} & \cdots & \\ & \vdots & \ddots & \\ & & & \frac{\partial^2 v_i}{\partial t_z^2} \end{pmatrix}$$

Dériver directement $r_i$ est délicat, puisqu'il fait intervenir la distance entre un point et une courbe quelconque, pour laquelle on ne dispose pas d'expression analytique. Nous avons donc utilisé une approximation du terme $\text{Dist}^2(\mathbf{m}'_{i,j},c_i)$ qui est valable dans le voisinage qui nous intéresse, c'est-à-dire le voisinage de $\mathbf{p}^*$. Pour cela, nous remplaçons cette distance par la distance entre le point $\mathbf{m}'_{i,j}$ et la reprojection du point 3D sur la courbe $C_i$ qui lui correspond au minimum, ce qui est illustré figure 3.1. Nous considérons donc tout d'abord le point $\mathbf{m}$ de la courbe $c_i$ obtenue pour le minimum $\mathbf{p}^*$ le plus proche de $\mathbf{m}'_{i,j}$ (donc $\text{Dist}^2(\mathbf{m}'_{i,j},\mathbf{m}) = \text{Dist}^2(\mathbf{m}'_{i,j},c_i)$ au minimum). On associe alors à $\mathbf{m}'_{i,j}$ le point 3D $\mathbf{M}_{i,j}$ qui se projette en $\mathbf{m}$ et qui appartient à la courbe 3D $C_i$. $r_i$ devient finalement:

$$r_i = \text{Dist}(\mathbf{m}'_i,\text{Proj}_{\mathbf{p}}(\mathbf{M}'_i))$$

forme qui peut se dériver plus aisément. Dans notre implantation, les expressions des dérivées secondes de $r_i$ et de $v_i$ ont été calculées à l'aide du logiciel Maple.

### 3.4.2 Choix du paramètre $\epsilon$

Il reste à choisir la valeur de $\epsilon$. On peut considérer d'après la définition de la région d'indifférence par l'équation 3.12 que $\epsilon$ correspond à l'incertitude de détection sur ces données. La précision de détection des courbes 2D est inférieure au pixel, celles des points 2D de l'ordre du

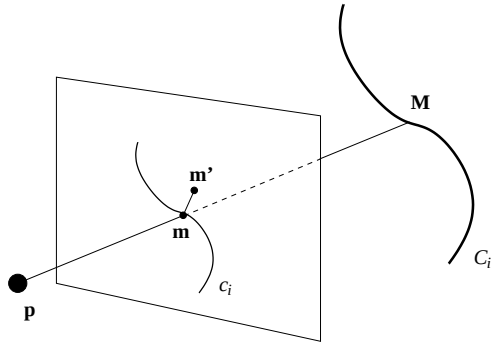



FIG. 3.1 – *Attribution du point 3D* $\mathbf{M}$ *au point 2D* $\mathbf{m}'$.

pixel. Comme nous avons normalisé le critère $\Phi$ par rapport au nombre de données, nous avons tout d'abord retenu $\epsilon = 1.0$ (pixel$^2$).

Cependant, il nous a semblé logique que cette région suive les variations du résidu au minimum $\Phi^*$: plus $\Phi^*$ est grand, plus cette région doit être grande. En effet, un résidu $\Phi^*$ grand traduit souvent le fait que les indices servant à calculer le point de vue ne sont pas dans une configuration permettant une détermination précise du point de vue. Nous avons donc défini $\epsilon$ comme étant proportionnel à $\Phi^*$:

$$\epsilon = k_\epsilon \Phi^*.$$

Les régions calculées en utilisant une valeur de 1.0 pour $k_\epsilon$ sont souvent manifestement surestimées (voir figures 3.4 et 3.5). L'intérêt de choisir une valeur de $\epsilon$ proportionnel au résidu apparaît surtout figure 3.6. Un des points de vue (encadré) est très mal retrouvé par la méthode hybride. La région retrouvée avec $\epsilon = 1.0$ est trop petite (figure 3.6.b). Comme le résidu pour ce point de vue est très important, en choisissant $\epsilon = \Phi^*$, la région retrouvée est plus grande pour ce point de vue que celles calculées pour les images proches, et elle contient manifestement le point de vue attendu.

Comme une surestimation est préférable à une sous-estimation, nous avons finalement retenu la solution $\epsilon = \Phi^*$.

### 3.4.3 Interprétation des résultats

Le résultat obtenu est donc un ellipsoïde $\Gamma$ à 6 dimensions centré en $\mathbf{p}^*$ d'équation

$$\Gamma : (\mathbf{p} - \mathbf{p}^*)^t \mathbf{H}(\mathbf{p}^*)(\mathbf{p} - \mathbf{p}^*) \leq 2\epsilon,$$

supposé contenir le point de vue réel $\overline{\mathbf{p}}$. Une bonne façon d'interpréter ce résultat consiste à considérer les axes et les sommets de $\Gamma$. Pour retrouver les axes de $\Gamma$, il suffit de mettre $\mathbf{H}(\mathbf{p}^*)$ sous la forme $\mathbf{VDV}^t$, où $\mathbf{D}$ est une matrice diagonale et $\mathbf{V}$ est la matrice de passage entre le repère de l'espace des paramètres $(\alpha,\beta,\gamma,t_x,t_y,t_z)$ et du repère propre de $\Gamma$. Chacun des douze sommets de $\Gamma$ est donné par:

$$\mathbf{s}_{\sigma,i} = \mathbf{p}^* + \sigma \sqrt{\frac{2\epsilon}{\mathbf{D}_{i,i}}} \mathbf{V}_i$$

où $\sigma = (+1)$ ou $(-1)$, $i$ est compris entre 1 et 6, et $\mathbf{V}_i$ est la $i^{\text{eme}}$ colonne de $\mathbf{V}$.

Il est à noter que le vecteur $\mathbf{t} = (t_x,t_y,t_z)^t$ est le vecteur intervenant dans la relation de passage entre le repère du monde et le repère de la caméra (voir chapitre 2): $\mathbf{M_c} = \mathbf{RM} + \mathbf{t}$ et ne

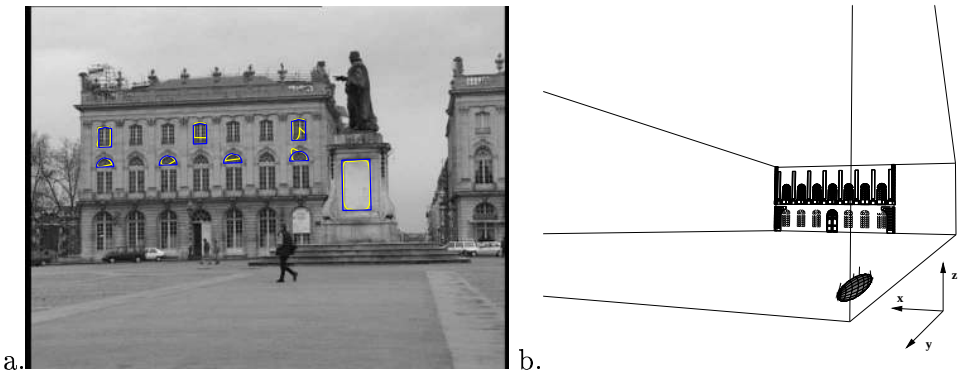



FIG. 3.2 – *a. Une image de la séquence Stanislas et les primitives 3D; b. ellipsoïde d'incertitude pour cette image (on a représenté également l'orientation de la caméra aux sommets).*

correspond pas aux coordonnées du centre $\mathbf{C}$ de la caméra dans le repère du monde (on a $\mathbf{C} = -\mathbf{R}^t\mathbf{t}$). Comme le centre de la caméra est plus facile à interpréter que ce vecteur, nous utiliserons le centre de la caméra correspondant aux sommets plutôt que $\mathbf{t}$ dans les tableaux de comparaison. De même, pour la visualisation des ellipsoïdes, que nous limiterons au monde 3D pour des raisons évidentes de représentation, nous représenterons les centres de caméra correspondant aux points à la surface de l'ellipsoïde.

### 3.4.4 L'incertitude est plus importante en profondeur

Un des premiers résultats qualitatifs qui apparaît sur la séquence Stanislas est que l'incertitude est plus importante dans un axe approximativement dirigé vers les primitives. Ce phénomène est signalé dans [Hartley et al.00]. Considérons une des images de la séquence Stanislas, présentée figure 3.2.a. Dans un premier temps, le point de vue a été évalué à l'aide de la méthode basée modèle ($\Phi = \Phi_{3D2D}$), avec des primitives 3D extraites de la façade de l'Opéra, sur un plan d'équation $y = 85$ m environ, et de la statue ($y \simeq 142$ m). Le tableau suivant présente les angles ($\alpha$,$\beta$,$\gamma$) et les coordonnées du centre de la caméra pour les 12 sommets de l'ellipsoïde d'incertitude, représentée figure 3.2.b:

| | $\alpha$ (rad) | $\beta$ (rad) | $\gamma$ (rad) | $\mathbf{C}_x$ (m) | $\mathbf{C}_y$ (m) | $\mathbf{C}_z$ (m) |
|---|---|---|---|---|---|---|
| $\mathbf{s}_{-1,1}$ | 3,107 | 0,187 | -1,479 | -2,25 | 192,93 | 0,37 |
| $\mathbf{s}_{+1,1}$ | 3,114 | 0,188 | -1,480 | -2,30 | 192,93 | 0,45 |
| $\mathbf{s}_{-1,2}$ | 3,110 | 0,187 | -1,480 | -2,21 | 192,93 | 0,44 |
| $\mathbf{s}_{+1,2}$ | 3,110 | 0,188 | -1,479 | -2,33 | 192,93 | 0,38 |
| $\mathbf{s}_{-1,3}$ | 3,110 | 0,188 | -1,480 | -2,30 | 192,93 | 0,47 |
| $\mathbf{s}_{+1,3}$ | 3,110 | 0,188 | -1,479 | -2,24 | 192,93 | 0,35 |
| $\mathbf{s}_{-1,4}$ | 3,110 | 0,186 | -1,480 | -2,18 | 192,84 | 0,45 |
| $\mathbf{s}_{+1,4}$ | 3,110 | 0,189 | -1,479 | -2,36 | 193,02 | 0,37 |
| $\mathbf{s}_{-1,5}$ | 3,102 | 0,188 | -1,482 | -2,32 | 192,94 | 0,60 |
| $\mathbf{s}_{+1,5}$ | 3,119 | 0,187 | -1,477 | -2,22 | 192,93 | 0,22 |
| $\mathbf{s}_{-1,6}$ | 3,110 | 0,191 | -1,478 | -2,34 | 191,84 | 0,43 |
| $\mathbf{s}_{+1,6}$ | 3,110 | 0,184 | -1,480 | -2,20 | 194,02 | 0,38 |
| Min | 3,102 | 0,184 | -1,482 | -2,36 | 191,84 | 0,22 |
| Max | 3,119 | 0,191 | -1,477 | -2,18 | 194,02 | 0,60 |
| Max - Min | 0,017 | 0,007 | 0,005 | 0,18 | 2,18 | 0,38 |

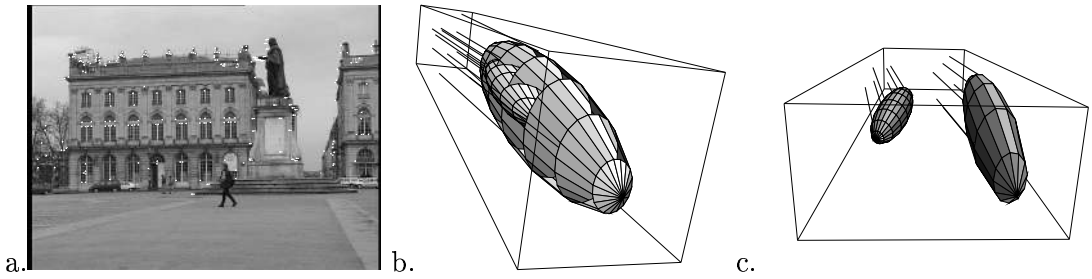


FIG. 3.3 – *a. Appariement de points utilisé par la méthode hybride; b. l'ellipsoïde obtenu par la méthode basée modèle, et plus petit, celui obtenu par la méthode hybride ($\epsilon = 1.0$); c. à gauche, l'ellipsoïde obtenu par la méthode hybride, à droite, celui obtenu par la méthode basée modèle, séparés pour une meilleure comparaison.*

La coordonnée $\mathbf{C}_y$ du centre de la caméra varie beaucoup plus que $\mathbf{C}_x$ et $\mathbf{C}_z$ : l'intervalle de variation de $\mathbf{C}_y$ s'étend sur un peu plus de 2 mètres, alors que cet intervalle pour $\mathbf{C}_x$ et $\mathbf{C}_z$ s'étend sur moins de 20 centimètres et 40 centimètres, respectivement.

### 3.4.5 La méthode hybride réduit l'incertitude

Comme on pouvait s'y attendre, la méthode hybride ($\Phi = \frac{1}{2}(\Phi_{3D2D} + \Phi_{2D2D})$) réduit l'incertitude sur le point de vue, et il n'est également pas surprenant que cette réduction ne soit pas isotrope. Si l'on compare les différences entre les valeurs maximales et minimales de chacun des paramètres obtenus par les deux méthodes:

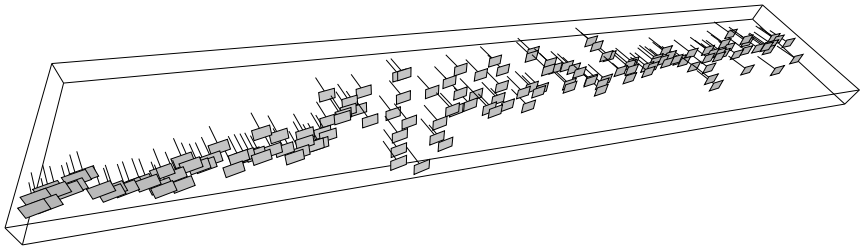
| | $\alpha$ (rad) | $\beta$ (rad) | $\gamma$ (rad) | $\mathbf{C}_x$ (m) | $\mathbf{C}_y$ (m) | $\mathbf{C}_z$ (m) |
|---|---|---|---|---|---|---|
| Max - Min (basée modèle) | 0,017 | 0,007 | 0,005 | 0,18 | 2,18 | 0,38 |
| Max - Min (hybride) | 0,005 | 0,004 | 0,001 | 0,14 | 1,10 | 0,12 |

on se rend compte que l'intervalle possible pour $\mathbf{C}_x$ n'a que peu diminué. La raison en est que le déplacement de la caméra entre l'image précédente et l'image considérée se fait selon une direction à peu près parallèle à l'axe des $x$. Or, les appariements de points 2D (voir figure 3.3.a) n'apportent qu'une contrainte sur la direction du déplacement, mais pas sur sa norme, ce qui explique que la méthode hybride ne permet pas de réduire l'incertitude dans la direction du déplacement de la caméra. La figure 3.3.b permet de comparer visuellement les ellipsoïdes obtenus par les deux méthodes.

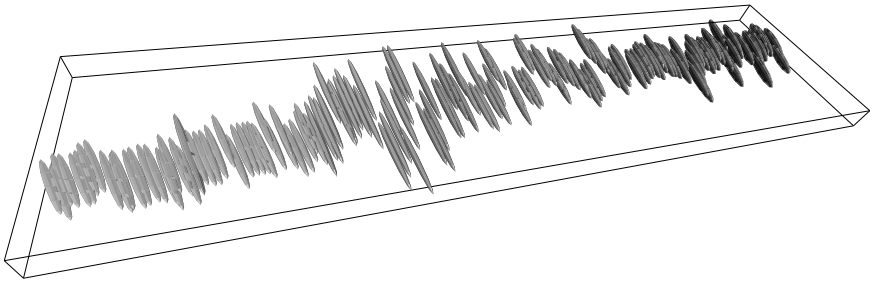
### 3.4.6 Évolution de l'incertitude en fonction de la distance aux primitives 3D

Pour l'instant, nous considérons le choix de la valeur 1.0 pour le paramètre $\epsilon$. Pour la séquence Stanislas, les ellipsoïdes d'incertitude retrouvés ont à peu près tous la même amplitude (figures 3.4.b et 3.5.b). Ceci est dû au fait que la distance entre la caméra et les primitives 3D n'évolue que très peu le long de cette séquence (voir les images figure 2.14).

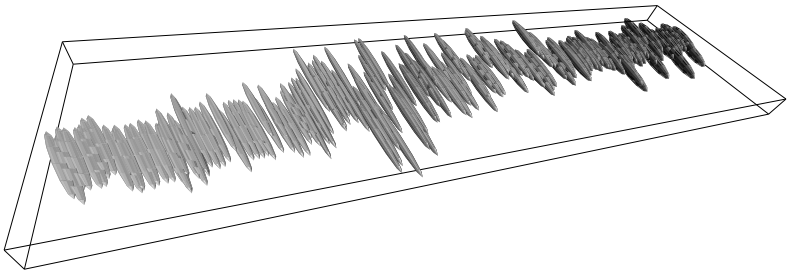
En revanche, pour la séquence du chalet (voir images figure 2.17), la caméra s'éloigne puis se rapproche des primitives 3D. On s'aperçoit sur la figure 3.6.b que l'incertitude augmente avec la distance entre la caméra et les primitives, comme on pouvait s'y attendre.



a.

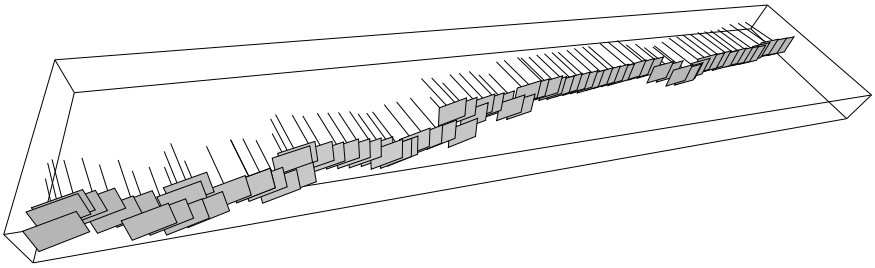


b.

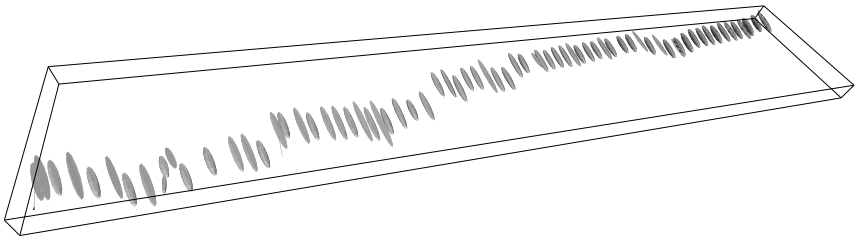


c.

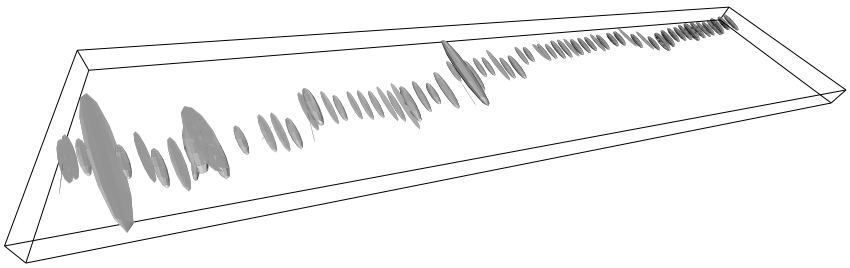
FIG. 3.4 – **Séquence Stanislas:** *a. Points de vue estimés à l'aide de la méthode basée modèle; b. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = 1.0$*; c. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = \Phi^*$*.*



a.

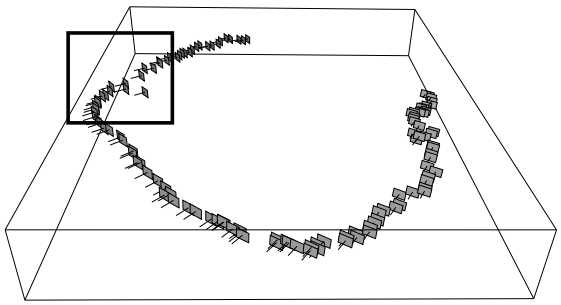


b.

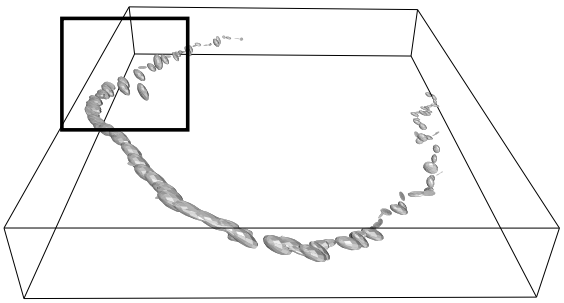


c.

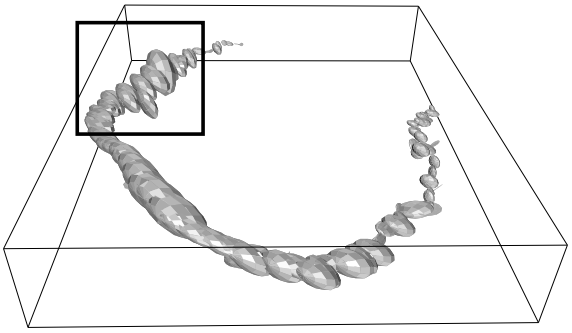
FIG. 3.5 – **Séquence Stanislas:** *a. Points de vue estimés à l'aide de la méthode hybride; b. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = 1.0$*; c. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = \Phi^*$.



a.



b.



c.

FIG. 3.6 – **Séquence du chalet:** *a. Points de vue estimés à l'aide de la méthode hybride; b. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = 1.0$*; c. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = \Phi^*$ *(voir partie 3.4.2 pour une discussion sur le choix de* $\epsilon$*).*

Enfin, on peut remarquer, pour la figure 3.5.c, plusieurs ellipsoïdes d'incertitudes qui ont une taille nettement plus grande que les autres et sont manifestement surestimés. Cette taille anormale est due à des erreurs d'appariements dans les images concernées : le résidu $\Phi^*$ est alors très grand. En prenant $\epsilon = \Phi^*$, les ellipsoïdes d'incertitude correspondants sont alors très grands. L'intérêt du choix $\epsilon = \Phi^*$ apparaît surtout dans la figure 3.6 (voir partie 3.4.2).

## 3.5 Cas de l'ajustement de faisceaux

Rappelons tout d'abord le critère minimisé par l'ajustement de faisceaux:

$$\Phi_{\text{ajust}}(\mathbf{P},\mathbf{M}) = \sum_{i,j} \sigma(i,j).\text{Dist}^2(\text{Proj}_{\mathbf{P}_i}(\mathbf{M}_j),\mathbf{m}_{ij}) \tag{3.15}$$

avec $\sigma(i,j) = 1$ si $\mathbf{M}_j$ est suivi dans l'image $i$, et 0 sinon, et $\mathbf{P}_i$ le point de vue de l'image $i$.

Le paramètre à estimer est alors un vecteur composé de l'ensemble des paramètres externes des projections $\mathbf{P}_i$ et des coordonnées des points $\mathbf{M}_j$:

$$\mathbf{y} = \left((\alpha_1,\beta_1,\gamma_1,t_{x1},t_{y1},t_{z1}) \dots (\alpha_n,\beta_n,\gamma_n,t_{xn},t_{y_n},t_{zn}),(M_{x1},M_{y1},M_{z1}) \dots (M_{xm},M_{y_m},M_{zm})\right).$$

### 3.5.1 Forme du hessien

Le hessien $\mathbf{H}$ a alors la forme suivante :

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_1}}\right)^t & & \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_1}}\right)^t & & \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \\ & \dots & & & \dots & \\ \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & & \frac{\partial}{\partial \mathbf{P_n}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \frac{\partial}{\partial \mathbf{P_n}}\left(\frac{\partial \Phi}{\partial \mathbf{M_1}}\right)^t & & \frac{\partial}{\partial \mathbf{P_n}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \\ \hline \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_1}}\right)^t & & \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_1}}\right)^t & & \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \\ & \dots & & & \dots & \\ \frac{\partial}{\partial \mathbf{M_m}}\left(\frac{\partial \Phi}{\partial \mathbf{P_1}}\right)^t & & \frac{\partial}{\partial \mathbf{M_m}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t & & \frac{\partial}{\partial \mathbf{M_m}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \end{array}\right).$$

Les éléments $\frac{\partial}{\partial \mathbf{P_i}}\left(\frac{\partial \Phi}{\partial \mathbf{M_j}}\right)^t$ et $\frac{\partial}{\partial \mathbf{M_j}}\left(\frac{\partial \Phi}{\partial \mathbf{P_i}}\right)^t$ pour lesquels $\sigma(i,j) = 0$ sont évidemment nuls. Le critère $\Phi_{\text{ajust}}$ est une somme de termes où chacun de ces termes ne faisant intervenir qu'un seul point de vue et un seul point 3D, les éléments de $\mathbf{H}$ $\frac{\partial}{\partial \mathbf{P_k}}\left(\frac{\partial \Phi}{\partial \mathbf{P_l}}\right)^t$ et $\frac{\partial}{\partial \mathbf{M_k}}\left(\frac{\partial \Phi}{\partial \mathbf{M_l}}\right)^t$ sont également nuls pour $k \neq l$. En revanche, les éléments $\frac{\partial}{\partial \mathbf{P_i}}\left(\frac{\partial \Phi}{\partial \mathbf{M_j}}\right)^t$ et $\frac{\partial}{\partial \mathbf{M_j}}\left(\frac{\partial \Phi}{\partial \mathbf{P_i}}\right)^t$ pour lesquels $\sigma(i,j) \neq 0$ ne sont pas nuls. Le hessien $\mathbf{H}$ est finalement de la forme:

$$\left(\begin{array}{ccc|ccc} \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_1}}\right)^t & & \mathbf{0} & \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_1}}\right)^t & & \frac{\partial}{\partial \mathbf{P_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \\ & \ddots & & & \dots & \\ \mathbf{0} & & \frac{\partial}{\partial \mathbf{P_n}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \frac{\partial}{\partial \mathbf{P_n}}\left(\frac{\partial \Phi}{\partial \mathbf{M_1}}\right)^t & & \frac{\partial}{\partial \mathbf{P_n}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \\ \hline \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_1}}\right)^t & & \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \frac{\partial}{\partial \mathbf{M_1}}\left(\frac{\partial \Phi}{\partial \mathbf{M_1}}\right)^t & & \mathbf{0} \\ & \dots & & & \ddots & \\ \frac{\partial}{\partial \mathbf{M_m}}\left(\frac{\partial \Phi}{\partial \mathbf{P_1}}\right)^t & & \frac{\partial}{\partial \mathbf{M_m}}\left(\frac{\partial \Phi}{\partial \mathbf{P_n}}\right)^t & \mathbf{0} & & \frac{\partial}{\partial \mathbf{M_m}}\left(\frac{\partial \Phi}{\partial \mathbf{M_m}}\right)^t \end{array}\right).$$

### 3.5.2 Calcul des matrices de covariance associés aux $\mathbf{P_i}$

Appliquer directement l'approche précédemment exposée ne nous fournit qu'une région de l'espace du paramètre $\mathbf{y}$, région qui contient les points de vue réels $\mathbf{P}_i$ et la position réelle des points $\mathbf{M}_j$, c'est-à-dire une région de l'espace $\mathbb{R}^{6n} \times \mathbb{R}^{3m}$. Or, il est beaucoup plus intéressant (surtout pour l'usage que nous souhaitons en faire, comme nous le verrons au chapitre 6) de connaître, pour chaque point de vue, une région de l'espace des paramètres $(\alpha,\beta,\gamma,t_x,t_y,t_z)$ qui contient le point de vue réel. Les régions d'indifférence associées aux points de vue après ajustement de faisceaux pourraient alors être utilisées comme celles obtenues pour la méthode basée modèle et la méthode hybride.

Or, $\Lambda_{\mathbf{y}}$ a la forme suivante :

$$\Lambda_{\mathbf{y}} = \left(\begin{array}{ccc|ccc} \Lambda_{\mathbf{P_1}} & & \Lambda_{\mathbf{P_1},\mathbf{P_n}} & \Lambda_{\mathbf{P_1},\mathbf{M_1}} & & \Lambda_{\mathbf{P_1},\mathbf{M_m}} \\ & \cdots & & & \cdots & \\ \Lambda_{\mathbf{P_1},\mathbf{P_n}} & & \Lambda_{\mathbf{P_n}} & \Lambda_{\mathbf{P_n},\mathbf{M_1}} & & \Lambda_{\mathbf{P_n},\mathbf{M_m}} \\ \hline \Lambda_{\mathbf{M_1},\mathbf{P_1}} & & \Lambda_{\mathbf{M_1},\mathbf{P_n}} & \Lambda_{\mathbf{M_1}} & & \Lambda_{\mathbf{M_1},\mathbf{M_m}} \\ & \cdots & & & \cdots & \\ \Lambda_{\mathbf{M_m},\mathbf{P_1}} & & \Lambda_{\mathbf{M_m},\mathbf{P_n}} & \Lambda_{\mathbf{M_1},\mathbf{M_m}} & & \Lambda_{\mathbf{M_m}} \end{array}\right).$$

où :

$$\Lambda_{\mathbf{a},\mathbf{b}} = \Lambda^t_{\mathbf{b},\mathbf{a}} = E[(\mathbf{a} - E[\mathbf{a}])(\mathbf{b} - E[\mathbf{b}])^t]$$

On peut donc obtenir les matrices de covariance $\Lambda_{\mathbf{P_i}}$ à un coefficient multiplicatif près en inversant $\mathbf{H}$, puisque $\Lambda_{\mathbf{y}} \sim \mathbf{H}^{-1}$. Malheureusement, la taille de la matrice $\mathbf{H}$ est très importante en pratique et rend son inversion prohibitive. Par exemple, dans le cas de la séquence Stanislas qui comprend 76 images et 2066 points 3D, $\mathbf{H}$ est d'ordre $76 \times 6 + 2066 \times 3 = 6654$. Pour calculer néanmoins les matrices $\Lambda_{\mathbf{P_i}}$, nous reprenons l'astuce de calcul utilisée par [Hartley94]. Écrivons :

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{U}_n \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} \mathbf{V}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{V}_m \end{pmatrix}, \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & & \mathbf{W}_{1m} \\ & \cdots & \\ \mathbf{W}_{n1} & & \mathbf{W}_{nm} \end{pmatrix},$$

$$\text{avec} \quad \mathbf{U}_i = \frac{\partial}{\partial \mathbf{P_i}} \left(\frac{\partial \Phi}{\partial \mathbf{P_i}}\right)^t, \quad \mathbf{V}_j = \frac{\partial}{\partial \mathbf{M_j}} \left(\frac{\partial \Phi}{\partial \mathbf{M_j}}\right)^t \quad \text{et} \quad \mathbf{W}_{ij} = \frac{\partial}{\partial \mathbf{P_i}} \left(\frac{\partial \Phi}{\partial \mathbf{M_j}}\right)^t.$$

$\mathbf{H}$ s'écrit alors :

$$\mathbf{H} = \begin{pmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^t & \mathbf{V} \end{pmatrix}$$

Notons $\mathbf{H}^{-1}$ par :

$$\mathbf{H}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^t & \mathbf{B} \end{pmatrix}$$

On a alors :

$$\begin{pmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^t & \mathbf{V} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^t & \mathbf{B} \end{pmatrix} = \mathbf{I} \tag{3.16}$$

Supposons que $\mathbf{V}$ soit inversible et multiplions l'équation 3.16 à gauche par la matrice :

$$\begin{pmatrix} \mathbf{I} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

qui donne la relation :

$$\begin{pmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^t & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^t & \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

d'où :

$$(\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^t)\mathbf{A} = \mathbf{I}$$

ou :

$$\mathbf{A} = (\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^t)^{-1} = \mathbf{G}^{-1}$$

$\mathbf{G}$ peut être calculé à moindre coût ; on a effectivement (avec $\delta_{ij} = 1$ si $i = j$, 0 sinon) :

$$\mathbf{G}_{ij} = \delta_{ij}\mathbf{U}_i - \sum_k \mathbf{W}_{ik}\mathbf{V}_k^{-1}\mathbf{W}_{jk}^{-1}$$

Calculer $\mathbf{A}$ revient maintenant à inverser $\mathbf{G}$. Le coût d'inversion de la matrice $\mathbf{G}$ est nettement inférieur à celui de la matrice $\mathbf{H}$. Dans le cas de la séquence Stanislas, $\mathbf{G}$ est d'ordre $76 \times 6 = 456$, à comparer avec celui de $\mathbf{H}$, plus de 10 fois plus grand.

On a finalement :

$$\Lambda_{\mathbf{P_i}} \sim \mathbf{A}_{ii},$$

en écrivant la matrice $\mathbf{A}$ par blocs :

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & & \mathbf{A}_{1n} \\ & \cdots & \\ \mathbf{A}_{n1} & & \mathbf{A}_{nn} \end{pmatrix}$$

### 3.5.3 Calcul des ellipsoïdes d'indifférence

Reprenons l'équation 3.11 ; on a alors :

$$(\mathbf{P_i} - E[\mathbf{P_i}])\mathbf{A}_{ii}{}^{-1}(\mathbf{P_i} - E[\mathbf{P_i}])^t \leq K_i'^2 \tag{3.17}$$

Par analogie avec ce qui a été fait précédemment, il nous a semblé naturel de considérer $K_i'^2$ proportionnel à l'erreur de reprojection dans l'image $i$, c'est-à-dire :

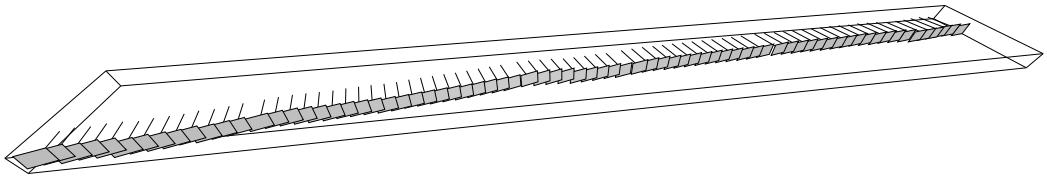
$$\Phi^i_{\text{ajust}}(\mathbf{P}_i) = \sum_j \sigma(i,j).\text{Dist}^2\left(\text{Proj}_{\mathbf{P}_i}(\mathbf{M}_j),\mathbf{m}_{ij}\right)$$

calculé au minimum retrouvé par l'ajustement de faisceaux. Les résultats présentés dans la suite ont donc été obtenus en posant $K_i'^2 = k'_\epsilon \Phi^i_{\text{ajust}}{}^*$, et $k'_\epsilon = 1$. Cette valeur pour $k'_\epsilon$ pourrait être modifiée en fonction des résultats obtenus. Cependant, nous avons constaté qu'elle convient pour nos expérimentations qui utilisent les ellipsoïdes d'incertitude (voir chapitre 7).

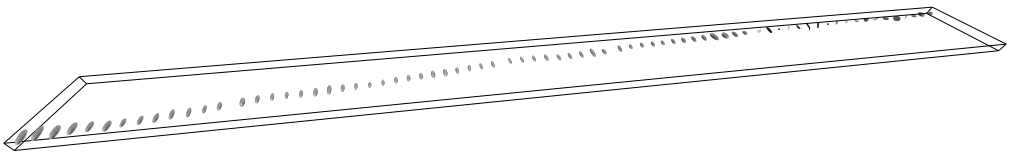
### 3.5.4 Expérimentations

La figure 3.7 montre les régions d'incertitude obtenues pour les points de vue calculés par ajustement de faisceaux. Il s'agit des points de vue obtenus à partir d'un jeu de points d'intérêt sans *outliers*, en optimisant uniquement les paramètres externes (voir partie 2.5.2). La figure 3.8 montre quant à elle les résultats pour la séquence du chalet.

Les conclusions sont identiques pour ces deux séquences: les points de vue obtenus par l'ajustement de faisceaux sont plus précis et il est donc normal que les régions d'indifférence soient



a.



b.

FIG. 3.7 – **Séquence Stanislas:** *a. Points de vue obtenus après ajustement de faisceaux; b. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = {\Phi^i_{ajust}}^*$.

plus réduites que précédemment (on pourra comparer avec les résultats présentés figures 3.5 et 3.6).

On notera toutefois que si les régions sont plus petites, elles suivent la même évolution que dans le cas de la méthode hybride (figure 3.6): les tailles des régions augmentent quand la caméra s'éloigne de la scène. Cela est simplement dû au fait que de nouveaux points d'intérêt n'apparaissent pas quand la caméra recule (figure 3.9).

## 3.6 Conclusion

On vient de présenter tout d'abord plusieurs méthodes statistiques permettant d'évaluer la matrice de covariance d'un résultat d'un algorithme. Cette matrice permet théoriquement de définir une zone autour de ce résultat contenant le résultat attendu, définie par l'équation 3.4. L'inconvénient majeur de ces méthodes statistiques est qu'elles supposent l'indépendance des données utilisées par l'algorithme, hypothèse qui ne peut être retenue dans notre cas.

Au delà de la matrice de covariance, nous recherchons une zone contenant le point de vue réel qui a été estimé à l'aide d'une de nos méthodes. Plutôt qu'une approche statistique, nous avons donc utilisé l'approche des régions d'indifférence, qui permet notamment de se passer de l'hypothèse d'indépendance des données.

Nous avons montré ensuite comment utiliser cette méthode dans le cas de la méthode basée modèle, la méthode hybride et l'ajustement de faisceaux. Nous montrerons comment utiliser ces ellipsoïdes d'incertitude pour estimer l'erreur de reconstruction et de reprojection dans les parties 6.4.1 et 6.4.2. Cette utilisation nous permettra également de valider les zones d'incertitude obtenues.

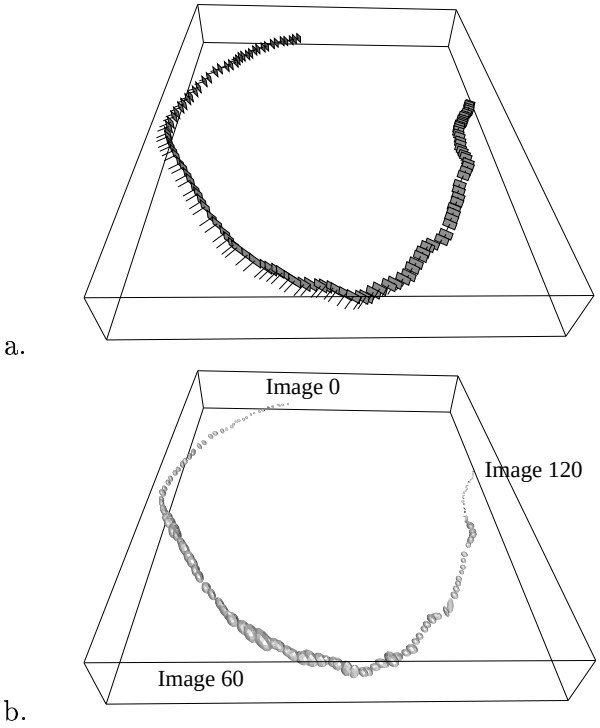



FIG. 3.8 – **Séquence du chalet:** *a. Points de vue obtenus après ajustement de faisceaux; b. ellipsoïdes d'incertitude pour ces points de vue et* $\epsilon = {\Phi^i_{ajust}}^*$.


FIG. 3.9 – **Séquence du chalet:** *Points d'intérêt utilisés par l'ajustement de faisceaux, dans a: l'image 0; b: l'image 60; c: l'image 120.*

# Chapitre 4

# Reconstruction 3D en vision par ordinateur

La reconstruction d'un modèle informatique d'une scène ou d'un objet filmés à partir d'une ou plusieurs caméras est un thème important de recherche en vision par ordinateur. L'enjeu est effectivement de taille puisque cette reconstruction peut être très intéressante pour de nombreuses applications. Nous avons déjà cité son importance pour la Réalité Augmentée (chapitre 1), à laquelle nous pouvons ajouter son utilisation pour la synthèse d'images. Par ailleurs, l'avantage d'une reconstruction par vision est qu'elle ne requiert pas de matériel spécifique (comme un laser, par exemple).

Dans notre cas, l'intérêt d'un tel modèle est qu'il permet théoriquement de résoudre les occultations en Réalité Augmentée. Nous présentons donc ici les nombreuses approches existantes de reconstruction, en gardant à l'esprit notre besoin d'utiliser le modèle acquis pour la gestion de ces occultations, et nos contraintes : nous devons retrouver les parties visibles de l'objet virtuel avec une précision de l'ordre du pixel, en utilisant des points de vue qui peuvent être imprécis. De plus, notre contexte est très général, puisque la caméra se déplace selon une trajectoire quelconque, et filme une scène qui peut être complexe tant sur le plan photométrique que sur le plan topologique. Nous verrons en fait que la reconstruction à proximité des frontières des objets occultants reste difficile, et pourquoi.

## 4.1 Occultations et contours

### 4.1.1 Masque d'occultation et contours occultants

Nous avons présenté dans le chapitre 1 le masque d'occultation, qui est la partie de l'objet virtuel cachée par la scène réelle. En l'absence d'intersection entre cet objet et la scène, la frontière du masque d'occultation est constituée d'une partie de la frontière (évidemment connue) de l'objet virtuel, et de frontières d'objets réels dans l'image (voir la figure 4.1 qui montre l'insertion d'une sphère virtuelle derrière des objets réels).

Retrouver précisément dans l'image les frontières des objets réels (que nous appellerons *contour occultant* dans la suite) est donc essentiel pour obtenir une incrustation de bonne qualité visuelle. Un contour occultant peut être soit la projection d'une arête vive, soit un *contour apparent* (également appelé *contour d'occultation*), c'est-à-dire un contour qui résulte de la projection d'une surface lisse. Les contours occultants ont plusieurs propriétés, qui peuvent être utilisées pour leur détection. Un indice possible est par exemple la présence de jonctions en T le long des

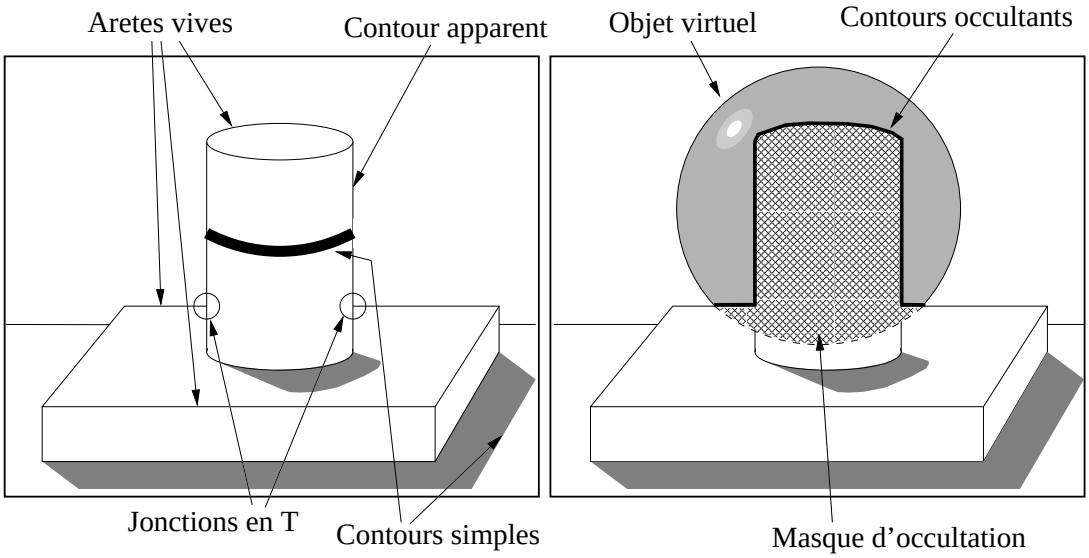



FIG. 4.1 – *Contours et masque d'occultation.*

contours occultants (figure 4.1), qui a été utilisé par [Rothwell95]. En pratique, la détection de telles jonctions est utilisable surtout pour des scènes très contraintes, où les objets réels sont de couleurs différentes et uniformes. Nous allons voir dans cette partie trois propriétés des contours occultants plus étudiées dans la littérature. La première est d'ordre photométrique et est valable surtout pour les arêtes vives; la deuxième est une propriété géométrique des contours apparents; la dernière est liée à la profondeur de la scène.

Enfin, les algorithmes de reconstruction 3D, qu'ils utilisent ou non ces propriétés, devraient permettre de retrouver ces contours. Nous les étudierons dans le reste de ce chapitre.

### 4.1.2 Régions occultées

Un contour occultant, quand il s'agit d'une arête vive, est la frontière d'une région de la scène réelle qui est occultée dans d'autres images (figure 4.2). On peut naturellement envisager d'utiliser cette propriété pour retrouver les contours occultants. Néanmoins, le fait que les points de telles régions soient visibles dans certaines images et pas dans d'autres est souvent source de difficultés pour les algorithmes de reconstruction, notamment ceux qui cherchent à mettre en correspondance des points 2D entre plusieurs images.

De plus, cette propriété n'est plus tout à fait valable pour les contours apparents puisqu'une partie de l'objet occultant n'est alors visible que dans une seule image. Utiliser cette propriété sur un contour apparent ferait que le contour occultant détecté serait délocalisé par rapport à sa position réelle.

Notons que dans le reste de ce chapitre, nous utiliserons le terme d'occultation pour faire référence à ces régions occultées, et non pas à l'occultation d'un objet virtuel par la scène réelle.

### 4.1.3 Contours apparents

Plusieurs travaux [Vaillant et al.92, Kutulakos et al.94, Yi et al.97] ont porté sur la discrimination des contours apparents par rapport aux autres contours présents dans l'image. En effet, les correspondants 3D des contours apparents dépendent du point de vue (figure 4.3.a), et cette propriété permet théoriquement, à partir de trois projections du contour considéré, de savoir

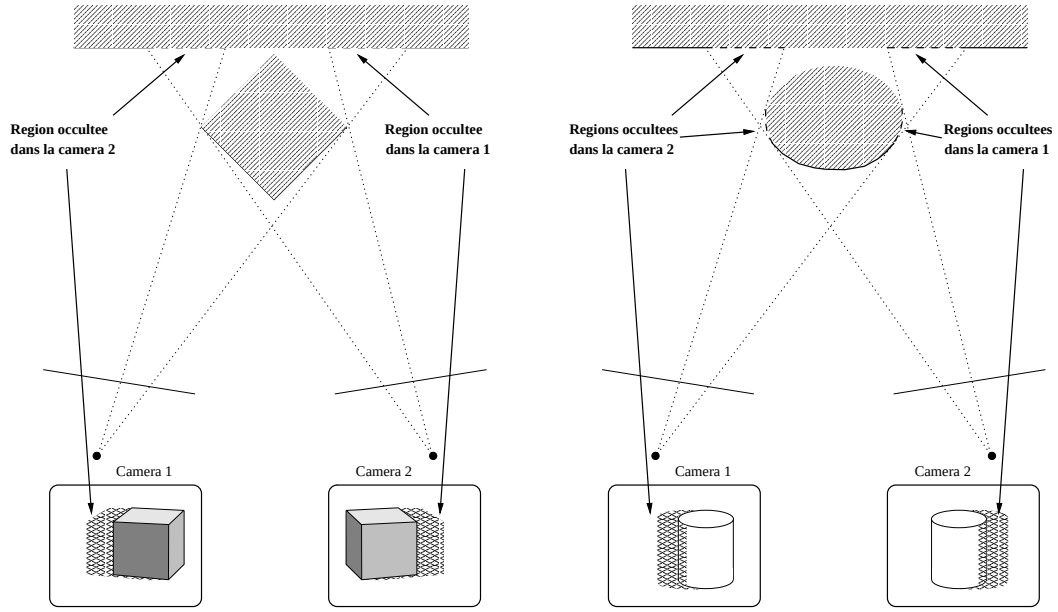



FIG. 4.2 – *Contours occultants et occultations : cas des arêtes vives et des contours apparents.*

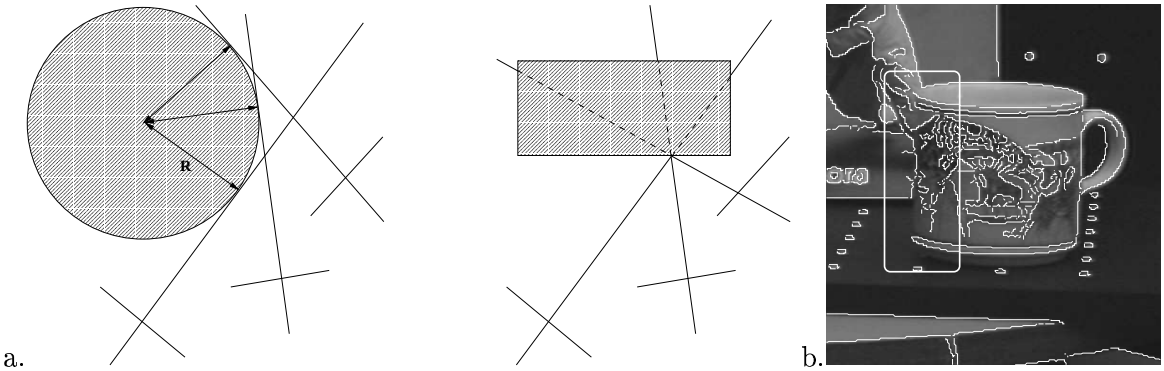



FIG. 4.3 – *a. Discrimination des contours apparents : en pratique, les deux cas sont difficiles à différencier; b. Les contours apparents sont souvent mal retrouvés par les détecteurs de contours.*

s'il s'agit d'un contour apparent ou non : trois projections permettent de calculer le rayon du cercle osculateur aux trois lignes de vue (en supposant que le mouvement de la caméra est plan), rayon qui est non nul dans le cas d'un contour apparent. Mais cette discrimination est difficile à effectuer en pratique : les points de vue doivent être connus très précisément et être dans une position favorable, ce qui fait que cette technique est plutôt utilisée en vision active. De plus, les contours apparents correspondent rarement à une variation de type marche dans le signal image, variation sur laquelle se basent les détecteurs de contours 2D [Canny86, Deriche87] : ils sont souvent au mieux détectés en plusieurs morceaux (voir figure 4.3.b). Enfin, pour notre problème, retrouver les contours apparents ne suffirait pas, puisqu'il nous faut également retrouver les contours occultants qui sont des arêtes vives.

### 4.1.4 Discontinuité de profondeur

Les contours occultants correspondent aussi à une discontinuité de profondeur de la scène. Cette propriété est souvent utilisée conjointement avec le fait qu'un contour occultant soit la frontière d'une région occultée par des algorithmes de reconstruction 3D que nous détaillerons dans la partie 4.3.3.

Avant d'étudier les algorithmes de reconstruction dans le but de détecter les contours occultants, nous allons faire tout d'abord plusieurs considérations sur les points de vue permettant de distinguer les différents contextes des algorithmes de reconstruction.

## 4.2 Points de vue utilisés pour la reconstruction

Dans le cadre très général de la reconstruction en vision par ordinateur, on peut distinguer plusieurs contextes : les images peuvent être acquises par plusieurs caméras fixes, ou par une caméra mobile. Certains algorithmes (dits de *shape from shading*) n'utilisent qu'une seule image, en se basant sur l'ombrage de la scène; ils imposent cependant aux matériaux de la scène d'être diffus et de couleur uniforme, nous n'en parlerons donc pas ici : les scènes qui nous intéressent sont au contraire très générales, comme des milieux extérieurs ou des intérieurs.

### 4.2.1 Précision des points de vue

Selon le contexte, la position et l'orientation des caméras sont connues plus ou moins précisément. Si le point de vue a pu être calculé à l'aide d'une mire de calibration, il est *a priori* connu très précisément, mais l'utilisation d'une mire reste contraignante en pratique. Dans des applications grandeur réelle, la précision des points de vue n'est pas forcément très grande, tester si plusieurs primitives 2D sont la projection d'une même primitive 3D devient alors délicat, et la reconstruction est imprécise.

Dans le cadre d'un algorithme de *structure from motion* (présentés dans le chapitre 2), le problème se pose en termes différents puisque les points de vue et la structure 3D sont retrouvés simultanément : l'appariement des primitives 2D est donc réalisé sans connaissance des matrices de projection. Cependant, ces algorithmes travaillent sur des séquences d'images, et l'appariement s'opère alors sur des images proches, ce qui permet d'obtenir de meilleurs résultats lors de la mise en correspondance.

### 4.2.2 Disposition des points de vue

Supposons que nous ayons trouvé deux points 2D, projections dans deux images (pour lesquelles nous connaissons la calibration de la caméra) d'un point 3D, nous pouvons reconstruire ce point par triangulation. Cependant, comme la position des points 2D n'est pas connue de façon exacte, la précision de la reconstruction dépend de la position relative des points de vue depuis lesquelles sont prises les images. On peut comprendre intuitivement sur la figure 4.4 que cette précision sera moins grande si les rayons entre le point 3D et les centres des caméras sont presque parallèles alors que la configuration optimale consiste en des rayons perpendiculaires (on trouvera une description plus formelle de l'erreur de reconstruction dans [Szeliski et al.96]).

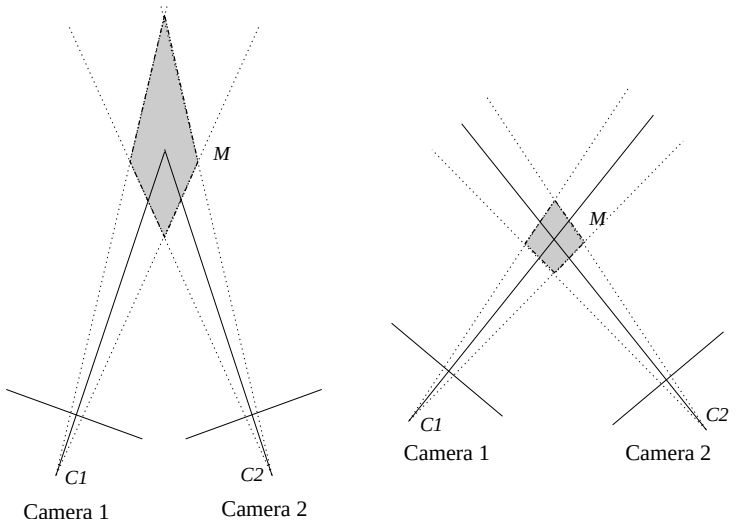



FIG. 4.4 – *Précision d'une reconstruction par triangulation suivant la disposition des caméras.*

### 4.2.3 Nombre de points de vue

Les différentes vues peuvent provenir d'une seule caméra monoculaire en mouvement, ou de plusieurs caméras placées en différentes positions. Si ces deux cas présentent des similitudes, le deuxième permet de considérer des scènes dynamiques, puisque les caméras multiples permettent alors d'acquérir plusieurs images de la scène au même instant, et de reconstruire un modèle pour chaque instant de la séquence (comme c'est le cas par exemple du projet 3D Dome [Kanade et al.99] qui utilise 51 caméras, figure 4.17). Notons également qu'en général, les projets utilisant plusieurs caméras statiques peuvent disposer les caméras suivant une configuration propice à la reconstruction, et que la calibration des caméras est également plus facile puisque les caméras sont fixes.

Une autre solution est d'utiliser une caméra munie de plusieurs objectifs [Kanade et al.95, Kanade et al.96] (voir figure 4.16.a), dont les centres optiques sont suffisamment décalés pour permettre une reconstruction par triangulation. Il suffit alors de connaître la position et l'orientation des objectifs les uns par rapport aux autres; on se retrouve cependant dans le cas peu favorable où les rayons sont presque parallèles (figure 4.4).

Dans le reste de ce chapitre, nous commençons par décrire la stéréoscopie binoculaire, qui a pour but d'obtenir une carte 3D dense d'une scène à partir de deux images. Nous insisterons sur cette approche parce que la majorité des travaux en reconstruction dense ont porté sur celle-ci, mais surtout parce qu'elle va nous permettre d'expliquer pourquoi une reconstruction est rarement précise à proximité des contours occultants. Nous verrons ensuite l'extension de cette méthode à plus de deux images, puis l'utilisation des données partielles fournies par les algorithmes de *structure from motion* pour construire un modèle dense d'une scène. Après ces travaux, qui travaillent essentiellement en appariant plusieurs images puis reconstruisent par triangulation, nous présenterons des travaux qui travaillent eux directement dans l'espace à reconstruire. Finalement, après ces méthodes qui sont toutes automatiques, nous verrons plusieurs produits qui font eux appel à l'utilisateur, ce qui est justifié par le manque de précision et de fiabilité des méthodes automatiques.

## 4.3 Stéréoscopie binoculaire dense

Les premiers algorithmes de reconstruction à avoir été proposés utilisaient deux caméras, décalées horizontalement et dirigées approximativement dans la même direction, par analogie avec la vision humaine [Marr et al.76]. Depuis, de nombreux travaux se sont appuyés sur cette idée et ont proposé de multiples améliorations à l'algorithme originel de Marr et Poggio. Le principe général de ces algorithmes est de retrouver la correspondance de l'ensemble des points entre les deux images, et de reconstruire ces points par triangulation. Ils reposent donc sur la *contrainte d'unicité* énoncée par Marr et Poggio : chaque point a un et un seul correspondant, ou si l'on veut pouvoir prendre en compte les occultations, au plus un correspondant.

### 4.3.1 Corrélation d'intensités

#### Restriction aux objets Lambertiens

La recherche du correspondant s'appuie sur le fait qu'un même point 3D $\mathbf{M}$ doit avoir la même couleur (ou intensité pour des images en niveaux de gris) dans les deux images :

$$I_1(\mathbf{m}_1) = I_2(\mathbf{m}_2) \tag{4.1}$$

où $\mathbf{m_1}$ (respectivement $\mathbf{m_2}$) est la projection de $\mathbf{M}$ dans l'image $I_1$ ($I_2$). Ce n'est pas le cas des points appartenant à des surfaces ayant une composante spéculaire, comme la surfaces des métaux, ou des surfaces partiellement transparentes. On se limite donc à des objets Lambertiens, c'est-à-dire à des objets dont la surface renvoie la lumière de façon isotrope.

#### Fenêtres de corrélation

Il est également très important de tenir compte en pratique des limites des caméras réelles en tant que capteurs. Leur résolution est évidemment finie, et l'image obtenue n'est que le résultat d'un échantillonnage de la projection idéale de la scène. De plus, cette image est généralement bruitée. Supposons maintenant que l'on souhaite comparer deux points 2D dans deux images différentes pour savoir s'ils peuvent être la reprojection d'un même point 3D. À cause des deux limites citées plus haut, on ne peut pas se contenter de comparer leurs couleurs (ou leurs intensités), la comparaison est donc généralement étendue à une fenêtre autour de ces 2 points (figure 4.7.a), et effectuée selon une fonction de corrélation, par exemple :

$$\begin{aligned}&\mathrm{Corr}\,(\mathbf{m}_1,\mathbf{m}_2) = \mathrm{Corr}\,((u_1,v_1),(u_2,v_2)) = \\ &\frac{\displaystyle\sum_{i=-n}^{n}\sum_{j=-m}^{m}\Big[I_1(u_1+i,v_1+j)-\overline{I_1(u_1,v_1)}\Big]\Big[I_2(u_2+i,v_2+j)-\overline{I_2(u_2,v_2)}\Big]}{(2n+1)(2m+1)\sqrt{\sigma^2(I_1)\sigma^2(I_2)}}\end{aligned}$$

où

$$\overline{I_k(u,v)} = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k(u+i,v+j)}{(2n+1)(2m+1)}$$

est la moyenne des intensités autour du point $(u,v)$, et

$$\sigma(I_k) = \sqrt{\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} {I_k}^2(u+i,v+j)}{(2n+1)(2m+1)} - \Big[\overline{I_k(u,v)}\Big]^2}$$

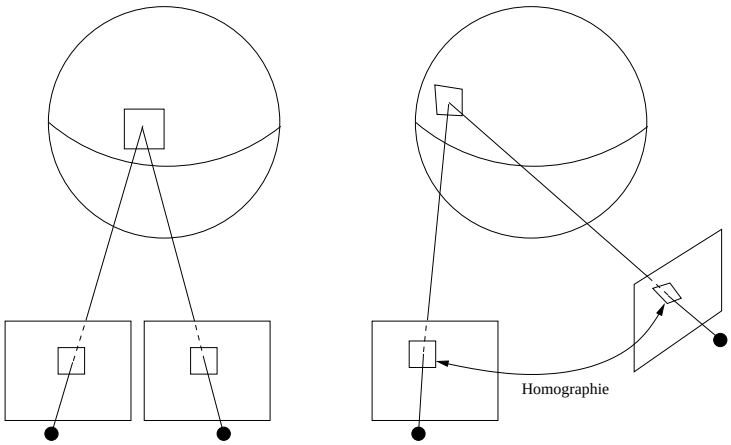



FIG. 4.5 – *A gauche : pour une surface parallèle aux plans images, l'utilisation de fenêtres de corrélation rectangulaires est fondée ; à droite : pour une disposition différente, les fenêtres ne devraient pas être rectangulaires.*

est l'écart-type des intensités autour du point $(u,v)$. Une valeur typique pour $m$ et $n$ est de 7 pixels. Le résultat est compris entre -1 pour des fenêtres de corrélation complètement dissemblables et $+1$ pour des fenêtres identiques. Cette fonction permet de retrouver des correspondants, même sous un changement affine de l'intensité, mais l'intérêt de telles fonctions varie selon le type de scènes : pour des scènes texturées, c'est-à-dire les scènes avec de hautes fréquences du signal image, comme les scènes d'extérieur, la corrélation permet de caractériser correctement les points 2D. Elle est par contre inefficace sur des scènes peu texturées, pour lesquelles les images présentent de grandes zones d'intensité uniforme.

### Tenir compte du plan tangent à la surface

Les fenêtres de corrélation telles qu'elles viennent d'être introduites supposent implicitement que la surface qui se projette dans ces fenêtres soit « en face » et parallèle aux plans images des caméras. Dans le cas contraire, les fenêtres ne correspondent pas à la même partie de la surface, et ceci est d'autant plus marqué que les caméras sont éloignées. On remarquera qu'une disposition permettant une reconstruction précise (partie 4.2.2) n'est pas propice à une bonne corrélation entre les projections d'un même point 3D (le tableau 4.12 répertorie les avantages et les inconvénients des différentes dispositions de points de vue).

Pour compenser cette différence d'apparence, [Devernay et al.94] ont proposé de déformer l'une des fenêtres de corrélation : une petite surface vue comme un carré de pixels dans une image est vue dans l'autre image approximativement selon un carré déformé (voir figure 4.7.b). En effet, si l'on considère que la surface est localement plane, la relation entre les deux fenêtres est théoriquement une homographie, qui est approximée dans cet article par une transformation affine (définie par 6 paramètres, au lieu de 8 pour l'homographie). Comme le plan tangent à la surface n'est pas connu (on recherche justement cette surface), la transformation entre les fenêtres ne peut être calculée directement : les points sont donc d'abord mis grossièrement en correspondance, puis une des fenêtres est transformée selon une transformation affine pour obtenir une meilleure corrélation et une précision subpixel du correspondant. Néanmoins, la mise en correspondance initiale reste obtenue à l'aide de fenêtres classiques, qui peut échouer si les caméras sont trop éloignées. [Faugeras et al.97] utilisent également ces fenêtres déformées, d'une

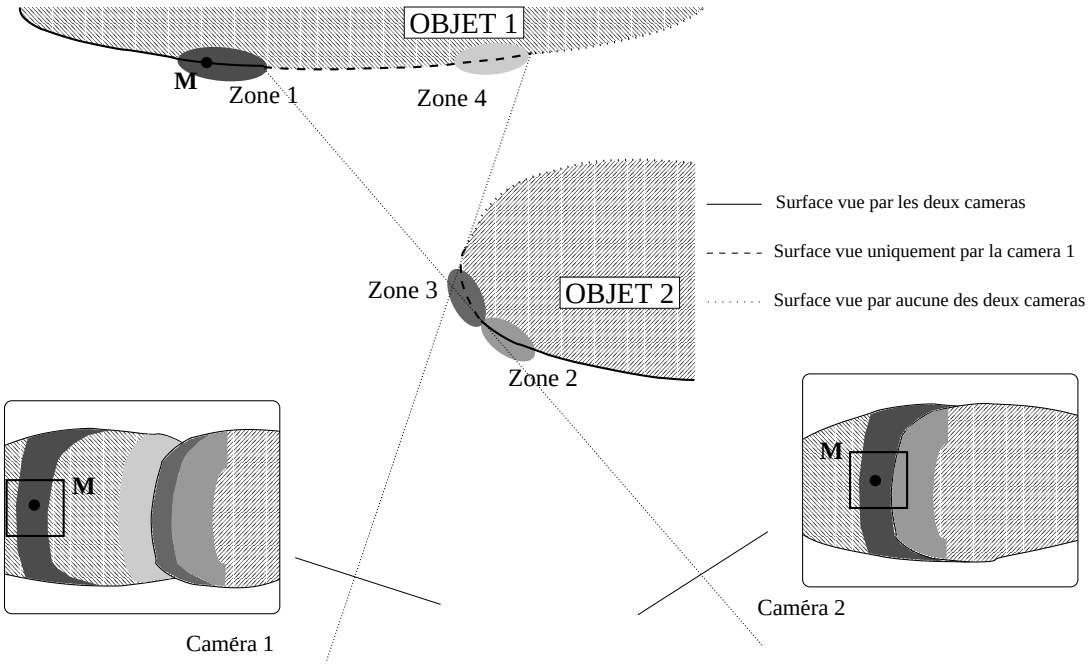



FIG. 4.6 – *Fenêtres de corrélation à proximité d'un contour occultant (Zones 1 et 2 : la fenêtre de corrélation dans la caméra 2 recouvre les objets 1 et 2; zones 3 et 4 : la fenêtre de corrélation dans la caméra 1 recouvre les objets 1 et 2).*

façon un peu différente que nous détaillons plus bas (partie 4.6).

### Tenir compte des occultations des surfaces

A proximité des contours occultants, utiliser des fenêtres de corrélation s'avère délicat. Pour s'en persuader, considérons la figure 4.6 qui représente un objet 1 occulté partiellement par un objet 2. Une fenêtre de corrélation autour de la projection d'un point de la zone 1 définie par cette figure dans l'image de la caméra 1 recouvre une portion de l'objet 1, mais dans l'image de la caméra 2, une telle fenêtre recouvre à la fois l'objet 1 et l'objet 2. La zone 2 présente le même phénomène, en inversant le rôle des images; notons qu'en plus, la surface au voisinage de cette zone est presque perpendiculaire au plan image de la caméra 2, ce qui pose également le problème cité dans le paragraphe précédent. Les fenêtres autour des projections des points des zones 3 et 4 recouvrent également les deux objets dans la caméra 1, mais ces zones ne sont pas visibles par la caméra 2 : nous verrons dans la partie 4.3.3 ce qui a été proposé pour détecter les points d'une image qui n'ont pas de correspondant dans la deuxième image. Nous nous intéressons ici aux techniques évitant à la fenêtre de corrélation de recouvrir deux objets.

[Kanade et al.94] ont développé un algorithme permettant d'ajuster en chacun des points la fenêtre de corrélation. Ils définissent tout d'abord une mesure de la variance d'une fenêtre basée sur la variance de la disparité et de l'intensité. À partir d'une mise en correspondance initiale sur l'ensemble des images, l'algorithme recherche en chaque point une fenêtre adaptée aux conditions locales, en calculant tout d'abord la variance de la fenêtre $3 \times 3$ centrée sur le point considéré. Il étend ensuite la fenêtre en ajoutant des lignes et des colonnes (figure 4.7.c), à condition que l'ajout n'augmente pas la variance de la fenêtre. Une nouvelle mise en correspondance est effectuée en utilisant les fenêtres calculées. Le processus est itéré jusqu'à la convergence, ou pour un certain

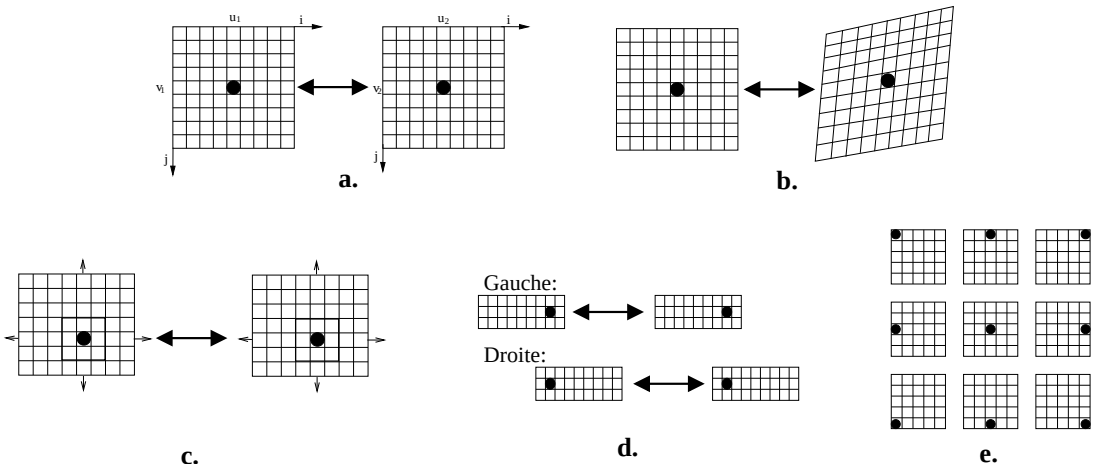



FIG. 4.7 – *a. Fenêtres de corrélation classiques; b. Fenêtres utilisées par [Devernay et al.94]; c. par [Kanade et al.94]; d. par [Geiger et al.95]; e. par [Intille et al.94].*

nombre d'opérations. Ce processus améliore les résultats obtenus avec des fenêtres classiques, sans parvenir à éliminer toutes les erreurs. De plus, il est particulièrement coûteux en temps de calcul.

Une solution plus simple a été proposée par [Geiger et al.95]. Afin d'éviter que, près des occultations, la fenêtre de corrélation ne recouvre à la fois les surfaces occultée et occultante, ils calculent deux corrélations $\text{Corr}_{\text{gauche}}$ et $\text{Corr}_{\text{droite}}$ sur deux fenêtres, l'une débordant sur la gauche des points, l'autre sur la droite (voir figure 4.7.d), la corrélation finale étant $\text{Corr} = \min(\text{Corr}_{\text{gauche}}, \text{Corr}_{\text{droite}})$. Ces fenêtres étant prévues uniquement pour des contours occultants verticaux, [Intille et al.94] utilisent une version plus générale, avec 9 fenêtres de corrélation (figure 4.7.e). Malheureusement, ces auteurs ne comparent pas leurs méthodes avec l'utilisation de fenêtres plus classiques, il est donc difficile de juger de l'intérêt de telles fenêtres.

### Considération du gradient de l'image

[Oisel et al.00] utilise la relation 4.1 d'une autre manière que la corrélation. Ils décomposent le déplacement entre $\mathbf{m}_1$ et $\mathbf{m}_2$ en : $\mathbf{m}_2 = \mathbf{m}_1 + \mathbf{n} + \lambda\mathbf{v}$ où $\mathbf{v}$ est le vecteur directeur de la droite épipolaire associée à $\mathbf{m}_1$ (voir partie 2.2.2). En développant $I_2(\mathbf{m}_2)$ au voisinage de $I_2(\mathbf{m}_1 + \mathbf{n})$, la relation 4.1 devient

$$I_1(\mathbf{m}_1) \simeq I_2(\mathbf{m}_1 + \mathbf{n}) + \lambda\mathbf{v}\nabla I_2(\mathbf{m}_1 + \mathbf{n}),$$

ce qui permet d'utiliser la valeur $\|I_1(\mathbf{m}_1) - I_2(\mathbf{m}_1 + \mathbf{n}) + \lambda\mathbf{v}\nabla I_2(\mathbf{m}_1 + \mathbf{n})\|$ à la place de la fonction de corrélation. On n'évite néanmoins pas tous les problèmes inhérents aux fenêtres de corrélation puisque le gradient $\nabla I_2(\mathbf{m}_1 + \mathbf{n})$ est calculé à partir de l'intensité des pixels sur une fenêtre autour de $\mathbf{m}_1 + \mathbf{n}$.

Le développement de Taylor suppose un faible déplacement entre $\mathbf{m}_1$ et $\mathbf{m}_2$. On peut contourner ce problème en utilisant une approche multi-résolution de mise en correspondance (voir plus bas, partie 4.3.2) pour pouvoir considérer des déplacements importants.

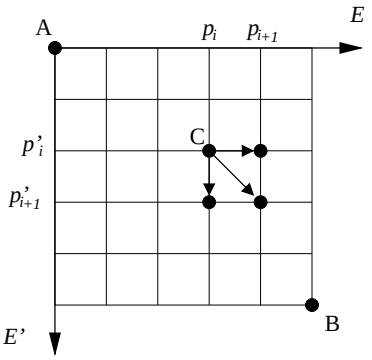



FIG. 4.8 – *Espace de recherche des mises en correspondance.*

### 4.3.2 Réduction des ambiguïtés

#### Contrainte épipolaire

Plutôt que de rechercher le correspondant d'un point **m** d'une image sur l'ensemble de l'autre image, on peut évidemment utiliser la contrainte épipolaire (voir partie 2.2.2) pour limiter la recherche du correspondant à la droite épipolaire associée à **m**. *On notera cependant que cela suppose en pratique une très bonne calibration des caméras.*

En pratique, la recherche est encore réduite, jusqu'à un segment de la droite épipolaire. [Ayache89] montre comment calculer ce segment quand on connaît les dimensions de la scène à reconstruire. On limite ainsi non seulement le temps de recherche mais le nombre de correspondants possibles acceptés par la corrélation d'intensités.

#### Contrainte d'ordre

Les contraintes épipolaire et de corrélation laissent évidemment de nombreuses ambiguïtés de mises en correspondance, notamment pour les régions d'intensité uniforme, ou celles qui présentent des motifs répétitifs. Dans le but de réduire ces ambiguïtés, [Baker et al.81] ont imposé dans leur algorithme la contrainte d'ordre le long des droites épipolaires (également appelée contrainte de monotonie) : les correspondants des points sur une droite épipolaire gardent souvent le même ordre que ces points (ce n'est cependant pas toujours le cas, en particulier quand un objet est très près des caméras par rapport au reste de la scène). Cette contrainte permet en pratique d'améliorer considérablement les résultats.

La mise en correspondance se fait alors entre deux droites épipolaires, par recherche récursive : considérons un point $p_i$ de la droite épipolaire $E$ et un point $p'_j$ de $E'$, conjuguée de $E$. Trois choix se présentent :

1. on apparie $p_i$ et $p'_j$, on considère alors $p_{i+1}$ et $p'_{j+1}$;
2. on n'attribue pas de correspondant à $p_i$, on considère alors l'appariement entre $p_{i+1}$ et $p'_j$;
3. on n'attribue pas de correspondant à $p'_j$, on considère alors l'appariement entre $p_i$ et $p'_{j+1}$.

Chaque choix a un coût, et on recherche l'appariement global entre $E$ et $E'$ de coût total minimal. La programmation dynamique [Vintsyuk68] permet de trouver efficacement cette mise en correspondance optimale. [Baker et al.81] se sont limités à l'appariement de points appartenant à des contours 2D, sans doute pour des raisons de temps de calcul. Depuis, différents coûts ont été proposés dans le cadre de l'appariement dense, comme le score de corrélation pour les appariements, et un coût fixe quand il n'y a pas appariement [Blanc94].

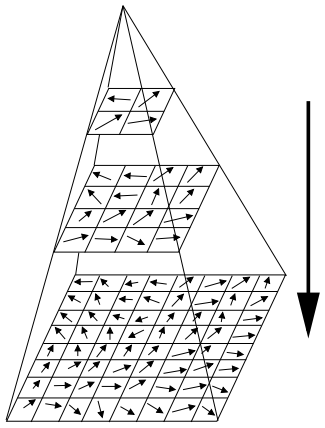


FIG. 4.9 – *Appariement multi-résolution, les flèches représentant les appariements obtenus aux différentes résolutions (figure extraite de [Oisel98]).*

### Cohérence entre les droites épipolaires

Pour l'instant, nous nous sommes limités à considérer la mise en correspondance entre droites épipolaires, sans tenir compte de la cohérence de l'image. [Ohta et al.85] ont étendu l'algorithme de [Baker et al.81] en définissant une relation d'ordre sur les contours (de la gauche vers la droite de l'image), pour étendre la mise en correspondance à l'ensemble de l'image, tout en continuant d'utiliser la programmation dynamique. Cette approche est malheureusement limitée à la mise en correspondance de points de contours.

Dans le cadre de l'appariement dense, [Cox et al.96] ont cherché à utiliser cette cohérence de l'image en minimisant les discontinuités verticales de la profondeur entre deux droites épipolaires consécutives. Ils déterminent tout d'abord une mise en correspondance droites épipolaires par droites épipolaires, puis recherchent une nouvelle fois la mise en correspondance au niveau de chaque droite épipolaire, en minimisant alors un critère qui tient compte de l'appariement obtenu pour les droites épipolaires supérieure et inférieure. Cette technique permet d'améliorer les résultats, mais de façon limitée puisque la cohérence n'est utilisée que localement.

Signalons qu'un simple filtre médian bidimensionnel, appliqué sur la carte de disparité ou de profondeur permet de diminuer le nombre de défauts de la carte [Birchfield et al.98] : un tel filtre supprime le bruit impulsionnel dû aux erreurs d'appariement tout en préservant les discontinuités, qu'il est important de conserver puisqu'elles apparaissent au voisinage des occultations.

Cependant, une bien meilleure approche est sans doute une mise en correspondance multi-résolution [Weng et al.92, Oisel et al.00]. Dans ces travaux, la résolution des images à mettre en correspondance est diminuée, par exemple jusqu'à un niveau $2 \times 2$. Les blocs résultants sont appariés, et cet appariement est utilisé pour initialiser la recherche des correspondants à la résolution supérieure, jusqu'à la résolution réelle des images (voir figure 4.9). On considère alors correctement la cohérence des images. Cette méthode permet également de retrouver la correspondance même pour de grandes valeurs de disparité. La contrainte d'ordre n'est en revanche pas imposée dans ces travaux.

## 4.3.3 Gestion des occultations

Les algorithmes présentés jusqu'ici ne traitent pas explicitement les occultations. Ceux qui attribuent à chaque point un correspondant ne peuvent retrouver correctement les occultations,

mais plusieurs post-traitements ont été proposés pour les détecter à partir des appariements obtenus avec ces algorithmes. Néanmoins, une meilleure approche consiste à tenir compte de la présence éventuelle d'occultations lors de la recherche des correspondants.

### Détection *a posteriori*

[Egnal et al.00] ont répertorié quatre types de post-traitement des cartes de mise en correspondance pour détecter les occultations :

- BMD pour *Bimodality* : Les points à proximité d'un contour occultant sont mis en correspondance avec des points soit de la surface occultante, soit de la surface occultée, créant une distribution bimodale dans la carte de disparité, que les algorithmes BMD cherchent à repérer ([Little et al.90, Wildes91]...)
- MGJ pour *Match Goodness Jumps* : Les algorithmes MGJ supposent que les points qui ont obtenu de mauvais scores de corrélation lors de la mise en correspondance sont en fait des points occultés ([Smitley et al.84]...)
- LRC pour *Left-Right Checking* : Les algorithmes LRC utilisent deux cartes de disparité : une résultant de la mise en correspondance image 1 vers image 2, et l'autre, image 2 vers image 1. Ces cartes sont normalement symétriques, sauf pour les points occultés ([Weng et al.88, Luo et al.95]...)
- ORD pour *Ordering* : Les algorithmes ORD supposent que les points dont la mise en correspondance violent la contrainte d'ordre sont des points qui sont en fait occultés ([Baker et al.81, Yuille et al.84]...)

Les auteurs ont cherché à comparer ces post-traitements empiriquement, sur des images synthétiques, des images réelles de laboratoire et des images de scènes extérieures. Leur conclusion est que LRC est l'algorithme qui donne les meilleurs résultats mais essentiellement pour des scènes suffisamment texturées. ORD donne parfois de meilleurs résultats mais détecte de petites occultations là où il n'y en a pas, en particulier dans les zones faiblement texturées. BMD détecte la plupart des contours occultants, mais sous réserve d'avoir les bons paramètres; en fait, cette méthode est la plus sensible aux paramètres. Enfin, MGJ donne de mauvais résultats quand les régions occultées ou occultantes sont peu texturées.

Cependant, même pour LRC, les résultats restent peu acceptables, surtout pour nos impératifs de qualité. Il vaut mieux tenir compte des occultations dès la mise en correspondance.

### Prise en compte lors de la mise en correspondance

[Geiger et al.95] ont fait remarquer qu'une discontinuité de profondeur (c'est-à-dire un contour occultant) dans une image doit correspondre à une région visible uniquement dans l'autre image (figure 4.2, comme nous l'avons déjà fait remarquer, ceci est surtout valable pour les arêtes vives), qui leur permet d'associer un coût en l'absence d'appariement, proportionnel au nombre de points non appariés. L'appariement a un coût qui est la somme de la corrélation entre les points appariés, et d'un terme permettant d'obtenir une surface reconstruite lisse.

Malheureusement, le coût attribué aux occultations par Geiger et al. n'est pas satisfaisant. En fonction des paramètres choisis, soit il devient trop gros et une large région occultée pourrait être remplacée par des mises en correspondance (erronées), soit il est trop faible et l'algorithme trouve trop d'occultations. On peut remarquer que les travaux présentés jusqu'ici n'utilisaient que des paires d'images avec de petites régions occultées, larges d'une vingtaine de pixels, sauf ceux utilisant une mise en correspondance multi-résolution. [Intille et al.94] ont proposé, pour tenir

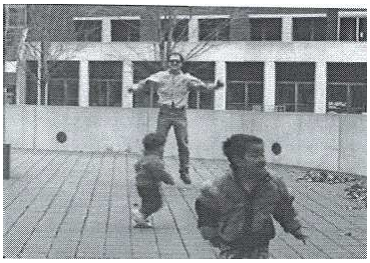
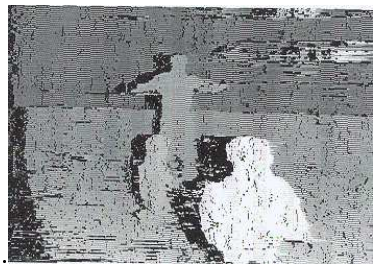
a.  b.  c. 

FIG. 4.10 – *a. et b. Paire stéréoscopique utilisée par [Intille et al.94]; c. Carte de profondeur obtenue (les zones noires sont les occultations détectées).*

a.  b.  c. 

FIG. 4.11 – *a. et b. Paire stéréoscopique utilisée par [Birchfield et al.98]; c. Carte de profondeur obtenue.*

compte de régions plus larges, de forcer le chemin dans le tableau des mises en correspondance à passer par des appariements jugés très sûrs, appelés *Ground Control Points*, qui sont des couples de points dans des zones très texturées qui ont obtenu un bon score de corrélation (voir résultats figure 4.10).

Plutôt que de définir un coût pour les occultations, [Oisel et al.00] adoptent une démarche inverse. Ils définissent alors un critère de régularisation de la surface reconstruite en chaque point $\mathbf{m}$ :

$$H(\mathbf{m}) = \sum_{\mathbf{v} \in \mathcal{V}(\mathbf{m})} ||\text{disparité}(\mathbf{m}) - \text{disparité}(\mathbf{v})||$$

où $\mathcal{V}(\mathbf{m})$ représente les voisins de $\mathbf{m}$. Pour permettre des discontinuités de ce critère, ils introduisent un estimateur robuste $\rho$ (voir partie 2.3.3), et cherchent alors à minimiser, en plus d'un critère de ressemblance d'intensité :

$$H'(\mathbf{m}) = \rho \left( \sum_{\mathbf{v} \in \mathcal{V}(\mathbf{m})} ||\text{disparité}(\mathbf{m}) - \text{disparité}(\mathbf{v})|| \right)$$

Enfin, [Birchfield et al.98] ont proposé d'imposer aux régions occultées de commencer ou de finir sur une importante variation de l'intensité de l'image. Cette variation correspond alors au contour occultant, à condition que la couleur de l'objet soit suffisamment différente de la zone occultée. Dans le cas contraire, l'algorithme se base sur une autre variation d'intensité et la reconstruction est alors très mauvaise (voir résultats 4.11). Signalons de plus que [Birchfield et al.98] ne montrent pas de résultats sur des scènes très texturés, où les variations d'intensité ont peu de chance de correspondre à des contours d'objets.

| **Points de vue proches :** | **Points de vue éloignés :** |
|---|---|
| ⊕ Mise en correspondance facilitée :<br>– Bonne corrélation entre les projections d'un même point 3D ;<br>– Intervalle de recherche de disparité réduit ;<br>– Régions occultées petites. | ⊖ Mise en correspondance plus difficile :<br>– Risque de mauvaise corrélation entre les projections d'un même point 3D;<br>– Intervalle de recherche de disparité large;<br>– Régions occultées larges. |
| ⊖ Reconstruction incertaine | ⊕ Reconstruction plus précise |

FIG. 4.12 – *Avantages ⊕ et inconvénients ⊖ de l'utilisation de points de vue proches et éloignés pour la stéréoscopie binoculaire*

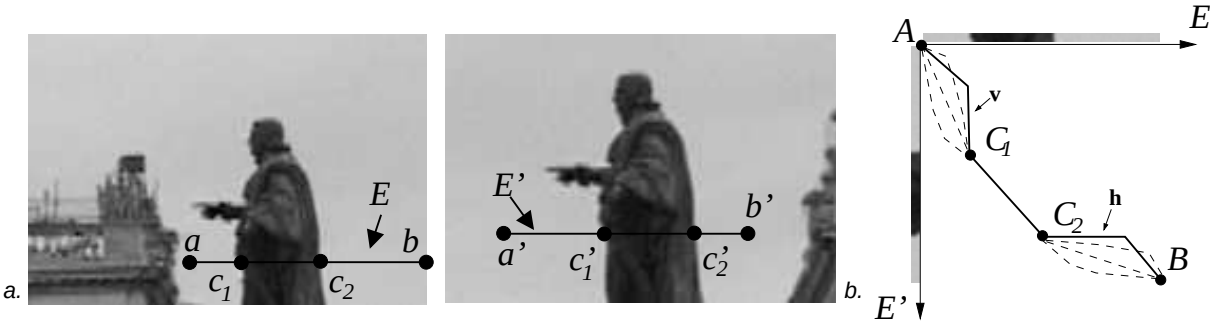



FIG. 4.13 – *a. Paire d'images stéréoscopiques; b. en trait plein, la mise en correspondance entre E et E′, en trait pointillé des mises en correspondance possibles.*

### 4.3.4 Discussion

En dépit de nombreux travaux, la stéréoscopie binoculaire n'est souvent pas suffisante pour permettre une reconstruction précise au voisinage des occultations dans le cas général, pour différentes raisons :

- on a déjà signalé qu'une région d'intensité uniforme pose des difficultés de reconstruction dues aux ambiguïtés d'appariement, qui peuvent être partiellement résolues en utilisant la contrainte d'ordre. Il subsiste cependant des ambiguïtés en particulier quand cette région est partiellement occultée, comme le ciel de la figure 4.13.a. Dans cet exemple, on peut légitimement supposer que les points $c_1$ et $c_2$ sur le contour de la statue peuvent être aisément mis en correspondance. Par contre, l'appariement des régions $[a; c_1]$ et $[c_2; b]$ est beaucoup plus difficile : la mise en correspondance correcte (en trait plein sur la figure 4.13.b) n'a en fait aucune raison d'être retrouvée, notamment les parties **v** et **h**, qui correspondent aux régions occultées.
- quand un objet est placé devant une surface de même apparence, la transition est peu marquée (cf figure 4.14) : une partie de la surface risque d'être reconstruite comme appartenant à l'objet ou inversement (c'est le cas de la reconstruction figure 4.11.c). On pourrait réduire ce phénomène en imposant une contrainte de régularité de la frontière des objets, mais cela va à l'encontre de notre souhait d'un détourage précis. Une telle régularisation risquerait en effet d'éliminer certains détails de la frontière des objets.
- si la surface placée derrière un objet présente un motif répétitif, l'objet, en occultant un ou

FIG. 4.14 – *Absence de variation d'intensités à la frontière d'un objet.*


FIG. 4.15 – *Disparition d'un motif répétitif entre deux images stéréoscopiques.*

plusieurs motifs dans l'une des images, risque fort de perturber la mise en correspondance (cf figure 4.15.b).

- les paramètres que nécessite une contrainte de régularité des surfaces reconstruites restent relativement arbitraires. De plus, cette contrainte n'est pas valable aux frontières des objets puisqu'elles correspondent à des discontinuités de profondeur. Imposer une continuité « presque partout » reste délicat.

Les points que nous venons de soulever sont directement liés aux occultations et ne sont pas du tout exceptionnels : une séquence prise en extérieur a toutes les chances de réunir ces difficultés. La stéréoscopie binoculaire ne permet donc pas d'obtenir les résultats précis dont nous avons besoin.

## 4.4 Extension de la stéréoscopie binoculaire à $n$ images

Pour réduire les ambiguïtés d'appariement, certains travaux utilisent plus de deux images, mais se limitent à chercher une carte de profondeur pour une image de référence. D'autres cherchent à intégrer l'ensemble des informations fournies par les images, pour construire un modèle 3D plus complet de la scène, tout en restant dans le cadre de la reconstruction stéréoscopique.

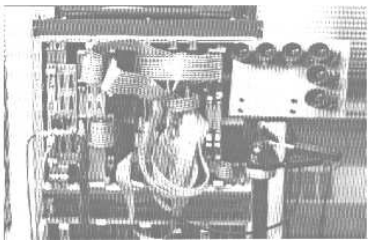
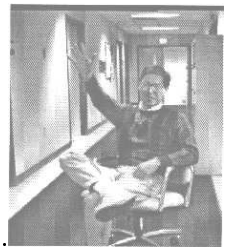
a.  b.  c. 

FIG. 4.16 – *a. Dispositif à 6 caméras de [Okutomi et al.93]; b. Scène vue par la caméra de référence; la carte de profondeur obtenue.*

### 4.4.1 Stéréoscopie « multi baseline »

En vue d'obtenir un système temps réel [Okutomi et al.93] ont développé un système utilisant plusieurs caméras alignées, comme celui présenté figure 4.16.a. Nous sommes donc ici dans le cas d'une faible distance entre les points de vue. Les caméras multiples permettent de réduire les ambiguïtés : le choix d'un correspondant dans une image pour un point de l'image de référence donne l'ensemble des reprojections du point 3D dans toutes les autres images. Le correspondant qui fournit la meilleur corrélation entre toutes les reprojections est retenu. Ce principe risque donc de donner un mauvais appariement si le point 3D est occulté dans au moins une des images. Ce système permet d'obtenir une carte de profondeur dense de $200 \times 200$ pixels, à une fréquence de 30 images par secondes. Leurs résultats (figure 4.16.c) présentent des artefacts à proximité des contours occultants, mais ont le mérite d'être obtenus en temps réel.

### 4.4.2 Fusion de cartes 3D denses

Un moyen simple d'utiliser plusieurs images serait de reconstruire des cartes 3D à partir de plusieurs paires d'images, et de fusionner ces cartes. Mais la plupart des algorithmes de fusion de cartes 3D travaillent en général sur des cartes de profondeur obtenues par laser, et donc bien plus précises que celles obtenues par stéréoscopie binoculaire. [Narayanan et al.98] ont pu néanmoins utiliser un tel algorithme, celui de [Curless et al.96], qui présente la particularité de tenir compte de l'incertitude des reconstructions et de la réduire au moment de la fusion en utilisant la redondance des observations. Le volume à reconstruire est d'abord divisé en voxels (petits éléments de volume cubiques, disposés régulièrement); à chaque fois qu'une nouvelle carte 3D est considérée, la distance de chaque voxel à cette surface est calculée et accumulée. Quand toutes les cartes ont été fusionnées, on suppose que les valeurs contenues par les voxels sont celles d'une fonction $f(x,y,z)$, et que $f(x,y,z) = 0$ définit la surface reconstruite (extraite à l'aide de l'algorithme *marching cubes*). Narayanan et al. ont mis en œuvre cette fusion dans le cadre du « 3D Dome », constitué par 51 caméras sur un dôme de 5 mètres de diamètre, chaque caméra étant dirigée vers le centre du dôme (figure 4.17.a). Les cartes 3D sont obtenues à partir de 3 à 6 caméras, en utilisant l'algorithme « multi baseline » présenté dans le paragraphe précédent. Malgré le nombre de caméras impliquées, les mêmes artefacts restent présents (figure 4.17.b) près des contours occultants.

### 4.4.3 Chaînage de mises en correspondance binoculaires

Pour reconstruire une scène à partir d'une séquence, [Koch et al.98] commencent par mettre en correspondance chaque paire composée de deux images consécutives de la séquence (ils utilisent l'algorithme de [Cox et al.96]). Ensuite, les correspondants sont chaînés : chaque chaîne est un

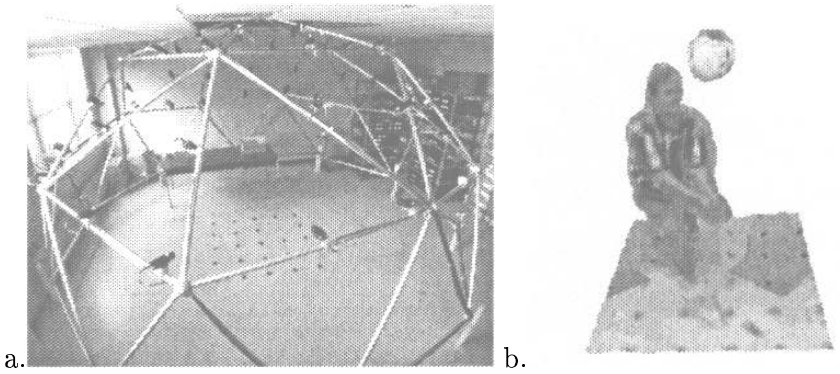


FIG. 4.17 – *a. Dome 3D; b. un exemple de reconstruction obtenue (figures tirées de [Narayanan et al.98]).*

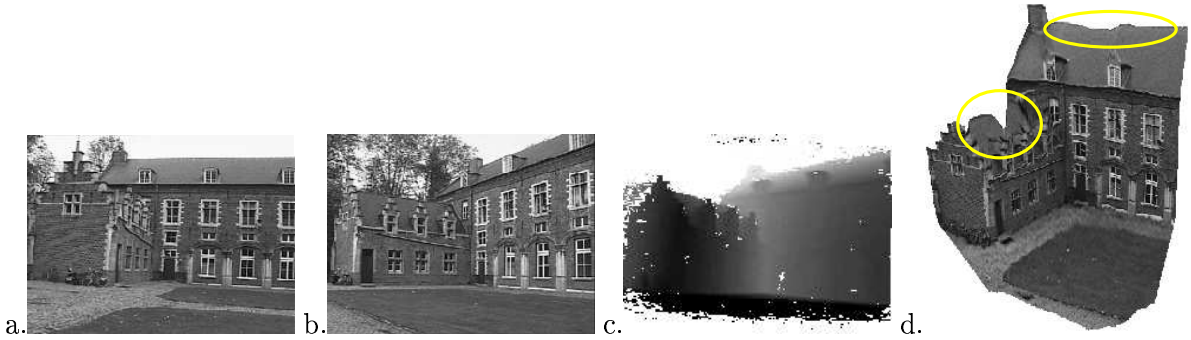


FIG. 4.18 – *a. et b. Première et dernière image de la séquence utilisée par [Koch et al.98]; c. carte de profondeur obtenue; d. modèle reconstruit texturé.*

ensemble de projections d'un point 3D, qui est reconstruit à partir de ces projections à l'aide d'un filtre de Kalman. Afin d'éviter la présence d'erreurs d'appariement, le chaînage est interrompu quand on cherche à intégrer un point 2D qui n'est pas compatible avec les précédentes projections. Cette approche réunit donc les avantages de l'utilisation de points de vue proches (pour effectuer la mise en correspondance) et de points de vue éloignés (au moment de la reconstruction). De fait, les résultats sont assez satisfaisants : le sol et les murs du bâtiment de la séquence présentée figure 4.18 sont convenablement reconstruits; on notera cependant que certains contours occultants (notamment près des fenêtres du toit) restent mal retrouvés. On retombe ici dans une moindre mesure sur les mêmes problèmes que la stéréoscopie binoculaire.

### 4.4.4 Discussion

L'utilisation de plusieurs images devrait permettre de réduire l'ambiguïté des appariements. Cependant, soit les travaux négligent le problème des occultations, soit ils chaînent les mises en correspondance entre deux images. Aucun travail n'a cherché à mettre en correspondance les points sur l'ensemble des images disponibles tout en tenant compte des occultations, sans doute à cause du nombre d'hypothèses d'appariement à considérer. De fait, les résultats ne sont pas encore convaincants.

Nous allons voir maintenant ce qui a été proposé pour obtenir une reconstruction dense mais en s'appuyant sur un appariement partiel mais fiable, obtenu sur un ensemble d'images.

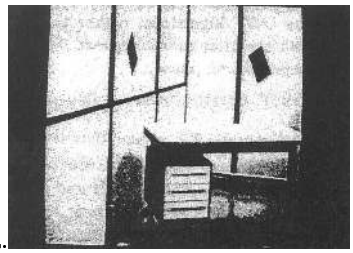
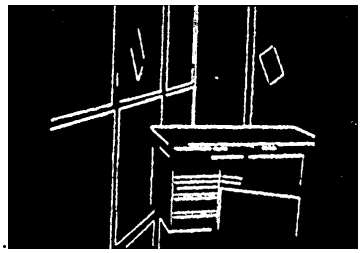
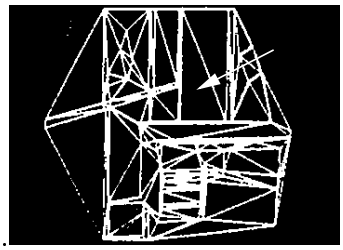
a.  b.  c. 

FIG. 4.19 – *a. Scène à reconstruire; b. Segments détectés et reconstruits; c. Résultat d'une triangulation de Delaunay contrainte sur ces segments (figures tirées de [Bruzzone et al.92]).*

## 4.5 Reconstruction par appariement de primitives 2D

Plutôt que de rechercher un correspondant pour l'ensemble des points d'une image, une autre approche consiste à se limiter à des primitives 2D telles que des points d'intérêt ou des segments 2D. Ces primitives présentent en effet l'intérêt de pouvoir être mises en correspondance de façon robuste, permettant d'éviter les erreurs de reconstruction.

Les algorithmes de *structure from motion* utilisent déjà des appariements de telles primitives, pour reconstruire leurs correspondants 3D et calculer simultanément le mouvement de la caméra. Nous avons déjà présenté la mise en correspondance de points d'intérêt le long d'une séquence d'images dans le chapitre 2. La mise en correspondance de segments de droite s'opère de manière équivalente, en se basant sur un critère de corrélation le long des segments et le tenseur trifocal [Schmid et al.97], ou l'orientation des segments 2D [Ayache88]. L'utilisation de segments 2D est néanmoins plus délicate que les points d'intérêt. Par exemple, le détecteur de segments peut retrouver, pour un même segment 3D, un long segment 2D dans une image, et deux petits segments dans l'image suivante, compliquant ainsi la mise en correspondance. Néanmoins, les segments 2D apportent une information plus dense que les points d'intérêt.

Le problème est justement ici la densification de la reconstruction. Les points ou les segments ne donnent effectivement qu'une information partielle, ne permettant notamment pas de déterminer les occultations entre la scène réelle et les objets virtuels. Nous présentons donc les méthodes permettant de densifier un ensemble de points ou de segments 3D. Elles reposent essentiellement sur une triangulation 2D ou 3D des éléments déjà reconstruits. Elles sont donc limitées à des scènes planes par morceaux. De plus, les parties pertinentes de la scène doivent être présentes parmi les points et les segments reconstruits. Dans le cas contraire, le maillage obtenu se révèle évidemment très imprécis.

### 4.5.1 Densification par triangulation

La plupart des travaux de *structure from motion* qui présentent une reconstruction dense utilisent une triangulation de Delaunay 2D [Watson81]. Par exemple, [Bruzzone et al.92] reconstruisent des segments, et la triangulation de Delaunay contrainte [Chew87] pour que les segments 2D correspondent à des arêtes du maillage (figure 4.19). Évidemment, des problèmes subsistent, comme le triangle marqué par une flèche, dont une des arêtes est un contour occultant : ce triangle ne correspond pas à une partie de la scène. On remarque ici que la triangulation 2D ne permet pas à elle seule de préserver les discontinuités de profondeur essentielles pour la détermination des occultations. De plus, elle impose que les primitives (segments ou points) soient présentes dans une même image. Effectuer le maillage sur les primitives 3D permet d'éviter cette contrainte.

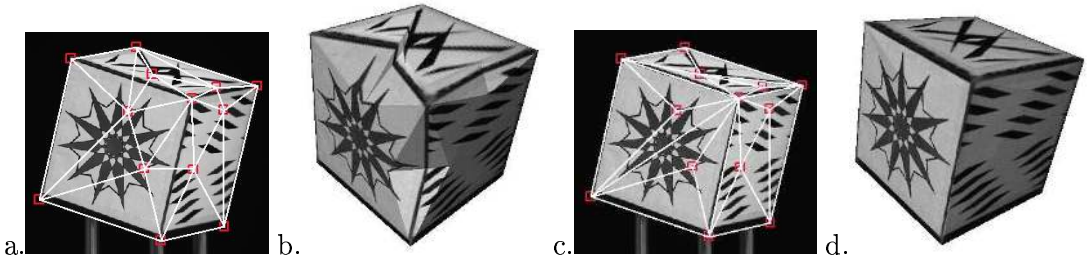



FIG. 4.20 – *a. Maillage initial et modèle résultant; c. et d. Maillage obtenu par [Morris et al.00] et modèle résultant (figures tirées de l'article).*

En synthèse d'images, de nombreux travaux ont porté sur l'obtention d'une surface triangulée à partir d'un ensemble de points 3D, en cherchant notamment à obtenir une surface régulière. Ces algorithmes obtiennent généralement de bons résultats pour des ensembles de points suffisamment denses, ce qui n'est généralement pas le cas des points obtenus en vision à partir d'une séquence d'images, et la triangulation obtenue ne modélise pas la surface à reconstruire.

Pour éviter d'obtenir une reconstruction arbitraire, [Morris et al.00] ont proposé d'utiliser les images de la séquence lors de la triangulation, pour vérifier que les triangles créés correspondent effectivement à des parties planes de la scène à reconstruire. Ils projettent les images de la séquence sur une triangulation initiale, en moyennant les intensités quand plusieurs images se projettent sur une même facette, texturant ainsi le maillage. Ils projettent ensuite le maillage et sa texture selon les points de vue de la séquence, pour obtenir une prédiction $\{\hat{I}_i\}$ des images $\{I_i\}$ de la séquence. Il s'agit alors de minimiser :

$$\Sigma = \sum_i \sum_{\mathbf{x} \in \hat{I}_i} \left[ I_i(\mathbf{x}) - \hat{I}_i(\mathbf{x}) \right]^2$$

Pour cela, ils opèrent la modification locale du maillage qui permet la plus grande diminution du critère $\Sigma$. Le processus est alors itéré jusqu'à ce qu'on ne puisse plus trouver de modification qui améliore le critère $\Sigma$. Un exemple de résultat est visible figure 4.20.c où notamment les arêtes vives sont correctement retrouvées, la figure 4.20.a montrant la triangulation initiale.

### 4.5.2 Discussion

On peut regretter que Morris et al. ne montrent pas d'exemples de scènes d'une topologie plus complexe que celle présentée figure 4.20, puisque leur algorithme peut *a priori* tenir compte en particulier des occultations. Un autre regret est qu'ils n'utilisent pas de segments (en plus des points) pour contraindre la triangulation. Enfin, rappelons que les parties pertinentes de la scène (comme les sommets du cube de l'exemple) doivent appartenir aux primitives reconstruites pour éviter un résultat trop dégradé. En la présence d'occultations, la densification de données partielles reste donc non résolue, même pour une scène plane par morceaux.

Nous venons de voir l'approche consistant essentiellement à apparier les images disponibles pour reconstruire la scène par triangulation. Nous allons décrire maintenant des travaux qui suivent la démarche inverse, puisqu'ils considèrent en premier lieu l'espace à reconstruire.

## 4.6 Reconstruction par « méthode d'ensemble de niveaux » (*level set method*)

### 4.6.1 Présentation

[Faugeras et al.97] ont proposé, pour reconstruire une scène à partir de plusieurs vues, de rechercher directement la surface réelle $\mathcal{S}$. Ils cherchent donc à minimiser le critère, en paramétrant cette surface par $S(v,w)$ :

$$\iint_{\mathcal{S}} \sum_{i,j=1;i\neq j}^{\text{nombre d'images}} \text{Corr}\,(S(v,w),i,j)\,|S_v \times S_w| dvdw, \tag{4.2}$$

Corr $(S(v,w),i,j)$ étant une fonction calculant la corrélation entre les projections du point $S(v,w)$ dans les images $i$ et $j$; le terme $|S_v \times S_w|$ permet lui de régulariser la surface obtenue. De plus, au moment de calculer le critère, les auteurs tiennent compte de l'apparence locale de $\mathcal{S}$ lors de la corrélation en utilisant des fenêtres du même type que [Devernay et al.94] (voir figure 4.7.b). De même, la corrélation est effectuée uniquement pour les images où le point $S(v,w)$ est visible, ce qui permet de prendre en compte les occultations.

Pour rechercher la surface qui réalise le minimum, Faugeras et al. s'inspirent de la méthode des *level sets* [Sethian96] (littéralement ensembles de niveaux), qui a déjà trouvée de multiples applications (en physique des fluides, en robotique, en imagerie médicale...). L'idée de base en est d'ajouter une dimension (qui représente le temps $t$) au problème : ici, $\mathcal{S}$ sera en fait vue comme l'intersection d'une surface de $\mathbb{R}^4$ et de l'espace $\mathbb{R}^3$, l'intérêt étant de permettre des topologies complexes pour cette surface, en particulier plusieurs objets non connectés.

### 4.6.2 Implantation

Détaillons l'implantation d'une telle méthode :

1. **Discrétisation :** On choisit un volume qui englobe la scène à reconstruire, qui est ensuite discrétisé selon un maillage $M_{i,j,k}$; le temps est également discrétisé selon un pas $\delta t$ : on va construire une suite de surface $\mathcal{S}^t$ indexée par le temps et qui converge vers la surface à reconstruire.
2. **Initialisation :** $\mathcal{S}$ est initialisée par une surface initiale $\mathcal{S}^{t=0}$ englobant la surface à reconstruire. À chaque point $M_{i,j,k}$ on affecte la distance algébrique de ce point à $\mathcal{S}^{t=0}$ (positive à l'extérieur, négative à l'intérieur) : $\Phi^{t=0}_{i,j,k} = d(M_{i,j,k},\mathcal{S}^{t=0})$; $\Phi^{t=0} = 0$ définit donc $\mathcal{S}^{t=0}$.
3. **Itération :** À chaque pas de temps, la valeur de $\Phi$ évolue suivant le schéma numérique :

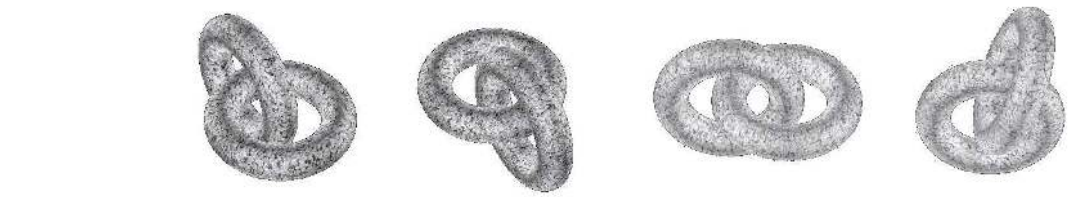
   $$\Phi^{t+\delta t}_{i,j,k} = \Phi^t_{i,j,k} - \beta^t_{i,j,k} |\nabla_{i,j,k} \Phi^t_{i,j,k}|$$

   où $\beta$ est une fonction calculée à partir du critère 4.2, et dépendant notamment des dérivées secondes de $\Phi$. $\mathcal{S}^t$ est définie par la surface de niveau 0 de $\Phi^t$, et peut être extraite grâce à l'algorithme des *marching cubes*.

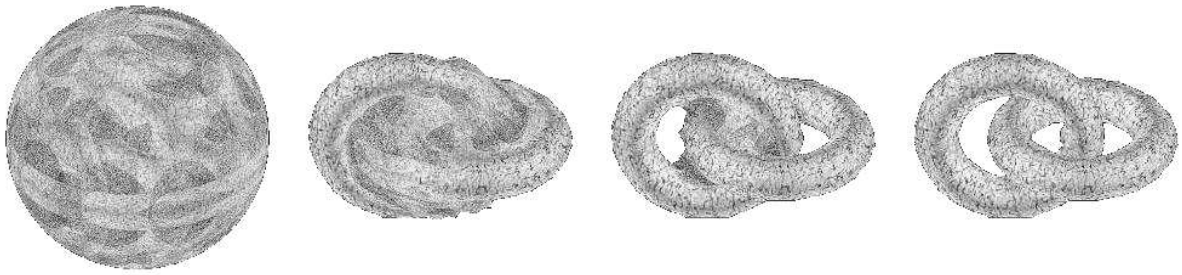
### 4.6.3 Discussion

Un exemple de reconstruction est présenté figure 4.21. Cette approche est très intéressante, puisqu'elle permet :

- de reconstruire des surfaces de topologie complexe tout en respectant une certaine régularité;



a.



b.

FIG. 4.21 – *a. 4 images sur les 20 utilisés pour reconstruire deux tores imbriqués; b. Plusieurs étapes de la convergence de la surface $\mathcal{S}$.*

- de prendre en compte plusieurs images, sans privilégier une image de référence;
- de tenir compte dans une certaine mesure des occultations et de l'orientation des surfaces au moment du calcul de la corrélation, en se basant sur la surface courante.

Un inconvénient est que la surface à reconstruire doit être $\mathcal{C}^2$, ce qui empêche de reconstruire correctement notamment des arêtes vives. De plus, la plupart des exemples de reconstruction présentés ne sont pas des cas pratiques : ils sont obtenus à partir d'images synthétiques très texturées et non bruitées, permettant à la corrélation d'être très discriminante; on peut se demander comment se comporte l'algorithme en présence de zones d'intensité uniforme. De plus, l'utilisation d'images de synthèse permet de disposer de points de vue connus avec une précision dont on ne dispose pas forcément en pratique.

## 4.7 Reconstruction par « découpage de l'espace » (*space carving*)

### 4.7.1 Présentation

Le *space carving* (littéralement découpage ou creusage d'espace) a été introduit par [Seitz et al.97] et prolongé par [Kutulakos et al.98] pour reconstruire une scène à partir d'un ensemble d'images calibrées. Comme la méthode par ensemble de niveaux, il ne cherche plus à mettre en correspondance les images en vue de la reconstruction, mais travaille au contraire directement dans l'espace à reconstruire. Le principe en est très simple : intuitivement, on part d'un volume plein englobant la scène, qui est ensuite « creusé » jusqu'à s'approcher de cette scène. Plus précisément (voir également la figure 4.22) :

1. **Initialisation :** on ne conserve qu'un volume $\nu$ de l'espace, qui doit englober la scène à reconstruire, et qui est discrétisé en voxels.
2. **Pour chaque voxel $v$ à la surface de $\nu$ :**
   (a) $v$ est projeté dans toutes les images où il n'est pas occulté par d'autres voxels;
   (b) si les projections de $v$ ne sont pas toutes de la même couleur, alors $v$ ne correspond

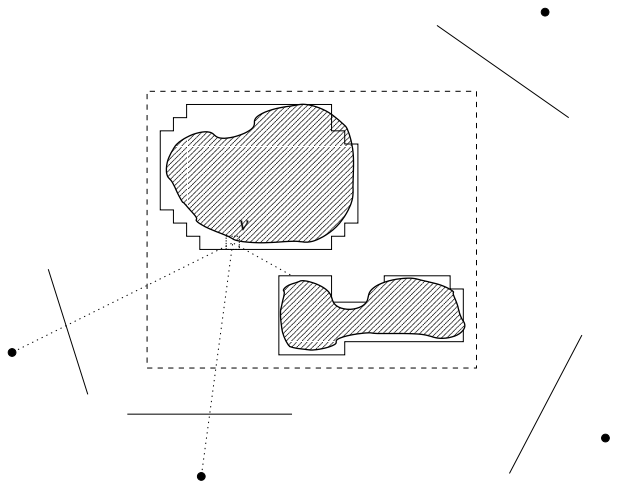



FIG. 4.22 – *Schéma de principe du* space carving.

pas à un point physique de la scène, et il est retiré de $\nu$. Sinon, $v$ est coloré selon la couleur de ses projections.

3. **Itération :** une fois que chaque voxel de $\nu$ ait été considéré, le processus est itéré sur les voxels restants. On s'arrête quand aucun voxel $v$ à la surface de $\nu$ ne peut être retiré.

On notera que cet algorithme prend en compte les occultations d'une manière similaire à celle de la méthode par ensemble de niveaux, puisque la structure courante de la scène reconstruite est utilisée pour évaluer si un voxel est occulté ou non.

Il reste à définir le test sur la couleur des projections de $v$. [Seitz et al.99] supposent qu'un voxel appartenant effectivement à la scène se projette dans chaque image sur plusieurs pixels de même couleur. Si l'écart-type des pixels appartenant à l'ensemble des projections est inférieur à un seuil, le voxel est conservé, sinon il est rejeté. [Kutulakos et al.98] se contente d'effectuer ce test sur les projections du centre du voxel considéré. Ces méthodes ont un inconvénient dû à la discrétisation de l'espace : si le pas choisi est trop important, certaines parties de la scène peuvent ne pas être reconstruites, pour peu qu'aucun voxel n'en soit suffisamment proche pour que toutes ses projections soient de la même couleur.

Pour éviter cela, et pour tenir compte de l'imprécision des points de vue en pratique, [Kutulakos00] considère les disques $\{C_i\}$ de quelques pixels de rayon centrés sur les projections du centre du voxel. Le voxel est retenu si et seulement si on peut trouver un ensemble de pixels $\{\mathbf{p}_i\}$, tels que $\mathbf{p}_i \in C_i$ et que les $\mathbf{p}_i$ soient approximativement de la même couleur. Néanmoins, les reconstructions obtenues de cette façon sont plus larges que la scène originale, et on remarquera qu'on retrouve un problème similaire à celui posé par les fenêtres de corrélation : un disque $C_i$ peut recouvrir deux objets différents.

### 4.7.2 Notion de reconstruction maximale

Si le pas de discrétisation est suffisamment fin, les points de vue suffisamment précis et en l'absence de bruit dans les images, on a la garantie, en projetant $\nu$ dans les différents plans image, d'obtenir les images qui ont servi à la reconstruction [Kutulakos et al.98]. Ces auteurs montrent que ceci ne veut néanmoins pas dire que $\nu$ correspond exactement à la scène à reconstruire. Ces auteurs montrent en effet que dans le cas général, il existe une famille de reconstructions possibles qui donneront les mêmes images à partir des mêmes points de vue. Ce phénomène est illustré

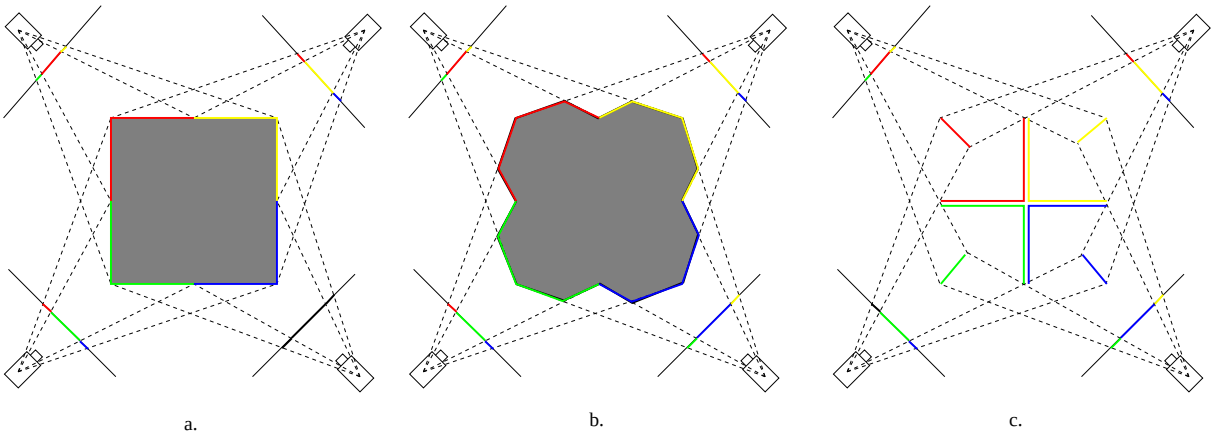



FIG. 4.23 – *a : Scène à reconstruire, vue par 4 caméras; b : Reconstruction maximale (figures tirées de [Kutulakos et al.98]); c : Une autre reconstruction possible.*

figure 4.23 : la figure 4.23.a présente la scène à reconstruire, un carré dont les quatre coins ont une couleur différente des autres, et vu par quatre caméras. À partir des quatre images du carré et de la position des caméras, on ne peut que déterminer que la scène appartient à une famille de reconstruction dont les figures 4.23.b et 4.23.c montrent deux représentants. La reconstruction de la figure 4.23.b, appelée *reconstruction maximale*, est particulière en ce sens qu'elle englobe toutes les reconstructions possibles. Kutulakos et al. montrent que leur algorithme retrouve cette reconstruction maximale.

*Dans le cas général, on ne peut donc pas reconstruire exactement une scène à partir d'un ensemble d'images de cette scène, sans ajouter de connaissances autres que les images.* Ceci est vrai en particulier en présence de zones d'intensité uniforme. L'algorithme de *space carving* reconstruit ces zones plus bombées qu'elles ne le sont en réalité. Néanmoins, il n'est pas interdit de supposer en pratique que de telles zones sont planes, contrainte que cet algorithme ne peut imposer, contrairement à d'autres algorithmes comme celui de [Faugeras et al.97].

### 4.7.3 Discussion

On remarquera une certaine analogie entre le *space carving* et [Faugeras et al.97], analogie qui apparaît surtout quand on considère l'implantation de ce dernier. Le *space carving* a pour avantage son temps de convergence : quelques minutes contre quelques heures pour la méthode précédente.

Un inconvénient du *space carving* a été mis en évidence par [Seitz et al.99], qui apparaît quand la fréquence des textures présentes dans la scène est trop grande par rapport à la résolution des images. La figure 4.24.a montre une surface vue par 4 caméras. Les figures 4.24.b à 4.24.j montrent quant à elles la reconstruction de cette surface par *space carving*, quand on augmente progressivement la fréquence de la texture de la surface. La figure 4.24.b montre la reconstruction quand la surface est d'une couleur uniforme. Au fur et à mesure que la fréquence augmente, la reconstruction converge vers la zone correcte (figure 4.24.h). À partir d'un certain point, la reconstruction devient de plus en plus mauvaise (figure 4.24.i), voire vide (figure 4.24.j). Un défaut analogue apparaît peut-être également pour [Faugeras et al.97], bien qu'il soit difficile d'en juger en l'absence d'expérimentation.

Un tel phénomène a toutes les chances de se produire en particulier dans les scènes d'extérieur, généralement très texturées. À l'inverse, les scènes d'intérieur présentent elles des zones d'intensité

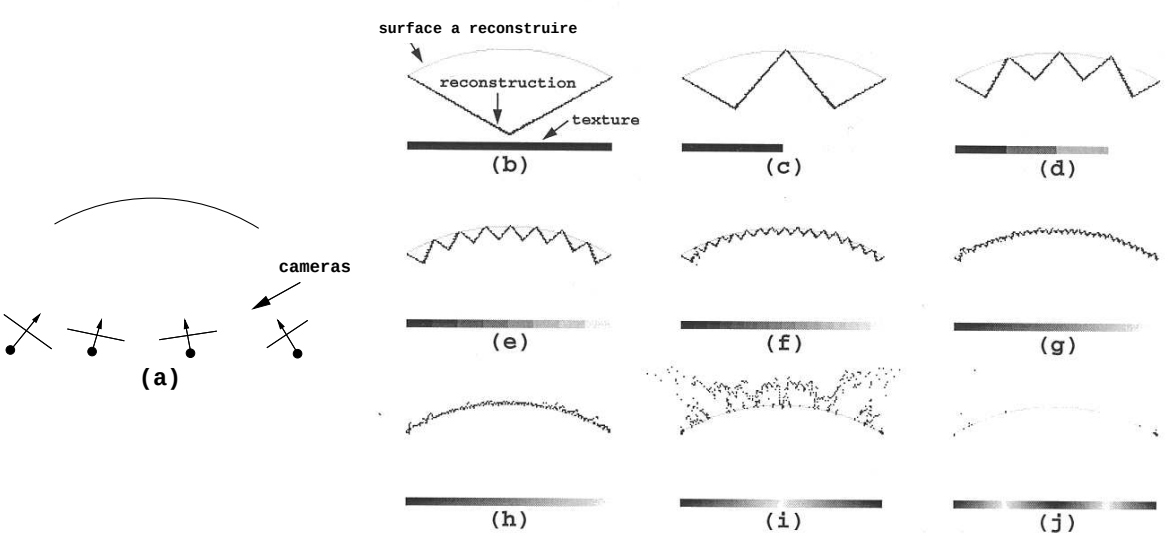



FIG. 4.24 – *a : Surface à reconstruire, vue par 4 caméras; b. à j : évolution de la reconstruction par space carving quand on augmente la fréquence de la texture de la surface (figure tirée de [Seitz et al.99]).*

uniforme, qui sont reconstruites un peu « bombées » comme on l'a déjà souligné. Le *space carving* semble donc peu intéressant en pratique.

Face aux difficultés pour reconstruire une scène de façon automatique, quelques propositions ont été faites reposant sur la mise à contribution d'un opérateur humain. La difficulté est alors de définir une interface utilisateur, qui minimise l'intervention de l'opérateur en automatisant ce qui peut l'être sans risque d'erreur de reconstruction.

## 4.8 Reconstruction avec intervention de l'utilisateur

### 4.8.1 Façade

[Debevec et al.96] ont défini, dans leur logiciel « Façade », une méthode qui requière une intervention importante de l'utilisateur, mais qui permet d'obtenir efficacement des résultats précis. L'utilisateur commence par choisir un ensemble d'images de la scène à reconstruire, pour lesquelles les paramètres internes de la caméra sont connues. Il dispose de plusieurs primitives polyédrales (parallélépipède rectangle, prisme, pyramide...) qu'il assemble à l'aide d'une interface pour modéliser la scène (figure 4.25.b). Il définit ensuite les segments 2D dans les images qui correspondent aux arêtes visibles des primitives utilisées (figure 4.25.a). À l'aide de ces correspondances, l'algorithme peut alors retrouver la position spatiale des primitives, ce qu'on peut vérifier en reprojetant le modèle ainsi reconstruit dans les images utilisées (figure 4.25.c). Le modèle peut être ensuite raffiné en reconstruisant les détails présents sur les facettes des primitives à l'aide d'un algorithme de vision stéréoscopie binoculaire. La mise en correspondance est grandement facilitée grâce au modèle défini par l'utilisateur, qui permet de connaître les occultations et donne une bonne idée de la disparité des correspondants.

Cette méthode a prouvé son efficacité en permettant de générer des films de synthèse de sites

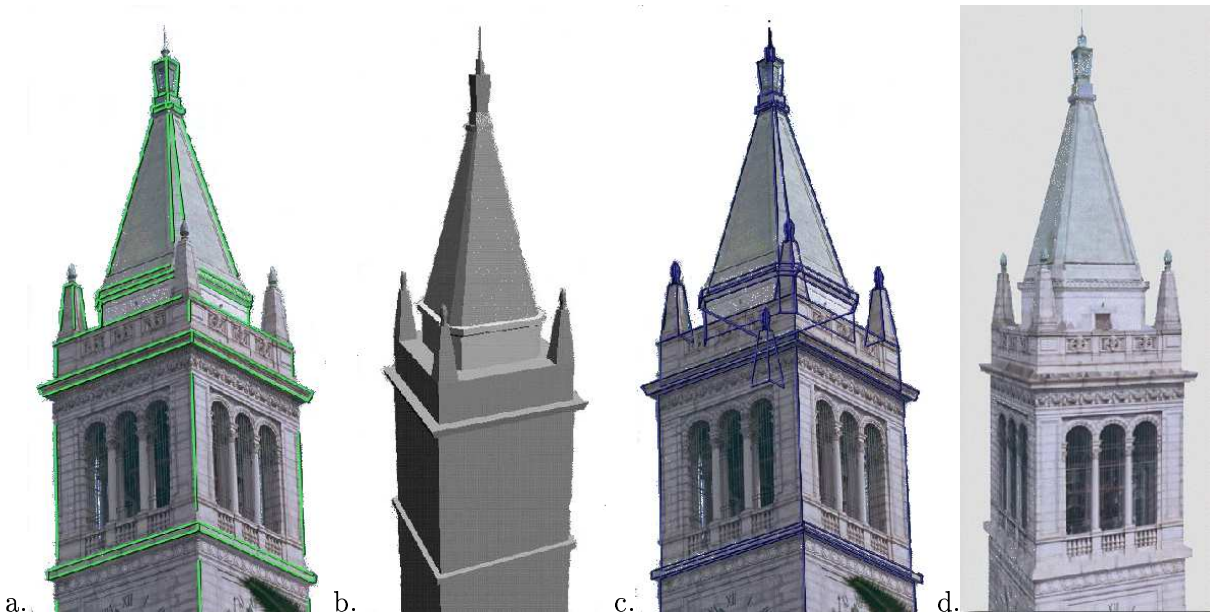


FIG. 4.25 – *a. Une des images utilisées dans Façade et les segments ayant un correspondant dans le modèle construit par l'utilisateur (b); c. Reprojection de ce modèle dans l'image; d. Vue du modèle texturé (images extraites de [Debevec et al.96]).*

architecturaux existants. Signalons également que la société MetaCreations [Metacreations00] a commercialisé un logiciel basé sur cette idée, Canoma, limité à une seule image, et dont l'interface utilisateur présente le modèle en train d'être construit projeté dans cette image. Cette méthode est néanmoins limitée à des scènes de géométrie relativement simple.

### 4.8.2 Image Modeler

Le logiciel Image Modeler (voir figure 4.26) de la société RealViz [Realviz00] reprend la possibilité d'utiliser des primitives pour modéliser une scène, mais en y ajoutant la création de facettes 3D polyédriques pour étendre les capacités de modélisation. L'utilisateur désigne les projections de points 3D pertinents pour la reconstruction dans plusieurs vues calibrées d'une scène. Ces points sont ensuite triangulés automatiquement ou manuellement quand le maillage automatique n'est pas correct [Goncalves00].

## 4.9 Conclusion

Nous venons de passer en revue les méthodes existantes de reconstruction, qu'on peut répartir en trois catégories :

- méthodes qui adoptent une démarche « images vers l'espace 3D » ;
- méthodes qui adoptent une démarche « espace 3D vers les images » ;
- méthodes faisant appel à l'utilisateur.

Les méthodes automatiques se basent sur l'hypothèse qu'un point 3D de la scène a la même couleur (ou intensité) dans toutes les images où il est visible, vérifiée pour des objets Lambertiens et utilisent des contraintes géométriques. Cette hypothèse n'est en général pas suffisante puisqu'elle laisse subsister de nombreuses ambiguïtés sur la scène à reconstruire. Dans certaines

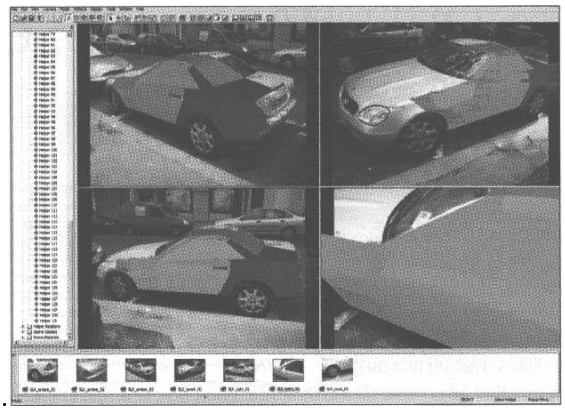
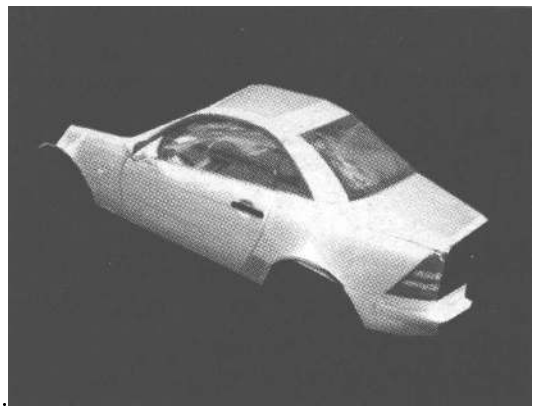
a.  b. 

FIG. 4.26 – *a. Interface de ImageModeler 2.0; b. Modèle reconstruit (images extraites de [Goncalves00]).*

conditions, en particulier quand le nombre d'images est faible, la scène reconstruite puis reprojetée peut donner les images de départ, sans qu'elle corresponde à la scène réelle. Il faut alors utiliser une heuristique sur la régularité de la surface.

D'autres facteurs viennent compliquer en pratique l'hypothèse de départ :

- les limites des caméras, à la fois la présence de bruit dans les images, qui fait qu'un point 3D n'a pas exactement la même couleur dans toutes les images, et également la résolution finie des images. La solution adoptée est alors l'utilisation de fenêtres de corrélation;
- l'imprécision des points de vue, qui n'est qu'exceptionnellement prise en compte pour la reconstruction dense.

En l'absence d'objets occultants dans la scène, les résultats peuvent être acceptables. En effet, la cause majeure d'imprécision des méthodes automatiques est bien la présence d'occultations dans la scène, pour différentes raisons que nous avons pu voir dans ce chapitre, notamment :

- le fait que des points soient visibles dans des images, mais cachés dans d'autres, difficile à prendre en compte;
- les fenêtres de corrélation qui recouvrent plusieurs objets près des contours occultants.

Toutes les approches envisageables n'ont pas forcément été explorées. Nous avons déjà relevé partie 4.4.4 qu'aucun algorithme de type « images vers l'espace 3D » ne considère des hypothèses de mise en correspondance sur plusieurs images ([Koch et al.98] se limitent à chaîner des appariements sur des paires d'images). Cette absence de travaux en ce sens est sans doute au nombre d'informations qu'il faudrait alors traiter. La dimension du problème n'avait d'ailleurs pas échappé aux auteurs des premiers travaux de reconstruction 3D [Marr et al.76]. Une autre piste pourrait être la fusion avec d'autres indices que la stéréoscopie, comme l'illumination (*shape from shading*) mais cette fusion est difficile et a été peu explorée (voir tout de même [Fua et al.96]). Enfin, le lecteur intéressé pourra se rapporter aux résultats du *workshop* sur la stéréovision qui se tiendra en décembre 2001, en conjonction avec la conférence *Computer Vision and Pattern Recognition*, qui prévoit notamment de comparer les différents algorithmes de reconstruction par stéréovision.

Dans une perspective à moyen terme, aucune des méthodes automatiques n'est encore suffisamment fiable pour fournir, à partir d'une séquence quelconque, un modèle que nous pourrions utiliser sans correction importante de la part de l'utilisateur. Dans des contextes plus contraints, une reconstruction en vision par ordinateur peut donner de bien meilleurs résultats : en particu-

lier, la reconstruction d'un objet filmé sur une table tournante fournit des modèles 3D de bonne qualité (voir par exemple [Szeliski93, Boyer et al.97, Sullivan et al.98]).

Les solutions interactives restent une approche réaliste de la reconstruction, dans le cadre d'une séquence d'images quelconque. Elles sont malgré tout fastidieuses et surtout limitées à des formes géométriques relativement simples. Face aux difficultés posées par la reconstruction, quelques auteurs ont proposé des solutions spécifiques pour la gestion des occultations, que nous allons voir maintenant.

# Chapitre 5

# État de l'Art sur la gestion des occultations en Réalité Augmentée

Si la littérature traitant de la reconstruction en vision est très développée, ce n'est pas la cas des travaux qui considèrent la gestion des occultations dans un contexte similaire au nôtre, c'est-à-dire dans une séquence quelconque, sans connaissance *a priori* sur la structure 3D de la scène réelle. Il est pourtant indispensable de considérer le problème en lui-même : comme nous l'avons exposé dans le chapitre précédent, les méthodes générales de reconstruction en vision ne permettent pas d'obtenir les résultats précis que nous souhaitons. On pourra s'en rendre compte sur les quelques exemples présents dans la littérature (voir [Wloka et al.95], ou [Kanade et al.95] dans le cadre de la Réalité Mixte dont nous avons déjà montré une image de résultat figure 1.9).

Nous présentons donc dans ce chapitre les deux travaux traitant explicitement des occultations. Le premier propose une méthode automatique basée sur les contours. Nous discutons ensuite des méthodes de suivi d'objets dans une séquence d'images car il nous a semblé naturel d'envisager de les utiliser pour résoudre notre problème. La présentation des limites du suivi nous permettra d'introduire le deuxième article, faisant intervenir l'utilisateur pour obtenir un résultat plus précis, ainsi que notre méthode que nous présenterons dans le chapitre suivant.

## 5.1 Approche basée contours [Berger97]

### 5.1.1 Description

[Berger97] commence par mettre en évidence la relation entre les contours occultants et le masque d'occultation, telle que nous l'avons décrite dans la partie 4.1.1. Partant de cette remarque, la méthode développée consiste à déterminer les contours de l'image qui sont devant l'objet virtuel, et à partir de ces contours, retrouver le masque d'occultation. Plus précisément, pour chaque image de la séquence à augmenter, on effectue les étapes suivantes :

- **Étape 1 : Initialisation**
  On commence par calculer la région, notée $m_I$, de l'image considérée qui correspond à la projection de l'objet virtuel, en supposant que cet objet soit entièrement visible (voir figure 5.1.b).
- **Étape 2 : Suivi des contours**
  On considère alors les chaînes de contour $\mathcal{C}_i$ qui intersectent $m_I$. Elles sont suivies dans l'image suivante (figure 5.1.d) à l'aide d'un outil présenté dans [Berger94], basé notamment sur le flot optique entre les deux images consécutives. Ces chaînes sont ensuite mises en

correspondance à l'aide de la contrainte épipolaire. Pour chaque point $m$ d'une chaîne $\mathcal{C}$, son correspondant est calculé comme étant l'intersection entre la droite épipolaire de $m$ et la courbe suivie. S'il existe plusieurs intersections, des heuristiques sont utilisées pour déterminer l'intersection correcte.

- **Étape 3 : Étiquetage des points de contour**
  La profondeur des points de contour est comparée à celle du point l'objet virtuel qui se projette au même endroit, ce qui permet de les étiqueter *Devant*, *Derrière* ou *Douteux* quand la précision de détection et de suivi de courbes ne permet pas de comparer les profondeurs de façon sure (les points de vue sont obtenus à l'aide d'une mire de calibration et donc très précis). En fait, la comparaison n'est pas effectuée en 3D, mais en 2D, entre la disparité du point de contour et celle du point de l'objet virtuel. Ainsi, on peut étiqueter *Douteux* le point quand la différence des disparités est inférieure à la précision de détection des courbes (inférieure au pixel).
- **Étape 4 : Séparation des chaînes de contours entre objets différents**
  On considère à partir de maintenant uniquement les chaînes qui ne sont composées que de points étiquetés *Devant*. Cette étape groupe les chaînes qui appartiennent au même objet occultant. Elle est nécessaire quand plusieurs objets sont susceptibles d'occulter l'objet virtuel simultanément.
  On définit tout d'abord la distance entre deux courbes $\mathcal{C}_i$ et $\mathcal{C}_j$ par $D(\mathcal{C}_i,\mathcal{C}_j) = \min_{x\in\mathcal{C}_i, y\in\mathcal{C}_j} d(x,y)$, et on considère que deux courbes telles que $D(\mathcal{C}_i,\mathcal{C}_j) < s$, avec $s$ un seuil de quelques pixels, appartiennent au même objet. Ceci permet de construire un graphe de proximité, où les nœuds sont les chaînes de contour : deux nœuds sont connectés si et seulement si la distance entre les chaînes correspondantes est inférieure à $s$. Il ne reste plus qu'à chercher les cliques maximales $\mathcal{H}_i$ du graphe de proximité : chaque clique $\mathcal{H}_i$ correspond à un objet occultant.
- **Étape 5 : Calcul du masque d'occultation** Il reste maintenant à déterminer le masque d'occultation correspondant à chaque clique $\mathcal{H}_i$. Ce problème est rendu difficile par le fait qu'une partie du contour du masque d'occultation n'a pas été détectée ou pas suivie (voir figure 5.1.e), et nous allons détailler sa résolution car il est important.
  Pour pouvoir déduire des courbes appartenant à $\mathcal{H}_i$ un masque d'occultation unique, on fait l'hypothèse que le contour du masque d'occultation est lisse, ce qui permet d'utiliser un contour actif. Un contour actif [Kass et al.88] est une courbe $v$ à laquelle on attribue une énergie $E(v)$ dont le minimum constitue le contour d'intérêt. $E(v)$ est définie par :

$$E(v) = E_{ext}(v) + \lambda E_{int}(v) \tag{5.1}$$

  avec $E_{int} = \int_v(\alpha|v'|^2 + \beta|v''|^2)$ qui permet de régulariser $v$, et $E_{ext} = \int_v(-|\nabla(v))$ qui permet de faire converger $v$ vers les contours proches en maximisant le gradient le long du contour. Comme le contour actif se contracte quand le gradient est nul, on l'initialise par un contour fermé autour de l'ensemble des courbes de $\mathcal{H}_i$ (voir figure 5.1.g). La carte de gradient est calculée comme suit : soit $I_0$ la carte de contours avec $I_0(x,y) = 255$ si $(x,y)$ appartient à une des chaînes de $\mathcal{H}_i$, et 0 sinon. La carte de gradient est alors le produit de convolution entre $I_0$ et un filtre Gaussien. Le contour actif converge alors vers une courbe régulière qui « englobe » $\mathcal{H}_i$ (voir figure 5.1.h).
  Finalement, l'intérieur du contour actif définit le masque d'occultation. En fait, ce masque (voir figure 5.1.i) est étendu à l'objet occultant (il n'est pas uniquement composé de points occultés de l'objet virtuel) mais cela n'est pas gênant pour la composition (voir figure 5.1.j).

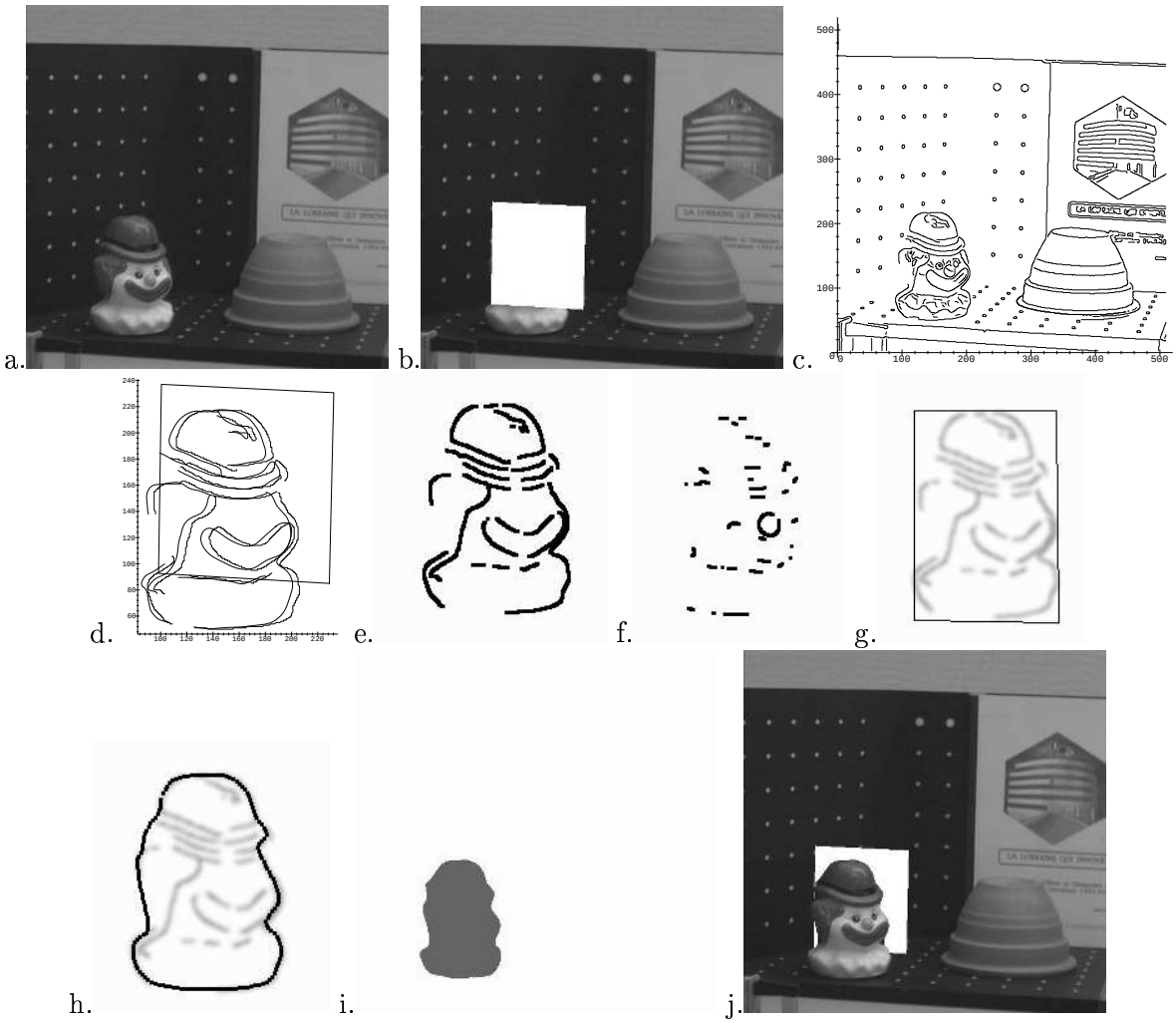


FIG. 5.1 – *Incrustation d'un rectangle virtuel (figures extraites de [Berger97]). a : image réelle considérée; b : région $m_I$; c : carte de contours; d : résultat du suivi de courbes; e : contours étiquetés* Devant*; f : points étiquetés* Douteux*; g : champ de gradient créé par les contours étiquetés* Devant *et le contour actif initial; h et i : masque après la convergence du contour actif; j : résultat de l'incrustation.*

### 5.1.2 Discussion

Ce travail est intéressant car il montre clairement les relations entre les contours de l'image et le masque d'occultation. Il fournit également une méthode pour retrouver le masque de l'objet occultant malgré l'absence de certaines portions de contour.

Malheureusement, cette méthode doit pour cela supposer que le contour du masque est régulier, hypothèse qui n'est pas toujours valide (si on considère un objet polyédrique par exemple), ce qui peut rendre le détourage imprécis. Le contour actif peut également converger vers un mauvais contour, si le contour attendu n'a pas été détecté ou reconstruit. Enfin, l'approche basée contour est évidemment moins pertinente dans le cas de scènes très texturées, puisqu'on risque alors de ne pas détecter suffisamment de contours pertinents.

## 5.2 Approche par suivi temporel 2D des objets occultants

Une idée naturelle pour retrouver les masques d'occultation sur l'ensemble d'une séquence est de suivre le ou les objets occultants d'une image à l'autre, à partir d'un détourage initial, obtenu manuellement ou par la méthode décrite précédemment.

Pour certaines applications, quand les objets à suivre sont connus, on peut disposer de leur modèle 3D [Gennery92], ce qui facilite évidemment le suivi. Dans un cadre un peu moins restrictif, on peut utiliser un modèle 3D générique, par exemple de véhicules automobiles [Koller et al.92] ou de personnes [Delamarre et al.99]. Dans le cas général qui nous intéresse, on ne dispose pas de modèle 3D et le suivi s'effectue uniquement à partir de données 2D. La silhouette de l'objet à suivre est alors définie par une courbe 2D (un ensemble de segments de droite, de B-splines ou de courbes de Bézier). Cette silhouette correspond donc au masque d'occultation, éventuellement étendu à tout l'objet occultant comme dans [Berger97]. Étant donnée la silhouette $S$ dans l'image $I(.,t)$, on cherche la silhouette $S'$ dans l'image suivante $I(.,t+1)$.

De nombreux travaux ont été consacré au suivi, et en particulier au suivi de courbes et de régions d'intérêt. Afin de prendre en compte la variation de forme de l'objet dans les images, due aux variations de points de vue, aux contours apparents et éventuellement au fait que l'on considère un objet non rigide, ces méthodes reposent essentiellement sur le concept de contours actifs [Berger93, Bascle et al.94] (voir l'équation 5.1). Cependant, les contours actifs nécessitent une initialisation suffisamment proche de la position attendue pour être efficace. C'est pourquoi on utilise généralement une approche hiérarchique pour le suivi : on commence par établir une prédiction de la position du contour en se basant sur des estimations locales du mouvement 2D, puis le résultat de cette prédiction est affiné par un ajustement local souvent à base de contours actifs. Ceci permet de suivre assez facilement des contours même quand le déplacement apparent de la silhouette est important entre les deux images. Nous détaillons ci-dessous les méthodes employées pour les phases de prédiction et d'ajustement local.

### 5.2.1 Prédiction

#### Prédiction par filtre de Kalman

Une première possibilité de prédiction est l'utilisation d'un filtre de Kalman [Deriche et al.90, Blake et al.93]. Cependant, cela nécessite que le modèle dynamique utilisé par le filtre soit adapté au mouvement considéré. Ceci est loin d'être toujours le cas. En effet les mouvements humains lors des prises de vue ne peuvent être pris en compte par un modèle dynamique simple comme le souligne [Ravela et al.96] dans le cadre de la Réalité Augmentée.

#### Prédiction par calcul du mouvement

Pour ces raisons, de nombreuses méthodes essaient d'inférer la prédiction à partir de l'information locale de mouvement 2D fournie par le flot optique ou la corrélation. Comme l'information de mouvement fournie par le flot ou la corrélation peut être très bruitée, voire erronée, on cherche à estimer globalement le mouvement du contour (ou de la région définie par le contour), qui correspond le mieux au mouvement calculé en chaque point. Comme le calcul du flot optique, particulièrement sa composante tangentielle, est peu fiable, le calcul est en général effectué sous l'hypothèse d'un mouvement paramétrique. Les modèles de mouvement les plus couramment utilisés sont les modèles affines et parfois homographiques [Meyer et al.92, Bonnaud et al.94]. Le choix du modèle de mouvement est important : un modèle trop complexe peut mener à l'échec car il considérera davantage le bruit sur le flot calculé. Comme la phase de prédiction est suivie

par une phase d'affinement local, nous n'avons besoin que d'une estimation assez grossière de la prédiction et nous avons donc retenu le modèle affine.

Le modèle de mouvement étant choisi, on peut l'estimer de plusieurs façons soit en considérant une approche de type contour, soit une approche de type région. Dans l'approche de type contour, on va utiliser seulement l'information de mouvement sur ou au voisinage immédiat du contour à suivre. Dans l'approche région, on utilisera toute l'information de mouvement à l'intérieur du contour.

- Approche de type contour. En utilisant une approche par corrélation, on va chercher le mouvement affine $D$ tel que les intensités sur les contours $S$ et $S_D = D(S)$ soient similaires. $D$ est donc la transformation qui minimise

$$\sum_{\mathbf{m}\in S} (I(D(\mathbf{m}),t+1) - I(\mathbf{m},t))^2$$

où $I(\mathbf{m},t)$ est l'intensité au pixel $\mathbf{m}$ à l'instant $t$.
Cette estimation est particulièrement coûteuse car elle ne peut s'effectuer que par une méthode d'optimisation itérative. C'est pourquoi il est nettement plus rapide d'utiliser la propriété d'invariance de l'intensité à l'ordre 1 en utilisant le flot optique. En dérivant l'équation d'invariance $I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t)$ et en utilisant le développement de Taylor de $I$ au voisinage de $(x,y)$, on obtient la relation $\nabla I(x,y,t).\overrightarrow{f} + \frac{\partial I}{\partial t}(x,y,t) = 0$, qui permet d'estimer (au premier ordre) la composante normale du flot $\overrightarrow{f} = (\delta x, \delta y)$.
Comme seule la composante normale du flot est véritablement fiable, on cherche $D$ minimisant [Wohn et al.91, Berger et al.99]

$$\sum_{\mathbf{m}\in S} ||(\overrightarrow{\mathbf{m}D(\mathbf{m})}.\overrightarrow{n}(\mathbf{m}))\overrightarrow{n}(\mathbf{m}) - \overrightarrow{f}^{\perp}(\mathbf{m})||^2$$

où $\overrightarrow{n}(\mathbf{m})$ est la normale à $S$ au point $\mathbf{m} = (x,y)$ et $\overrightarrow{f}^{\perp}(\mathbf{m})$ est la composante normale du flot optique. Cet estimation est raffinée itérativement entre l'image $I(.,t)$ compensée par $D$ et $I(.,t+1)$.
Chaque étape se réduit à une estimation aux moindres carrés classique, ce qui accélère évidemment le calcul global de $D$. Il faut noter que de telles méthodes peuvent échouer si le mouvement apparent est trop grand car le flot calculé est alors erroné. Dans ce cas, une méthode plus coûteuse en temps de calcul mais donnant de meilleurs résultats consiste à d'abord rechercher par corrélation d'intensité les correspondants d'un certain nombre de points de la silhouette [Lamberti et al.93, Berger et al.99], d'une manière semblable à celle déjà présentée partie 2.4.2 puis à utiliser ensuite la méthode à base de flot
- Approche de type région. Dans ce cas, on cherche le mouvement permettant la meilleure corrélation d'intensité entre les régions intérieures aux courbes $S$ et $S_D = d(S)$ [Bascle et al.94]. $D$ minimise donc

$$\sum_{\mathbf{m}\in R_S} (I(D(\mathbf{m}),t+1) - I(\mathbf{m},t))^2$$

où $R_S$ est la région délimitée par le contour $S$.
La recherche de la meilleure corrélation peut être effectuée par descente de gradient [Bascle et al.94], par une méthode stochastique de type recuit simulé [Kervrann et al.94]... Cette méthode est encore plus coûteuse puisque la corrélation n'est plus limitée au contour mais à l'ensemble de l'intérieur de la silhouette. Par contre elle est plus fiable que l'approche contour

dans la mesure où la texture à l'intérieur de la région est prise en compte pour le calcul du mouvement. Par contre la précision de la localisation de la frontière de l'objet suivi est en général moins bonne car une modification très légère de la frontière n'aura que peu d'influence sur la corrélation calculée sur toute la région.

Il faut être conscient qu'une approche contour est moins fiable qu'une approche région puisque l'information utilisée est plus locale; par contre, elle permet *a priori* une meilleure localisation des contours. C'est pourquoi nous utilisons un compromis entre ces deux méthodes en limitant le calcul de la corrélation à une bande intérieure plutôt qu'à l'ensemble de l'intérieur des courbes. La corrélation d'intensité permet d'obtenir des résultats plus fiables, sans que la partie intérieure aux courbes qui n'améliore pas la localisation des contours n'influence l'estimation de $D$.

### 5.2.2 Ajustement local

L'étape précédente doit fournir une région proche de la silhouette de l'objet dans la nouvelle image, mais elle n'est en général pas suffisante pour un suivi sur un grand nombre d'images, pour deux raisons :

- elle ne permet pas de prendre en compte l'apparition de nouveaux détails le long de la silhouette;
- les modèles de mouvement utilisable pour la première étape ne sont pas assez généraux pour prendre en compte un mouvement quelconque. L'homographie permet de prendre en compte les changements de perspective mais de manière exacte uniquement pour les objets plans. Pour des objets non plans, les modèles de mouvement ne sont valables que pour un mouvement de faible amplitude.

C'est pourquoi elle est suivie par un affinement local, lui-même éventuellement décomposé en plusieurs niveaux [Chen et al.98].

La méthode la plus populaire d'affinement est l'utilisation d'un contour actif: à partir de la position prédite $S_D$, on laisse un contour actif converger dans l'image $I'$ en espérant que le contour atteint soit bien le contour correspondant. Dans un contexte photométrique simple où le contour à suivre est un contour très marqué, cette méthode réussira sans problème. Par contre, des problèmes vont surgir dès que le contour à suivre est un contour faible ou lorsque l'objet est très texturé, le contour actif convergeant alors vers un mauvais contour.

Depuis l'article initial de Kass [Kass et al.88], de nombreux travaux ont eu pour but d'améliorer la convergence des modèles actifs de contours ou d'introduire des contraintes sur son comportement. Parmi les plus importants, on peut citer [Cohen91] qui a proposé l'utilisation d'une « force de ballon » permettant l'expansion d'un contour actif sans qu'il soit retenu par des gradients forts créés par du bruit. L'étude des paramètres intervenant dans le modèle numérique a été faite dans [Fua et al.89, Samadani91, Berger91, Cohen92] afin d'améliorer la convergence des modèles actifs de contour et de réduire leur comportement parfois oscillant. Dans le même but, des modifications des forces externes ont été envisagées: utilisation d'une force de gradient normalisée pour éviter que les forts gradients aient trop d'influence sur le contour actif [Cohen92], utilisation d'une fonction du module du gradient ($\ln|\nabla I|$ ou $\exp|\nabla I|$) pour réduire ou accélérer la convergence... Malgré tout, ces travaux n'ont pas réellement permis de mieux contrôler la convergence des contours actifs. C'est pourquoi des approches de type *Ziplock snakes* [Neuenschwander et al.95] qui permettent d'insérer des points de passage dans le contour actif ont été proposées dans des stratégies de type croissance de *snake*.

Mais l'avancée la plus marquante de ces dernières années a consisté en le développement d'un modèle géométrique de contour actif [caselles et al.92, Malladi et al.95] permettant d'avoir une

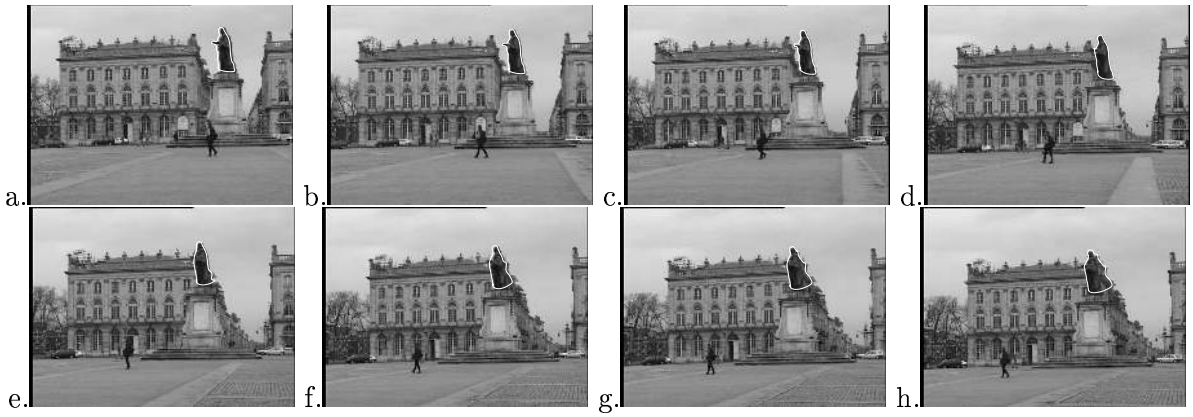



FIG. 5.2 – *Suivi de la statue sur la séquence Stanislas.*

courbe active à topologie variable, en modélisant la courbe active comme courbe de niveau d'une surface définie implicitement. Ceci permet de détecter sans problème des objets constitués de plusieurs parties, contrairement au modèle initial de Kass et al.

En conclusion, on peut dire que les problèmes topologiques liés aux contours actifs sont maintenant résolus ainsi que certains problèmes de convergence de l'algorithme numérique. Cependant, la phase d'initialisation reste prépondérante dans la qualité du résultat obtenu et il est à l'heure actuelle très délicat d'utiliser ce concept pour détecter des contours faibles environnés par des contours forts.

### 5.2.3 Discussion

Face à la difficulté voire à l'impossibilité d'obtenir automatiquement un détourage précis des contours occultants, dans un contexte de post-production il n'est pas interdit d'imaginer de demander à l'utilisateur de déterminer les objets occultants (cette tâche est facile puisque c'est lui qui impose la trajectoire des objets virtuels), et de les détourer manuellement dans une image, détourage qui servirait à l'initialisation du suivi.

#### Mauvaise stabilité du suivi

Mais outre la nécessité d'employer un contour actif qui risque d'altérer localement la qualité de la localisation de la silhouette, un gros inconvénient du suivi est que la qualité de la silhouette dans une image dépend de celle retrouvée dans l'image précédente. Une petite erreur de localisation peut dégénérer en une grande imprécision. Par exemple, nous avons testé le suivi de la statue dans la séquence Stanislas (voir figure 5.2). Le suivi se passe correctement pour les premières images, mais quand la statue est proche de l'opéra, le contour actif est attiré par le toit des bâtiments (figure 5.2.e), ce qui génère une erreur importante dans les images suivantes (figure 5.2.f-h).

#### Pas ou peu de prise en compte des changements d'aspect

Imaginons que l'objet suivi soit un objet polyédrique. Si une face de cet objet apparaît lors de la séquence, le suivi risque de ne pas prendre en compte cette nouvelle face. L'étape d'affinement peut éventuellement retrouver cette face si le contour actif converge vers ses arêtes mais ceci

reste aléatoire. Le même phénomène peut se produire pour un objet non polyédrique quand de nouvelles parties de l'objet apparaissent le long de la séquence.

Si l'on reste dans une approche semi-automatique, on peut envisager que l'utilisateur détoure à nouveau l'objet, quand un changement d'aspect se produit. Cependant, il est dommage que le suivi ne puisse prendre en compte l'information apportée par ce nouveau contour dans les images précédentes.

#### Limitation des modèles de mouvement

Enfin, le suivi 2D prend mal en compte les changements de perspective, sauf pour les objets plans pour lesquels on peut utiliser une homographie comme modèle de déformation globale. Pourtant, dans le cas d'un objet rigide, la déformation de la silhouette le long de la séquence ne dépend que de la géométrie 3D de cet objet et du mouvement de la caméra. Il serait intéressant de prendre en compte cette géométrie 3D pour déterminer la silhouette.

Nous allons voir que l'approche de Ong et al. permet d'éviter les trois points que nous venons de soulever.

## 5.3 Approche interactive [Ong et al.98]

### 5.3.1 Description

[Ong et al.98] ont également proposé de faire appel à l'utilisateur. Celui-ci doit en effet choisir un certain nombre d'images de la séquence, appelées images-clé, dans lesquelles il détoure l'objet occultant. Contrairement au suivi décrit plus haut qui est une approche purement bidimensionnelle, leur approche tient compte de la nature tridimensionnelle des objets occultants. Les auteurs décrivent leur méthode pour retrouver alors automatiquement la silhouette de l'objet occultant dans les autres images de la séquence, méthode basée sur la construction de ce qu'ils appellent un clone de l'objet réel. Ce clone est une reconstruction 3D de l'objet modélisé par un ensemble de voxels, qui n'est pas forcément très exacte, mais qui doit permettre de résoudre correctement les occultations. En reprojetant ce clone dans chaque image de la séquence (pas seulement les images-clé), on retrouve la silhouette de l'objet réel occultant et sa profondeur.

Décrivons en détail la construction et l'utilisation d'un clone :

- **Choix des images-clé**
  Chaque image où un changement important de l'apparence de l'objet réel intervient (comme l'apparition ou la disparition d'une face) doit être retenue comme images-clé par l'utilisateur. Celui-ci peut évidemment définir plus d'images-clé : plus elles seront nombreuses, plus les résultats seront précis. Il doit ensuite détourer l'objet occultant dans chaque image-clé.
- **Définition d'une boite englobante à l'objet réel**
  On considère ensuite les points d'intérêt 2D qui se trouvent dans les silhouettes définies par l'utilisateur, et leurs correspondants 3D reconstruits, qui se situent donc sur l'objet occultant. La boite englobante de ces points fournit une première approximation de la boite englobante de l'objet réel. Elle peut également être corrigée par l'utilisateur. Elle est ensuite étendue pour que sa projection dans les images-clé recouvre les silhouettes détourées par l'utilisateur. Cette boite permet de restreindre l'espace à un volume proche de l'objet réel à reconstruire.

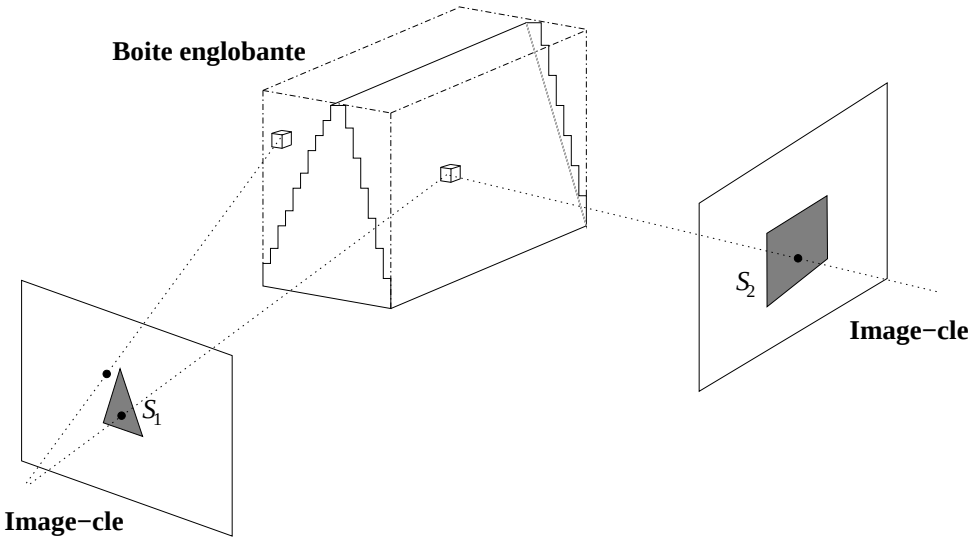



FIG. 5.3 – *Principe de construction du clone.*

- **Construction du clone**
  La boite englobante est ensuite divisée en voxels, suffisamment petits pour que la projection d'un voxel dans les images ait à peu près la taille d'un pixel. Chacun de ces voxels est alors projeté dans les images-clé : si l'une de ses projections se trouve à l'extérieur du contour détouré, le voxel n'appartient pas à l'objet occultant et il est supprimé. Les voxels restants constituent le « clone » de l'objet (voir figure 5.3).
- **Création de voxels dynamiques**
  Quand on projette le clone dans les images clés, il ne recouvre pas entièrement les silhouettes détourées manuellement : à cause de l'imprécision des points de vue, le clone est trop « creusé » en certains endroits, en particulier près de la frontière des silhouettes. Ong et al déterminent alors, pour chaque image clé, un ensemble de voxels, appelés voxels dynamiques. Ces voxels se projettent dans la partie non recouverte de la silhouette, et sont situés à une profondeur comprise entre la profondeur moyenne de la surface visible du clone et la profondeur moyenne de la surface invisible (voir figure 5.4). Ils permettent de limiter les erreurs de reconstruction.
- **Gestion des occultations**
  Pour chaque image de la séquence, la réunion des projections du clone et des voxels dynamiques associés à l'image-clé la plus proche correspond à peu près à la silhouette de l'objet réel considéré.
  Pour déterminer les occultations entre l'objet réel et les objets virtuels dans chaque image de la séquence, les auteurs calculent, pour l'image considérée, la carte de profondeur de ces deux ensembles de voxels. Il suffit alors, pour chaque pixel où se projette un objet virtuel, de comparer la profondeur de cet objet et celle contenue dans la carte de profondeur, pour savoir si l'objet virtuel est occulté ou non en ce pixel.

### 5.3.2 Discussion

Cette technique de reconstruction est en fait inspirée de travaux comme celui de [Szeliski93]. Ces travaux (auxquels nous avons déjà fait référence dans la conclusion du chapitre précédent

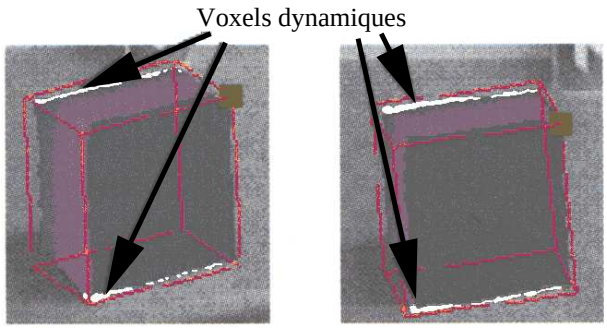



FIG. 5.4 – *Voxels dynamiques utilisés par [Ong et al.98] (figure extraite de l'article).*

sont généralement utilisés dans un contexte beaucoup plus contraint, où la caméra filme l'objet à reconstruire qui opère une rotation de 360 degrés (tout se passe donc comme si c'était la caméra qui tournait autour de l'objet). Dans ce contexte, les paramètres de la caméra peuvent être connus beaucoup plus précisément. De plus, l'objet étant filmé devant un fond connu, sa silhouette peut être retrouvée automatiquement dans chaque image. On peut alors retrouver un modèle 3D assez précis de l'objet.

Cette différence de contexte explique l'utilisation ici des voxels dynamiques, qui servent à compenser l'erreur sur les points de vues utilisés pour la reconstruction, mais qui ne permettent de corriger le modèle que d'une façon limitée. Si l'erreur reste trop importante, il faut alors ajouter une image-clé. Considérons l'un des exemples de [Ong et al.98], présenté figure 5.5, où l'objet réel occultant considéré est l'immeuble situé au premier plan. Alors que l'apparence de cet objet change très peu au cours de la séquence (aucune face n'apparaît ni ne disparaît), 6 images-clé doivent être définies (et donc 6 silhouettes doivent être détourées manuellement), alors qu'il faudrait réduire la tâche de l'utilisateur.

## 5.4 Conclusion

Malgré nos critiques de [Ong et al.98], nous pensons que demander à l'utilisateur de choisir des images-clé et de détourer les objets occultants dans celle-ci est une approche réaliste du problème. En effet, nous avons déjà vu que les méthodes automatiques de reconstruction 3D décrites dans le chapitre précédent se révèlent trop imprécises, et la méthode automatique présentée dans ce chapitre [Berger97] est trop sensible aux erreurs de détection des contours pour garantir ses résultats sur toutes les séquences.

De plus, l'utilisation des silhouettes définies par l'utilisateur pour construire un modèle 3D de l'objet occultant pas forcément précis mais permettant de retrouver les silhouettes dans les images intermédiaires donne des résultats plus stables qu'un suivi purement 2D, qui modélise mal les changements de perspective. La reconstruction et la reprojection nécessitent de connaître les points de vue pour chaque image, mais ceci n'est pas contraignant puisque ceux-ci doivent connus pour synthétiser l'image de l'objet virtuel.

Cependant, la représentation d'un tel modèle 3D par un ensemble de voxels ne nous semble pas adaptée si l'on souhaite pouvoir prendre en compte les erreurs de points de vue, ce qui est indispensable en pratique pour obtenir des résultats précis.

L'utilisation de toutes les images-clé pour construire un clone unique est également maladroite. On pourra remarquer en effet (et cela sera justifié dans le chapitre suivant) que la partie visible d'un objet dans une image située entre deux images-clé peut être déduite des seules

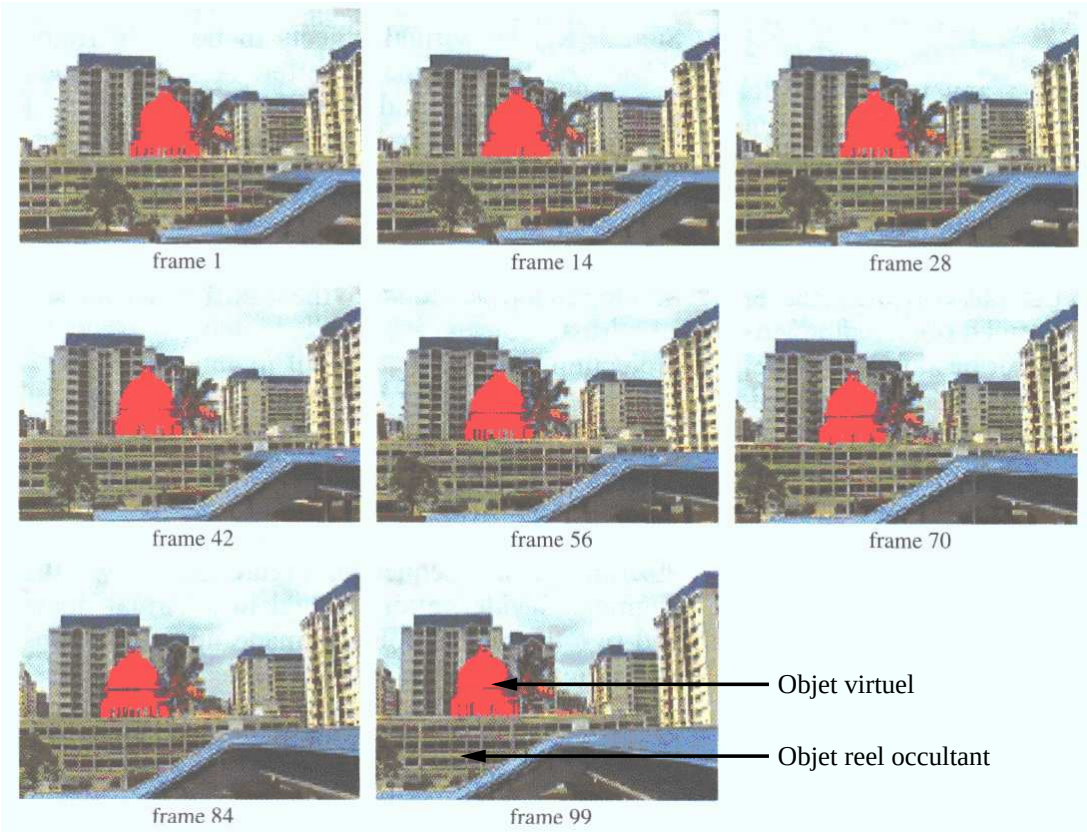



FIG. 5.5 – *Exemple d'incrustation extrait de [Ong et al.98].*

silhouettes détourées dans ces images-clé.

Retenons donc les points suivants :

- l'importance des images où l'aspect de l'objet occultant change, et l'intérêt de connaître de façon sure la silhouette de l'objet dans ces images;
- l'intérêt d'un modèle 3D pour déterminer la déformation 2D de la silhouette de l'objet occultant;
- le fait que ce modèle 3D n'a pas à être précis, nous cherchons essentiellement la silhouette de l'objet occultant. En particulier, un modèle 3D peut n'être valable qu'entre deux images-clé consécutives;
- la nécessité de tenir compte des erreurs des points de vue.

Dans le chapitre suivant, nous décrivons la méthode que nous proposons, qui tient compte de ces quatre remarques.

# Chapitre 6

# Gestion semi-automatique des occultations

Nous présentons dans ce chapitre notre méthode de gestion des occultations. Comme celle de [Ong et al.98], elle est basée sur un détourage préliminaire des objets occultants effectué par l'utilisateur, dans un nombre limité d'images de la séquence. Ce détourage va nous permettre de retrouver précisément les masques d'occultation sur l'ensemble de la séquence. Cet aspect non automatique de la méthode est justifié par le manque de précision des approches purement automatiques (voir chapitres 4 et 5), et par le fait que cette tâche est facile à réaliser. Néanmoins, elle pourrait devenir fastidieuse, et les différences de notre méthode avec celle de [Ong et al.98] permettent de minimiser le nombre de détourages manuels.

Nous avons en effet choisi de représenter l'objet occultant par un ensemble de contours 3D. L'idée générale de notre méthode est de réaliser une reconstruction locale du contour 3D occultant, reconstruction valable uniquement entre deux images-clé consécutives et qui nous permet de retrouver la silhouette de l'objet dans les images entre ces images-clé. Il est important de noter que l'estimation de l'erreur des points de vue décrite chapitre 3 est utilisée pour tenir compte de l'imprécision des points de vue utilisés lors de la reconstruction.

Nous présentons tout d'abord notre méthode sur un exemple simple. Les différentes étapes de la méthode sont détaillées dans les sections suivantes. Nous discutons également du cas où l'objet est composé au moins en partie de surfaces courbes, ce cas étant plus délicat pour la reconstruction qu'un objet composé uniquement d'arêtes vives. Nous donnons finalement les potentialités de la méthode, qui n'est en fait pas limitée à la gestion des occultations.

Les expérimentations de cette méthode seront présentées dans le chapitre suivant.

## 6.1 Description générale

### 6.1.1 Cas d'un objet ne présentant que des arêtes vives

L'idée générale est donc de reconstruire un contour 3D à partir de chaque paire de contours 2D détourés par l'utilisateur dans deux images-clé successives. La projection de ce contour 3D dans les images intermédiaires fournit une bonne prédiction de la silhouette des objets occultants dans ces images. Cette prédiction est ensuite corrigée par corrélation d'intensités. Cette étape de correction utilise l'estimation de l'erreur des points de vue décrits chapitre 3, puisqu'elle est rendue nécessaire en grande partie par l'imprécision des points de vue utilisés lors de la reconstruction et de la projection du contour 3D.

Nous allons présenter notre méthode sur l'exemple simple d'une scène composée d'un cube réel et d'un objet virtuel (figure 6.1.a), en montrant comment retrouver les images où le cube occulte l'objet virtuel, et les masques d'occultation pour ces images. Nous considérerons pour l'instant que l'objet occultant est toujours constitué, comme notre cube, d'arêtes vives.

1. **Choix des images-clé**
   L'utilisateur doit tout d'abord retenir certaines images de la séquence, appelés images-clé, pour lesquelles un changement d'aspect de l'objet occultant se produit. Pour notre exemple, l'image où disparaît une des faces du cube doit être retenue comme image-clé (voir figure 6.1.b). En vue de l'étape de reconstruction, la première et la dernière image doivent également être retenues, nous avons donc besoin ici de 3 images-clé $\mathrm{I}_{\text{clé-1}}$, $\mathrm{I}_{\text{clé-2}}$, et $\mathrm{I}_{\text{clé-3}}$.
2. **Détourage manuel de l'objet occultant dans les images-clé**
   L'utilisateur trace les contours 2D $\mathbf{c}_{\text{clé-}i}$ dans chaque image-clé $\mathrm{I}_{\text{clé-}i}$.
3. **Reconstruction des contours 3D**
   A partir de chaque paire de contours consécutifs $\mathbf{c}_{\text{clé-}i}$ et $\mathbf{c}_{\text{clé-}i+1}$, on reconstruit un contour 3D $\mathbf{C}_{i,i+1}$ par stéréoscopie. Si les images-clé ont été correctement choisies par l'utilisateur, ce contour 3D correspond aux frontières de l'objet occultant vues dans les images intermédiaires entre $\mathrm{I}_{\text{clé-}i}$ et $\mathrm{I}_{\text{clé-}i+1}$ (voir figures 6.2.a et 6.2.b).
4. **Projection des contours 3D dans les images intermédiaires (prédiction des silhouettes)**
   En reprojetant ce contour 3D $\mathbf{C}_{i,i+1}$ dans ces images intermédiaires, on obtient donc les silhouettes $\mathbf{c}_{\text{repr-}j}$ de l'objet occultant dans ces images (voir figures 6.3.a et 6.3.b). En pratique, en raison des imprécisions des points de vue, les contours $\mathbf{c}_{\text{repr-}j}$ ne correspondent pas exactement aux silhouettes. C'est pourquoi une étape de correction est nécessaire.
5. **Correction des projections**
   La reprojection $\mathbf{c}_{\text{repr-}j}$ du contour n'est donc qu'une prédiction, qui est donc corrigée pour obtenir le contour $\mathbf{c}_{\text{corr-}j}$, par corrélation d'intensités entre les régions intérieures à $\mathbf{c}_{\text{repr-}j}$ et au contour $\mathbf{c}_{\text{clé-}i}$ dans l'image-clé la plus proche de l'image numéro $j$. Grâce à l'estimation de l'incertitude des points de vue utilisés pour la reconstruction et la projection, nous déterminons une région de l'image où doit se trouver $\mathbf{c}_{\text{corr-}j}$, ce qui nous permet de contraindre la recherche de ce contour à cette région.
6. **Gestion des occultations**
   Nous disposons donc pour chaque image de la silhouette de l'objet réel susceptible d'occulter l'objet virtuel. En comparant la profondeur du contour 3D reprojeté dans l'image considérée et celle de l'objet virtuel, on peut déterminer si l'objet réel est effectivement devant l'objet virtuel. Bien sûr, l'utilisateur n'a considéré que les objets réels occultants; ce test est néanmoins intéressant quand plusieurs objets virtuels sont présents, et ne sont pas forcément tous occultés.
   Le cas échéant, il suffit de retrancher à l'image de l'objet virtuel la silhouette $\mathbf{c}_{\text{corr-}j}$ de l'objet réel.
   Cette comparaison des profondeurs est relativement grossière puisqu'on ne dispose pas du modèle complet de l'objet réel mais seulement de sa frontière 3D. Elle est cependant suffisante si la géométrie de l'objet réel n'est pas trop particulière et si les objets ne sont pas trop proches les uns des autres.

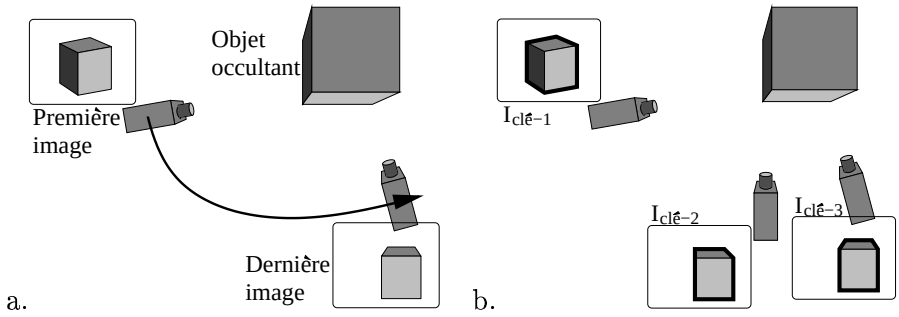



FIG. 6.1 – *a : Séquence d'exemple; b :* **Choix des images-clé** $I_{clé\text{-}1}$, $I_{clé\text{-}2}$ *et* $I_{clé\text{-}3}$, *et* **détourage manuel**.

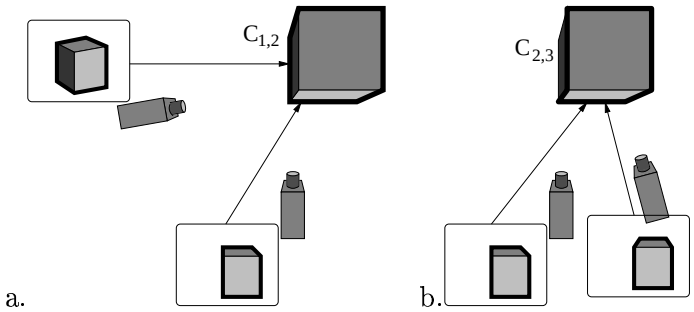



FIG. 6.2 – **Reconstruction des contours 3D** *a :* $\mathbf{C}_{1,2}$ *et b :*$\mathbf{C}_{2,3}$.

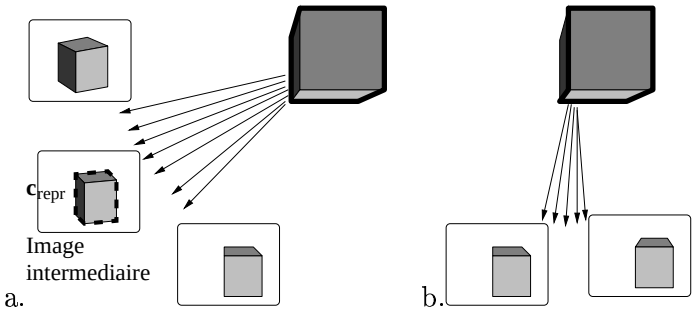



FIG. 6.3 – **Projection des contours 3D dans les images intermédiaires** *a. entre* $I_{clé\text{-}1}$ *et* $I_{clé\text{-}2}$ *et b. entre* $I_{clé\text{-}2}$ *et* $I_{clé\text{-}3}$.

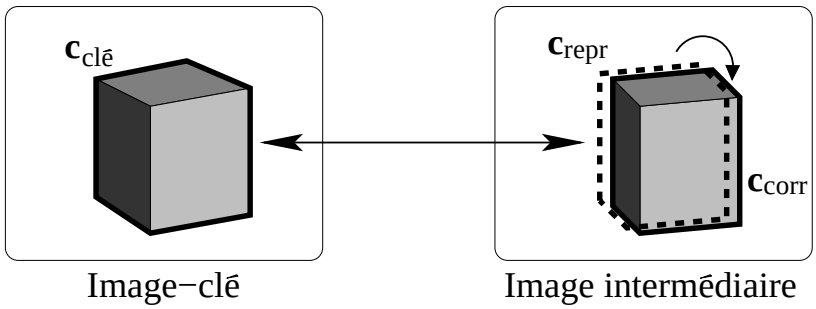



FIG. 6.4 – **Correction d'une projection**.

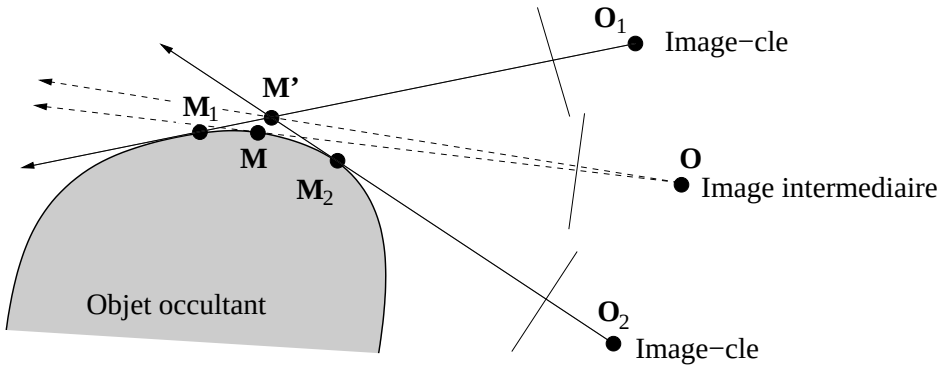



FIG. 6.5 – *Erreur produite par les contours apparents.*

### 6.1.2 Cas d'un objet présentant des surfaces courbes

Considérons maintenant un objet occultant non polyédrique. La silhouette de cet objet dans les images est alors composée, au moins en partie de contours apparents. Nous avons déjà signalé le fait (partie 4.1.3) qu'on ne peut pas reconstruire par stéréoscopie ces contours puisqu'ils ne correspondent pas à un contour physique de l'objet.

Mais considérons la figure 6.5 qui montre ce qui se passe quand on applique la méthode que nous venons de présenter à de tels contours. $\mathbf{O}_1$ et $\mathbf{O}_2$ sont les centres des caméras pour les images-clé, $\mathbf{O}$ le centre de la caméra pour une image intermédiaire $\mathrm{I}_{\mathrm{inter}}$. La position du contour de l'objet occultant dans $\mathrm{I}_{\mathrm{inter}}$ est la projection du point $\mathbf{M}$ dans cette image. Notre méthode utilise la projection de la reconstruction $\mathbf{M}'$ qui est faite en appariant les projections des points $\mathbf{M}_1$ et $\mathbf{M}_2$. Ces deux projections sont évidemment différentes mais intuitivement, on peut déjà s'apercevoir que si l'angle entre les droites $(\mathbf{O}_1\mathbf{M}_1)$ et $(\mathbf{O}_2\mathbf{M}_2)$ n'est pas trop grand et si la caméra pour l'image intermédiaire n'est pas trop proche de l'objet, ces deux projections ne sont pas très éloignées l'une de l'autre. On peut remarquer qu'appliquer notre méthode sur des contours apparents revient à approximer l'arc $\overset{\frown}{\mathbf{M}_1\mathbf{M}_2}$ par les segments $[\mathbf{M}_1\mathbf{M}]$ et $[\mathbf{M}\mathbf{M}_2]$.

Nous verrons dans la partie 6.8 une estimation quantitative de l'erreur engendrée par cette approximation. Si les positions des caméras pour les images-clé sont trop éloignées, nous pouvons ajouter une image-clé intermédiaire pour diminuer les erreurs de reconstruction et de projection. C'est pourquoi la méthode que nous venons d'exposer est souvent valable même en présence de contours apparents. Nous le verrons à travers différents exemples dans le chapitre suivant.

## 6.2 Reconstruction des contours 3D

Afin de calculer par triangulation un contour 3D, nous apparions tout d'abord les contours de la paire d'images-clé consécutives qui lui correspondent. Comme le contour reconstruit à partir de ces appariements est bruité, nous lui appliquons un filtre médian. Enfin, comme les appariements sont parfois ambigus, certaines parties du contour n'ont pas pu être reconstruites, et nous estimons leur position spatiale par interpolation.

### 6.2.1 Mise en correspondance

Considérons deux images-clé $\mathrm{I}_{\text{clé-1}}$ et $\mathrm{I}_{\text{clé-2}}$, dans lesquelles l'opérateur a détouré les contours $\mathbf{c}_{\text{clé-1}}$ et $\mathbf{c}_{\text{clé-2}}$, représentés par les chaînes de points $\{\mathbf{c}^i_{\text{clé-1}}\}$ et $\{\mathbf{c}^i_{\text{clé-2}}\}$. Afin de déterminer la

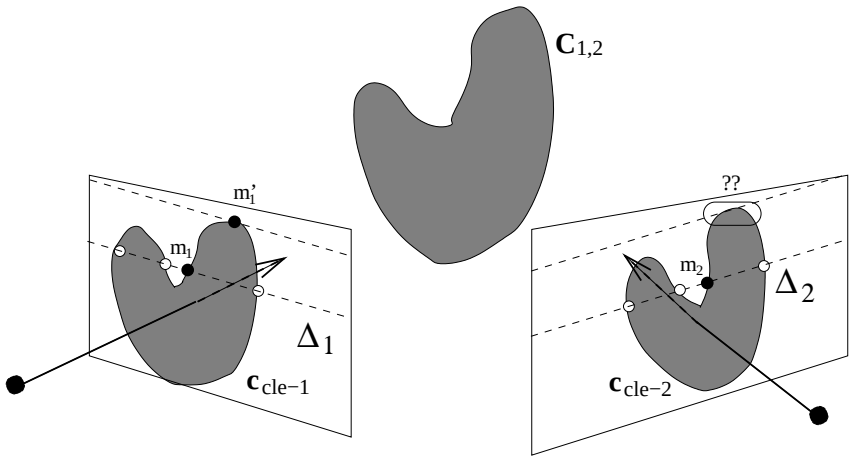



FIG. 6.6 – *Appariement des contours 2d.*

position 3D du contour $\mathbf{C}_{1,2}$ se reprojetant en $\mathbf{c}_{\text{clé-1}}$ et $\mathbf{c}_{\text{clé-2}}$, on commence par mettre en correspondance ces deux contours. Par exemple, on peut chercher pour chaque point $\mathbf{c}^i_{\text{clé-1}}$ son correspondant situé sur $\mathbf{c}_{\text{clé-2}}$ ; on peut évidemment inverser les rôles de $\mathbf{c}_{\text{clé-1}}$ et $\mathbf{c}_{\text{clé-2}}$.

Supposons donc qu'on cherche le correspondant $\mathbf{m}_2$ d'un point $\mathbf{m}_1$ de $\mathbf{c}_{\text{clé-1}}$ dans l'image $\mathrm{I}_{\text{clé-2}}$. Si l'opérateur a correctement détouré l'objet réel, $\mathbf{m}_2$ appartient à l'intersection de $\mathbf{c}_{\text{clé-2}}$ et de la droite épipolaire $\Delta_2$ associée à $\mathbf{m}_1$ dans l'image $\mathrm{I}_{\text{clé-2}}$ (voir figure 6.6); l'intersection est calculée en considérant que $\mathbf{c}_{\text{clé-2}}$ est composé de segments de droites $[\mathbf{c}^j_{\text{clé-2}};\mathbf{c}^{j+1}_{\text{clé-2}}]$. Comme $\mathbf{c}_{\text{clé-2}}$ est un contour fermé, l'intersection est généralement composée de deux points ou plus. Afin de lever cette ambiguïté sur le point $\mathbf{m}_2$, on utilise la contrainte d'ordre : si les images-clé sont correctement choisies, cette heuristique est respectée.

Considérons donc la droite épipolaire $\Delta_1$ passant par $\mathbf{m}_1$. $\Delta_1$ intersecte $\mathbf{c}_{\text{clé-1}}$ en $N_1$ points $\{\mathbf{n}^i_1\}$, $\Delta_2$ intersecte $\mathbf{c}_{\text{clé-2}}$ en $N_2$ points $\{\mathbf{n}^j_2\}$. Si $N_1 \neq N_2$, une erreur est survenue, due à l'imprécision de la géométrie épipolaire ou aux contours qui ont été mal détourés. Sinon, il existe un rang $i$ pour lequel $\mathbf{n}^i_1 = \mathbf{m}_1$, et le correspondant $\mathbf{m}_2$ est le point $\mathbf{n}^i_2$.

Un autre type d'erreur peut survenir quand les épipolaires sont proches d'une tangente aux contours : l'intersection épipolaire/contour risque d'être déterminée de façon imprécise. C'est le cas du point $\mathbf{m}'_1$ de la figure 6.6. C'est pourquoi on n'affectera pas de correspondant au point $m_1$ quand $\Delta_1$ est approximativement tangente à $\mathbf{c}_{\text{clé-1}}$, ou $\Delta_2$ à $\mathbf{c}_{\text{clé-2}}$.

A ce stade, on dispose des couples $(\mathbf{c}^i_{\text{clé-1}},\mathbf{c'}^i_{\text{clé-1}})$, avec $\mathbf{c'}^i_{\text{clé-1}} \in \mathbf{c}_{\text{clé-2}} \cup \{\omega\}$, $\omega$ désignant le fait que le correspondant n'a pas été trouvé. Les couples tels que $\mathbf{c'}^i_{\text{clé-1}} \neq \omega$ permettent de reconstruire un ensemble de points $\mathbf{M}^i$ appartenant au contour 3D $\mathbf{C}_{1,2}$. La figure 6.8 montre un tel contour 3D dans un cas pratique (la figure 6.7 montre les images-clé correspondantes, et les parties non appariées). Il présente un certain nombre d'artefacts, provenant d'erreurs d'appariement. C'est pourquoi nous appliquons un filtre permettant de limiter l'influence de ces erreurs d'appariement.

### 6.2.2 Filtre médian

Au vu de la nature des erreurs, c'est-à-dire des « pics » dans le contour reconstruit, nous avons retenu le filtre médian [Ataman et al.81], qui permet de réduire le bruit impulsionnel. Dans le cas où l'ensemble des valeurs possibles d'une donnée $X$ à filtrer est fini $(x_0,\ldots,x_{2n})$, et où la distribution de probabilité des $x_i$ est uniforme, la valeur donnée par le filtre médian est la

a.  b. 

FIG. 6.7 – *a et b : Deux images-clé de la séquence Stanislas, avec en noir, les points non appariés.*

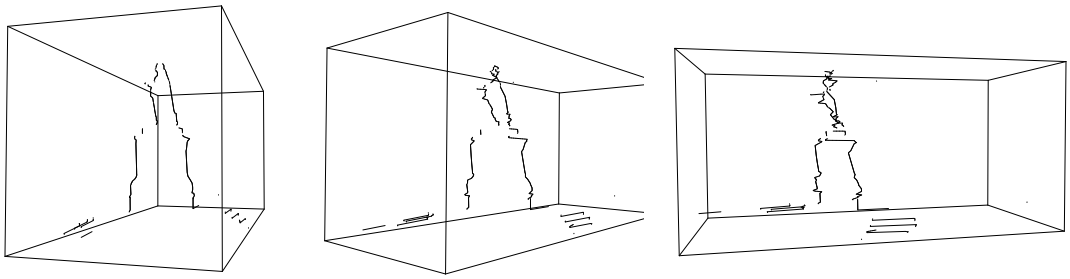


FIG. 6.8 – *Reconstruction du contour détouré dans les images-clé de la figure 6.7.*

médiane des $x_i$, c'est-à-dire la valeur $x_{\sigma(n)}$ avec $\sigma$ telle que $x_{\sigma(0)} \leq \ldots \leq x_{\sigma(2n)}$ :

$$\mathcal{F}_{\mathcal{M}}(X) = \text{médiane}(x_0, \ldots, x_{2n}) = x_{\sigma(n)}$$

Le choix des données à filtrer est important; en particulier, il faut veiller à ce que le contour 3D après le passage du filtre se reprojette encore en les contours détourés dans les images-clé. C'est pourquoi nous avons choisi d'appliquer le filtre sur la *profondeur* des points du contour 3D. Comme la profondeur d'un point est définie en fonction d'une image, nous reconstruisons deux contours (notés $\mathbf{C}_{1,2}$ et $\mathbf{C}_{2,1}$) au lieu d'un seul. Le premier sera reprojeté dans les images intermédiaires plus proches de la première image-clé, le deuxième dans les images intermédiaires plus proches de la deuxième image-clé.

Le contour $\mathbf{C}_{1,2}$ est défini en fonction de $\mathrm{I}_{\text{clé-1}}$. Pour chaque point $\mathbf{M}^i$ de $\mathbf{C}_{1,2}$, nous calculons :

$$\text{prof}\,'_1(\mathbf{M}^i) = \text{médiane}\left(\text{prof}_1(\mathbf{M}^{i-h}), \ldots, \text{prof}_1(\mathbf{M}^{i+h})\right),$$

où $h$ est la taille du filtre médian et $\text{prof}_2(\mathbf{M}^j)$ la profondeur de $\mathbf{M}^j$ dans l'image $\mathrm{I}_{\text{clé-1}}$. $\mathbf{M}^i$ se reprojette en $\mathbf{c}^i_{\text{clé-1}}$, le filtre médian remplace donc ce point par le point 3D de profondeur $\text{prof}\,'_1(\mathbf{M}^i)$ et se projetant également en $\mathbf{c}^i_{\text{clé-1}}$. Le contour $\mathbf{C}_{2,1}$ est défini pareillement, en fonction de $\mathrm{I}_{\text{clé-2}}$.

La figure 6.9 montre le résultat du filtre sur le contour déjà présenté figure 6.8.

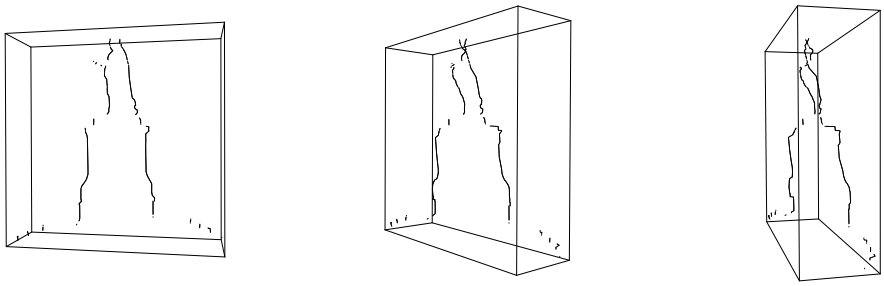


FIG. 6.9 – *Contour de la figure 6.8 après filtrage.*

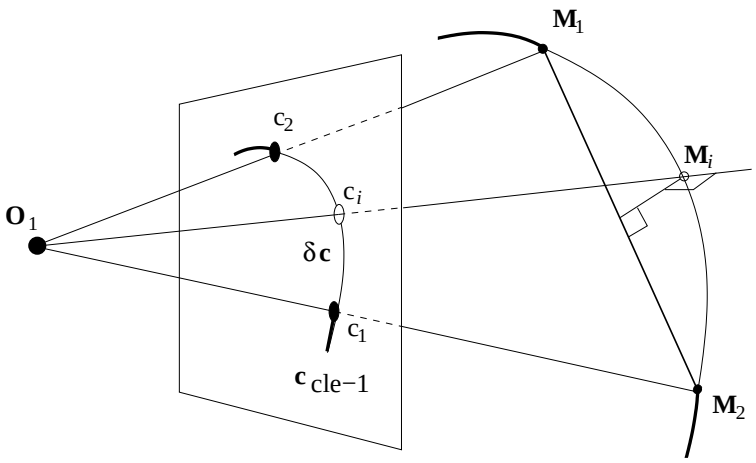


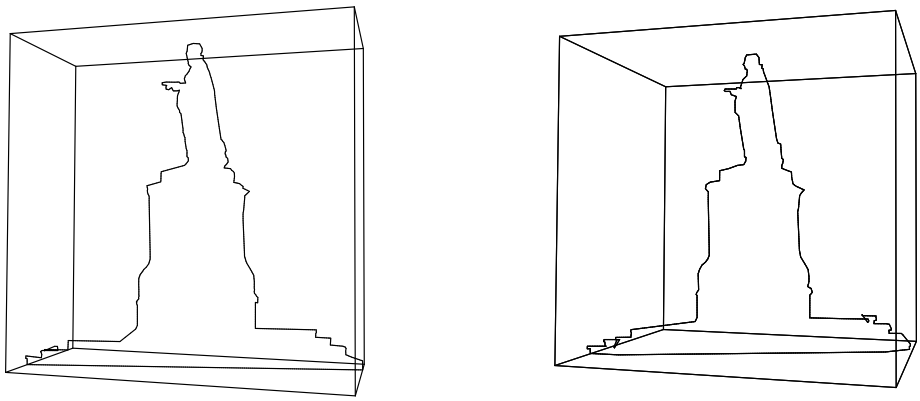

FIG. 6.10 – *Estimation des parties non reconstruites du contour 3D.*

### 6.2.3 Interpolation des parties non reconstruites

Il reste à estimer la position spatiale des parties de $\mathbf{C}_{1,2}$ et $\mathbf{C}_{2,1}$ qui n'ont pas pu être reconstruites, faute d'appariement.

Soit $\delta_{\mathbf{c}}$ un ensemble de points contigus de $\mathbf{c}_{\text{clé-1}}$ pour lesquels on n'a pas pu trouver de correspondant, et $\mathbf{c}_1$ et $\mathbf{c}_2$ ses extrémités déjà reconstruites en $\mathbf{M}_1$ et $\mathbf{M}_2$. Estimer la courbe 3D de $\mathbf{C}_{1,2}$ correspondante à $\delta_{\mathbf{c}}$ par le segment $[\mathbf{M}_1;\mathbf{M}_2]$ serait maladroit, puisque la reprojection de ce segment dans $\mathrm{I}_{\text{clé-1}}$ ne serait pas exactement $\delta_{\mathbf{c}}$. La figure 6.10 montre une meilleure interpolation : le correspondant 3D $\mathbf{M}_i$ d'un point $\mathbf{c}_i$ sur $\delta_{\mathbf{c}}$ est estimé par le point le plus proche du segment $[\mathbf{M}_1;\mathbf{M}_2]$ qui appartient à la droite $[\mathbf{O}_1\mathbf{c}_i)$ (avec $\mathbf{O}_1$ le centre de la caméra pour l'image $\mathrm{I}_{\text{clé-1}}$). La partie 3D estimée ainsi se reprojette exactement sur $\delta_{\mathbf{c}}$.

La figure 6.11 montre le contour $\mathbf{C}_{1,2}$ résultat de cette interpolation sur le contour déjà présenté figure 6.9, ainsi que le contour $\mathbf{C}_{2,1}$.



a. b.

FIG. 6.11 – *a : Contour final* $\mathbf{C}_{1,2}$*; b : contour final* $\mathbf{C}_{2,1}$.

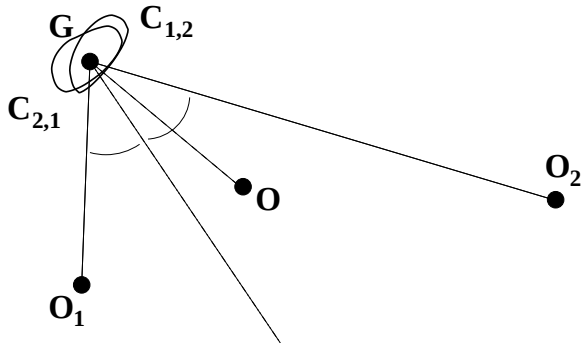



FIG. 6.12 – *Choix du contour à reprojeter en fonction de l'image intermédiaire.*

### 6.2.4 Choix du contour à reprojeter

Les contours ainsi reconstruits peuvent être maintenant reprojetés dans les images intermédiaires :

$$\mathbf{c}_{\text{repr}} = \text{Proj}(\mathbf{C}).$$

Il reste à choisir, de $\mathbf{C}_{1,2}$ ou de $\mathbf{C}_{2,1}$, quel contour reprojeter. Nous retenons le contour associé à l'image-clé la plus proche de l'image intermédiaire, en terme de direction par rapport à l'objet. Plus exactement :

$$\mathbf{C} = \begin{cases} \mathbf{C}_{1,2} \text{ si } \widehat{\mathbf{O}_1\mathbf{G}\mathbf{O}} < \widehat{\mathbf{O}\mathbf{G}\mathbf{O}_2} \\ \mathbf{C}_{2,1} \text{ sinon} \end{cases}$$

où $\mathbf{O}$ est le centre de la caméra pour l'image intermédiaire, et $\mathbf{O}_1$ et $\mathbf{O}_1$ sont les centres de la caméra pour les images-clé. $\mathbf{G}$ est le centre de gravité de l'objet occultant, estimé à partir de l'un des deux contours 3D. La figure 6.12 illustre ce choix.

Pour illustrer l'intérêt de reconstruire deux contours 3D plutôt qu'un seul, nous avons reprojeté le contour $\mathbf{C}_{1,2}$ dans l'image-clé $\text{I}_{\text{clé-2}}$ et le contour $\mathbf{C}_{2,1}$ dans l'image-clé $\text{I}_{\text{clé-1}}$, figure 6.13. Ces reprojections sont éloignées de la position attendue, et elles seraient une mauvaise initialisation de l'étape suivante, consistant à déformer le contour reprojeté selon un modèle de mouvement

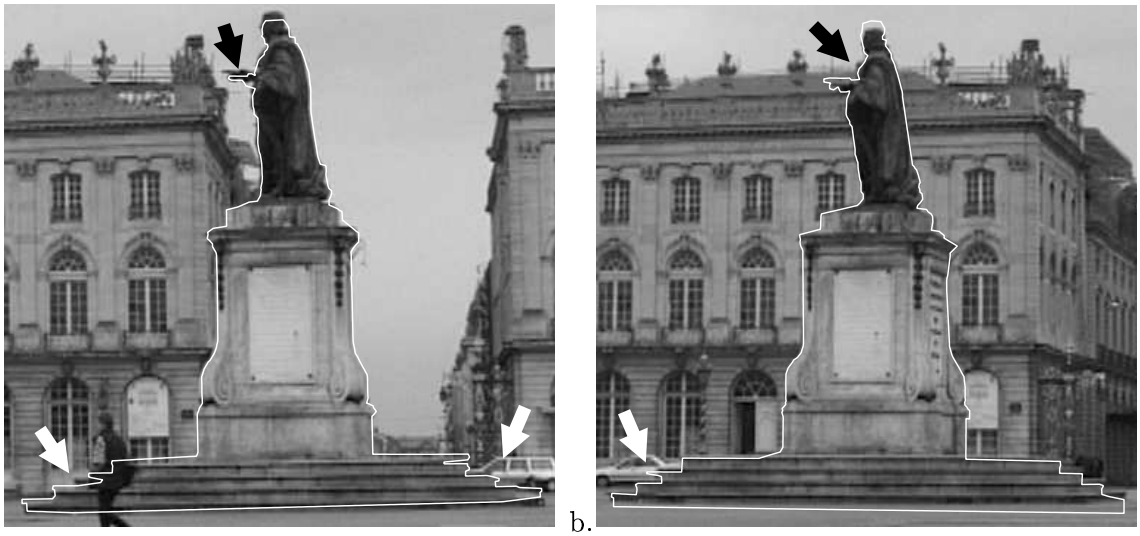


FIG. 6.13 – *Illustration de l'intérêt des deux contours. a : Reprojection de* $\mathbf{C}_{2,1}$ *dans la première image-clé; b : reprojection de* $\mathbf{C}_{2,1}$ *dans la deuxième image-clé. Les flèches indiquent les défauts majeurs de la prédiction.*

global pour retrouver la position attendue. On pourra comparer ces reprojections avec celle de la figure 6.14, qui respecte elle le choix du contour 3D que l'on vient de présenter.

## 6.3 Détermination des masques d'occultation

### 6.3.1 Nécessité de l'étape de correction

La projection $\mathbf{c}_{\text{repr}}$ du contour 3D dans les images intermédiaires fournit une bonne estimation de la silhouette de l'objet occultant. En pratique, cette estimation n'est généralement pas exacte (voir par exemple la figure 6.14), pour deux raisons :

- l'imprécision des points de vue utilisés pour la reconstruction et la projection ;
- le fait que les contours utilisés pour la reconstruction aient pu être des contours apparents.

### 6.3.2 Correction par modèle de mouvement

C'est pourquoi la projection ne constitue qu'une prédiction, et une étape de correction est nécessaire. Cette étape est similaire au suivi basé région par corrélation, déjà présenté partie 5.2 : elle consiste à estimer la transformation globale entre la projection et la position effective de la silhouette dans l'image considérée. Imposer un modèle de mouvement global permet de stabiliser et d'améliorer l'estimation de la correction à effectuer.

Nous recherchons donc la transformation $D$ telle que :

$$\mathbf{c}_{\text{corr}} = D(\mathbf{c}_{\text{repr}})$$

### 6.3.3 Choix du modèle de mouvement

Plusieurs modèles de mouvement sont utilisés par la communauté vision, allant d'un déplacement (dépendant de 3 paramètres, translation et rotation) à une transformation homographique

FIG. 6.14 – *Image 118 : un exemple de reprojection; une étape de correction est nécessaire pour retrouver la position exacte.*

(8 paramètres). La transformation affine est le modèle le plus couramment utilisé : elle est suffisamment générale pour cette étape de correction sans dépendre de trop de paramètres. Plusieurs auteurs ont constaté que les paramètres supplémentaires dont dépend le modèle homographique sont souvent mal estimés [Meyer et al.92, Bonnaud et al.94].

Nous avons donc retenu pour $D$ une transformation affine. Nous avons constaté expérimentalement, comme nous le verrons, qu'une telle transformation est effectivement bien adaptée à notre problème. La transformation d'un point $\mathbf{m}$ par $D$ est alors :

$$D(\mathbf{m}) = \begin{pmatrix} D_1 & D_2 \\ D_4 & D_5 \end{pmatrix} \begin{pmatrix} \mathbf{m}_u \\ \mathbf{m}_v \end{pmatrix} + \begin{pmatrix} D_3 \\ D_4 \end{pmatrix} = \begin{pmatrix} D_1 & D_2 & D_3 \\ D_4 & D_5 & D_6 \end{pmatrix} \begin{pmatrix} \mathbf{m}_u \\ \mathbf{m}_v \\ 1 \end{pmatrix}$$

### 6.3.4 Critère à minimiser

Suivant le schéma du suivi basé région, la recherche de $D$ est effectuée en cherchant à minimiser un critère mesurant la corrélation d'intensités entre les régions intérieures à $\mathbf{c}_{\text{repr}}$ et $D(\mathbf{c}_{\text{repr}})$. Or, notre représentation de l'objet par un simple contour 3D ne nous permet de prédire que le contour 2D, et non pas directement l'apparence de l'objet.

On peut néanmoins utiliser le contour reprojeté $\mathbf{c}_{\text{repr}}$ pour prédire indirectement cette apparence dans l'image intermédiaire considérée. Pour chaque point $\mathbf{c}^k_{\text{repr}}$ du contour reprojeté, nous estimons la déformation locale entre le contour $\mathbf{c}_{\text{repr}}$ et $\mathbf{c}_{\text{clé}}$, le contour détouré par l'utilisateur dans une des deux images-clé utilisées pour calculer $\mathbf{c}_{\text{repr}}$. Le choix de cette image-clé est réalisé de la même façon que pour le choix du contour 3D. Cette déformation est ici encore modélisée

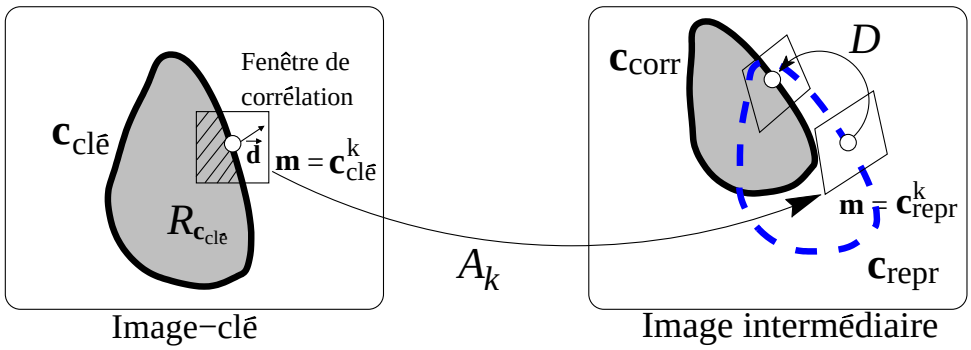



FIG. 6.15 – *Correction de la prédiction par corrélation entre l'image-clé et l'image intermédiaire.*

par un modèle affine $A_k$, estimé aux moindres carrés :

$$\min_{A_k} \sum_{j=-\delta k}^{j=+\delta k} \left(\mathbf{c}^{k+j}_{\text{repr}} - A_k(\mathbf{c}^{k+j}_{\text{clé}})\right)^2$$

$\delta k$ doit être choisi suffisamment grand, nous utilisons la valeur de $\delta k$ telle que $2\delta k + 1$ vaille la moitié du nombre de points de $\mathbf{c}_{\text{clé}}$. La déformation $A_k$ est ensuite appliquée localement sur la texture autour du point $\mathbf{m} = \mathbf{c}^k_{\text{clé}}$ pour prédire localement la texture autour du point $\mathbf{m}' = \mathbf{c}^k_{\text{repr}}$ :

$$\mathrm{I}^k_{\text{prédite}}\left(\mathbf{m}'_u + \mathbf{d}_u, \mathbf{m}'_v + \mathbf{d}_u\right) = \mathrm{I}_{\text{clé}}\big((\mathbf{m}_u, \mathbf{m}_v) + A_k(\mathbf{d}_u, \mathbf{d}_v)\big)$$

Le calcul de la corrélation n'est pas effectué sur la totalité des surfaces intérieures aux contours mais limité à une bande intérieure aux contours, ce qui permet une meilleure localisation de $\mathbf{c}_{\text{corr}}$. $D$ est supposé minimiser la somme :

$$\Psi(D) = \sum_k \psi_D(k)$$

$\psi_D(k)$ étant une mesure de corrélation entre les images $\mathrm{I}^k_{\text{prédite}}$ et $\mathrm{I}_{\text{inter}}$, sur une fenêtre centrée sur le point $\mathbf{c}^k_{\text{repr}}$ et une fenêtre centrée sur le point $D(\mathbf{c}^k_{\text{repr}})$, déformée suivant $D$ :

$$\psi_D(k) = \frac{1}{N_k} \sum_{\substack{\mathbf{d}_u, \mathbf{d}_v = +T \\ \mathbf{d}_u, \mathbf{d}_v = -T \\ (\mathbf{c}^k_{\text{repr}} + \vec{\mathbf{d}}) \in R_{\mathbf{c}_{\text{repr}}}}} \left(\mathrm{I}^k_{\text{prédite}}(\mathbf{c}^k_{\text{repr}} + \vec{\mathbf{d}}) - \mathrm{I}_{\text{inter}}\left(D(\mathbf{c}^k_{\text{repr}} + \vec{\mathbf{d}})\right)\right)^2 \qquad (6.1)$$

où $\vec{\mathbf{d}} = (\mathbf{d}_u, \mathbf{d}_v)$, $2T + 1$ est la taille des fenêtres de corrélation (typiquement $T = 7$ pixels) et $R_{\mathbf{c}_{\text{repr}}}$ est la région intérieure à $\mathbf{c}_{\text{repr}}$. La portion de la fenêtre recouverte par $R_{\mathbf{c}_{\text{repr}}}$ n'est pas de la même taille pour tous les points $\mathbf{c}^k_{\text{repr}}$. Le calcul de la corrélation est donc pondéré par le facteur $\frac{1}{N_k}$, avec $N_k$ le nombre de pixels de la fenêtre de corrélation appartenant à $R_{\mathbf{c}_{\text{repr}}}$ : on évite ainsi de donner artificiellement une plus grande importance aux points pour lesquels cette portion est grande.

Dans notre implantation, $\mathrm{I}^k_{\text{prédite}}$ n'est pas calculée, nous composons directement $A_k$ et $D$ pour calculer la corrélation entre l'image-clé et l'image intermédiaire (voir figure 6.15) :

$$\psi_D(k) = \frac{1}{N_k} \sum_{\substack{\mathbf{d}_u, \mathbf{d}_v = +T \\ \mathbf{d}_u, \mathbf{d}_v = -T \\ (\mathbf{c}^k_{\text{clé}} + \vec{\mathbf{d}}) \in R_{\mathbf{c}_{\text{clé}}}}} \left(\mathrm{I}_{\text{clé}}(\mathbf{c}^k_{\text{clé}} + \vec{\mathbf{d}}) - \mathrm{I}_{\text{inter}}(D(\mathbf{c}^k_{\text{repr}} + A_k(\vec{\mathbf{d}})))\right)^2$$

FIG. 6.16 – *Image 118 : en pointillés, la prédiction par reprojection; en trait plein, la correction sans la contrainte des régions* $\Lambda_i$.

ce qui est sensiblement équivalent à l'expression 6.1 ($N_k$ est ici le nombre de pixels de la fenêtre de corrélation appartenant à $R_{\mathbf{c}_{\text{clé}}}$).

### 6.3.5 Minimisation du critère

La minimisation d'une fonction à plusieurs paramètres partant d'une estimée initiale $D_0$ est généralement décomposée en un ensemble de minimisations à une dimension effectuées successivement selon plusieurs directions. Par exemple, la méthode du gradient conjugué utilisée par [Bascle et al.94] effectue à l'étape $i+1$ une minimisation dans la direction $-\nabla\Psi(D_i)$. Nous avons préféré utiliser la méthode de convergence quadratique proposée par Powell et son implantation par [Press et al.88], pour laquelle la direction utilisée à l'étape $i+1$ est $D_i - D_{i-m}$, où $m$ est le nombre de paramètres de la fonction. Cette méthode évite d'avoir à calculer le gradient $\nabla\Psi$, et l'implantation de Press et al. a un risque moindre de converger vers un minimum local.

Nous avons tout d'abord fixé $D_0$ à la transformation Identité :

$$D_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

et estimé le minimum $D$ à l'aide de la méthode de Powell. Les résultats sont tout à fait satisfaisants quand la prédiction $\mathbf{c}_{\text{repr}}$ est suffisamment proche de la silhouette de l'objet dans l'image considéré (voir un exemple figure 6.16).

Des problèmes se sont posés quand la prédiction est trop éloignée de la position attendue, comme par exemple la figure 6.17.b : le résultat de la minimisation par une telle méthode n'est

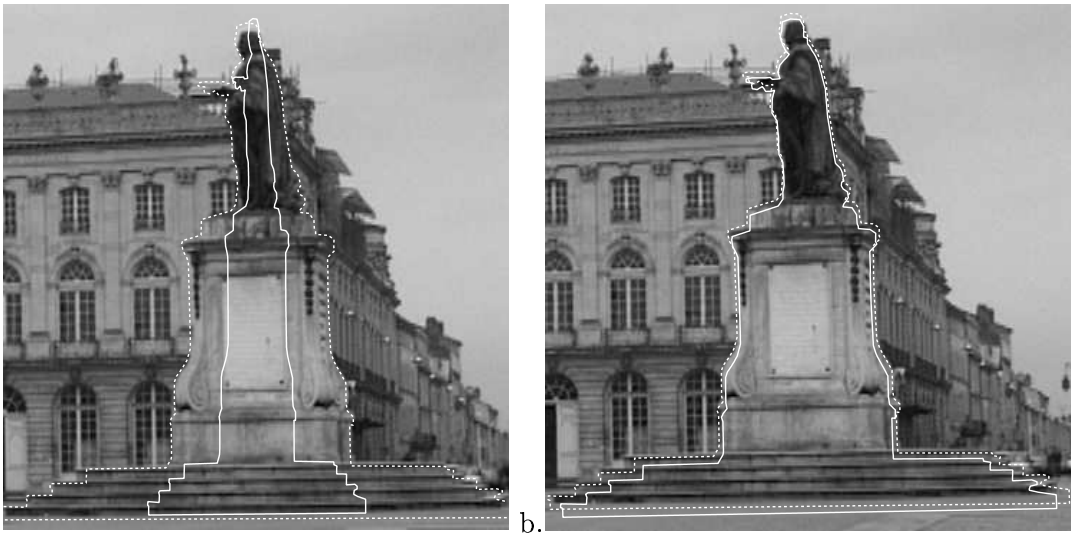


FIG. 6.17 – *Images 98 (a) et 99 (b) : en pointillés, la prédiction par reprojection; en trait plein, la correction sans la contrainte des régions* $\Lambda_i$.

qu'un minimum local. Les minima locaux surviennent notamment quand l'objet occultant présente des motifs répétitifs, comme les marches sous la statue de cet exemple. La corrélation entre la première marche et la deuxième, par exemple, est en effet très bonne (les marches ont toutes la même apparence), ce qui crée un minimum local de la fonction de corrélation $\Psi$. Un autre cas de correction erronée est présenté figure 6.17.a : en l'absence de contrainte, la correction par corrélation peut donner un résultat très éloigné de la prédiction, et de la position attendue.

Une recherche semi-exhaustive consisterait à discrétiser l'espace de recherche

$$D_{ijklmn} = \begin{pmatrix} 1 + id_1 & 0 + jd_2 & 0 + kd_3 \\ 0 + ld_4 & 1 + md_5 & 0 + nd_6 \end{pmatrix}$$

en fixant les pas $d_1$, $d_2$, $d_3$, $d_4$, $d_5$ et $d_6$, et à déterminer le paramètre $D_{ijklmn}$ qui fournit la valeur minimale de $\Psi$ quand on fait varier les entiers $i$, $j$, $k$, $l$, $m$ et $n$ sur un intervalle. Ce paramètre $D_{ijklmn}$ pourrait être ensuite utilisé pour initialiser $D_0$. Malheureusement, une telle recherche est particulièrement coûteuse en temps de calcul puisque l'espace de recherche possède 6 dimensions. On pourrait utiliser un algorithme de recherche stochastique, de type tabou ou recuit simulé. Ils restent cependant coûteux en temps de calcul, sans garantir de retrouver le minimum global.

Un compromis peut être alors d'effectuer un certain nombre de tirages aléatoires de valeurs de $D$ autour de l'identité. Le tirage qui obtient le score de corrélation minimum est utilisé pour initialiser la minimisation numérique. Cette solution permet d'améliorer les résultats à moindre coût, sans éliminer toutes les erreurs de convergence.

Il faut se rappeler que l'étape de correction est nécessaire à cause des erreurs des points de vue utilisés pour la reconstruction et la projection du contour 3D. Nous disposons d'une estimation de ces erreurs (voir chapitre 3), et nous allons montrer comment utiliser cette estimation pour estimer l'erreur de reprojection du contour. Cette nouvelle estimation va nous fournir pour chaque point du contour reprojeté une région de l'image contenant la position réelle de ce point. Ainsi, nous pouvons limiter l'espace de recherche et diminuer le risque de retrouver un minimum local.

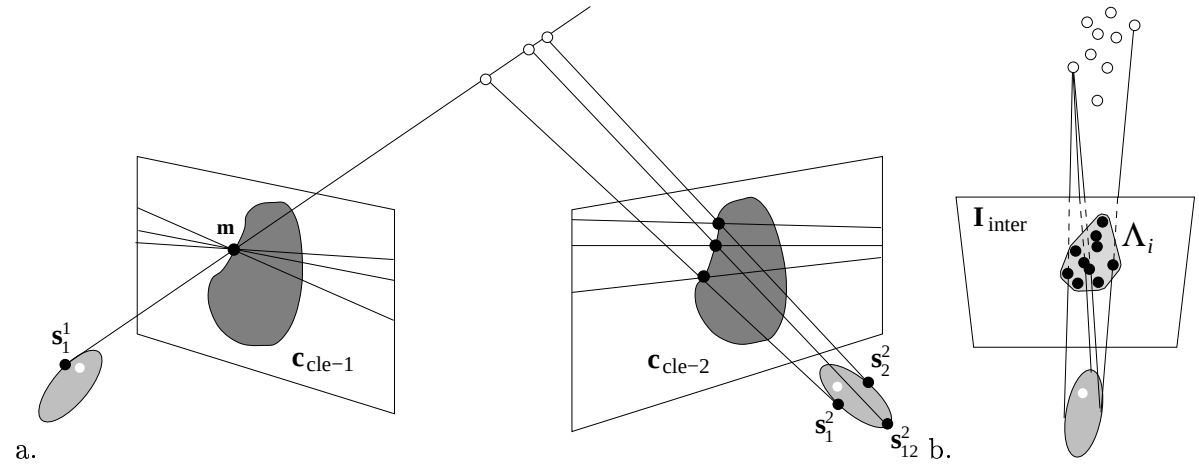



FIG. 6.18 – *a. Estimation de l'erreur de reconstruction; b. estimation de l'erreur de reprojection.*

## 6.4 Estimation et prise en compte de l'erreur du contour reprojeté

Nous allons tout d'abord, pour chaque point du contour, estimer l'erreur de reconstruction. De cette erreur de reconstruction, nous déduirons l'erreur de reprojection pour chacun de ces points.

### 6.4.1 Erreur de reconstruction

Dans le cas d'une reconstruction à partir d'appariements de points 2D, [Ayache88], par exemple, a montré que l'on pouvait estimer l'erreur de reconstruction quand on connaît l'incertitude des points de vue utilisés, grâce à un filtre de Kalman.

Dans notre cas, malheureusement, nous ne disposons que d'un appariement entre contours, et la mise en correspondance des points des contours dépend des points de vue puisqu'elle est effectuée en calculant l'intersection des droites épipolaires avec l'un des deux contours. Ces contours sont définis par un ensemble de points reliés par des segments pour permettre une forme très générale, ce qui nous empêche de calculer analytiquement l'erreur de reconstruction.

Nous avons donc recours à une approche exhaustive. Nous considérons les 12 sommets des ellipsoïdes d'incertitude de chacun des deux points de vue utilisés pour la reconstruction, que l'on notera $\{\mathbf{s}_1^1,\mathbf{s}_2^1,\ldots,\mathbf{s}_{12}^1\}$ et $\{\mathbf{s}_1^2,\mathbf{s}_2^2,\ldots,\mathbf{s}_{12}^2\}$. Soit $\mathbf{m}$ un point appartenant au contour $\mathbf{c}_{\text{clé-1}}$ d'une des deux images-clé. A partir d'un sommet $\mathbf{s}_i^1$ de cette image-clé, on peut trouver 12 reconstructions possibles de $\mathbf{c}$ avec les 12 sommets $\mathbf{s}^2$ de l'autre image-clé (voir figure 6.18.a). En considérant ainsi les 12 sommets $\mathbf{s}_i^1$, on obtient alors $12^2$ reconstructions extrémales de $\mathbf{m}$. L'enveloppe convexe de ces 144 points est une bonne approximation de l'erreur de reconstruction de $\mathbf{m}$, c'est-à-dire que le correspondant 3D $\mathbf{M}$ de $\mathbf{m}$ est situé dans cette enveloppe convexe.

### 6.4.2 Erreur de reprojection

Nous pouvons maintenant chercher l'incertitude de la projection du contour 3D, en tenant compte de l'incertitude du point de vue de l'image intermédiaire, et l'erreur de reconstruction qui vient d'être calculée. Nous utilisons pour cela la même approche que précédemment : pour chaque point $\mathbf{m}$ de $\mathbf{c}_{\text{clé-1}}$, les $12^2$ reconstructions sont projetées dans l'image intermédiaire selon les 12

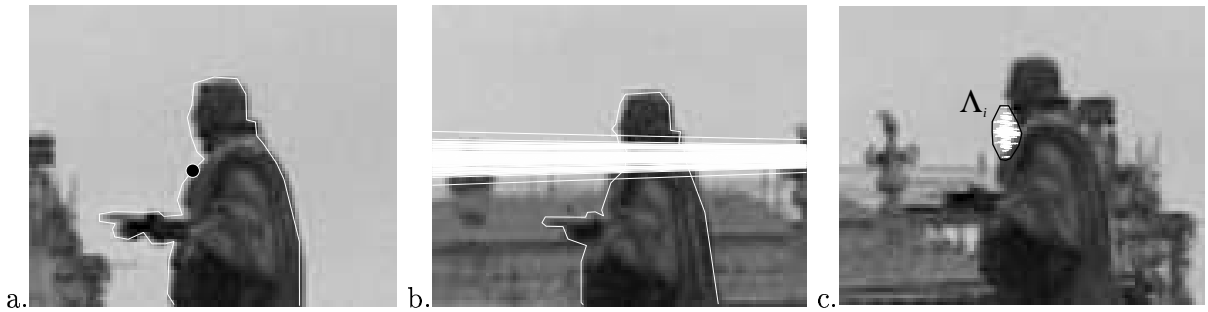



FIG. 6.19 – *a : Un point du contour dans une image-clé; b : les droites épipolaires associées à ce point dans l'autre image-clé; c : les reprojections associées à ce point dans une image intermédiaire.*

sommets de l'ellipsoïde d'incertitude estimé pour le point de vue de cette image. L'enveloppe convexe des $12^3$ points 2D qui résultent de cette étape définit une région de l'image qui contient la projection du correspondant 3D $\mathbf{M}$ de $\mathbf{m}$, c'est-à-dire le correspondant 2D du point $\mathbf{m}$ dans l'image intermédiaire (voir figure 6.18.b). La figure 6.19 montre les droites épipolaires et les reprojections dans un cas concret.

A chaque point $\mathbf{c}^i_{\text{repr}}$ du contour reprojeté dans l'image $\mathrm{I}_{\text{inter}}$, on associe donc l'enveloppe convexe des $12^3$ reprojections qui lui correspondent, que l'on notera $\Lambda_i$. Ainsi, la correction $D(\mathbf{c}^i_{\text{repr}})$ d'un point du contour reprojeté doit appartenir à la région $\Lambda_i$, ce qui permet de contraindre la recherche de $D$.

### 6.4.3 Contraindre la recherche du minimum par les régions $\Lambda_i$

En pratique, comme le calcul des régions $\Lambda_i$ est relativement coûteux, il n'est effectué que sur un sous-ensemble des points du contour, régulièrement répartis, typiquement 2% du nombre total.

Les tirages aléatoires effectués pour l'initialisation de la minimisation numérique ne sont alors considérés que si les points corrigés appartiennent bien à leur région $\Lambda_i$.

Imposer des contraintes à la minimisation dans le cas non linéaire est un problème difficile. Pour pouvoir imposer à la minimisation numérique de rechercher le minimum dans l'espace restreint par les régions $\Lambda_i$, nous avons modifié le critère de corrélation par $\Psi'(D)$ :

$$\Psi'(D) = \sum_k \psi'_D(k)$$

avec

$$\psi'_D(k) = \begin{cases} \psi_D(k) & \text{si } D(m_k) \in \Lambda_k \\ \Omega & \text{sinon.} \end{cases}$$

$\Omega$ est un terme de pénalité constant, très supérieur à l'ordre de valeur l'ancien terme de corrélation $\Psi(D)$. Ainsi, une transformation $D$ pour laquelle la correction $D(\mathbf{m})$ d'un point $\mathbf{m}$ sort de sa région $\Lambda$ ne peut pas être un minimum de $\Psi'$.

Les figures 6.29 et 6.28 comparent les résultats de la correction selon que les régions $\Lambda_i$ sont utilisées ou non pour contraindre la recherche du minimum. Elles montrent clairement l'intérêt de cette contrainte, rendue possible par l'estimation de l'erreur des points de vue, que l'on a pu propager pour évaluer l'erreur de reconstruction et de reprojection.

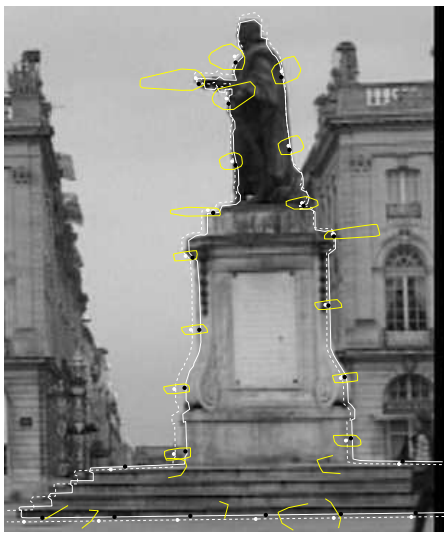


FIG. 6.20 – *Exemple de correction pour un objet partiellement visible.*

## 6.5 Entrée ou sortie de l'objet occultant

Jusqu'ici, nous n'avons considéré l'étape de correction que dans le cas où l'objet occultant était entièrement visible dans l'image. Pour pouvoir traiter des séquences où cet objet n'est que partiellement visible dans certaines images, le critère de corrélation a été modifié et vaut finalement :

$$\Psi''(D) = \begin{cases} \frac{1}{N}\sum_k \psi'_D(k) & \text{si } N \neq 0 \\ \Omega' & \text{sinon} \end{cases}$$

où $N$ est le nombre de points du contour $\mathbf{c}_{\text{repr}}$ transformé par $D$ visibles dans l'image, et $\Omega'$ un terme suffisamment grand pour éviter que la minimisation ne renvoie un contour entièrement hors de l'écran (on peut prendre $\Omega' = n\Omega$, avec $n$ le nombre de points composant le contour $\mathbf{c}_{\text{clé}}$). La figure 6.20 montre un exemple de prédiction pour un objet partiellement visible, et la correction obtenue.

## 6.6 Discussion sur l'affinement par un contour actif

L'étape de correction telle qu'elle vient d'être présentée consiste à corriger *globalement* le contour reprojeté. La dernière étape des méthodes de suivi consiste généralement en une correction locale. Cette dernière étape est nécessaire pour le suivi puisque le modèle de mouvement global ne peut à lui seul permettre de modéliser le changement de perspective sur un grand nombre d'images (voir partie 5.2).

Dans notre cas, le contour reprojeté tient déjà compte du changement de perspective. Néanmoins, il serait *a priori* intéressant d'avoir également une étape de correction locale, permettant d'affiner la position du contour, notamment quand il s'agit d'un contour apparent.

Nous avons déjà évoqué les difficultés à concevoir un ajustement local dans un contexte général. Notamment les contours actifs sont encore délicats à utiliser quand le contour recherché peut être un contour faible, alors qu'il peut être environné par des contours forts (partie 5.2.2).

Pour illustrer ces difficultés, nous avons testé l'utilisation d'un contour actif après l'étape de correction globale (voir un exemple figure 6.21). Celui-ci permet d'améliorer la position du

a.  b. 

FIG. 6.21 – *a. Un exemple de contour corrigé, utilisé pour initialiser un contour actif; b. contour actif après convergence.*

contour quand la position attendue correspond à un gradient fort de l'image (voir par exemple la tête et le bras de la statue). Dans le cas contraire, le contour actif peut être attiré par un gradient ne correspondant pas à la position attendue. De plus, il fait disparaître certains détails du contour initial (par exemple la main de la statue), là où la courbure du contour initial est forte.

Étant donné le contexte très général d'application de notre méthode, nous ne pouvons faire d'hypothèses sur la correspondance entre le contour de l'objet et un gradient fort. Dans de telles conditions, le contour actif dégrade plus le résultat de la correction globale qu'il ne l'améliore. C'est pourquoi nous n'avons pas conservé cette dernière étape.

## 6.7 Choix des images-clé

### 6.7.1 Graphe d'aspects

Le choix des images-clé est fortement lié au graphe d'aspects [Plantinga et al.90] de l'objet occultant, tout au moins dans le cas d'un objet polyédrique. Le graphe d'aspects est une représentation qualitative d'un objet qui énumère toutes ses apparences topologiquement distinctes. Il partitionne l'espace 3D en un ensemble de régions maximales $\mathcal{R}_i$, où tous les points voient les mêmes indices de l'objet, et chaque nœud du graphe d'aspects représente une région maximale, les arcs du graphe reliant les régions voisines. La figure 6.22 montre ces régions maximales définies par le cube de notre exemple.

### 6.7.2 Choix des images-clé

Ainsi, pour retrouver la silhouette d'un objet dans une image intermédiaire située dans une des régions $\mathcal{R}_i$, il faut disposer du contour 3D correspondant à cette région. La reconstruction de ce contour nécessite deux images-clé appartenant à $\mathcal{R}_i$. Donc, chaque région $\mathcal{R}_i$ doit contenir au moins deux images-clé. De plus, des images-clé consécutives $I_{\text{clé-1}}$ et $I_{\text{clé-2}}$ doivent appartenir à la même région pour que les contours détourés manuellement dans ces images correspondent

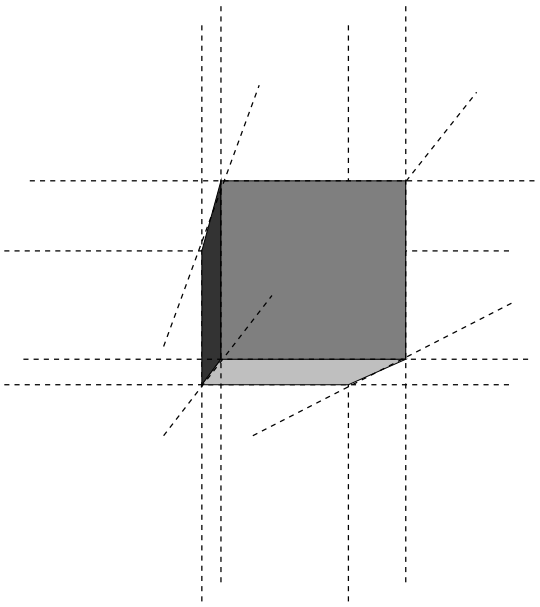


FIG. 6.22 – *Les 26 régions maximales appartenant au graphe d'aspects d'un cube.*

au même contour physique. C'est pourquoi une image-clé doit être retenue quand la caméra traverse la frontière entre deux régions maximales, comme c'est le cas de l'image-clé $I_{clé\text{-}2}$ de notre exemple.

Nous verrons dans les expérimentations qu'il faut parfois plus de deux images-clé pour une région quand l'erreur sur les points de vue est trop importante, mais cet ajout reste exceptionnel.

Dans le cas d'un objet composé de surfaces courbes, la notion de graphe d'aspects est moins pertinente pour notre problème. Par exemple, le graphe d'aspects d'une sphère ne présente qu'une seule région maximale. Il faut alors disposer les images-clé suffisamment régulièrement pour que l'approximation due aux contours apparents n'engendre pas d'erreur trop importante. Nous allons maintenant discuter plus précisément du cas de ces contours.

## 6.8 Cas des contours apparents

Nous allons maintenant donner une estimation quantitative de l'erreur de projection quand des contours apparents sont utilisés pour reconstruire le contour 3D.

### 6.8.1 « Reconstruction » d'un contour apparent

Considérons deux caméras de centres respectifs $\mathbf{O}_1$ et $\mathbf{O}_2$, et plaçons-nous dans un plan épipolaire (voir figure 6.23). Pour des raisons de simplicité, nous supposerons que les points de vue sont connus exactement, et que la section de l'objet occultant par ce plan est un cercle (centré en $\mathbf{O}$ et de rayon $R$).

Soit le point $\mathbf{I}_1$ tel que $(\mathbf{OI}_1)$ soit perpendiculaire à $(\mathbf{OI}_1)$. L'image du point $\mathbf{I}_1$ dans la caméra 1 correspond à un contour apparent. On définit de la même façon le point $\mathbf{I}_2$ pour la caméra 2. En prenant pour simplifier $\mathbf{OO}_1 = \mathbf{OO}_2 = D$, d'où $\delta_1 = \delta_2 = \delta$, et en choisissant correctement

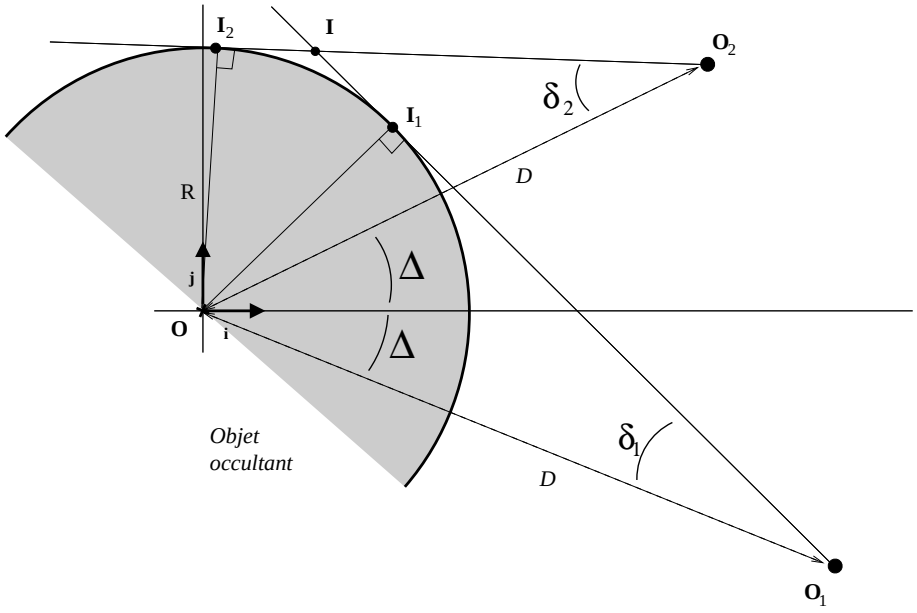



FIG. 6.23 – *Construction du point $I$.*

le repère (**Oij**) :

$$\mathbf{O}_1 = D\begin{pmatrix}\cos(-\Delta)\\ \sin(-\Delta)\end{pmatrix}, \mathbf{O}_2 = D\begin{pmatrix}\cos\Delta\\ \sin\Delta\end{pmatrix}, \mathbf{I}_1 = R\begin{pmatrix}\sin(\delta+\Delta)\\ \cos(\delta+\Delta)\end{pmatrix}, \mathbf{I}_2 = R\begin{pmatrix}\sin(\delta-\Delta)\\ \cos(\delta-\Delta)\end{pmatrix},$$

où $\Delta = \frac{\widehat{\mathbf{O}_1\mathbf{O}\mathbf{O}_2}}{2} \in ]0;\frac{\pi}{2}[$.

La mise en correspondance a apparié les projections des points $\mathbf{I}_1$ et $\mathbf{I}_2$, et le point 3D du contour est reconstruit à l'intersection des droites $(\mathbf{O}_1\mathbf{I}_1)$ et $(\mathbf{O}_2\mathbf{I}_2)$, notée $\mathbf{I}$. La droite $(\mathbf{O}_1\mathbf{I}_1)$ a pour équation $\sin(\delta+\Delta)x + \cos(\delta+\Delta)y = R$, la droite $(\mathbf{O}_2\mathbf{I}_2)$ $\sin(\delta-\Delta)x + \cos(\delta-\Delta)y = R$, d'où :

$$\mathbf{I} = \frac{R}{\sin(\delta+\Delta)\cos(\delta-\Delta) - \sin(\delta-\Delta)\cos(\delta+\Delta)}\begin{pmatrix}\cos(\delta-\Delta) - \cos(\delta+\Delta)\\ \sin(\delta+\Delta) - \sin(\delta-\Delta)\end{pmatrix}$$

qu'on simplifie aisément en :

$$\mathbf{I} = \frac{2R\sin\Delta}{\sin 2\Delta}\begin{pmatrix}\sin\delta\\ \cos\delta\end{pmatrix}.$$

En supposant que $R \ll D$, c'est-à-dire que les caméras restent éloignées de l'objet par rapport à son rayon :

$$\mathbf{I} \simeq \frac{2R\sin\Delta}{\sin 2\Delta}\begin{pmatrix}\frac{R}{D}\\ 1\end{pmatrix}.$$

### 6.8.2 Erreur de reprojection

Supposons maintenant qu'on ajoute une troisième caméra, et qu'on recherche la reprojection du contour apparent dans cette nouvelle image. Dans quelle mesure la reprojection de $\mathbf{I}$ dans cette image est-elle une bonne prédiction ?

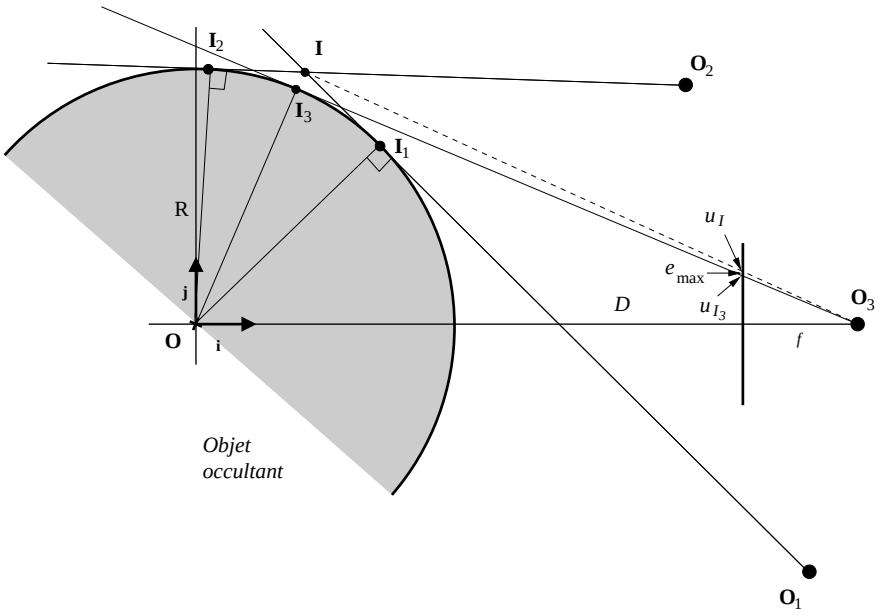



FIG. 6.24 – *Erreur engendrée par un contour apparent.*

Soit $\mathbf{O}_3 = (D\cos\alpha, D\sin\alpha)$ le centre de cette nouvelle caméra et $f$ sa distance focale ($\alpha$ est l'angle entre $[\mathbf{OO}_3)$ et $[\mathbf{Oi})$). La position réelle du contour dans cette caméra est la projection du point $\mathbf{I}_3$, avec $\mathbf{I}_3$ tel que $(\mathbf{OI}_3)$ soit perpendiculaire à $(\mathbf{O}_3\mathbf{I}_3)$. Toujours si $R \ll D$, pour des raisons de symétrie, l'erreur de prédiction maximale est obtenue pour $\alpha = 0$ (voir figure 6.24).

Le point $\mathbf{I}_3$ a alors pour coordonnées : $\mathbf{I}_3 = (-R\sin(\alpha-\delta), R\cos(\alpha-\delta))$.

$$\mathbf{I}_3 = R\begin{pmatrix} \sin(\delta) \\ \cos(\delta) \end{pmatrix} \simeq R\begin{pmatrix} \frac{R}{D} \\ 1 \end{pmatrix},$$

et la matrice de projection de la nouvelle caméra est :

$$\mathbf{P} = A[\mathbf{R}| - \mathbf{R}^T\mathbf{t}] = \begin{pmatrix} 0 & f \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & -D \\ 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{P} = \begin{pmatrix} 0 & f & 0 \\ 1 & 0 & -D \end{pmatrix}$$

$\mathbf{I}_3$ se projette donc en $u_{\mathbf{I}_3}$, avec :

$$u_{\mathbf{I}_3} = \frac{(\mathbf{PI}_3)_1}{(\mathbf{PI}_3)_2} = \frac{fRD}{R^2 - D^2} \simeq -\frac{fR}{D},$$

car on suppose $R \ll D$. La prédiction de la position du contour occultant est la projection du point $\mathbf{I}$ dans cette caméra, c'est-à-dire le point d'abscisse $u_{\mathbf{I}}$, avec :

$$u_{\mathbf{I}} = \frac{(\mathbf{PI})_1}{(\mathbf{PI})_2} = \frac{2fRD\sin\Delta}{\sin 2\Delta(R - D^2)} \simeq -\frac{2fR\sin\Delta}{D\sin 2\Delta}.$$

Soit $e_{\max} = |u_{\mathbf{I}} - u_{\mathbf{I}3}|$, l'erreur maximale de reprojection. On a :

$$e_{\max} \simeq \frac{Rf}{D}\left|1 - \frac{2\sin\Delta}{\sin 2\Delta}\right|. \tag{6.2}$$

FIG. 6.25 – *La balise a un rayon de 12 cm et est à une distance de 5m.*

| | 10 deg | 20 deg | 30 deg | 40 deg |
|---|---|---|---|---|
| $\frac{1}{50}$ | 0.3 | 1.3 | 3.1 | 6.1 |
| $\frac{1}{20}$ | 0.8 | 3.2 | 7.7 | 15.3 |
| $\frac{1}{10}$ | 1.5 | 6.4 | 15.5 | 30.5 |

FIG. 6.26 – *Valeurs de $e_{max}$ en pixels pour différentes valeurs de $\frac{R}{D}$ et de $\Delta$ ($f = 1000$).*

Pour se représenter ce que peut valoir le rapport de $D$ et $R$ dans un cas pratique, on peut se rapporter à l'image 6.25 : la balise lumineuse dans cette image a un rayon $R = 12$ centimètres, et sa distance à la caméra vaut approximativement $D \simeq 5$ mètres (la distance focale $f$ a une valeur approximative de 1000). Le rapport $\frac{R}{D}$ vaut alors à peu près $\frac{1}{50}$.

De plus, le tableau 6.26 présente les valeurs de $e_{\max}$ en pixels pour différentes valeurs du rapport $\frac{R}{D}$ et de l'angle $\Delta$, toujours pour une distance focale de 1000. Les valeurs de $e_{\max}$ proches du pixel sont acceptables. On peut espérer raisonnablement que les valeurs de l'ordre de la dizaine de pixels peuvent être diminuées par l'étape de correction et atteindre alors une valeur acceptable. Au delà, l'erreur maximale devient très importante. Elle peut être réduite en ajoutant une image-clé, et nous étudions dans la partie suivante l'évolution de cette erreur après cet ajout.

### 6.8.3 Réduction de l'erreur en ajoutant une image intermédiaire

Si on ajoute une troisième image de l'objet pour améliorer la reconstruction, l'arc $\widehat{\mathbf{I}_1\mathbf{I}_2}$ est alors approximé par les 3 segments $[\mathbf{I}_1\mathbf{J}']$, $[\mathbf{J}'\mathbf{J}]$ et $[\mathbf{J}\mathbf{I}_2]$ (voir figure 6.27).

On peut se rendre compte sur la figure qu'ajouter une vue intermédiaire permet de diminuer significativement l'erreur de reconstruction. Plus formellement, on peut évaluer l'influence d'un tel ajout sur l'erreur de reprojection maximale, calculée plus haut. Le minimum du rapport de l'ancienne erreur maximale sur la nouvelle erreur maximale $\frac{e_{max}(\Delta)}{e_{max}(\Delta/2)}$ est atteint quand $\Delta$ tend vers 0, et vaut 4. Ajouter une vue intermédiaire permet donc de diviser l'erreur maximale de reprojection par plus de 4.

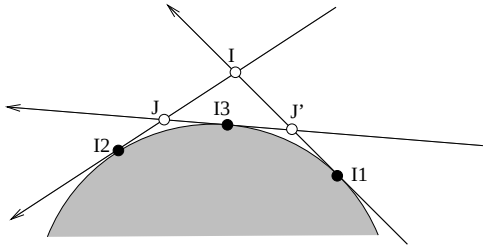



FIG. 6.27 – *Comparaison de l'erreur de reconstruction d'un contour apparent pour 2 et 3 images-clé.*

### 6.8.4 Conclusion sur l'erreur due aux contours apparents

On voit donc que l'erreur due aux contours apparents reste raisonnable quand l'objet occultant et les caméras sont suffisamment éloignés, et souvent très inférieure à l'erreur due à l'imprécision des points de vue qui peut atteindre une dizaine de pixels. L'étape de correction doit alors permettre de corriger cette erreur.

Si cette erreur devient trop importante et ne permet pas d'obtenir un bon résultat même après correction, on peut décider d'ajouter des images-clé, en sachant qu'ajouter une image-clé permet de diviser l'erreur par plus de 4.

Bien sûr, la méthode peut échouer si les caméras sont trop proches de l'objet. La formule 6.2 ou le tableau 6.26 peuvent aider à estimer par avance l'erreur due aux contours apparents.

Après cette description de notre méthode, nous présentons, dans le chapitre suivant les expérimentations effectuées, selon différentes trajectoires de caméra et de types d'objet occultant. Nous discuterons alors, au vu des résultats obtenus, de l'adéquation de cette méthode avec les objectifs décrits dans le chapitre 1.

On pourra constater que l'outil développé ici ne se limite pas à la gestion des occultations. En fournissant la silhouette d'un objet dans un ensemble d'images, il peut être utilisé comme outil de segmentation temporelle d'une séquence vidéo. Nous présenterons donc également dans le chapitre suivant des exemples d'une telle application de cet outil.

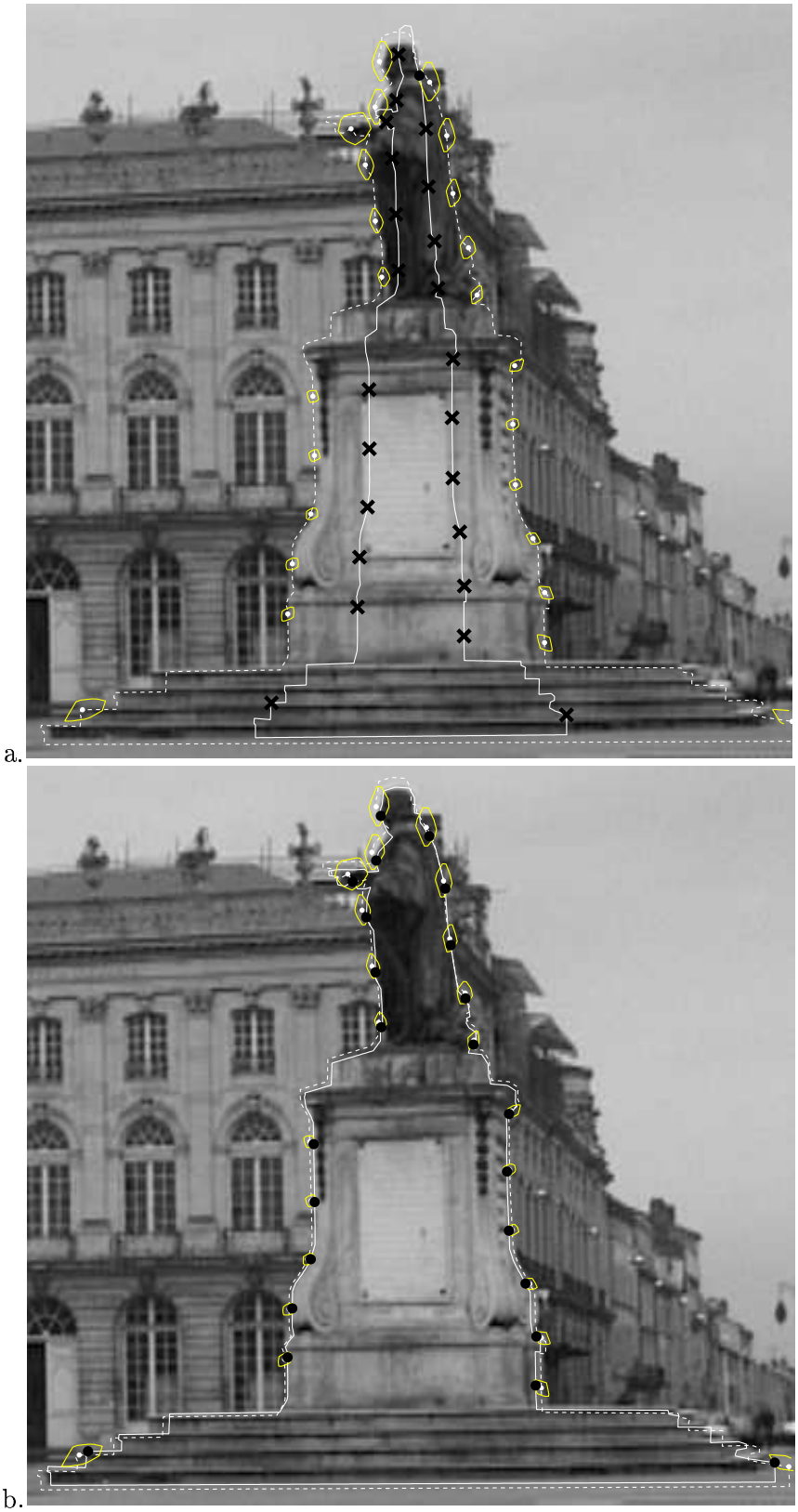


FIG. 6.28 – *Image 98 : résultat de la correction a : sans et b : avec la contrainte des régions $\Lambda_i$. Le contour reprojeté est en pointillés, la correction en trait plein. Les points corrigés qui sont restés dans leur région $\Lambda_i$ sont représentés par un point noir, ceux qui sont en dehors par une croix noire.*

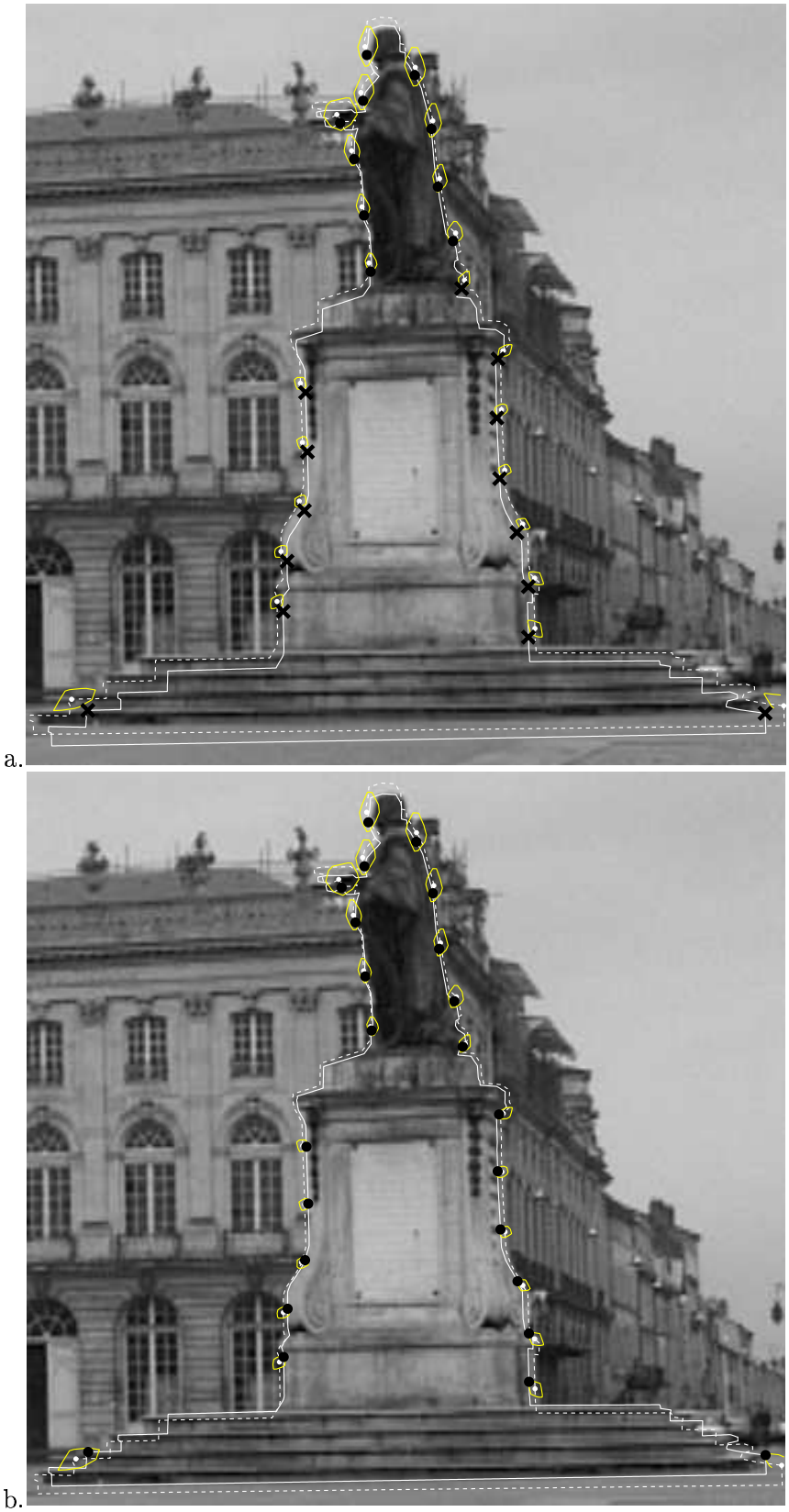


FIG. 6.29 – *Image 99 : résultat de la correction a : sans et b : avec la contrainte des régions $\Lambda_i$. Le contour reprojeté est en pointillés, la correction en trait plein. Les points corrigés qui sont restés dans leur région $\Lambda_i$ sont représentés par un point noir, ceux qui sont en dehors par une croix noire.*

# Chapitre 7

# Gestion semi-automatique des occultations: expérimentations

Dans ce chapitre, nous présentons les expérimentations de notre méthode décrite au chapitre précédent. Elles ont été réalisées en considérant différents types de séquences: nous avons fait varier la trajectoire de la caméra (certaines trajectoires étant plus propices à la reconstruction), la qualité de l'estimation des points de vue ainsi que les caractéristiques des objets occultants (présence ou non de contours apparents, complexité du graphe d'aspects) et la nature des images (prises en milieu extérieur ou intérieur, texturées ou faiblement texturées). Nous montrons que cette méthode est également utilisable pour un objet mobile et rigide. Ces expérimentations nous permettront de valider les critères que nous avions fixés au chapitre 1 : développer un système général, précis, robuste, avec une interactivité réduite et intuitive.

Nous commençons par présenter l'outil développé pour l'aide au détourage manuel qui doit être effectué dans les images-clé, permettant de faciliter et d'accélérer la tâche de l'utilisateur. Nous donnons pour chaque expérimentation une évaluation du temps consacré à ce détourage, ainsi que le temps de calcul du détourage automatique dans les images intermédiaires (pour une station Solaris Ultra 5, cadencée à 233MHz).

Les séquences présentant les scènes augmentées sont disponibles au format MPEG sur notre site Internet, à l'adresse :

`http://www.loria.fr/~lepetit/Occlusions`

## 7.1 Outil semi-automatique de détourage dans une image

Même si le détourage manuel n'est effectué que dans un petit nombre d'images, il est intéressant pour l'utilisateur de disposer d'un outil interactif pouvant permettre un détourage précis, sans être laborieux. Plusieurs outils sont envisageables : les contours actifs et les *Intelligent Scissors* (« ciseaux intelligents »).

### 7.1.1 Contours actifs

Une méthode souvent utilisée pour définir manuellement et facilement un contour est l'utilisation d'un contour actif [Kass et al.88] : un tracé manuel relativement grossier sert à initialiser un contour actif permettant après convergence d'améliorer le tracé. Rappelons qu'un contour actif converge vers un minimum d'énergie, énergie définie par un terme interne (comme l'intégrale de

la courbure) permettant de lisser le contour, et un terme externe basé par exemple sur la magnitude du gradient. Cependant, nous avons déjà vu que le résultat obtenu n'est pas forcément la position attendue (voir figure 6.21) : les endroits à forte courbure ont pu être lissés et le contour actif a pu être attiré par un fort gradient. En cas d'erreur, le contour doit être réinitialisé ou édité manuellement.

### 7.1.2 *Intelligent Scissors*

Nous avons opté pour un outil appelé *Intelligent Scissors* [Mortensen et al.95, Mortensen et al.00], permettant d'obtenir un détourage plus précis au prix d'une interactivité plus forte. Décrivons tout d'abord son principe: l'utilisateur clique tout d'abord à l'aide de la souris sur un point du contour à tracer, point appelé graine (*seed point*). Quand il déplace le pointeur de la souris, l'interface graphique propose un chemin (déterminé à partir du gradient de l'image, voir plus bas) reliant la graine à la position actuelle du pointeur de souris. Plus la distance entre ces points est grande, plus le chemin proposé risque d'être incorrect. Quand le chemin proposé correspond bien à une portion du contour que veut tracer l'utilisateur, celui-ci clique à nouveau pour retenir le chemin proposé, et la position de la souris devient la nouvelle graine. Si aucun des chemins ne convient (le contour de l'objet ne correspond pas toujours à un contour physique de l'image), l'utilisateur peut choisir de tracer un segment de droite entre la graine et la position courante de la souris. Le procédé est itéré jusqu'à ce que l'objet soit entièrement détouré.

### 7.1.3 Calcul du chemin proposé

Quand l'utilisateur choisit un point graine, un chemin optimal est calculé pour chaque pixel de l'image (éventuellement une portion de l'image pour un temps de calcul compatible avec l'interactivité), chemin reliant le pixel considéré au point graine. [Mortensen et al.95] définissent un chemin optimal comme le chemin minimisant la somme du Laplacien le long du chemin. Nous avons préféré utiliser le gradient de l'image, le calcul du gradient donnant un résultat moins bruité que celui du Laplacien. Nous minimisons donc plutôt la somme des valeurs :

$$\frac{1}{1 + \|\overrightarrow{\text{grad}}(u,v)\|}$$

le long du chemin. Les chemins optimaux sont calculés efficacement grâce à un algorithme de type $\mathcal{A}^*$.

### 7.1.4 Résultats

La figure 7.1 montre deux résultats d'utilisation de l'outil *Intelligent scissors*. Les contours de la statue (figure 7.1.a) étant très marqués, ils sont retrouvés correctement par l'outil; seul un détail de la main a été perdu. Le contour des marches n'est lui pas présent dans l'image, et le tracé proposé est systématiquement attiré par un mauvais contour. La vache jouet de la séquence du chalet pose le même type de problème, les tâches noires et blanches attirant le tracé proposé à l'intérieur de la silhouette.

Ces exemples montrent l'intérêt et les limites de cet outil : quand les contours de l'image le permettent, il peut fournir rapidement un détourage précis. Dans le cas contraire, il faut se contenter d'un ensemble de segments de droite. Enfin, si l'on souhaite un détourage très précis, il faut pouvoir (comme nous le ferons par la suite) éditer le contour obtenu à l'aide d'un outil plus classique (déplacement, suppression et ajout de points).

a.  b. 

FIG. 7.1 – *Deux résultats d'utilisation de l'outil* Intelligent scissors *: les points correspondent aux clics souris, les traits pleins aux contours retrouvés automatiquement et les traits pointillés aux segments de droite définis quand la méthode automatique a échoué.*

## 7.2 Séquence Stanislas

Cette séquence a été filmée d'une voiture en mouvement autour de la place Stanislas à Nancy. L'objet occultant sera ici la statue située au milieu de la place. Les difficultés présentées par cette séquence sont :

- la scène présente de nombreux éléments mobiles (voitures, piétons) ;
- la séquence est relativement saccadée ;
- l'objet occultant et le fond sont de couleurs assez semblables, notamment le piédestal de la statue et les bâtiments : les contours de l'objet occultant sont donc très peu marqués.

### 7.2.1 Statue seule

Commençons par nous limiter à la statue seule, comme dans l'exemple de suivi partie 5.2. Elle change peu d'apparence le long de la séquence, seule une partie du bras disparaît progressivement. Nous n'avons donc tout d'abord retenu que deux images-clé, présentées figure 7.2 : définir l'image 50 comme image-clé permet un détourage manuel très rapide grâce à l'outil *Intelligent scissors* puisque les contours de la statue sont très marqués dans cette image. La deuxième image-clé est la dernière image de la séquence. Les points de vue utilisés sont ceux obtenus à l'aide de la méthode basée modèle. Les résultats obtenus sont corrects, sauf pour la main de la statue (voir images 99 et 127). On notera que les images 15 et 43 ne sont pas situés entre les images-clé. Les temps de réalisation sont présentés tableau 7.4.

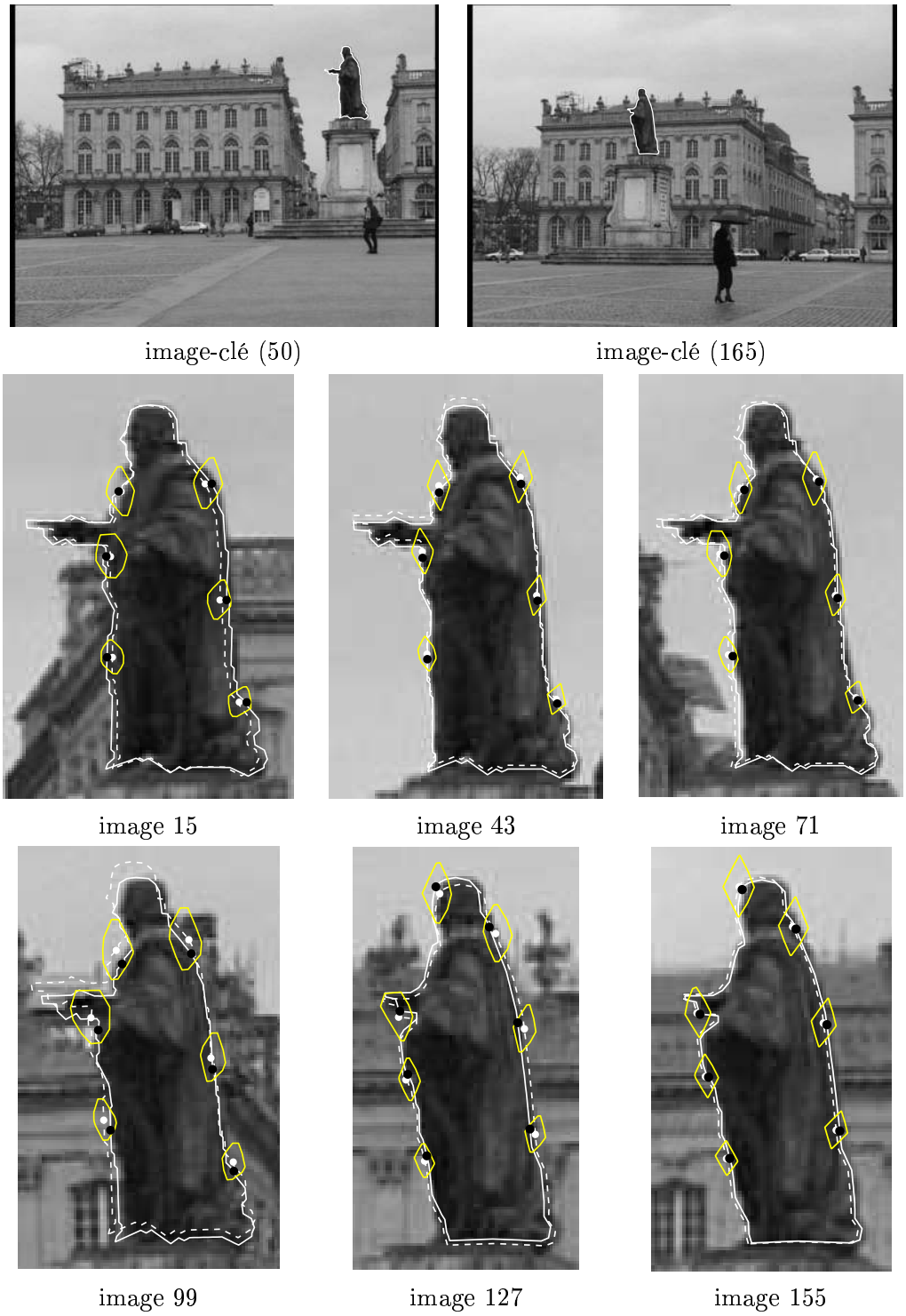



FIG. 7.2 – *Résultats sur la séquence Stanislas, statue seule, 2 images-clé. Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

image-clé supplémentaire (110)

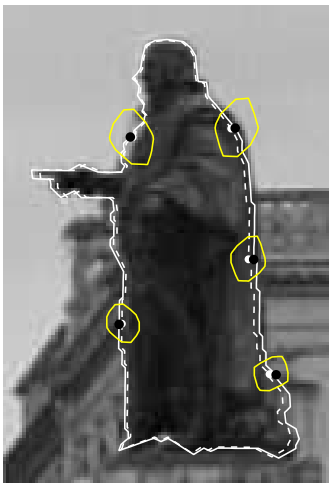


image 15

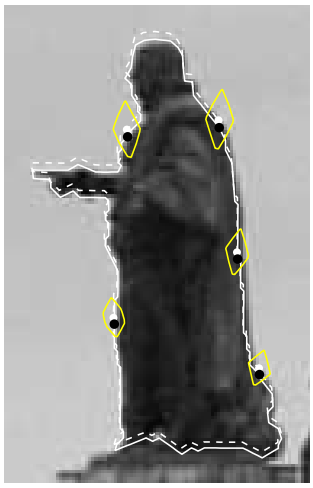


image 43

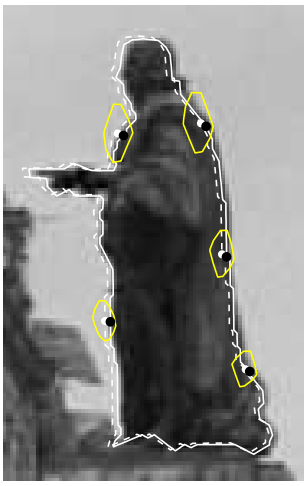


image 71

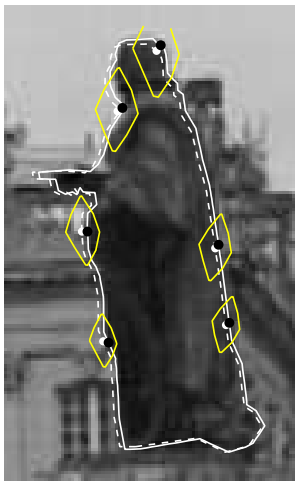


image 99

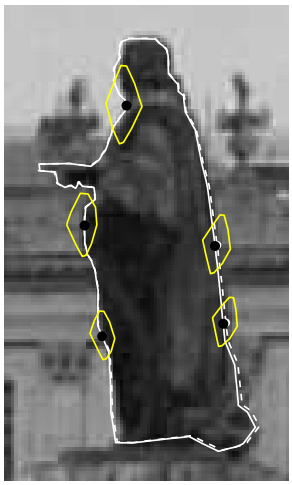


image 127

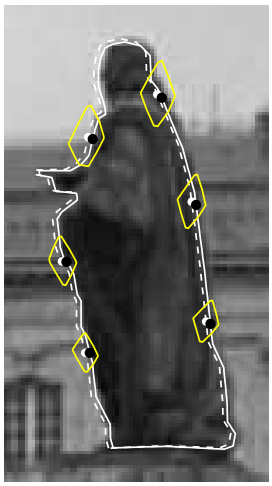


image 155

FIG. 7.3 – *Résultats sur la séquence Stanislas, statue seule, 3 images-clé. Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

| **Temps totaux** | | |
|---|---|---|
| Détourage manuel | $\simeq$ 1 min 30,00 sec | |
| Reconstruction | 0,10 sec | |
| Reprojection | 0,12 sec | |
| Calcul des régions $\Lambda_i$ | 9,17 sec | |
| Correction | 4 min 35,34 sec | |
| Total | $\simeq$ 6 min 15,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | 45,00 sec | par image-clé |
| Reconstruction | 0,10 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Calcul des régions $\Lambda_i$ | 0,06 sec | par image |
| Correction | 1,82 sec | par image |

FIG. 7.4 – *Temps de réalisation de la séquence Stanislas pour la statue seule (2 images-clé, 151 images, 250 points environ par contour).*

Si on souhaite une meilleure précision au niveau de la main de la statue, on peut ajouter une image-clé. Cet ajout est très rapide puisqu'il suffit de modifier localement un des contours déjà calculés : c'est ce que nous avons fait dans l'image 110, situées entre les deux premières images-clé. La main est maintenant détourée plus précisément (voir les images 99 et 127 de la figure 7.3 et les temps de réalisation tableau 7.5).

### 7.2.2 Statue, piédestal et marches

Considérons maintenant comme objet occultant la réunion de la statue, son piédestal et les marches de la base. Les difficultés supplémentaires sont :

- l'objet occultant occupe une majeure partie de chaque image ;
- non seulement le contour des marches est peu marqué mais les ombres créent un contour fort qui ne correspond pas au contour de l'objet ;
- les marches ont une apparence similaire entre elles, ce qui crée de nombreux minima locaux de la fonction de corrélation, comme signalé dans la partie 6.3.5 ;
- les marches sont partiellement occultées par un piéton.

Une des faces du piédestal apparaît image 72, qui a donc été retenue comme image-clé. La première image-clé est la première image de la séquence (image 60) où l'objet est entièrement visible. Enfin, la troisième image-clé est la dernière image de la séquence où les marches ne sont pas encore occultées par un deuxième piéton (voir figure 7.7).

Quand les points de vues utilisés sont ceux obtenus par la méthode basée modèle, les résultats ne sont pas toujours satisfaisants, en particulier le haut du contour retrouvé dans l'image 103, figure 7.8). Ici, le modèle affine utilisé au moment de la correction n'est pas suffisamment souple pour retrouver la position attendue de l'ensemble du contour.

| **Temps totaux** | | |
|---|---|---|
| Détourage manuel | $\simeq$ 1 min 40,00 sec | |
| Reconstruction | 0,20 sec | |
| Reprojection | 0,12 sec | |
| Calcul des régions $\Lambda_i$ | 10,23 sec | |
| Correction | 4 min 37,33 sec | |
| Total | $\simeq$ 6 min 25,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | 35,00 sec | par image-clé |
| Reconstruction | 0,10 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Calcul des régions $\Lambda_i$ | 0,07 sec | par image |
| Correction | 1,84 sec | par image |

FIG. 7.5 – *Temps de réalisation de la séquence Stanislas pour la statue seule (3 images-clé, 151 images, 250 points environ par contour).*

| **Temps totaux** | | |
|---|---|---|
| Détourage manuel | $\simeq$ 6 min 00,00 sec | |
| Reconstruction | 1,58 sec | |
| Reprojection | 0,44 sec | |
| Calcul des régions $\Lambda_i$ | 34,85 sec | |
| Correction | 13 min 56,01 sec | |
| Total | $\simeq$ 20 min 00,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | $\simeq$ 2 min 00,00 sec | par image-clé |
| Reconstruction | 0,79 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Calcul des régions $\Lambda_i$ | 0,23 sec | par image |
| Correction | 5,54 sec | par image |

FIG. 7.6 – *Temps de réalisation de la séquence Stanislas pour la statue et le piédestal (3 images-clé, 151 images, 780 points par contour environ, points de vue obtenus par la méthode basée modèle).*

Une solution peut être d'ajouter une nouvelle image-clé, entre les images 72 et 153. Nous disposons cependant des points de vue estimés après ajustement de faisceaux, qui sont plus précis que ceux retrouvés directement par la méthode basée modèle. Les résultats sont bien meilleurs quand on utilise ces nouveaux points de vue. Les zones d'incertitude sont plus réduites et la projection du contour 3D est déjà très proche de la position attendue : l'étape de correction est presque inutile (figures 7.9 et 7.10). Nous verrons cependant que dans certaines configurations, cette étape reste nécessaire même après ajustement de faisceaux (partie 7.5).

On remarquera que le contour est correctement retrouvé malgré l'occultation partielle par un piéton (voir images 39 et 65).

Enfin, la figure 7.25 montre un exemple d'incrustation où un avion passant derrière la statue, les parties occultées ayant été déterminées à l'aide de la méthode en utilisant les points de vue après ajustement de faisceaux.

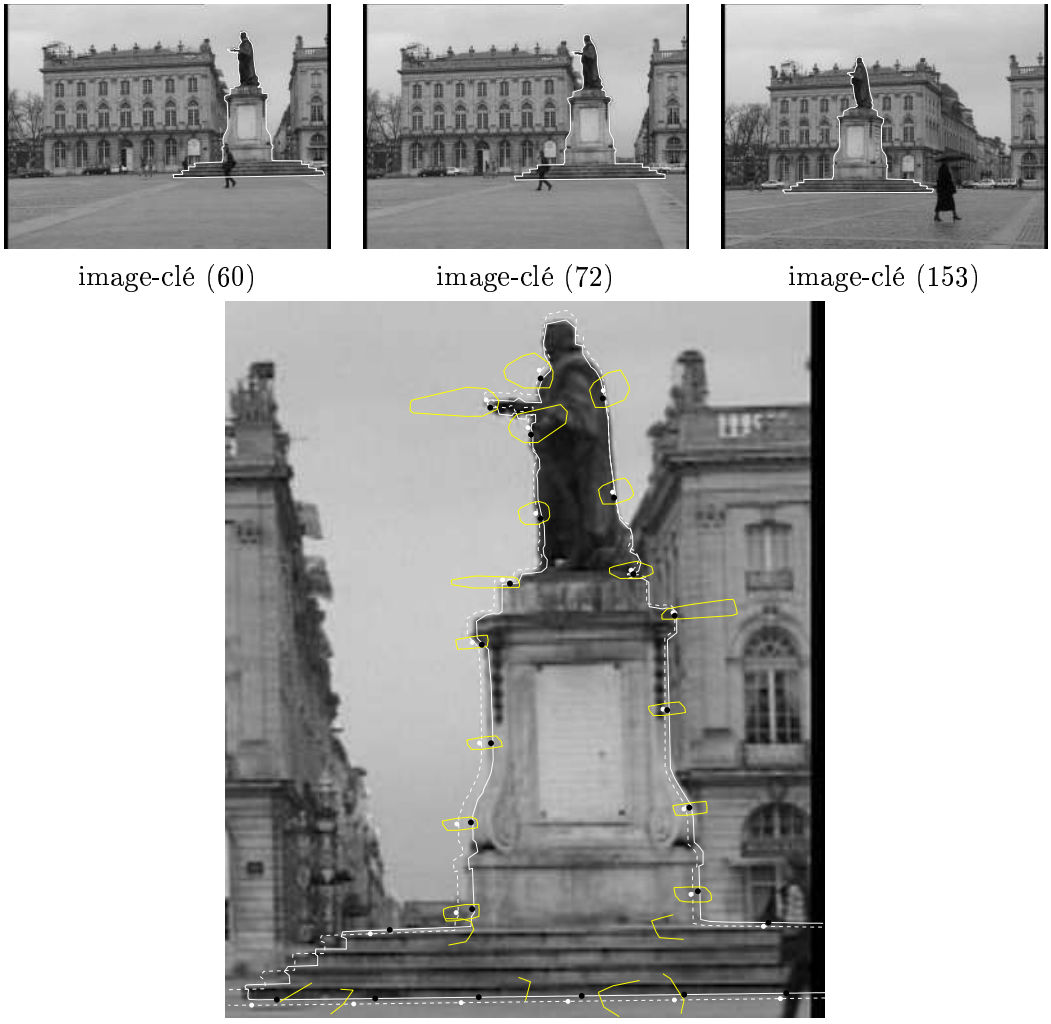


image-clé (60) image-clé (72) image-clé (153)

image 39

FIG. 7.7 – *Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus par la méthode basée modèle.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

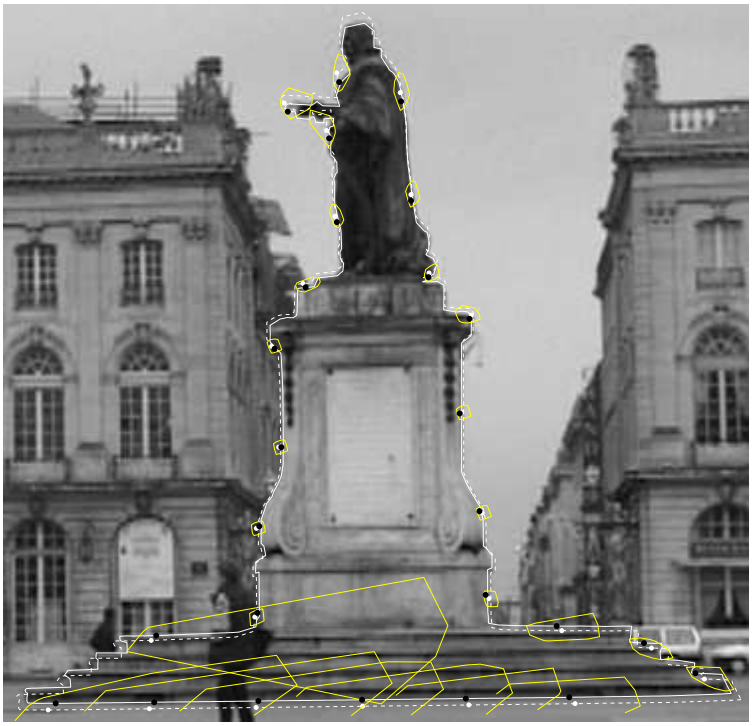


image 65

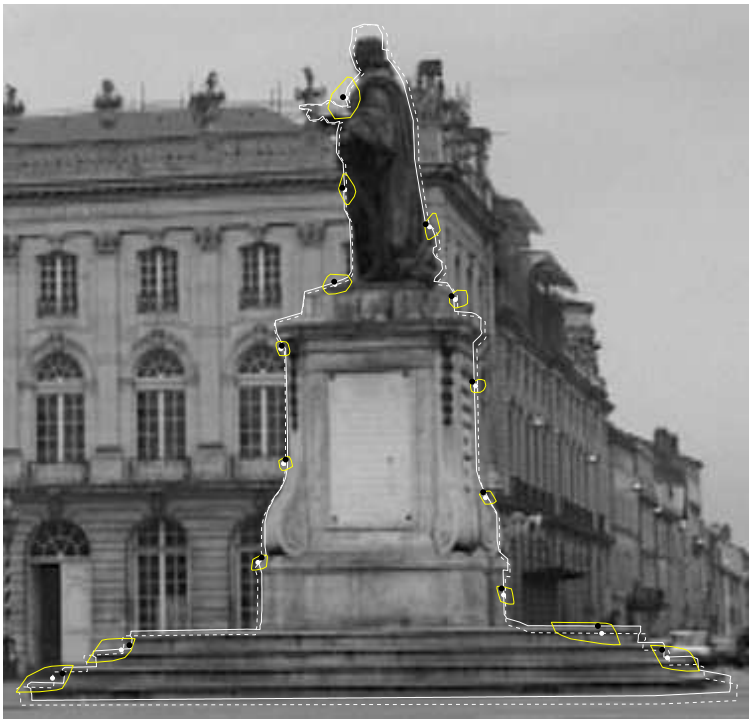


image 103

FIG. 7.8 – *Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus par la méthode basée modèle.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

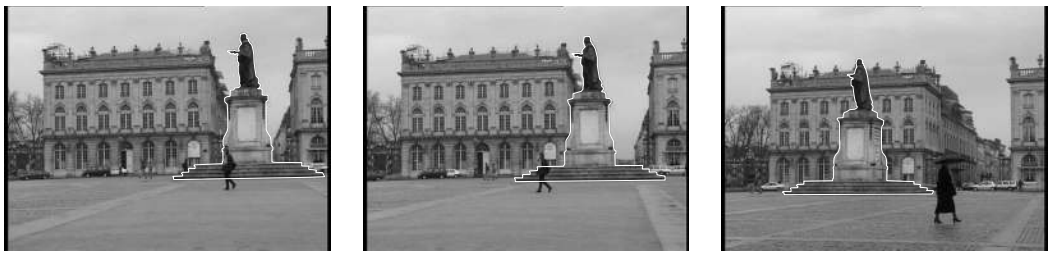


image-clé (60) image-clé (72) image-clé (153)

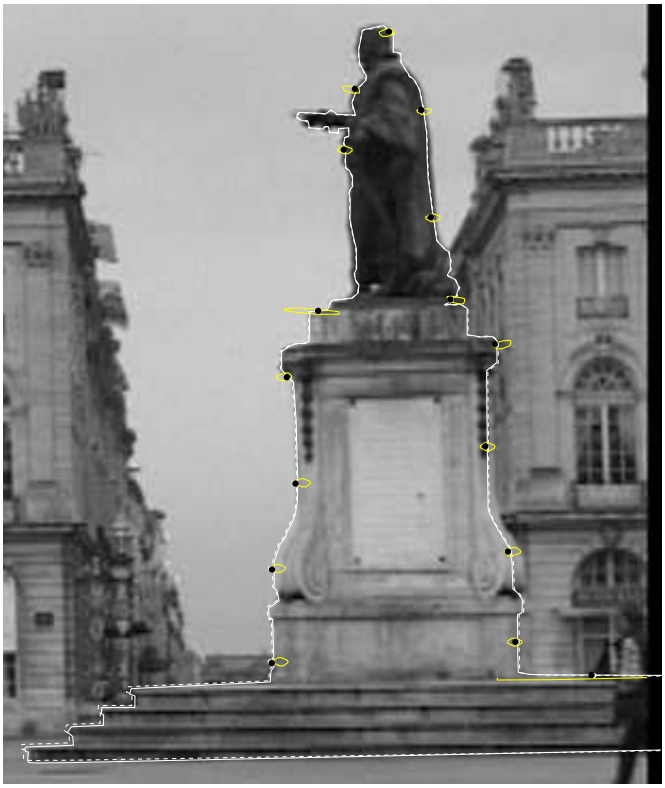


image 39

FIG. 7.9 – *Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus après ajustement de faisceaux.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

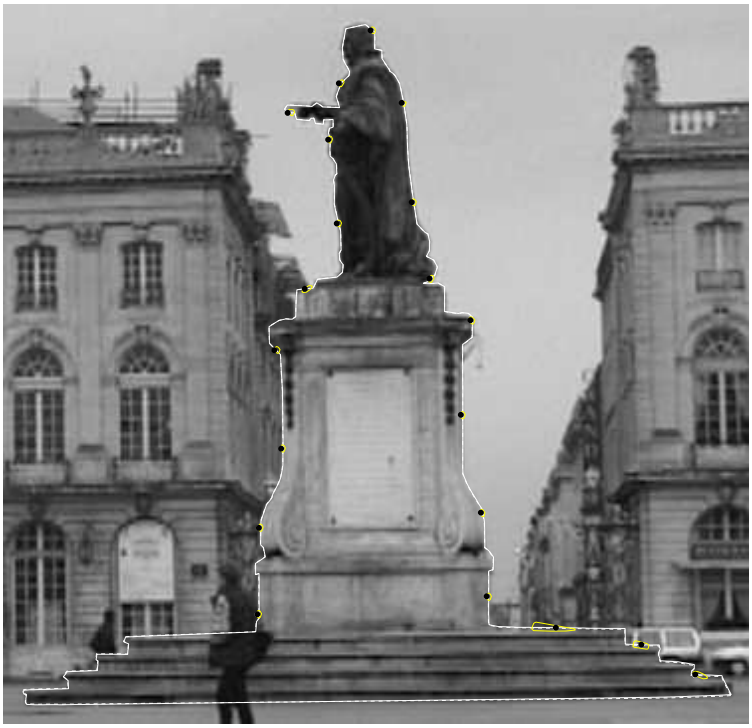


image 65

image 103

FIG. 7.10 – *Résultats sur la séquence Stanislas, statue, piédestal et marches, points de vue obtenus après ajustement de faisceaux.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

Image-clé (0)

Image-clé (30)

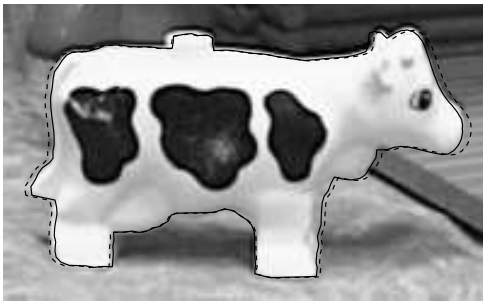


image 3

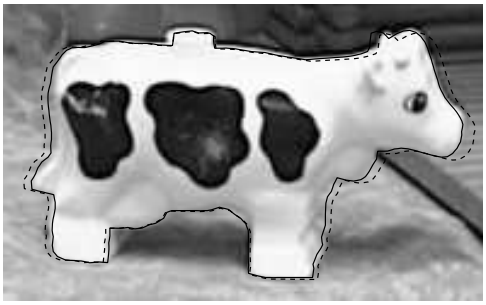


image 6

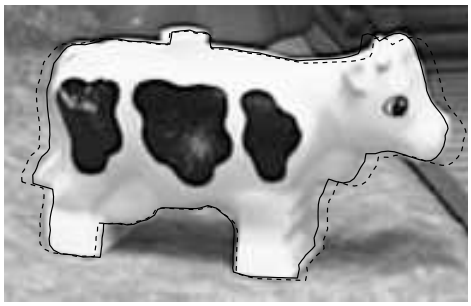


image 9

image 12

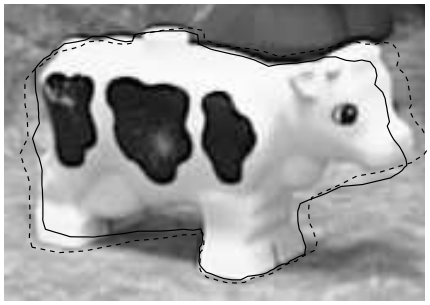


image 15

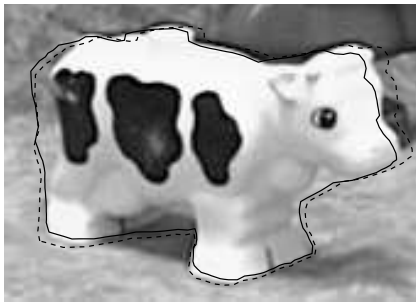


image 19

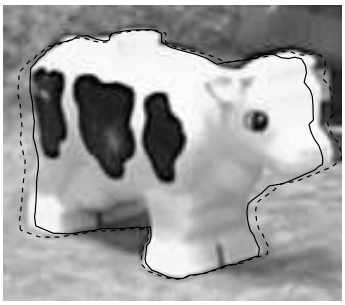


image 23

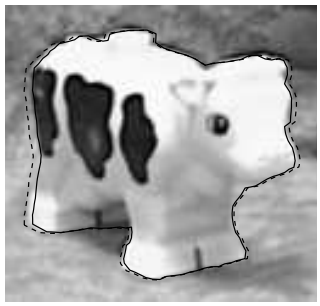


image 27

FIG. 7.11 – *Résultats sur la première séquence du chalet (2 images-clé). Trait pointillé : contour reprojeté; trait plein : contour corrigé.*

Image-clé supplémentaire (15)

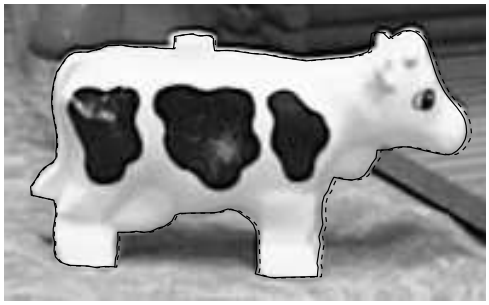


image 3

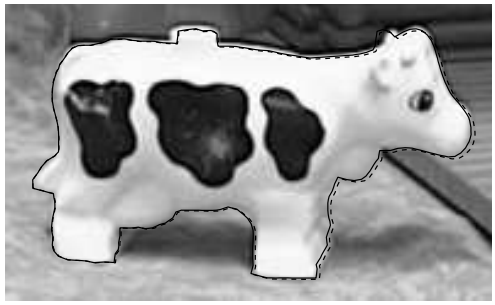


image 6

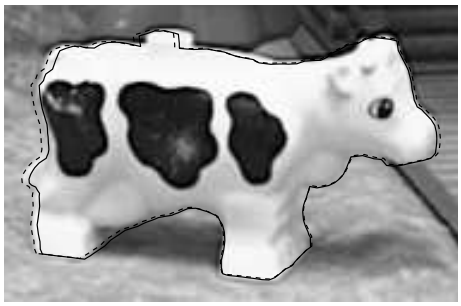


image 9

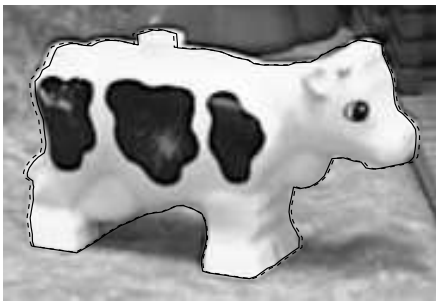


image 12

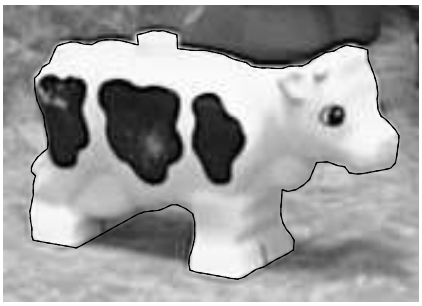


image 15

image 19

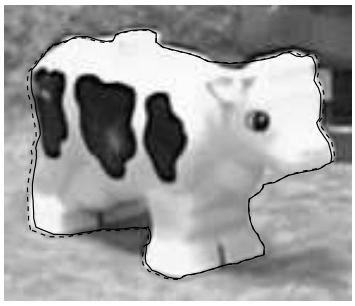


image 23

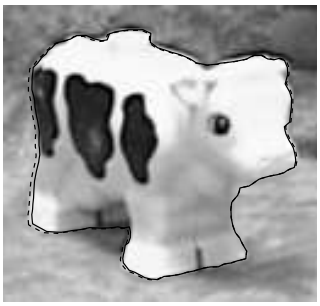


image 27

FIG. 7.12 – *Résultats sur la première séquence du chalet (3 images-clé). Trait pointillé : contour reprojeté; trait plein : contour corrigé.*

| Temps totaux | | |
|---|---|---|
| Détourage manuel | $\simeq$ 1 min 40,00 sec | |
| Reconstruction | 0,12 sec | |
| Reprojection | 0,03 sec | |
| Correction | 1 min 04,79 sec | |
| Total | $\simeq$ 2 min 45,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | 50,00 sec | par image-clé |
| Reconstruction | 0,13 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Correction | 1,86 sec | par image |

FIG. 7.13 – *Temps de réalisation de la première séquence du chalet (31 images, 2 images-clé, 370 points par contour environ).*

## 7.3 Première séquence du chalet : influence des contours apparents

L'intérêt de cette séquence est d'illustrer l'erreur due aux contours apparents. Afin de faire abstraction de l'erreur d'estimation des points de vue, nous avons utilisé une mire de calibration et une table micrométrique pour déterminer les points de vue de façon très précise. L'étape de correction a été effectuée sans la contrainte des zones d'incertitude.

L'objet occultant est une vache jouet, composée de surfaces lisses : les contours de cet objet dans les images sont donc des contours apparents (voir figure 7.11). Entre la première et la dernière image de la séquence (retenues comme images-clé), la caméra effectue une rotation de 60 degrés, et l'apparence de l'objet subit une importante modification.

Comme on pouvait s'y attendre, la projection du contour 3D « déborde » de chaque côté de l'objet, ce contour 3D ayant été reconstruit à partir de contours apparents. L'étape de correction n'est pas suffisante pour retrouver une solution acceptable (voir images de 9 à 19).

Nous avons donc ajouté une troisième image-clé : entre chaque image-clé, la caméra effectue maintenant une rotation de 30 degrés. Les résultats sont beaucoup plus précis (figure 7.12), et peuvent être utilisés pour incruster correctement un objet virtuel derrière la vache jouet (voir figure 7.26).

## 7.4 Séquence du chalet

Les caractéristiques de cette séquence filmée en milieu intérieur sont :

- la caméra se déplace selon des mouvements variés : deux translations le long de l'axe optique et une rotation autour de l'objet occultant;
- le graphe d'aspects de l'objet occultant est relativement complexe.

Cet objet occultant est également une vache jouet, cette fois plus réaliste, ce qui complique le graphe d'aspects et le choix des images-clé. Entre les images 30 et 31, la topologie du contour de l'objet change; il en est de même entre les images 40 et 41. Ces quatre images ont donc été retenues comme images-clé, ainsi que la première et la dernière image de la séquence (voir figure 7.16). La figure 7.17 montre les trois contours 3D reconstruits à partir de ces images-clé. Les points de vue utilisés ont été obtenus à l'aide de la méthode hybride.

| **Temps totaux** | | |
|---|---|---|
| Détourage manuel | $\simeq$ 2 min 30,00 sec | |
| Reconstruction | 0,26 sec | |
| Reprojection | 0,04 sec | |
| Correction | 57,77 sec | |
| Total | $\simeq$ 3 min 30,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | 50,00 sec | par image-clé |
| Reconstruction | 0,13 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Correction | 1,86 sec | par image |

FIG. 7.14 – *Temps de réalisation de la première séquence du chalet (31 images, 3 images-clé, 370 points par contour environ).*

| **Temps totaux** | | |
|---|---|---|
| Détourage manuel | $\simeq$ 12 min 00,00 sec | |
| Reconstruction | 0,66 sec | |
| Reprojection | 0,17 sec | |
| Calcul des régions $\Lambda_i$ | 8,80 sec | |
| Correction | 7 min 40,37 sec | |
| Total | $\simeq$ 20 min 00,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | 2 min 00,00 sec | par image-clé |
| Reconstruction | 0,22 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Calcul des régions $\Lambda_i$ | 0,07 sec | par image |
| Correction | 3,92 sec | par image |

FIG. 7.15 – *Temps de réalisation de la séquence du chalet (120 images, 6 images-clé. 750 points environ).*

La figure 7.18 montre les résultats obtenus (bien que ce ne soit pas visible sur les gros plans, la taille apparente de la vache augmente entre les images 72 et 105, elle passe de 120 pixels à 200 pixels). On notera que l'erreur due aux contours apparents reste inférieure à l'erreur due aux points de vue, puisque la position attendue des points pour lesquels on a calculé la zone d'incertitude se situe bien dans cette zone.

Entre les images 41 et 120, le contour retrouvé n'est pas toujours précis (images 50, 61 et 72). Encore une fois, on peut ajouter une image-clé à moindre coût, en modifiant localement un des contours déjà estimés. En utilisant ainsi le contour dans l'image 50, les résultats sont plus proches de la position attendue (voir figure 7.19), et permettent l'incrustation d'objets virtuels (figure 7.27). L'arbre virtuel positionné devant la vache réelle permet de rappeler que le contour 3D est utilisé pour déterminer si l'objet virtuel est occulté ou non par l'objet réel détouré.

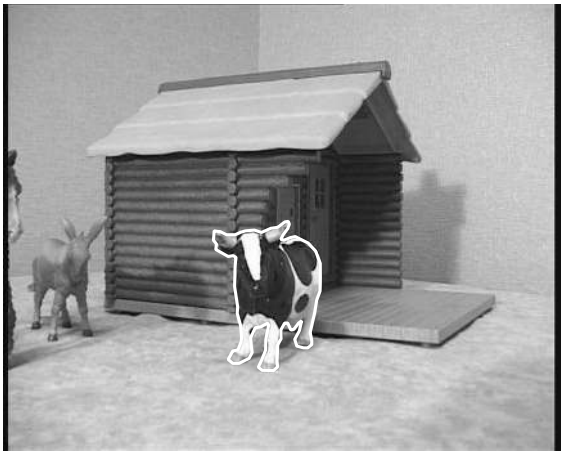


image 0

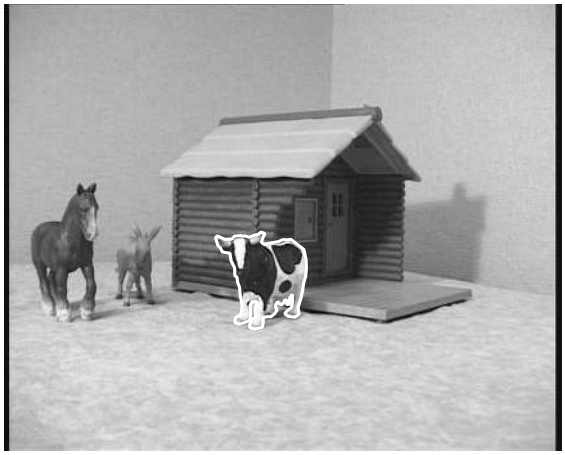


image 30

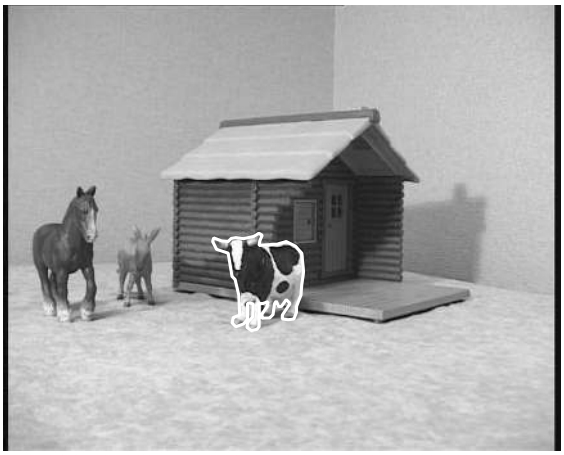


image 31

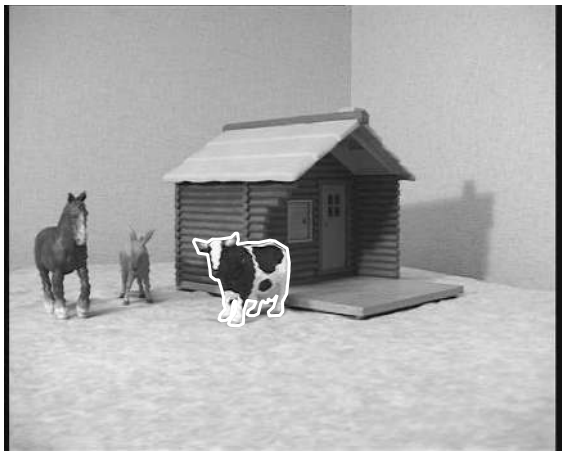


image 40

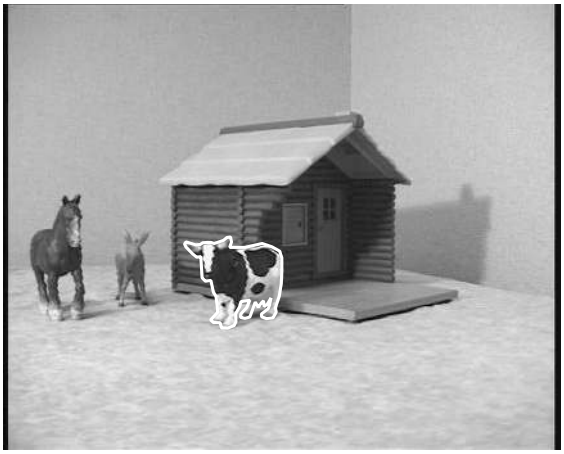


image 41

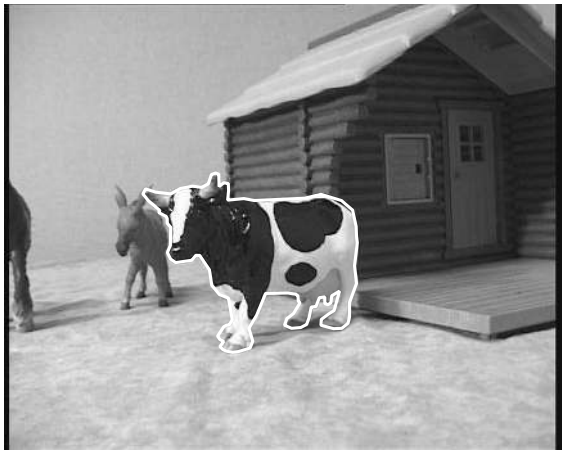
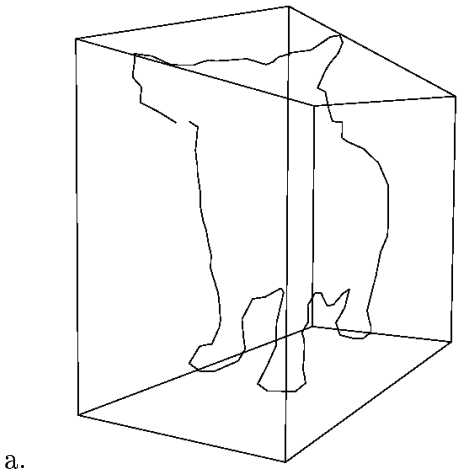
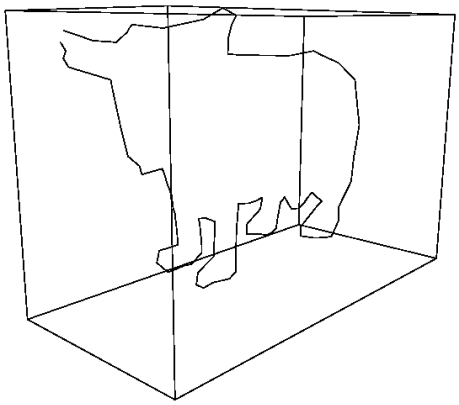


image 120

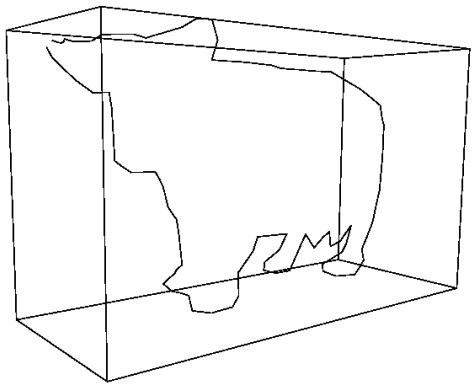
FIG. 7.16 – *Séquence du chalet : les 6 images-clé retenues.*



a.



b.



c.

FIG. 7.17 – *Séquence du chalet. Contours 3D reconstruits à partir des images a : 0 et 30; b : 31 et 40 c : 41 et 120.*

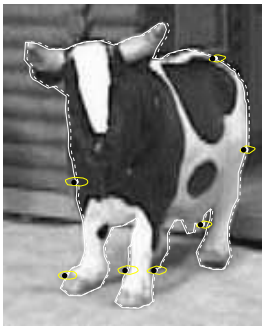


image 15

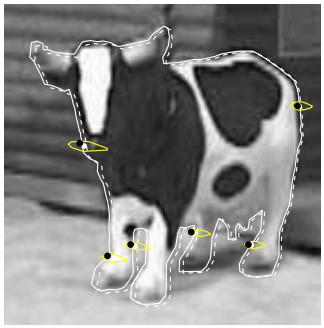


image 35

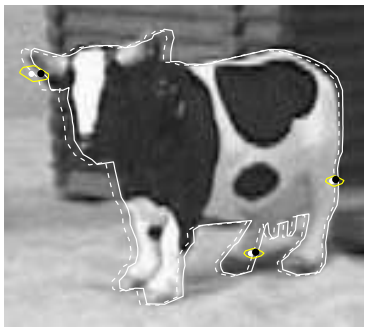


image 50

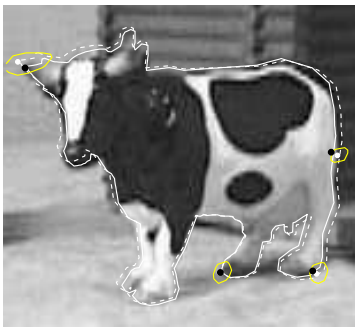


image 61

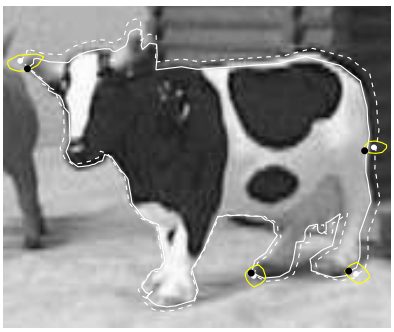


image 72

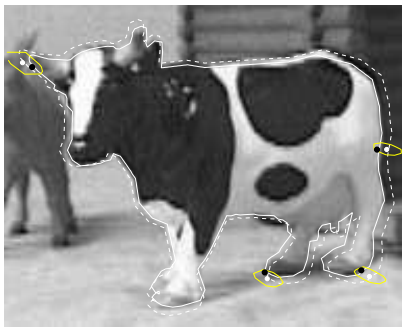


image 83

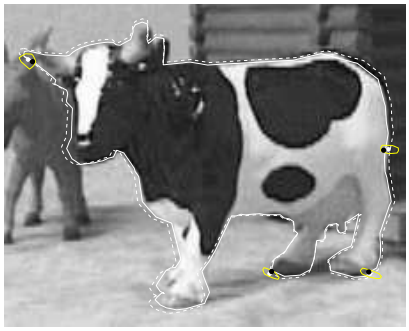


image 94

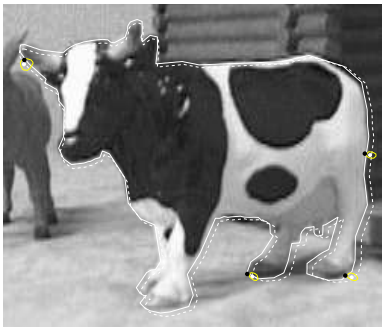


image 105

FIG. 7.18 – *Séquence du chalet : résultats obtenus avec 6 images-clé.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

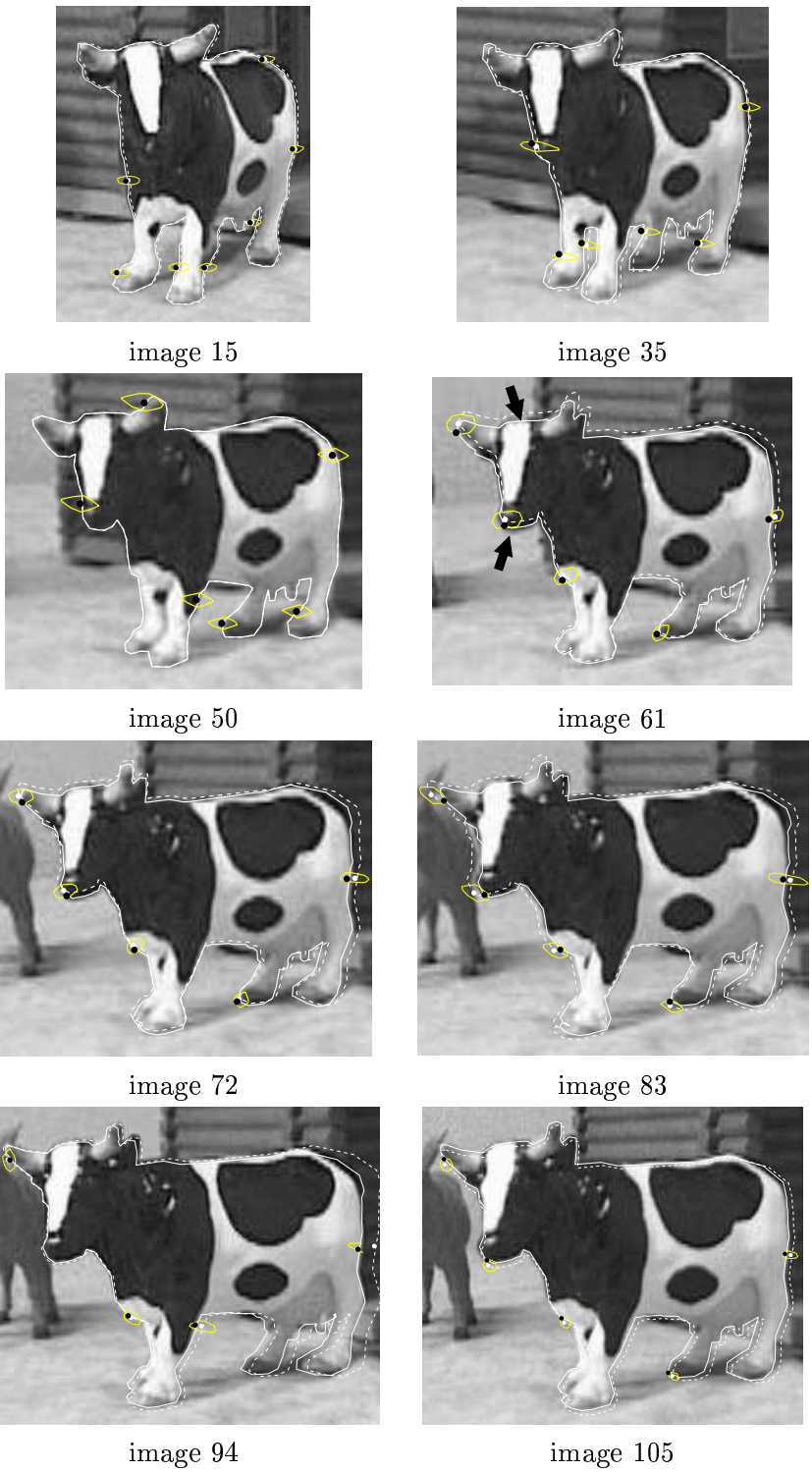



FIG. 7.19 – *Séquence du chalet : résultats obtenus en ajoutant l'image 50 comme image-clé supplémentaire.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

| Temps totaux | | |
|---|---|---|
| Détourage manuel | $\simeq$ 1 min 20,00 sec | |
| Reconstruction | 0,16 sec | |
| Reprojection | 0,63 sec | |
| Calcul des régions $\Lambda_i$ | 36,66 sec | |
| Correction | 10 min 58,89 sec | |
| Total | $\simeq$ 13 min 00,00 sec | |
| **Temps moyens** | | |
| Détourage manuel | 0 min 40,00 sec | par image-clé |
| Reconstruction | 0,16 sec | par contour 3D |
| Reprojection | $\simeq$ 0,00 sec | par image |
| Calcul des régions $\Lambda_i$ | 0,24 sec | par image |
| Correction | 4,11 sec | par image |

FIG. 7.20 – *Temps de réalisation de la séquence du Loria (2 images-clé, 483 images, 170 points environ).*

## 7.5 Séquence du Loria

Cette séquence a été filmée près du laboratoire du Loria. Ses difficultés sont:

- la caméra se déplace le long de l'axe optique, ce qui risque de rendre imprécis la reconstruction;
- la caméra est tenue par un piéton, ce qui rend la séquence saccadée; en particulier, le mouvement apparent de l'objet occultant n'est pas linéaire;
- la caméra passe très près de l'objet occultant, à peu près cylindrique.

Comme l'objet occultant (une balise lumineuse, voir figure 7.21) change peu d'apparence, seules deux images-clé ont été définies. Entre ces deux images, la caméra se déplace le long de son axe optique. Les rayons utilisés pour la reconstruction du contour par triangulation sont donc quasiment parallèles, et une faible erreur de localisation du contour 2D peut résulter en une grande erreur de reconstruction.

Considérons tout d'abord les résultats obtenus en utilisant les points de vue calculés à l'aide de la méthode basée modèle. Les primitives 3D utilisées pour déterminer ces points de vue sont situées sur le bâtiment, et l'objet occultant est éloigné de ces primitives. Suivant un phénomène similaire à celui déjà décrit partie 2.3.5, la projection du contour 3D est très éloignée de la position attendue. A partir de l'image 400, cette projection sort même de l'image (voir figure 7.21).

En utilisant plutôt les points de vue obtenus après ajustement de faisceaux, la projection est beaucoup plus proche. On notera cependant que l'étape de correction reste nécessaire (contrairement à la séquence Stanislas), et que l'erreur de projection augmente au fur et à mesure que la caméra s'approche de l'objet. Sur les dernières images de la séquence, le détourage devient moins précis (voir figure 7.22).

La figure 7.28 montre l'incrustation d'une voiture virtuelle entre la balise lumineuse et le bâtiment. L'ombre de la voiture a été rendue en plaçant manuellement un plan virtuel (non rendu) au niveau du sol et une source lumineuse dans la position approximative du soleil. Les reflets sur la voiture ont été obtenus grâce à une mosaïque d'images prises à partir de la position de la voiture (voir image 7.23).

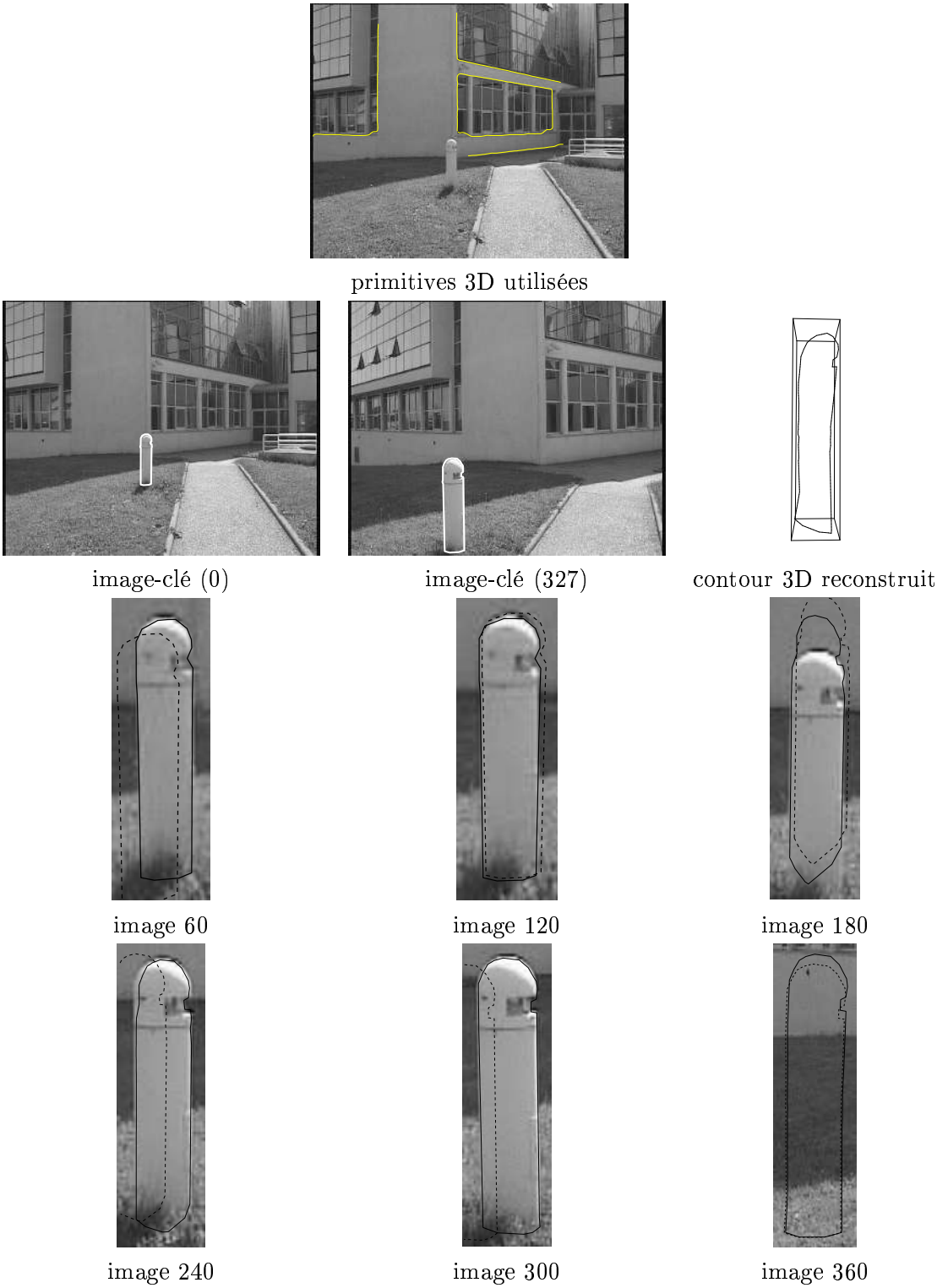



FIG. 7.21 – *Séquence du Loria : résultats obtenus en utilisant les points obtenus par la méthode 3d2d.*

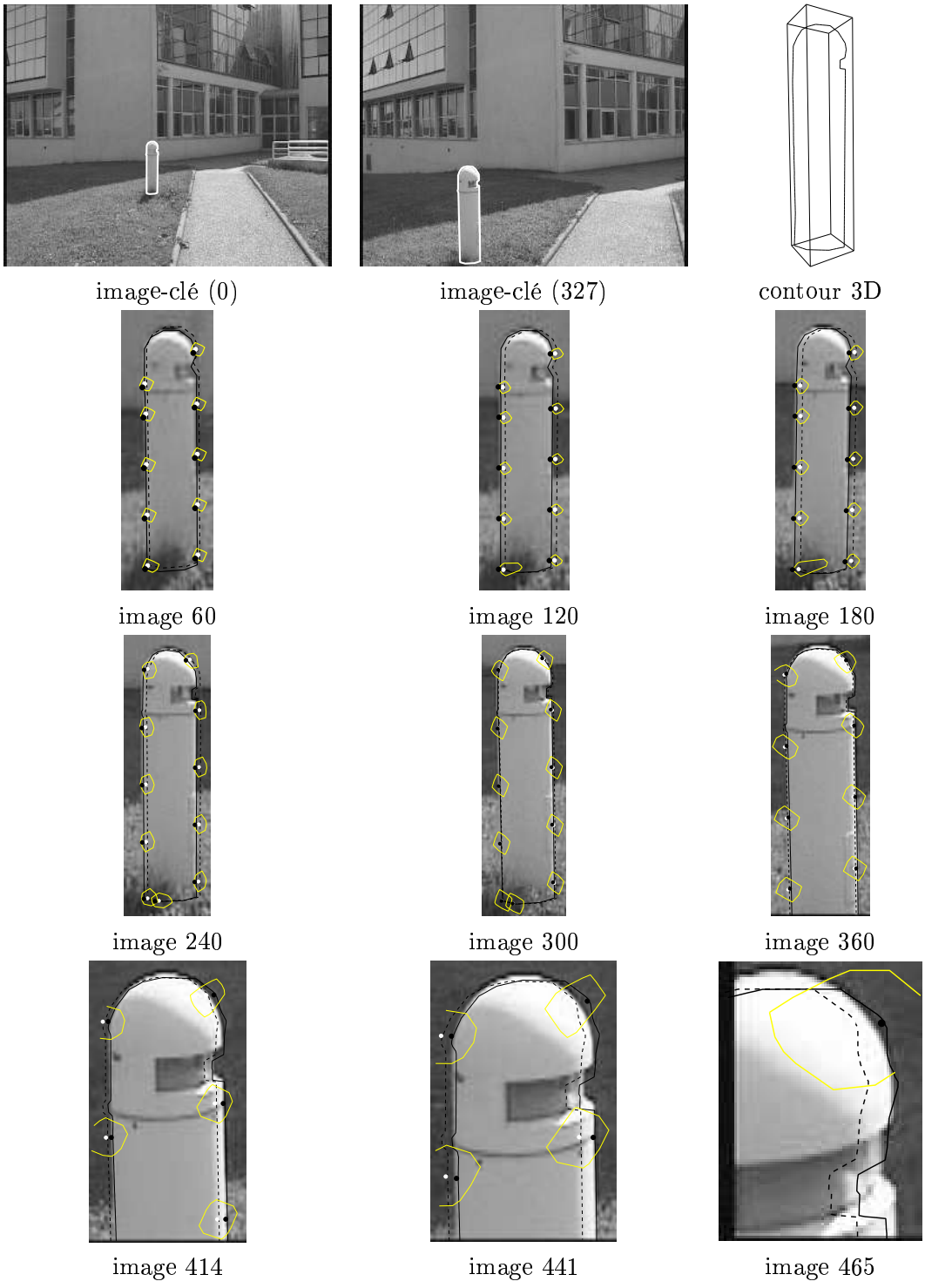



FIG. 7.22 – *Séquence du Loria : résultats obtenus en utilisant les points obtenus après ajustement de faisceaux.*
*La projection est très éloignée de la position attendue, jusqu'à ne plus recouvrir l'objet occultant à partir de l'image 360.*
*Trait pointillé : contour reprojeté; trait plein : contour corrigé; ronds blancs : points contraints à rester dans leur région $\Lambda_i$ (également représentées); ronds noirs : position de ces points après correction.*

FIG. 7.23 – *Mosaïque utilisée pour le rendu des reflets sur la voiture virtuelle de la séquence du Loria.*

## 7.6 Séquence de la voiture : objet occultant mobile et rigide

Cette séquence montre que notre méthode peut être utilisée également dans le cas d'un objet mobile et rigide. Nous avons jusqu'ici supposé que nous connaissions les points de vue dans le référentiel du monde et considéré des objets fixes. Pour pouvoir considérer des objets mobiles, il suffit de connaître les points de vue dans un référentiel lié à l'objet occultant.

Néanmoins, pouvoir déterminer les points de vue par rapport à l'objet considéré nécessite que l'objet soit suffisamment étendu dans l'image, et permettre le suivi de primitives (courbes 3D ou points d'intérêt) suffisamment bien réparties dans l'espace, comme c'est le cas pour l'estimation des points de vue pour une scène fixe.

Pour illustrer la gestion des occultations dues à un objet mobile, nous avons filmé une voiture en mouvement, et déterminé la trajectoire par rapport à cette voiture à l'aide de la méthode basée modèle, à partir de primitives 3D situées sur cette voiture (les fenêtres et le coffre). La figure 7.24 montre les deux images-clé utilisées, les résultats obtenus sur quelques images de la séquence et un exemple d'incrustation. On constatera que la reprojection du contour 3D dans l'image 2 est très éloignée de la position attendue. En effet, dans cette image (comme toutes celles avant l'image-clé 12), le coffre n'est plus visible et ne peut pas être utilisé pour l'estimation des points de vue. Pour retrouver une position correcte malgré l'éloignement de la prédiction, nous avons utilisé une recherche semi-exhaustive suivie d'une minimisation numérique, coûteuse en temps de calcul (plus de deux minutes par image).

Bien que nous ne l'ayons pas testé, on pourrait également prendre en compte des points d'intérêt, que ce soit pour la méthode hybride ou l'ajustement de faisceaux. Afin de distinguer les points d'intérêt situés sur l'objet considéré des points qui appartiennent au reste de la scène, il suffit de se limiter aux points d'intérêt qui apparaissent dans au moins une des silhouettes détourées dans les images-clé.

## 7.7 Séquence Saint-Epvre : cas d'un panoramique

Dans le cas d'un panoramique, le centre de la caméra est fixe, ce qui empêche de réaliser une reconstruction des objets occultants. Cependant, ce cas est beaucoup plus simple, puisque la transformation entre deux images de la séquence est une homographie 2D, qui peut être déterminée quand les points de vue pour ces images sont connus. Il suffit alors de connaître la silhouette de l'objet occultant dans une des images de la séquence pour retrouver cette silhouette dans toutes les autres images. On ne peut en revanche pas connaître la profondeur de l'objet considéré, et il faut définir manuellement si l'objet virtuel est devant ou derrière l'objet considéré.


FIG. 7.24 – *Séquence de la voiture : images-clé, primitives 3D utilisées pour l'estimation des points de vue, résultats (trait pointillé : contour reprojeté; trait plein : contour corrigé) et incrustation.*

La séquence Saint-Epvre illustre le cas d'un mouvement panoramique de la caméra. L'objet occultant est l'ensemble de la statue située au milieu de la place (voir figure 7.29). Les points de vue ont été déterminés à l'aide de la méthode décrite dans [Simon et al.00], basée sur le suivi d'un plan 3D de la scène (ici, le sol). Son intérêt est qu'elle ne nécessite pas de connaissances 3D sur la scène, contrairement aux méthodes basée modèle et hybride. Comme la silhouette de l'objet présente des trous (voir la barrière autour de la statue), nous avons préféré représenter la silhouette par une région 2D plutôt que par une représentation basée sur la frontière.

Une étape de correction est dans ce cas également nécessaire. Nous ne connaissons pas l'incertitude des points de vue, qui ont été calculés à l'aide d'une méthode différente de celles décrites dans cette thèse. Cependant, on remarquera que le critère de corrélation utilisé par l'étape de correction n'a pas à être ici limité au masque de l'objet occultant : il peut être étendu à l'ensemble de l'image, ce qui confère au résultat obtenu une grande précision. Nous recherchons donc pour chaque image $\mathrm{I}_{\mathrm{inter}}$ de la séquence l'homographie 2D $H$ qui minimise:

$$\sum_{\mathbf{p} \text{ pixel de } \mathrm{I}_{\mathrm{inter}}} \left(\mathrm{I}_{\mathrm{inter}}\left(\mathbf{p}\right) - \mathrm{I}_{\mathrm{clé}}\left(H\left(\mathbf{p}\right)\right)\right)^2$$

$H$ est initialisée par l'homographie calculée à partir des points de vue des images $\mathrm{I}_{\mathrm{clé}}$ et $\mathrm{I}_{\mathrm{inter}}$,

et la minimisation est effectuée à l'aide de la même méthode itérative que précédemment.

La région 2D représentant l'objet occultant et les résultats obtenus sont présentés figure 7.29.

## 7.8 Un outil de segmentation temporelle d'objets dans des séquences vidéos

Comme nous l'avons déjà fait remarquer, l'outil développé ici ne se limite pas à la gestion des occultations. Il peut également être utilisé comme outil de segmentation temporelle d'objets dans des séquences vidéos, et avoir ainsi de multiples applications utiles pour la post-production, dont nous présentons maintenant quelques exemples.

### 7.8.1 Colorisation

Puisque la silhouette de l'objet est connue dans chaque image, elle peut être utilisée pour modifier les couleurs de l'objet dans la séquence. Par exemple, nous avons multiplié les composantes rouge, vert et bleu des pixels correspondant à la statue de la séquence Stanislas par des coefficients lui donnant une apparence dorée (figure 7.30).

### 7.8.2 Composition vidéo

De même, la connaissance des silhouettes permet d'extraire l'objet considéré de la séquence vidéo pour l'insérer dans une autre séquence.

Pour illustrer cette application, nous avons réalisé le rendu d'une scène virtuelle à l'aide des points de vue calculés sur la séquence Stanislas, puis incrusté dans les images synthétisées la statue, le piédestal et les marches extraits de la séquence vidéo (figure 7.31). La profondeur de l'objet incrusté est connue et pourrait permettre, comme précédemment, de gérer automatiquement les occultations entre les parties virtuelles et cet objet.

### 7.8.3 Réalité diminuée

La dernière application que nous présenterons consiste en la suppression d'objets réels de la séquence vidéo. Outre les effets spéciaux, l'intérêt d'une telle application est de permettre, par exemple, la visualisation d'une scène urbaine après la suppression d'un bâtiment.

Une fois que la silhouette de l'objet à supprimer est connue dans chacune des images de la séquence, il faut pouvoir la remplacer par le fond de la scène. Nous avons défini pour cela deux stratégies. Si l'objet est mince, une simple interpolation de l'intensité (ou des composantes rouge, vert et bleu) de l'image permet d'obtenir une suppression réaliste (voir figure 7.32 qui montre la suppression d'une chaîne à l'avant-plan d'une séquence).

Pour des objets plus larges, une telle interpolation ne suffit plus. A la place, nous utilisons une reconstruction 3D basée sur le maillage de points d'intérêt suivis dans la séquence. Ceci n'est évidemment possible que si la séquence permet de reconstruire la partie de la scène située derrière l'objet, et si cette partie est suffisamment simple pour permettre une reconstruction fiable. En considérant uniquement les points d'intérêt qui ne se trouvent pas dans une des silhouettes de l'objet, on reconstruit alors la scène sans l'objet à supprimer. Cette reconstruction (texturée) est ensuite projetée dans les images pour remplacer la silhouette de l'objet supprimé (voir figure 7.33). On trouvera plus de détail dans [Lepetit et al.01].

## 7.9 Conclusion

Nous pensons qu'à travers les multiples exemples de séquences qui viennent d'être décrits, nous avons montré que le système développé dans cette thèse répond aux critères fixés dans le chapitre 1. En effet, ce système devait être:

- **général :** nous avons montré que les objets occultants peuvent être de formes très générales, et ne sont pas limités à des formes polyédriques (voir par exemple la statue de la séquence Stanislas ou les vaches jouets). De même, le système n'impose pas de contraintes sur le mouvement de la caméra, ce mouvement n'étant pas obligatoirement favorable pour la reconstruction 3D (voir la séquence du Loria).
  Une limite à la généralité qui apparaît est que l'aspect de l'objet occultant ne doit pas varier trop souvent le long de la séquence, l'intervention de l'utilisateur deviendrait très forte le cas échéant. De plus, les objets occultants sont contraints à être rigides, même s'ils peuvent être mobiles.
- **précis :** les exemples montrent que la précision requise pour une incrustation réaliste est souvent obtenue : les contours retrouvés correspondent bien aux contours de l'objet occultant, les détails fins de cet objet peuvent également pris en compte (voir en particulier la main de la statue de la séquence Stanislas).
  Cette précision peut être limitée quand le contour considéré est un contour apparent, comme nous l'avons en particulier sur la première séquence du chalet. Il faudrait faire suivre l'étape de correction d'une étape d'adaptation locale. Or, nous avons vu (chapitre 5) qu'une telle correction reste difficile. Une solution dans le système actuel est alors d'ajouter une image-clé supplémentaire permettant d'affiner la prédiction du contour.
- **robuste :** par la prise en compte explicite des erreurs de points de vue, le système est robuste à ces erreurs. Les limites de cette robustesse ont été montrées sur la séquence Stanislas et la séquence du Loria, pour les points de vue obtenus par la méthode basée modèle. Si des points de vue plus précis (obtenus après ajustement de faisceaux par exemple) ne sont pas disponibles, l'utilisateur a toujours la possibilité d'ajouter une image-clé pour améliorer les résultats.
  De plus, la scène peut contenir des objets mobiles, qui peuvent éventuellement occulter l'objet considéré (voir les voitures et les piétons de la séquence Stanislas).
- **intuitif, avec une interaction réduite :** même si la méthode est basée sur une reconstruction tridimensionnelle des objets occultants, l'utilisateur n'a à considérer que les images de la séquence. Le choix des images-clé est parfois délicat (voir la séquence du chalet), mais reste simple dans la plupart des cas. Le détourage des objets ne nécessite pas de connaissances particulières. Du point de vue de l'utilisateur, tout se passe comme si la méthode interpolait les silhouettes entre les images-clé, d'une manière beaucoup plus fiable que d'une interpolation 2D puisqu'on prend en compte la nature tridimensionnelle des objets. De même, si l'utilisateur décide de corriger un des contours obtenus (comme nous l'avons fait pour l'image 110 de la séquence Stanislas, ou l'image 50 de la séquence du chalet), cette correction est répercutée automatiquement dans les autres images, en ajoutant l'image dans laquelle est effectuée la correction au nombre des images-clé.
  La gestion des occultations ne se fait qu'à partir des images de la séquence, sans nécessiter de connaissances sur la scène considérée. Bien sur, les points de vue sont calculés ici à partir de la connaissance de primitives 3D de la scène. Mais on peut envisager d'utiliser un algorithme basé image d'estimation des points de vue, des produits fondés sur de tels algorithmes commençant à être disponibles [Realviz00, 2d300].

image 39

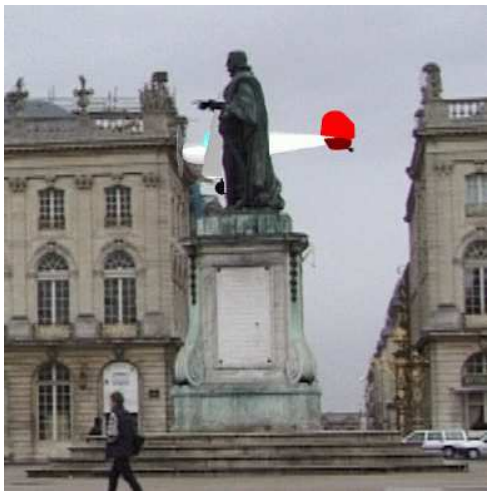


image 65

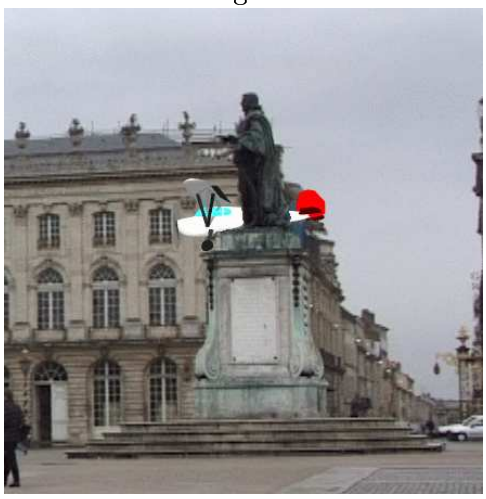


image 93

image 103

image 135

image 145

FIG. 7.25 – *Incrustation d'un avion virtuel dans la séquence Stanislas.*

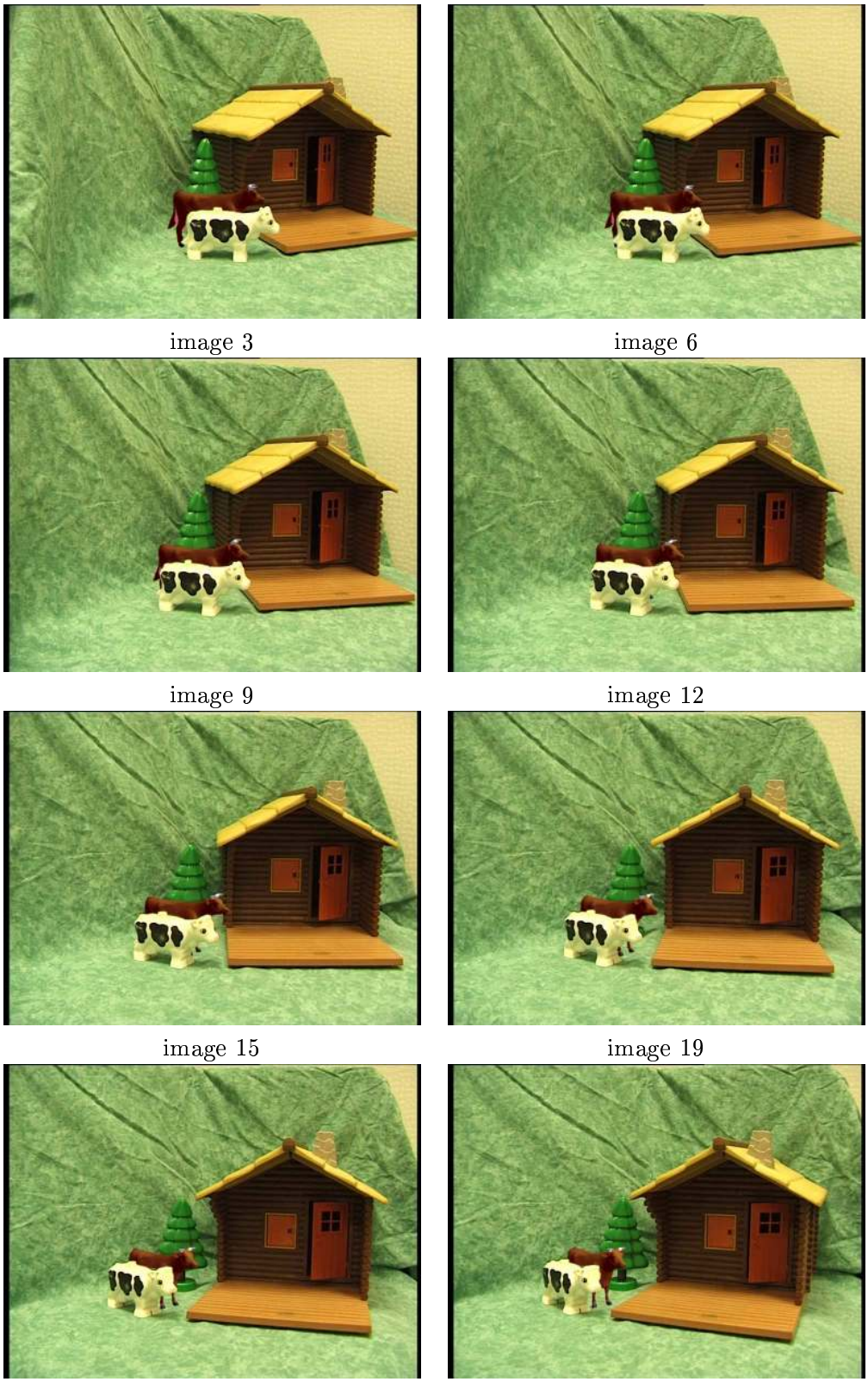


image 3 image 6

image 9 image 12

image 15 image 19

image 23 image 27

FIG. 7.26 – *Incrustation d'une vache virtuelle dans la première séquence du chalet.*

image 15

image 35

image 50

image 61

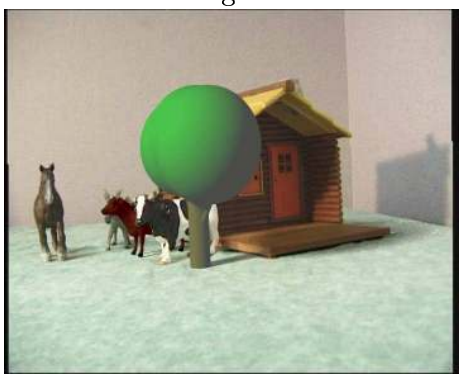


image 72

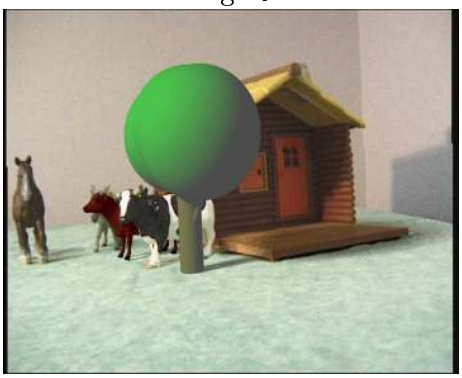


image 83

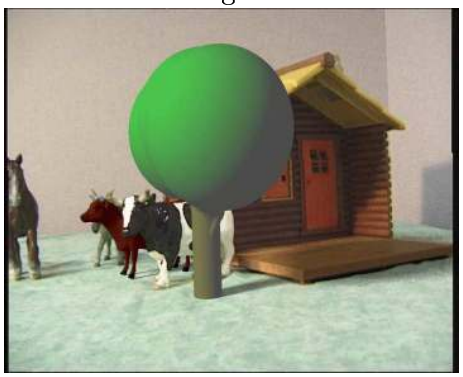


image 94

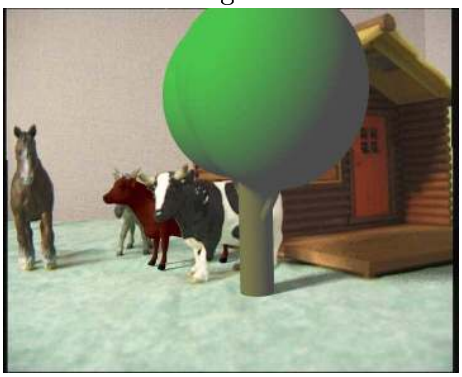


image 105

FIG. 7.27 – *Séquence du chalet : incrustation d'objets virtuels.*

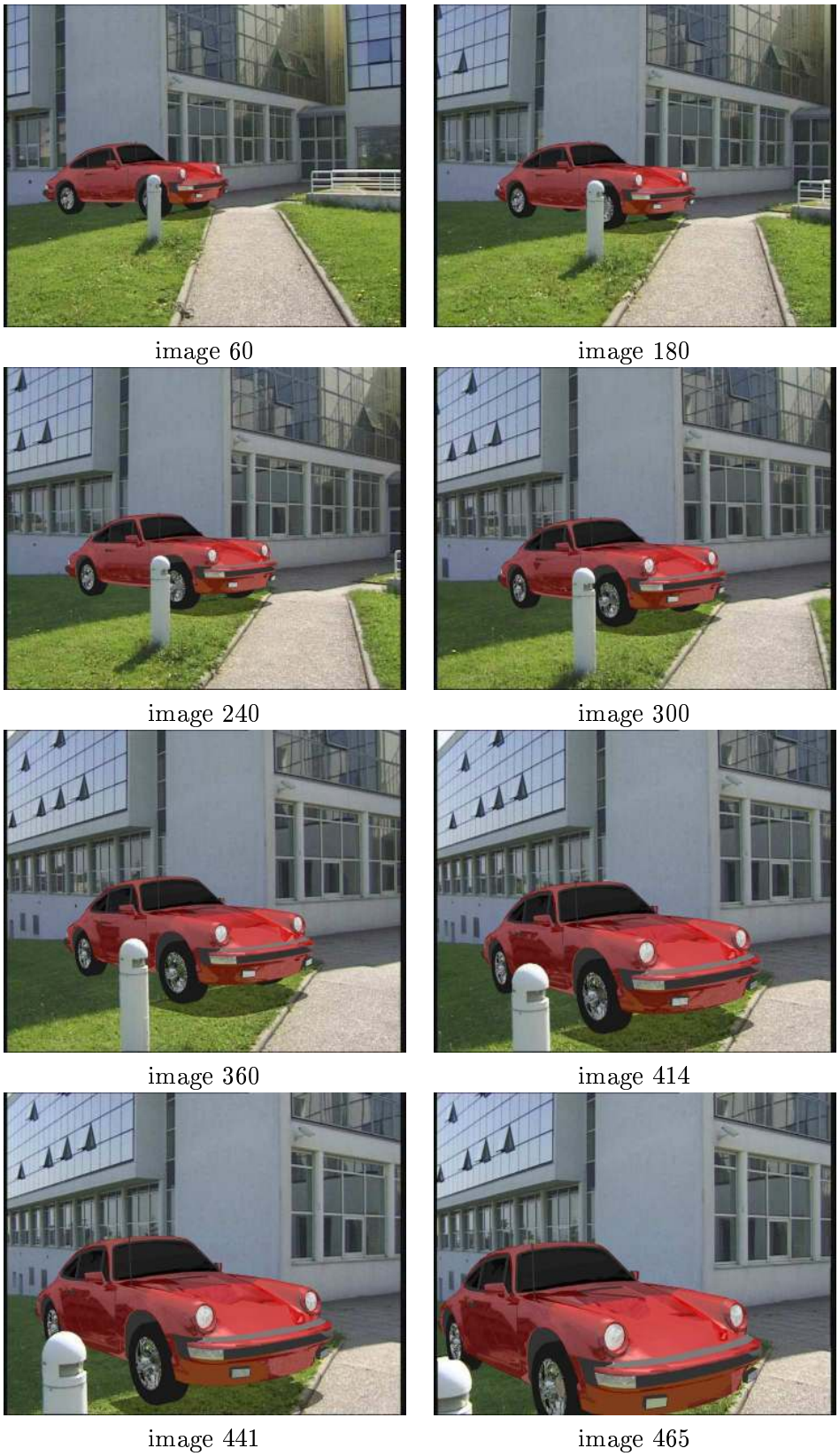


image 60 image 180

image 240 image 300

image 360 image 414

image 441 image 465

FIG. 7.28 – *Séquence du Loria : incrustation d'une voiture virtuelle.*

masque de l'objet occultant
défini dans l'image 40

image 40

image 20

image 55

image 105

image 60

FIG. 7.29 – *Résultats sur la séquence Saint-Epvre.*


FIG. 7.30 – *a. Une image originale; b-d : images après modification des couleurs de la statue.*

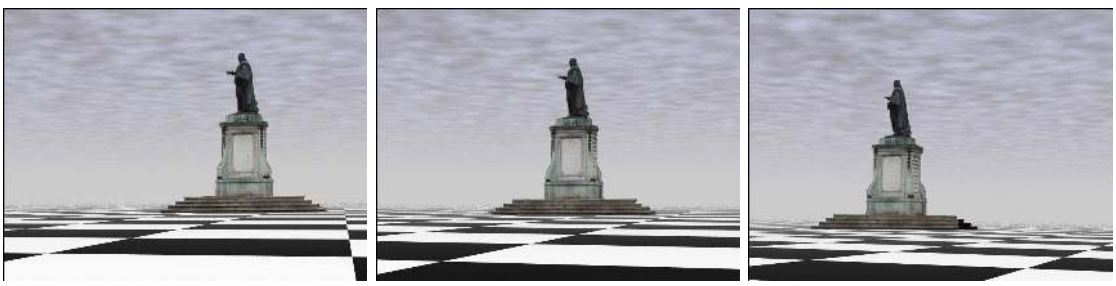


FIG. 7.31 – *Composition de la statue avec une séquence d'images de synthèse.*

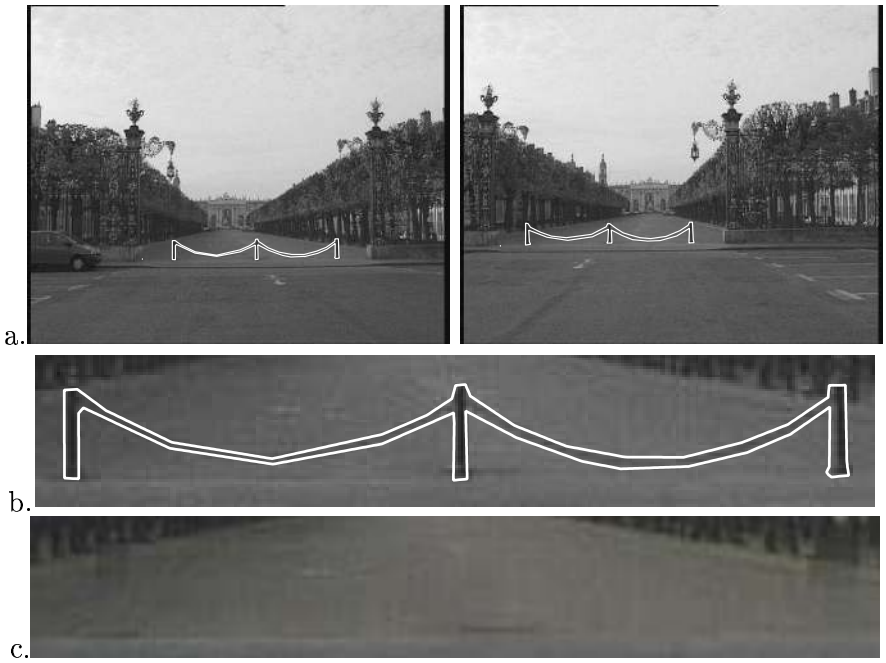


FIG. 7.32 – *Séquence de la chaîne : a. les deux images-clé; b. résultat du détourage dans une image intermédiaire; c. suppression de la chaîne par interpolation.*

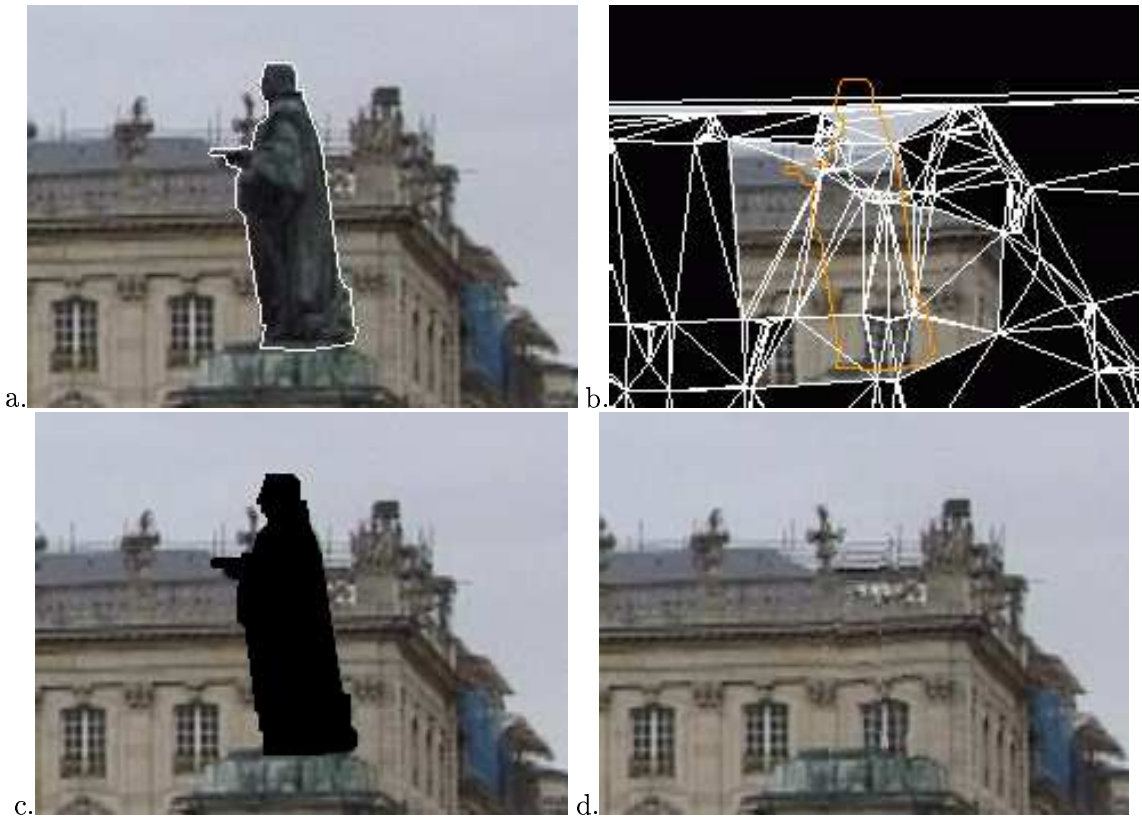


FIG. 7.33 – *a. Résultat du détourage dans une des images; b. reprojection de la scène reconstruite sans l'objet à reconstruire; c. masque de l'objet supprimé; d. résultat final.*

# Chapitre 8

# Conclusion

## 8.1 Apports de la thèse

Notre travail considère la gestion des occultations en Réalité Augmentée. Il est motivé par l'importance des applications de l'incrustation d'objets virtuels dans des séquences vidéo, rendues possibles par le développement récent d'algorithmes d'estimation de points de vue. De telles applications, en particulier en post-production, demandent de retrouver précisément les parties cachées des objets virtuels par la scène réelle.

### 8.1.1 Estimation de l'incertitude des points de vue

Nous avons montré comment estimer l'erreur sur les points de vue, selon qu'ils soient obtenus par la méthode basée modèle, hybride ou après ajustement de faisceaux. Cette estimation est utilisée par la suite dans la méthode de gestion des occultations développée dans la thèse. Au delà de cette méthode, elle est très intéressante pour estimer l'erreur de reconstruction et de reprojection. En particulier, l'estimation de l'erreur des points de vue obtenu après ajustement de faisceaux est très générale puisque cet ajustement constitue souvent la dernière étape des algorithmes de calcul de points de vue en post-production.

### 8.1.2 Une approche semi-automatique pour une grande précision

Un point fort de la méthode développée dans cette thèse est qu'elle se traduit en un outil intuitif et simple d'emploi. En effet, l'utilisateur travaille directement sur les images de la séquence, sans avoir à fournir d'informations plus complexes que le détourage 2D des objets. Les points de vue doivent être connus, mais ceci est par ailleurs nécessaire pour l'incrustation des objets virtuels. La gestion des occultations peut alors être réalisée efficacement et facilement, moyennant une intervention relativement réduite de l'utilisateur. La part d'interactivité de la méthode est justifiée par le manque de fiabilité des méthodes automatiques, et permet de garantir de bons résultats. Cette interactivité est en accord avec la philosophie actuelle de développement de produits utilisant la vision par ordinateur.

La prise en compte de l'incertitude des points de vue par cette méthode permet de l'utiliser dans des cas pratiques, où l'erreur sur les points de vue calculés peut être relativement importante.

### 8.1.3 Un outil de suivi dans des séquences vidéo

Nous avons également montré que cette méthode pouvait être utilisée dans un cadre plus général que la gestion des occultations, en offrant une segmentation temporelle d'objets dans une séquence vidéo. Elle trouve alors de multiples applications pour la post-production: colorisation et suppression d'objets, composition de séquences réelles et virtuelles...

## 8.2 Limites de la méthode

### 8.2.1 Une étape d'affinement local?

Nos expérimentations nous ont permis d'identifier une limite de la méthode, à savoir un manque de précision quand le contour occultant est un contour apparent. Une étape de correction locale ne serait parfois pas inutile, cependant on a vu que l'utilisation d'un contour actif, solution de correction locale souvent utilisée, risque de dégrader les résultats plutôt que de les améliorer, l'information de contours n'étant pas suffisamment pertinente dans des séquences vidéo de scènes extérieures.

On retiendra que l'information de contours, si elle peut être utilisée dans d'autres contextes (par exemple, dans des milieux industriels contraints), n'est pas réellement utilisable dans notre cadre qui est plus général.

Le problème est effectivement difficile : le contour détouré correspond à un contour apparent. Or, nous avons montré que le voisinage de ces contours est très difficile à reconstruire précisément, à identifier ou à localiser. Ceci est surtout vrai dans notre cas où l'on ne dispose que de quelques positions du contour de l'objet.

Une solution pour obtenir une plus grande précision de détourage des contours apparents dans les images intermédiaires consiste évidemment à augmenter le nombre d'images-clé. Cette solution n'est pas forcément très satisfaisante puisqu'elle demande un surcroît de travail pour l'utilisateur.

### 8.2.2 Objets occultants avec un graphe d'aspects complexe

Il est difficile de considérer un objet occultant avec un graphe d'aspects complexe. Plus exactement, il vaut mieux éviter d'avoir à traiter des séquences pour lesquelles la caméra traverse fréquemment les frontières des régions maximales du graphe d'aspects de l'objet. Nous avons montré qu'il était possible de prendre en compte un objet avec un graphe d'aspects relativement complexe (voir la séquence du chalet). Certains objets complexes pourraient demander un grand nombre d'images-clé selon le déplacement de caméra, et la méthode perdrait de son intérêt. Pour donner un exemple, les séquences où la caméra tourne autour d'un arbre en hiver sont à proscrire...

## 8.3 Application interactive

La méthode a été conçue pour une utilisation en post-production, c'est-à-dire dans une étape postérieure à l'acquisition de la séquence. Ceci nous permet de demander à l'utilisateur d'effectuer un détourage dans les images-clé, et de réaliser une reconstruction adaptée à la séquence.

On peut tout de même suggérer un emploi de cette méthode en temps réel, dans un cadre relativement contraint. Imaginons par exemple une application de type studio virtuel: la caméra se déplace dans une zone restreinte d'un plateau de tournage, et on souhaite incruster des éléments

virtuels dans les images retransmises par cette caméra. En pré-traitement, un opérateur peut déplacer la caméra pour définir un ensemble d'images-clé, dans lesquelles les objets considérés peuvent être détourés. Les contours 3D correspondant peuvent alors être pré-calculés.

Si les points de vue sont retrouvés par l'intermédiaire du bras mécanique déplaçant la caméra, ils sont connus avec une grande précision. Pour des objets polyédriques, l'étape de correction devient alors inutile. Pour chaque image, ne reste que la reprojection du contour 3D, étape quant à elle peu gourmande en temps de calcul, surtout grâce aux techniques de rendu temps réel.

# Bibliographie


[2d300] 2d3. – http://www.2d3.com.

[Ataman et al.81] Ataman (E.), Aatre (V. K.) et Wong (K. M.). – Some statistical properties of median filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-29 (5), 1981, p. 1073.

[Ayache88] Ayache (N.). – *Construction et Fusion de Représentations Visuelles 3D, Applications à la Robotique Mobile.* – Thèse d'Etat, Université de Paris-Sud, Centre d'Orsay, 1988.

[Ayache89] Ayache (N.). – *Vision stéréoscopique et perception multisensorielle. Applications à la robotique mobile.* – InterEditions, 1989.

[Azuma97] Azuma (R. T.). – A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, vol. 6 (4), August 1997, pp. 355–385.

[Baker et al.81] Baker (H. H.) et Binford (T. O.). – Depth from Edge and Intensity Based Stereo. *In : Proceedings of 7th International Joint Conference on Artificial Intelligence, Vancouver, B.C. (Canada)*, pp. 631–636.

[Balcisoy et al.00] Balcisoy (S.), Torre (R.), Ponder (M.), Fua (P.) et Thalmann (D.). – Augmented Reality for Real and Virtual Humans. *In : Proceedings of the Conference on Computer Graphics International.* pp. 303–308. – Los Alamitos, CA, June 19–24 2000.

[Bard74] Bard (Y.). – *Nonlinear Parametric Estimation.* – Academic Press, 1974.

[Bascle et al.94] Bascle (B.), Bouthemy (P.), Deriche (R.) et Meyer (F.). – Tracking Complex Primitives in an Image Sequence. *In : Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem (Israel).*

[Berger et al.99] Berger (M.-O.), Winterfeldt (G.) et Lethor (J.-P.). – Contour Tracking in Echocardiographic Sequences without Learning Stage: Application To the 3D Reconstruction of The Beating Left Ventricule. *In : Medical Image Computing and Computer assisted Intervention, Cambridge (England)*, pp. 508–515.

[Berger91] Berger (M.-O.). – *Les contours actifs : modélisation, comportement et convergence.* – Vandœuvre-lès-Nancy, Thèse de doctorat, Institut National Polytechnique de Lorraine, February 1991, 47–54p.

[Berger93] Berger (M.-O.). – Tracking Rigid and non Polyhedral Objects in an Image Sequence. *In : Proceedings of 8th Scandinavian Conference on Image Analysis, Tromsø (Norway)*, pp. 945–952.

[Berger94] Berger (M.-O.). – How to Track Efficiently Piecewise Curved Contours with a View to Reconstructing 3D Objects. *In : Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem (Israel)*, pp. 32–36.

[Berger97] Berger (M.-O.). – Resolving occlusion in augmented reality: a contour-based approach without 3d reconstruction. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 91–96.

[Birchfield et al.98] Birchfield (S.) et Tomasi (C.). – Depth Discontinuities by Pixel-to-Pixel Stereo. *In : Proceedings of 6th International Conference on Computer Vision, Bombay (India)*, pp. 1073–1080.

[Blake et al.93] Blake (A.), Curwen (R.) et Zisserman (A.). – Affine-Invariant Contour Tracking with Automatic Control of Spatiotemporal Scale. *Fourth International Conference on Computer Vision*, 1993, pp. 66–75.

[Blanc94] Blanc (J.). – *Reconstruction 3D pour la Synthèse d'Images.* – Rapport de DEA, Institut National Polytechnique de Grenoble, July 1994.

[Bonnaud et al.94] Bonnaud (L.) et Labit (C.). – *Etude d'algorithme de suivi temporel de segmentation basée mouvement pour la compression de séquences d'images.* – Rapport de recherche 2253, INRIA, 1994.

[Bougnoux98] Bougnoux (S.). – From Projective to Euclidiean Space under any Practical Situation, a Criticism of Self-calibration. *In : Proceedings of 6th International Conference on Computer Vision, Bombay (India)*, pp. 790–796.

[Boyer et al.97] Boyer (Edmond) et Berger (Marie-Odile). – 3D Surface Reconstruction Using Occluding Contours. *International Journal of Computer Vision*, vol. 22 (3), March 1997, pp. 219–233.

[Bruzzone et al.92] Bruzzone (E.), Cazzanti (M.), Floriani (L. De) et Mangili (F.). – Applying Two–Dimensional Delaunay Triangulation to Stereo Data Interpolation. *In : Proceedings of Computer Vision (ECCV '92)*, éd. par Sandini (Giulio). pp. 368–372. – Berlin, Germany, mai 1992.

[Canny86] Canny (J.). – A Computational Approach to Edge Detection. *IEEE Transactions on PAMI*, vol. 8 (6), 1986, pp. 679–698.

[Caprile et al.90] Caprile (B.) et Torre (V.). – Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, vol. 4, 1990, pp. 127–140.

[caselles et al.92] caselles (V.), Catte (F.), Coll (T.) et Dibos (F.). – *A Geometric Model for Active Contours in Image Processing.* – Technical Report 9210, Ceremade, 1992. Cahier de mathématiques de la décision.

[Chen et al.98] Chen (M.), Kanade (T.), Rowley (H.) et D.Pommerleau. – Anomaly Detection through Registration. *In : IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998.*

[Chevrier et al.95] Chevrier (C.), Belblidia (S.) et Paul (J.-C.). – Compositing Computer Generated Images and Video Films: An application for Visual Assessment in Urban Environments. *In : Computer Graphics: Development in Virtual Environments*, éd. par Earnshaw (R. A.) et Vince (J. A.), pp. 115–125. – Academic Press, June 1995.

[Chew87] Chew (L. P.). – Constrained Delaunay Triangulations. *In : Proceedings of the 3rd Annual Symposium on Computational Geometry (SCG '87)*, éd. par Wood (Derick). pp. 215–222. – Waterloo, ON, Canada, June 1987.

[Cohen91] Cohen (L.). – On Active Contours Models and Balloons. *Computer Vision, Graphics and Image Processing*, vol. 53 (2), 1991, pp. 211–218.

[Cohen92] Cohen (I.). – *Modèles déformables 2D et 3D: application à la segmentation d'images médicales.* – PhD thesis, Université de Paris Dauphine, UER mathématiques de la décision, 1992.

[Cox et al.96] Cox (Ingemar J.), Hingorani (Sunita L.), Rao (Satish B.) et Maggs (Bruce M.). – A Maximum Likelihood Stereo Algorithm. *Computer Vision and Image Understanding: CVIU*, vol. 63 (3), May 1996, pp. 542–567.

[Csurka et al.97] Csurka, Zeller (C.), Zhang (Z.Y.) et Faugeras (O.D.). – Characterizing the Uncertainty of the Fundamental matrix. *Computer Vision and Image Understanding*, vol. 68 (1), May 1997, pp. 18–36.

[Curless et al.96] Curless (B.) et Levoy (M.). – A Volumetric Method for Building Complex Models from Ranges Images. *In : Computer Graphics (Proceedings Siggraph New Orleans)*, pp.??–??

[Curtis et al.98] Curtis (D.), Mizell (D.), Gruenbaum (P.) et Janin (A.). – Several Devils in the Details: Making an AR App Work in the Airplanee Factory. *In : First International Workshop on Augmented Reality, San Francisco.*

[Debevec et al.96] Debevec (P. E.), Taylor (C. J.) et Malik (J.). – Modeling and Rendering Architecture from Photographs. *In : Proc. SIGGRAPH.*

[Debevec98] Debevec (P.). – Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography. *Computer Graphics*, vol. 32 (Annual Conference Series), August 1998, pp. 189–198.

[Delamarre et al.99] Delamarre (Q.) et Faugeras (O.). – 3d articulated models and multiview tracking with silhouettes. *In : Proceedings of 7th International Conference on Computer Vision, Greece*, pp. 716–721.

[Dementhon et al.95] Dementhon (D.) et Davis (L.). – Model Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*, vol. 15, 1995, pp. 123–141.

[Deriche et al.90] Deriche (R.) et Faugeras (O.). – Tracking line segments. *Image and Vision Computing*, vol. 8 (4), November 1990, pp. 261–270.

[Deriche87] Deriche (R.). – Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, vol. 1 (2), 1987, pp. 167–187.

[Devernay et al.94] Devernay (F.) et Faugeras (O. D.). – Computing Differential Properties of 3D Shapes from Stereoscopic images Without 3D Models. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA (USA)*, pp. 208–213.

[Dhome et al.89] Dhome (M.), Richetin (M.), Lapresté (J.T.) et Rives (G.). – Determination of the Attitude of 3-D Objects from a Single Perspective View. *IEEE Transactions on PAMI*, vol. 11 (12), 1989, pp. 1265–1278.

[Drettakis et al.97] Drettakis (G.), Robert (L.) et Bougnoux (S.). – Interactive Common Illumination for Computer Augmented Reality. *In : 8th Eurographics workshop on Rendering, St. Etienne, France.*

[Egnal et al.00] Egnal (G.) et Wildes (R.P.). – Detecting Binocular Half-Occlusions: Empirical Comparisons of Four approaches. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (USA)*, pp. 466–273.

[Faugeras et al.92] Faugeras (O. D.), Luong (Q.-T.) et Maybank (S. J.). – Camera Self-Calibration: Theory and Experiments. *In : Proceedings of 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 321–334.

[Faugeras et al.97] Faugeras (O.) et Keriven (R.). – Level Set Methods and the Stereo Problem. *Lecture Notes in Computer Science*, vol. 1252, 1997, pp. 272–??

[Faugeras93] Faugeras (O.). – *Three-Dimensional Computer Vision: A Geometric Viewpoint.* – MIT Press, 1993, *Artificial Intelligence.*

[Faugeras98] Faugeras (O.). – De la géométrie au calcul variationnel : théorie et applications de la vision tridimensionnelle. *In : Actes du 11e Congrès de Reconnaissance des Formes et Intelligence Artificielle (RFIA'98), Clermont-Ferrand*, pp. 15–34.

[Feiner et al.93] Feiner (S.), MacIntyre (B.) et Seligmann (D.). – Knowledge-based Augmented Reality. *Communications of the ACM*, vol. 36 (7), July 1993, pp. 52–62.

[Feldmar et al.97] Feldmar (J.), Ayache (N.) et Betting (F.). – 3D-2D Projective Registration of Free Form Curves and Surfaces. *Computer Vision and Image Understanding*, vol. 65 (3), 1997, pp. 403–424.

[Ferri et al.93] Ferri (M.), Mangili (F.) et Viano (G.). – Projective Pose Estimation of Linear and Quadratic Primitives in Monocular Computer Vision. *CVGIP: Image Understanding*, vol. 58 (1), July 1993, pp. 66–84.

[Fitzgibbon et al.98] Fitzgibbon (A.W.) et Zisserman (A.). – Automatic Camera Recovery for Closed or Open Images Sequences. *In : Proceedings of 5th European Conference on Computer Vision, University of Freiburg (Germany)*, pp. 311–326.

[Fournier et al.93] Fournier (A.), Gunawan (A. S.) et Romanzin (C.). – Common Illumination between Real and Computer Generated Scenes. *In : Proceedings of Graphics Interface'93*, pp. 254–262.

[Fua et al.89] Fua (P.) et Leclerc (Y.). – Model Driven Edge Detection. *Machine Vision and Applications*, 1989.

[Fua et al.96] Fua (P.) et Leclerc (Y.). – Taking Advantage of Image-Based and Geometry-Based Constraints to Recover 3–D Surfaces. *Computer Vision and Image Understanding*, vol. 64 (1), July 1996, pp. 111–127.

[Geiger et al.95] Geiger (D.), Ladendorf (B.) et Yuille (A.). – Occlusions and Binocular Stereo. *International Journal of Computer Vision*, vol. 14, 1995, pp. 211–226.

[Gennery92] Gennery (D.). – Visual Tracking of Known Three Dimensional Objects. *International Journal of Computer Vision*, vol. 7 (3), 1992, pp. 243–270.

[Gibbs et al.96] Gibbs (S.) et Baudisch (P.). – Interaction in the virtual studio. *Computer Graphics*, vol. 30 (4), November 1996, pp. 29–??

[Goncalves00] Goncalves (M.). – RealViz Image Modeler 2. *Pixel*, no54, November 2000, pp. 24–25.

[Haralick et al.89] Haralick (R. M.), Joo (H.), Lee (C. N.), Zhuang (X.), Vaidya (V.G.) et Kim (M. B.). – Pose Estimation from Corresponding Point Data. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19 (6), 1989.

[Harris et al.88] Harris (C.) et Stephens (M.). – A Combined Corner and Edge Detector. *In : Proceedings of 4th Alvey Conference.* – Cambridge, August 1988.

[Hartley et al.00] Hartley (R. I.) et Zisserman (A.). – *Multiple View Geometry in Computer Vision.* – Cambridge University Press, ISBN: 0521623049, 2000.

[Hartley94] Hartley (R. I.). – Euclidean reconstruction from uncalibrated views. *Lecture Notes in Computer Science*, vol. 825, 1994, pp. 237–256.

[Hartley97] Hartley (R. I.). – Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, vol. 22 (2), March 1997, pp. 125–140.

[Huber81] Huber (P. J.). – *Robust Statistics.* – Wiley, New York, 1981.

[Intille et al.94] Intille (S. S.) et Bobick (A. F.). – Disparity-Space Images and Large Occlusion Stereo. *In : European Conference on Computer Vision*, éd. par Eklundh (Jan-Olof). pp. 179–186. – Stockholm, Sweden, May 1994.

[Kanade et al.94] Kanade (T.) et Okutomi (M.). – A Stereo Matching Algorithm with an Adaptative Window: Theory and Experiment. *IEEE Transactions on PAMI*, vol. 16 (9), September 1994, pp. 920–932.

[Kanade et al.95] Kanade (T.), Oda (K.), Yoshida (A.), Tanaka (M.) et Kano (H.). – *Video-Rate Z Keying: A New Method fo Merging Images.* – CMU-RI-TR 38, The Robotics Institute, 1995.

[Kanade et al.96] Kanade (T.), Yoshida (A.), Oda (K.), Kano (H.) et Tanaka (M.). – A Video-Rate Stereo Machine and Its New Applications. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA (USA).*

[Kanade et al.99] Kanade (T.), Rander (P.), Vedula (S.) et Saito (H.). – Virtualized Reality: Digitizing a 3D Time-Varying Event As Is and in Real Time. *In : Proc. ISMR'99 (International Symposium on Mixed Reality)*, pp. 41–57. – Yokohama, Japan, March 1999.

[Kass et al.88] Kass (M.), Witkin (A.) et Terzopoulos (D.). – Snakes: Active Contour Models. *International Journal of Computer Vision*, vol. 1, 1988, pp. 321–331.

[Kato et al.99] Kato (H.) et Billinghurst (M.). – Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System.

*In : Proceedings of the 2nd International Workshop on Augmented Reality, San Francisco.*

[Kerrien et al.99] Kerrien (E.), Berger (M.-O.), Mauricomme (E.), Launay (L.), Vaillant (R.) et Picard (L.). – Fully Automatic 3D/2D Substracted Angiography Registration. *In : Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, pp. 664–671.

[Kervrann et al.94] Kervrann (C.) et Heitz (F.). – A Hierarchical Statistical Framework for the Segmentation of Deformable Objects in Image Sequences. *In : Proceedings of the Conference on Computer Vision and Pattern Recognition.* pp. 724–728. – Los Alamitos, CA, USA, June 1994.

[Klinker et al.97] Klinker (G. J.), Ahlers (K. H.), Breen (D. E.), Chevalier (P.-Y.), Crampton (C.), Greer (D. S.), Koller (D.), Kramer (A.), Rose (E.), Tuceryan (M.) et Whitaker (R. T.). – Confluence of Computer Vision and Interactive Graphics for Augmented Reality. *Presence: Teleoperators and Virtual Environments*, vol. 6 (4), August 1997, pp. 433–451.

[Koch et al.98] Koch (R.), Pollefeys (M.) et Van Gool (L.). – Multi Viewpoint Stereo from Uncalibrated Video Sequences. *Lecture Notes in Computer Science*, vol. 1406, 1998, pp. 55–??

[Koller et al.92] Koller (D.), Daniilidis (K.) et Nagel (H. H.). – Model-Based Object Tracking in Traffic Scenes. *In : Proceedings of 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 437–452.

[Kriegman et al.90] Kriegman (D.) et Ponce (J.). – On Recognizing and Positioning Curved 3D Objects from Image Contours. *IEEE Transactions on PAMI*, vol. 12 (12), December 1990, pp. 1127–1137.

[Kumar et al.94] Kumar (R.) et Hanson (A.). – Robust Methods for Estimating Pose and a Sensitivity Analysis. *CVGIP: Image Understanding*, vol. 60 (3), 1994, pp. 313–342.

[Kutulakos et al.94] Kutulakos (K.) et Dyer (C.). – Occluding Contour Detection Using Affine Invariants and Purposive Viewpoint Control . *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA (USA).*

[Kutulakos et al.98] Kutulakos (K. N.) et Seitz (S. M.). – *A Theory of Shape by Space Carving.* – Technical Report TR692, University of Rochester, Computer Science Department, May 1998.

[Kutulakos00] Kutulakos (K. N.). – Approximate N-View Stereo. *In : Proceedings of 6th European Conference on Computer Vision, Trinity College Dublin (Ireland).*

[Lamberti et al.93] Lamberti (C.), Botazzi (P.) et Sarti (A.). – Region Based Matching Field Computation in 2D Echocardiography. *In : Computers in Cardiology, 1993*, pp. 739–742.

[Lepetit et al.01] Lepetit (V.) et Berger (M.-O.). – An intuitive tool for outlining objects in video sequences : Applications to augmented and diminished reality (to appear). *In : Proceedings of International Symposium of Mixed Reality, Yokohama (Japan).*

[Liebowitz et al.99] Liebowitz (D.), Criminisi (A.) et Zisserman (A.). – Creating Ar-

chitectural Models from Images. *In : EUROGRAPHICS'99, Milano, Italy.*

[Little et al.90] Little (J. J.) et Gillett (W. E.). – Direct Evidence for Occlusion in Stereo and Motion. *In : Proceedings of 1st European Conference on Computer Vision, Antibes (France)*, pp. 336–340.

[Loscos et al.00] Loscos (C.), Drettakis (G.) et Robert (L.). – Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6 (3), 2000.

[Luo et al.95] Luo (H.), Agam (G.) et Dinstein (I.). – Directional Mathematical Morphology Approach for Line Thinning and Extraction of Character Strings from Maps and Line Drawings. *In : Proceedings of 3rd International Conference on Document Analysis and Recognition, Montréal (Canada)*, pp. 257–260.

[Luong92] Luong (Q. T.). – *Matrice fondamentale et calibration visuelle sur l'environnement, vers une plus grande autonomie des systèmes robotiques.* – Thèse de doctorat, Université de Paris Sud, centre d'Orsay, December 1992.

[Malladi et al.95] Malladi (R.), Sethian (J.) et Vemuri (B.). – Shape Modeling with Front Propagation: A level Set Approach. *IEEE Transactions on PAMI*, vol. 2 (17), 1995, pp. 158–175.

[Marr et al.76] Marr (D.) et Poggio (T.). – Cooperative Computation of Stereo Disparity. *Science*, vol. 194, 1976, pp. 283–287.

[Maver et al.85] Maver (T. W.), Purdie (C.) et Stearn (D.). – Visual Impact Analysis — Modelling and Viewing the Natural and Built Environment. *Comput. & Graphics*, vol. 9 (2), 1985, pp. 117–124.

[Maybank et al.92] Maybank (S. J.) et Faugeras (O. D.). – A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, vol. 8 (2), 1992, pp. 123–152.

[Mclauchlan00] McLauchlan (P. F.). – A batch/recursive algorithm for 3d scene reconstruction. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (USA)*, pp. 738–743.

[Metacreations00] MetaCreations. – http://www.metacreations.com.

[Meyer et al.92] Meyer (F.) et Bouthemy (P.). – Region Based Tracking in an image sequence. *In : Proceedings of 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 476–483.

[Morris et al.00] Morris (D.) et Kanade (T.). – Image-Consistent Surface Triangulation. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (USA)*, pp. 332–338.

[Mortensen et al.95] Mortensen (E.) et Barrett (W.). – Intelligent Scissors for Image Composition. *In : Computer Graphics (Proceedings Siggraph)*, pp. 191–198.

[Mortensen et al.00] Mortensen (E.), Reese (L.) et Barrett (W.). – Intelligent selection tools. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (USA).*

[Narayanan et al.98] Narayanan (P.J.), Rander (P.W.) et Kanade (T.). – Constructing Virtual Worlds Using Dense Stereo. *In : Proceedings of 6th International Conference on Computer Vision, Bombay (India)*, pp. 3–10.

[Neuenschwander et al.95] Neuenschwander (W.), Fua (P.), Szekely (G.) et Kubler (O.). – From Ziplock Snakes to Velcro Surfaces. *In : Ascona'95.*

[Ohshima et al.99] Ohshima (T.), Satoh (K.), Yamamoto (H.) et Tamura (H.). – RV-Border Guards: A Multi-player Entertainment in Mixed Reality Space. *In : Proceedings of the 2nd International Workshop on Augmented Reality, San Francisco.*

[Ohta et al.85] Ohta (Y.) et Kanade (T.). – Stereo by Intra- and Inter-Scanline Search. *IEEE Transactions on PAMI*, vol. 7 (2), March 1985, pp. 139–154.

[Oisel et al.00] Oisel (L.), Memin (E.) et Morin (L.). – Geometric Driven Optical Flow Estimation and Segmentation for 3D Reconstruction. *In : Proceedings of 6th European Conference on Computer Vision, Trinity College Dublin (Ireland)*, pp. 849–863.

[Oisel98] Oisel (L.). – *Reconstruction 3D de Scènes Complexes à artir de Séquences Vidéo Non Calibrées: Estimation et Maillage d'un Champ de Disparité.* – Thèse de doctorat, Université de Rennes I, November 1998.

[Okutomi et al.93] Okutomi (M.) et Kanade (T.). – A Multiple-Baseline Stereo. *IEEE Transactions on PAMI*, vol. 15 (4), 1993, pp. 353–63.

[Ong et al.98] Ong (K. C.), Teh (H. C.) et Tan (T. S.). – Resolving Occlusion in Image Sequence Made Easy. *The Visual Computer*, vol. 14, 1998, pp. 153–165.

[Orad00] Orad. – http://www.orad.co.il.

[Park et al.98] Park (J. I.) et Inoue (S.). – Real-image-based virtual studio. *In : Proceedings of the 1st International Conference on Virtual Worlds (VW-98)*, éd. par Heudin (Jean-Claude). pp. 117–122. – Berlin, July 1–3 1998.

[Plantinga et al.90] Plantinga (H.) et Dyer (C.). – Visibility, Occlusion, and the Aspect Graph. *International Journal of Computer Vision*, vol. 5 (2), 1990, pp. 137–160.

[Press et al.88] Press (W. H.), Flannery (B. P.), Teukolsky (S. A.) et Vetterling (W. T.). – *Numerical Recipes in C, The Art of Scientific Computing.* – Cambridge University Press, 1988.

[Ravela et al.96] Ravela (S.), Draper (B.), Lim (J.) et Weiss (R.). – Tracking Object Motion Across Aspect Changes for Augmented Reality. *In : ARPA Image Undertanding Worshop, Palm Spring (USA).*

[RD94] Renault-Design. – RACOON. *In : Siggraph 94 Screening Room - Issue 102.*

[Realviz00] Realviz. – http://www.realviz.com.

[Reiners et al.98] Reiners (D.), Stricker (S.), Klinker (G.) et Müller (S.). – Augmented Reality for Construction Tasks: Doorlock Assembly. *In : First International Workshop on Augmented Reality, San Francisco.*

[Rose et al.94] Rose (E.), Breen (D.), Ahlers (K.), Crampton (C.), Tuceyran (M.), Whitaker (R.) et Greer (D.). – *Annotating Real-World Objects Using Augmented Reality.* – Technical report, ECRC, Munich, 1994.

[Roth et al.99] Roth (M.), Brack (C.), Burgkart (R.), Czopf (A.), Götte (H.) et Schweikard (A.). – Multi-view contourless registration of bone structures using a single calibrated X-ray fluoroscope. *In : Computer Assisted Radiology and Surgery (CARS'99)*, pp. 756–761.

[Rothwell95] Rothwell (C.). – *The Importance of Reasoning about Occlusions during Hypothesis Verification in Object Recognition.* – Rapport de recherche 2673, INRIA, 1995.

[Rousseeuw et al.87] Rousseeuw (Peter J.) et Leroy (Annick M.). – *Robust Regression and Outlier Detection.* – John Wiley and Sons, 1987, *Wiley Series in Probability and Mathematical Statistics.*

[Samadani91] Samadani (R.). – Adaptative Snakes: Control of Damping and Material parameters. *In : Proceedings SPIE 1991 International Symposium on Optical Applied Science and Engineering, Geometric Methods in Computer Vision, San Diego, California*, pp. 202–213.

[Sato et al.99] Sato (I.), Sato (Y.) et Ikeuchi (K.). – Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene. *IEEE Transactions on Visualization and Computer Graphics*, vol. 5 (1), March 1999.

[Schmid et al.97] Schmid (C.) et Mohr (R.). – Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on PAMI*, vol. 19 (5), August 1997, pp. 530–535.

[Schmitt00] Schmitt (P.). – Escapade. *Pixel*, no51, May 2000, p. 47.

[Seitz et al.97] Seitz (S. M.) et Dyer (C. R.). – Photorealistic Scene Reconstruction by Voxel Coloring. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 1067–1073.

[Seitz et al.99] Seitz (S.) et Dyer (C.). – Photorealistic Scene Reconstruction by Voxel Coloring. *International Journal of Computer Vision*, vol. 35 (2), 1999, pp. 151–173.

[Sethian96] Sethian (J. A.). – *Level Set Methods.* – Cambridge University Press, 1996.

[Shakunaga93] Shakunaga (T.). – Robust Line Based Pose Enumeration From a Single Image. *In : Proceedings of 4th International Conference on Computer Vision, Berlin (Germany)*, pp. 545–550.

[Simon et al.99] Simon (G.) et Berger (M.-O.). – Registration with a Zoom Lens Camera for Augmented Reality Applications. *In : Second International Workshop on Augmented Reality, San Francisco.*

[Simon et al.00] Simon (G.) et Berger (M.-O.). – Registration with a Zoom Lens Camera for Augmented Reality Applications. *In : Proceedings of 6th European Conference on Computer Vision, Trinity College Dublin (Ireland).*

[Simon95] Simon (G.). – *Détermination du point de vue à partir d'une observation d'un objet 3D dont le modèle est connu.* – Rapport de DEA, Université Henri Poincaré Nancy I, September 1995.

[Simon99] Simon (G.). – *Vers un système de Réalité Augmentée autonome.* – Vandœuvre-lès-Nancy, Thèse de doctorat, Université Henri Poincaré Nancy I, December 1999.

[Smitley et al.84] Smitley (D. L.) et Bajcsy (R.). – Stereo Processing of Aerial, Urban Images. *In : Seventh International Conference on Pattern Recognition (Montreal, Canada, July 30-August 2, 1984).* IEEE, pp. 433–435. – IEEE.

[State et al.96] State (A.), Livingston (M. A.), Hirota (G.), Garrett (W. F.), Whitton (M. C.) et Fuchs (H.). – Technologies for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. *In : Siggraph Conference Proceedings.* ACM Siggraph, pp. 439–446.

[Sullivan et al.98] Sullivan (S.) et Ponce (J.). – Automatic Model Construction and Pose Estimation From Photographs Using Triangular Splines. *IEEE Transactions on PAMI*, vol. 20 (10), October 1998, pp. 1091–1097.

[Symahvision00] SymahVision. – Epsis http://www.epsis.com.

[Szeliski et al.96] Szeliski (R.) et Kang (S. B.). – Shape Ambiguities in Structure from Motion. *Lecture Notes in Computer Science*, vol. 1064, 1996, pp. 709–??

[Szeliski93] Szeliski (R.). – Rapid Octree Construction from Image Sequences. *CVGIP: Image Understanding*, vol. 58 (1), July 1993, pp. 23–32.

[Thalmann et al.97] Thalmann (N. M.) et Thalmann (D.). – Animating Virtual Actors in Real Environments. *ACMMS'97*, vol. 5 (2), 1997, pp. 113–125.

[Torr et al.97] Torr (P.H.S.) et Zisserman (A.). – Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, vol. 15, 1997, pp. 591–605.

[Torre et al.00] Torre (R.), Balcisoy (S.), Fua (P.), Ponder (M.) et Thalmann (D.). – Interaction Between Real and Virtual Humans : Playing Checkers. *In : Eurographics Workshop on Virtual Environments, Amsterdam, Netherlands.*

[Triggs et al.00] Triggs (W.), McLauchlan (P. F.), Hartley (R. I.) et Fitzgibbon (A.). – Bundle Adjustment for Structure from Motion. *In : Vision Algorithms: Theory and Practice.* – Springer-Verlag.

[Vaillant et al.92] Vaillant (R.) et Faugeras (O.). – Using Extremal Boundaries for 3-D Object Modeling. *IEEE Transactions on PAMI*, vol. 14 (2), February 1992, pp. 157–173.

[Vintsyuk68] Vintsyuk (T. K.). – Speech Discrimination by Dynamic Programming. *Kibernetica*, vol. 1, 1968, pp. 81–88.

[Watson81] Watson (D. F.). – Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes. *The Computer Journal*, vol. 24 (2), 1981, p. 167.

[Weng et al.88] Weng (J.), Ahuja (N.) et Huang (T. S.). – Two-View Matching. *In : Second International Conference on Computer Vision (Tampa, FL, December 5–8, 1988).* pp. 64–73. – Washington, DC, 1988.

[Weng et al.92] Weng (J.), Ahuja (N.) et Huang (T.S.). – Matching Two Perspective Views. *IEEE Transactions on PAMI*, vol. 14 (8), 1992, pp. 806–825.

[Wildes91] Wildes (R. P.). – Direct Recovery of Three-Dimensional Scene Geometry from Binocular Stereo Disparity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13 (8), August 1991, pp. 761–774.

[Wloka et al.95] Wloka (M.) et Anderson (B.). – Resolving Occlusions in Augmented Reality. *In : Symposium on Interactive 3D Graphics Proceedings, (New York)*, pp. 5–12.

[Wohn et al.91] Wohn (K.), Wu (J.) et Brockett (R.). – A Contour based Recovery of Image Flow: Iterative Transformation Method. *IEEE Transactions on PAMI*, vol. 13 (8), August 1991, pp. 746–760.

[Yi et al.97] Yi (X.) et Camps (O.). – Robust Occluding Contour Detection Using the Hausdorff Distance. *In : Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, PR (USA)*, pp. 962–968.

[Yuille et al.84] Yuille (A. L.) et Poggio (T.). – *A Generalized Ordering Constraint for Stereo Correspondence.* – Technical Report AIM-777, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, May 1984.

[Zeller et al.96] Zeller (Cyril) et Faugeras (Olivier). – *Camera Self-Calibration from Video Sequences: the Kruppa Equations Revisited.* – Rapport de recherche 2793, INRIA, February 1996.

[Zhang et al.94] Zhang (Z.), Deriche (R.), Faugeras (O.) et Luong (Q.). – *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry.* – Rapport de recherche 2273, INRIA, 1994.

[Zhang et al.95] Zhang (Z.), Deriche (R.), Faugeras (O.) et Luong (Q.). – A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*, vol. 78, October 1995, pp. 87–119.

[Zisserman et al.99] Zisserman (A.), Fitzgibbon (A.) et Cross (G.). – VHS to VRML: 3D graphical models from video sequences. *In : Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia.*

[Zoghlami et al.96] Zoghlami (I.), Faugeras (O.) et Deriche (R.). – Traitement des occlusions pour la modification d'objet plan dans une séquence d'image. *In : ORASIS'96 - Journées francophones des jeunes chercheurs en analyse d'images et perception visuelle.*