



HAL
open science

Suivi Hybride en présence d'Occultations pour la Réalité Augmentée

Madjid Maidi

► **To cite this version:**

Madjid Maidi. Suivi Hybride en présence d'Occultations pour la Réalité Augmentée. Sciences de l'ingénieur [physics]. Université d'Evry-Val d'Essonne, 2007. Français. NNT: . tel-00375685

HAL Id: tel-00375685

<https://theses.hal.science/tel-00375685>

Submitted on 15 Apr 2009

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

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

UNIVERSITÉ D'ÉVRY VAL D'ESSONNE
ÉCOLE DOCTORALE SITEVRY

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ D'ÉVRY

Spécialité : ROBOTIQUE

présentée et soutenue publiquement

par

Madjid MAIDI

le 14 novembre 2007

Titre

*Suivi Hybride en présence d'Occultations pour la
Réalité Augmentée*

—————
Directeur de thèse : Malik MALLEM
—————

JURY

M. Éric Marchand	Chargé de recherche INRIA (IRISA), HDR Université de Rennes 1	Rapporteur
M. Didier Stricker	Directeur du Fraunhofer Institute for Computer Graphics (Darmstadt)	Rapporteur
M. Jean-Marc Lavest	Prof. Université d'Auvergne (Clermont-Ferrand)	Examineur
M. Florent Chavand	Prof. Université d'Évry Val d'Essonne	Examineur
M. Malik Mallem	Prof. Université d'Évry Val d'Essonne	Examineur
M. Fakhreddine Ababsa	MdC Université d'Évry Val d'Essonne	Examineur

Table des matières

Table des matières	3
Table des figures	9
Liste des tableaux	13
Remerciements	15
Introduction générale	17
1 Réalité Augmentée : État de l'art	19
1.1 Introduction	19
1.2 Les applications de la réalité augmentée	21
1.2.1 Médical	21
1.2.2 Industrie-maintenance	23
1.2.3 Visualisation	25
1.2.4 Téléopération	26
1.2.5 Divertissement	27
1.2.6 Militaire	28
1.2.7 Architecture	28
1.3 Dispositifs et capteurs utilisés en réalité augmentée	29
1.3.1 Systèmes de visualisation	29
1.3.2 Capteurs de position et d'orientation	31
1.3.3 Les systèmes de tracking visuel	33
1.4 Problèmes liés à la réalité augmentée	34
1.4.1 Alignement spatial : erreurs statiques	34
1.4.2 Alignement temporel : temps de latence	34
1.4.3 Éclairage	35
1.4.4 Occultation	35
1.4.5 Flou de mouvement	36
1.5 Le tracking dans les systèmes de réalité augmentée	36
1.5.1 Systèmes de tracking en vision	36
1.5.1.1 Tracking temps réel	36
1.5.1.2 Asservissement visuel virtuel	37
1.5.1.3 Travaux récents en tracking visuel	37
1.5.2 Tracking pour les applications de réalité augmentée	37
1.5.2.1 Tracking des cibles passives	37
1.5.2.2 Tracking des cibles actives	38
1.5.3 Bilan sur les méthodes de tracking	38
1.6 Travaux récents dans la réalité augmentée	39
1.6.1 Travaux récents en vision indirecte au sein du laboratoire IBISC	39
1.6.1.1 Travaux de Ababsa	39
1.6.1.2 Travaux de Merad	40

1.6.2	Travaux récents en vision directe	40
1.6.2.1	Travaux de Foxlin et al.	41
1.6.2.2	Travaux de Foxlin et Naimark	41
1.6.2.3	Travaux de Chai et al.	43
1.6.2.4	Travaux de You et Neumann	44
1.7	Contexte	45
1.8	Motivations	46
1.9	Objectif et solutions proposées	46
1.10	Conclusion	48
2	Méthodes basées vision pour l'estimation de la pose	49
2.1	Introduction	49
2.2	Position du problème	49
2.3	Mise en correspondance 2D-3D	51
2.3.1	Identification de cibles codées	53
2.3.1.1	Méthodes d'identification de cibles en réalité augmentée	53
2.3.1.2	Approche développée	56
2.3.2	Suivi sans-marqueur	61
2.3.2.1	Contours actifs	62
2.3.2.2	Suivi d'objets avec les contours actifs	65
2.3.2.3	Amélioration du suivi par le flot optique	66
2.3.2.4	Résultats et analyse de performances	69
2.3.2.5	Analyse et critique du suivi par contours actifs	69
2.3.3	Bilan sur les méthodes de reconnaissance de cible	73
2.4	Estimation de la pose	73
2.4.1	Algorithme analytique	74
2.4.2	Itération orthogonale	74
2.4.3	Filtre de Kalman étendu	76
2.4.3.1	Prédiction	77
2.4.3.2	Correction	77
2.4.4	Filtre de Kalman étendu hybride	78
2.4.5	Résultats expérimentaux	79
2.4.5.1	Temps d'exécution	79
2.4.5.2	Erreur de reconstruction	80
2.4.5.3	Erreur de généralisation	81
2.4.5.4	Estimation de la distance réelle entre la caméra et la cible	82
2.4.5.5	Résultats de recalage	86
2.4.5.6	Éléments de comparaison avec d'autres études équivalentes	88
2.4.6	Discussion	89
2.5	Conclusion	90
3	Gestion d'occultations en réalité augmentée	91
3.1	Introduction	91
3.2	Les estimateurs robustes	91
3.3	Domaines d'applications en vision	93
3.4	Contraintes liées au suivi de points d'intérêts	94
3.4.1	La contrainte du mouvement	94
3.4.2	L'unicité de la mise en correspondance	94
3.4.3	Caractéristiques des primitives	95
3.4.4	Le suivi robuste des points d'intérêts	95

3.5	La méthode RANSAC	95
3.6	Gestion d'occultations en réalité augmentée	98
3.7	Notre méthode de suivi robuste	102
3.7.1	Conditions de fonctionnement	103
3.7.2	Détection des points d'intérêts	104
3.7.3	Mise en correspondance 2D-2D	104
3.7.4	Mise en correspondance 3D-2D	105
3.7.5	Mise en correspondance 3D-2D et 2D-2D	105
3.7.6	Mise en correspondance robuste	106
3.7.7	Résumé de la méthode de suivi robuste	106
3.8	Résultats expérimentaux	107
3.8.1	Conditions expérimentales	107
3.8.2	L'algorithme	107
3.8.3	Suivi de points	109
3.8.4	Occultations partielles	109
3.8.5	Changement de luminosité	111
3.8.6	Changement d'échelle	111
3.8.7	Changement d'orientation	114
3.8.8	Temps d'exécution	115
3.8.9	Erreur de reconstruction	116
3.8.10	Discussion	117
3.9	Conclusion	118
4	Suivi hybride en réalité augmentée	119
4.1	Introduction	119
4.2	Systèmes vision-inertiels existants	120
4.3	Calibration du capteur hybride	121
4.3.1	Description	121
4.3.2	Les repères utilisés	121
4.3.3	Procédure de calibration	122
4.3.3.1	La centrale inertielle	122
4.3.3.2	La caméra	123
4.3.4	Résolution de la rotation R_{CI}	124
4.4	Modélisation du mouvement	125
4.4.1	Vitesse linéaire	125
4.4.2	Accélération	126
4.4.3	Vitesse angulaire	126
4.4.4	Prédiction de l'orientation	127
4.4.5	Prédiction de la translation	128
4.5	Suppléance de données	128
4.5.1	Module d'orientation	130
4.5.1.1	Facteur de pondération	130
4.5.1.2	Gain du filtre	130
4.5.1.3	Facteur d'adaptation du champ magnétique	130
4.5.2	Module des données calibrées	131
4.5.2.1	Modèle physique du capteur	131
4.5.3	Le filtre de Kalman pour la translation	131
4.5.3.1	Prédiction	132
4.5.3.2	Correction	132
4.6	Résultats expérimentaux	133

4.6.1	Protocole expérimental	133
4.6.2	Caractérisation de la centrale inertielle : MTi	133
4.6.2.1	Données de la MTi : rotations et données calibrées	133
4.6.2.2	Filtrage des données	135
4.6.2.3	Soustraction de la gravité	135
4.6.3	Suivi hybride	137
4.6.3.1	Calibration de l'orientation	137
4.6.3.2	Calibration de la translation	138
4.6.3.3	Paramètres du filtre de Kalman pour la translation	139
4.6.3.4	Évaluation du filtre avec des données synthétiques	140
4.6.3.5	Évaluation du filtre avec des données réelles	147
4.6.4	Résultats de suivi et de recalage	152
4.6.4.1	Cible visible	152
4.6.4.2	Cible occultée	155
4.6.4.3	Flou de mouvement	156
4.6.4.4	Gestion d'occultations partielles et totales de la cible	157
4.7	Conclusion	159
	Conclusion générale	161
	Bibliographie	165
	Annexes	175
A	Calibration de la caméra par la méthode de Zhang	175
A.1	Introduction	175
A.2	Équations de base	175
A.2.1	Notations	175
A.2.2	L'homographie entre le modèle planaire et son image	176
A.2.3	Contraintes des paramètres intrinsèques	176
A.2.4	Interprétation géométrique	176
A.3	Résolution des paramètres de la calibration	177
A.3.1	Solution analytique	177
A.3.2	Estimation par maximum de vraisemblance	179
A.3.3	Les distortions radiales	179
A.3.3.1	Estimation des distortions radiales par alternance	179
A.3.3.2	Estimation par maximum de vraisemblance globale	179
A.4	Résumé	180
B	Filtre de Kalman	181
B.1	Le filtre de Kalman discret	181
B.1.1	Le processus à estimer	181
B.1.2	Les origines calculatoires du filtre	181
B.1.3	Algorithme du filtre de Kalman discret	182
B.2	Filtre de Kalman étendu	183
B.3	Conclusion	186
C	Le filtre de Harris	187

D	Représentation des rotations	189
D.1	Matrice de rotation	189
D.2	Angles d'Euler	189
D.3	Quaternions	191

Table des figures

1.1	Sensorama Simulator.	19
1.2	Le continuum réel/virtuel de Milgram [109].	20
1.3	Le projet KARMA [45].	21
1.4	Visualisation à l'aide d'un visiocasque les images d'échographie [15].	22
1.5	Visualisation du ventre à l'aide du système de réalité augmentée développé à l'UNC [71].	22
1.6	Représentation virtuel de l'intérieur du sein [146].	23
1.7	Application de câblage d'un système électrique dans un prototype de Boeing [25] [76] [143].	23
1.8	Le système ARVIKA [10]. (a) Développement : système de réalité augmentée pour le crash-test. (b) Production : assistance visuelle pour l'utilisateur dans une tâche de maintenance. (c) Service : en cas de difficulté l'opérateur est relié au centre de service.	24
1.9	Exemple de vue en transparence fournie à l'opérateur - Projet AMRA [35].	25
1.10	Fenêtres virtuelles affichées sur des objets spécifiques du monde réel pour l'annotation [44].	25
1.11	Augmentation de la scène avec des lignes virtuelles [38].	26
1.12	Téléopération d'un robot réel en utilisant son prototype virtuel [37].	26
1.13	Téléopération du robot ARITI [122].	27
1.14	Le projet ALIVE : l'utilisateur donne des ordres à son chien [97].	27
1.15	Le jeu RV-Border : scène telle qu'elle est vue par le joueur [120].	28
1.16	Les casques utilisés dans les avions de combats. (a) HUD (Head Up Display). (b) HMS (Helmet Mounted Sight).	28
1.17	Les visiocasques. (a) Optical see through. (b) Video see through. (c) NOMAD.	29
1.18	Écran LCD 3D pour la vision stéréoscopique en réalité augmentée.	30
1.19	Écran portable à la main.	30
1.20	Capteurs magnétiques.	31
1.21	Capteur à ultrasons.	32
1.22	Capteur GPS.	32
1.23	Centrale inertielle MTi de Xsens [166].	33
1.24	Différents systèmes de tracking visuel utilisés en réalité augmentée.	33
1.25	Suivi d'objet avec la méthode de squelettisation [108].	40
1.26	Prototype du système de tracking VIS-Tracker d'Intersense [52]. (a) Le dispositif de tracking. (b) Utilisateur portant le système de tracking.	42
1.27	Évaluation de la robustesse du système [52]. (a) Marqueurs partiellement occultés. (b) Identification de cibles en variant la luminosité.	43
1.28	Système de tracking développé au CSM [26].	44
1.29	Visiocasque de réalité augmentée instrumenté de capteurs : caméras CCD, centrales inertielles, LEDs [26].	44
1.30	Capteur hybride : caméra et gyroscope [167].	45
1.31	Extraction des cibles de l'image [167].	45

1.32	Schéma synoptique du système de suivi et de gestion d'occultations.	47
2.1	Les repères utilisés pour l'estimation de la pose.	50
2.2	Repères géométriques associés à la calibration de la caméra.	51
2.3	Ordonnancement des points 2D dans l'image.	52
2.4	Processus de suivi et d'estimation de la pose.	52
2.5	Modèles de marqueurs utilisés par ARToolKit.	53
2.6	Processus de reconnaissance de cibles d'ARToolKit.	54
2.7	Processus de reconnaissance de cibles du CyberCode.	55
2.8	Exemple de cibles codées du système d'Intersense.	56
2.9	Processus de détection de cible. (a) Image originale. (b) Détection de contours. (c) Lissage de l'image. (d) Dilatation de l'image. (e) Approximation polygonale des contours.	57
2.10	Échantillonnage de la cible codée.	58
2.11	Identification de la cible.	58
2.12	Codes correspondant aux différentes orientations de la cible.	59
2.13	Identification de cible lors d'un changement d'intensité.	60
2.14	Identification de cible effectuant des rotations.	60
2.15	Identification de cible en effectuant un changement d'échelle. (a) Distance caméra-cible : 0.5m. (b) Distance caméra-cible : 1m. (c) Distance caméra-cible : 1.5m. (d) Distance caméra-cible : 2m.	61
2.16	Algorithme de Kass.	65
2.17	Évolution du snake pour la détection d'objets.	66
2.18	Suivi spatio-temporel avec le snake et le flot optique.	67
2.19	Suivi de cible par les contours actifs et le flot optique.	68
2.20	Suivi d'objet en effectuant une translation.	70
2.21	Suivi d'objet en effectuant une rotation.	70
2.22	Détection d'objet lors d'un changement d'échelle.	71
2.23	Détection d'objet subissant une occultation partielle.	71
2.24	Détection d'objet dans des conditions normales de luminosité.	72
2.25	Détection d'objet lors de modifications de l'intensité lumineuse.	72
2.26	Détection d'objet dans des conditions de luminosités dégradées.	73
2.27	Erreur de colinéarité de l'espace d'objet et l'espace d'image.	75
2.28	Diagramme d'estimation de la pose avec l'itération orthogonale hybride à l'instant t	76
2.29	Diagramme d'estimation de la pose avec le filtre de Kalman étendu hybride à l'instant t	78
2.30	Erreur de reconstruction en fonction des distances caméra-cible.	80
2.31	Cibles employées pour tester l'erreur de généralisation de chacun des algorithmes.	81
2.32	Erreur de généralisation en fonction de la distance inter-cibles.	82
2.33	Banc de calibration utilisé pour l'évaluation de la distance caméra-cible.	84
2.34	Erreur moyenne des données classifiées.	84
2.35	Variances des données classifiées.	85
2.36	Évaluation des distances mesurées par rapport aux distances réelles.	85
2.37	Résultats de recalage dans une séquence de tracking.	86
2.38	Généralisation de recalage sur un objet 3D en utilisant une cible coplanaire.	87
2.39	Généralisation de recalage sur un objet 3D en utilisant une cible 3D.	87
3.1	Détection des lignes avec la transformée de Hough. (a) Image originale. (b) Résultat de l'application de la transformée de Hough.	92
3.2	Ajustement d'une ligne avec les données correctes en utilisant RANSAC.	96
3.3	L'algorithme générique RANSAC.	97
3.4	Système de gestion d'occultation de Kiyokawa et al. [85].	98

3.5	Approche de gestion d'occultations en utilisant les contours [17].	99
3.6	Occultation des objets virtuels en mouvements [149].	99
3.7	Occultation des graphiques virtuels par l'opérateur [53].	100
3.8	Gestion d'occultations avec la technique de Fischer et Baratoff [48].	100
3.9	Suivi d'un objet partiellement occulté dans une application de maintenance [29]. . .	101
3.10	Suivi avec occultations multiples [30].	101
3.11	Schéma synoptique de la gestion d'occultations.	103
3.12	Détection de points avec le filtre de Harris.	104
3.13	Mise en correspondance des points. (a) Appariements 3D-2D. (b) Appariements 2D-2D.	106
3.14	Fonctionnement de l'algorithme robuste.	108
3.15	Étapes de traitement et de suivi de points avec la méthode robuste. (a) Cibles visibles. (b) Cibles partiellement occultées. (c) Appariements des points 2D. (d) Recalage des cubes sur les cibles.	109
3.16	Suivi de cibles en présence d'occultations partielles.	110
3.17	Échec de suivi de cibles.	110
3.18	Changement de luminosité lors d'un suivi robuste.	112
3.19	Suivi robuste en présence d'une forte luminosité.	112
3.20	Suivi robuste avec une forte diminution de luminosité.	113
3.21	Changement d'échelle lors d'un suivi robuste.	113
3.22	Changement d'orientation des cibles dans un suivi robuste.	114
3.23	Changement d'orientation de la caméra dans un suivi robuste.	115
3.24	Temps d'exécution de la méthode robuste en fonction du nombre de points visibles. .	116
4.1	Système de capteurs utilisé.	121
4.2	Représentation des repères utilisés.	122
4.3	Représentation des repères de la centrale inertielle : MTi.	123
4.4	Définition du nouveau repère global.	123
4.5	Estimation de la pose : rotation du repère monde par rapport au repère de la caméra. .	124
4.6	Représentation des rotations entre les repères utilisés.	124
4.7	Suppléance de données du capteur hybride.	129
4.8	Diagramme temporel de suppléance de données.	129
4.9	Données calibrées de la MTi.	134
4.10	Représentation des rotations de la MTi. (a) Angles d'Euler. (b) Quaternions.	134
4.11	Filtrage des signaux d'accélération.	136
4.12	Représentation de l'accélération de la MTi avant et après soustraction de la gravité. .	136
4.13	Les composantes du quaternion représentant la rotation R_{CI}	138
4.14	Simulation des accélérations constantes. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.	142
4.15	Simulation des accélérations constantes bruitées. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.	143
4.16	Simulation des accélérations sous forme d'un signal carré bruité. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.	145
4.17	Simulation des accélérations sinusoïdales bruitées. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.	146
4.18	Changement de fréquence de la MTi. (a) 25Hz. (b) 100Hz. (c) 200Hz. (d) 400Hz. . .	147
4.19	Changement de temps de correction des positions calculées par la MTi. (a) Correction après 40ms. (b) Correction après 200ms. (c) Correction après 480ms. (d) Correction après 600ms.	148
4.20	Dérive de la MTi en fonction du temps.	149

4.21	Suivi lors d'un mouvement uniforme.	150
4.22	Suivi lors d'un mouvement brusque. (a) Fréquence de la MTi : $25Hz$. (b) Fréquence de la MTi : $100Hz$	151
4.23	Résultats de suivi et de recalage avec le capteur hybride lorsque la cible est visible.	154
4.24	Erreurs d'estimation de la pose entre la caméra et la MTi. (a) Erreurs de translation. (b) Erreurs de rotation.	154
4.25	Résultats de suivi et de recalage avec le capteur hybride lorsque la cible est occultée.	155
4.26	Résultats de suivi et de recalage avec le capteur hybride en présence de flou de mouvement.	156
4.27	Suivi avec le capteur hybride et gestion d'occultations partielles et totales.	157
4.28	Diagramme de suivi et de gestion d'occultations avec le système global de tracking.	158
B.1	Filtre de Kalman linéaire.	183
B.2	Filtre de Kalman étendu.	185
C.1	Filtre de voisinage utilisé.	188
C.2	Classification des courbures suivant les valeurs propres de M	188
D.1	Représentation des angles d'Euler.	190
D.2	Représentation des angles de Bryan.	190
D.3	Représentation de la rotation avec un quaternion.	191

Liste des tableaux

2.1	Paramètres intrinsèques de la caméra Sony XC-555P, utilisée dans les expérimentations.	79
2.2	Estimation des différentes erreurs de reconstruction.	80
2.3	Estimation des différentes erreurs de généralisation.	81
2.4	Paramètres des régressions utilisées.	83
2.5	Résultats des différentes erreurs d'estimation de distance.	83
2.6	Récapitulatif de quelques méthodes d'estimation de pose.	89
3.1	Temps de traitement de l'algorithme robuste en fonction du nombre de points suivis.	115
3.2	Erreur de reconstruction en fonction du nombre de points visibles.	116
4.1	Variables cinématiques utilisées.	125
4.2	Coefficients du filtre passe-bas.	135
4.3	Nombre de données en fonction des fréquences.	147

Remerciements

Mes remerciements les plus sincères vont vers Malik Mallem qui m'a accueilli dans son équipe, a dirigé mon travail et m'a conseillé tout au long de cette thèse. Ma profonde gratitude est pour Fakhreddine Ababsa pour son encadrement et ses conseils qui m'ont permis d'y voir plus clair et plus loin. Une reconnaissance particulière pour Jean-Yves Didier qui m'a beaucoup aidé sur la partie de localisation par la vision ainsi que tous les échanges constructifs que nous avons pu avoir et qui m'ont permis d'apporter des réponses aux difficultés rencontrées.

Je tiens à remercier les membres du jury, Éric Marchand et Didier Stricker pour avoir accepté de rapporter ce travail et apporter une contribution critique à celui-ci et aussi pour toutes leurs remarques positives et constructives. Je remercie sincèrement Jean-Marc Lavest d'avoir participé au jury de soutenance et d'examiner ce travail.

Florent Chavand fera également l'objet de ma reconnaissance à double titre : en tant qu'examineur faisant partie du jury et en tant qu'ancien directeur du laboratoire qui a mis à disposition sa structure pour m'accueillir lors du commencement de cette thèse. Son successeur à la tête du laboratoire, Étienne Colle, sera tout autant remercié.

Je remercie tous les doctorants pour l'ensemble des moments partagés ainsi que le personnel technique et administratif et toutes les personnes qui m'ont soutenu ou aidé d'une manière directe ou indirecte pendant cette thèse.

Enfin, j'aimerais exprimer ma gratitude à ma famille pour le support et l'aide qu'elle m'a fournis et dont le soutien et les encouragements n'ont jamais fait défaut.

Introduction générale

Les systèmes de réalité augmentée permettent la superposition des images virtuelles sur des scènes réelles. Le concept de la réalité augmentée enrichit notre perception du monde réel, en y ajoutant des informations de manière dynamique et interactive. Ses applications sont multiples et touchent à plusieurs domaines : jeux vidéo, télévision, industrie, médecine, etc.

L'objectif de la réalité augmentée est d'apporter un réalisme et une cohérence visuelle entre le flux réel et virtuel. Afin de maintenir le recalage dynamique des objets de synthèse sur le monde réel, les systèmes d'acquisition doivent en permanence calculer le point de vue de l'opérateur pour donner l'illusion que ces objets virtuels appartiennent au monde réel. Le positionnement de ces objets dans la scène nécessite de connaître la position et l'orientation de la caméra par rapport à un repère lié au monde.

Le problème de la localisation de la caméra est donc important, divers travaux sont menés dans ce domaine pour le résoudre. Plusieurs capteurs peuvent être utilisés pour déterminer le positionnement du point de vue de l'utilisateur dans l'environnement comme : les capteurs magnétiques, les capteurs optiques, les encodeurs ou évidemment les caméras. Toutefois, dans la littérature, la pose est souvent calculée par la caméra en utilisant des méthodes géométriques. Ces méthodes emploient des modèles d'objets et leurs projections dans l'image pour résoudre la transformation entre les repères où les données sont exprimées.

Si l'on considère uniquement les informations acquises par la caméra, le problème de réalité augmentée se ramène à un problème de vision par ordinateur. Les techniques de vision permettent à la fois la reconstruction 3D des points de la scène, la calibration, la localisation 3D, etc. Cependant, le temps de traitement avec les systèmes de vision dépend de la complexité algorithmique des méthodes implémentées. En outre, de tels dispositifs sont souvent confrontés à des problèmes de conditions d'acquisition dans l'environnement réel de l'utilisateur.

Les applications récentes de la réalité augmentée ont ouvert un nouveau champ pour l'utilisation des systèmes multi-capteurs afin de surmonter les problèmes des dispositifs de tracking basés vision. Ces capteurs permettent de : localiser rapidement la caméra par rapport à la scène, accélérer les traitements, résoudre certains problèmes liés à l'environnement et servent aussi à adapter le modèle virtuel de la scène pour mieux gérer les augmentations. Les objets virtuels subissent alors des transformations géométriques qui permettront de les recalculer dans l'image réelle et les intégrer de la manière la plus naturelle possible.

L'un des verrous lors de la réalisation d'un système de réalité augmentée est le manque de précision et de robustesse des dispositifs de tracking utilisés. La plupart des applications de réalité augmentée exigent un environnement contrôlé et imposent des contraintes de mouvements et de visibilité. Selon le type d'application, deux solutions sont envisageables, la première repose sur l'utilisation des systèmes multi-capteurs, la seconde emploie les méthodes robustes. Le suivi multi-capteurs ou hybride emploie généralement des capteurs de localisation : une centrale inertielle, un capteur magné-

tique, etc. Une centrale inertielle est rapide et ne nécessite pas d'éléments extérieurs. Cependant, le problème avec ce genre de capteur est l'accumulation des erreurs de dérives et la faible précision. Le suivi avec des capteurs magnétiques est aussi souvent utilisé, mais il possède l'inconvénient de confiner l'utilisateur dans un environnement petit et instrumenté. Les méthodes basées vision prennent en compte les propriétés de l'image et permettent un recalage précis des éléments virtuels ajoutés à la scène. Ces méthodes empêchent, toutefois, le suivi dans le cas d'occultations des cibles. L'utilisation des méthodes robustes est donc, nécessaire pour réaliser des tâches en environnement réel et pallier les problèmes d'occultations et des changements des conditions d'environnement. Ces techniques sont efficaces, mais elles requièrent un certain nombre de primitives visibles et le temps de calcul est souvent trop élevé et ne permet pas une utilisation en temps réel.

Cette thèse porte sur le suivi robuste et hybride de cibles dans une application de réalité augmentée. L'objectif principal est de contribuer à l'amélioration de la précision et de la robustesse des systèmes de réalité augmentée basés vision. Une architecture multimodale constituée de différents modules de suivi est élaborée. L'approche multimodale consiste à combiner différents capteurs et techniques, associés en fonction des conditions réelles de l'environnement. En effet, les contraintes liées à l'environnement réel de l'utilisateur, le temps réel et de précision du tracking, sont prises en compte tout le long du travail présenté.

Dans le chapitre 1, nous introduisons le domaine de recherche de la réalité augmentée. L'ensemble des systèmes et des prototypes existants dans la littérature sont étudiés et analysés. Les principaux travaux réalisés et les solutions disponibles sont évoqués et enfin nous exposons notre problématique et les solutions apportées afin de contribuer à lever quelques verrous liés à l'utilisation d'un système de réalité augmentée.

Le chapitre 2 présente notre système de localisation par la vision. Ce dernier, repose sur l'utilisation de cibles codées qui permettent de calculer en temps réel la pose de la caméra par rapport aux objets suivis. Des algorithmes spécifiques de calcul de pose sont introduits et des solutions hybrides sont adoptées pour améliorer la précision. Les divers systèmes de localisation du point de vue de la caméra sont évalués afin de déterminer leurs performances et permettre de s'affranchir des problèmes liés aux différentes méthodes.

Dans le chapitre 3, nous nous intéressons à la gestion d'occultations afin de maintenir le recalage des objets de synthèse sur les images réelles. Nous présentons notre méthode de suivi robuste qui permet la gestion d'occultations partielles des cibles. Une étude bibliographique sur les estimateurs robustes et les systèmes de gestion d'occultations en réalité augmentée est effectuée. Ensuite, notre méthode de suivi robuste est présentée et nous détaillons les étapes nécessaires à l'élaboration de l'algorithme de tracking. Enfin, nous terminons par une série d'expérimentations pour évaluer les performances de notre méthode.

Enfin, dans le chapitre 4, nous présentons une approche hybride de tracking en réalité augmentée. Notre dispositif multi-capteurs est constitué d'une caméra et d'une centrale inertielle. Notre méthode hybride permet de surmonter les défauts des deux capteurs. Cette méthode exploite la complémentarité des capteurs afin d'enrichir l'information perçue et pallier les éventuelles défaillances du système. L'étude porte sur la description de notre approche de tracking hybride ainsi que les différentes solutions proposées pour la gestion de l'information issue du système multi-capteurs. Enfin, le protocole expérimental et les résultats obtenus sont présentés dans ce chapitre.

Nous allons à présent commencer par introduire notre domaine de recherche qu'est la réalité augmentée.

Chapitre 1

Réalité Augmentée : État de l'art

1.1 Introduction

Un système de réalité virtuelle est une modélisation par ordinateur dans laquelle le graphisme est utilisé pour créer un monde qui semble réaliste. Cette modélisation n'est pas statique car elle répond aux ordres de l'utilisateur (gestes, paroles ou toutes autres commandes extérieures). La notion de réalité virtuelle date de plus de cinquante ans, avec l'invention "Sensorama Simulator" (figure 1.1). Le Sensorama Simulator fut inventé par le réalisateur de documentaires américain Morton Heilig au milieu des années 50.

Son dispositif était constitué d'un système vidéo 3D, des ventilateurs et d'un siège vibrant. Les ventilateurs situés près de la tête de l'utilisateur fournissent des odeurs et du vent. Le siège vibrant donne la sensation de mouvement à l'utilisateur. Ainsi, il était possible de simuler un voyage à moto-cyclette ou des voyages en automobile.



FIG. 1.1 – Sensorama Simulator.

La réalité virtuelle est une façon pour l'humain de visualiser, manipuler, interagir avec des données complexes à l'aide d'un ordinateur. La réalité augmentée regroupe l'ensemble des techniques permettant d'associer un monde réel avec un monde virtuel, en utilisant des images réelles et des entités virtuelles (images de synthèse, objets virtuels, graphiques, etc). Plusieurs classifications de la réalité augmentée peuvent être envisagées selon les approches techniques, fonctionnelles ou applications.

La réalité augmentée a vu le jour avec les travaux de Sutherland [151] [152]. En effet, Sutherland a fait des recherches sur les casques de réalité virtuelle au début des années 70. En affichant simplement quelques polygones, il construit le premier casque de réalité virtuelle transparent "See Through System". L'utilisation d'un casque virtuel à travers lequel on peut voir le monde réel, répond au problème des scientifiques : ça permet à l'utilisateur de garder le contact avec le monde réel tout en étant dans un monde virtuel.

Selon Ronald Azuma, l'un des pionnier de la réalité augmentée, trois règles fondamentales sont nécessaires pour le fonctionnement d'un système de réalité augmentée [13], et qui sont :

- Combiner le réel et le virtuel.
- Respecter les contraintes d'interactivité et de temps réel.
- Respecter l'homogénéité et la cohérence entre les deux mondes réel et virtuel.

Cette définition exclut donc les simples superpositions 2D qui ne respectent pas la cohérence 3D, ainsi que la composition en post-production qui n'est pas temps réel. En revanche, aucune contrainte n'est imposée sur le réalisme photométrique des compositions (par exemple, les objets virtuels peuvent être affichés en mode filaire). La visualisation peut se faire à l'aide des dispositifs particuliers permettant de voir la réalité en même temps que les objets synthétiques représentés en relief. Sur un axe représentant le passage de l'environnement réel à un monde virtuel, la réalité augmentée se situe dans la partie gauche de l'axe, en opposition à la virtualité augmentée qui concerne l'ajout d'éléments réels (personnes, textures, etc.) dans des environnements virtuels. La réalité augmentée et la virtualité augmentée sont regroupées sous le terme de "réalité mixte" [109] (figure 1.2).



FIG. 1.2 – Le continuum réel/virtuel de Milgram [109].

Les technologies des environnements virtuels immergent complètement l'utilisateur dans un environnement synthétique. L'utilisateur, dans ce cas, ne peut pas voir le monde réel autour de lui. En revanche, la réalité augmentée permet à l'utilisateur de voir l'environnement réel avec des objets virtuels superposés. Par conséquent, la réalité augmentée complète la réalité plutôt que de la remplacer, l'utilisateur est immergé dans un environnement où les objets virtuels et réels coexistent ensemble dans le même espace.

Un des premiers systèmes de réalité augmentée est réalisé par Feiner et al. [45], c'est le projet KARMA (Knowledge-based Augmented Reality for Maintenance Assistance). Comme démonstration, les auteurs choisissent un système de support du personnel pour configurer et gérer une imprimante laser (figure 1.3). Plusieurs capteurs sont attachés à l'imprimante pour détecter leurs positions et leurs orientations. Le monde virtuel est un complément et un enrichissement du monde réel. Dans la figure 1.3, on observe ce que voit l'utilisateur à travers le casque. Dans cette application l'utilisateur change le papier dans l'imprimante, le système de réalité augmentée offre une aide visuelle représentant le bac du papier et son fonctionnement.

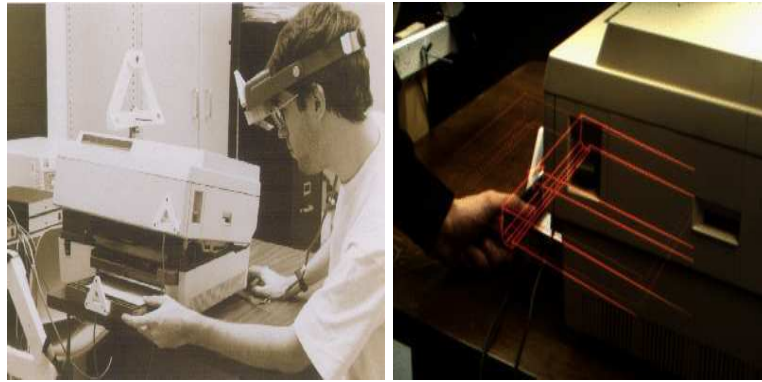


FIG. 1.3 – Le projet KARMA [45].

Actuellement, les avancées technologiques permettent de réaliser des objets virtuels à l'aide des bibliothèques graphiques en 3D. On peut considérer que la plupart des domaines de la vie courante peuvent et pourront un jour bénéficier de la réalité augmentée.

1.2 Les applications de la réalité augmentée

La réalité augmentée est un domaine de recherche relativement récent et en plein essor. Les applications sont nombreuses dans des cadres différents tels que la robotique ou la médecine. L'objectif de la réalité augmentée est l'amélioration de la perception et de l'interaction de l'utilisateur avec le monde réel. Les objets virtuels permettent l'affichage des informations que l'utilisateur ne peut pas voir directement ou détecter avec ses propres sens. Ces informations virtuelles aident l'utilisateur à exécuter ses tâches dans le monde réel.

Les applications potentielles de la réalité augmentée sont dans les domaines suivants : médical, visualisation, industrie/maintenance, robotique, militaire, etc. Nous allons maintenant détailler ces applications ainsi que les technologies utilisées.

1.2.1 Médical

La médecine est un domaine de prédilection de la réalité augmentée. Bajura et al. [15] ont décrit un système de visualisation médicale, ce dernier montre les résultats d'une échographie visualisée à l'intérieur du ventre d'une femme enceinte (figure 1.4). La visualisation est effectuée en utilisant une caméra miniature montée sur le visiocasque de l'opérateur. Les images de la caméra et les images ultrasonores générées par ordinateur sont superposées pour représenter un environnement réel/virtuel riche en information.

Des travaux similaires ont été réalisés à l'Université de Caroline du Nord (UNC) [71], un casque de réalité augmentée a été développé et utilisé pour l'échographie des femmes enceintes. Ce casque permet une superposition géométriquement correcte des données ultrasonores sur le ventre de la mère permettant au gynécologue d'examiner la position du fœtus dans l'utérus de la mère. Auparavant le médecin avait des difficultés à regarder l'écran de contrôle et intégrer cette image à la position de sa main sur le ventre de la mère (volume 3D). Avec la réalité augmentée, le système supprime cette difficulté, en permettant à l'utilisateur de voir les images scannées superposées sur le ventre de la mère (figure 1.5).

Plusieurs projets explorent ce domaine d'application. Des efforts récents se sont concentrés sur une biopsie d'aiguille d'une tumeur de sein. La figure 1.6 montre une maquette d'une opération de biopsie du sein, où les objets virtuels identifient l'endroit de la tumeur et guident l'aiguille à sa cible [146].

D'autres travaux ont été réalisés pour faire du recalage des images tomographique ou a résonance magnétique dans une séquence d'images à rayon X et aussi la reconstruction 3D de la vascularisation cérébrale dans des images d'angiographie numérique soustraite [13]. Ainsi en neuroradiologie interventionnelle par exemple, les travaux de Kerrien [82] permettent au radiologue de savoir à tout instant où se trouve son cathéter dans le corps du patient.

La réalité augmentée pourrait également être utile pour la visualisation médicale générale dans les salles de chirurgie. Les chirurgiens peuvent voir des choses avec l'oeil nu qu'ils ne peuvent pas voir avec les dispositifs et vice-versa. La réalité augmentée donne accès aux chirurgiens aux deux types d'informations simultanément. Ceci pourrait également guider la précision de la tâche, montrer par exemple, comment faire une opération sur le crâne pour une chirurgie de cerveau ou exécuter une biopsie d'aiguille d'une petite tumeur. L'information des capteurs non-invasifs est directement affichée sur le patient, montrant exactement où effectuer l'opération.

La réalité augmentée est aussi utilisée pour la formation des chirurgiens. Les instructions virtuelles rappellent le chirurgien débutant les étapes nécessaires, sans avoir à consulter un manuel. Les objets virtuels peuvent également identifier des organes et indiquer leurs endroits [41].

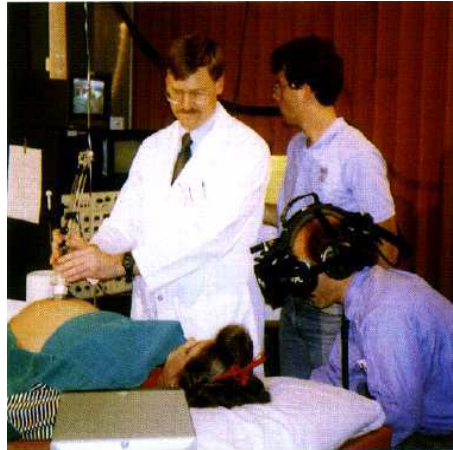


FIG. 1.4 – Visualisation à l'aide d'un visiocasque les images d'échographie [15].



FIG. 1.5 – Visualisation du ventre à l'aide du système de réalité augmentée développé à l'UNC [71].

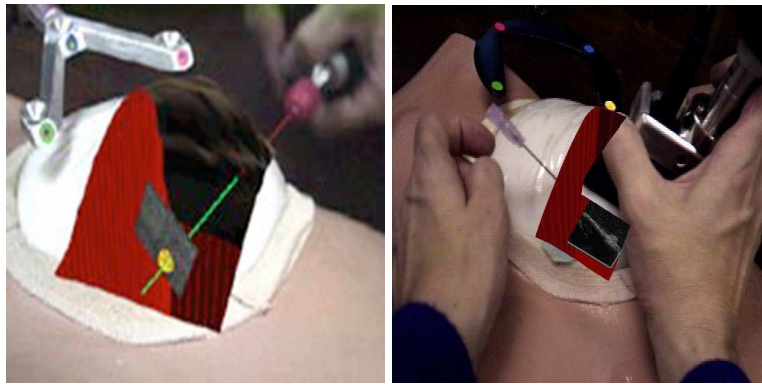


FIG. 1.6 – Représentation virtuel de l'intérieur du sein [146].

1.2.2 Industrie-maintenance

Une autre catégorie des applications de la réalité augmentée est l'assemblage, l'entretien, et la réparation des machines industrielles complexes. Les instructions pourraient être plus faciles à comprendre si elles étaient disponibles, pas sous forme de manuels avec le texte et l'image, mais plutôt sous forme de schémas en 3D superposés sur le dispositif réel, montrant étape par étape, la tâche à exécuter.

Plusieurs projets de recherche ont abouti à la réalisation de prototypes dans ce secteur. Feiner et al. [45], à l'université de Columbia, ont réalisé une application pour la maintenance d'une imprimante laser appelé KARMA (voir l'introduction).

Un groupe chez Boeing s'est intéressé à un système de réalité augmentée pour guider un technicien dans la réalisation d'un harnais de câblage (Wiring Harness) qui forme le système électrique d'une aile d'avion. Actuellement, les techniciens utilisent de grands tableaux pour construire de tels systèmes, et Boeing a besoin de beaucoup de pièces pour stocker ces systèmes. Une projection virtuelle sur un tableau unique (figure 1.7) est utilisée, ce qui permet de conserver de la place et de réduire les coûts [25] [76] [143].



FIG. 1.7 – Application de câblage d'un système électrique dans un prototype de Boeing [25] [76] [143].

Un autre projet appelé ARVIKA (Augmented Reality for Development, Production and Servicing) a été réalisé dans le domaine industriel [10]. Le projet est sponsorisé par le ministère allemand de l'Éducation et de la Recherche et réalisé par plusieurs partenaires industriels (SIEMENS, AIRBUS, Audi, BMW...). ARVIKA a pour objectif de réaliser un système de réalité augmentée permettant le

développement, la production et le service pour les systèmes technologiques complexes tels que les véhicules et les avions (figure 1.8). Le système permet d'améliorer la perception de l'utilisateur et lui fournit une aide visuelle pour l'assister dans ses opérations. À travers le recalage virtuel des objets sur des scènes réelles, la technologie développée permet d'agir sur le monde réel d'une manière interactive et précise pour diverses tâches de développement et de maintenance. En cas de difficulté, l'utilisateur est relié à un centre de service pour l'assister dans les opérations complexes.

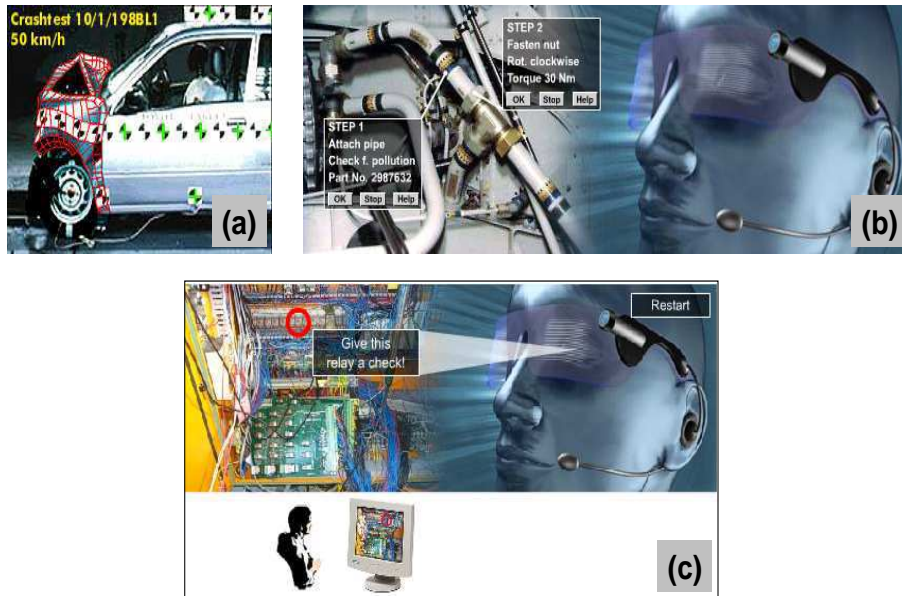


FIG. 1.8 – Le système ARVIKA [10]. (a) Développement : système de réalité augmentée pour le crash-test. (b) Production : assistance visuelle pour l'utilisateur dans une tâche de maintenance. (c) Service : en cas de difficulté l'opérateur est relié au centre de service.

En France, et dans notre laboratoire IBISC (Informatique, Biologie Intégrative et Systèmes Complexes), un projet intitulé AMRA (Assistance et Maintenance en Réalité Augmentée) du RNTL (Réseau National des Technologies Logicielles) placé sous la tutelle du ministère de la Recherche est réalisé [35]. Le projet a commencé en 2002 et s'est achevé en 2004. Le but était d'implémenter un système de réalité augmentée à usage mobile pour une utilisation en milieu industriel, et plus spécifiquement dans le domaine de la maintenance industrielle. Ce projet a plusieurs objectifs :

- Fournir une aide contextuelle à des mainteneurs inexpérimentés, leur permettant d'être formés sur site.
- Apporter aux agents de maintenance une assistance permettant d'accéder sur leur poste de travail à des informations pertinentes (documentation de maintenance, modes opératoires, films de montage).
- Augmenter la disponibilité de l'information sur le lieu de maintenance en utilisant les techniques de réalité augmentée.

Le prototype AMRA est un système de réalité augmentée en vision indirecte constitué d'une tablette-PC (un ordinateur portable allégé pourvu d'un écran tactile) pour la visualisation des informations, qui agit comme une fenêtre augmentée sur le monde réel, grâce à la caméra embarquée sur ce dernier. Ce type de système aborde plusieurs problématique : celle de l'informatique nomade (en anglais mobile computing), celle du recalage temps réel des entités virtuelles sur les images du monde réel, et enfin celle du développement d'une aide graphique contextuelle adaptée (figure 1.9).

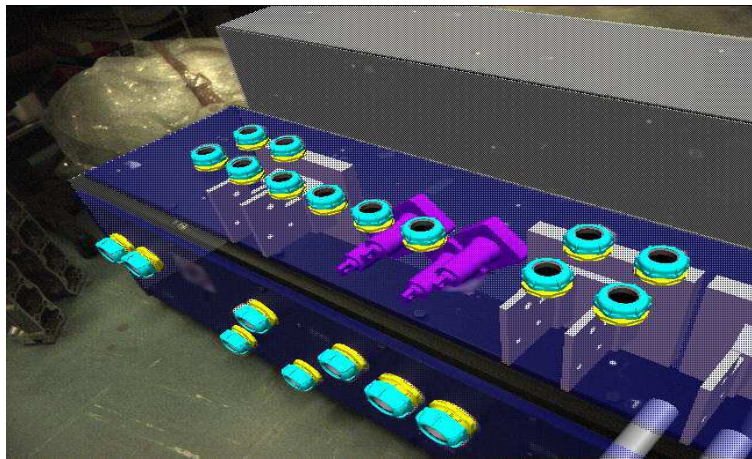


FIG. 1.9 – Exemple de vue en transparence fournie à l’opérateur - Projet AMRA [35].

1.2.3 Visualisation

La réalité augmentée est aussi utilisée pour annoter des objets dans un environnement particulier en affichant des informations permettant de le rendre plus compréhensif et interactif avec l’opérateur. Par exemple un écran-afficheur à main (hand-held display) pour fournir des informations concernant les bouquins d’une librairie lorsque l’utilisateur se ballade dans une bibliothèque. Des chercheurs de l’université de Columbia [44] ont développé une application permettant d’afficher des fenêtres d’une interface standard sur le monde réel ou sur des objets particuliers afin de les mémoriser et augmenter la perception de l’utilisateur de son environnement (figure 1.10).



FIG. 1.10 – Fenêtres virtuelles affichées sur des objets spécifiques du monde réel pour l’annotation [44].

La réalité augmentée sert aussi d’aide générale à la visualisation, un architecte peut regarder par une vision rayon X à l’intérieur d’un bâtiment pour observer l’installation électrique, les tuyaux d’eau, les structures des murs, etc.

Des chercheurs de l’université de Toronto ont développé un système de vision stéréoscopique pour la réalité augmentée (ARGOS). Ce système permet d’augmenter les scènes réelles dans des conditions difficiles de visualisations [38]. La figure 1.11 montre le modèle filaire de l’intérieur d’une navette spatiale en orbite. Les lignes tracées permettent de voir plus facilement la géométrie de la navette. Ces lignes et objets virtuels facilite la navigation et la compréhension de l’environnement quand les conditions de visibilité sont médiocres.



FIG. 1.11 – Augmentation de la scène avec des lignes virtuelles [38].

1.2.4 Téléopération

La téléopération d'un robot est un problème difficile, particulièrement quand le robot est sur un site distant ce qui provoque des retards dans les transmissions. Dans ces circonstances, au lieu de commander le robot directement, il est préférable de commander son modèle virtuel à sa place. L'utilisateur manipule le robot virtuel en temps réel et le robot réel suit. Une fois la tâche est définie, l'utilisateur demande au robot de l'exécuter réellement. Le système permet aussi de prédire et d'assister l'utilisateur qui manipule le robot et lui offre un outil de visualisation et de planification pour l'aider dans l'exécution de la tâche désirée. Le système ARGOS a démontré que la vision stéréoscopique est plus efficace et facile pour faire la planification de trajectoire d'un robot, à l'inverse des interfaces en monovision traditionnelles [37] [110]. D'autres auteurs ont également utilisé des modèles virtuels enregistrés pour un système de télé-présence [83] [84] [123] [153]. La figure 1.12 montre comment le robot virtuel représente la position prédite du robot réel.

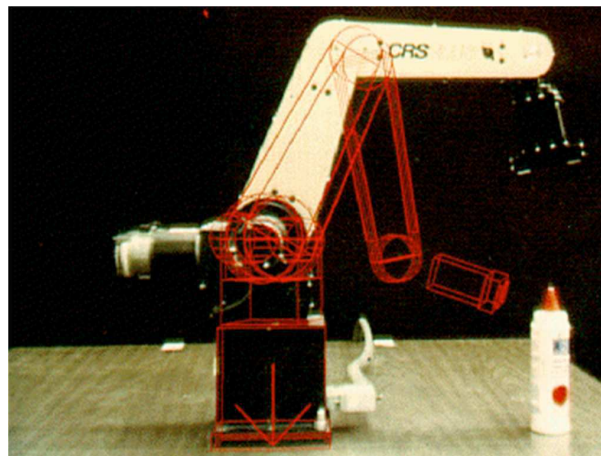


FIG. 1.12 – Téléopération d'un robot réel en utilisant son prototype virtuel [37].

Un autre système de téléopération a été développé au laboratoire IBISC [122], le système s'appelle ARITI (Augmented Reality Interface for Teleoperation via Internet) qui est le premier système de téléopération en réalité augmentée apparu en France, permettant le contrôle à distance d'un robot à l'aide d'un robot virtuel et de guides virtuels (figure 1.13).

ARITI a été mis pour la première fois en ligne sur le web en 1998 et depuis Janvier 2000, il est référencé par la NASA Space Telerobotics Program. Le système ARITI est réalisé dans le but de permettre à un opérateur humain de contrôler et de commander un robot à distance. L'interface Homme-Machine est basée sur le concept de la réalité mixée, regroupant ainsi la réalité virtuelle et la réalité augmentée. Pendant une tâche de téléopération, l'utilisateur est assisté par des aides appelées

guides virtuels actifs qui améliorent ses performances. Outre les assistances offertes à l'utilisateur, des possibilités de travail en commun lui sont également proposées.



FIG. 1.13 – Téléopération du robot ARITI [122].

1.2.5 Divertissement

À SIGGRAPH'95, plusieurs exposants ont montré des systèmes virtuels qui permettent l'immersion et l'interaction des utilisateurs avec un monde virtuel en temps réel et en 3D. Les utilisateurs se tiennent devant un grand écran bleu, alors qu'une caméra commandée par ordinateur enregistre la scène. La caméra localise les utilisateurs et l'ordinateur génère des acteurs virtuels qui reproduisent les mouvements des utilisateurs.

L'industrie de divertissement voit ceci comme une manière de réduire les coûts de production : créer et stocker des environnements virtuels est un marché très rentable par rapport à la réalisation d'un décor réel. Le projet ALIVE du laboratoire MIT Media [97] a mis en oeuvre un environnement avec des créatures virtuelles intelligentes qui réagissent aux actions de l'utilisateur. L'utilisateur devant son écran interagit avec des entités virtuelles (figure 1.14).



FIG. 1.14 – Le projet ALIVE : l'utilisateur donne des ordres à son chien [97].

Un autre projet dans le domaine du jeu et loisirs est RV-Border Guards qui a été présenté au Workshop international de réalité augmentée [120], 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 : les armes, les ennemis, etc (figure 1.15). Ce type d'application est très prometteur et envisage une nouvelle génération de jeu.



FIG. 1.15 – Le jeu RV-Border : scène telle qu'elle est vue par le joueur [120].

1.2.6 Militaire

Pendant plusieurs années, les avions militaires et les hélicoptères ont employé des dispositifs d'affichage à tête haute (HUD : Head Up Display) et les casques de vue (HMS : Helmet Mounted Sight) pour superposer des graphiques sur l'écran d'affichage du pilote (figure 1.16). Ces systèmes permettent de fournir des informations de base pour la navigation et le vol. Les graphiques sont superposés sur des cibles dans l'environnement réel. Par exemple, la tourelle dans un hélicoptère militaire peut être asservie au HMS du pilote, ainsi le pilote peut viser avec la tourelle en regardant seulement la cible. De futures générations d'avion de combat seront développées avec des HMDs intégrés aux casques des pilotes [158].

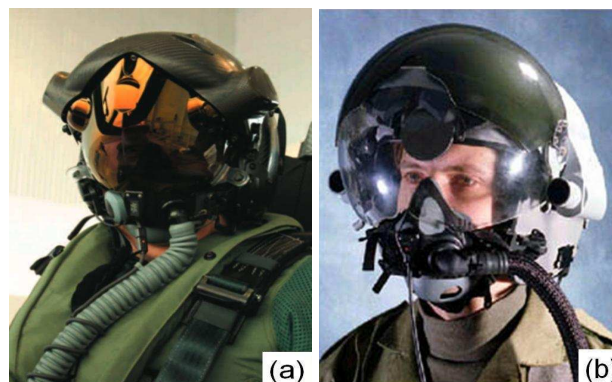


FIG. 1.16 – Les casques utilisés dans les avions de combats. (a) HUD (Head Up Display). (b) HMS (Helmet Mounted Sight).

1.2.7 Architecture

L'architecture est particulièrement concernée par les applications de la réalité augmentée. En effet, les produits mis au point permettent de visualiser les maquettes à des échelles diverses et de voir aussi à l'intérieur et à l'extérieur des bâtiments.

La réalité augmentée permet de concevoir des extensions ou des modifications d'une architecture existante, définir ou modifier les aménagements d'ameublement, et établir des scénarios concernant des événements imprévus. Toutes ces opérations sont difficiles à réaliser réellement et nécessitent des moyens matériels importants. Avec la réalité augmentée l'environnement est enrichi avec des entités virtuelles ce qui permet de modéliser tout ce que l'utilisateur imagine.

1.3 Dispositifs et capteurs utilisés en réalité augmentée

Plusieurs dispositifs et capteurs sont utilisés pour réaliser un système de réalité augmentée. Néanmoins, les dispositifs les plus employés sont : les systèmes d'affichage qui jouent un rôle primordial pour le rendu visuel et la génération des scènes virtuelles et les systèmes de suivi nécessaires à la localisation dans le repère du monde. La plupart des capteurs utilisés en réalité augmentée sont dédiés pour des applications en intérieur. Cependant, de nouvelles technologies nécessitant l'utilisation de nouveaux dispositifs sont destinées pour des applications en extérieur ou sans fil.

1.3.1 Systèmes de visualisation

Les systèmes d'affichage sont utilisés pour visualiser des objets virtuels. Des dispositifs d'affichage dédiés permettent alors de mixer le réel et le virtuel. On peut distinguer différentes classes de systèmes :

- les afficheurs de types visiocasque (HMD) : le système est porté sur la tête de l'utilisateur, et se distingue en deux catégories. Les casques dits semi-transparent optique (optical see-through HMD), constitué d'un écran LCD couplé à un miroir semi-transparent (figure 1.17-a) dont le mixage réel et virtuel est fait par l'oeil de l'utilisateur [151]. Pour ce type de casques, on peut citer le NOMAD, qui est un système d'affichage à haute résolution permettant l'accès à tout type d'information tout en regardant le monde réel (figure 1.17-c). Il y a aussi les casques dits semi-transparent vidéo (video see-through HMD) dont le mixage est fait entre un rendu graphique et l'image provenant d'une caméra (figure 1.17-b), ce mélange étant alors présenté à l'utilisateur [42].
- les afficheurs de type écran : la visualisation est faite à travers un écran, à l'aide d'une caméra couplée à cet élément qui peut être : un écran de bureau (figure 1.18) [5], un ordinateur portables [137], un écran portable à la main (figure 1.19) [114] [134] ou une solution mixte [20].
- les affichages par projection : la visualisation est faite sur une surface du monde réel, la projection pouvant être sur une table [130], un mur [131] ou sur une surface quelconque [124] [125] [128].

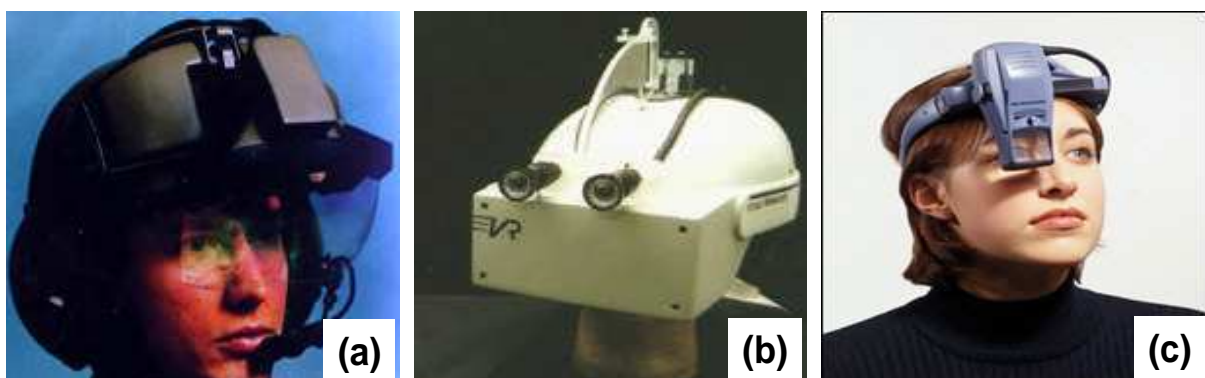


FIG. 1.17 – Les visiocasques. (a) Optical see through. (b) Video see through. (c) NOMAD.



FIG. 1.18 – Écran LCD 3D pour la vision stéréoscopique en réalité augmentée.



FIG. 1.19 – Écran portable à la main.

1.3.2 Capteurs de position et d'orientation

Les capteurs de position et d'orientation servent à localiser un objet rigide en mouvement par rapport à un repère absolu dans l'espace. On s'en sert principalement pour mesurer le mouvement de l'utilisateur. En général, on mesure une série de 6 grandeurs (3 positions et 3 angles) avec une fréquence d'échantillonnage compatible avec la précision recherchée. Il existe de nombreux types de capteurs. On peut les classer par catégories : mécaniques, magnétiques, ultrasonores, optiques. Pour les comparer, les critères suivants sont utilisés : les degrés de libertés (ddl), la résolution, l'erreur, le temps de réponse, l'espace de travail et la sensibilité.

Capteur mécanique : Ils se présentent en général sous la forme de bras articulés avec 2 ou 3 segments. Ils utilisent des dispositifs de mesures analogiques ou numériques de grande précision attachés au bras pour obtenir la position et l'orientation de l'extrémité du bras. N'utilisant pas de technologies sensibles à l'environnement (électromagnétique, sonores) ils n'ont aucune interférences avec l'extérieur. Les systèmes mécaniques ont une bonne précision, résolution, et temps de réponse, mais leur champ de travail est limité et ils sont très peu maniables.

Capteur magnétique : Ces capteurs mesurent la distorsion du champ magnétique pour la localisation. Un champ basse fréquence est modulé pour déterminer la position et l'orientation de l'objet mobile. Le signal reçu du récepteur est échantillonné grâce à un algorithme contenu dans l'unité de traitement pour déterminer la position du récepteur par rapport à l'émetteur. Le principal défaut de ce type de capteur est sa grande sensibilité par rapport à tout matériau magnétique ou appareillage électromagnétique qui distord le champ émis et rend la précision faible. De plus la présence de câblage sur le système augmente la complexité de manipulation. L'avantage de ces capteurs est leur coût peu cher et leur poids (figure 1.20).

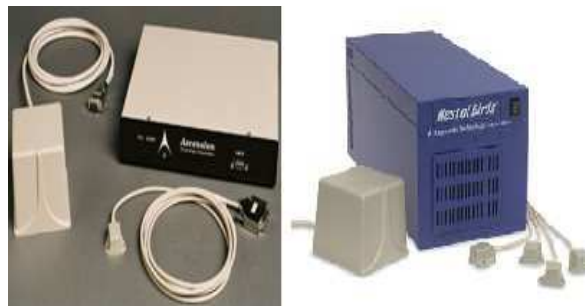


FIG. 1.20 – Capteurs magnétiques.

Capteur à ultrasons : Le capteur à ultrasons (figure 1.21) est un capteur télémétrique mesurant uniquement les positions, composé d'un émetteur, d'un récepteur et d'un microcontrôleur PIC. Il sert à mesurer la distance qui le sépare d'un obstacle se situant devant lui. Son fonctionnement correspond à celui des sonars : lorsque l'on veut mesurer une distance, l'émetteur envoie une salve d'ultrasons de fréquence de 40KHz par exemple, puis le PIC commence à compter le temps. Lorsque le récepteur reçoit l'onde réfléchie, le PIC arrête de compter et selon le temps parcouru par l'onde, aller/retour, et en connaissant la vitesse du son dans le milieu où l'onde se propage, le capteur nous donne la distance le séparant de l'obstacle. En connaissant la vitesse de l'obstacle, on peut mesurer aussi la vitesse de l'objet portant le capteur. L'inconvénient avec les ultrasons c'est l'interaction des ondes avec la matière qu'elles traversent. Il se produit plusieurs phénomènes qui aboutissent à leur atténuation :

- une absorption dans les milieux homogènes,
- des réflexions, réfractions et diffusions aux changements de milieux.



FIG. 1.21 – Capteur à ultrasons.

Capteur optique : Les capteurs optiques sont nombreux et peuvent être classés en diverses catégories suivant le type de lumière utilisée : visible, infrarouge, source laser. Ces capteurs sont très employés dans les films et les réalisations de jeu pour animer les mouvements de personnages 3D. Il existe principalement deux techniques employées pour les systèmes optiques. Dans la première, des marqueurs réfléchissants sont fixés sur l'utilisateur. Des caméras munies de LEDs infrarouges, éclairent ces marqueurs. Le mouvement de l'opérateur est ensuite enregistré et les marqueurs sont identifiés à partir des images. Dans la deuxième technique, les marqueurs sont des LEDs infrarouges alimentées par des piles (les capteurs sont auto-éclairants). Une caméra infrarouge est alors utilisée pour enregistrer le mouvement des marqueurs. Le principal défaut de ces capteurs est le temps de traitement souvent long, et l'absence de réponse si un obstacle est présent entre la caméra et le marqueur. Les systèmes lasers encore très peu utilisés servent principalement pour des grandes distances de l'ordre du kilomètre.

Capteur GPS (Global Positioning System) : Utilisé principalement pour les applications de réalité augmentée en extérieur. Le GPS se base sur la position retournée par les mesures des satellites (figure 1.22). Le système GPS comprend au moins 24 satellites artificiels orbitant à $20200km$ d'altitude. Ces satellites émettent en permanence un signal complexe (code pseudo-aléatoire) daté précisément grâce à leur horloge atomique, ainsi que des éphémérides permettant le calcul de leurs coordonnées prédites. Ainsi un récepteur GPS qui capte les signaux d'au moins quatre satellites peut, en mesurant les écarts relatifs des horloges, connaître sa distance par rapport aux satellites par triangulation. Le GPS est ainsi utilisé pour localiser des véhicules roulants, des navires, des avions, des missiles et même des satellites évoluant en basse orbite.

Certains systèmes GPS conçus pour des usages très particuliers peuvent fournir une localisation à quelques centimètres près. Le GPS différentiel (DGPS), corrige ainsi la position obtenue par GPS conventionnel par les données envoyées par une station terrestre de référence localisée dans un endroit très précis.



FIG. 1.22 – Capteur GPS.

Centrale inertielle : La centrale inertielle est une unité miniaturisée de mesure d'inertie fournissant un signal numérique d'accélération 3D, les taux de rotation 3D (compas gyroscopiques) et les données du champ magnétique terrestre (figure 1.23). Elle fournit des données précises sur l'orientation 3D en temps réel, un module de stabilisation est inclus à l'intérieur du capteur pour éliminer les dérives dues aux perturbations extérieures. La centrale inertielle permet le suivi de mouvement et l'orientation du corps humain, de véhicules non-pilotés et de robots mobiles, etc.



FIG. 1.23 – Centrale inertielle MTi de Xsens [166].

1.3.3 Les systèmes de tracking visuel

Les systèmes de tracking visuel sont composés essentiellement d'une ou plusieurs caméras. La caméra est un capteur de vision extéroceptif qui fournit non pas une simple information, comme pour les capteurs proximétriques (qui mesurent la distance à la surface la plus proche), mais un signal vidéo que l'on peut visualiser sur divers dispositifs d'affichage : téléviseur, écran d'ordinateur, etc. Le choix d'une caméra dépend du type d'application et de ses caractéristiques : capteur utilisé (CCD, CMOS), résolution, distance focale, illumination minimale, communication (ethernet, série, wifi, USB, etc), poids, consommation électrique, etc.

Plusieurs systèmes de tracking visuel sont commercialisés, ces systèmes contiennent des caméras et aussi d'autres capteurs légers et confortables. Ces dispositifs sont configurables à des fréquences et résolutions variables et possèdent aussi des options d'affichage : vision sans parallaxe, vision monoculaire/binoculaire, etc (figure 1.24).



FIG. 1.24 – Différents systèmes de tracking visuel utilisés en réalité augmentée.

Le système de tracking visuel est utilisable dans des situations où l'opérateur porte des lunettes de vues, et doit avoir une totale liberté de mouvement. Un tel système inclut tout l'équipement nécessaire

pour un travail immédiat. L'unité de commande du système est très compacte et légère et elle est portée par l'utilisateur (par exemple : comme une ceinture). La scène est enregistrée avec la caméra et le système de traitement transmet les images sur le dispositif d'affichage.

1.4 Problèmes liés à la réalité augmentée

L'un des problèmes les plus importants dans les applications de la réalité augmentée est celui de l'alignement virtuel-réel. Les objets dans le monde virtuel doivent être superposés correctement sur les objets réels, de telle manière à ce que l'utilisateur ait l'impression que les deux mondes réel et virtuel coexistent ensemble. Les applications de réalité augmentée nécessitent beaucoup de précision afin de réaliser un bon alignement.

Les erreurs d'alignement sont difficiles à contrôler et elles sont divisées en deux types : statique et dynamique. L'erreur statique apparaît quand le point de vue de l'utilisateur et l'objet dans le monde restent fixes. L'erreur dynamique apparaît quand le point de vue de l'utilisateur ou de l'objet est en mouvement dans l'environnement.

Dans les systèmes actuels, les erreurs dynamiques sont généralement la source principale des erreurs d'alignement. Les problèmes liés à l'éclairage, aux occultations et aux mouvements représentent aussi des verrous dans l'élaboration d'un système de réalité augmentée efficace.

1.4.1 Alignement spatial : erreurs statiques

On définit l'erreur statique par l'erreur d'alignement entre objet réel et objet virtuel, déterminée lorsque l'utilisateur est en position statique.

Trois types d'erreurs statiques sont définies :

- distorsion optique,
- erreur du système de suivi,
- mauvaise estimation des paramètres.

La distorsion optique est un problème bien connu et qui est dû aux propriétés des lentilles des caméras. L'erreur du système de suivi est générée par les capteurs électromagnétiques sensibles aux distorsions du champ magnétique. La mauvaise estimation des paramètres est l'erreur la plus étudiée en réalité augmentée. Une estimation fiable nécessite de mettre en oeuvre des méthodes robustes basées sur une définition de contraintes dans la procédure de calibration. À partir d'une modélisation du système, on s'intéresse alors à la calibration de ces différentes parties. On peut alors utiliser des méthodes automatiques (par vision, auto-calibrage) ou des méthodes basées utilisateurs (méthodes simples et efficaces) [59].

1.4.2 Alignement temporel : temps de latence

Le recalage dynamique des images de synthèse sur le monde réel est une étape cruciale dans l'élaboration d'un système de réalité augmentée. Les données des différents systèmes d'acquisition, tel que les dispositifs de suivi d'objet et de formation d'images, doivent être enregistrées spatio-temporellement avec le point de vue de l'utilisateur. Chaque dispositif a un retard associé à l'observation du monde et à l'affichage sur le système de visualisation. La différence des retards entre les divers dispositifs d'acquisition ou systèmes de traitement est appelée : latence relative. Elle représente une source du mauvais repérage et doit être réduite. Les sources de retard sont classifiées en six catégories et représentées par : retard d'arrivée des données, retard de traitement, retard du rendu, retard d'affichage, retard de synchronisation et retard de mise à jour.

Beaucoup d'auteurs ont traité la latence et ont essayé de réduire ses effets. Dans [54], [70], [162], les auteurs ont développé une technique de calcul de temps critique qui est très commercialisé pour

sa vitesse. L'application se rend constamment compte du temps de traitement et tâche de le réduire. La réduction de la latence est basée sur la réduction de la précision des calculs, ce qui ne pourrait pas être toujours recommandée.

La plupart des systèmes graphiques en temps réel sont dotés de hautes performances calculatoires afin de réduire la latence. Dans [121], les auteurs étudient un système fonctionnant en parallèle, ce qui minimise la latence et permet de réduire les retards dus aux systèmes de visualisation. La parallélisation diminue la latence en augmentant la vitesse des calculs, ainsi, dans [163], l'auteur utilise un processeur très performant pour permettre le traitement rapide des données.

La prédiction des positions et orientations du système de tracking peut être employée pour réduire le retard perçu par l'utilisateur [13]. Ces positions et orientations sont estimées par l'extrapolation des valeurs courantes et anciennes. Cette prédiction peut être alors employée pour produire une image intermédiaire pendant que l'image courante est visualisée.

Dans [16] et [145], les auteurs synchronisent les données du flux vidéo par le suivi de la tête de l'utilisateur en réduisant l'erreur de suivi avec l'utilisation des indices visuels afin de faciliter la recherche des cibles dans les images et réduire ainsi le temps de traitement.

1.4.3 Éclairage

Une augmentation réaliste demande de tenir en compte de l'éclairage par les sources lumineuses présentes dans la scène réelle. La cohérence, le modèle d'éclairage et la gestion des ombres est en effet, un problème de synthèse d'images et demande une connaissance des caractéristiques des sources lumineuses, leurs positions, couleurs, puissances, etc.

Le degré de luminosité des points de l'image définit le quotient de l'intensité lumineuse d'une surface. Une bonne luminance entraîne une meilleure exploitation des détails dans les zones de l'image et permet de caractériser les objets présents dans l'image et facilite leur détection.

De nombreux travaux de gestion d'éclairage ont été menés pour incruster les objets virtuels dans les images (travaux de [32], [50], [57], [92], [138], [147]). Récemment plusieurs auteurs se sont intéressés à des solutions temps réel simplifiées : estimation de direction [66], estimation par lightprobe [4], estimation et éclairage basé image [79], re-éclairage du modèle initiale avant intégration [23] [105], gestion des ombres [60], modification par projection [129]. Dans ce contexte, Sugano et al. [150] ont étudié la relation réel/virtuel par rapport à la notion d'ombres et montrent que la direction de l'ombre est l'information dominante dans un cas de scènes statiques et que dans des scènes dynamiques seule la forme globale est un paramètre important.

1.4.4 Occultation

La réalisation de tâches en réalité augmentée suppose que les indices visuels nécessaires sont disponibles à tout instant : ils ne doivent donc jamais être occultés partiellement ou totalement. Différentes approches ont été proposées pour faire du suivi en temps réel par des méthodes de vision afin de résoudre le problème d'occultations. Cela consiste en des techniques de suivi de plusieurs marqueurs positionnés dans divers endroits de l'environnement. Toutefois, la non-fiabilité de ces techniques de vision pour la gestion d'occultations, peut alors nécessiter la conservation d'un contrôle utilisateur pour ré-initialiser le suivi quand le système n'arrive plus à poursuivre les cibles.

Une approche hybride avec un suivi par vision (lent) et un autre capteur de position (rapide mais peu précis) reste sans doute une des meilleures solutions [12] [145]. Foxlin et Naimark [52] ont réalisé un système de tracking vision-inertiel afin de gérer les occultations dans des conditions de luminosité variées. D'autres travaux s'intéressant à la gestion d'occultations en utilisant les estimateurs robustes ont été réalisés dans la littérature. Un état de l'art sur ces méthodes sera présenté dans le chapitre 3.

1.4.5 Flou de mouvement

Les méthodes basées vision prennent en compte directement les propriétés de l'image, elles permettent un recalage précis de la scène et des éléments qui y sont ajoutés dans le cas des applications de réalité augmentée. Ces méthodes empêchent, cependant, le suivi lors d'un mouvement brusque du dispositif de tracking à cause du flou de mouvement et de la disparité importante générée dans l'image.

Le flou de mouvement se produit en raison de l'acquisition non-instantanée des images par la caméra. Quand un objet ou la caméra se déplace, les projections se déplacent aussi à travers la formation d'image pendant la durée de l'exposition dans la direction du mouvement. Des déplacements dépassant la fréquence d'acquisition de la caméra provoquent un flou de mouvement et peuvent sévèrement compromettre la visualisation de l'image et toutes les opérations qui y sont appliquées (filtrage, seuillage, détection, segmentation, etc).

Le flou de mouvement surgit en raison des contraintes technologiques des capteurs ou des conditions d'acquisition. Dans ce cas, l'image représente la scène seulement sur une certaine période de temps. Dans une telle situation, les objets se déplacent à travers les séquences et les images de la scène ne représentent plus une intégration de toutes les positions de ces objets à partir du point de vue de la caméra. Les objets en mouvement sembleront brouillés le long de la direction du mouvement.

L'effet provoqué par le flou de mouvement entre la caméra et les objets de la scène peut être évité en appliquant des filtres sur l'image ou bien en utilisant des capteurs de tracking dont la fréquence d'acquisition est beaucoup plus élevée que celle de la caméra [86].

1.5 Le tracking dans les systèmes de réalité augmentée

1.5.1 Systèmes de tracking en vision

1.5.1.1 Tracking temps réel

Harris [64] a développé un système appelé RAPiD (Real Time Attitude and Position Determination) qui était l'un parmi les premiers systèmes de tracking visuel basé sur un modèle 3D en temps réel sans marqueurs visuels. Harris a présenté une technique qui minimise le nombre de données à extraire en employant un matériel spécifique. Beaucoup de systèmes de tracking visuel partagent les principes du système RAPiD car il contient une description de la pose de la caméra par rapport à un modèle 3D. La pose est représentée par 6 paramètres, 3 pour la rotation et 3 pour la translation et pour chaque image, la pose de la caméra est mise à jour : premièrement par prédiction par rapport à la dynamique du modèle et après par mesure de l'entrée vidéo. Le système RAPiD a démontré son aptitude au tracking en temps réel. Ce système requiert des pré-traitements d'image pour déterminer la visibilité des points caractéristiques à partir de diverses positions de la caméra.

Lowe [94] a utilisé la détection de contour et la convolution d'images pour réaliser le suivi. Un calcul de probabilité est effectué pour ajuster les segments de contour du modèle par rapport à la distance relative à leurs projections. Ces vraisemblances guident le modèle à la recherche du meilleur appariement des contours détectés. Une mise en correspondance est ensuite réalisée pour mettre à jour les paramètres de minimisation. Si les résidus des moindres carrés sont importants, l'appariement est rejeté et la recherche est répétée jusqu'à ce qu'une solution à faible résidu soit trouvée. Le coût calculatoire de cette méthode limite les opérations à $3-5Hz$, ce qui est assez lent comparant à RAPiD qui tourne à $50Hz$. Cependant, le système permet le suivi de modèles complexes et a montré qu'il est robuste face aux arrières plans encombrés.

Gennery [56] a utilisé un dispositif spécialisé dans la détection de contour, la méthode de recherche des contours est similaire à celle de RAPiD. Pour chaque contour du modèle, des imagerie sont corrélées par des recherches de ressemblance. À l'inverse de RAPiD qui utilise un nombre limité de points de contours, Gennery place des points de contrôle chaque 3 pixels. De plus, les mesures

sont pondérées par rapport à la qualité du contour. Les contours de l'arrière plan sont supposés être de haute qualité alors que les contours internes sont pondérés par rapport au contraste et aux angles des faces adjacentes.

1.5.1.2 Asservissement visuel virtuel

Espiau et al. [43] ont utilisé l'asservissement visuel pour guider un robot vers une cible. À chaque instant, une matrice jacobienne reliant la vitesse du robot aux mesures de l'image est calculée. Le robot se déplace alors de telle manière à minimiser l'erreur sur l'image. L'idée est de considérer le système de vision dans la boucle de correction et de commande pour réaliser le suivi. Les résultats expérimentaux ont démontré la stabilité et la précision de l'approche.

Marchand et Chaumette [104] ont proposé une formulation originale d'estimation de la pose en employant le principe d'asservissement visuel. La méthode consiste à modifier les paramètres de la caméra virtuelle en utilisant un asservissement visuel pour aligner la projection du modèle avec les primitives dans l'image.

Comport et al. [28] ont présenté une étude comparative entre deux algorithmes de tracking en temps réel. La première méthode proposée par Drummond et Cipolla [39] basée sur l'algèbre de Lie. La deuxième méthode est celle proposée dans [31] utilisant l'asservissement visuel virtuel. La précision et la robustesse des deux méthodes ont été étudiées et les deux systèmes ont présentés des similarités quant à l'amélioration des performances et aux réglages optimaux des paramètres.

1.5.1.3 Travaux récents en tracking visuel

Vacchetti et al. [156] ont combiné le suivi basé contour avec la méthode de Harris de suivi de points caractéristiques. Pour chaque image, les points d'intérêts sont déterminés et appariés avec l'image précédente. Les points 3D de l'objet étant connus, la mise en correspondance avec les points 2D permet d'estimer la pose de la caméra. Pour éviter les problèmes d'appariements, une optimisation de la pose est calculée en intégrant les informations issues des contours et des points d'intérêts.

Pupilli et Calway [126] ont présenté un système de tracking basé sur les filtres particulaires. Les filtres particulaires, sont utilisés pour représenter une distribution quelconque en utilisant suffisamment d'échantillons pour se rapprocher du filtre bayésien optimal et réaliser un système de tracking robuste. Cependant, le nombre de particules augmente avec le nombre de points d'intérêt ce qui ralentit le système de tracking.

1.5.2 Tracking pour les applications de réalité augmentée

1.5.2.1 Tracking des cibles passives

Nous allons citer maintenant les stratégies de tracking utilisées dans les applications récentes de réalité augmentée. Plusieurs technologies de tracking emploient des systèmes multi-capteurs, cela a été décrit dans la dissertation d'Azuma [14] et Rolland [135]. Toutefois, le faible coût des caméras et leur disponibilité, les classent comme étant le dispositif idéal pour le tracking. Comme, nous l'avons vu dans les sections précédentes, plusieurs systèmes de tracking sans-marqueur ont été proposés dans la littérature, mais cela exige des hypothèses et des restrictions sur la forme de l'objet d'intérêt et exige parfois l'intervention de l'utilisateur pour l'initialisation du système de tracking.

La plupart des systèmes de réalité augmentée, utilisent des marqueurs facilement reconnaissables dans l'environnement, ces marqueurs peuvent être passifs (cibles imprimées) ou actifs (contenant des diodes émettrices), ces deux type de marqueurs sont très utilisés dans les applications de réalité augmentée.

Hoff et al. [69] ont utilisé un système basé sur des cibles codées circulaires pour un casque de réalité augmentée dans une application de maintenance. Les cibles sont placées partout dans l'envi-

ronnement et l'utilisateur voit des instructions et des augmentations sur la scène réelle par le biais de son casque. Les cibles sont détectées par la segmentation d'image en régions noires et blanches, les régions connexes sont ensuite calculées pour déterminer les centroïdes. L'estimation de la pose est calculée en utilisant 5 cibles codées. Le système calcule la pose de la caméra pour chaque image, ce qui limite les erreurs par rapport aux systèmes utilisant les informations des images précédentes.

Koller et al. [87] ont employé aussi des marqueurs visuels en faisant une modélisation du mouvement de la caméra avec un système d'équations exprimant la cinématique du mouvement. Les cibles utilisées sont des carrés contenant un code binaire pour permettre l'identification de la cible dans l'image. Le tracking est effectué localement autour des cibles et le filtre de Kalman étendu est utilisé pour l'estimation des états du système. Le système s'auto-initialise quand les cibles ne sont plus visibles par la caméra.

Stricker et al. [149] ont développé une approche différente d'identification de cibles se basant sur la détection de contour des objets pour la localisation de la caméra. Un M-estimateur est appliqué sur les mesures afin de localiser les points des contours et rendre la détection plus robuste aux données aberrantes, l'identification est effectuée en utilisant des barres placées à côté des cibles. Le système de tracking permet le suivi des cibles dans l'image, il est rapide et robuste et peut être ré-initialisé en cas d'échec du suivi.

1.5.2.2 Tracking des cibles actives

Les systèmes de tracking des cibles actives, utilisent des LEDs montées sur un visiocasque et traquées par des caméras pour localiser la tête de l'utilisateur. Les marqueurs actifs permettent une localisation très précise et rapide, ce qui rend le système très performant et permet de régler le problème lié aux mouvements brusques. L'un des systèmes les plus connus à base de LEDs a été développé à l'université de Caroline du Nord (UNC) par Wang et al. [157] et Ward et al. [159]. Des photo-diodes sont montées sur un visiocasque permettent la réception des rayons infrarouges émis par les diodes posées sur les objets dans la scène, ces capteurs possèdent une grande résolution et ils sont très rapides. L'avantage avec ce type de système c'est qu'il n'a pas de problème d'identification de cibles ou de mise en correspondance. Welch et Bishop [160] ont remplacé les fondements mathématiques du système par une architecture basée sur le filtre de Kalman étendu appelé SCAAT (Single Constraint At A Time) pour optimiser et commercialiser le système.

1.5.3 Bilan sur les méthodes de tracking

Dans les sections précédentes, nous avons subdivisé le problème de suivi en deux parties : tracking pour la vision et tracking pour la réalité augmentée. Concernant le suivi par système de vision, plusieurs méthodes ont été présentées, elles sont basées sur l'extraction des primitives géométriques dans l'image, telles que, les contours, les points ou les segments. Ces algorithmes représentent le fondement théorique pour le développement d'autres méthodes spécifiques pour un type d'application visée.

Un intérêt particulier a été donné par la suite aux systèmes de tracking en réalité augmentée. Les techniques de suivi dans ce domaine repose sur le type d'amer visuel utilisé. En effet, si les cibles sont désignées par des motifs visuels imprimées, il s'agit donc de suivi de cibles passives. Ces cibles sont caractérisées par un certain code qui les distingue par rapport aux autres objets présents dans la scène. Le problème avec les cibles passives est le temps de traitement nécessaire pour extraire la forme de la cible et identifier son motif. D'autre part, les technologies basées sur les cibles actives représentent des solutions matérielles et s'appuient sur des dispositifs qui émettent un signal (généralement des LEDs) pour se localiser dans un environnement instrumenté. Ces systèmes sont rapides et précis, cependant, de tels dispositifs nécessitent des sources d'énergie ce qui limite leur utilisation dans des endroits en intérieur.

1.6 Travaux récents dans la réalité augmentée

Les applications de la réalité augmentée sont subdivisées en deux types de systèmes : les systèmes en vision indirecte où l'utilisateur perçoit par le biais d'une caméra le monde réel et les systèmes en vision directe où l'utilisateur voit directement le monde réel par des interfaces dédiées de type visières ou semi transparentes.

1.6.1 Travaux récents en vision indirecte au sein du laboratoire IBISC

En vision indirecte, le système de réalité augmentée est contraint à faire un tracking de cibles dans un flux vidéo, les traitements sont réalisés en temps différé et la notion de temps de retard n'est pas abordée. Par contre, dans les systèmes de réalité augmentée en vision directe, la notion du temps est très importante. Tous les traitements sont faits en temps réel et différents capteurs sont utilisés pour accélérer le rendu visuel. Dans un système de réalité augmentée en vision indirecte seul l'alignement spatial qui est important. Pour l'obtenir, il est nécessaire que le recalage des objets virtuels sur le monde réel soit exact, précis et conservé dans le repère de visualisation de l'utilisateur. Si la superposition n'est pas correcte, des erreurs de recalages sont générées.

1.6.1.1 Travaux de Ababsa

Les travaux de Ababsa [1] se sont basés sur le problème du recalage 3D automatique d'objets de forme libre en vision indirecte. L'auteur a utilisé la clinométrie des objets de forme libre pour faire du recalage 3D en utilisant un schéma traditionnel de la reconnaissance d'objets en vision qui est constitué de trois étapes : l'extraction de primitives, la construction de la base de données du modèle et enfin l'appariement et le recalage.

Deux approches clinométriques ont été élaborées dont les modules diffèrent en fonction du type d'images utilisé. Dans la première approche qui est la photoclinométrie, l'auteur a démontré que, sous certaines conditions, la photoclinométrie permet d'extraire à partir d'une seule image de luminance d'un objet de forme libre un invariant projectif de type photométrique (non géométrique) permettant ainsi de reconnaître les points du modèle vus dans l'image, il s'agit en l'occurrence des répartitions angulaires. L'exploitation de cet invariant permet de construire une base de données qui répertorie d'une manière simple et efficace les informations pertinentes sur le modèle permettant ainsi de l'apparier avec les primitives image. Afin de démontrer l'efficacité de cette méthode, une série d'images synthétiques représentant le modèle dans des poses différentes a été employée. Cependant, la mise en oeuvre pratique de la solution photoclinométrique reste difficile à réaliser. En effet, l'hypothèse de base qui impose que la source de lumière illuminant la scène soit directionnelle est trop contraignante et difficile à mettre en oeuvre, ce qui a amené à travailler sur des images de synthèse. Ainsi, les conclusions établies à partir de ces images possèdent une portée plus ou moins limitée.

Dans l'approche clinométrique, l'exploitation des normales à la surface d'un objet de forme libre permet de construire des invariants géométriques (à la rotation et à la translation) locaux qui peuvent être utilisés pour établir des appariements 3D/3D entre des régions de l'objet de la scène et celles du modèle. La solution clinométrique, a été mise en oeuvre dans un système de vision constitué d'un capteur stéréo actif (caméra CCD + projecteur de grille laser). Une méthode de détection subpixel des primitives dans l'image a été développée afin d'assurer une bonne précision de reconstruction des points 3D (précision $< 1\%$). L'originalité dans cette application réside dans la discrimination entre les différentes régions de surface d'un seul modèle grâce aux invariants clinométriques, ceci a une implication directe dans l'efficacité de la mise en correspondance. En effet, le calcul des invariants clinométriques permet des appariements robustes et rapides entre les régions de l'objet de la scène et celles du modèle de la base de données. Ceci simplifie le processus de recalage et le rend fiable. L'approche clinométrique a été testée sur un objet réel de forme libre, les résultats obtenus pour les tests effectués sont très probants. Ainsi, deux méthodes de recalage 3D automatique d'objets de forme

libre ont été développées en utilisant uniquement la clinométrie. Les deux approches développées ont été utilisées pour faire ensuite du recalage d'objet en vision indirecte.

1.6.1.2 Travaux de Merad

Merad [108] a présenté une méthode originale qui permet à l'aide d'une représentation en squelettes, de reconnaître des objets (figure 1.25). Le processus de squelettisation permet d'obtenir deux graphes, un graphe 3D et un autre 2D. Le graphe 3D est obtenu à partir du squelette 3D homotopique et le graphe 2D est construit à partir du squelette 2D. Cette technique permet de construire des squelettes robustes et ainsi de faciliter l'étape de mise en correspondance. Dans l'étape de mise en correspondance, l'auteur effectue un isomorphisme entre le graphe 3D de la base de données et le graphe 2D obtenu à partir de l'image. Dans cette approche, cette tâche est difficile car cette mesure doit évaluer le degré de ressemblance entre structures similaires de deux sous-graphes, afin de gérer le problème d'occultations. Merad a adopté la méthode d'isomorphisme de Siddiqi [141] [142] pour l'étape de mise en correspondance de graphes. Cette méthode utilise les signatures topologiques des noeuds et permet d'indexer une base de donnée et d'apparier un-à-un les noeuds des deux graphes.

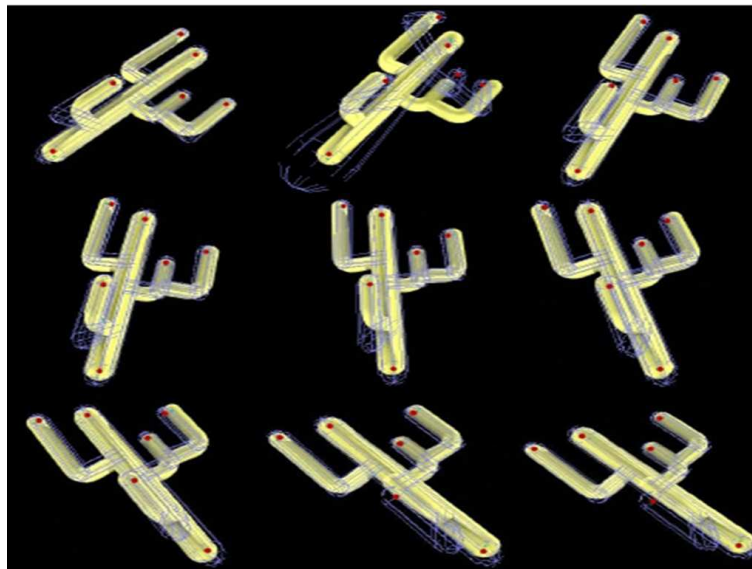


FIG. 1.25 – Suivi d'objet avec la méthode de squelettisation [108].

Afin de démontrer l'efficacité de cette approche, des tests sur une série d'images réelles représentant un modèle dans des poses différentes ont été effectués. L'algorithme a démontré sa robustesse par rapport aux occultations et aux auto-occultations et son efficacité en présence d'autres objets. Un algorithme de suivi d'objets 3D de forme libre par les squelettes a été proposé aussi.

1.6.2 Travaux récents en vision directe

Dans les systèmes de la réalité augmentée en vision directe, différents dispositifs sont employés pour compenser les limitations des divers capteurs et réduire le temps de traitement. Les données des systèmes d'acquisition tels que les dispositifs de suivi d'objet (caméras) et les capteurs de mouvements (centrale inertielle, traqueur optique, etc) doivent être synchronisés spatio-temporellement avec le point de vue de l'utilisateur. Cette synchronisation est très difficile à obtenir lorsque les dispositifs sont en mouvement. Ceci représente un challenge pour les applications de réalité augmentée.

1.6.2.1 Travaux de Foxlin et al.

Les travaux de Foxlin et al. [51] se sont basés sur le tracking hybride pour les applications en temps réel de réalité augmentée. Les capteurs de mouvements souffrent généralement de latence, de portée restreinte, et coûtent très cher. Pour résoudre ce problème, un système de tracking hybride a été proposé, composé de plusieurs capteurs, ce système permettra de compenser les limitations de chaque capteur.

Les auteurs ont réalisés des produits commerciaux de tracking pour la navigation et la localisation dans des environnements de réalité augmentée. En effet, deux dispositifs de tracking hybrides ont été commercialisés : le *IS – 300* et le *IS – 600*. Le *IS – 300* permet de traquer l'orientation avec trois degrés de liberté, en utilisant un gravimètre sensible à l'inclinaison pour éviter toute dérive gyroscopique.

D'autre part, le *IS – 600* est un système de tracking hybride acoustique/inertiel avec six degrés de liberté, trois en position et trois en orientation. Il suit le changement en orientation et en position en intégrant les sorties de ses gyroscopes et accéléromètres.

Les deux systèmes de tracking de mouvement, le *IS – 300* et le *IS – 600*, reposent sur des mini-capteurs inertiels (InertiaCube d'Intersense [75]) et considèrent leurs mesures primaires. Cette technologie offre plusieurs avantages potentiels :

- Faible latence.
- Prédiction du mouvement.
- Bonne résolution.
- Grande portée.
- Pas d'interférence.
- Portabilité (ne nécessite pas une source d'énergie).

Les données fournies par un système de navigation inertielle sont les mesures de base pour les deux systèmes de tracking *IS – 600* et *IS – 300*, ils mesurent l'orientation d'un objet en intégrant les vitesses angulaires obtenues à partir des trois gyroscopes orthogonaux attachés à l'objet. Pour obtenir la position, trois accéléromètres sont aussi rattachés aux trois axes orthogonaux de l'objet en mouvement. Ils mesurent le vecteur d'accélération totale de l'objet par rapport au repère de la centrale inertielle.

Ce vecteur d'accélération peut être converti en position dans le référentiel de la terre en utilisant l'orientation instantanée de l'objet déterminée par le gyroscope. La position est obtenue en soustrayant l'accélération de la gravité aux accélérations mesurées et en effectuant une double intégration partant d'une position initiale connue.

1.6.2.2 Travaux de Foxlin et Naimark

Les auteurs dans [52][118] ont établi un système portable de tracking hybride composé de caméras et d'une centrale inertielle (figure 1.26-a). Ce système est considéré comme étant le premier système commercial de la réalité augmentée car il répond à plusieurs critères :

- Il est petit, léger et peut être confortablement porté à la ceinture.
- Les capteurs inertiels sont fixés sur le visiocasque ou à un dispositif tenu à la main.
- Il peut être facilement réalisé, étant donné qu'il utilise des composants peu coûteux et disponibles.
- Il dispose d'un logiciel de calibration automatique.
- Il est robuste face aux variations de luminosité, à l'occultation et au mouvement brusque.

Ce dispositif est plus petit et plus performant que la plupart des technologies disponibles dans le commerce pour le guidage automatique des robots et des véhicules. Le système comporte un noyau de fusion des données de capteurs pour assurer trois fonctions principales :

- Tracking : pendant le tracking, le noyau de fusion des capteurs utilise les algorithmes optimaux du filtre de Kalman pour fusionner les données de la centrale inertielle et de la caméra quand

ces dernières sont disponibles. Cette combinaison permet d'aboutir à un système de tracking ayant une possibilité prédictive et une exactitude absolue des enregistrements des données.

- Calibration automatique : les auteurs utilisent des algorithmes d'identification pour estimer et corriger les paramètres intrinsèques de la caméra (focale, facteurs d'échelle, point principal, distorsions) et les paramètres extrinsèques (position et orientation des caméras par rapport à la centrale inertielle). Ces algorithmes d'auto calibration permettent d'obtenir une bonne estimation des paramètres sans avoir à utiliser des équipements coûteux.
- Auto apprentissage : avant d'utiliser le système de tracking dans un nouvel environnement, les cibles artificielles sont initialisées et un apprentissage du système dans cet environnement initial est effectué pour tester les capacités de généralisation de l'algorithme.

Le prototype du système est appelé VIS-Tracker, il est composé de plusieurs caméras et d'une centrale inertielle de type InertiaCube2 d'Intersense [75] : une très petite et très puissante centrale inertielle.

La figure 1.26-b montre un utilisateur portant le système de tracking sur un HMD se déplaçant autour de la zone d'essai avec des marqueurs installés. La plupart des marqueurs sont placés au plafond car le but est de permettre le tracking de l'utilisateur.

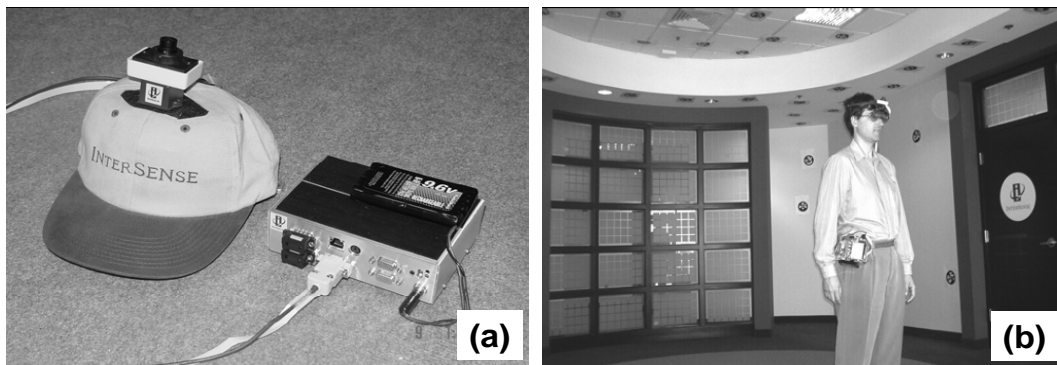


FIG. 1.26 – Prototype du système de tracking VIS-Tracker d'Intersense [52]. (a) Le dispositif de tracking. (b) Utilisateur portant le système de tracking.

En raison de la conception hybride, le système de tracking est relativement robuste aux occultations. Quand le système détecte une cible partiellement occultée (figure 1.27-a), il arrive à maintenir le tracking, tant qu'il y a au moins deux marqueurs non occultés. Avec un seul marqueur, le tracking s'arrête après quelques minutes. Par contre, lorsque le marqueur est entièrement occulté, il peut maintenir le tracking grâce à la centrale inertielle pendant environ cinq secondes.

Les auteurs ont développé des algorithmes de traitement d'images. Ces algorithmes sont robustes à un éventail de conditions d'éclairage qu'on pourrait rencontrer dans des applications pratiques. Pour tester la robustesse du système, l'expérience a été effectuée en se déplaçant dans l'espace avec les lumières commutant en marche et en arrêt. La caméra a été orientée vers un mur de verre ensoleillé. L'expérience s'est prolongée dans un couloir avec un arrière plan très sombre. Le système a été robuste et a continué le tracking malgré les variations de la lumière. La figure 1.27-b montre quelques images de cette expérience.

En conclusion, Foxlin et Naimark ont développé un système de tracking hybride constitué d'une caméra et d'une centrale inertielle. Ils ont conçu un nouveau modèle de marqueurs basé sur les cibles codées. L'algorithme proposé permet l'extraction des cibles à partir de la scène au cours du temps. À la différence des autres méthodes basées sur les système vision-inertiel pour la réalité augmentée, ce système a pour objectif de réaliser un produit commercial qui sera vendu aux industriels et qui pourra être utilisé dans des environnements semi-hostiles.

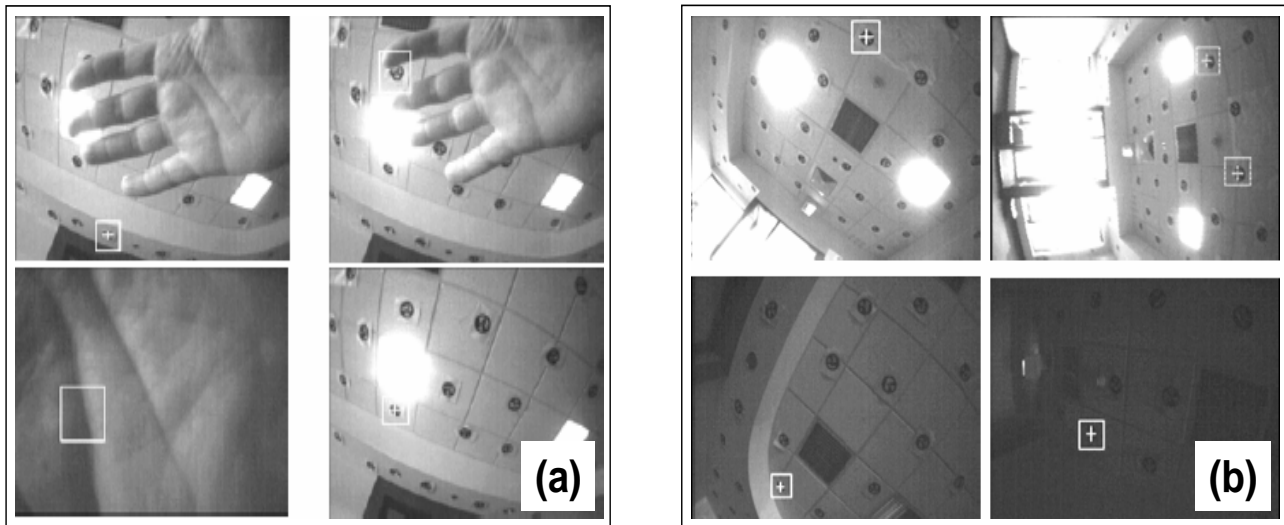


FIG. 1.27 – Évaluation de la robustesse du système [52]. (a) Marqueurs partiellement occultés. (b) Identification de cibles en variant la luminosité.

1.6.2.3 Travaux de Chai et al.

Dans leur étude, Chai et al. [26] ont proposé un système basé sur des caméras et des centrales inertielles montées sur un casque de réalité augmentée pour prédire la pose et améliorer le recalage. Le dispositif permet l'élaboration d'un système de réalité augmentée portable, les capteurs inertiels et les caméras CCD sont légers et peuvent être facilement montés sur un casque. Les objets d'intérêts sont localisées à l'aide d'un système d'identification de cibles, aucune connaissance a priori de l'objet n'est nécessaire, ce qui rend le système de tracking autonome et plus rapide. Les centrales inertielles détectent rapidement les mouvements et maintiennent le tracking en temps réel et les caméras sont utilisées pour corriger les dérives des centrales inertielles.

Pour l'expérimentation, les auteurs se sont basés sur un prototype de système de réalité augmentée développé au CSM (Colorado School of Mines). Le système se compose d'un casque instrumenté d'un capteur de position OPTOTRAK 3020, monté sur le mur et d'un PC lié par une carte d'interface (figure 1.28). Le contrôleur illumine les LEDs montées sur le casque ce qui permet la détermination de la position du casque.

L'objectif de l'étude est de développer et implémenter un algorithme pour fusionner les données de deux sources de capteurs afin d'améliorer le recalage des objets dans un système de réalité augmentée. Le casque d'affichage contient un ensemble de capteurs à ultrasons et des caméras CCD (figure 1.29). Deux filtres de Kalman étendu sont utilisés pour prédire le mouvement de la tête de l'utilisateur et la structure 3D des points observés par l'utilisateur dans la scène. Une boucle récursive est employée entre les deux filtres de Kalman afin que les estimations initiales des positions de la tête de l'utilisateur et les points caractéristiques puissent être mis à jour pour approcher la solution désirée.

L'algorithme proposé est testé en utilisant des mouvements synthétiques et réels. Les résultats de prédiction obtenus en utilisant un mouvement synthétique sont précis. Même avec un mouvement réel et de vraies images, les erreurs de prédictions sont acceptables. Les tests expérimentaux ont démontrés que l'utilisation d'un système stéréoscopique améliore les performances du recalage et du tracking.



FIG. 1.28 – Système de tracking développé au CSM [26].



FIG. 1.29 – Visiocasque de réalité augmentée instrumenté de capteurs : caméras CCD, centrales inertielles, LEDs [26].

1.6.2.4 Travaux de You et Neumann

You et Neumann [167] ont présenté une approche hybride pour le recalage en réalité augmentée en utilisant une technologie de tracking basée sur un système composé de caméra et de capteur inertiel (figure 1.30). Le capteur inertiel a l'avantage d'être rapide et robuste, cependant, l'inconvénient avec ce capteur c'est qu'il dérive au bout d'un certain temps de fonctionnement. Le tracking basé vision est précis pour des durées importantes mais le problème avec les caméras c'est l'occultation et le temps de calcul. Les auteurs ont exploité la complémentarité de ces deux technologies pour compenser les faiblesses de chaque capteur. Une analyse quantitative de la sensibilité par rapport aux erreurs d'orientation dans le tracking a été étudiée. Pour fusionner les données du capteur inertiel et de la caméra, une calibration précise des repères liés aux capteurs est effectuée.

Les auteurs ont proposé une approche pour la reconnaissance de cibles, cette méthode basée sur l'analyse des composantes principales (ACP) a été développée pour détecter et reconnaître les marqueurs de forme carrée dans l'image en temps réel. La procédure de détection est réalisée en trois étapes :

- Détection globale : la position prédite de l'image précédente est utilisée pour déterminer les régions de l'image où le gradient est élevé.
- Compensation de la projection : le gradient est utilisé pour normaliser l'intensité afin de compenser les déformations géométriques des régions identifiées.
- Détection fine : ajuster le marqueur et le comparer avec les modèles pré-définis.

Dans l'étape d'identification de la cible, chaque marqueur détecté est projeté dans l'espace ACP et ses coefficients de projection sont déterminés. Ensuite ces coefficients sont comparés avec d'autres coefficients enregistrés dans une base de données pour faire la mise en correspondance (figure 1.31).



FIG. 1.30 – Capteur hybride : caméra et gyroscope [167].

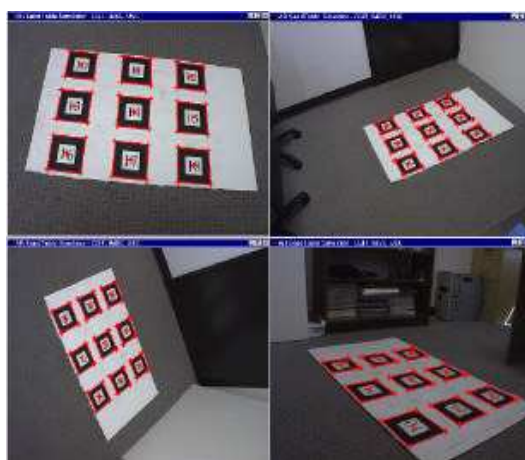


FIG. 1.31 – Extraction des cibles de l'image [167].

1.7 Contexte

L'élaboration d'un système de réalité augmentée effectif exige la précision et la robustesse pour la localisation des objets dans leur environnement. La plupart des systèmes de réalité augmentée sont basés vision et ne permettent pas la gestion d'occultations ou le flou de mouvement.

Dans la littérature plusieurs systèmes de réalité augmentée ont été développés. Les technologies basées sur les cibles actives occupent une place prépondérantes dans diverses applications de tracking en environnement instrumenté. Cependant, ce type d'application nécessite des sources d'énergie ce qui limite leur utilisation dans des endroits spécifiques isolés des interférences et des perturbations.

D'autres systèmes utilisent des cibles passives, mais le problème avec ce genre de systèmes est le temps de traitement dû au nombre important de calculs pouvant introduire une instabilité numérique. Les systèmes multi-capteurs permettent l'amélioration des systèmes basés vision et ils sont employés pour leur robustesse et leur rapidité. Ces systèmes hybrides exploitent la complémentarité des capteurs et compensent les erreurs de chaque capteur. Toutefois, les difficultés de mise en oeuvre de ces systèmes, notamment la calibration, la fusion de données hétérogènes et les erreurs générées, compromettent leur utilisation.

L'analyse et la synthèse des principaux travaux réalisés dans la littérature, a permis d'en dégager les dispositifs utilisés, les problèmes rencontrés et les solutions proposées. Dans le cadre de cette thèse, nous nous sommes particulièrement intéressés à la classe des travaux qui portent sur le suivi multi-capteurs. Dans ce domaine, les travaux de Foxlin et Naimark [52] et You et Neumann [167] sont les plus éminents. Ces auteurs ont orienté leurs recherches vers les systèmes hybrides de fusion de données et les architectures de suivi robuste et de recalage en temps réel. Toutefois, un certain nombre de problématiques non résolues par les solutions existantes, nous a conduit à la proposition d'une approche originale de suivi multi-capteurs.

1.8 Motivations

L'absence aujourd'hui d'un système complet qui propose une solution globale pour le suivi d'objets et la gestion d'occultations a constitué notre principale motivation dans cette thèse. La mise en place d'un tel système, représente un aspect intéressant à approfondir et la conception des divers modules qui le composent, permettra une analyse exhaustive de son fonctionnement.

Des verrous technologiques et scientifiques associés à l'élaboration d'une architecture de suivi en présence d'occultations, nous ont poussés à proposer des solutions originales à plusieurs niveaux de notre étude. En effet, nous suggérons un système qui comporte des modules de tracking réalisant un suivi d'objets d'intérêts quelques soient les conditions réelles de l'environnement. Cette solution multimodale a permis d'orienter nos travaux vers des perspectives prometteuses et de définir les critères fondamentaux pour la mise en oeuvre d'une architecture efficace de suivi en réalité augmentée.

De nombreux travaux menés à ce jour dans le cadre de suivi d'objets, sont fondés sur des systèmes basés vision. Toutefois, les méthodes de reconnaissance de cibles développées dans la littérature, présentent des limitations concernant le temps de calcul ou bien le nombre d'amers visuels utilisés. Ceci nous a motivé à proposer notre propre système de localisation par la vision. En outre, les limitations des méthodes d'estimation de la pose existantes, nous a permis d'élaborer une technique novatrice qui surmonte les problèmes de précision et de temps de traitement. Cette partie en particulier, constituera un point d'étude privilégié de notre travail.

Des solutions limitées concernant la réalisation des systèmes multi-capteurs, nous ont incité à développer notre système hybride de tracking. Ceci a conduit aussi à proposer des solutions originales pour résoudre les problèmes de calibration du dispositif de tracking et la suppléance et la gestion de l'information issue du système multi-capteurs. Le système hybride ainsi réalisé permet de régler les problèmes d'occultations, du flou de mouvement et de variations des conditions expérimentales.

1.9 Objectif et solutions proposées

L'objectif de cette thèse est d'établir une architecture multimodale permettant de gérer différents modules de suivi selon les conditions de visibilité de la cible. L'approche multimodale proposée emploie plusieurs capteurs et méthodes, associés de manière à surmonter les variations des conditions réelles de l'environnement. Nous nous intéressons aux problèmes d'alignement visuel des entités virtuelles sur la scène réelle et au suivi robuste des cibles dans les séquences d'images. Afin d'obtenir une composition géométriquement correcte de l'incrustation d'éléments virtuels, il est nécessaire de

calculer la pose de la caméra à chaque instant pour positionner correctement l'objet virtuel dans la scène. Nous nous intéressons tout d'abord à l'extraction des indices visuels des cibles et à l'augmentation avec des objets virtuels, en développant des algorithmes d'identification et d'estimation de pose. Nous proposons un algorithme de reconnaissance de cibles carrées et une méthode hybride d'estimation de la pose de la caméra. Cette première étape de tracking de cibles visibles constitue le module initial de notre architecture de suivi.

Dans un second temps, nous voulons établir un système de réalité augmentée robuste qui permet la gestion d'occultations. En effet, plusieurs systèmes de réalité augmentée utilisent des systèmes de vision pour la localisation et permettent un suivi visuel des cibles lorsque celles-ci sont visibles par la caméra. Dans notre étude, nous voulons, tout d'abord, définir un système capable de résoudre le problème d'occultations partielles, en utilisant une méthode de suivi robuste de points d'intérêt. L'algorithme d'estimation robuste conçu, permettra le maintien du suivi de cibles malgré les incertitudes sur le modèle, les erreurs de mesure et les changements de l'environnement.

Ensuite, nous nous intéressons au problème d'occultations totales des indices visuels. Nous proposons, donc, d'inclure un module de tracking hybride dans l'architecture globale de suivi. Cette partie est basée sur un algorithme de suivi multi-capteurs pour la localisation de la caméra lorsque tous les indices visuels sont occultés. Notre système hybride combine un capteur inertielle à un système de vision. Cette approche collaborative est motivée par le fait que les systèmes de vision peuvent échouer lors du suivi (cibles occultées, flou de mouvement, etc), alors que les capteurs inertiels sont plus robustes. De plus, cette approche tiendra en compte des contraintes liées à l'utilisation de ces deux capteurs ainsi que le réglage optimal des paramètres internes du dispositif. Enfin, nous voulons intégrer les différentes solutions proposées dans une architecture globale de suivi, permettant ainsi d'établir un système de tracking multimodal (figure 1.32). Ce système utilisera l'approche robuste pour le suivi lorsqu'un certain nombre de points sont visibles par la caméra et la méthode hybride quand tous les indices visuels sont occultés, ce qui permettra de localiser le dispositif de tracking si le système de vision n'arrive plus à identifier les cibles dans l'image.

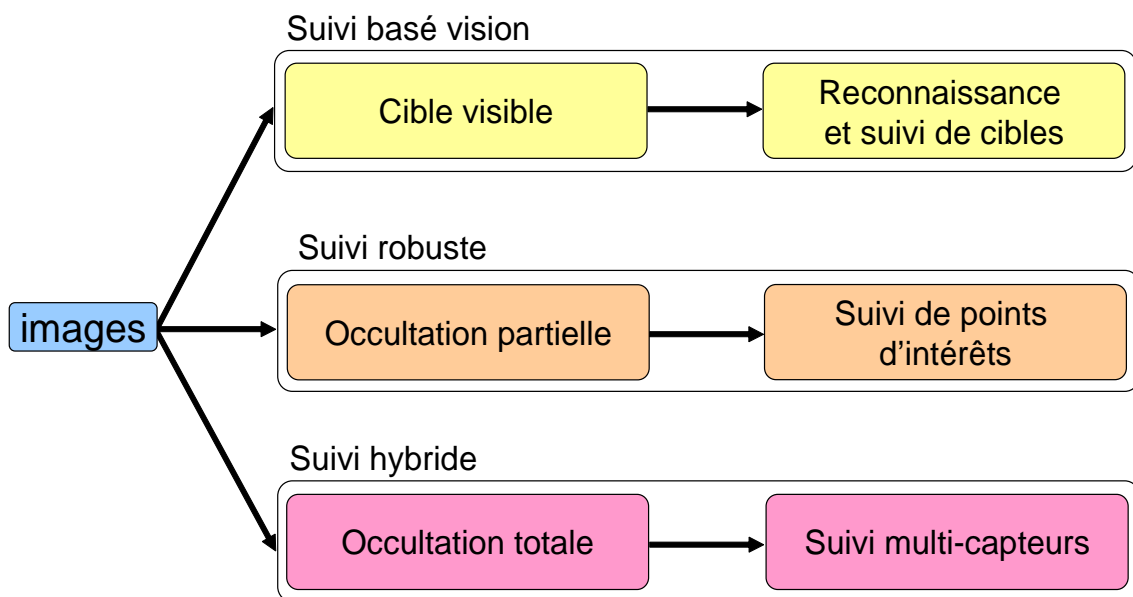


FIG. 1.32 – Schéma synoptique du système de suivi et de gestion d'occultations.

1.10 Conclusion

La réalité augmentée est un domaine dont beaucoup de travaux sont en cours. Son approche pluridisciplinaire convergeant vers un environnement rassemblant plusieurs champs de recherche. Les limitations sont aujourd'hui technologiques et scientifiques qui nécessitent une grande avancée pour passer à des solutions industrielles. Les techniques d'alignements doivent aussi progresser, les solutions hybrides semblent ouvrir une nouvelle voie d'utilisation pour combler les erreurs statiques et dynamiques.

Ce chapitre a été consacré à la définition de la réalité augmentée et à la présentation de sa taxonomie technique et fonctionnelle ainsi que les principaux verrous rencontrés pour l'élaboration d'un système effectif. Nous avons présenté dans ce chapitre plusieurs systèmes de réalité augmentée décrits dans la littérature. L'étude de ces systèmes a permis d'en extraire les principales fonctionnalités à savoir l'architecture, le suivi temps réel, la prédiction du point de vue dans le cas de la vision directe ainsi que les modalités de présentation des informations visuelles. À ce tour d'horizon s'ajoute la description de systèmes et les solutions techniques et scientifiques correspondant aux différents verrous abordés par les chapitres qui constituent cette thèse.

Enfin, le contexte de notre travail, les principales motivations et l'objectif de la thèse ont été présentés. L'approche multimodale proposée combine des capteurs et des techniques pour gérer diverses conditions expérimentales. Le système développé se démarque des travaux existants en élaborant une architecture hybride de suivi de cibles et de gestion d'occultations pour les applications de réalité augmentée. Le but de notre approche est de mettre en oeuvre un système de suivi hybride qui servira comme base pour d'autres systèmes de tracking multi-capteurs.

Dans le chapitre suivant, nous nous intéressons au suivi de cibles et à l'estimation de la pose de la caméra. Cette première partie constitue le premier module de notre architecture multimodale de suivi basé sur un système de localisation par la vision.

Chapitre 2

Méthodes basées vision pour l'estimation de la pose

2.1 Introduction

L'estimation de la pose de la caméra est une étape importante dans les applications de réalité augmentée. Elle permet le recalage correct des graphiques virtuels sur le monde réel. Le problème de l'estimation de la pose est un problème non-linéaire, souvent résolu par des algorithmes d'optimisations tels que Gauss-Newton [8] [93] ou Levenberg-Marquardt [116]. La plupart des solutions itératives dépendent de l'optimisation non-linéaire de quelques contraintes géométriques. Pour des applications en temps réel, les solutions analytiques indépendantes de toute initialisation sont les plus adaptées. Néanmoins, l'estimation des distances réelles et la précision dans le recalage sont nettement inférieures à celles obtenues par les méthodes itératives.

Dans ce chapitre, nous nous intéressons au premier module constituant notre architecture multimodale de suivi. Cette première partie fonctionne lorsque les cibles sont visibles et reconnaissables. Nous avons développé deux méthodes d'identification de cibles. La première, repose sur la reconnaissance des codes à l'intérieur des cibles de forme rectangulaire présentes dans la scène. La deuxième méthode se base sur une localisation par contour actif de l'objet d'intérêt. La localisation de la caméra est effectuée à l'aide de notre algorithme hybride d'estimation de pose. Cet algorithme combine à la fois une méthode analytique et une autre itérative pour améliorer la précision et la rapidité du recalage en temps réel.

Nous allons présenter tout au long du chapitre, d'une part, les outils théoriques et mathématiques pour modéliser la caméra et la localiser dans l'espace de travail et d'autre part, l'évaluation des divers systèmes développés pour la localisation du point de vue par la vision.

2.2 Position du problème

L'estimation de la pose est la détermination de la transformation (position et orientation) entre le repère de la caméra et le repère du monde (figure 2.1). Plusieurs algorithmes d'estimation de pose ont été développés dans la littérature, cependant, ils sont généralement de types : analytique ou itératif. Les méthodes analytiques sont des algorithmes directs admettant un nombre fini de solutions. Les méthodes itératives ou numériques sont basées sur la minimisation d'un certain critère d'erreur. L'estimation de la pose de la caméra est basée sur l'extraction géométrique des primitives qui permettent la mise en correspondance des points 2D de l'image (exprimés dans le repère image) avec les points 3D de l'objet (exprimés dans le repère monde). Nous allons évoquer dans ce qui suit quelques méthodes de calcul de pose proposées dans la littérature. Nous présentons seulement les techniques qui peuvent être calculées en temps réel, car ce facteur est un point important pour nos applications.

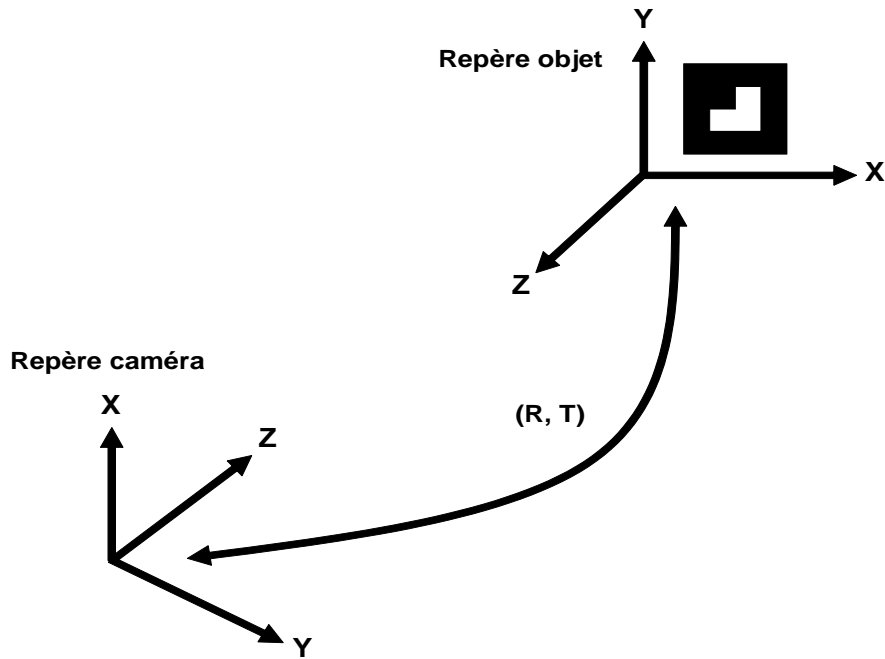


FIG. 2.1 – Les repères utilisés pour l'estimation de la pose.

L'estimation de la pose de la caméra repose sur la minimisation d'une fonction qui lie les points 3D exprimés dans le repère objet à leurs projections 2D exprimés dans l'image, par la relation suivante :

$$F(p, P, M_{int}, R, T) = 0 \quad (2.1)$$

Soit un point dans le repère objet $P_i = (X, Y, Z)^T$ et soit $p_i = (u, v)^T$, sa projection dans le repère image. En appliquant le modèle perspectif de la caméra à l'équation 2.1, on obtient l'équation suivante :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = M_{int} \begin{pmatrix} R & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.2)$$

où, s est un facteur d'échelle, (R, T) est la rotation et translation du repère lié à l'objet par rapport au repère de la caméra et M_{int} est la matrice des paramètres intrinsèques de la caméra.

Afin d'estimer la pose de la caméra, il faut tout d'abord détecter les points 2D de l'objet d'intérêt projeté sur l'image et faire une mise en correspondance entre les points 2D-3D. Ensuite, il faut résoudre la transformation perspective qui lie ces points pour déterminer les paramètres de la pose. La résolution du problème de l'estimation de la pose nécessite la calibration de la caméra.

La calibration de la caméra est une représentation de la rotation et de la translation en une transformation rigide suivie d'une projection perspective sur le plan image des points 3D. Cette modélisation est composée de deux transformations linéaires : monde-caméra et caméra-image (figure 2.2), les repères associés sont :

- (R_O, X_O, Y_O, Z_O) : est le repère associé au monde. O est l'origine de ce repère.
- (R_C, X_C, Y_C, Z_C) : est le repère associé à la caméra. C est le centre optique de la caméra.
- (R_I, u, v) : est le repère associé à l'image visualisée.

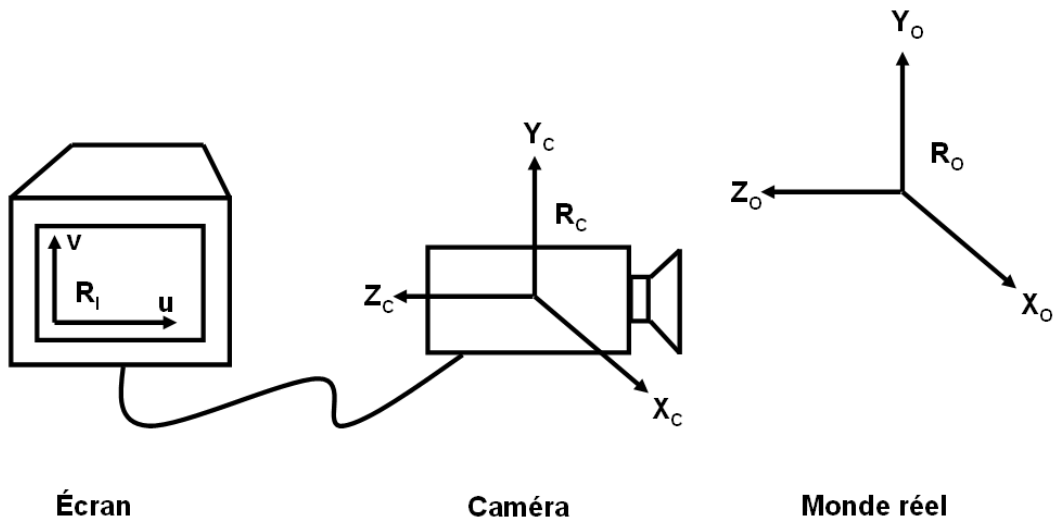


FIG. 2.2 – Repères géométriques associés à la calibration de la caméra.

Soit P_i , un point dans le repère monde R_O , sa projection p_i dans le repère image R_I est donnée par l'équation suivante :

$$\begin{pmatrix} su_i \\ sv_i \\ s \end{pmatrix} = \begin{pmatrix} k_u f & \gamma & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{CO} & t_x \\ & t_y \\ & t_z \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \quad (2.3)$$

(p_i) (M_{int}) (M_{ext}) (P_i)

Les paramètres employés dans ce modèle sont divisés en deux catégories : les paramètres intrinsèques qui sont internes à la caméra et représentés par la matrice M_{int} , et les paramètres extrinsèques, définis par M_{ext} qui peuvent varier suivant la position de la caméra dans le repère monde. Les paramètres intrinsèques sont :

- La distance focale : f .
- Les facteurs d'échelle de l'image : k_u et k_v . Les facteurs d'échelle et la focale sont souvent regroupés sous la forme suivante : $\alpha_u = k_u f$, $\alpha_v = k_v f$.
- Les coordonnées de la projection du centre optique de la caméra sur le plan image : u_0 et v_0
- La non-orthogonalité des lignes et des colonnes de cellules électroniques photosensibles qui composent le capteur de la caméra : γ . La plupart du temps, ce paramètre est négligé et prend donc une valeur nulle.

Les paramètres de la matrice extrinsèque sont représentés par :

- R_{CO} : la matrice de rotation permettant de passer du repère monde au repère lié à la caméra.
- t_x, t_y et t_z : les composantes du vecteur de translation permettant de passer du repère lié monde au repère lié à la caméra.

2.3 Mise en correspondance 2D-3D

L'identification des cibles a pour objectif d'extraire les indices visuels de l'image pour déterminer les mises en correspondances de ces points 2D avec les points 3D du modèle de l'objet.

La mise en correspondance consiste à localiser, dans les images, les projections de la même entité du modèle de la scène. La mise en correspondance s'avère être un problème délicat car de nombreuses difficultés peuvent entraîner des résultats erronés comme, par exemple : les bruits, les changements de

luminosité et les occultations. Il existe de nombreuses méthodes de mise en correspondance, cependant, dans ce travail, nous avons proposé une technique générique s'appuyant sur l'ordonnement des points caractéristiques de l'objet d'intérêt.

Cette méthode, calcule tout d'abord, l'enveloppe convexe de l'objet dans l'image pour déterminer le sens de parcours des points 2D. Ensuite, le sommet le plus proche de l'origine de l'image permet d'ordonner les points restants (figure 2.3). Les points 3D du modèle sont répertoriés dans notre système d'identification et par conséquent l'appariement des points 3D avec leurs homologues 2D est effectué en connaissant le positionnement de ces points dans l'image et dans le fichier de données 3D.

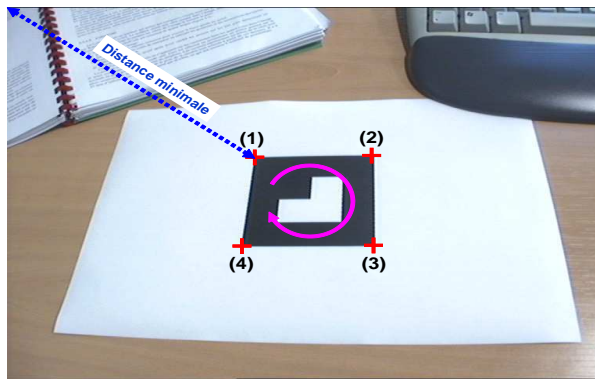


FIG. 2.3 – Ordonnement des points 2D dans l'image.

Nous allons à présent détailler le processus de reconnaissance de cibles, cette tâche est nécessaire pour extraire les primitives visuelles de l'objet d'intérêt dans l'image afin de réaliser la mise en correspondance. Nous proposons deux solutions complètement différentes, la première se base sur l'échantillonnage spatial des objets de forme quadrilatérale ayant des motifs particuliers. Quant à la deuxième méthode, elle emploie les contours actifs pour retrouver l'objet d'intérêt dans l'image. Pour cette technique, aucune connaissance a priori sur le modèle de cible n'est requise pour identifier les marqueurs.

Ces deux méthodes de reconnaissance de cibles permettent la mise en correspondance 2D-3D des points caractéristiques et représente l'étape primordiale pour les algorithmes d'estimation de la pose. Ainsi, la figure 2.4 illustre le processus de suivi et d'estimation de pose élaboré et dont les différentes parties constituantes sont développées dans ce chapitre.

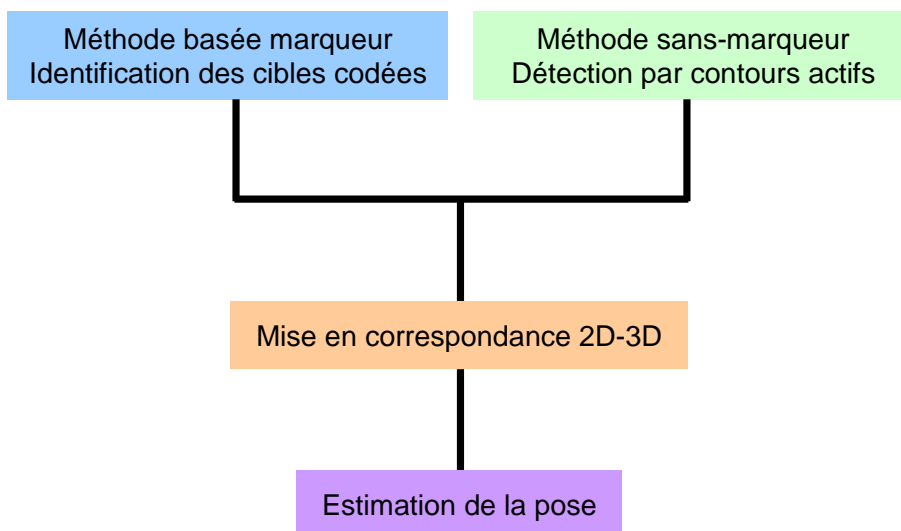


FIG. 2.4 – Processus de suivi et d'estimation de la pose.

2.3.1 Identification de cibles codées

Nous allons décrire, maintenant, les méthodes d'identification de cibles les plus connues en réalité augmentée. Nous présentons, ensuite, deux méthodes que nous avons développées pour l'extraction des cibles et la mise en correspondance des points 2D-3D. La première approche consiste à identifier les cibles à l'aide d'un algorithme de détection et d'identification de cibles codées dans l'image. La deuxième technique est basée sur une localisation spatiale des objets d'intérêts dans la scène. Cette méthode repose sur la minimisation d'un certain critère pour converger vers le contour de l'objet recherché. Cette technique sans-marqueur (markerless) présente l'avantage de ne pas requérir la forme du motif de la cible et révèle donc, une souplesse concernant le processus de reconnaissance de cible.

Plusieurs systèmes d'identification de cibles codées ont été présentés dans la littérature. Toutefois les trois systèmes les plus connus sont : ARToolKit, CyberCode et le système de cibles circulaires d'Intersense.

2.3.1.1 Méthodes d'identification de cibles en réalité augmentée

ARtoolKit : C'est un système de marqueurs utilisé pour la réalité augmentée. Grâce à ses performances de robustesse, il est utilisé dans un grand nombre d'applications en réalité augmentée et en vision par ordinateur. ARToolKit regroupe plusieurs modèles bidimensionnels de marqueurs fiduciaux. Il se prête bien à la recherche de marqueurs, la reconnaissance et l'identification. Toutefois, ses performances en matière de détection de marqueurs pourraient être améliorées. En effet, il arrive souvent que les marqueurs soient confondus entre eux ou qu'ils soient détectés par erreur en avant-plan. Les marqueurs d'ARToolKit se composent d'une bordure carrée noire entourant un modèle qui se compare à d'autres modèles pré-enregistrés dans une base pour la mise en correspondance (figure 2.5).

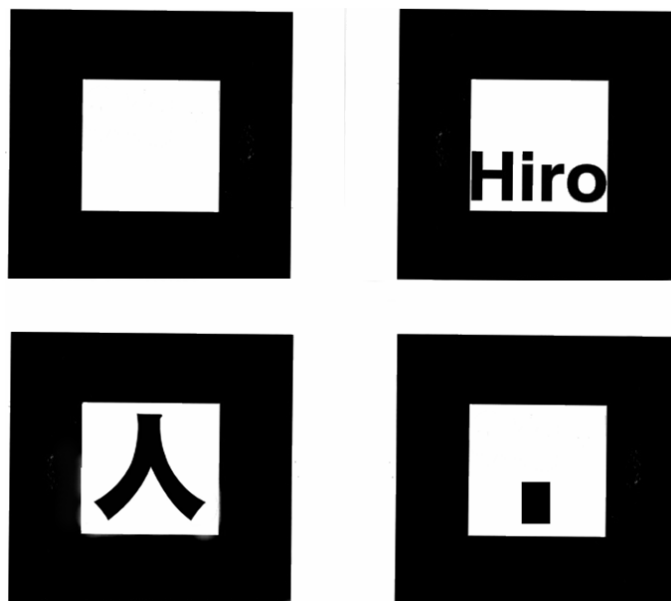


FIG. 2.5 – Modèles de marqueurs utilisés par ARToolKit.

Pour pouvoir utiliser cette technique, on a besoin d'informations sur la scène (repères, description des objets à suivre, etc.) pas forcément disponibles. C'est pourquoi les outils les plus utilisés font appel à des formes prédéfinies : des marqueurs qui serviront de repères dans le monde physique. Ils peuvent être de plusieurs types, le plus courant est une image en noir et blanc décrivant un symbole. Ces marqueurs sont utilisés par la librairie ARToolKit [81]. À partir d'un dispositif de capture, cette librairie recherche dans l'image binarisée tout objet "encadré de noir". Une fois les marqueurs détec-

tés, elle les compare avec des modèles pré-définis afin de les identifier. Elle calcule alors une matrice de transformation pour exprimer les coordonnées du marqueur dans le repère 3D associé.

ARToolKit est composé de plusieurs étapes, la première consiste à binariser l'image en utilisant un seuil approprié. Ensuite une recherche de composants connexes est effectuée afin de déterminer les régions connexes dans l'image. Les arrêtes et les coins sont alors extraits de l'image et enfin un cube virtuel est recalé sur la cible (figure 2.6).

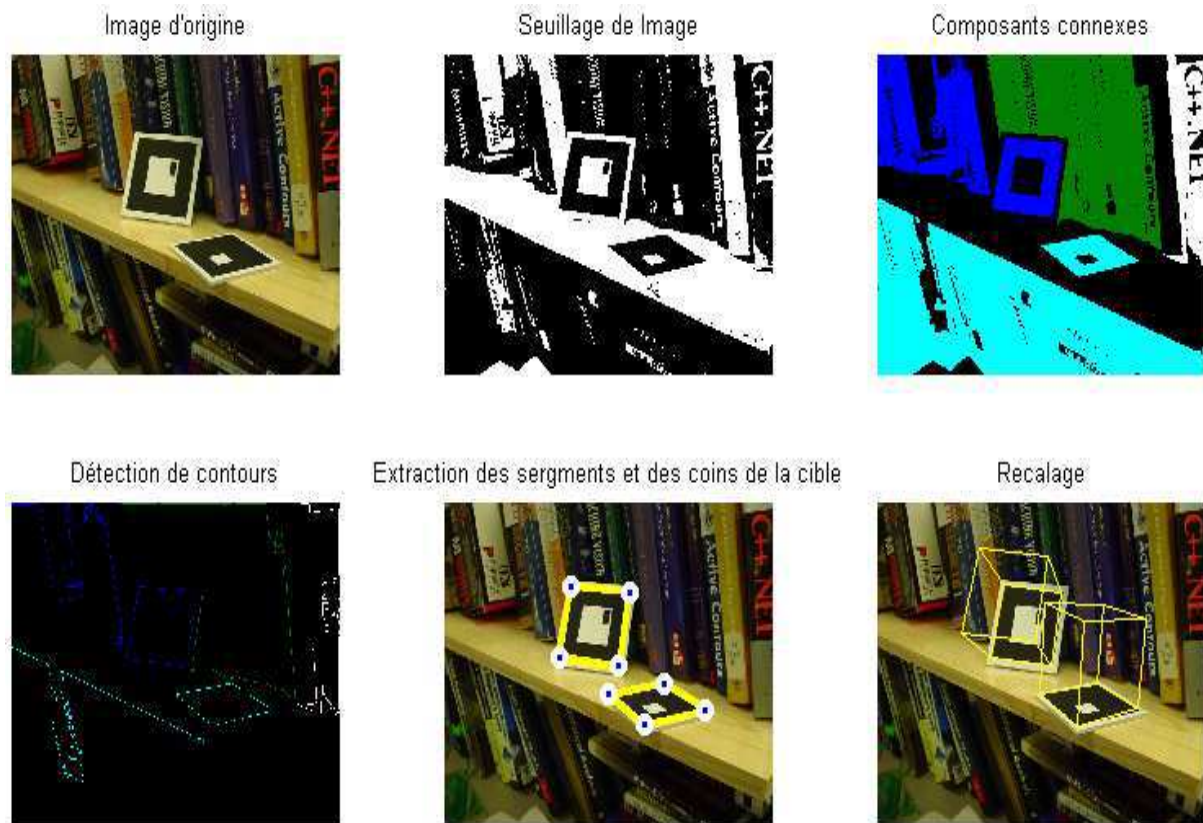


FIG. 2.6 – Processus de reconnaissance de cibles d'ARToolKit.

ARToolKit permet ainsi d'identifier tout objet du monde physique en y associant un simple marqueur, ce qui rend l'opération très facile à expérimenter. Cette librairie a subi de nombreuses modifications afin d'améliorer sa robustesse aux erreurs d'identification tout en diminuant les coûts de calcul. Cependant cette méthode a un grand inconvénient qui est, l'occultation du marqueur par un autre objet. Si le marqueur n'est pas entièrement visible, alors il n'est pas détecté. Pour résoudre ce problème, Malik et al. [102] ont proposé une méthode hybride qui utilise un marqueur pour initialiser une recherche des coins. Ainsi sur chaque marqueur, on recherche tout d'abord des régions caractéristiques (formes géométriques blanches sur fond noir) pour identifier approximativement les coins de ces régions. Une recherche locale autour de ces points est alors effectuée pour trouver précisément leurs coordonnées et ainsi identifier le marqueur. Si certains points sont occultés, l'application peut tout de même identifier le marqueur. Bien évidemment, ce type de traitement augmente considérablement les coûts de calcul.

Fiala [46] quant à lui, a proposé une modification de l'ARToolKit : l'ARTag qui n'utilise pas de fichier de description de marqueur. Pour cela, il a créé une série de 2002 marqueurs uniques qui correspondent à un codage binaire sur 36 bits. Le fait d'utiliser ce principe, alors que l'ARToolKit effectue un calcul de corrélation entre niveau de gris sur 4 positions, permet d'obtenir des taux d'erreurs d'identification (confusion entre deux marqueurs, marqueurs détectés absents et inversement) quasi-nuls. De plus, cette méthode autorise les faibles occultations par d'autres objets.

CyberCode : Le CyberCode est un système qui utilise des marqueurs visuels basé sur des cibles codées proposé par Rekimoto [132] [133]. Le système utilise une cible unique dans l'image, différentes étapes sont nécessaires pour détecter et extraire les cibles de l'image et estimer la pose.

L'algorithme du CyberCode est constitué de quatre étapes (figure 2.7) :

- a) Binarisation de l'image.
- b) Détermination des régions connexes pour trouver la barre-guide du motif.
- c) Recherche des quatre coins du marqueur en utilisant les positions et les orientations des barres-guides trouvées dans b.
- d) Décodage du code du motif après avoir vérifié les erreurs sur les bits pour déterminer un CyberCode correct.

La pose est ensuite déterminée en s'appuyant sur les contraintes liant les quatre coins dans l'image et leurs coordonnées dans le monde réel. Ceci est effectué en minimisant un critère portant sur la normale au plan grâce à l'algorithme du simplexe. Les vecteurs représentant les arêtes de la cible sont alors recalculés et les paramètres de la pose de la caméra sont estimés à partir de ces derniers. Il y a 33 bits d'information dans le code porté par la cible, ce qui fait environ 8 milliard de possibilités.

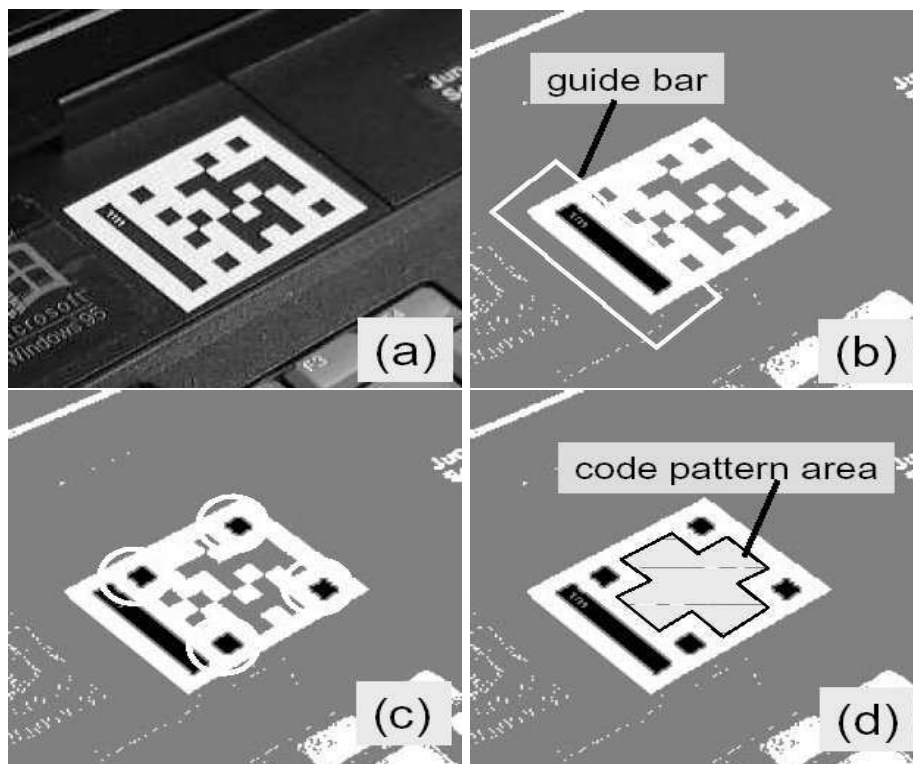


FIG. 2.7 – Processus de reconnaissance de cibles du CyberCode.

Intersense : En 2002, la société Intersense [118] a développé son propre système de cibles codées. Celui-ci s'appuie sur des cibles circulaires comme nous pouvons le voir à la figure 2.8. Bien qu'il ne s'agisse pas du premier du genre (Cho et Neumann [27] ont développé un système similaire en 1998), il est de loin le plus abouti puisque les traitements sont actuellement embarqués dans l'électronique d'un système hybride comprenant une caméra et une centrale inertielle. Les codes sont stockés sur 15 bits ce qui fait en tout 32768 possibilités.

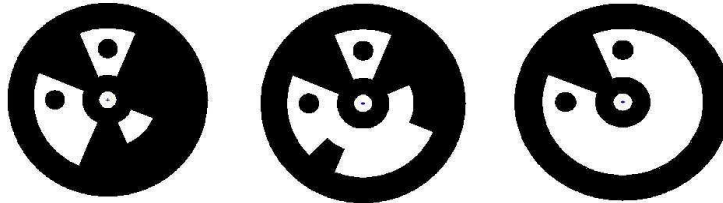


FIG. 2.8 – Exemple de cibles codées du système d'Intersense.

L'algorithme de reconnaissance des cibles, directement câblé à la suite du capteur CCD, comporte les étapes de traitement suivantes :

1. un rehaussement du contraste,
2. une détection des arêtes en employant un filtre de Sobel de noyau de taille 3×3 ,
3. une binarisation suivie d'une érosion de l'image, les zones blanches de l'image contiennent des cibles potentielles : qui sont filtrées en fonction de leurs formes,
4. le code de la cible est lu en s'aidant des points blancs au centre de la cible et des deux points noirs situés dans les quadrants. Cet ensemble de trois points forme un repère qui permet d'extraire le code.

Comme les autres systèmes de cibles circulaires, il est nécessaire d'avoir plusieurs amers dans le champ de vision pour calculer la pose de la caméra, une cible donnant un point particulier. Dans le système présenté, un minimum de quatre cibles de ce type est nécessaire.

2.3.1.2 Approche développée

Nous avons développé un algorithme de reconnaissance de cibles qui consiste à réaliser un échantillonnage spatial des zones rectangulaires de l'image pour l'extraction du code de la cible. Cette méthode est constituée de deux parties, une partie de détection et une autre pour l'identification. La reconnaissance d'objet d'intérêt est élaborée en employant des cibles codées avec un motif que nous définissons et qui permettra de distinguer nos cibles parmi les autres objets présents dans la scène.

Le système proposé apporte des améliorations en terme de rapidité et de souplesse par rapport à ARToolKit [81] ou le système de cibles circulaires d'Intersense [118]. En effet, notre méthode extrait en temps réel l'objet d'intérêt de l'image en calculant le code binaire situé à l'intérieur de la cible. Le code utilisé est composé de 16 bits, ce qui permet de réduire le temps de calcul comparé au CyberCode qui emploie 33 bits [133]. De plus, un seul marqueur est suffisant pour estimer la pose de la caméra (la pose est déterminée à partir des 4 sommets de la cible) contrairement au système d'Intersense qui exige plusieurs amers visuels pour calculer cette même pose (ce système requiert 4 cibles pour déterminer la pose).

Détection : Pour extraire l'objet cible à partir de l'image, il est nécessaire tout d'abord de détecter sa forme pour l'identifier. Afin de réduire le taux d'erreur de la détection, les images sont pré-traitées avant d'être analysées. L'image est binarisée en utilisant un seuil approprié. Puis, d'autres opérations sont appliquées pour détecter la forme d'objet. L'algorithme de la détection d'objet se compose des étapes suivantes (figure 2.9) :

1. appliquer le filtre de Canny [24] pour détecter les contours dans l'image,
2. lisser l'image à l'aide d'un filtre gaussien pour éliminer les variations de pixels selon les segments des contours en joignant les valeurs moyennes,
3. dilater l'image lissée pour supprimer les trous potentiels entre les segments de bord,
4. faire une approximation polygonale du contour,
5. trouver le nombre de sommets d'objet,
6. identifier les frontières de l'objet en tant que quatre lignes sécantes en testant la colinéarité des sommets,
7. trouver l'angle minimum entre les arêtes voisines, si les cosinus des quatre angles sont proches de zéro, alors un carré est détecté.

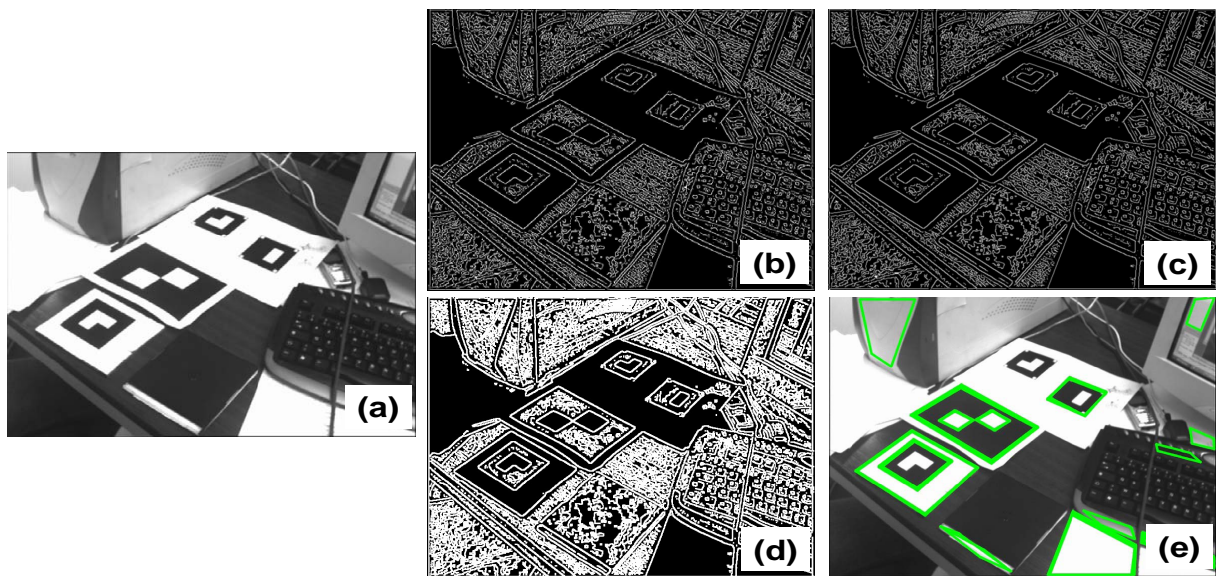


FIG. 2.9 – Processus de détection de cible. (a) Image originale. (b) Détection de contours. (c) Lissage de l'image. (d) Dilatation de l'image. (e) Approximation polygonale des contours.

En conclusion, seulement les objets avec quatre sommets et dont les angles sont pratiquement droits sont recherchés et considérés comme des carrés. Une fois les objets carrés détectés, la prochaine étape est de les identifier avec un modèle pré-défini pour ne retenir que les objets cibles.

Identification : Le but de cette étape est de concevoir un modèle d'objet qui pourra être extrait de manière robuste et en temps réel de la scène. Nous avons défini un modèle de cible avec un motif particulier représenté sur la figure 2.10. Le code interne de notre objet cible est calculé à partir d'un échantillon de points de l'objet cible 3D. On effectue une projection de ces échantillons sur le plan image 2D en utilisant la matrice de projection, H , qui a été calculée à partir des quatre sommets de la cibles dans l'image et leurs correspondants 3D du modèle de la cible, par la formule suivante :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}_H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (2.4)$$

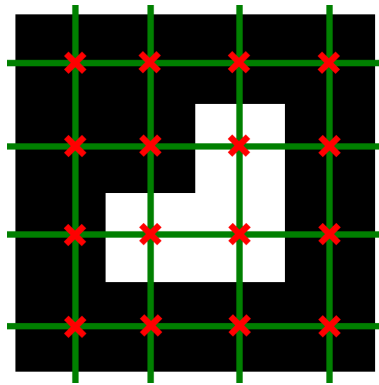


FIG. 2.10 – Échantillonnage de la cible codée.

À partir de l'équation précédente, nous pouvons écrire :

$$\begin{pmatrix} X & Y & 1 & 0 & 0 & 0 & -uX & -uY \\ 0 & 0 & 0 & X & Y & 1 & -vX & -vY \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} h_{33}u \\ h_{33}v \end{pmatrix} \quad (2.5)$$

Afin de ne pas obtenir une solution triviale, nous fixons $h_{33} = 1$. Donc, le nombre de paramètres à estimer est égal à 8. Le système à résoudre contient 8 équations. Nous utilisons alors les 4 points coplanaires qui représentent les sommets de notre objet cible dans l'image et leurs correspondants du modèle de la cible. La matrice de projection permet le passage des points du repère de la cible vers le repère de l'image. Nous définissons une grille sur notre cible, puis nous projetons les échantillons de points 3D sur l'image 2D en utilisant la matrice de projection, H .

Nous calculons le code correspondant aux échantillons de points de la grille du modèle projeté sur l'image 2D. Ce code est composé de 16 bits. Nous remarquons que notre modèle à quatre codes suivant les quatre possibilités d'orientation de notre carré. Par exemple pour la figure 2.11, notre cible a les quatre codes suivants : 608, 1120, 1568 ou 1600, selon son orientation dans l'image .

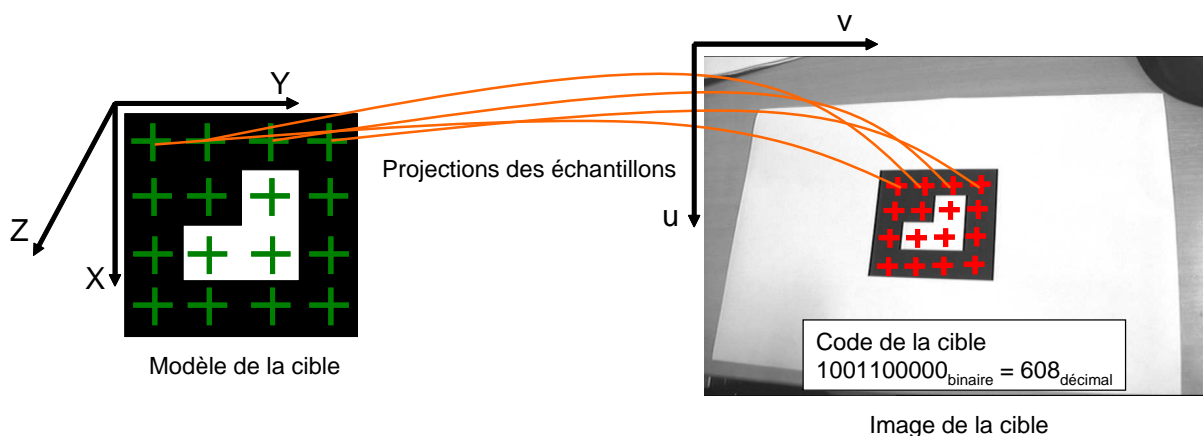


FIG. 2.11 – Identification de la cible.

Notre système de codage contient 4 bits utiles pouvant déterminer le code effectif de la cible. Les 8 autres bits servent à détecter le bord de l'objet. Au final, notre système donne 2^4 possibilités soit 16 marqueurs différents. Toutefois, le système de cibles doit respecter une contrainte qui est de permettre la détection de l'orientation de la cible, ce qui veut dire que chaque cible qui est tournée d'un quart de tour possède un code différent (figure 2.12). En conséquence, cela divise par 4 le nombre de classes d'équivalence que l'on peut attribuer par rapport au nombre de codes possibles. De plus, les cibles ne doivent pas présenter de symétrie centrale car nous ne pourrions pas distinguer de code plus petit qu'un autre et donc connaître l'orientation de la cible. Finalement, nous obtenons 4 classes d'équivalence de code et chaque cible a un code unique selon son orientation [2].

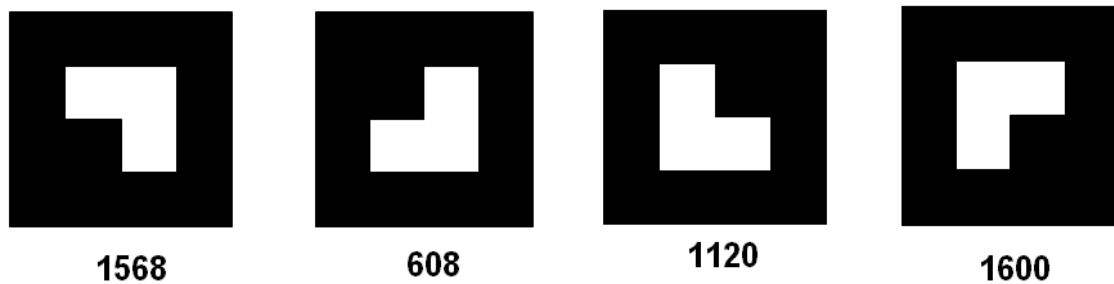


FIG. 2.12 – Codes correspondant aux différentes orientations de la cible.

Résultats expérimentaux

Temps d'exécution : Nous avons commencé l'évaluation de notre méthode de reconnaissance de cibles codées, par le calcul du temps d'exécution. Le temps nécessaire pour les différentes procédures de traitement de notre méthode de détection et d'identification de cibles, est estimé à $\approx 20ms$. Ce temps varie considérablement en fonction de la complexité de la scène, le nombre d'objets qui y sont présents et aussi de la configuration matérielle employée.

Changement d'éclairage : Nous avons testé notre algorithme d'identification dans le suivi de cibles, en variant la luminosité de l'environnement. Les résultats obtenus montrent que le suivi est maintenu quand les conditions d'éclairage sont normales (figure 2.13-b, c, d, e), mais lors d'une très forte augmentation ou baisse de lumière, l'objet cible n'est plus détecté (figure 2.13-a, f). En effet, dans des conditions extrêmes d'éclairage, les opérateurs de traitement d'images ne peuvent pas être appliqués convenablement car la lumière constitue une source d'information du signal analysé.

Rotation de la cible : On remarque que l'algorithme d'identification extrait la cible de l'image quelque soit son orientation, ceci est dû au fait que nous prenons en compte les 4 orientations de la cible dans l'algorithme d'identification. Le suivi de cible effectuant une rotation ne pose donc aucun problème particulier, et les résultats obtenus sont satisfaisants (figure 2.14).

Changement d'échelle : L'algorithme de reconnaissance a été aussi testé lors du changement d'échelle. On a varié le positionnement de la cible par rapport à la caméra, on remarque pour une distance allant jusqu'à $1.5m$ (figure 2.15-a, b, c), la cible est identifiée, cependant, au delà de cette distance, l'algorithme n'arrive plus à extraire la cible de l'image (figure 2.15-(d)). Le facteur d'échelle conditionne la visibilité des cibles, ceci permet donc, la détection ou non des objets d'intérêts dans l'image.

Inconvénients de la méthode : Les cibles présentent l'inconvénient de devoir être fixées sur les objets à suivre. De plus, le système présenté utilise des cibles planes, ce qui suppose que l'objet possède des surfaces planes sur lesquelles les cibles peuvent être collées. Enfin, si les cibles sont occultées, le système ne peut plus estimer la pose de la caméra. Ceci implique que les cibles doivent être placées de manière à ce qu'au moins une de celles-ci soit toujours visible par la caméra.

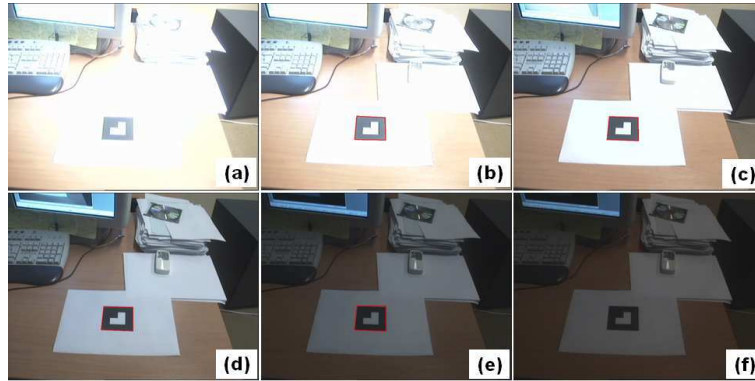


FIG. 2.13 – Identification de cible lors d'un changement d'intensité.

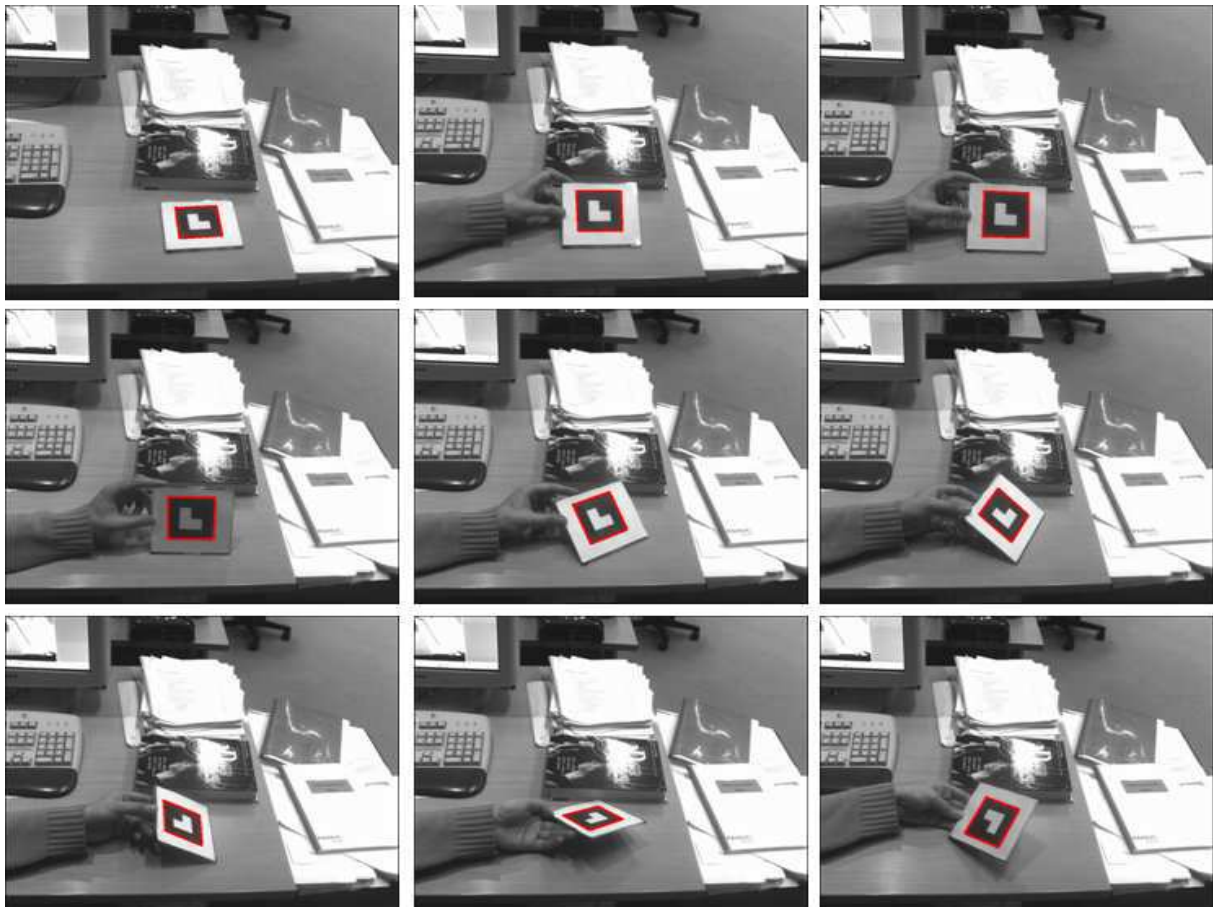


FIG. 2.14 – Identification de cible effectuant des rotations.

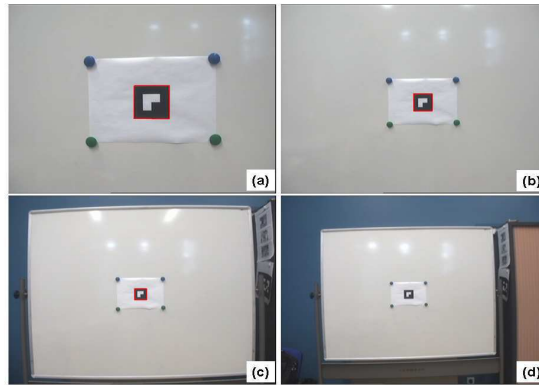


FIG. 2.15 – Identification de cible en effectuant un changement d'échelle. (a) Distance caméra-cible : $0.5m$. (b) Distance caméra-cible : $1m$. (c) Distance caméra-cible : $1.5m$. (d) Distance caméra-cible : $2m$.

2.3.2 Suivi sans-marqueur

Le suivi d'objets (ou tracking) constitue l'un des axes essentiels de recherche dans le domaine de la vision par ordinateur. Il s'agit d'un problème difficile mais occupe une place importante dans des tâches comme la mesure du mouvement ou l'asservissement visuel et ce pour des applications allant de l'imagerie médicale à la sécurité en passant par l'analyse du trafic routier.

Plusieurs méthodes de suivi sont développées dans la littérature. Il existe deux catégories d'algorithmes : ceux basés sur l'extraction de données 1D (points ou segments) de l'image et également les techniques basées sur des informations 2D comme les textures ou les contours.

Parmi les différentes méthodes de suivi existantes, il existe trois méthodes génériques couramment utilisées :

- Méthode de Gennery [55] [56] : cette méthode consiste à projeter le modèle de l'objet sur le plan image, ensuite mesurer l'erreur de projection entre le modèle et l'image réelle et enfin ajuster la fonction de la mesure. Il n'y a qu'une seule itération parce que la pose initiale est supposée correcte. Cette méthode offre de bons résultats à condition bien sûr que l'estimation de départ soit correcte.
- Méthode de Lowe [93] [94] : le principe de base de cette méthode est d'évaluer une fonction d'erreur de la pose entre la projection du modèle de la cible et l'image 2D. Cette fonction est exprimée par rapport aux paramètres de la pose. Le but est de minimiser cette fonction afin de déterminer les paramètres de la pose. L'algorithme converge assez rapidement, toutefois, si la caméra subit un mouvement brusque, les paramètres estimés sont entachés d'erreur.
- Méthode de Harris [63] : à partir d'une pose initiale, chaque point d'intérêt est projeté sur l'image 2D, le gradient de l'image est alors calculé autour de ce point mais dans une direction privilégiée qui est précisée dans le modèle. À chaque point d'intérêt est associé un point compagnon. Une fois projetés dans l'image 2D, les deux points forment une droite et la direction du gradient est choisie perpendiculairement à cette droite.

Toutes ces méthodes présentées permettent un tracking de cibles quand celles-ci sont visibles par la caméra. D'autres auteurs, tels que Naimark et Foxlin [52], se sont intéressés à l'aspect robustesse, en élaborant un système de tracking hybride composé d'une caméra et d'une centrale inertielle qui gère les occultations et qui fonctionne dans des conditions variées d'éclairage.

Nous présentons maintenant, une méthode de suivi que nous avons développée pour la détection et le tracking des cibles dans les séquences d'images. Cette méthode de tracking sans-marqueur, se base sur les contours actifs pour la détection d'objet et en s'appuyant sur le flot optique pour initialiser la région d'intérêt [99].

2.3.2.1 Contours actifs

La méthode des contours actifs ou snakes¹ a été proposée par Kass en 1987 [80] et a depuis bénéficié d'une certaine popularité au sein de la communauté de vision par ordinateur. Elle est utile dans un grand nombre de domaines, comme la visualisation médicale, l'extraction de forme, ou encore le suivi d'objets.

Définition : Un contour actif est défini comme un modèle élastique contrôlé par une contrainte de continuité, qui se déforme en minimisant un certain critère vers le contour de l'objet à segmenter. Il s'agit d'une méthode semi interactive : l'opérateur place a priori dans l'image au voisinage de la forme à détecter une forme initiale de contour qui sera amenée à se déformer par itérations successives pour converger vers le contour réel.

Le critère de déformation est une fonction d'énergie représentant des caractéristiques géométriques du contour et des caractéristiques de l'image. Pour repérer les variations sur l'image, la méthode est souvent basée sur le gradient, qui est plus élevé au niveau des contours. Le contour actif évolue au cours du temps de la façon suivante : dans un premier temps, l'énergie pour chaque point est calculée, puis les points sont classés par ordre d'énergie croissante, et le point d'énergie maximale est déplacé, puis on recalcule l'énergie pour ce point, et on réorganise la liste. Si la distance entre deux points est trop grande, on ajoute un point entre les deux. Cette évolution s'arrête par un critère d'arrêt qui correspond à une condition de stabilité.

Modélisation : Le snake est modélisé par une courbe paramétrée :

$$s \mapsto v(s) = (x(s), y(s)), s \in [a, b] \quad (2.6)$$

avec s abscisse curviligne (paramètre sur la courbe appartenant au domaine spatial $\Omega = [a, b]$) et $v(s)$ vecteur de position du point de contour de coordonnées $x(s)$ et $y(s)$. Cette courbe peut être fermée ou non.

Interprétation en terme de force et d'énergie

Approche physique : Le contour évolue suivant plusieurs conditions, comme la proximité des pixels présentant une grande variation. Ces conditions sont analogues à des conditions physiques, et sont rassemblées dans une fonction d'énergie que l'on cherche à minimiser. Kass [80] explique que le snake est amené à se déformer sous l'action de plusieurs forces :

1. une énergie propre, assimilée à l'énergie mécanique de tension et de torsion d'une ligne matérielle, due uniquement à la forme du contour,
2. une énergie potentielle externe imposée par l'image qui vise à plaquer la courbe sur les contours,
3. une énergie traduisant les contraintes spécifiques du problème jugées pertinentes pour augmenter la précision de la segmentation et introduites par l'utilisateur.

Fonction d'énergie du contour actif : L'énergie du contour s'écrit alors :

$$E_{totale} = \int_0^1 E_{snake}(v(s)) = \int_0^1 [E_{interne}(v(s)) + E_{image}(v(s)) + E_{externe}(v(s))] ds \quad (2.7)$$

¹Les expressions "contour actif" et "snake" sont synonymes et seront utilisées de façon interchangeables dans la suite de ce mémoire.

avec $E_{interne}$ l'énergie propre du contour, E_{image} son énergie potentielle due aux contraintes de l'image et $E_{externe}$ l'énergie des autres contraintes. L'énergie interne $E_{interne}$ représente les propriétés intrinsèques du contour, comme la longueur, la courbure, la distance entre les points. Son but est de maintenir le contour de telle sorte que chaque point du snake soit au milieu de ses deux voisins. $E_{interne}$ représente l'influence de l'élasticité et de la rigidité sur la forme du contour lors de sa déformation, on peut donc écrire :

$$E_{interne} = \alpha(s) E_{elastique} + \beta(s) E_{courbure} \quad (2.8)$$

où α est le facteur d'élasticité du contour, et β est le facteur de rigidité du contour. L'énergie élastique tend à rétrécir le snake tout en maintenant une répartition homogène des espaces entre les points de contrôle. Plus le paramètre d'élasticité est grand, et plus la tendance du contour à se contracter est grande (forme limite : le point). L'énergie de courbure ou rigidité réduit la possibilité d'une discontinuité locale en un noeud : elle tend à redresser le snake et à limiter la courbure (forme limite : le cercle). L'énergie interne s'écrit donc :

$$E_{interne} = \alpha E_{elastique} + \beta E_{courbure} = \alpha(s) \left(\frac{dv}{ds} \right)^2 + \beta(s) \left(\frac{d^2v}{ds^2} \right)^2 \quad (2.9)$$

L'énergie due à l'image E_{image} dépend des caractéristiques de l'image et agit de telle sorte qu'un point du snake se déplace vers un point de fort contour. Elle représente donc les éléments vers lesquels on veut attirer le snake, et contient un terme associé aux changements abruptes de l'image, c'est à dire les lignes de fort gradient. On choisit de représenter ces forces par une force relative au gradient de l'image :

$$E_{image} = -\lambda(s) |\nabla I(v(s))|^2 \quad (2.10)$$

Une autre approche, moins utilisée, consiste à définir l'énergie de l'image $E_{image} = -I$, afin d'attirer le snake avec les lignes claires de l'image. L'énergie due aux autres contraintes $E_{externe}$: elle est définie par l'utilisateur selon les spécificités du problème. On peut citer quelques exemples d'énergies de contraintes fréquemment utilisées, comme l'énergie correspondant aux zones brillantes et aux zones sombres ($E_{externe} = I$), ou l'énergie correspondant à la répulsion d'une zone donnée ($E_{externe}(x) = \frac{\lambda}{1+|x-\omega|^2}$). Kass propose également une énergie dont le but serait d'obliger certains points du snake à fixer leur position. Il imagine ainsi d'attacher certains points du snake par un ressort fictif à des points de l'image. La force de ce ressort s'écrit $k(x - x_0)$ produit l'énergie $kx^2/2$, qui peut être utilisée comme énergie de contraintes dans l'équation des contours actifs. Finalement, on obtient donc :

$$E_{totale}(v(s)) = - \int_a^b \left[-\lambda(s) |\nabla I(v(s))|^2 + \alpha(s) \left(\frac{dv}{ds} \right)^2 + \beta(s) \left(\frac{d^2v}{ds^2} \right)^2 + E_{externe}(v(s)) \right] ds \quad (2.11)$$

Mise en équation : Le problème est donc de minimiser la fonction d'énergie suivante :

$$E_{totale}(v(s)) = - \int_a^b [\lambda(s) |\nabla I(v(s))|^2 + E_{externe}(v(s))] ds + \int_a^b \alpha(s) v'(s)^2 ds + \int_a^b \beta(s) v''(s)^2 ds \quad (2.12)$$

Le contour actif est attiré par les zones de fort gradients qui correspondent aux contours de l'image, et dépend des contraintes définies par l'utilisateur, ce qui nous amène à poser $P = -\lambda(s) |\nabla I(v(s))|^2 + E_{externe}(v(s))$. Pour minimiser cette fonction, on utilise une approche variationnelle : si la courbe a

un minimum, ce minimum est atteint lorsque l'équation associée d'Euler-Lagrange est satisfaite. On obtient donc :

$$-(\alpha v')' + (\beta v'')'' + \nabla P(v) = 0 \quad (2.13)$$

soit :

$$-(\alpha v')' + (\beta v'')'' = F(v(s)) \quad (2.14)$$

avec $F(v) = -\nabla P(v)$ la somme des forces liées à l'image.

Résolution des équations d'énergie

Discrétisation du contour : On considère le snake comme étant un ensemble de points. Il faut donc discrétiser la courbe en un nombre n de points séparés par un pas h . Pour cela, on pose : $V^T = [v_0, v_1, \dots, v_{n-1}]^T$.

En posant V_i (avec $i \in [0, n-1]$) la solution approchée de la solution exacte $v(ih)$ au point ih , nous pouvons approcher l'équation 2.13 par :

$$\sum_{i=0}^{n-1} [-(\alpha v'(ih))' + \beta v''(ih)'' - F(v(ih))] = 0 \quad (2.15)$$

soit :

$$-(\alpha v_i')' + (\beta v_i'')'' = F(v_i) \quad (2.16)$$

Les dérivées sont approximées par des différences finies (méthode de Cohen) :

$$\begin{aligned} v_i' &= \frac{v_{i+1}v_i}{h} + O(h) \\ v_i'' &= \frac{2v_{i+1}v_i + v_{i-1}}{h^2} + O(h^2) \\ v_i^{(4)} &= \frac{4v_{i+2}v_{i+1} + 24v_i v_{i-1} + v_{i-2}}{h^4} + O(h^4) \end{aligned} \quad (2.17)$$

En remplaçant les dérivées par les différences finies :

$$-\frac{\alpha}{h^2} (2v_{i-1}v_i + v_{i+1}) + \frac{\beta}{h^4} (4v_{i-2}v_{i-1} + 24v_i v_{i+1} + v_{i+2}) = F(v_i) \quad (2.18)$$

$$av_{i-2} + bv_{i-1} + cv_i + bv_i + 1 + av_{i+2} = F(v_i) \quad (2.19)$$

avec :

$$a = \frac{\beta}{h^4}, \quad b = \frac{\alpha}{h^2} \frac{4\beta}{h^4}, \quad c = \frac{2\alpha}{h^2} + \frac{6\beta}{h^4} \quad (2.20)$$

Résolution : Les paramètres α , β et λ étant des constantes, on a donc :

$$-\alpha v_i'' + \beta v_i^{(4)} = F(v_i) \quad (2.21)$$

Ce qui peut s'écrire sous la forme d'une équation matricielle : $AV = F$ où :

$$A = \begin{pmatrix} c & b & a & 0 & \dots & \dots & 0 \\ b & c & b & a & 0 & & \vdots \\ a & b & c & b & a & \ddots & \vdots \\ 0 & a & b & c & b & \ddots & 0 \\ \vdots & 0 & a & b & c & \ddots & a \\ \vdots & & \ddots & \ddots & \ddots & \ddots & b \\ 0 & \dots & \dots & 0 & a & b & c \end{pmatrix} \quad (2.22)$$

On peut remarquer que cette équation prend en compte le fait que chaque point du snake est directement lié à ses 4 voisins.

Généralisation : En réalité, v dépend de s mais aussi de $t : v(s, t)$. La mise en équation donne alors :

$$\frac{\partial v(s, t)}{\partial t} \frac{\partial}{\partial s} \left(\alpha(s) \frac{\partial v(s, t)}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left(\beta(s) \frac{\partial^2 v(s, t)}{\partial s^2} \right) = F(v) \quad (2.23)$$

La résolution de cette équation se fait de la même façon, à l'aide de l'approximation par les différences finies. On obtient alors l'équation matricielle suivante :

$$(I + \tau A) V^T = V^{t-1} + \tau F(V^{t-1}) \quad (2.24)$$

Algorithme de minimisation - Algorithme de Kass : Le premier algorithme de minimisation de la fonction d'énergie des contours actifs a été proposé par Kass pour l'équation qu'il avait lui-même définie [80], et est basé sur des calculs variationnels. L'équation 2.13 est écrite sous la forme suivante :

$$\alpha v''(x) - \beta v^{(4)}(x) - \gamma \nabla E_{\text{externe}} = 0 \quad (2.25)$$

Une projection de l'équation 2.25 sur les axes x et y , donne :

$$\begin{aligned} \alpha x''(s) - \beta x^{(4)}(s) - \gamma \frac{\partial E}{\partial x} &= 0 \\ \alpha y''(s) - \beta y^{(4)}(s) - \gamma \frac{\partial E}{\partial y} &= 0 \end{aligned} \quad (2.26)$$

L'objectif de l'algorithme est de minimiser ces deux équations. Cette résolution est une des plus simples, et a une complexité en $O(n)$. La boucle principale de traitement de l'algorithme de Kass est illustrée sur la figure 2.16.

```

Tant que (nombre de point qui
          bougent > seuil)
  pour chaque point (xi, yi)
    • calcul de l'énergie locale de la courbe au
      point (xi, yi)
    • estimation des dérivées partielles x'', x(4),
      y'', y(4)
    • calcul des forces [αv''(s)] et [γ(dEext/ds)]
      selon les directions x puis y
    • calcul de l'énergie totale dans chaque
      direction pour trouver le
      nouveau pixel (xnv, ynv)
    • calcul de l'énergie locale au point
      (xnv, ynv)
    • si (energie(xnv, ynv) < energie(xi, yi)) mise à
      jour de (xi, yi) par (xnv, ynv)
  fin_pour
fin_tant_que

```

FIG. 2.16 – Algorithme de Kass.

2.3.2.2 Suivi d'objets avec les contours actifs

Le suivi d'objet permet d'étendre sur plusieurs images la segmentation spatiale définie précédemment. Il s'agit de suivre une zone d'intérêt au cours d'une séquence d'images. Cette tâche, appelée également segmentation temporelle ou tracking, est devenue une des plus utilisées parmi celles développées pour la vision par ordinateur.

Approche classique de suivi : La segmentation vidéo se fait en deux grandes étapes :

- la segmentation spatiale de l'image initiale,
- le suivi de cette segmentation sur les images suivantes.

Plusieurs approches sont alors possibles pour la seconde étape : soit on recommence la segmentation sur l'image suivante, puis on essaye de mettre en correspondance le résultat obtenu avec le résultat précédent à l'aide de critère de recouvrement, soit on utilise le résultat précédent comme initialisation de la nouvelle recherche. La seconde approche nécessite que la nouvelle segmentation soit peu éloignée de la précédente, mais permet une plus grande rapidité et stabilité.

Suivi d'objets et contours actifs : Le suivi d'un objet doit être robuste, c'est à dire qu'il doit permettre de suivre un objet même si celui-ci change de position ou est partiellement occulté. La méthode de suivi à partir des contours actifs offre justement l'intérêt de ne pas utiliser de représentation explicite de la forme que doit avoir le contour. Au contraire, le contour évolue jusqu'à ce qu'il converge (figure 2.17). Elle semble donc adaptée à la résolution du problème du tracking. L'état initial de la cible est défini sur la première image de la séquence, une fois que la minimisation de l'énergie totale du snake est atteinte. Le suivi consistera ensuite à trouver pour chaque instant t de la séquence d'images la position du snake qui minimise son énergie.

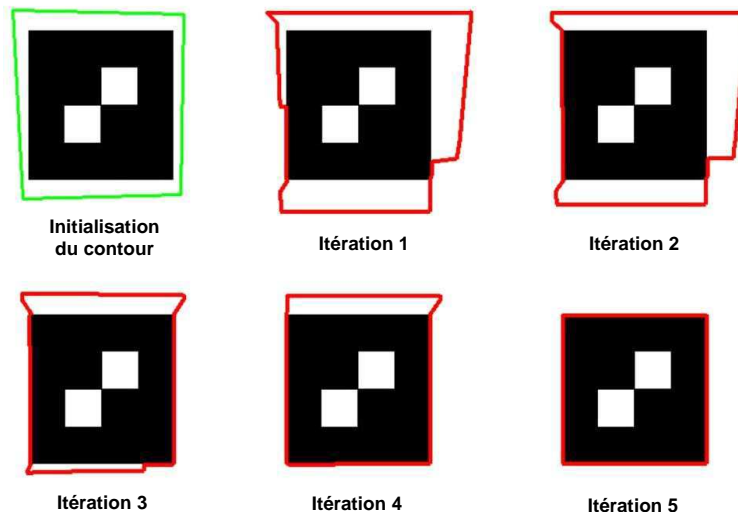


FIG. 2.17 – Évolution du snake pour la détection d'objets.

2.3.2.3 Amélioration du suivi par le flot optique

Le contour actif réalise uniquement la localisation spatiale de l'objet d'intérêt. Cette procédure de détection est divisée en deux étapes : initialisation et localisation. L'inconvénient de la méthode du contour actif est l'initialisation : l'utilisateur garde le contrôle sur l'application et doit placer le contour initial autour de l'objet cible pour le détecter. Dans une application de suivi, l'objet ou la caméra est en mouvement et ceci mettra en échec le tracking basé sur le snake car le contour initial reste figé à son emplacement initial dans la première image de la séquence. Nous utilisons le flot optique afin d'initialiser le contour initial du snake à chaque instant, par la suite le contour actif effectue la localisation spatiale. Le schéma de la figure 2.18 montre le principe du suivi de la méthode du snake et le flot optique. Au début, l'utilisateur initialise le contour autour de l'objet d'intérêt, dans la suite de la séquence d'images, le flot optique calcule les nouvelles positions de ce contour ce qui permet d'automatiser le processus d'initialisation du contour actif. Enfin, le snake est appliqué pour localiser le contour réel de la cible et les points d'intérêts 2D sont déterminés.

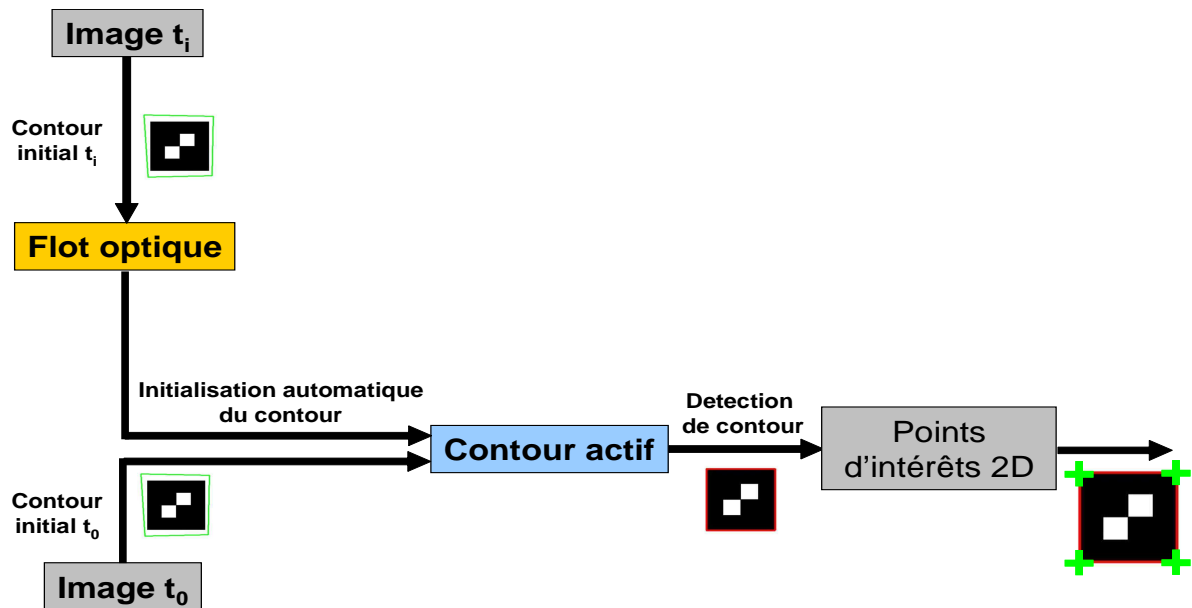


FIG. 2.18 – Suivi spatio-temporel avec le snake et le flot optique.

Définition du flot optique : Le flot optique est un champ de déplacement visuel qui permet de décrire les évolutions d'une image à l'autre dans une séquence vidéo. Cet outil est couramment utilisé, par exemple dans des applications comme la reconstruction 3D, l'interpolation temporelle de séquences d'images en tenant compte du mouvement, ou dans le domaine qui nous intéresse, la segmentation temporelle et le suivi d'objets. Il s'agit de considérer les variations de l'intensité dans la vidéo en terme de déplacement de points d'images. Pour calculer le flot optique, on se base sur la règle de continuité suivante : l'intensité d'un point particulier d'un objet sur une vidéo est constante au cours du temps.

Mise en équation : On pose $I(x, y, t)$ l'intensité de l'image au point (x, y) à l'instant t . La règle de continuité peut alors se traduire mathématiquement par :

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (2.27)$$

En appliquant la différentiation par chaîne, on peut écrire :

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2.28)$$

On note le flot optique par :

$$v_x = \frac{dx}{dt}, v_y = \frac{dy}{dt} \quad (2.29)$$

et le gradient de l'image :

$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t} \quad (2.30)$$

d'où :

$$I_x v_x + I_y v_y = -I_t \quad (2.31)$$

Cette dernière équation est usuellement appelée "Équation de Contrainte du Flot Optique". Le vecteur $V = (v_x, v_y)^T$ correspondant au vecteur vitesse est défini comme étant le flot optique à déterminer. Nous sommes confrontés ici au problème d'ouverture, puisqu'il faut calculer deux inconnues à partir d'une seule équation. Pour le résoudre, il faut donc poser des contraintes supplémentaires sur le flot optique. Plusieurs méthodes ont été proposées dans cette optique, nous avons choisi d'utiliser celle formulée par Lucas et Kanade [96].

Algorithme de Lucas et Kanade : Lucas et Kanade [96] ont ajouté une contrainte locale à l'équation du flot optique : ils considèrent, en effet, que dans le voisinage du point considéré, en supposant que tous les points du voisinage ont la même vitesse, l'intensité est également constante.

La méthode consiste à diviser l'image originale en petits intervalles : $\Omega_1 \cup \Omega_2 \dots = \Omega$, on considère que la vitesse est constante dans chaque intervalle et on minimise l'erreur via une expression de moindres carrés pondérés :

$$\sum_{x \in \Omega} C^2(x) [\nabla I(x, t) \cdot v_y + I_t(x, t)]^2 \quad (2.32)$$

La solution est de la forme :

$$a^T C^2 a v_y = a^T C^2 b \quad (2.33)$$

où, pour n points $x_i \in \Omega$ et pour un temps t :

$$\begin{cases} a = [\nabla I(x_1), \dots, \nabla I(x_n)]^T \\ C = \text{diag}[C(x_1), \dots, C(x_n)] \\ b = -[I_t(x_1), \dots, I_t(x_n)]^T \end{cases} \quad (2.34)$$

La solution est donnée par :

$$v_y = [a^T C^2 a]^{-1} a^T C^2 b \quad (2.35)$$

à partir des équations 2.35 et 2.31, le vecteur V du flot optique est déterminé.

La figure 2.19 montre le fonctionnement de la méthode des contours actifs et le flot optique pour la détection d'objet dans une séquence d'image. Le flot optique permet le suivi du contour initial de l'objet d'intérêt et le contour actif réalise la localisation spatiale de l'objet dans chaque image.

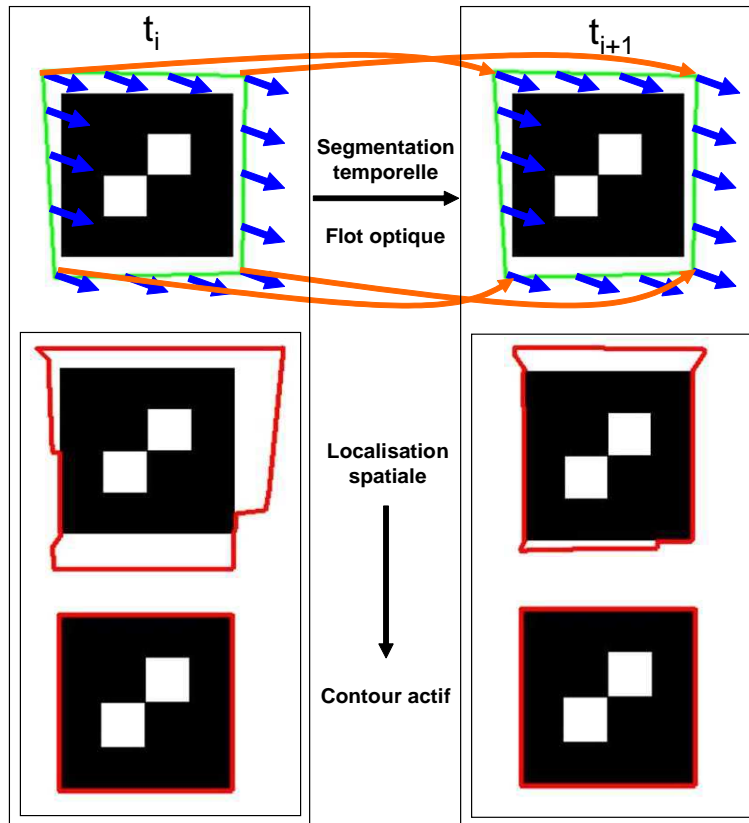


FIG. 2.19 – Suivi de cible par les contours actifs et le flot optique.

2.3.2.4 Résultats et analyse de performances

Afin de pouvoir vérifier les performances de l'algorithme du contour actif, et soulever les avantages et les inconvénients de l'utilisation de la méthode des contours actifs pour le suivi vidéo, nous avons réalisé une série de tests, séparés en deux catégories principales : d'abord les tests concernant des changements dus au mouvement de la cible elle-même, puis ceux considérant l'environnement du suivi.

Mouvement de translation : Le mouvement de translation est un mouvement simple, que nous avons utilisé pour vérifier la pertinence de l'algorithme des contours actifs. Lorsque cette translation a lieu à une vitesse raisonnable, le suivi est réussi (figure 2.20). Cependant, lors de translations plus rapides, le contour est parfois attiré par des points extérieurs à la zone de recherche, comme les bords de l'image.

Rotation de la cible : Le snake ne prend pas en compte l'orientation de la cible, le suivi d'un objet effectuant une rotation ne pose donc aucun problème particulier, et les résultats obtenus sont acceptables (figure 2.21).

Changement d'échelle : Ce test a pour but de vérifier la capacité du snake à détecter un objet soumis à un changement d'échelle, se rapprochant ou s'éloignant. N'étant pas basé sur une forme a priori de l'objet à détecter, le snake ne rencontre pas de difficultés notables lors de ce test (figure 2.22).

Occultation partielle : En ce qui concerne les occultations, tant qu'elles ne sont pas trop importantes, les résultats sont satisfaisants (figure 2.23-a). Par contre, si cette occultation change la nature de la cible, le snake perd une partie de l'objet (figure 2.23-b).

Ainsi, sur le deuxième test, la cible est un carré noir contenant deux petits carrés blanc en son milieu, et lorsqu'elle sort partiellement de l'image, ce carré blanc est considéré par le snake comme étant le contour à suivre. Même si par la suite l'objet réapparaît dans son entier, le snake garde un appui sur le contour du milieu et n'en sortira pas (figure 2.23-c).

Changements d'intensité lumineuse : Nous avons testé la détection dans le cas des changements de luminosité. Le snake arrive à détecter l'objet en un nombre réduit d'itérations quand les conditions de luminosité sont normales (figure 2.24). En effet, quand l'intensité diminue progressivement, la détection et le suivi perdent de leurs précisions et le snake converge en un nombre plus élevé d'itération (figure 2.25). Enfin, lors d'une baisse plus forte de lumière, l'objet n'est plus détecté et le snake a un comportement aléatoire (figure 2.26). Il s'avère que l'algorithme n'est pas robuste aux changements trop grands au niveau de l'intensité lumineuse de l'image.

2.3.2.5 Analyse et critique du suivi par contours actifs

Le travail effectué montre qu'on peut obtenir des résultats satisfaisants avec la méthode des contours actifs appliquée à une image fixe. Sur une vidéo par contre, l'efficacité de la méthode est plus discutable. Les contours s'écartent facilement de la cible, et restent parfois bloqués dans des zones bruitées. Cependant, les tests effectués se sont révélés assez positifs lorsqu'un suivi par flot optique est appliqué en renfort des contours actifs. Ces tests mettent ainsi en avant les avantages des snakes sur d'autres méthodes antérieures, à savoir qu'ils gèrent tout à fait convenablement des caractères liés à la cible à suivre. Ils sont capables de détecter un objet a priori inconnu, et de réaliser le suivi d'objets déformables, puisqu'ils n'intègrent aucun modèle prédéfini, ce qui constitue sans doute leur plus grande réussite. Les occultations dans une mesure raisonnable et les petites variations de luminosité

sur la cible ne sont pas non plus des freins au suivi, les snakes pouvant réaliser un suivi acceptable malgré ces contraintes. Néanmoins, il s'agit d'une technique exigeante du point de vue des situations où elle est applicable. Elle nécessite en effet un bon contraste, et ne résiste pas à de forts changements d'intensité lumineuse. En outre, l'inconvénient le plus gênant à notre avis est cette dépendance au niveau de l'initialisation.

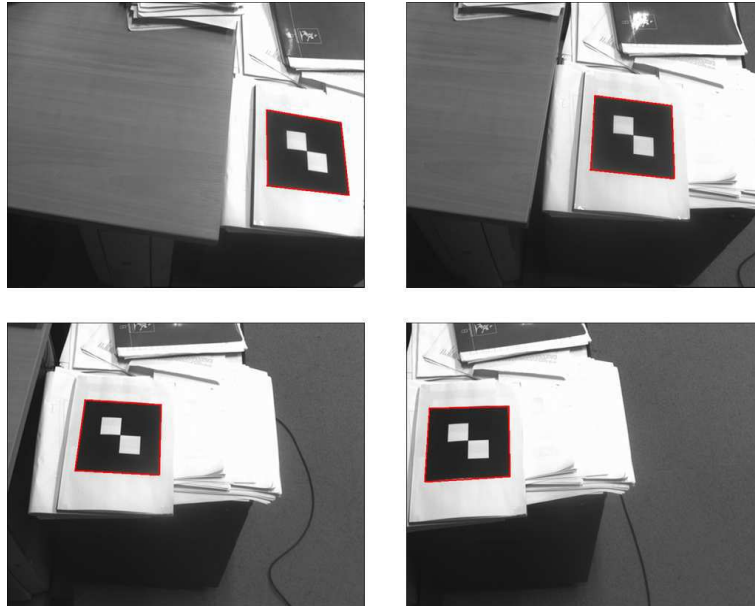


FIG. 2.20 – Suivi d'objet en effectuant une translation.

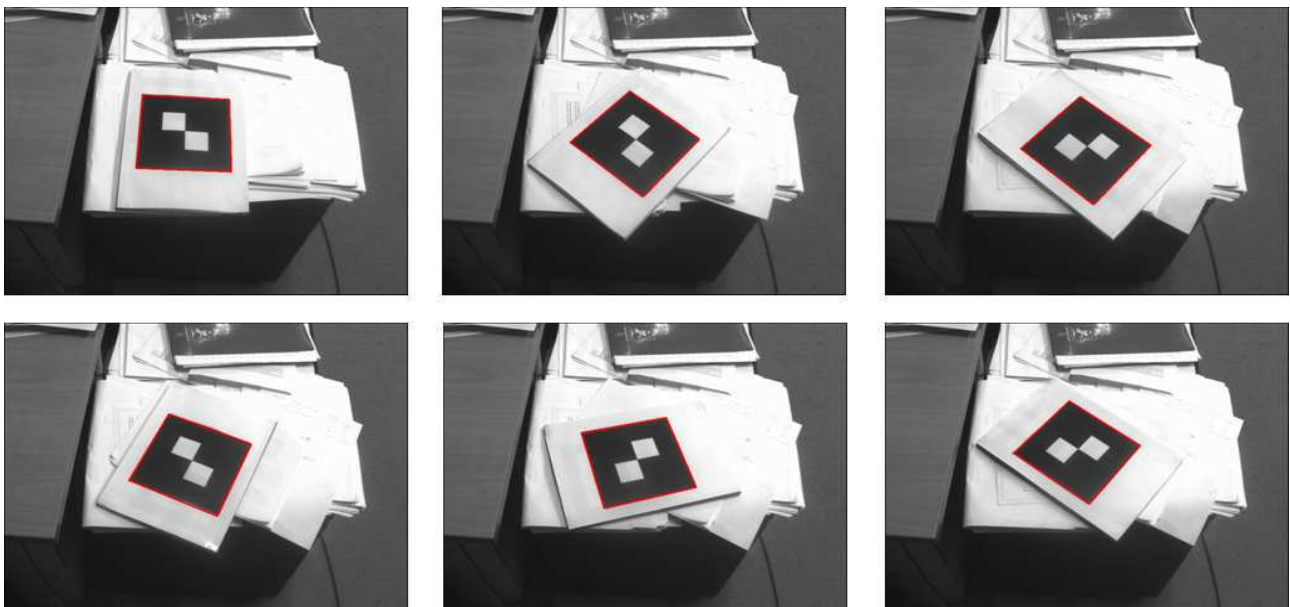


FIG. 2.21 – Suivi d'objet en effectuant une rotation.

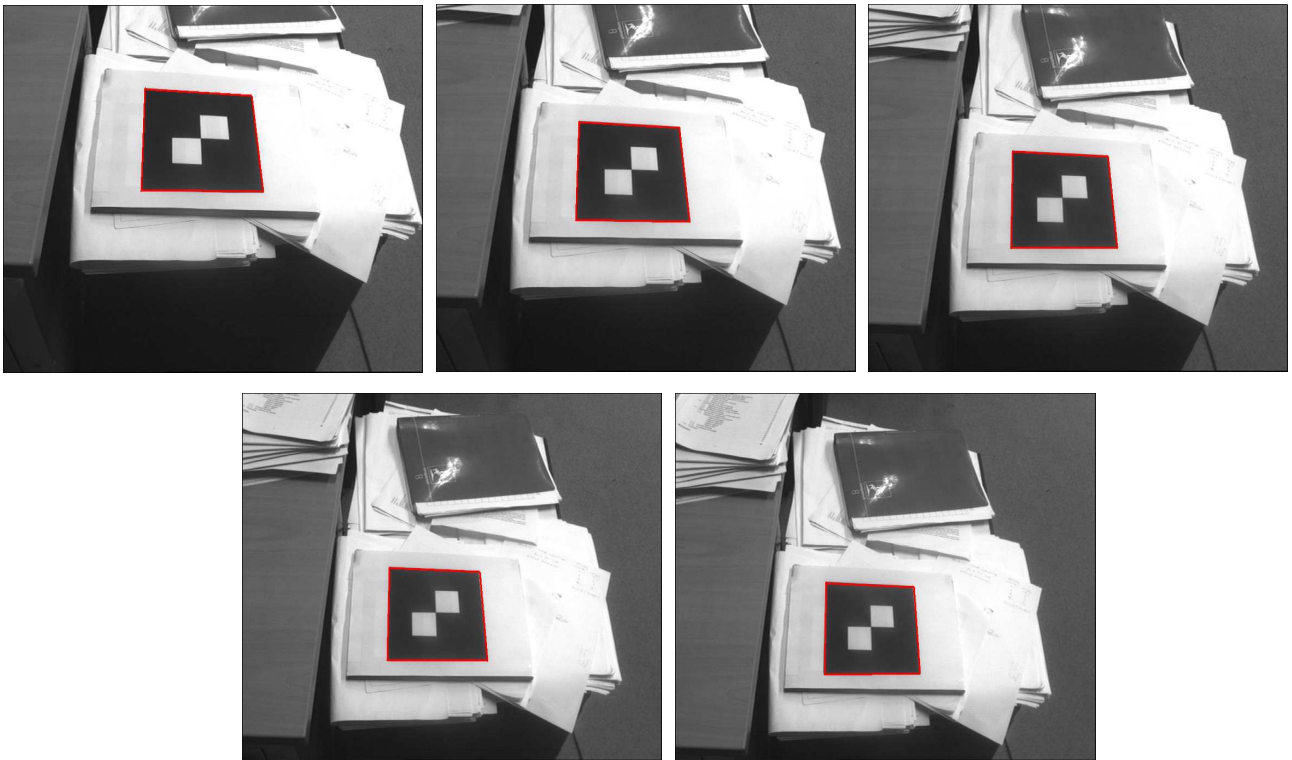


FIG. 2.22 – Détection d'objet lors d'un changement d'échelle.



FIG. 2.23 – Détection d'objet subissant une occultation partielle.

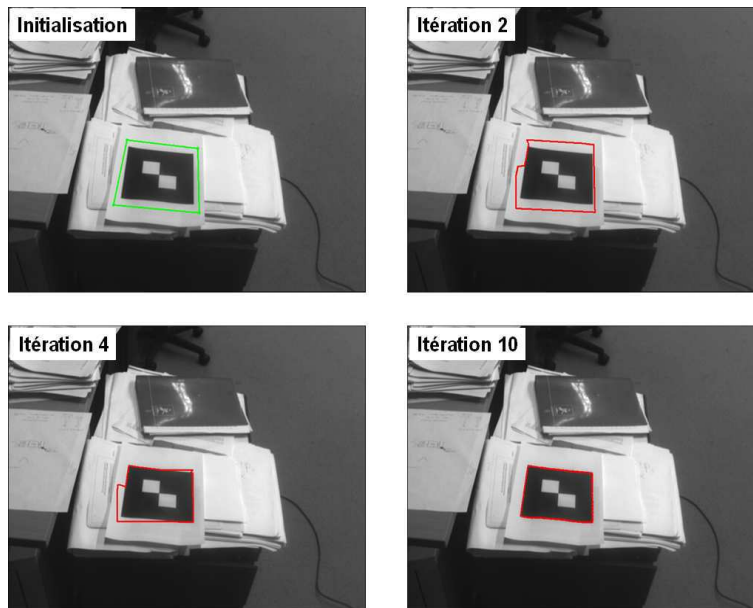


FIG. 2.24 – Détection d'objet dans des conditions normales de luminosité.

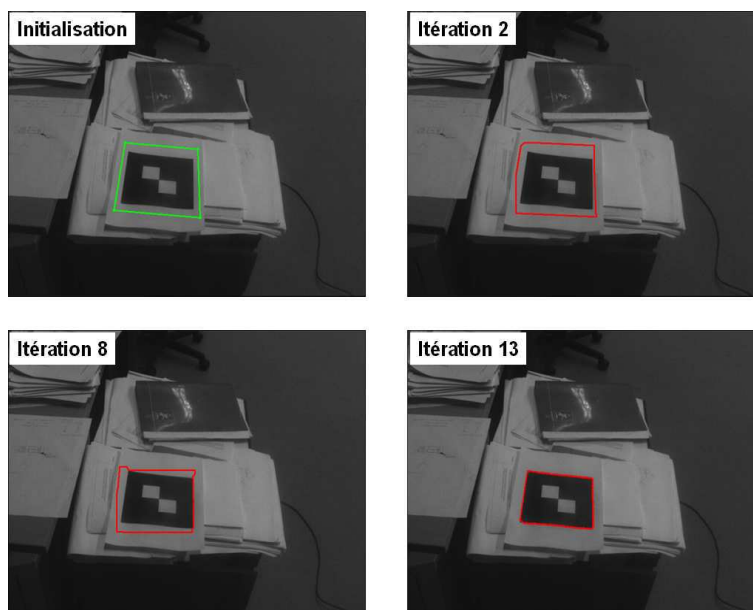


FIG. 2.25 – Détection d'objet lors de modifications de l'intensité lumineuse.

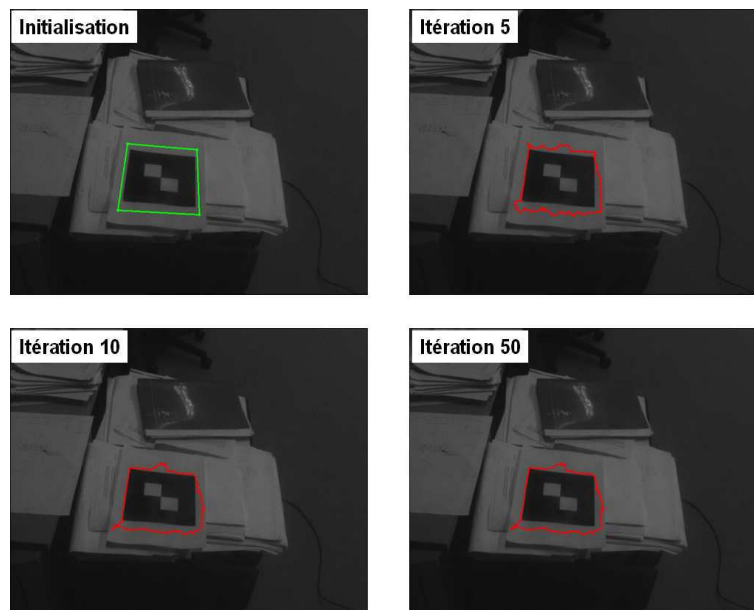


FIG. 2.26 – Détection d’objet dans des conditions de luminosités dégradées.

2.3.3 Bilan sur les méthodes de reconnaissance de cible

Nous avons présenté précédemment deux méthodes d’identification de cibles carrées pour un système de réalité augmentée. La première méthode basée sur les cibles codées réalise un échantillonnage spatial de l’image pour rechercher les primitives de forme quadrilatérale et calcule ensuite le code binaire de l’objet trouvé. Cette méthode de localisation spatio-temporelle est rapide et efficace mais son inconvénient est l’obligation de fixer les amers visuels sur l’objet à suivre.

La deuxième méthode développée, repose sur une optimisation d’un critère pour tendre vers le contour d’objet d’intérêt. Cette technique sans-marqueur n’exige aucune information sur la cible, les objets suivis sont de formes libres. Toutefois, le défaut majeur de cette méthode est l’initialisation du contour. En effet, l’utilisateur doit garder le contrôle sur l’application pour assurer la détection et la reconnaissance de la cible. De plus, la méthode du contour actif réalise uniquement la localisation spatiale de l’objet d’intérêt. C’est la raison pour laquelle le flot optique a été employé pour faire une segmentation temporelle dans les images.

D’après l’étude effectuée, la méthode des cibles codées s’est révélée plus performante et plus robuste que la méthode sans-marqueur. Dans la suite de notre étude, cette méthode sera utilisée pour la reconnaissance des cibles

2.4 Estimation de la pose

Le calcul de la pose s’appuie sur l’extraction des primitives géométriques pour permettre l’appariement des points 2D avec les points 3D de l’objet exprimés dans leurs repères associés.

Comme nous l’avons déjà évoqué, il existe deux types d’algorithmes d’estimation de la pose de la caméra qui sont distingués en deux grandes familles :

- Les méthodes analytiques (méthodes directes).
- Les méthodes numériques (méthodes itératives).

Nous allons présenter maintenant l’ensemble des méthodes proposées dans la littérature et qui sont dédiées aux applications temps réel. Par la suite, nous décrivons notre méthode de localisation de la caméra qui emploie un algorithme hybride d’estimation de pose. Cet algorithme combine à la fois une méthode analytique et une autre itérative pour améliorer la précision et la rapidité du recalage.

La méthode analytique est représentée par l'algorithme direct de calcul de pose de Didier [35] et la méthode numérique est fondée sur le filtre de Kalman étendu [161]. Des expérimentations seront réalisées pour évaluer les performances de notre méthode hybride et les autres techniques utilisées dans le cadre d'une étude comparative.

2.4.1 Algorithme analytique

Les méthodes analytiques emploient un nombre réduit de points. La complexité dans ces méthodes et le temps d'exécution sont faibles. Les paramètres de pose sont précis, cependant, la profondeur n'est pas bien estimée. Plusieurs algorithmes analytiques ont été présentés dans la littérature, ces méthodes diffèrent essentiellement dans la technique de résolution et le nombre de points utilisés. Didier [35] a développé un nouvel algorithme analytique basé sur les cibles carrées. La position et l'orientation sont calculées quand le code de la cible est correct. La méthode exige la connaissance de :

- Paramètres intrinsèques de la caméra.
- Coordonnées des quatre coins de la cible dans l'image.
- Mesure d'un côté de la cible carrée.

L'algorithme est composé de deux parties. La première partie consiste à calculer la profondeur réelle des coins de la cible et la deuxième partie calcule la pose. La cible a une forme carrée, ainsi nous avons la propriété suivante :

$$\overrightarrow{AB} = \overrightarrow{DC}$$

En appliquant le modèle perspectif de la caméra, nous obtenons l'équation suivante :

$$\begin{pmatrix} u_B & -u_C & u_D \\ v_B & -v_C & v_D \\ -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Z_B \\ Z_C \\ Z_D \end{pmatrix} = \begin{pmatrix} v_A \\ u_A \\ -1 \end{pmatrix} \quad (2.36)$$

En résolvant l'équation 2.36, la profondeur des coins de la cible sont données par les formules suivantes :

$$\begin{aligned} Z_B &= \frac{1}{\det M} [u_A (v_C - v_D) + v_A (u_D - u_C) + (u_C v_D - u_D v_C)] \\ Z_C &= \frac{1}{\det M} [u_A (v_B - v_D) + v_A (u_D - u_B) + (u_D v_B - u_B v_D)] \\ Z_D &= \frac{1}{\det M} [u_A (v_B - v_C) + v_A (u_C - u_B) + (u_B v_C - u_C v_B)] \\ \det M &= (u_C v_D - u_D v_C) + (u_D v_B - u_B v_D) + (u_B v_C - u_C v_B) \end{aligned} \quad (2.37)$$

Une fois, la profondeur réelle est déterminée, on calcule la translation et la rotation de la cible par rapport à la caméra. La translation est déterminée en utilisant le centre de la cible calculé à partir de ses sommets A, B, C, D . La matrice de rotation est donnée par les trois vecteurs suivants :

$$r_1 = \frac{\overrightarrow{AB} + \overrightarrow{DB}}{\|\overrightarrow{AB} + \overrightarrow{DB}\|} \quad r_2 = \frac{\overrightarrow{AC} - \overrightarrow{DB}}{\|\overrightarrow{AC} - \overrightarrow{DB}\|} \quad r_3 = r_1 \times r_2 \quad (2.38)$$

2.4.2 Itération orthogonale

Dans cette méthode, l'estimation de pose est formulée comme une minimisation d'erreur métrique basée sur la colinéarité dans l'espace d'objet. En utilisant l'erreur de colinéarité de l'espace objet (figure 2.27), un algorithme itératif est dérivé pour calculer les matrices orthogonales de rotation [95]. La projection et la mise en correspondance des points de référence 3D et les points 2D de l'image est formalisé comme suit : étant donné un ensemble de points 3D non-colinéaires $P_i = (X_i, Y_i, Z_i)^T$ où $i = 1 \dots n, n \geq 3$ exprimés dans un repère objet centré, l'espace des coordonnées de la caméra est donné par $q_i = (X'_i, Y'_i, Z'_i)^T$ qui sont reliés par une transformation rigide : $q_i = RP_i + T$ où R et T sont respectivement la matrice de rotation et le vecteur de translation. Les points de référence p_i sont projetés sur le plan image. Soit $p_i = (u_i, v_i)^T$ la projection de P_i sur le plan de l'image normalisée.

Sous le modèle idéal de sténopé, p_i , q_i et le centre de la projection sont alignés. Ceci est exprimé par l'équation suivante :

$$u_i = \frac{r_1^T P_i + t_x}{r_3^T P_i + t_z} \quad (2.39)$$

$$v_i = \frac{r_2^T P_i + t_y}{r_3^T P_i + t_z} \quad (2.40)$$

et :

$$p_i = \frac{1}{r_3^T P_i + t_z} (R P_i + T) \quad (2.41)$$

où : $R = (r_1, r_2, r_3)$ et $T = (t_x, t_y, t_z)$.

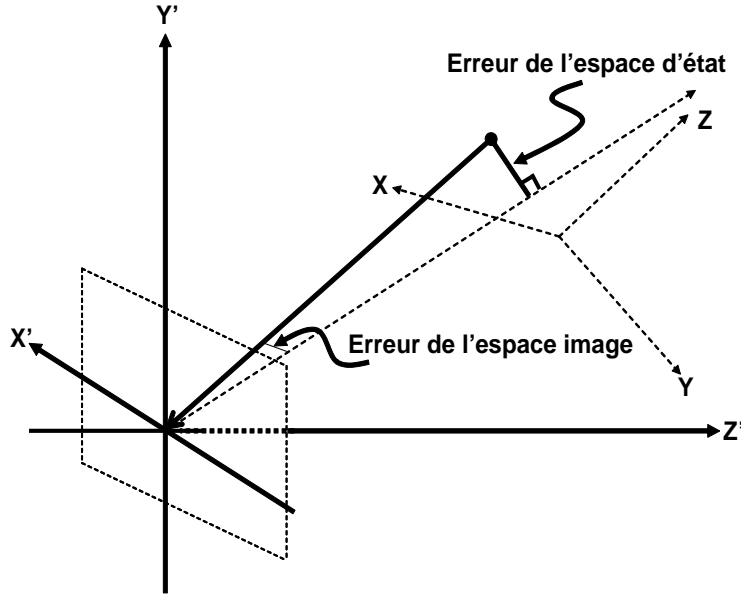


FIG. 2.27 – Erreur de colinéarité de l'espace d'objet et l'espace d'image.

L'algorithme d'itération orthogonale permet de déterminer dynamiquement les paramètres externes de la caméra en utilisant les mises en correspondances 2D-3D établies par l'algorithme de suivi de cible 2D à partir de l'image courante [3]. L'algorithme d'itération orthogonale calcule d'abord le vecteur d'erreur de colinéarité de l'espace d'objet [95] :

$$e_i = (I - \hat{V}_i) (R P_i + T) \quad (2.42)$$

où \hat{V}_i est la ligne de vue observée à travers la matrice de projection définie par :

$$\hat{V}_i = \frac{\hat{p}_i \hat{p}_i^T}{\hat{p}_i^T \hat{p}_i} \quad (2.43)$$

où \hat{p}_i est la projection de P_i dans l'image. Ensuite, une minimisation de l'erreur quadratique est effectuée :

$$E(R, T) = \sum_{i=0}^n \|e_i\|^2 = \sum_{i=0}^n \left\| (I - \hat{V}_i) (R P_i + T) \right\|^2 \quad (2.44)$$

L'algorithme d'itération orthogonale converge à un optimum pour un quelconque ensemble de points observés et n'importe quel point de départ [3]. Cependant, afin d'assurer la convergence de l'algorithme de calcul de la pose en un temps minimal, une bonne initialisation de paramètres de pose est requise. Par conséquent, l'algorithme analytique d'estimation de pose présenté dans la section 2.4.1 est utilisé pour calculer les paramètres initiaux de la pose et permettre à l'algorithme d'itération orthogonale de raffiner ces paramètres par la suite (figure 2.28).

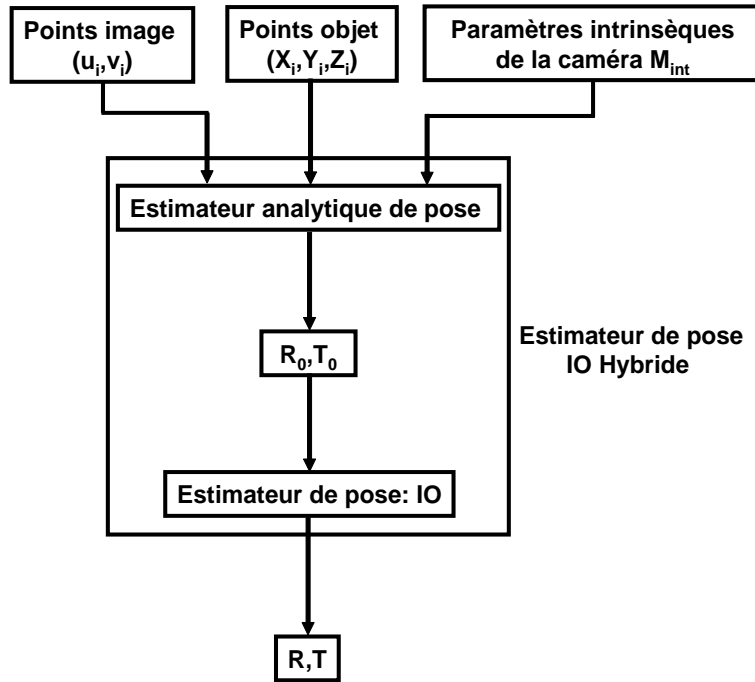


FIG. 2.28 – Diagramme d'estimation de la pose avec l'itération orthogonale hybride à l'instant t .

2.4.3 Filtre de Kalman étendu

Nous avons utilisé le filtre de Kalman étendu pour estimer la position et l'orientation du repère lié à l'objet par rapport au repère de la caméra. Le filtre de Kalman est un estimateur récursif optimal (annexe B). Le filtre de Kalman est un ensemble d'équations mathématiques permettant l'estimation des états d'un processus en minimisant l'erreur quadratique [161]. Le filtre de Kalman étendu est appliqué aux systèmes non-linéaires où le bruit est blanc et gaussien. Le modèle de processus est donné par l'équation suivante :

$$x_k = f(x_{k-1}, w_{k-1}) \quad (2.45)$$

où x_k est le vecteur d'état et w_{k-1} et le bruit de processus. L'équation de mesure est donnée par :

$$z_k = h(x_k, \nu_k) \quad (2.46)$$

où z_k est le vecteur de mesure et ν_k le bruit de mesure.

Dans la première étape du filtre de Kalman étendu, qui est l'étape de prédiction, le vecteur d'état et la matrice de covariance de l'erreur sont prédites en utilisant les estimations de \hat{x}_k et P_k . Une fois cette étape est finie, ces estimations deviennent les entrées de l'étape de correction. Avec les états mis à jour, la nouvelle matrice de covariance d'erreur est estimée. Ces deux étapes sont réalisées récursivement et enfin le vecteur d'état est estimé.

Comme nous l'avons décrit, l'étape de prédiction projette le vecteur d'état \hat{x} et la matrice de covariance d'erreur à partir de l'état courant k vers l'état suivant $k+1$. Le modèle de mesure représente la relation entre le vecteur d'état du système et les mesures de la caméra.

Nous devons tout d'abord définir le vecteur d'état du filtre de Kalman étendu. Puisque nous voulons estimer la pose de la caméra, nous utilisons les angles de rotation et les composantes de translation $(\phi, \theta, \psi, t_X, t_Y, t_Z)$ pour représenter l'état du système. Les mesures sont fournies par la caméra. Nous devons estimer les six variables du vecteur d'état, les mesures sont regroupées dans un vecteur 8×1 :

$$z = \left(u_1 \ u_2 \ u_3 \ u_4 \ v_1 \ v_2 \ v_3 \ v_4 \right)^T \quad (2.47)$$

où :

$$u_i = \frac{M_1 P_i + M_{14}}{M_3 P_i + M_{34}} \quad (2.48)$$

$$v_i = \frac{M_2 P_i + M_{24}}{M_3 P_i + M_{34}} \quad (2.49)$$

P_i représente le point 3D dans le repère objet et $M_i = (M_{i1}, M_{i2}, M_{i3})$.

M est la matrice de projection perspective de la caméra, donnée par :

$$M = \begin{pmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_x + u_0 t_z \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \quad (2.50)$$

2.4.3.1 Prédiction

La prédiction permet l'estimation du vecteur d'état, \hat{x} , et la matrice de covariance de l'erreur, P . Les équations de projections sont données par :

$$\hat{x}_k^- = \hat{x}_{k-1} \quad (2.51)$$

$$P_k^- = A P_{k-1} A^T + Q_{k-1} \quad (2.52)$$

où Q représente la matrice de covariance du bruit de processus et A est la matrice de transition exprimée par une matrice identité de dimension 6×6 :

$$A = I_6 \quad (2.53)$$

2.4.3.2 Correction

Le modèle de mesure relie le vecteur d'état au vecteur de mesures. Le vecteur de mesure est représenté par les points caractéristiques 2D de la cible dans l'image. En utilisant les positions de ces points dans le repère caméra et le modèle perspective de projection, on obtient les relations suivantes :

$$u_i = f(M, P_i) \quad (2.54)$$

$$v_i = f(M, P_i) \quad (2.55)$$

La fonction de mesure est donnée par :

$$z_k = h(\hat{x}_k) \quad (2.56)$$

h est donné par :

$$h(\hat{x}_k) = M P x_k + \nu_k \quad (2.57)$$

et x_k est le vecteur d'état défini auparavant.

Pour effectuer la mise à jour des mesures, on calcule tout d'abord le gain de Kalman :

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (2.58)$$

où :

$$\begin{cases} H_{ij} = \frac{\partial h_i}{\partial x_j}(\tilde{x}_k, 0) \\ V_{ij} = \frac{\partial h_i}{\partial \nu_j}(\tilde{x}_k, 0) \end{cases} \quad (2.59)$$

L'estimation est mise à jour avec la mesure z_k :

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (2.60)$$

Enfin, la covariance de l'erreur est mise à jour par :

$$P_k = (I - K_k H_k) P_k^- \quad (2.61)$$

où :

$$H = \begin{pmatrix} \frac{\partial h_1}{\partial \phi} & \frac{\partial h_1}{\partial \theta} & \frac{\partial h_1}{\partial \psi} & \frac{\partial h_1}{\partial t_X} & \frac{\partial h_1}{\partial t_Y} & \frac{\partial h_1}{\partial t_Z} \\ \frac{\partial h_2}{\partial \phi} & \frac{\partial h_2}{\partial \theta} & \frac{\partial h_2}{\partial \psi} & \frac{\partial h_2}{\partial t_X} & \frac{\partial h_2}{\partial t_Y} & \frac{\partial h_2}{\partial t_Z} \end{pmatrix} \quad (2.62)$$

Le vecteur d'état et la matrice de covariance d'erreur sont mis à jour en utilisant les mesures issues de la caméra. Après cela, ces mesures sont introduites à l'entrée de l'étape de correction, le vecteur d'état et la matrice de covariance d'erreur sont projetés à l'état prochain. En exécutant ces deux étapes récursivement, nous pouvons estimer correctement les angles de rotation et le vecteur de translation de la caméra par rapport au repère monde.

2.4.4 Filtre de Kalman étendu hybride

Cette méthode est simplement la combinaison des deux algorithmes présentés auparavant : le filtre de Kalman étendu et l'algorithme analytique. En effet, le problème avec le filtre de Kalman étendu est l'initialisation des paramètres, donc nous utilisons l'algorithme analytique afin de donner des valeurs initiales aux paramètres de la pose et estimer correctement le vecteur d'état du filtre de Kalman étendu (figure 2.29).

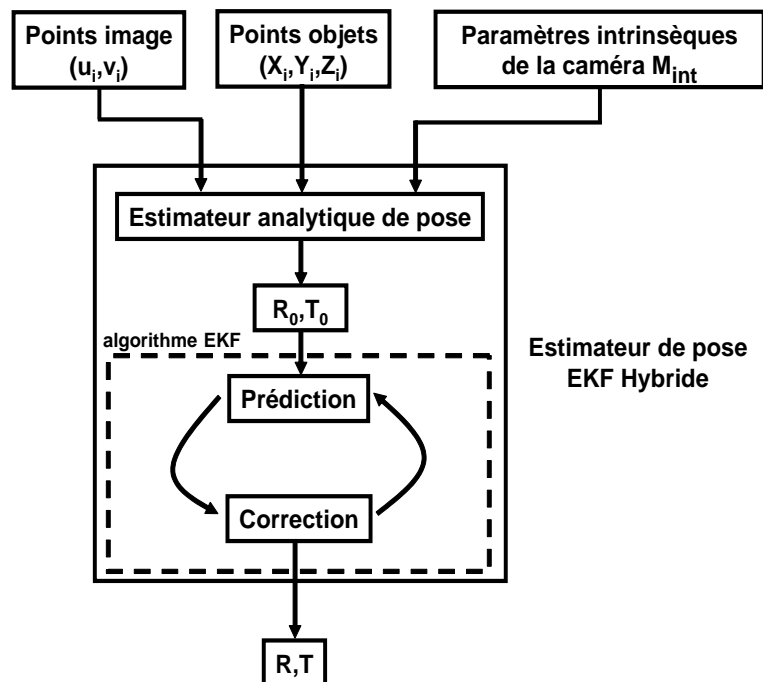


FIG. 2.29 – Diagramme d'estimation de la pose avec le filtre de Kalman étendu hybride à l'instant t .

2.4.5 Résultats expérimentaux

Nous présentons dans cette section, les résultats expérimentaux et une évaluation détaillée des différentes méthodes de localisation que nous avons présentées. Nous allons ensuite faire une comparaison entre ces méthodes afin de déterminer les performances de chaque méthode. Nous avons comparé la méthode du filtre de Kalman étendu hybride avec les trois autres méthodes qui sont : l'algorithme analytique, l'itération orthogonale hybride et le filtre de Kalman étendu. La comparaison entre ces algorithmes est réalisée sur les critères de performances suivants :

- temps d'exécution,
- l'erreur de reconstruction : mesure l'écart en pixel des points caractéristiques de la cible dans l'image et la projection du modèle 3D en utilisant les paramètres de la pose calculée,
- l'erreur de généralisation : est calculée en projetant les cibles qui n'étaient pas utilisées dans l'estimation de la pose sur le plan image et mesurer les variations pixeliques entre les points projetés des modèles 3D et les cibles correspondants détectés dans l'image,
- estimation de la distance réelle caméra-cible : c'est la différence entre l'évaluation de la distance estimée par l'algorithme d'estimation de la pose et la distance réelle calculée par le robot.

Les tests expérimentaux ont été réalisés en utilisant la configuration matérielle suivante :

- Pentium III 1.1GHz,
- carte d'acquisition Matrox Meteor II,
- caméra Sony XC-555P, focale : 6mm.

La caméra est calibrée en utilisant la méthode de Zhang [170]. Les paramètres intrinsèques sont illustrés sur le tableau 2.1.

Taille de l'image (pixels)		736 × 571	
Paramètres de projection		Paramètres de distortion	
Facteurs d'échelle		Coefficients de distortions radiales	
α_u	706.1	k_1	-0.2279
α_v	731.1	k_2	0.1479
Centre de projection optique		Coefficients de distortions Tangentielles	
u_0	388.0	p_1	-0.0007985
v_0	269.6	p_2	0.0006245

TAB. 2.1 – Paramètres intrinsèques de la caméra Sony XC-555P, utilisée dans les expérimentations.

2.4.5.1 Temps d'exécution

Notre première analyse concernant le temps d'exécution des différents algorithmes, montre que l'algorithme analytique est la méthode la plus rapide avec 19.3994 μs pour l'estimation d'une pose, le filtre de Kalman étendu hybride mets 112.2672 μs pour estimer la même pose, 153.2480 μs sont nécessaires pour l'itération orthogonale hybride. Enfin, le filtre de Kalman étendu estime les paramètres de la pose en 13530.3030 μs . En ce qui concerne le temps de calcul, nous pouvons dire que l'algorithme analytique est le plus rapide à l'inverse du filtre de Kalman étendu qui est très lent et semble inapproprié aux applications temps réel.

La méthode directe présente le meilleur temps d'exécution. En effet, la résolution des paramètres de pose dans ce cas, se base sur une résolution analytique et la solution est donnée immédiatement en résolvant le système d'équations. Par ailleurs, le traitement récursif des méthodes itératives engendre une lenteur dans les calculs. La convergence de ces méthodes est atteinte après plusieurs itérations. Pour ces méthodes numériques, l'initialisation est un paramètre crucial pour assurer la convergence rapide.

2.4.5.2 Erreur de reconstruction

Dans cette expérimentation, la caméra se déplace autour de l'objet cible, les quatre algorithmes estiment les paramètres de la pose et évaluent l'erreur de reconstruction dans l'image. Les algorithmes calculent 1400 poses, l'erreur est estimée en re-projetant le modèle d'objet sur l'image. Pour chaque estimation de pose, nous re-projetons le modèle de cible sur l'image et nous mesurons l'écart entre les coins de la cible et les coins du modèle projeté sur le plan image. Dans la figure 2.30, on remarque que les erreurs de reconstruction pour l'algorithme analytique, l'itération orthogonale hybride, le filtre de Kalman étendu hybride sont faibles par rapport à l'algorithme du filtre de Kalman étendu. Par conséquent, ces algorithmes permettent une meilleure estimation de pose et un recalage précis des objets virtuels ajoutés à la scène.

Les résultats numériques obtenus sur les différentes erreurs de reconstruction sont regroupés dans le tableau 2.2. Dans les valeurs données, certaines ont été retirées lorsqu'elles biaisaient les statistiques retournées (retrait des "outliers"). Nous remarquons que l'algorithme du filtre de Kalman étendu hybride est plus précis que les autres méthodes employées. Cette précision est due à l'association de deux méthodes dont l'utilisation permet de compenser les défauts de chaque technique. L'algorithme d'itération orthogonale présente pratiquement les mêmes performances que le filtre de Kalman étendu hybride, par contre, la méthode analytique s'est révélée moins performante que les deux autres méthodes car la résolution, dans ce cas, est directe et aucune optimisation n'est effectuée pour affiner le calcul des paramètres.

Toutefois, si la méthode de filtre de Kalman étendu semble présenter l'erreur la plus grande, c'est parce qu'elle nécessite une bonne initialisation des paramètres de la pose, ce qui implique plus d'itérations dans les calculs, et cela augmente d'autant l'erreur de reconstruction et le temps d'exécution de l'algorithme.

Algorithme	Algo. anal.	Itér. orth. hybr.	Filt. Kalm. éten.	Filt. Kalm. éten. hybr.
Err. reconst. (pixel)	0.5421	0.4671	3.2050	0.4651
Variance	0.0694	0.0373	0.5272	0.0360
Écart type	0.2634	0.1932	0.7261	0.1897

TAB. 2.2 – Estimation des différentes erreurs de reconstruction.

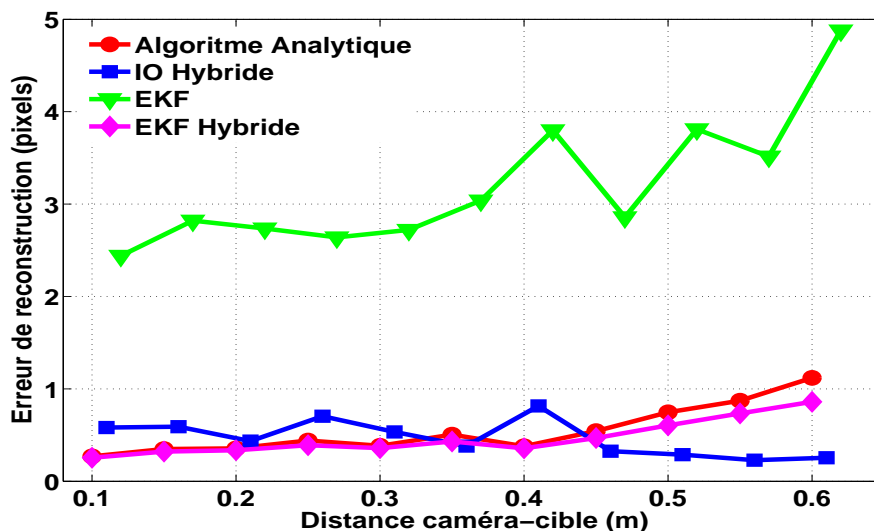


FIG. 2.30 – Erreur de reconstruction en fonction des distances caméra-cible.

2.4.5.3 Erreur de généralisation

Pour déterminer l'erreur de généralisation, nous avons utilisé un papier sur lequel nous avons imprimé quatre cibles carrées de 5cm de côté (figure 2.31). L'une des cibles est utilisée pour calculer les paramètres de la pose et les trois autres sont utilisées pour évaluer l'erreur de généralisation. Cette erreur est calculée en re-projetant les modèles d'objets qui n'ont pas été utilisés pour l'estimation de la pose et les projeter sur l'image. Les résultats obtenus sont représentés sur la figure 2.32. Le filtre de Kalman étendu hybride, présente les meilleures performances en terme d'erreur de généralisation en comparaison avec les autres algorithmes. La table 2.3, donne les valeurs numériques des erreurs de généralisations, ainsi que les variances et les écarts types des différentes méthodes.

Lors de nos tests, nous nous sommes aperçus que l'initialisation en utilisant un modèle de perspective faible n'assurait pas nécessairement la convergence de l'itération orthogonale vers la pose réelle de la cible. Il peut lui arriver de converger vers un autre minimum local. Ceci a des répercussions sur le temps de calcul nécessaire à cet algorithme pour converger. D'autre part, comme nous pouvons le constater sur la figure 2.32, l'algorithme du filtre de Kalman étendu donne des résultats disparates. Ceci s'explique par le critère de convergence de l'algorithme lorsque le résidu de la pose initiale et la pose réelle est très important. Le nombre d'itérations pour converger sera important pour le filtre et cela compromet le temps de calcul et la précision de la pose calculée. De ces valeurs (tableau 2.3), il ressort que c'est la méthode du filtre de Kalman étendu hybride qui procure les meilleurs résultats en termes d'erreur de généralisation.

Algorithme	Algo. anal.	Itér. orth. hybr.	Filt. Kalm. éten.	Filt. Kalm. éten. hybr.
Err. génér. (pixel)	10.9562	17.4712	16.0005	9.8435
Variance	48.0473	93.5641	450.2530	28.9559
Écart type	6.9316	9.6729	21.2192	5.3811

TAB. 2.3 – Estimation des différentes erreurs de généralisation.

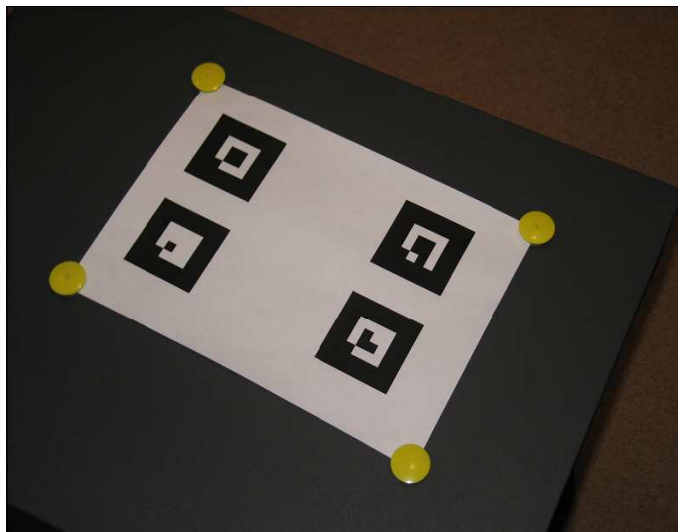


FIG. 2.31 – Cibles employées pour tester l'erreur de généralisation de chacun des algorithmes.

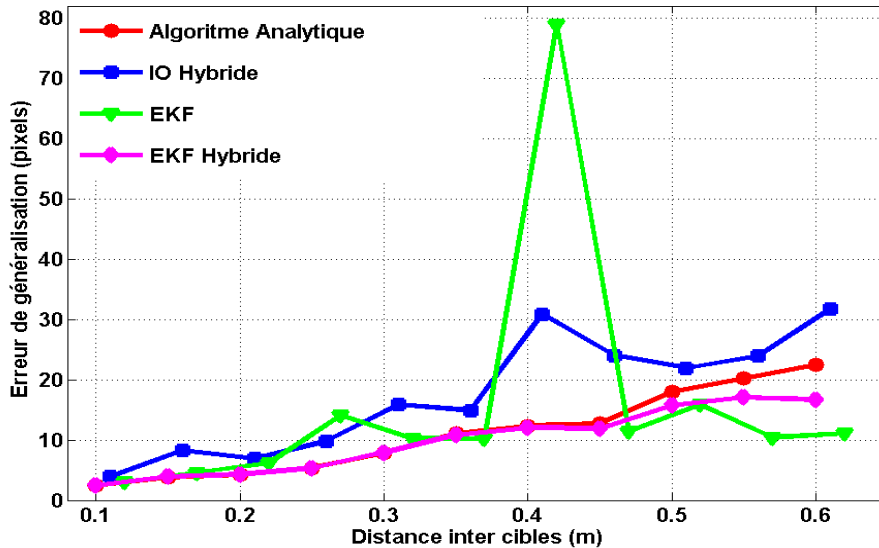


FIG. 2.32 – Erreur de généralisation en fonction de la distance inter-cibles.

2.4.5.4 Estimation de la distance réelle entre la caméra et la cible

Pour évaluer l'erreur de distance caméra-cible des différents algorithmes, nous utilisons un banc de calibration constitué d'un robot qui se déplace dans les deux directions X et Y (figure 2.33). La caméra est montée sur le banc, la cible est fixée sur l'autre côté du banc. Ce banc de calibration permet de contrôler le mouvement du robot et compare la distance calculée avec celle estimée par les différents algorithmes d'estimation de la pose. Nous avons échantillonné l'espace de déplacement du robot pour calculer la pose des différentes méthodes. Nous avons 1939 positions du robot pour lesquelles chaque algorithme estime les paramètres de pose et calcule la distance entre le centre optique de la caméra et la cible.

Nous avons classifié les résultats de pose obtenus en 10 classes et nous avons calculé les erreurs moyennes (figure 2.34) et les variances (figure 2.35) des méthodes d'estimation de la pose. Les résultats illustrés permettent de comparer les erreurs générées sur les distances réelles données par le robot (position du robot) et la position estimée par l'algorithme de pose. On remarque que la méthode analytique présente une importante erreur moyenne par rapport aux autres méthodes, cependant sa variance est relativement faible. Le filtre de Kalman étendu hybride et l'itération orthogonale présentent les meilleures performances à l'inverse de l'algorithme de filtre de Kalman étendu qui présente une grande variance autour de son erreur moyenne. La figure 2.36 représente la distance réelle calculée par le robot par rapport à la distance estimée par les différents algorithmes d'estimation de pose. En effet, cette évaluation détermine, avec précision, l'erreur de distance générée de chaque algorithme de pose. L'interprétation de ces erreurs est effectuée par une approximation des courbes représentées dans la figure 2.36 avec une régression non-linéaire pour l'itération orthogonale hybride, le filtre de Kalman étendu et le filtre de Kalman étendu hybride et une régression quadratique pour l'algorithme analytique. L'erreur moyenne de l'algorithme analytique est de 0.81% (une erreur moyenne de 8.1mm pour une distance de 1m). L'itération orthogonale hybride présente une erreur de 0.84%, quant à le filtre de Kalman étendu, il dégénère et présente une erreur moyenne de 2.6%. La valeur la plus faible de l'erreur est obtenue avec le filtre de Kalman étendu hybride où elle est estimée à 0.72%. On conclut que le filtre de Kalman étendu hybride est le meilleur estimateur de distance par rapport aux autres algorithmes.

Les paramètres des différentes régressions utilisées sont représentées sur le tableau 2.4. Le tableau 2.5 récapitule les erreurs d'estimation de distance des méthodes utilisées. Les meilleurs résultats de

chaque critères de comparaison sont coloriés en vert, alors que les mauvais résultats sont coloriés en rouge. À la lecture des résultats de ce tableau, il semblerait que l'algorithme du filtre de Kalman étendu hybride soit le meilleur compromis. Toutefois, il est possible d'améliorer cet algorithme, plus spécifiquement sa vitesse de convergence en effectuant des optimisations.

Algorithme	Valeurs finales des paramètres	Erreur asymptotique standard
Algo. anal.	$a_1 = 1.0081$	0.0009
	$b_1 = 0.0068$	0.0007
Itér. orth. hybr.	$a_2 = 1.0084$	0.0004
	$b_2 = 0.0068$	0.0005
Filt. Kalm. éten.	$a_3 = 2.6034$	0.2849
	$b_3 = -1.1006$	0.3488
Filt. Kalm. éten. hybr.	$a_4 = 1.0071$	0.0003
	$b_4 = 0.0087$	0.0003

TAB. 2.4 – Paramètres des régressions utilisées.

Algorithme	Algo. anal.	Itér. orth. hybr.	Filt. Kalm. éten.	Filt. Kalm. éten. hybr.
Erreur moyenne (m)	0.0168	0.0057	0.0030	0.0046
Variance	6.9574e-6	7.0165e-6	0.3567	3.6445e-6
Écart type	0.0026	0.0026	0.5973	0.0019
Temps (μs)	660	21420	1894200	15680

TAB. 2.5 – Résultats des différentes erreurs d'estimation de distance.

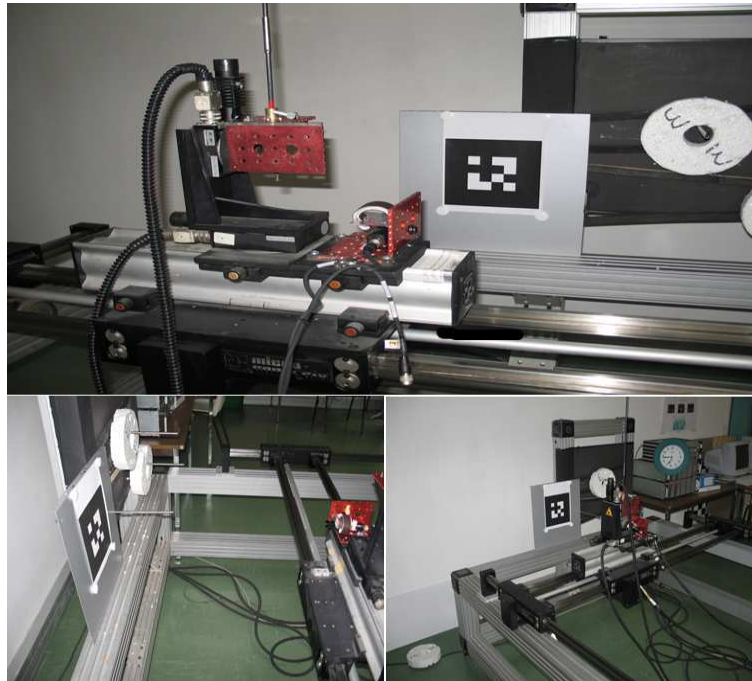


FIG. 2.33 – Banc de calibration utilisé pour l'évaluation de la distance caméra-cible.

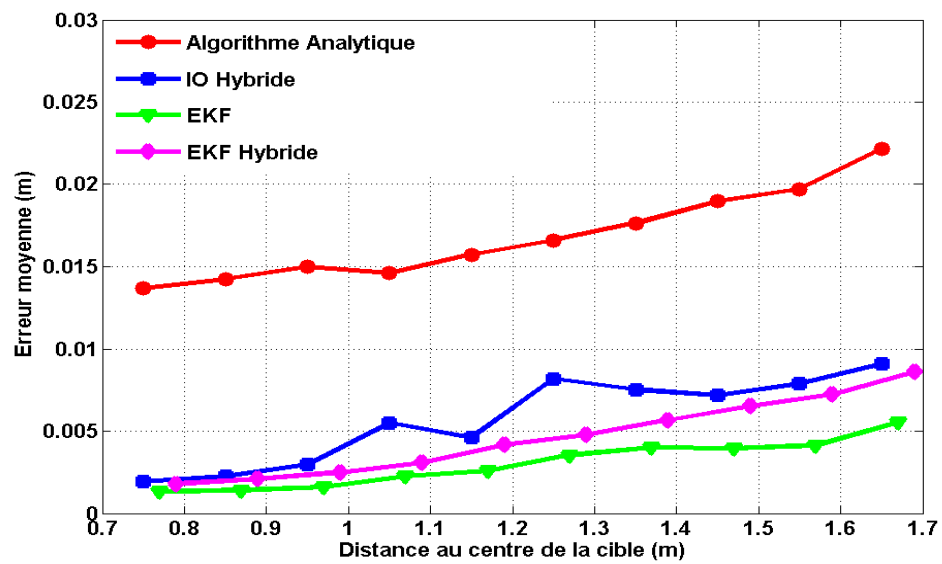


FIG. 2.34 – Erreur moyenne des données classifiées.

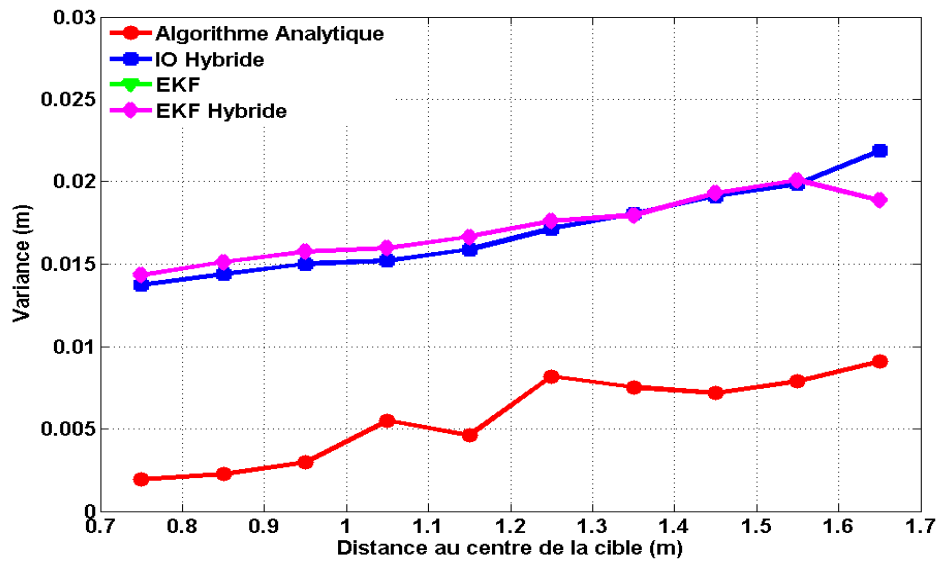


FIG. 2.35 – Variances des données classifiées.

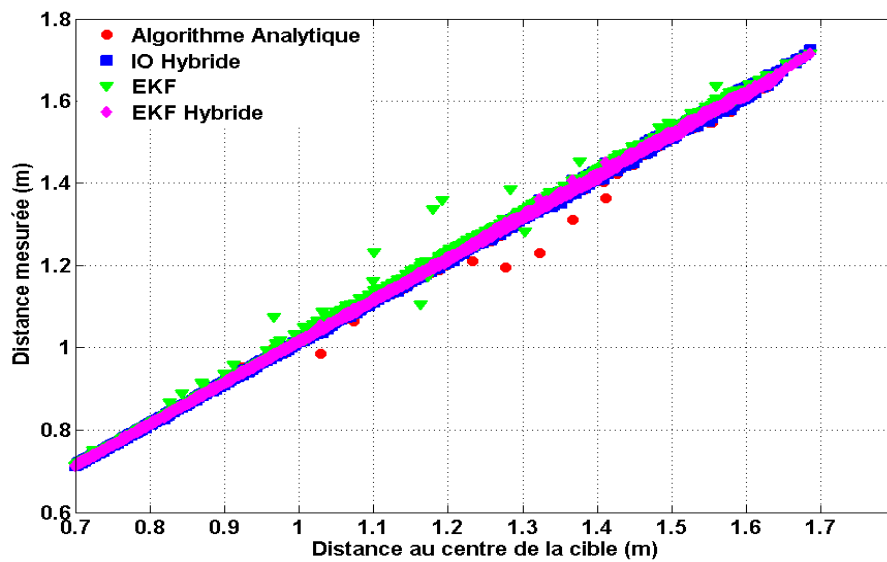


FIG. 2.36 – Évaluation des distances mesurées par rapport aux distances réelles.

2.4.5.5 Résultats de recalage

Les paramètres de la pose étant déterminés, nous allons maintenant projeter un objet virtuel sur les cibles détectées dans l'image pour évaluer visuellement le rendu graphique. Dans cette expérience, la caméra est déplacée librement autour des cibles. L'algorithme d'identification détecte et traque les cibles et le filtre de Kalman étendu hybride estime la pose de la caméra. Nous pouvons voir sur la figure 2.37 que les objets virtuels, représentés par des cubes, sont bien superposés sur les images réelles et ils restent collés sur les cibles pour les différentes pose de la caméra.

Dans un deuxième temps, nous avons testé le recalage sur des objets 3D pour vérifier les capacités de généralisation de notre algorithme sur ce type d'objets. Nous avons utilisé notre modèle de cible carrée pour estimer la pose de la caméra, ces paramètres de pose ont été employés pour projeter un modèle simplifié 3D sur un objet industriel représenté par une culasse de voiture (figure 2.38). Sur cette figure, des lignes droites sont superposées avec précision sur les contours de la culasse ce qui permet de confirmer l'aptitude de généralisation de l'algorithme d'estimation de pose sur des modèles 3D.

Nous avons réalisé un autre test qui consiste à employer deux objets 3D sous forme de boîtes. Les points caractéristiques d'une boîte servent à déterminer la pose de la caméra, le modèle 3D du deuxième objet est ensuite projeté en utilisant la pose calculée (figure 2.39). Les résultats obtenus montrent l'efficacité de notre méthode pour le recalage d'objets sur des modèles 3D.

Ces tests expérimentaux prouvent la précision de recalage du filtre de Kalman étendu hybride dans un processus de suivi de cibles carrées. De plus, cet estimateur de pose est aussi efficace pour la réalisation des tâches d'alignement sur des modèles d'objets 3D. Enfin, des évaluations concernant la généralisation sur des objets 3D ont démontré la flexibilité de cet algorithme.

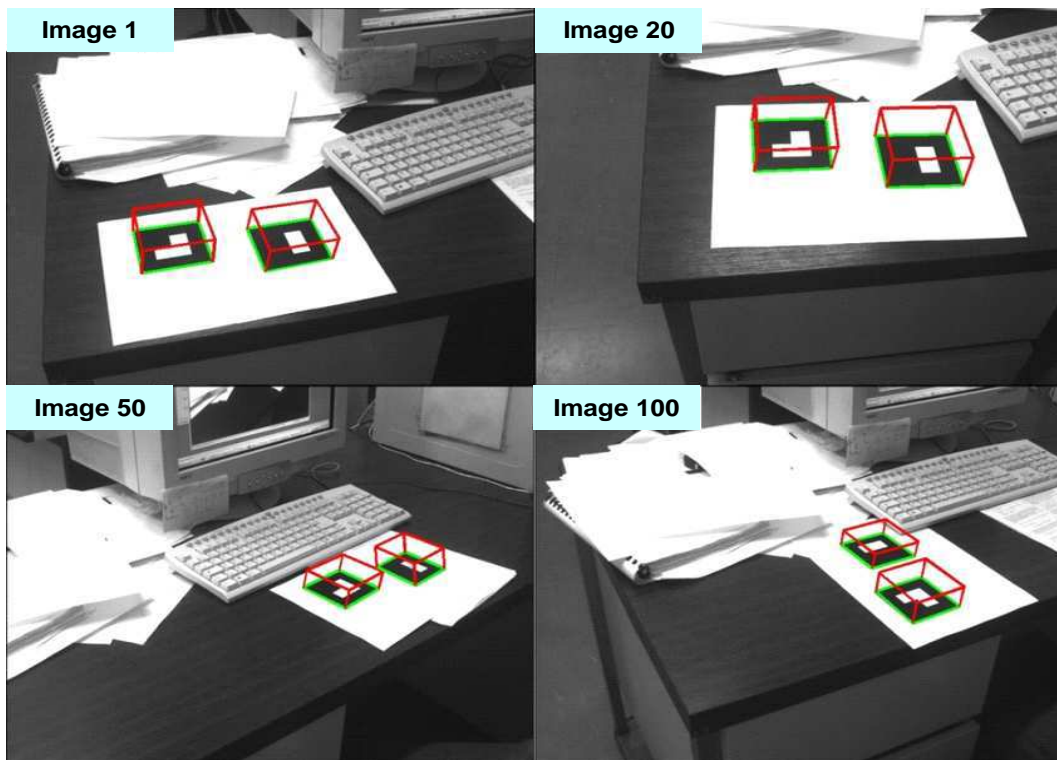


FIG. 2.37 – Résultats de recalage dans une séquence de tracking.

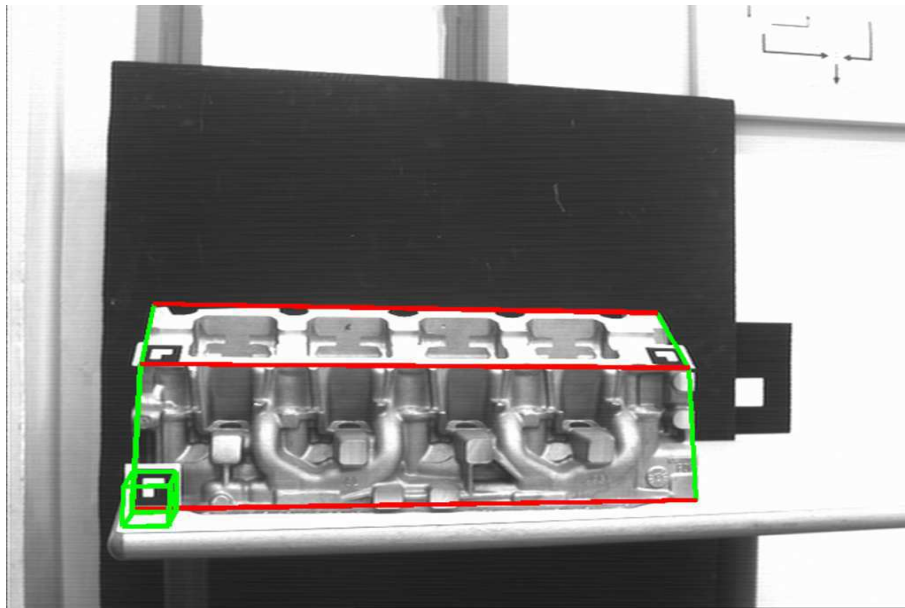


FIG. 2.38 – Généralisation de recalage sur un objet 3D en utilisant une cible coplanaire.

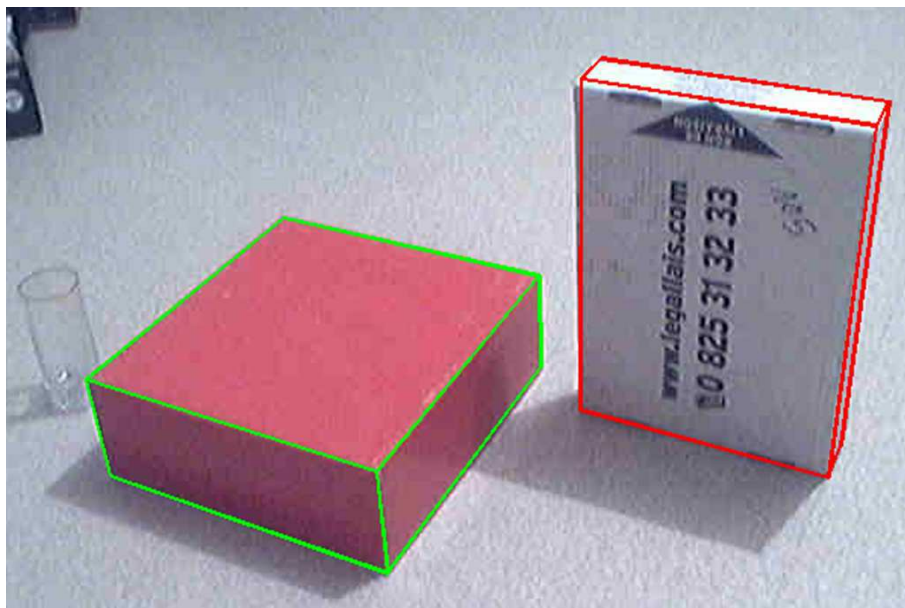


FIG. 2.39 – Généralisation de recalage sur un objet 3D en utilisant une cible 3D.

2.4.5.6 Éléments de comparaison avec d'autres études équivalentes

Des études comparatives entre des algorithmes d'estimation de pose ont été présentées dans la littérature. Ces travaux ont été réalisés pour évaluer les performances des méthodes selon plusieurs critères d'appréciation.

Ansar et Daniilidis [7] ont développé des algorithmes à n -points et n -lignes pour une estimation rapide et précise de la pose. Leurs méthodes ont été évaluées selon plusieurs critères de performances. Les auteurs ont présenté une approche mathématique pour la résolution des algorithmes linéaires en présence de bruit. Une analyse de sensibilité des algorithmes face aux erreurs dans l'image a été effectuée. Les résultats obtenus ont montré que les algorithmes à n -points sont plus performants que les méthodes itératives employées dans l'étude comparative.

Quan et Lan [127] ont présenté une famille de méthodes linéaires employant n -points ($n = 4, 5$) pour l'estimation de la pose. Ces algorithmes linéaires donnent des solutions uniques si les points utilisés sont bien répartis dans l'environnement. Les auteurs ont effectué une évaluation expérimentale de leurs méthodes et ont montré que leurs algorithmes sont précis et présentent des solutions stables pour diverses configurations.

Didier [35] a présenté une étude comparative de plusieurs algorithmes d'estimation de pose. L'auteur a évalué les performances globales de son système de localisation en le comparant avec des algorithmes connus. L'étude a porté sur divers critères de performances tels que : le temps d'exécution, l'erreur de reconstruction, etc. L'auteur a proposé en définitive un système hybride de calcul de la pose reposant sur un algorithme spécifique qu'il a développé et l'itération orthogonale. L'étude comparative réalisée a permis d'affirmer que son système apporte des solutions intéressantes pour effectuer la localisation par la vision à l'aide des cibles codées.

Les travaux réalisés Quan et Lan [127] n'ont pas permis de faire directement une comparaison entre les performances de leurs algorithmes et notre système de localisation, notamment les erreurs de re-projection ou les erreurs d'appréciation des distances. Les conditions expérimentales et le type des méthodes employées dans cette étude, ne nous permettent pas de trouver des éléments de comparaison tangibles avec notre étude expérimentale.

Par ailleurs, une analyse portant sur l'erreur de re-projection, nous permet de trouver un élément comparatif entre les méthodes d'estimation de pose employées dans notre étude et celle d'Ansar et Daniilidis [7]. Ces auteurs ont estimé l'erreur de re-projection, déterminée sous forme d'erreur quadratique moyenne. Leur méthode à n -points linéaires NPL (N Point Linear) présente une erreur $< 5pixels$ lorsque la caméra est distante de $10 \times$ taille du côté de la cible. Les résultats obtenus ont montré que l'algorithme à n -points linéaires est plus performant que les autres estimateurs de pose utilisés dans l'étude comparative, comme, l'algorithme de Fiore [47] ou celui de Quan et Lan [127]. Dans notre analyse, la plus faible erreur de re-projection est donnée par le filtre de Kalman étendue hybride. Cette erreur est estimée à $< 0.5pixels$ pour une distance $< 0.6m$ ce qui représente $12 \times$ taille du côté de la cible. Nous concluons donc, que notre méthode présente une meilleure précision dans l'estimation des paramètres de la pose comparée à la méthode à n -points linéaires d'Ansar et Daniilidis.

L'étude effectuée par Didier [35], nous permet aussi de trouver un espace commun d'évaluation car les critères de comparaison employés sont les mêmes que les nôtres, à savoir : le temps d'exécution, l'erreur de reconstruction, l'erreur de généralisation, et l'estimation de la distance réelle. De plus, l'algorithme analytique et l'itération orthogonale hybride utilisés pour la comparaison dans le travail de Didier, sont aussi employés dans notre étude. Dans [35], la méthode d'itération orthogonale présentait le meilleur compromis en terme de temps de calcul, d'erreur de généralisation et d'erreur sur l'estimation des distances par rapport aux algorithmes des moindres carrés et l'algorithme analytique. Alors que dans nos évaluations, le filtre de Kalman étendu hybride réalise pratiquement les meilleures performances dans les différents tests expérimentaux effectués.

Nous pouvons conclure que les techniques hybrides sont mieux adaptées à l'estimation de la pose. Ce type de méthodes contribue à l'amélioration de la rapidité et la précision du recalage dynamique. Ceci nous permet d'affirmer que leur utilisation s'avère novatrice et prometteuse pour effectuer des tâches de suivi dans les applications de réalité augmentée.

2.4.6 Discussion

Dans cette étude, nous avons comparé les performances de 4 algorithmes d'estimation de pose. Nous avons évalué ces méthodes en utilisant un protocole expérimental pour calculer différentes sources d'erreurs et estimer les distances réelles. Nous avons utilisé 3 méthodes itératives et une nouvelle méthode analytique basée sur le calcul direct des paramètres de la pose.

Les algorithmes d'estimation de pose utilisés sont basés sur les points caractéristiques des cibles carrées détectées. Pour extraire ces points de l'image, nous avons développé un algorithme de reconnaissance de cibles qui effectue l'échantillonnage spatial des zones rectangulaires de l'image afin de déterminer le code de la cible. Cette méthode comporte une partie de détection de la forme carrée des objets d'intérêts, et une autre partie pour identifier le code binaire du motif de la cible. Cette méthode présente l'avantage de définir nos propres codes qui permettront de distinguer nos cibles parmi les autres objets de la scène.

Les principales contributions de cette étude sont :

- L'utilisation d'un nouvel algorithme d'identification de cibles basé sur les cibles codées.
- Un algorithme hybride d'estimation de pose combinant une méthode directe et une autre itérative.
- Une comparaison entre différentes méthodes d'estimation de pose selon les critères suivants : temps d'exécution, erreurs et estimation de distances.

Divers travaux ont été réalisés dans la littérature concernant l'estimation de la pose en utilisant les méthodes directes ou itératives, quelques auteurs se sont intéressés à la comparaison et l'évaluation de ces méthodes [7] [33] [35] [127]. Dans le tableau 2.6, nous comparons différentes méthodes d'estimation de pose, où nous précisons l'année, la nature de l'algorithme et les conditions d'applications.

Enfin, les deux types d'algorithmes : itératif ou analytique ont des avantages et des inconvénients. Les méthodes itératives sont précises mais très gourmandes en temps de calcul à cause de la mauvaise initialisation des paramètres à estimer. D'autres part, les méthodes analytiques sont rapides mais peu précises. Nous avons donc, exploité la complémentarité de ces deux type de méthodes pour compenser les inconvénients de chaque approche.

Méthode	Année	Type	Conditions d'application
Dhome et al. [34]	1989	Analytique	3 lignes
Dementhon et Davis [33]	1992	Analytique	3 points
POSIT	1995	Itératif	4 points non-coplanaires
Itération orthogonale [95]	2000	Itératif	3 points
Filtre de Kalman étendu [26]	2002	Itératif	3 points
Ansar et Daniilidis [7]	2003	Analytique	4 points
Didier [35]	2005	Analytique	4 points coplanaires
Filtre de Kalman étendu hybride [2]	2007	Analytique et itératif	4 points coplanaires

TAB. 2.6 – Récapitulatif de quelques méthodes d'estimation de pose.

2.5 Conclusion

Dans ce chapitre, nous avons présenté quatre algorithmes dédiés à la résolution du problème de localisation par la vision. Dans un premier temps, nous nous sommes intéressés à l'extraction des indices visuels 2D des objets d'intérêts dans l'image. Nous avons développé deux approches de reconnaissance de cibles. La première méthode permet l'identification des zones rectangulaires de l'image en appliquant un échantillonnage spatial sur les objets de forme quadrilatérale. La deuxième méthode est basée sur une localisation par contour actif. Des tests expérimentaux ont montrés que la technique de suivi par contour actif présente de multiples limitations liées aux changements de conditions d'environnement, de plus, l'inconvénient majeur de cette méthode est sa dépendance par rapport au contour initial.

Ensuite, nous avons développé une nouvelle approche pour l'estimation de la pose de la caméra basée sur la combinaison de deux méthodes qui sont un algorithme analytique et le filtre de Kalman étendu. L'algorithme analytique calcule les paramètres initiaux de la pose, ces paramètres sont utilisés après pour initialiser un second algorithme d'estimation de pose basé sur le filtre de Kalman étendu. Cette méthode nous l'avons intitulée le filtre de Kalman étendu hybride.

Enfin, nous avons effectué une étude comparative entre les différents algorithmes d'estimation de la pose de la caméra en utilisant les cibles codées. Nous avons évalué les performances de notre système de localisation en le comparant avec d'autres méthodes. Cette étude a porté sur les critères de performances suivants : le temps d'exécution, l'erreur de reconstruction, l'erreur de généralisation et l'estimation de la distance caméra-cible. Les tests expérimentaux pour évaluer la distance réelle entre la caméra et les cibles, ont été réalisés en utilisant un banc de calibration. Les résultats obtenus de notre algorithme d'estimation de pose étaient robustes et ont prouvés que notre système fournit une solution intéressante quant à la localisation de la caméra en utilisant des cibles codées. Finalement, notre système a été testé pour gérer les augmentations dans des applications de réalité augmentée, les résultats de recalage obtenus étaient corrects et précis.

Les algorithmes que nous avons proposés sont destinés à résoudre le problème de la localisation par la vision. Les méthodes employées pour l'identification des cibles codées et l'estimation de la pose de la caméra, constituent le premier module de notre architecture multimodale de suivi. Cette partie permet le suivi de cibles lorsqu'elles sont reconnaissables par le système de vision. Cependant, une occultation d'un indice visuel de la cible, mettra en échec la reconnaissance et par conséquent l'estimation de la pose. Le chapitre suivant s'intéressera au suivi robuste des points d'intérêts dans le cas d'occultations partielles.

Chapitre 3

Gestion d'occultations en réalité augmentée

3.1 Introduction

Le suivi de cibles en temps réel est une tâche complexe qui doit se faire séquentiellement de manière précise et stable. Les avancées dans le domaine de la vision par ordinateur ouvrent d'importantes perspectives permettant d'envisager des systèmes beaucoup plus autonomes et qui s'adaptent aux changements imprévus de leurs environnements. Le problème d'occultations des indices visuels est souvent considéré comme l'un des principaux verrous dans un processus de suivi d'objet.

Dans ce chapitre, nous nous sommes intéressés à la gestion d'occultations afin de maintenir le recalage des objets de synthèse sur les images. Nous avons développé une méthode de suivi robuste permettant de gérer les occultations partielles des cibles.

Notre approche est basée sur l'algorithme RANSAC, qui permet d'estimer une transformation entre les points appariés de deux images successives afin de déterminer les bonnes mises en correspondances. Cet algorithme consiste à estimer les paramètres recherchés avec le minimum de mesures nécessaires puis vérifier si d'autres mesures confirment cette première estimation. Si un consensus est obtenu, l'estimation est retenue.

Dans ce chapitre, nous allons donc, dans un premier temps, passer en revue les estimateurs robustes et les systèmes de gestion d'occultations en réalité augmentée. Puis, nous décrivons la méthode que nous proposons pour apporter une solution à ce problème. Enfin, nous terminons avec une série d'expérimentations que nous avons menées afin d'évaluer notre approche.

3.2 Les estimateurs robustes

Les méthodes robustes se classent en deux grandes familles : les méthodes qui compensent les erreurs de mesures liées aux systèmes d'acquisition, et les méthodes qui compensent l'influence des erreurs grossières et distinguent explicitement les données correctes et les données aberrantes par des mesures robustes, comme les M-, R- et L-estimateurs. Cette deuxième famille de méthodes contient aussi des techniques classiques comme la transformation de Hough, RANSAC et LMedS, ainsi que des estimateurs robustes développés récemment en vision :

- CBD-estimateur (Consensus By Decomposition) [113].
- MF-estimateur (Model Fitting) [172].
- MINPRAN-estimateur (MINimization of the Probability of RANdomness) [148].
- RESC-estimateur (RESidual Consensus) [169].
- ALKS-estimateur (Adaptive Least K-th order Square) [89].
- MUSE-estimateur (Minimum Unbiased Scale Estimator) [111].
- MOM-estimateur (Method Of Moments)[119].

Les techniques robustes ont pour objectif de définir une fonction de coût qui minimise le minimum global et qui ne soit pas affecté par les données aberrantes. Ces approches cherchent par ailleurs à estimer de façon robuste l'écart type des mesures correctes. La caractéristique principale d'une méthode robuste est son "point de rupture". Le point de rupture est le pourcentage de mesures aberrantes qui mettent en défaut l'algorithme. À titre d'exemple, l'algorithme des moindres carrés a un point de rupture de 0% car il suffit d'une seule mesure aberrante pour obtenir une estimation fautive des paramètres. D'autres caractéristiques très importantes dans les applications robotiques sont la complexité algorithmique et la vitesse de convergence des algorithmes. Ces deux facteurs différents se traduisent en une diminution de la fréquence des boucles de commande. De plus, il faut tenir compte du compromis entre la vitesse de convergence et le point de rupture des algorithmes [103].

Plusieurs estimateurs robustes ont été développés dans la littérature, nous allons dans ce qui suit décrire quelques estimateurs robustes les plus connus et les plus utilisés en vision par ordinateur.

La transformée de Hough : La technique de Hough [73] est parmi les premières méthodes robustes en vision dans le sens où elle tolère les erreurs grossières. C'est une technique de reconnaissance de formes inventée en 1962 par Paul Hough, utilisée dans le traitement d'images numériques. L'application la plus simple permet de reconnaître les lignes d'une image. Toutefois, des modifications peuvent être apportées à la méthode pour reconnaître n'importe quelle forme : c'est la transformée généralisée de Hough développée par Richard Duda et Peter Hart en 1972 [40] (figure 3.1). La transformée de Hough est une technique pour détecter les droites dans les images très bruitées. Cela consiste à maximiser, de manière robuste, la cohérence dans l'ensemble de toutes les solutions possibles, elle est réalisée par une recherche exhaustive, est donc très coûteuse en terme de temps de calcul.

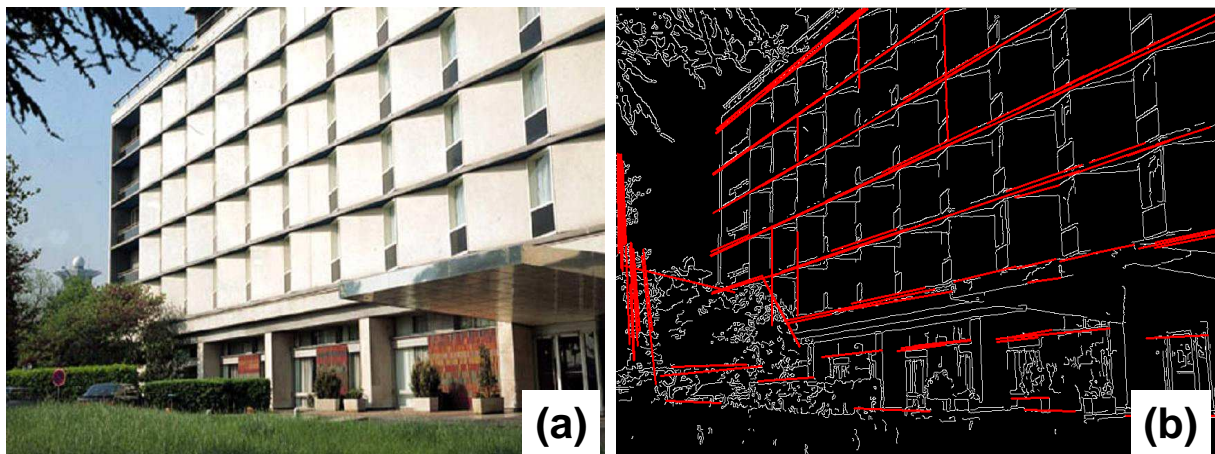


FIG. 3.1 – Détection des lignes avec la transformée de Hough. (a) Image originale. (b) Résultat de l'application de la transformée de Hough.

M-estimateurs : Les M-estimateurs [19] sont très populaires en vision par ordinateur. Ils minimisent la somme d'une fonction symétrique et définie positive $\rho(r_i)$ du résidu r_i , avec un seul minimum à $r_i = 0$. La M-estimation des paramètres est obtenue en convertissant la minimisation de $\sum_i \rho(r_i)$ en un problème de moindres carrés pondérés. Toutefois, cette méthode souffre du problème d'initialisation et de convergence. Des améliorations ont été proposées pour surmonter les défauts des M-estimateurs. Récemment, Li [91] a développé le AM estimateur (Annealing M-estimateur), une combinaison de M-estimateur et GNC (Graduated Non-Convexity [21]) pour améliorer la précision, la stabilité et le point de rupture.

Les M-estimateurs sont plus généraux qu'un estimateur au maximum de vraisemblance car ils permettent l'utilisation de différentes fonctions de minimisation qui ne correspondent pas nécessairement

à une distribution normale des données. Beaucoup de fonctions ont été proposées dans la littérature qui permettent soit d'accepter ou de rejeter les mesures incertaines. Le M-estimateur a pour objectif de réduire la sensibilité aux données aberrantes [30].

R-estimateurs : Les R-estimateurs sont basés sur un tri de l'ensemble des résidus. Ils proposent d'estimer les paramètres en minimisant $\sum_i a_n(R_i)r_i$ où r_i est le résidu, R_i son rang, et a_n est une fonction de coût. a_n doit être monotone et doit vérifier la condition $\sum_i a_n(R_i) = 0$ [74].

L-estimateurs : Les L-estimateurs utilisent la combinaison linéaire de la statistique d'ordre. Les méthodes basées sur la médiane et la moyenne appartiennent à cette classe. Des simulations réalisées par Heiler [67], ont démontré que les L-estimateurs donnent des résultats moins satisfaisants que ceux des deux autres classes. Le L-estimateur a un point de rupture inférieur à $\frac{1}{p+1}$, p est le nombre de paramètres [61].

LMedS-estimateur (Least Median of Squares) : L'estimateur des moindres carrés médians est très populaire [136], il possède un point de rupture proche de 0.5. L'efficacité relative de cette méthode peut être améliorée en la combinant avec d'autres techniques basées sur les moindres carrés et la complexité en temps peut être réduite par un échantillonnage aléatoire. Bien que la méthode LMedS ait dominé en vision pendant des années, cependant, elle présente l'inconvénient de tolérer au plus 50% d'erreurs grossières et elle n'a pas d'expression analytique.

CBD (Consensus By Decomposition) : Le principe de la méthode du consensus par décomposition est le suivant :

- Calculer un modèle candidat en choisissant un petit sous-ensemble de données.
- Appliquer ce modèle à l'ensemble des données.
- Calculer une mesure de qualité globale pour ce modèle.
- Répéter cette procédure pour optimiser la mesure de qualité.

Pour avoir une bonne estimation aux moindres carrés médians, il est nécessaire qu'au moins un modèle candidat apporte des valeurs correctes pour les paramètres. Quand le bruit aléatoire est important il y aura une estimation non biaisée mais avec une grande variance. Donc le modèle candidat ne donne plus une estimation fiable des paramètres du modèle réel [113].

MF-estimateur (Model Fitting) : C'est un estimateur robuste d'ajustement de modèle par régression. Cet estimateur modélise des fonctions de vraisemblances inconnues. Une modélisation partielle est réalisée par une règle statistique bayésienne avec un grand nombre de données heuristiques. Le MF-estimateur est utilisé pour la sélection de multiple seuils et l'estimation de mouvement [172].

3.3 Domaines d'applications en vision

Comme nous l'avons déjà cité, en vision, on peut distinguer deux types d'erreurs : les erreurs liées à la mesure, souvent modélisées par des distributions gaussiennes centrées, et les erreurs aberrantes ou les erreurs grossières issues des faux appariements. L'application des méthodes robustes permet de pallier les divers problèmes liés aux erreurs d'acquisition ou de traitement dans un système de vision. Nous citons maintenant les applications où les estimateurs robustes sont souvent utilisés.

Vision stéréoscopique : Les faux appariements en vision stéréoscopique sont inévitables et peuvent fausser le résultat de la reconstruction 3D. Très souvent, les estimateurs robustes sont utilisés pour corriger les mauvais appariements des points [144].

Flot optique : Les méthodes robustes sont aussi utilisées pour gérer les occultations lors d'un suivi par flot optique. L'estimation du flot optique concerne le calcul de deux composantes de la vitesse à partir d'une seule contrainte du flot optique d'image [72]. L'occultation crée de fausses mesures et la combinaison de ces mesures avec les contraintes du flot optique de différentes régions du mouvement, peut produire d'énormes erreurs dans l'estimation du flot optique, d'où la nécessité des méthodes robustes [139] [140].

Structure à partir du mouvement : Lors d'un suivi d'objet dans une séquence d'images, on suppose souvent que tous les points appartiennent au même objet rigide [68] [155]. Si les objets dans la scène ont des mouvements différents, les appariements des points dans les images peuvent être aberrants. Grâce aux méthodes robustes, on peut surmonter ce problème.

Autres applications : Les méthodes robustes sont très employées en traitement d'images [19] :

- la détection de contours [112],
- la segmentation [77],
- l'estimation de la matrice fondamentale [171],
- l'appariement visuel [88],
- l'estimation de pose [62],
- la reconstruction de surface [144],
- et enfin l'estimation du tenseur trilinéaire [154].

3.4 Contraintes liées au suivi de points d'intérêts

Dans ce chapitre nous nous sommes intéressés à l'application de la méthode robuste dans le cadre d'un suivi de points caractéristiques 2D dans une séquence d'images. Les méthodes de suivi de primitives des points ou des segments de droites sont fondées sur le même principe et emploient des informations adaptées pour chaque type de primitive. Lors d'un suivi, les primitives successives dans une image sont associées les unes aux autres en utilisant des contraintes permettant de lever l'ambiguïté d'association. Les algorithmes de suivi sont établis de manière à exploiter ces contraintes. Nous présentons maintenant les principales contraintes auxquelles il faut prêter attention afin de réaliser un suivi correct des points d'intérêts.

3.4.1 La contrainte du mouvement

En général, dans les séquences d'images, la fréquence employée permet une perception fluide du mouvement, toutefois, une variation brutale de la vitesse implique une certaine discontinuité du mouvement des objets observés. Cette contrainte permet de choisir entre plusieurs associations, les points concurrents. Le filtrage de Kalman est souvent utilisé pour prédire le mouvement et modéliser l'incertitude sur le mouvement. Cette incertitude définit la zone de recherche autour du point prédit.

3.4.2 L'unicité de la mise en correspondance

Cette contrainte permet d'associer un point donné dans une image à un seul point dans la deuxième image. Cependant, il n'y a pas toujours de correspondant puisque certains points peuvent apparaître et disparaître, par exemple lors d'une occultation, et donc il peut y avoir soit une seule association ou bien aucune.

3.4.3 Caractéristiques des primitives

Des mesures sur l'image sont utilisées pour faire l'appariement entre les primitives compatibles. Ces mesures concernent le voisinage du point, tels que les descripteurs de luminance, de couleur, de texture et des contraintes géométriques locales. Des études s'intéressant à la comparaison de ces caractéristiques et à l'amélioration des résultats en combinant différentes mesures.

3.4.4 Le suivi robuste des points d'intérêts

Les techniques de mise en correspondance peuvent ne pas être robustes à toutes les transformations (photométriques ou géométriques) que peuvent subir les images. La présence de bruit est l'une des raisons qui peut induire en erreur ces méthodes. Ceci implique la présence de mauvaises correspondances ou de faux appariements. Ces mauvaises correspondances faussent l'estimation de la transformation entre les deux images à recalculer. Il est nécessaire d'éliminer ces faux appariements pour ne garder que les couples de primitives appropriés.

Dans cette étude, un suivi de points d'intérêts en utilisant l'algorithme d'estimation robuste RANSAC est réalisé. L'utilisation de cette méthode robuste est nécessaire afin de réaliser des tâches en environnement réel. Les méthodes basées vision combinent les contraintes de mise en correspondance des points mesurés dans l'image. La vérification de ces contraintes n'est pas triviale car plusieurs points peuvent être candidats pour un point donné. La contrainte d'unicité peut induire à trouver des appariements susceptibles de correspondre à un même point donné et la non-régularité dans le mouvement compromettent le fonctionnement de l'algorithme. La méthode robuste que nous proposons basée sur RANSAC permet de pallier tous ces problèmes en calculant une transformation robuste qui permet de retenir uniquement les bons appariements. De plus, l'algorithme de suivi doit gérer toutes ces contraintes en réalisant un compromis entre complexité et performances qui dépendent des besoins de l'application.

Nous allons maintenant décrire l'algorithme RANSAC utilisé pour le suivi robuste des points 2D dans l'image.

3.5 La méthode RANSAC

RANSAC est une abréviation de "RANdom SAMple Consensus" (le consensus d'échantillons aléatoires), proposé par Fischler et Bolles en 1981 [49]. C'est un algorithme d'estimation de paramètres d'un modèle mathématique pour un ensemble de données observées. Les données se composent de données correctes (inliers) ou les points sont exprimés en fonction des paramètres du modèle, et des données aberrantes (outliers) qui sont des points qui ne permettent pas l'ajustement du modèle (figure 3.2).

Les entrées de l'algorithme RANSAC sont l'ensemble des données observées, les données aberrantes peuvent provenir du bruit ou bien des mesures erronées. RANSAC permet, à partir d'un ensemble de données, d'ajuster les paramètres du modèle en sélectionnant d'une manière itérative un sous ensemble aléatoire de données originales considérées comme des données correctes.

Une implémentation très simple de l'algorithme consiste, tout d'abord, à définir les notations suivantes :

- x , le vecteur paramètre du modèle.
- N , le nombre de données permettant l'estimation des paramètres du modèle.
- M , le nombre total de données.
- p_{succes} , la probabilité des données aléatoires permettant l'ajustement du modèle.
- p_{echec} , la probabilité que l'algorithme se termine sans trouver le bon ajustement des paramètres.

Les étapes de l'algorithme sont alors :

1. Sélectionner N données aléatoirement.
2. Estimer les paramètres du modèle x .
3. Trouver combien de données de M ajuste le modèle avec le vecteur paramètre x en utilisant une tolérance. Ce nombre est appelé K .
4. Si K est grand, accepter l'ajustement et quitter avec succès.
5. Répéter les étapes 1 à 4, L fois.
6. Échec sinon

L est défini par :

$$L = \frac{\log(p_{echec})}{\log(1 - (p_{succes})^N)} \quad (3.1)$$

Il est généralement possible de déterminer le seuil d'erreur tolérée expérimentalement. En perturbant les données, on calcule le modèle et on mesure l'erreur. Le seuil peut être pris égal à l'écart-type des erreurs. En supposant une distribution d'erreurs, ce seuil correspond à une certaine probabilité. Il dépend donc aussi de la distribution supposée des erreurs, mais l'hypothèse de la distribution gaussienne des erreurs est habituellement suffisante.

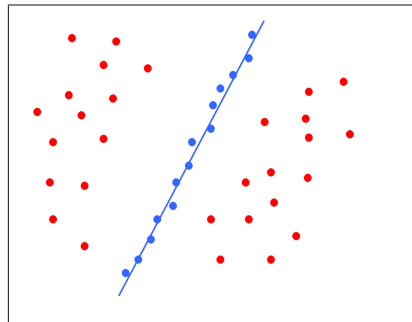


FIG. 3.2 – Ajustement d'une ligne avec les données correctes en utilisant RANSAC.

Une description générique de l'algorithme RANSAC est illustrée sur la figure 3.3. L'algorithme choisit d'une manière aléatoire des échantillons d'appariements de primitives du modèle et des données, qui sont utilisés pour calculer une transformation robuste. Cette transformation est immédiatement évaluée sur le reste des primitives. Si elle donne un consensus satisfaisant, l'algorithme termine, sinon, l'échantillonnage continue jusqu'à trouver une transformation satisfaisante, ou bien il n'arrive pas à ajuster le modèle s'il atteint le nombre maximal d'itérations.

L'algorithme RANSAC est employé dans différentes applications, néanmoins, ses trois domaines de prédilection sont :

- La stéréovision : pour rechercher les appariements corrects et lever les ambiguïtés sur les contraintes épipolaires.
- La reconnaissance d'objets : en calculant des alignements des points par une projection perspective en utilisant un nombre minimum de correspondances entre les primitives du modèle et celles des données observées dans l'image.
- L'analyse de séquence d'images : par le calcul d'une matrice fondamentale robuste, la méthode RANSAC permet la gestion d'occultations des primitives dans l'image, et surmonte les problèmes liés aux mouvements aléatoires et brusques des objets dans les séquences d'images.

```

entrées:
données - ensemble des points observés
modèle - modèle qui ajuste les points
N - le nombre minimum de données qui ajustent le modèle
K - le nombre maximum d'itération de l'algorithme
T - le seuil permettant de déterminer quand est ce que les données ajustent le modèle
D - le nombre de valeurs requises pour confirmer que le modèle ajuste bien les données

sorties:
ajustement - les paramètres du modèle qui donnent le meilleur ajustement des données
              (non_ajustement si aucun modèle n'est trouvé)

itérations      := 0
ajustement      := non_ajustement
meilleure_erreur := infini

tant que itérations < K
    peut_être_inlier := sélection aléatoire de N données
    peut_être_modèle := modèle ajusté à partir des données peut_être_inlier
    aussi_inlier     := ensemble vide

    pour chaque point de données qui n'est pas dans peut_être_inlier
        si le point ajuste le modèle avec une erreur < T rajouter le point à
            aussi_inlier
        fin si
        si le nombre d'éléments dans aussi_inlier est > D (cela implique qu'on a peut
            être trouvé le bon ajustement du modèle)
            meilleur_modèle := paramètres du modèle ajusté pour tous les points dans
                peut_être_inlier et aussi_inlier
            cette_erreur     := mesure de combien le modèle ajuste ces points
                si cette_erreur < meilleure_erreur
                    meilleur_ajustement := meilleur_modèle
                    meilleur_erreur     := cette_erreur
                fin si
            fin si
        fin pour
        incrémenter les itérations
fin tant que
retourner meilleur_ajustement

```

FIG. 3.3 – L'algorithme générique RANSAC.

3.6 Gestion d'occultations en réalité augmentée

Quand les objets virtuels et réels coexistent ensemble dans un monde réel, ils peuvent parfois être occultés par des éléments de la scène. Plusieurs approches ont été proposées dans la littérature pour résoudre ce problème, la vision stéréoscopique a souvent été utilisée pour estimer la profondeur des objets réels et gérer le problème d'occultations en utilisant les informations 3D permettant d'avoir une vue globale de la scène.

Wloka et Anderson [164] ont utilisé cette approche et l'ont validée dans plusieurs applications de réalité augmentée. L'utilisation de la vision monoculaire de la scène est peu réaliste et ne permet pas l'interaction de l'utilisateur avec son monde. Un algorithme de vision stéréoscopique a été alors développé, ceci exige deux caméras alignées de façon à ce que les lignes épipolaires soient parallèles aux lignes scannées (scan-lines). Les images sont échantillonnées et les pixels dans les deux images sont appariés par rapport à leur intensité et leur position. Les auteurs ont implémenté cet algorithme en imposant des contraintes afin de pouvoir reconstruire les objets occultés.

Breen et al. [22] ont traité le problème d'occultations dans des scènes statiques. Une carte de profondeur polygonale est générée pour chaque image et selon une pose donnée. Les auteurs enregistrent la carte 3D de l'environnement afin de recalculer à tout moment les objets virtuels sur le monde réel même si une ou plusieurs parties des objets ne sont pas visibles.

De récents travaux pour gérer les occultations avec la vision stéréoscopique ont été présentés par Kanbara et al. [78]. Leur approche diffère des approches précédentes en calculant la profondeur uniquement à partir des contours des objets virtuels, ce qui permet de garder les performances en terme de temps de calcul. Les auteurs ont utilisé des marqueurs dans l'image et un algorithme de mise en correspondance des points caractéristiques de la stéréovision. Les pixels du contour sont appariés en utilisant un calcul de distance entre les deux zones d'intérêts dans les images et en réalisant une interpolation de pixels.

Kiyokawa et al. [85] ont employé un système composé de 5 caméras pour générer une carte d'occultation dans une application de jeu. Le système a une résolution de $280 \times 240 \text{ pixels}$ et fonctionne avec une fréquence de 30 Hz . Le système requiert des caméras synchronisées fournissant un affichage global de l'environnement. Les résultats obtenus avec cette méthode ont présenté des limitations liées à l'affichage comme illustré sur la figure 3.4.

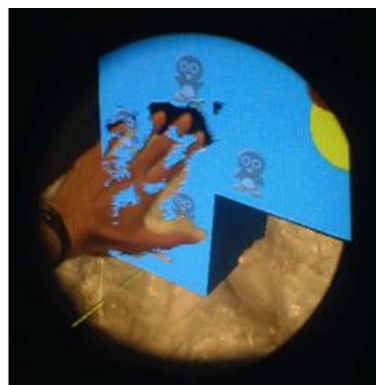


FIG. 3.4 – Système de gestion d'occultation de Kiyokawa et al. [85].

Berger [17] a développé une approche monoculaire pour la gestion d'occultations. L'auteur n'a pas estimé la profondeur mais a analysé le mouvement des contours au cours du temps, ces contours ont été libellés selon leurs positions : devant ou derrière les graphiques virtuels. Pour pallier les problèmes d'occultations, les contours actifs ont été employés. Le contour actif est initialisé autour du contour de l'objet à traquer et se déforme pour entourer l'objet d'intérêt. Avec cette méthode, même si une partie de l'objet est occultée, sa partie visible est suivie pour le recalage. De bons résultats étaient obtenus

avec des scènes non texturées (figure 3.5), cependant, les performances calculatoires de l'algorithme n'ont pas été évaluées.



FIG. 3.5 – Approche de gestion d'occultations en utilisant les contours [17].

Lepetit et Berger [90] ont amélioré la précision de la méthode de Berger [17] en temps différé en traquant des contours occultés dans une séquence vidéo. La reconstruction 3D des objets occultés est effectuée aussi. La précision de la segmentation a permis le recalage correct des objets virtuels.

Des travaux récents se sont focalisés sur l'occultation des objets virtuels en mouvements. Stricker et Klinker [149] ont présenté une application interactive pour pallier le problème. Dans l'application, l'utilisateur met un objet réel dans une scène virtuelle. Les occultations sont gérées en se localisant par rapport à la main de l'utilisateur et en soustrayant l'arrière plan, l'espace détecté ne faisant pas partie de l'arrière plan. Cette approche est faisable dans le cas d'arrière plans homogènes avec l'hypothèse que la caméra soit immobile (figure 3.6).

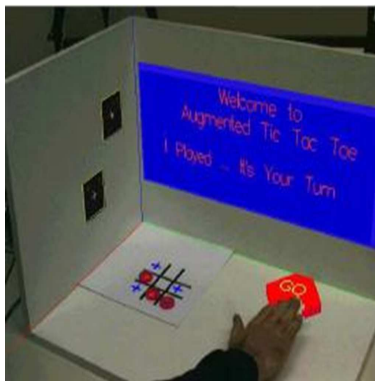


FIG. 3.6 – Occultation des objets virtuels en mouvements [149].

Fuhrmann et al. [53] se sont intéressés à l'occultation des graphiques virtuels par l'opérateur, des marqueurs posés sur l'utilisateur permettent de capturer et d'enregistrer son mouvement dans une tâche collaborative. Un modèle 3D d'humanoïde est aligné au capteur de mouvement, le rendu est calculé avec Z-buffer pour pallier les occultations des objets virtuels dans une application collaborative. Le dispositif permet de suivre les objets qui présentent une certaine densité de probabilité de la partie visible de la main. Les résultats de cette méthode sont montrés sur la figure 3.7.

Fischer et Baratoff [48] ont amélioré la technique de Stricker concernant la soustraction de l'arrière plan. Au lieu d'un simple arrière plan immobile, un modèle 3D texturé de la scène est utilisé. Le rendu est calculé pour chaque pose de la caméra obtenue à partir d'un suivi de cibles avec ARToolKit. Le rendu dans cette méthode est comparé avec une vidéo pré-enregistrée pour déterminer les endroits occultés. Une fois la comparaison faite, l'alignement obtenu entre la scène augmentée et la scène

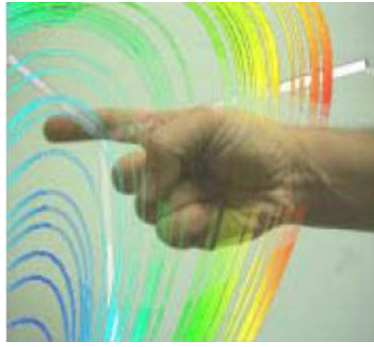


FIG. 3.7 – Occultation des graphiques virtuels par l'opérateur [53].

réelle est très précis. La scène est divisée en deux surfaces texturées et la position de chaque surface est affinée en alignant les points caractéristiques pré-calculés. La gestion d'occultations se fait par comparaison entre les pixels et en employant un critère pour minimiser l'influence du changement de luminosité. La complexité de l'algorithme diminue la rapidité, de plus, la soustraction de l'arrière plan présente des erreurs occasionnelles d'affichage comme illustré sur la figure 3.8.



FIG. 3.8 – Gestion d'occultations avec la technique de Fischer et Baratoff [48].

Mulder [117] a employé aussi la technique de soustraction d'arrière plan pour déterminer les objets occultés. Plusieurs caméras sont utilisées pour observer la main de l'utilisateur, les images résultantes sont binarisées et des masques sont employés pour déterminer les occultations par discrétisation des intervalles de profondeur en considérant plusieurs plans perpendiculaires dans la direction du regard de l'utilisateur. Pour chaque plan, tous les masques binaires des caméras sont projetés ce qui provoque un chevauchement des masques d'occultation. L'occultation de l'objet à cette profondeur existe et les parties virtuelles occultées sont reconstruites. L'algorithme utilise une combinaison de Stencil buffer et Z-buffer pour accomplir la procédure et déterminer la position des objets réels occultés par des objets virtuels.

Comport et al. [29] ont traité le problème de précision et de robustesse du recalage dans les systèmes de réalité augmentée en temps réel. Pour aborder l'aspect temps réel, un algorithme de tracking en vision monoculaire pour un casque semi-transparent de réalité augmentée a été proposé. L'estimation de la pose est basée sur l'asservissement visuel virtuel et des matrices d'interaction ont été utilisées pour détecter les différentes primitives géométriques telles que les lignes, les cercles, les cylindres et les sphères. L'algorithme de tracking réalise un suivi de points de contours pour satisfaire la contrainte du temps réel. La méthode élaborée combine les incertitudes locales sur les positions et les incertitudes globales de la pose. La robustesse est obtenue par l'intégration d'un M-estimateur dans un asservissement visuel via une implémentation itérative de moindres carrés pondérés. Les résultats obtenus sont robustes aux occultations et aux changements de luminosité (figure 3.9).



FIG. 3.9 – Suivi d'un objet partiellement occulté dans une application de maintenance [29].

Dans [30], les auteurs ont utilisé les principes de l'asservissement visuel virtuel. Dans ce contexte, des matrices d'interaction sont déterminées afin de minimiser la distance entre des points locaux et les contours auxquels ils appartiennent. Un suivi local en temps réel est obtenu par l'utilisation de la méthode des Éléments de Contour en Mouvement (ECM), qui effectue un suivi sur la normale du contour. La robustesse est assurée en intégrant une M-estimation dans la loi de commande, par une méthode des moindres carrés pondérés itérés. L'approche a été validée sur des applications d'asservissement visuel et de réalité augmentée. Plusieurs séquences d'images complexes sont considérées, y compris des environnements extérieurs. Les résultats obtenus ont montré la robustesse de la méthode face aux occultations, avec des performances de traitement satisfaisantes (figure 3.10).



FIG. 3.10 – Suivi avec occultations multiples [30].

3.7 Notre méthode de suivi robuste

L'objectif de cette partie est de présenter notre méthode robuste de gestion d'occultations basée sur l'algorithme RANSAC pour le suivi des points 2D dans l'image. Le choix de l'algorithme d'estimation robuste est effectué en respectant les contraintes dues aux : temps de calcul, la robustesse malgré les incertitudes sur le modèle, les erreurs de mesure et les changements de l'environnement.

Pour de nombreuses applications, des modèles photométriques et géométriques simples peuvent suffire. La géométrie projective est un outil mathématique très bien adapté à la modélisation de l'environnement et du processus d'acquisition des caméras. Toutefois, dès lors que l'on se confronte à des images réelles (applications robotiques) la modélisation devient inexacte et des algorithmes robustes sont nécessaires. De plus, des mesures aberrantes peuvent surgir lors d'un changement d'éclairage ou en présence d'occultations.

Nous allons maintenant utiliser l'algorithme d'estimation robuste que nous avons décrit précédemment, RANSAC, pour élaborer une méthode robuste de gestion d'occultations. L'objet d'intérêt est représenté par notre modèle de cibles définies dans le chapitre 2. Le suivi des points caractéristiques des cibles est résolu en estimant la matrice d'homographie qui lie les points 2D homologues dans deux images successives. Ces points 2D sont liés par l'homographie suivante :

$$\begin{pmatrix} \lambda u_2 \\ \lambda v_2 \\ \lambda \end{pmatrix} = \underbrace{\begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}}_G \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \quad (3.2)$$

tel que : (u_1, v_1) et (u_2, v_2) sont deux points homologues dans deux images successives et λ est un facteur d'échelle arbitraire. L'estimation de ces paramètres par une technique de moindres carrés donne généralement de très bons résultats même en présence de certaines perturbations. Cependant, dès que des erreurs aberrantes se produisent (comme par exemple une réflexion spéculaire sur la surface de l'objet ou une occultation), l'algorithme de suivi n'est plus capable d'estimer correctement les paramètres de la pose. Il est donc souhaitable d'utiliser une méthode d'estimation robuste.

Une fois l'étape de mise correspondance 2D-3D est établie, la matrice d'homographie d'estimation de la pose est déterminée. Cette matrice, H , de dimension 3×3 est définie à un facteur d'échelle près, par :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (3.3)$$

Les paramètres de l'homographie de l'équation 3.3 sont déterminés par la méthode de Zhang (annexe A). 4 points coplanaires sont suffisants pour résoudre l'équation 3.3.

Nous allons décrire la solution que nous avons proposée pour résoudre le problème d'occultations des cibles. Deux modèles de cibles ont été utilisés pour avoir au moins 4 points d'intérêts afin d'estimer l'homographie qui permet l'estimation de la pose. Le schéma de gestion d'occultations est illustré dans le diagramme de la figure 3.11. L'étape d'identification des cibles permet d'extraire les points d'intérêts initiaux. Cette étape permet dès l'image initiale, de proposer des candidats susceptibles d'être correctement appariés le long de la séquence d'images. Ces points sont ensuite suivis à l'aide de notre méthode robuste qui réalise la mise en correspondance et l'estimation de l'homographie par RANSAC en utilisant des points additionnels détectés par le filtre de Harris. Si le nombre de points caractéristiques est ≥ 4 la pose est estimée et le recalage d'objets virtuels est effectué. Par ailleurs, si cette condition n'est pas vérifiée, la pose de la caméra ne peut pas être déterminée d'où l'échec du recalage. Enfin, une ré-initialisation du tracking est effectuée par re-projection des modèles 3D d'objets sur l'image [100] [101].

L'algorithme de suivi robuste est composé de trois phases : une première phase extrait les points d'intérêts, vient ensuite une phase d'appariements de ces points d'une image à l'autre. Cette mise en correspondance est réalisée en deux temps : une mise en correspondance 2D-2D permettant de suivre les points dans les images, et une autre 3D-2D qui assure l'appariement des couples de points utilisés pour l'estimation de la pose. Finalement, une phase de validation ou de rejet éventuels des appariements est réalisée. Les points d'intérêts constituant les informations visuelles pertinentes à l'élaboration de l'homographie sont retenus, et les mauvais appariements sont éliminés par l'algorithme RANSAC.

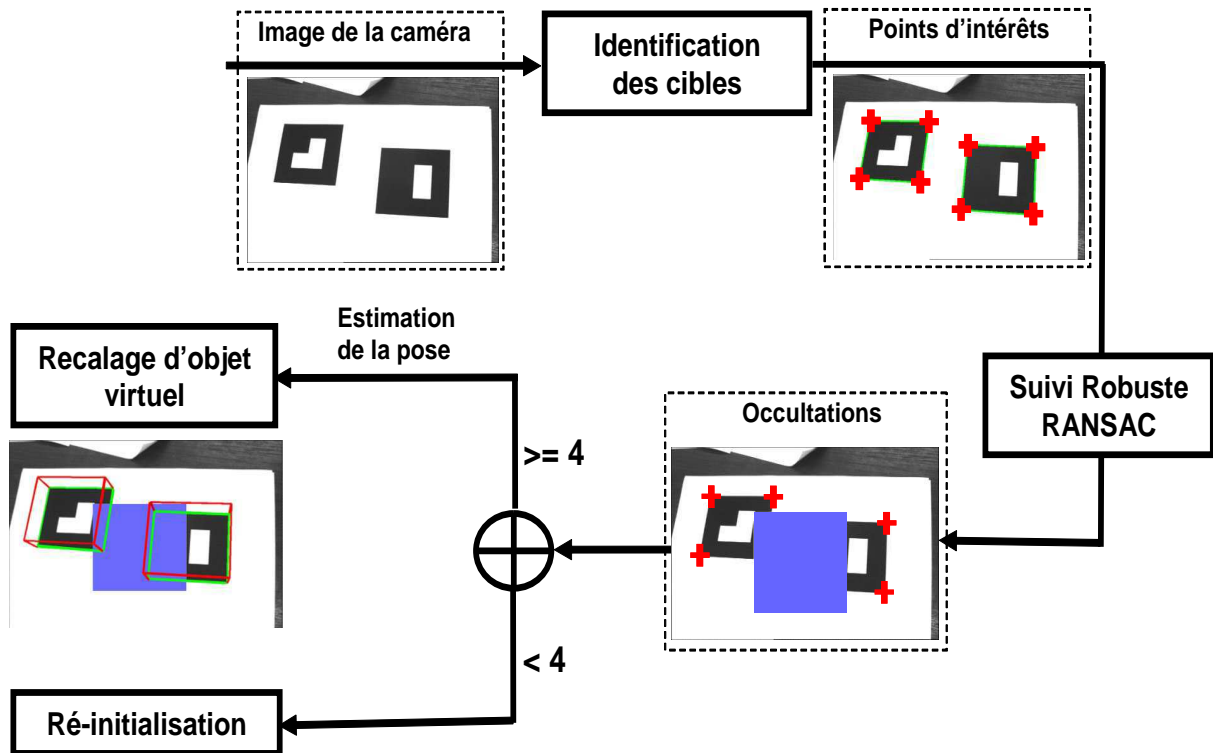


FIG. 3.11 – Schéma synoptique de la gestion d'occultations.

3.7.1 Conditions de fonctionnement

Selon le type d'application, les besoins sont différents et les paramètres varient suivant : la densité des points, leurs vitesses, la qualité de détection, le type de mouvement effectué. Tous ces paramètres doivent être pris en compte par l'algorithme de suivi.

Une attention particulière est accordée aux conditions de fonctionnement de notre méthode car, en effet :

- Le nombre de points d'intérêts suivis varie selon l'application. Par exemple, l'indexation vidéo nécessite une centaine de points pour le suivi, par contre pour un radar, un nombre faible de points est requis afin d'analyser plus rapidement le mouvement des cibles.
- Plus la densité des points est grande, plus on a une grande concurrence dans l'appariement des points, et cela augmente la complexité des traitements et les risques d'erreur.
- La vitesse maximale de déplacement des points dans l'image permet aussi de limiter le nombre de points candidats. De plus, il est important de tenir en compte la vitesse et la densité en fonction de la taille de l'image.
- Les perturbations et les bruits de mesures sont susceptibles de générer des fausses détections temporaires, ceci doit être pris en compte dans l'algorithme de suivi.

3.7.2 Détection des points d'intérêts

Les méthodes d'extraction de points d'intérêts dans l'image sont classées en deux catégories [36] :

- Les méthodes basées contours [11] [18] [107]. Ces méthodes extraient les primitives géométriques telles que les contours, les segments de droite et les courbes. Les coins sont ensuite extraits, ils correspondent aux courbures maximales de ces contours.
- Les méthodes basées sur le signal [65] [115]. Celles-ci, ont pour principe d'extraire directement les points d'intérêts à partir de l'information photométrique.

Pour la détection des points d'intérêts, nous avons utilisé le détecteur de Harris (annexe C). Le filtre de Harris permet de retrouver les points de fortes courbures dans l'image. Ces points sont ensuite appariés dans les séquences d'images et utilisés pour calculer la transformation robuste RANSAC. Les points d'intérêts sont maintenant déterminés (figure 3.12), l'étape suivante consiste à mettre en correspondance ces points dans une succession d'images.

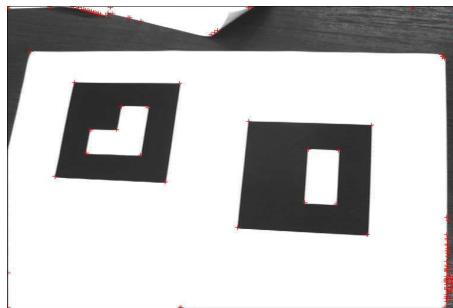


FIG. 3.12 – Détection de points avec le filtre de Harris.

3.7.3 Mise en correspondance 2D-2D

Nous allons à présent nous intéresser à la méthode d'appariement d'indices 2D. Gottesfeld [58] a présenté un état de l'art sur les méthodes d'appariements et les principales approches de mise en correspondance d'images 2D dont la classification peut être faite selon différents critères. Ces méthodes sont classées en quatre catégories principales :

- méthodes basées sur la corrélation,
- méthodes basées sur la transformée de Fourier,
- méthodes basées sur la correspondance de points,
- méthodes basées sur un modèle élastique (Elastic Model-Based Matching).

Dans notre étude, pour la mise en correspondance des points caractéristiques détectés par l'algorithme de Harris dans deux images successives, nous avons utilisé l'appariement par corrélation qui permet de rechercher les zones de ressemblance dans deux images successives.

Cette méthode utilise une mesure de la corrélation sur une fenêtre autour du point à mettre en correspondance. La mesure de la corrélation est donnée par l'équation suivante :

$$R_{uv} = \frac{\sum_x \sum_y w(x, y) I(x - u, y - v)}{\sqrt{\sum_x \sum_y I^2(x - u, y - v)}} \quad (3.4)$$

w est la fenêtre utilisée pour le calcul de la corrélation au point (x, y) considéré avec les points (u, v) , I est l'intensité de l'image à traiter.

La corrélation mesure la ressemblance de 2 pixels dans 2 images, un détail d'une image est reconnaissable grâce aux variations de son voisinage, on considère souvent des voisinages de forme rectangulaire ou carrée. Des fenêtres carrées sont généralement utilisées, leurs tailles sont relativement empiriques $3 \times 3 \text{ pixels}^2$ à $9 \times 9 \text{ pixels}^2$. Mesurer la ressemblance entre un pixel p_1 de l'image 1 et un pixel p_2 de l'image 2 revient donc à évaluer la ressemblance de 2 fenêtres contenant les deux pixels. On définit pour cela l'indice de ressemblance des fenêtres par :

- La distance $d_{ab} = |a - b|$.
- Le coefficient de corrélation linéaire (équation 3.4).

Une grande différence d'échelle ou d'orientation entre les imagerie ou une trop forte distorsion due au relief dégradent la mesure de ressemblance. Dans les méthodes itératives, on a donc intérêt à corriger géométriquement l'image de recherche en utilisant la parallaxe calculée à l'itération précédente. Cela améliore la ressemblance des imagerie de recherche par rapport à la référence, et la corrélation est calculée plus facilement. La réalisation d'un appariement d'imagerie permet de rechercher les points homologues suivant la méthode suivante :

1. On choisit une image de référence (image 1) et on se fixe une fenêtre centrée sur le pixel de l'objet d'intérêt.
2. On explore alors la seconde image en déplaçant la fenêtre de recherche et en calculant pour chaque position l'indice de ressemblance.
3. Le point homologue est celui pour lequel l'indice de ressemblance est le plus élevé (dans le cas le plus fréquent de la corrélation linéaire, proche de 1)

3.7.4 Mise en correspondance 3D-2D

Plusieurs méthodes peuvent être utilisées pour réaliser la mise en correspondance 3D-2D des points. La plupart de ces méthodes projettent les points 3D du modèle en employant une transformation pour passer dans le repère des points observés dans l'image. Le processus de vérification consiste à examiner si les primitives du modèle transformées par projection coïncident avec des points observés dans l'image.

Notre méthode de mise en correspondance 3D-2D est dépendante de l'estimation de la pose de la caméra. Les points 3D du modèle d'objet sont appariés avec leurs projections 2D dans l'image par l'utilisation de la matrice de projection. Cette mise en correspondance est réalisée initialement d'une manière approximative à une rotation près. Toutefois, lors de l'estimation de la première pose de la caméra, les points 3D sont appariés à leurs véritables homologues 2D.

Cette méthode présente deux avantages et qui sont :

- Au lieu d'explorer toutes les possibilités de mise en correspondance des données observées dans l'image et les points 3D du modèle, l'utilisation de la projection perspective de la caméra permet de retenir seulement les primitives du modèle correspondant à celles des données de l'image.
- L'assignation de primitives 2D à celles du modèle satisfait la contrainte de mise en correspondance. Cette méthode retrouve directement les bons appariements, ce qui réduit le nombre et le temps de recherche des candidats.

3.7.5 Mise en correspondance 3D-2D et 2D-2D

Nous avons vu dans les deux sections précédentes, les techniques de mise en correspondance que nous avons employées pour faire les appariements 2D-2D et 3D-2D des points. En effet, notre algorithme de suivi de points d'intérêts a besoin des points du modèle et leurs correspondances dans l'image à chaque instant. Cette étape est primordiale afin de réaliser un recalage dynamique sur les cibles. Les mises en correspondances entre les points 2D-2D dans la séquence d'images sont retrouvées par la méthode de corrélation (figure 3.13-b) et l'appariement 3D-2D est effectué par projection

des points du modèle sur l'image (figure 3.13-a). Cette étape de mise en correspondance 3D-2D et 2D-2D est très importante dans notre processus de suivi de points car elle permet d'identifier des différents attributs 2D et 3D soumis à une certaine relation projective. Le but de cette partie est d'obtenir une description géométrique entre les deux espaces image et modèle pour calculer la pose de la caméra.

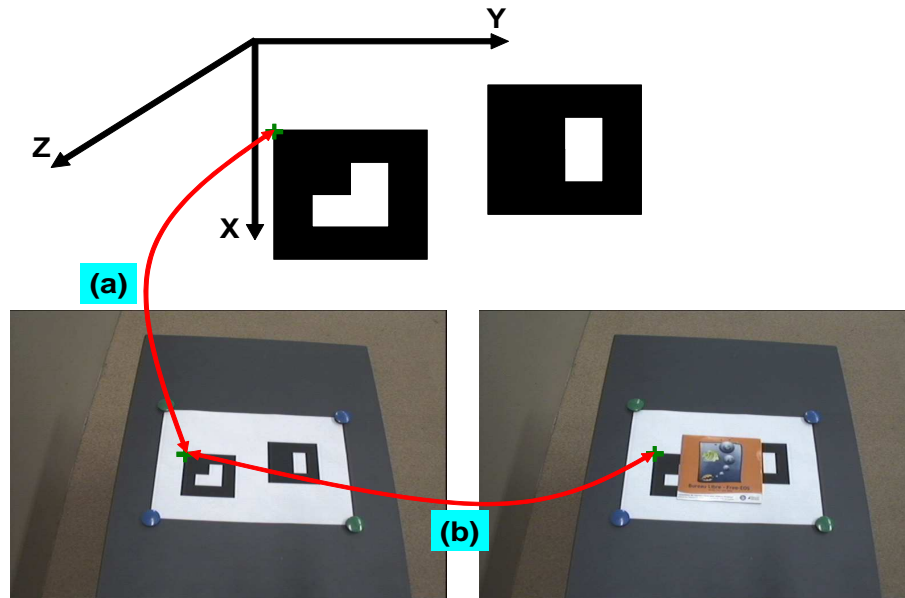


FIG. 3.13 – Mise en correspondance des points. (a) Appariements 3D-2D. (b) Appariements 2D-2D.

3.7.6 Mise en correspondance robuste

La méthode robuste de mise en correspondance à base de l'algorithme RANSAC, utilise les appariements 2D-2D des points homologues dans deux images successives. L'établissement de correspondances robustes de ces points d'intérêts est effectué par une estimation de l'homographie déterminée par RANSAC. Cette matrice homographie représente une contrainte géométrique entre les deux images. La détermination de cette matrice permet de retrouver la transformation qui relie les primitives d'une image à une autre. Les points caractéristiques susceptibles de représenter un bon appariement sont validés, tandis que, les faux candidats sont rejetés et éliminés par RANSAC.

3.7.7 Résumé de la méthode de suivi robuste

En résumé, la méthode que nous avons développée pour la mise en correspondance robuste des points, est constituée des étapes suivantes :

1. Extraction des points d'intérêts correspondants à chaque image de la séquence en utilisant le détecteur de Harris.
2. Établissement d'un ensemble de correspondances en utilisant la technique de corrélation. Cela permet de trouver les candidats à l'appariement pour chaque point et de construire de manière symétrique, des fenêtres de recherche dans le voisinage des points pour lever les ambiguïtés d'appariement.
3. Élimination des faux appariements par l'algorithme RANSAC : parmi les correspondances établies dans l'étape précédente, on trouve souvent de faux appariements. Ceci est principalement dû au fait que l'on a utilisé des contraintes heuristiques. Nous avons appliqué la méthode robuste

basée sur la technique RANSAC pour éliminer les faux appariements. La méthode développée consiste à estimer une homographie en cherchant dans l'espace de paramétrisation les points qui ajustent la transformation robuste.

4. Établissement d'un nouvel ensemble de correspondances : une fois l'homographie est estimée de manière précise, une nouvelle itération du processus de mise en correspondance est appliquée entre les deux images pour optimiser le résultat.
5. Mise en correspondance 3D-2D : ceci permet d'apparier les points 3D des modèles d'objets avec les points 2D des cibles dans l'image. Cette étape permet l'estimation de la pose de la caméra et le recalage virtuel.

3.8 Résultats expérimentaux

La validation de notre système de suivi robuste basé sur l'algorithme RANSAC nécessite une évaluation exhaustive pour tester son efficacité et sa robustesse dans différentes situations. Cela inclut une étude du comportement de la méthode en présence d'occultations partielles dans des conditions variables de l'environnement.

L'évaluation que nous proposons est réalisée en environnement réel. Dans ces conditions, nous commençons par l'étape de reconnaissance des cibles pour amorcer le tracking. Ensuite, ces mêmes cibles subissent des occultations partielles. Nous varions après, la luminosité, l'orientation et l'échelle. Les conditions expérimentales dans lesquelles les expérimentations sont réalisées, sont décrites dans la section suivante.

Remarque :

Notre méthode de suivi robuste a été validée dans un premier temps avec un programme sous Matlab. Pour le suivi robuste avec RANSAC, nous avons calculé la matrice homographie robuste entre les points appariés dans une succession d'images. Cependant, dans cette partie expérimentale, l'implémentation de RANSAC est effectuée en C++ en utilisant la librairie OpenCV. Une fonction dédiée permet le calcul la matrice fondamentale de RANSAC et non l'homographie.

3.8.1 Conditions expérimentales

L'implémentation de notre technique de suivi robuste est effectuée sur un PC possédant un processeur Intel Xeon 2.4 GHz et 1 GO de RAM. Nous avons utilisé une caméra CCD bâton de type Sony XC-555P avec une carte d'acquisition Meteor 2 standard. Les images sont en RGB avec une résolution $768 \times 576 \text{ pixel}^2$ et une focale de 6mm.

Dans les expériences, des séquences d'images ont été acquises en utilisant la caméra. Le suivi est effectué en temps réel avec une fréquence d'acquisition d'images égale à 25Hz. Dans de telles expériences, le traitement d'image est complexe. En effet, l'extraction et le suivi fiables des points dans un environnement réel est souvent difficile. Par conséquent, un réglage adéquat des paramètres de la méthode est primordial pour assurer un fonctionnement optimal.

3.8.2 L'algorithme

L'algorithme de suivi robuste que nous proposons est composé de plusieurs étapes (figure 3.14). Tout d'abord, une extraction des points caractéristiques dans l'image est réalisée par le filtre de Harris. Un point $(u_i, v_i)_t$ de l'image à l'instant t correspond à un point $(u_j, v_j)_{t+1}$ de l'image $t + 1$. Cela étant effectué par corrélation qui permet de calculer la disparité et génère des appariements putatifs entre les points caractéristiques. Ces points sont détectés dans deux images successives en calculant le coefficient de corrélation entre les points suivant un masque défini, w , exprimé en pixels.

La troisième étape consiste à déterminer la matrice fondamentale robuste reliant les appariements 2D-2D dans les images. Le suivi robuste de ces points caractéristiques est effectué avec RANSAC qui reçoit en entrée un ensemble de points appariés et calcule la distance de ces points par rapport au modèle de données. Si la distance entre les points et le modèle est inférieure à un certain seuil, alors les points correspondants sont considérés comme des bons appariements (inliers).

Ensuite, une mise en correspondance 3D-2D des points du modèle d'objet et leurs projections dans l'image est établie. Cette étape réalise l'appariement des couples de points 3D-2D pour calculer la pose de la caméra et permettre le recalage virtuel. Cette mise en correspondance est réalisée lorsque les cibles sont identifiées dans l'image à l'instant $t - 1$. Dans ce cas, les points caractéristiques sont déterminés et appariés avec les points 3D et la pose de la caméra est déterminée. Si un ou plusieurs points sont occultés, l'algorithme robuste permet de suivre les points visibles initialisés avec les sommets des cibles dans l'image à l'instant $t - 1$. Les points caractéristiques sont suivis dans l'image à l'instant t par RANSAC, et la mise en correspondance avec leurs homologues 3D est réalisée par projection des modèles 3D des cibles sur le plan image en utilisant la pose calculée à l'instant $t - 1$ (figure 3.15).

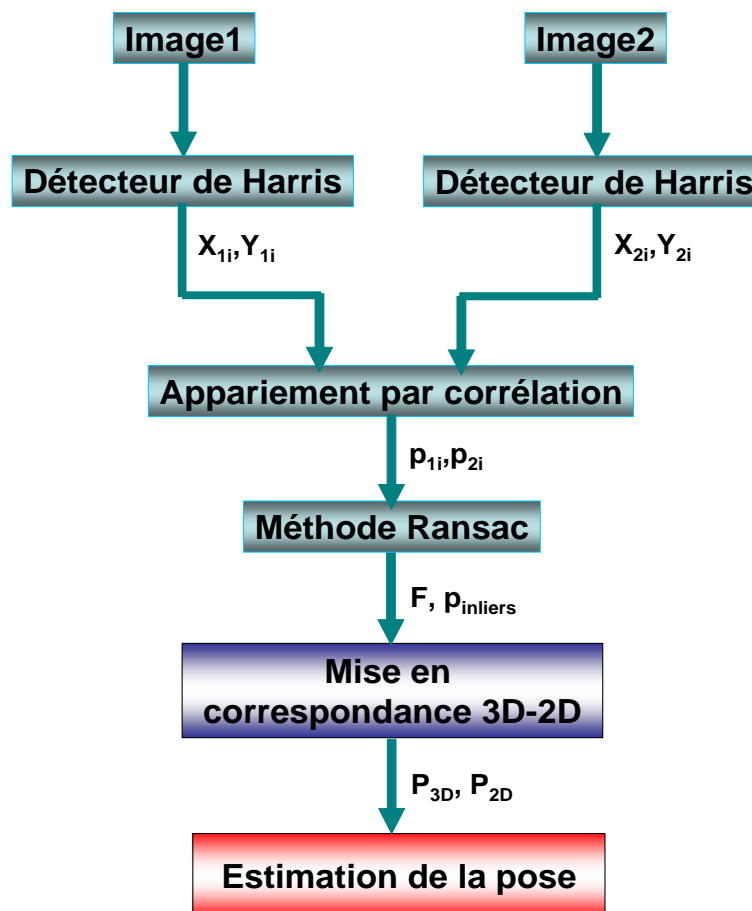


FIG. 3.14 – Fonctionnement de l'algorithme robuste.

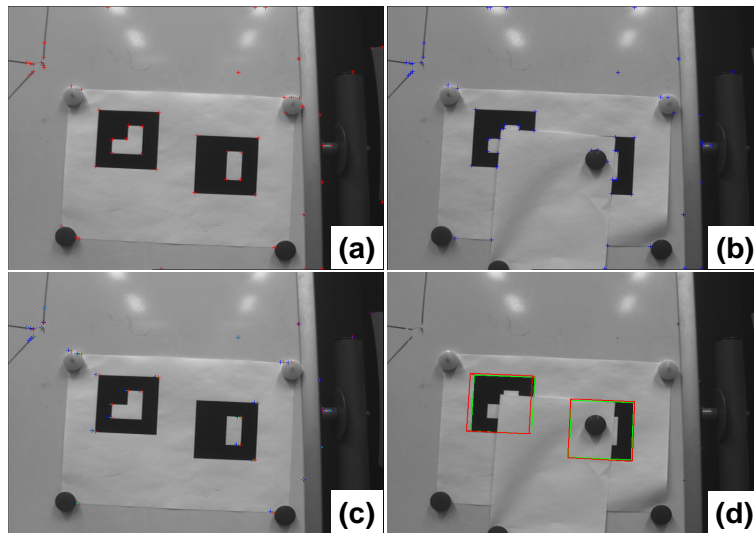


FIG. 3.15 – Étapes de traitement et de suivi de points avec la méthode robuste. (a) Cibles visibles. (b) Cibles partiellement occultées. (c) Appariements des points 2D. (d) Recalage des cubes sur les cibles.

3.8.3 Suivi de points

Nous avons effectué un premier test de suivi en utilisant les deux cibles codées. L'expérience consiste à employer la méthode robuste pour suivre les points d'intérêts au cours du temps selon plusieurs configurations. Cette expérimentation permet de tester l'algorithme robuste face aux occultations des cibles. Nous varions pour chaque test le nombre de points visibles par la caméra afin d'évaluer la robustesse de l'algorithme d'estimation de pose quant au calcul de l'homographie, sachant qu'au minimum 4 points sont nécessaires pour résoudre les paramètres de la pose.

Dans la figure 3.16, on voit les cibles imprimées déposées sur une table et des occultations de points caractéristiques de ces marqueurs sont réalisées. Les résultats obtenus montrent l'efficacité de l'algorithme robuste. Lorsque un point est occulté, la méthode robuste se déclenche et permet l'appariement robuste des points 2D indispensables à l'estimation de la pose et le recalage virtuel.

3.8.4 Occultations partielles

Dans la première expérience nous avons testé notre algorithme robuste dans le cas des occultations partielles des cibles. En effet, nous avons réalisé plusieurs occultations des cibles, les points caractéristiques visibles sont supérieurs à 4 afin de pouvoir calculer la pose de la caméra. On remarque que les deux cibles sont bien suivies et le recalage est maintenu pour différentes occultations. Lorsque les cibles sont visibles, elles sont identifiées ce qui permet d'initialiser les points qui sont suivis par l'algorithme robuste. Si le nombre de ces points 2D est ≥ 4 , l'homographie de la pose est résolue en utilisant les points 3D correspondants. Par contre, si ce nombre est < 4 , l'estimation de la pose ne peut pas être établie et ceci provoque la défaillance de la méthode de suivi robuste (figure 3.17).

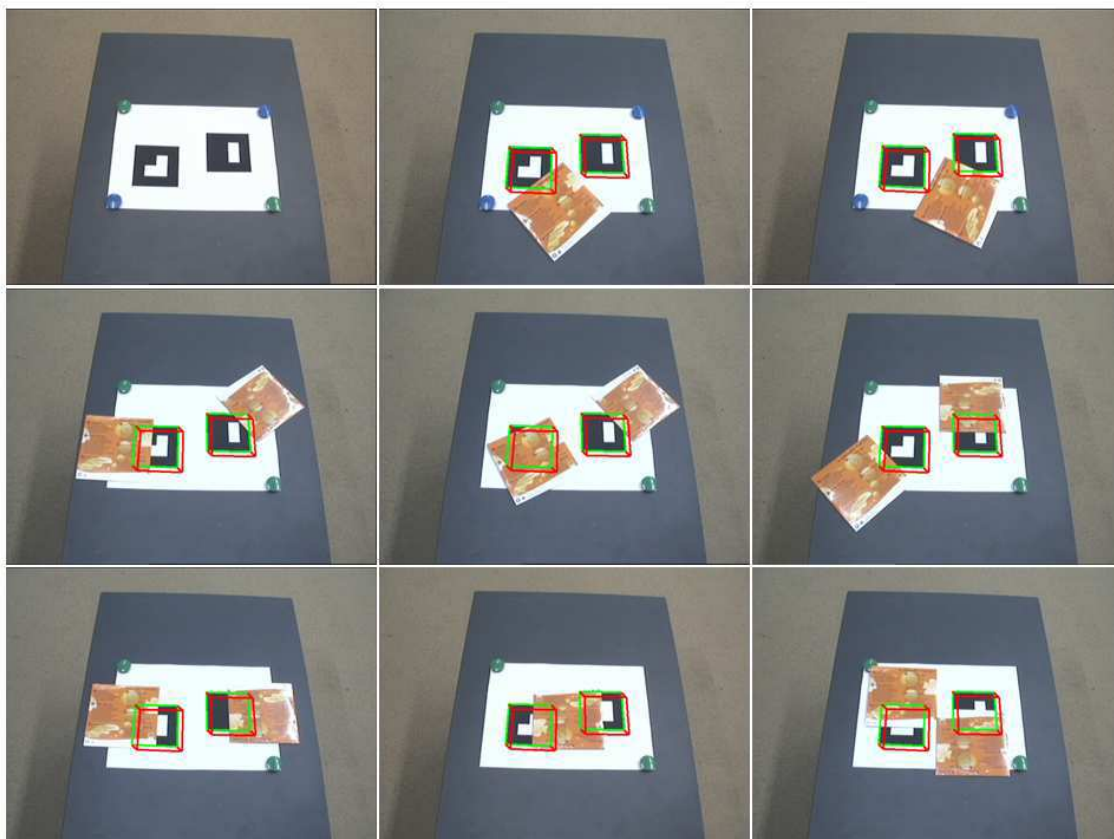


FIG. 3.16 – Suivi de cibles en présence d'occultations partielles.

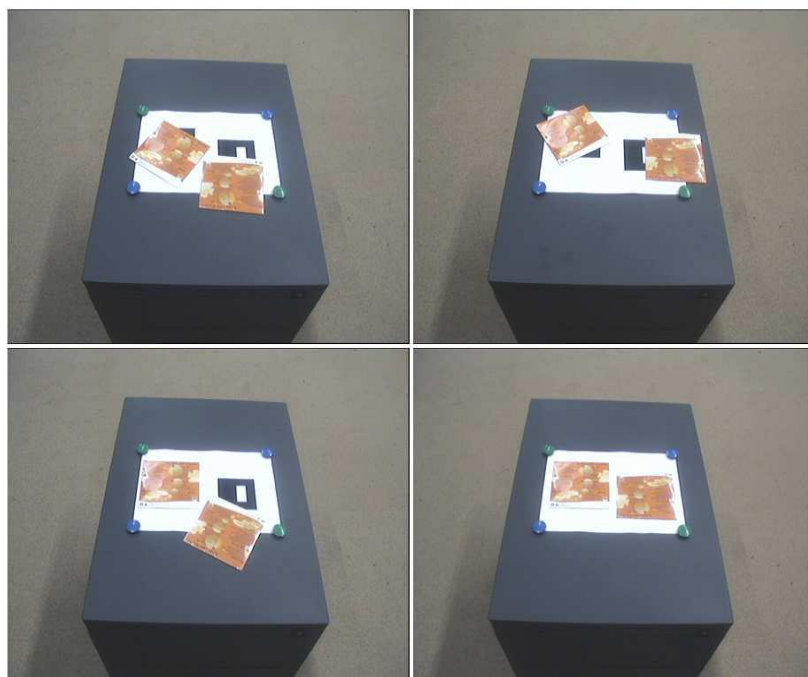


FIG. 3.17 – Échec de suivi de cibles.

3.8.5 Changement de luminosité

Dans ce test relatif aux changements de luminosité, l'objet est suivi malgré les grandes variations d'éclairage (figure 3.18). La lumière est augmentée dans les premières images et diminuée à la fin de la séquence. On remarque que le suivi de points avec l'algorithme robuste est maintenu. Ce test pourrait vérifier à la fois la robustesse de notre algorithme de reconnaissance de cibles ainsi que l'efficacité de notre méthode robuste face aux variations de la source lumineuse. Si le changement d'éclairage est réalisé lorsque les cibles sont toutes les deux visibles par la caméra, alors, on peut analyser au même temps l'évolution de la luminosité pour notre méthode d'identification de cible et la méthode robuste. En effet, l'initialisation des points caractéristiques des cibles est réalisée par la méthode d'identification qui extrait les sommets de la cible. Ces points sont ensuite suivis par l'algorithme robuste au cours du temps. Si les cibles ne sont pas détectées par l'algorithme de reconnaissance, cela conduit forcément à l'échec de l'algorithme robuste. Or, dans cette expérimentation, l'étape de détection a été effectuée dans des conditions normales de luminosité, et donc, la robustesse face aux changements d'éclairage ne dépend que de la méthode de suivi robuste de points d'intérêts.

Dans le cas où la scène comporte une grande zone de luminance, les objets sont peu contrastés par rapport au fond et la détection des points d'intérêts est compromise. Ceci influence sur le nombre de mises en correspondances nécessaires pour déterminer la matrice fondamentale RANSAC d'où la défaillance de l'algorithme de suivi robuste (figure 3.19). De même, pour une faible luminance, les points d'intérêts ne sont plus détectés et ceci entraîne l'échec du système de tracking (figure 3.20).

3.8.6 Changement d'échelle

Dans cette expérimentation, nous nous sommes confrontés à des changements d'échelle des images. Le changement d'échelle est lié aux variations de distances auxquelles les cibles sont capturées. Les résultats obtenus en variant l'échelle sont montrés sur la figure 3.21. Les cibles ont été placées à $0.5m$, $1m$ et $2m$ de la caméra, chaque distance est représentée par une série de trois images dans le sens horizontal. Pour chaque série nous avons varié le nombre de points occultés pour évaluer au même temps la précision de l'algorithme d'estimation de la pose. Nous pouvons dire que pour une distance allant jusqu'à $2m$ l'occultation est toujours gérée par notre algorithme et le recalage est précis malgré le changement de distance et le nombre de points visibles.

Contrairement à l'expérience précédente concernant le changement de luminosité, ce test ne requiert pas l'initialisation des points d'intérêts par la méthode d'identification dans des conditions favorables de l'environnement. En effet, ce test est moins contraignant pour l'algorithme de reconnaissance quant au réglage des paramètres des différents opérateurs utilisés pour le traitement des images de la scène.

Enfin, ce test a permis de vérifier la capacité de notre algorithme à suivre les cibles lors d'un changement d'échelle, en se rapprochant ou s'éloignant des objets d'intérêts. Dans les limites de visibilité des points d'intérêts des cibles, l'algorithme robuste ne rencontre pas de difficultés notables lors de ce test.

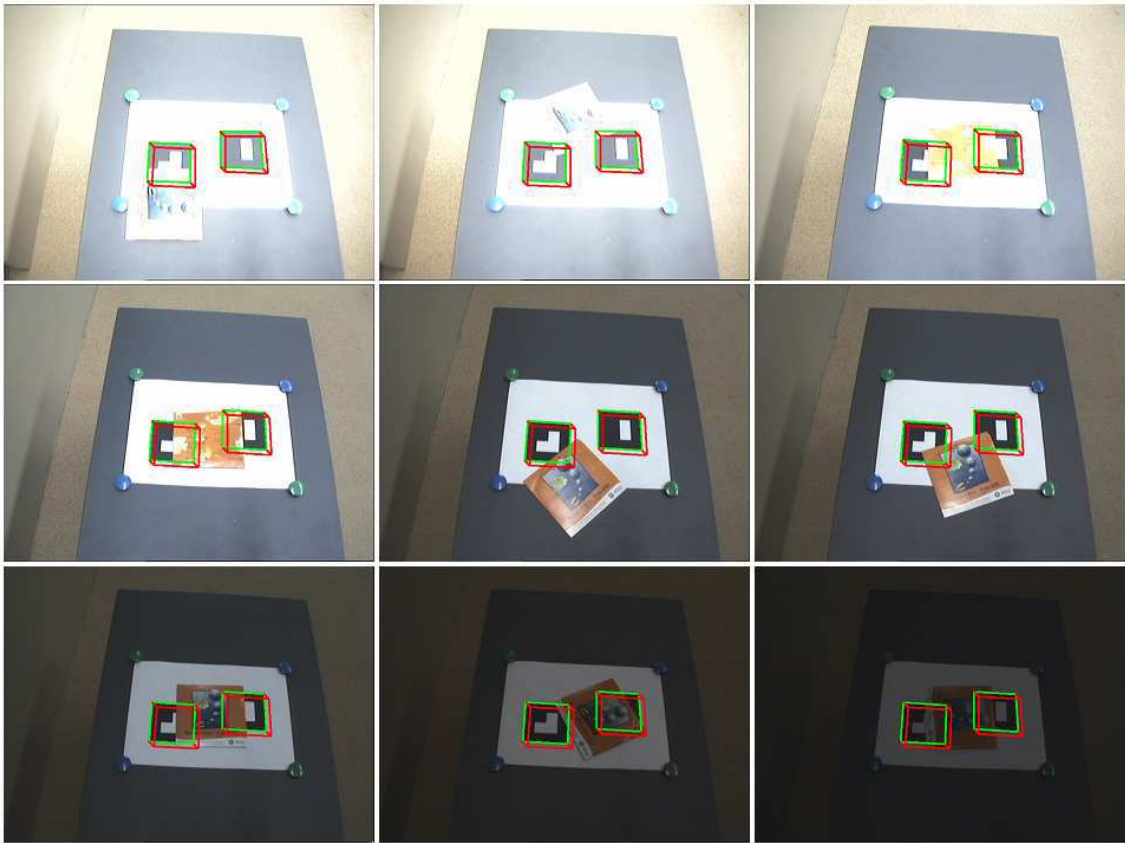


FIG. 3.18 – Changement de luminosité lors d'un suivi robuste.

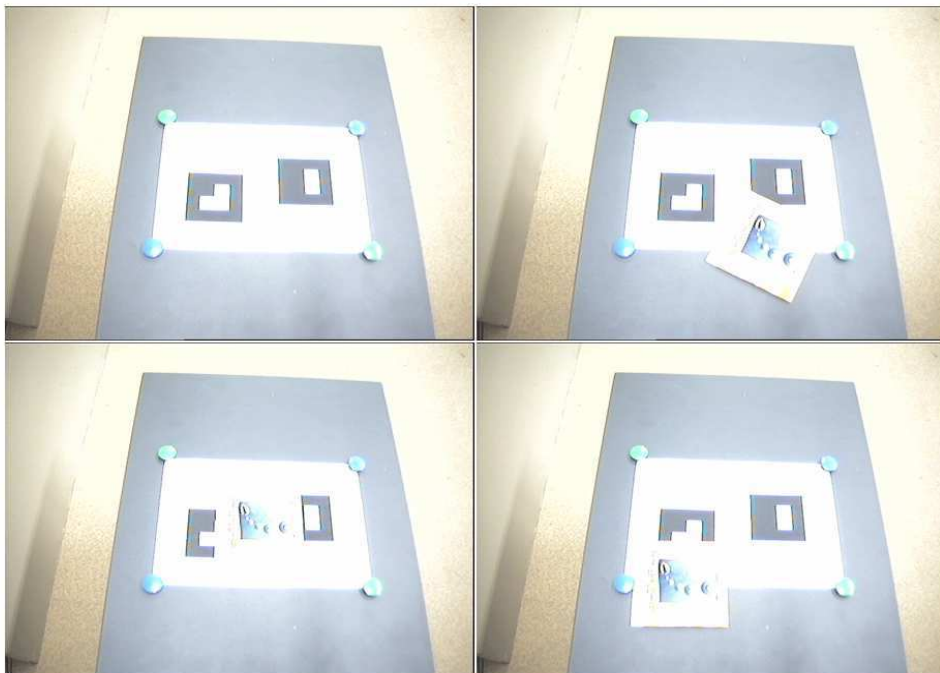


FIG. 3.19 – Suivi robuste en présence d'une forte luminosité.

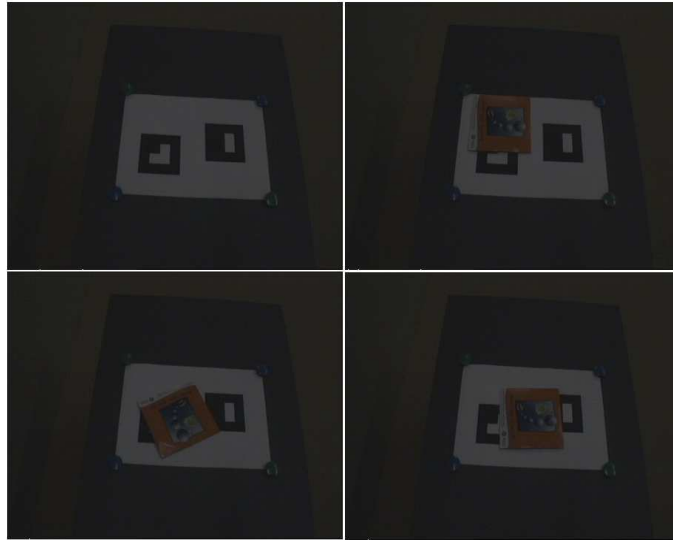


FIG. 3.20 – Suivi robuste avec une forte diminution de luminosité.

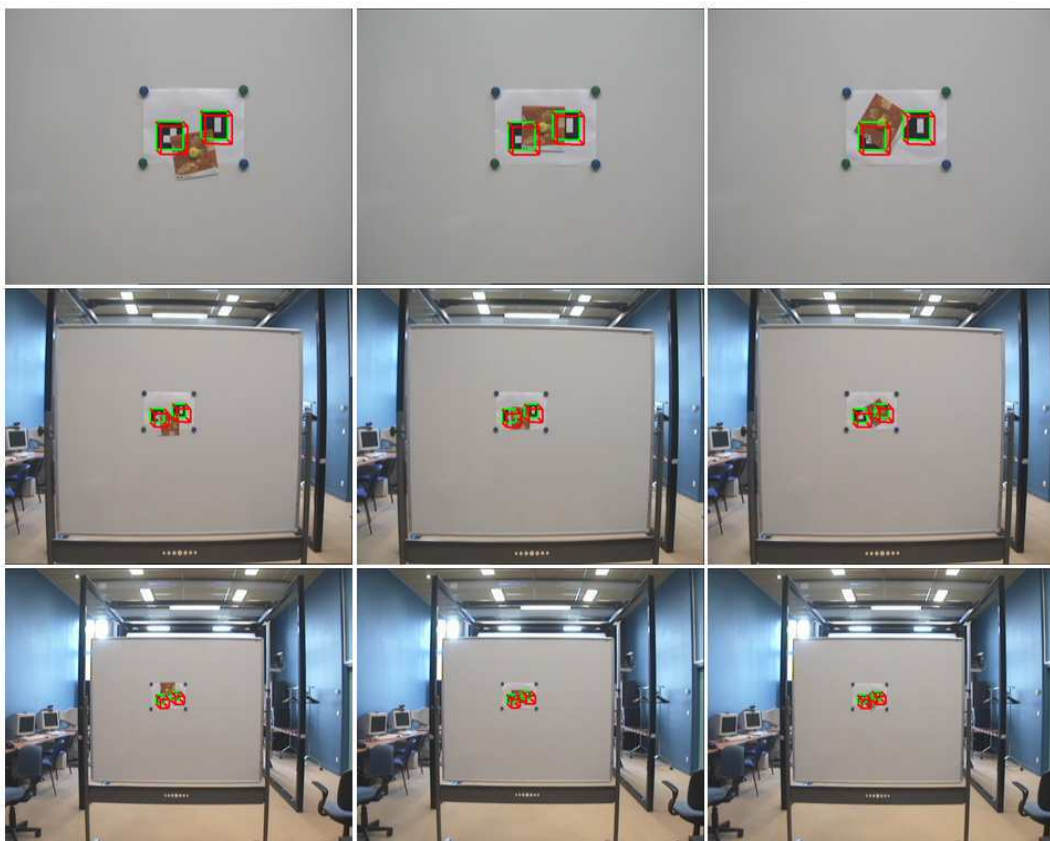


FIG. 3.21 – Changement d'échelle lors d'un suivi robuste.

3.8.7 Changement d'orientation

Le mouvement de la cible ou de la caméra entraîne des changements d'orientation, nous avons introduit dans ce test, ces variations d'orientations. Nous avons testé notre méthode robuste en réalisant diverses orientations des cibles (figure 3.22). L'algorithme de suivi robuste ne prend pas en compte l'orientation de la cible, puisqu'il s'agit d'un suivi de points d'intérêts, le suivi d'un objet effectuant une rotation ne pose donc aucun problème particulier, et les résultats obtenus sont acceptables.

Nous avons aussi testé avec d'autres type d'orientation en réalisant des mouvements de lacet et de roulis de la caméra autour des cibles. Les résultats illustrés sur la figure 3.23 confirme l'aptitude de la méthode à suivre les points caractéristiques sans se soucier de leurs positionnements dans l'espace ou le type d'inclinaison réalisée.

De plus, notre méthode de mise en correspondance 3D-2D et 2D-2D permet de préserver l'ordonnancement des points et par conséquent, la pose est estimée correctement. Cela permet donc, un recalage précis de l'objet virtuel sur les cibles. Enfin, nous concluons que l'orientation de la cible ne constitue pas un empêchement pour le suivi des points d'intérêts et l'algorithme robuste se comporte convenablement dans ce genre de situations.

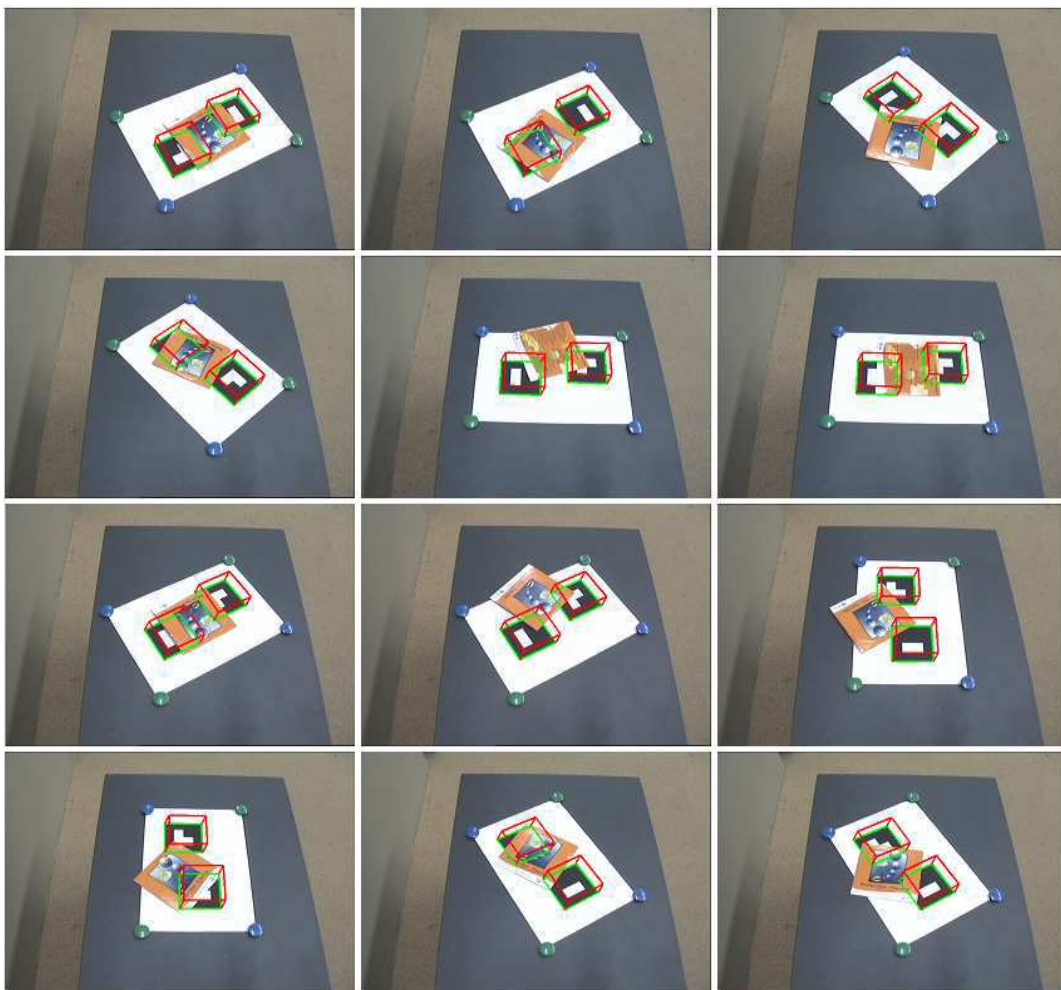


FIG. 3.22 – Changement d'orientation des cibles dans un suivi robuste.

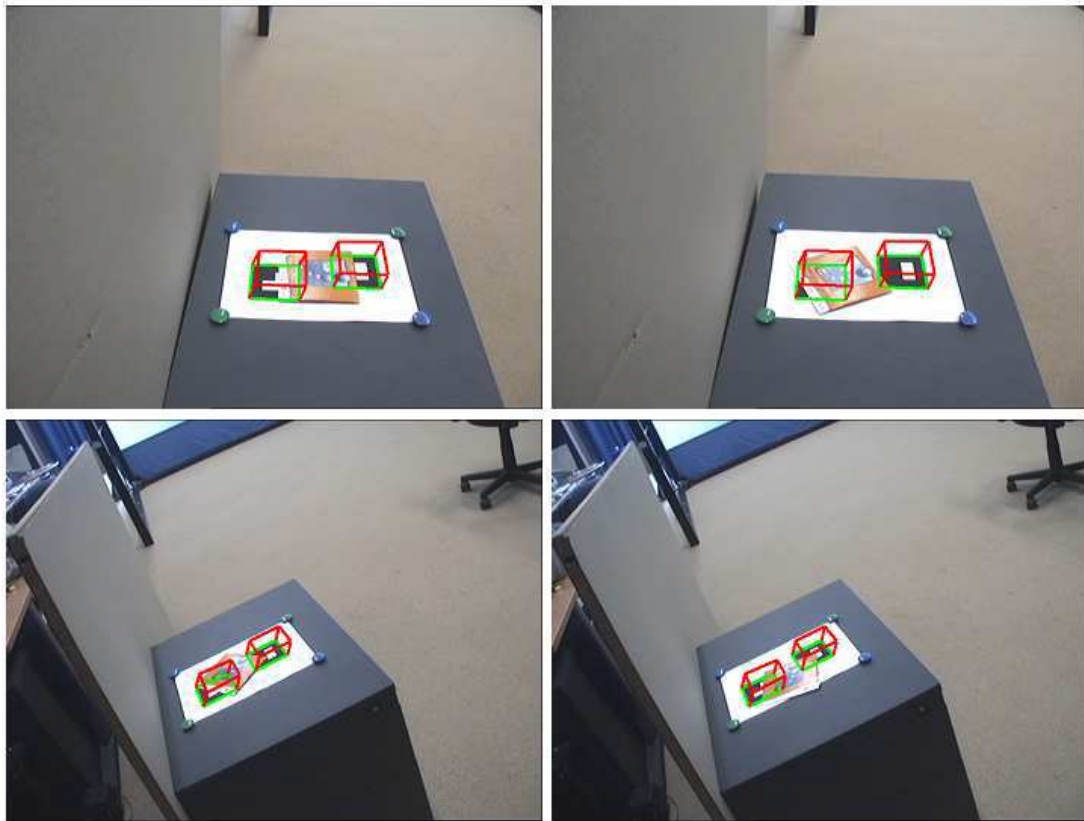


FIG. 3.23 – Changement d’orientation de la caméra dans un suivi robuste.

3.8.8 Temps d’exécution

Le temps d’exécution des différentes procédures développées pour le système de tracking robuste, pénalise le concept temps réel que nous voulons développer. En effet, sur la figure 3.24, on voit que la méthode est coûteuse en terme de temps de calcul par rapport à l’estimation de la pose de la caméra dont le temps d’exécution est estimé à $< 20\mu s$ avec l’algorithme analytique. Le temps de traitement moyen de la méthode robuste est de $37ms$ ce qui représente approximativement le temps d’acquisition d’images par la caméra ($40ms$).

Le tableau 3.1 montre le temps d’exécution de l’algorithme de suivi robuste en fonction du nombre de points visibles. On remarque que plus le nombre de points visibles augmente, plus le temps de traitement de l’algorithme de suivi augmente aussi.

Points visibles	Temps de traitement (ms)
4	35.3637
5	36.1106
6	36.9122
7	37.5684
8	38.2259

TAB. 3.1 – Temps de traitement de l’algorithme robuste en fonction du nombre de points suivis.

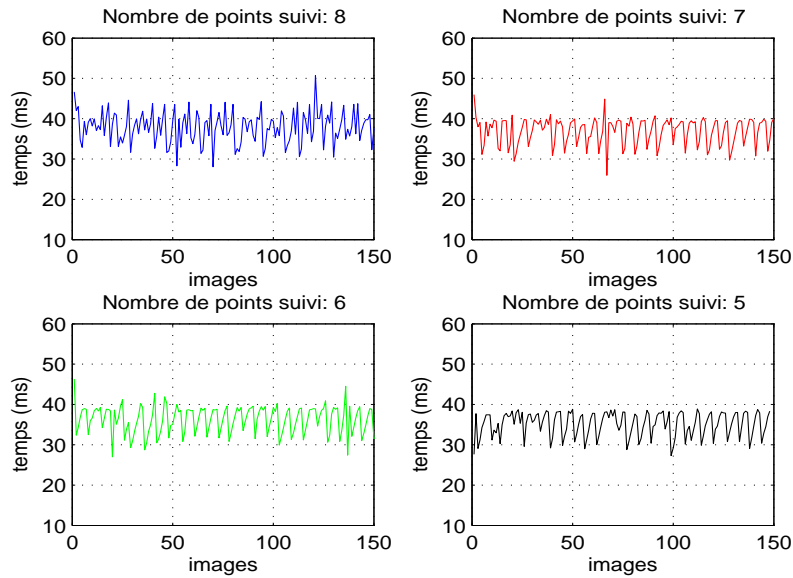


FIG. 3.24 – Temps d'exécution de la méthode robuste en fonction du nombre de points visibles.

3.8.9 Erreur de reconstruction

Nous avons calculé l'erreur de reconstruction des points 2D visibles à partir de l'estimation de la pose de la caméra et le modèle 3D des points. À partir du tableau 3.2, nous remarquons que l'erreur de reconstruction diminue lorsque le nombre de points visibles augmente. L'erreur de reconstruction est déterminée par la formule suivante :

$$err = \frac{1}{N} \sqrt{\sum_{i=1}^N (p_i - \hat{p}_i(M, P_i))^2} \quad (3.5)$$

Cette erreur détermine la précision dans l'estimation des paramètres de la pose et cela se traduit par un recalage précis des objets virtuels sur les cibles. Nous avons utilisé l'algorithme d'estimation de pose de Zhang (annexe A), qui est basé sur la détermination de l'homographie du modèle planaire des cibles et les points 2D correspondants. Nous avons opté pour cet estimateur de pose pour sa rapidité (temps d'exécution est $< 20\mu s$). On peut conclure que la précision dans l'estimation de la pose dépend du nombre de points utilisés et plus ce nombre est important, plus on a une meilleure estimation de la pose et donc un recalage précis des objets 3D.

Points visibles	Erreur de reconstruction (<i>pixel</i>)
4	0.7500
5	0.4140
6	0.3397
7	0.2956
8	0.1724

TAB. 3.2 – Erreur de reconstruction en fonction du nombre de points visibles.

3.8.10 Discussion

Dans ce chapitre, nous avons réalisé des séries de tests permettant d'évaluer l'application de suivi de cibles en présence d'occultations dans diverses conditions d'environnement. Lorsque les marqueurs sont visibles, ils sont suivis par l'algorithme de reconnaissance. Cependant, si un point d'une cible est occulté, celle-ci n'est plus reconnaissable par l'algorithme d'identification, ce qui provoque l'échec du recalage virtuel. Notre algorithme robuste permet alors de gérer ce problème d'occultation, en réalisant un suivi de points caractéristiques des cibles dans les séquences d'images en appariant ces points d'une manière robuste.

Lorsque les cibles ne sont pas identifiées, nous exploitons leurs points caractéristiques visibles, représentés par leurs sommets et nous réalisons des appariements 2D-2D pour le suivi de points dans l'image et 3D-2D afin d'estimer la pose de la caméra et réaliser le recalage virtuel. Nous avons vu aussi que le tracking échoue si les points caractéristiques visibles des cibles sont < 4 . Nous avons pris en compte cette éventualité dans notre méthode et nous avons envisagé dans ce cas précis de redémarrer le tracking.

Les séquences utilisées sont constituées de scènes réelles de l'environnement de l'utilisateur contenant des cibles codées. Nous avons prêté donc, une attention aussi bien à la détection des points caractéristiques des cibles qu'à l'analyse du mouvement et l'appariement robuste des points d'intérêts au cours de la séquence. En effet, nos méthodes d'identification et de suivi robuste sont basées vision, un réglage adaptatif et optimal des opérateurs de traitements d'image constitue un facteur important pour le bon fonctionnement des algorithmes.

Les expérimentations réalisées ont permis l'évaluation de système de tracking robuste dans des conditions variées de l'espace de travail, à savoir : le changement de luminosité, d'échelle, d'orientation et la gestion des occultations partielles des cibles.

Lors de changement de luminosité, l'objet est suivi malgré les variations de la source d'éclairage. Toutefois, dans des conditions extrêmes d'illumination, le suivi de points est compromis. Par ailleurs, l'algorithme robuste a démontré son efficacité dans le test de changement d'échelle. L'expérience réalisée a permis de vérifier la capacité de l'algorithme à suivre les cibles lors de variation de distances. En ce qui concerne le test de changement d'orientation de la cible, celui-ci, n'a pas constitué non plus un empêchement pour le suivi des points d'intérêts et l'algorithme robuste s'est comporté convenablement dans ce test. Enfin, l'inconvénient de la méthode de suivi robuste est le temps de calcul qui est considérable par rapport au temps d'acquisition des images. De plus, ce temps augmente en fonction du nombre de points visibles dans la scène.

Finalement, l'utilisation de la méthode de tracking robuste est nécessaire pour effectuer des tâches de suivi en présence d'occultations en environnement réel. Toutefois, dans un tel contexte particulier, les contraintes dues au réglage optimal des paramètres et au temps de calcul doivent être prises en compte pour assurer, d'une part, la flexibilité quant aux changements imprévus de l'environnement, et d'autre part, une utilisation de l'applications en temps réel.

3.9 Conclusion

Dans ce chapitre, nous avons passé en revue les méthodes d'estimation robuste le plus utilisées en vision par ordinateur. L'utilisation de ces méthodes est nécessaire afin de réaliser des tâches en environnement réel. L'inconvénient majeur avec ces méthodes est le temps de calcul important. Si les techniques de vote (Hough, RANSAC) sont très efficaces, le temps de calcul est souvent trop élevé pour assurer une utilisation des algorithmes de vision à une cadence compatible avec des applications en temps réel. Néanmoins, l'utilisation des estimateurs robustes sous certaines contraintes pourrait faire un bon compromis entre robustesse et efficacité algorithmique.

Nous avons développée une méthode de suivi robuste de points d'intérêts basée sur l'algorithme RANSAC. Cette méthode retrouve les correspondances robustes des points dans deux images successives par estimation de la matrice fondamentale. Cette matrice représente la contrainte géométrique qui lie les deux images. Notre méthode de suivi est constituée d'une première étape d'extraction des points d'intérêts en utilisant le détecteur de Harris. Les points sont ensuite appariés par une technique de corrélation. Enfin, les faux appariements sont éliminés par l'algorithme RANSAC qui cherche dans l'espace de paramétrisation seulement les points qui ajustent la matrice d'homographie robuste.

Un paramétrage standard de notre méthode de suivi robuste a été effectué pour que l'algorithme fonctionne dans la plupart des cas. En ce qui concerne les mouvements particuliers ou bien le changement de luminance et d'échelle, cela étant réalisé dans des limites du paramétrage de l'algorithme et il est nécessaire de faire une adaptation si l'on veut réaliser le tracking dans des conditions extrêmes.

Un autre problème concernant ce système de suivi robuste est le nombre de points visibles au cours de la séquence. En effet, on est contraint à avoir un nombre de points ≥ 4 pour pouvoir calculer la pose la caméra et permettre ainsi le recalage. Cependant, dans un environnement réel, le nombre de points visibles peut changer à tout moment et la contrainte que nous avons fixée n'est pas toujours respectée. Pour pallier le problème, nous allons alors, introduire dans le chapitre suivant un autre capteur qui est la centrale inertielle afin de localiser le système de vision quand les indices visuels sont < 4 .

Chapitre 4

Suivi hybride en réalité augmentée

4.1 Introduction

Afin de maintenir le recalage virtuel à tout instant dans un système de réalité augmentée, la pose de la caméra doit être calculée en permanence. Les systèmes de vision sont mis en échec dès que le nombre de primitives visuelles requis n'est pas suffisant pour résoudre la transformation reliant les points 3D du modèle d'objet et leurs correspondants 2D de l'image.

La localisation avec la méthode de suivi robuste présentée dans le chapitre précédent, exige au minimum 4 points visibles pour déterminer la pose de la caméra. Toutefois, dans un environnement réel, cette hypothèse n'est pas toujours vérifiée. Cela pourrait donc, compromettre l'estimation de la pose de la caméra en utilisant l'algorithme de suivi robuste des points d'intérêts. Dans de telles situations, l'utilisation des systèmes multi-capteurs s'avère inévitable pour garantir le recalage virtuel-réel au cours du temps. Les deux propriétés fondamentales d'un tel système sont, d'une part, la complémentarité des capteurs afin de mesurer des données physiques différentes et donc d'enrichir l'information perçue, et d'autre part, la redondance afin de pallier les éventuelles défaillances du système.

Les systèmes hybrides combinant plusieurs capteurs sont particulièrement utilisés pour améliorer le tracking basé vision. Ces systèmes multi-capteurs exploitent les contributions de chaque capteur pour compenser les dérives et les erreurs pouvant compromettre le suivi. Toutefois, ces dispositifs présentent des difficultés de mise en oeuvre, notamment la calibration, la synchronisation et la fusion de données hétérogènes.

Le dispositif de suivi de cibles que nous proposons dans cette étude, est composé d'un capteur inertiel et d'une caméra. Cette approche collaborative est justifiée par le fait que les systèmes basés vision peuvent échouer lors du suivi dans le cas d'un mouvement brusque ou lors de l'occultation des cibles, alors que les capteurs inertiels sont robustes et rapides mais moins précis que les systèmes de vision. La complémentarité de ces deux capteurs permet d'élaborer un dispositif hybride robuste et précis.

Dans ce chapitre, nous allons décrire notre approche de tracking hybride. Nous détaillons nos solutions originales concernant la calibration du système multi-capteurs, la synchronisation du flux de données et aussi la gestion de l'information issue des deux capteurs. Nous analysons aussi les réglages des paramètres pour une utilisation optimale du dispositif de tracking. Enfin, le protocole expérimental et les résultats obtenus sont présentés et interprétés.

4.2 Systèmes vision-inertiels existants

Plusieurs systèmes hybrides ont été proposés dans la littérature. Nous en avons cités quelques uns dans le premier chapitre de l'état de l'art. En effet, les travaux de Foxlin et al. [51][52] [118] et Chai et al. [26] se sont basés sur des dispositifs vision-inertiels pour la localisation dans les environnements de réalité augmentée. Par ailleurs, d'autres études allant dans le même sens se sont intéressées à cet aspect hybride. Un tour d'horizon sur quelques systèmes vision-inertiels est présenté dans ce qui suit.

Alves et al. [6] ont réalisé un système composé d'une caméra et d'une centrale inertielle. Un dispositif de calibration composé des deux capteurs, d'un pendule et une barre encodée sont utilisés pour caractériser les paramètres internes du capteur tels que : l'alignement des axes, le biais et le facteur d'échelle des accéléromètres et des gyroscopes. Le capteur inertiel est rigidement lié à la caméra, la rotation entre les deux repères est déterminée en déplaçant le système et en observant la direction verticale des points estimés par les deux capteurs. Ces mesures de mouvement de points verticaux permettent l'estimation de la rotation entre le capteur inertiel et la caméra.

Aron et al. [9] ont présenté une approche hybride pour la réalité augmentée composé d'un capteur inertiel et un système de suivi basé vision. Le capteur inertiel est utilisé seulement lorsque le système de vision échoue. Ce capteur inertiel détermine uniquement la rotation. La calibration de la rotation entre les deux capteurs est déterminée en effectuant plusieurs mesures de rotation du dispositif hybride pour résoudre la transformation permettant le passage entre les repères du système.

You et al. [168] ont présenté une approche hybride pour le recalage en réalité augmentée en utilisant une technologie de tracking basée sur un système composé d'une caméra et d'un gyroscope. Le prototype du système fusionne les données d'orientation du gyroscope avec les données image de la caméra pour stabiliser et corriger les défauts de chaque capteur. Les auteurs ont proposé une approche pour la reconnaissance de cibles basée sur l'analyse des composantes principales pour identifier les marqueurs de forme carrée dans l'image en temps réel. La calibration de la rotation entre les deux repères est effectuée par analyse du mouvement du dispositif hybride en exprimant la vitesse angulaire du gyroscope en fonction de la vitesse des points 2D.

Comparé aux approches existantes, notre système de tracking hybride permet l'estimation de la pose de la caméra (rotation et translation). La centrale inertielle est utilisée pour déterminer la rotation et la translation. Les systèmes de You et al. et Aron et al. utilisent seulement la rotation du capteur inertiel. You et al. réalisent une fusion de données des deux capteurs utilisés pour améliorer la précision et la robustesse du système. Dans notre étude, la caméra et la centrale inertielle se relayent pour le calcul de la rotation, concernant la translation, la caméra est utilisée pour corriger la dérive de la centrale inertielle. Notre approche effectue la suppléance de données pour rectifier la translation de la centrale inertielle et non pas pour stabiliser ou améliorer la précision du suivi. L'intérêt de notre application est le fonctionnement multimodal permettant d'aiguiller le système à chaque instant vers la procédure de tracking adéquate. Certes, une similitude existe avec le système de Aron et al. et le nôtre. Toutefois, la différence réside dans la calibration du capteur hybride, de plus, dans notre système la centrale inertielle permet de calculer la translation.

En outre, la méthode de calibration de notre système hybride est différente de celle présenté dans [6] et [168]. Notre méthode, comme nous le verrons dans les sections suivantes de ce chapitre, est rapide et nécessite seulement une seule pose de la caméra pour déterminer la rotation entre le repère de la centrale inertielle et celui de la caméra.

4.3 Calibration du capteur hybride

4.3.1 Description

Le système de capteurs utilisé est composé d'une centrale inertielle : une MTi¹ de Xsens [166] qui fournit les mesures suivantes :

- accélérations,
- vitesses angulaires,
- champs magnétiques,
- rotations angulaires.

et une caméra bâton de type Sony modèle XC-555P avec une focale de $6mm$, permettant l'acquisition, en temps réel, les images d'un flux vidéo. Les données des deux capteurs sont fournies instantanément, avec un temps d'échantillonnage défini, dans leurs repères respectifs. Les deux capteurs sont rigidement liés l'un à l'autre (figure 4.1).



FIG. 4.1 – Système de capteurs utilisé.

L'objectif de cette partie est la détermination de la rotation entre le repère de la caméra et le repère de la centrale inertielle. Dans l'expérimentation, on fera la calibration du système de capteurs, en utilisant les rotations de la caméra et celles de la MTi. La rotation entre les deux capteurs correspond alors à la composition des rotations.

4.3.2 Les repères utilisés

La caméra et la centrale inertielle possèdent toutes les deux leurs propres repère local. Le repère local de la caméra, R_C , est défini par calibration de celle-ci. Il permet la localisation de la caméra par rapport à un repère global, R_M , lié au monde et défini par l'utilisateur.

Le repère local de la centrale inertielle, R_I , est fixé au boîtier du capteur. Les données de rotations de la centrale inertielle sont mesurées par rapport au repère global qui est par défaut le repère du champ magnétique terrestre. Cependant, il est possible de définir un autre repère global différent de celui du champ magnétique terrestre, en utilisant les ré-initialisations [166]. Afin de simplifier la procédure de calibration, le repère global de la centrale inertielle est choisi de telle manière à ce qu'il coïncide avec le repère monde de la caméra, R_M (figure 4.2).

¹Les termes centrale inertielle ou MTi seront utilisés d'une manière interchangeable dans ce chapitre.

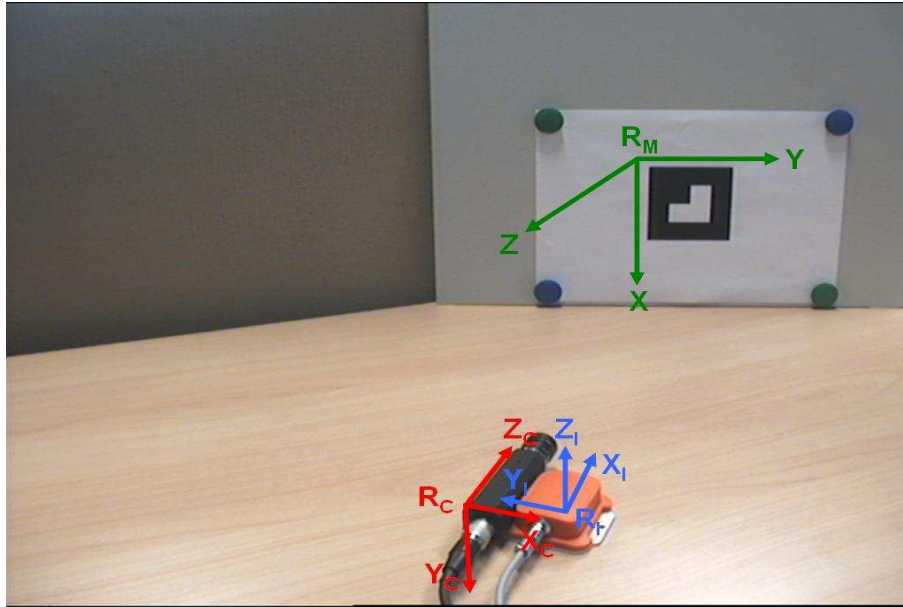


FIG. 4.2 – Représentation des repères utilisés.

4.3.3 Procédure de calibration

La procédure de calibration permet de déterminer la rotation entre le repère de la centrale inertielle, R_I , et le repère de la caméra, R_C . La rotation entre ces deux repères est exprimée par :

$$R_{CI} = R_{CM}R_{MI} \quad (4.1)$$

avec :

- R_{CI} : rotation du repère de la centrale inertielle, R_I , par rapport au repère de la caméra, R_C .
- R_{CM} : rotation du repère monde, R_M , par rapport au repère de la caméra, R_C .
- R_{MI} : rotation du repère de la centrale inertielle, R_I , par rapport au repère monde R_M .

La procédure de calibration de la rotation entre le repère de la caméra R_C , et le repère de la centrale inertielle R_I , est réalisée en deux étapes : tout d'abord une calibration manuelle du repère global de la MTi pour calculer les rotations R_{MI} et ensuite une estimation de pose de la caméra permet d'avoir la rotation R_{CM} .

4.3.3.1 La centrale inertielle

Comme nous l'avons déjà évoqué dans le paragraphe 4.3.2, la centrale inertielle détermine l'orientation de son repère local, R_I , par rapport au repère global lié au champ magnétique terrestre, R_G (figure 4.3).

Afin de résoudre l'équation 4.1, il est nécessaire de re-définir le repère global et exprimer les orientations de la centrale inertielle dans le repère monde de la caméra, R_M . On va utiliser alors les ré-initialisations de la MTi (reset) pour définir un nouveau repère global qui coïncide avec le repère monde, R_M . Il existe quatre type de ré-initialisations : globale, alignement, objet et cap [166]. Dans notre cas, on utilise la ré-initialisation globale qui permet de définir tous les axes du repère global selon la position actuelle de la MTi dans l'espace. Tous les angles sont mis à zéro lors de cette ré-initialisation et $R_G = R_I$. L'orientation de la MTi est par la suite calculée par rapport à ce nouveau repère global, soit la rotation R_{MI} (figure 4.4).

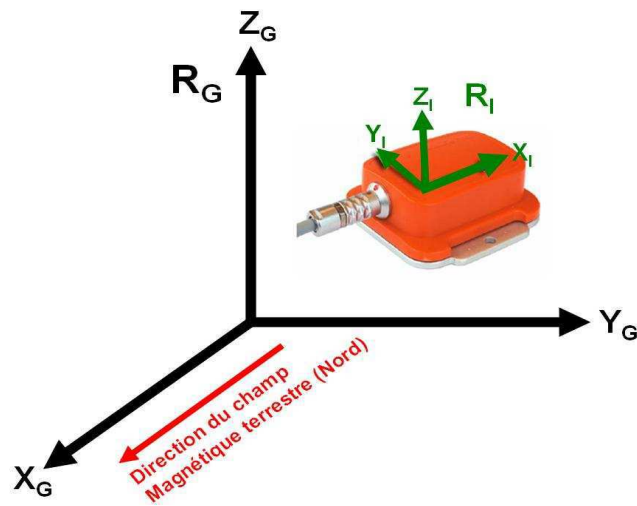


FIG. 4.3 – Représentation des repères de la centrale inertielle : MTi.

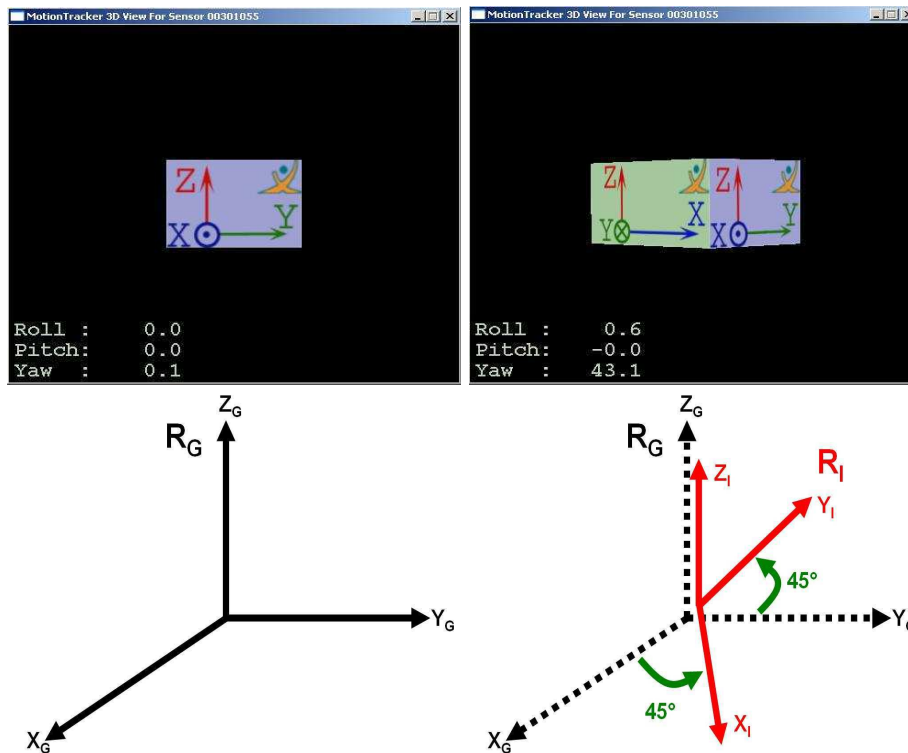


FIG. 4.4 – Définition du nouveau repère global.

4.3.3.2 La caméra

L'orientation du repère de la caméra, R_C , est donnée par l'algorithme d'estimation de la pose de Zhang (voir annexe A) qui est une méthode analytique dont le temps d'exécution est $< 20\mu s$. Cet estimateur de pose permet de calculer l'orientation du repère monde, R_M , par rapport au repère de la caméra, R_C , soit la rotation, R_{CM} (figure 4.5).

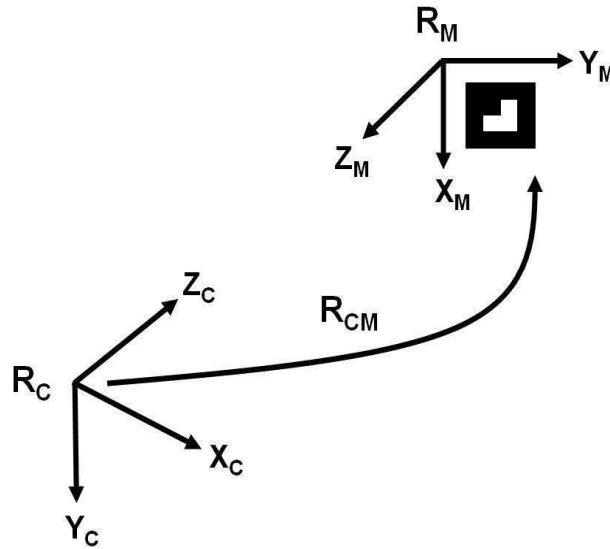


FIG. 4.5 – Estimation de la pose : rotation du repère monde par rapport au repère de la caméra.

4.3.4 Résolution de la rotation R_{CI}

La détermination de l'orientation du repère de la MTi par rapport au repère de la caméra est calculée par l'équation 4.1. En effet, on doit déterminer la rotation R_{CI} en utilisant les deux rotations : R_{CM} et R_{MI} , il s'agit donc de faire : $R_{CI} = R_{CM}R_{MI}$ (figure 4.6).

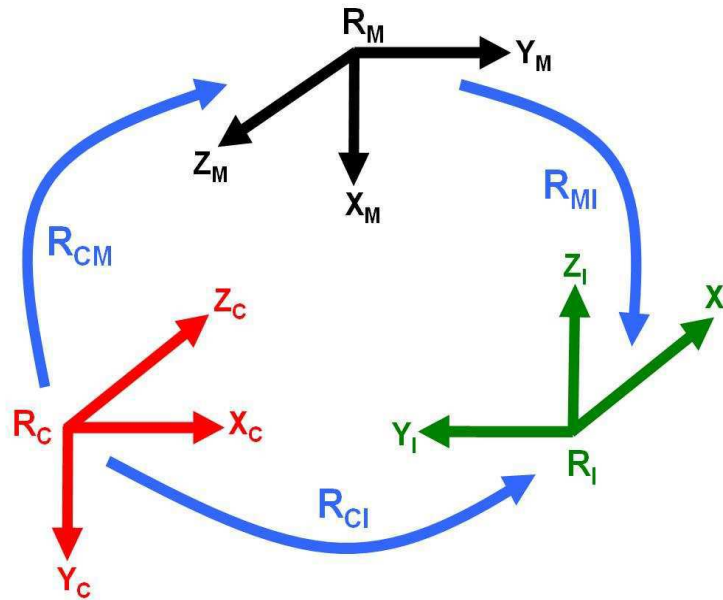


FIG. 4.6 – Représentation des rotations entre les repères utilisés.

La rotation R_{CI} est calculée en utilisant les quaternions car ceux-là ne présentent pas de singularités et donnent des solutions uniques pour l'orientation. Nous avons effectué une linéarisation autour du quaternion optimal. La solution finale est donnée par la moyenne suivante :

$$Q_{CI} = \frac{1}{N} \sum_{i=1}^N Q_{CI_i} \quad (4.2)$$

4.4 Modélisation du mouvement

Avant de présenter le processus de conception de notre approche de suivi hybride et de gestion d'occultations, nous devons d'abord étudier la cinématique du mouvement du capteur hybride constitué de la caméra et de la centrale inertielle. Nous allons analyser la position, l'orientation, la vitesse linéaire, la vitesse angulaire et l'accélération du mouvement. Cette étude nous permet d'établir les équations nécessaires régissant notre système pour le modéliser et exprimer les données dans un repère de référence. Les différents repères utilisés dans le système global sont illustrés sur la figure 4.2.

Nous définissons le repère de la caméra, R_C , comme étant le repère de référence de notre système de capteurs, c'est-à-dire toutes les données sont projetées dans ce repère.

Avant de procéder à la définition des équations cinématiques du mouvement, nous avons besoin de définir quelques notations qu'on utilisera dans cette étude. Ces termes ont été rassemblés dans le tableau 4.1.

\mathbf{V}_{IC}	Vitesse de R_C par rapport à R_I
\mathbf{V}_{MC}	Vitesse totale de R_C par rapport à R_M
$\mathbf{V}_{MC}^{linéaire}$	Composante de V_{MC} relative à la vitesse linéaire
$\mathbf{V}_{MC}^{rotationnelle}$	Composante de V_{MC} relative à la vitesse rotationnelle
\mathbf{V}_{MI}	Vitesse de R_I par rapport à R_M
\mathbf{A}_{MC}	Accélération de R_C par rapport à R_M
\mathbf{A}_{MI}	Accélération de R_I par rapport à R_M
\mathbf{T}_{CI}	Position de l'origine de R_I dans le repère R_C
\mathbf{T}_{MC}	Position de l'origine de R_C dans le repère R_M
ω_{MI}	Vitesse angulaire de R_I par rapport à R_M
α_{MC}	Orientation de R_C par rapport à R_M

TAB. 4.1 – Variables cinématiques utilisées.

4.4.1 Vitesse linéaire

Considérons que le repère de la centrale inertielle, R_I , est en mouvement avec une vitesse linéaire par rapport au repère monde, R_M , et qu'il n'y a pas de rotation entre R_I et R_M . La vitesse linéaire de R_C par rapport à R_M est alors la même que celle de R_I par rapport à R_M :

$$\mathbf{V}_{MC}^{linéaire} = \mathbf{V}_{MI} \quad (4.3)$$

R_I est rigidement lié à R_C , si R_I effectue une rotation propre autour de R_M avec une vitesse angulaire ω_{MI} , une vitesse rotationnelle induite par la rotation est produite. Cette vitesse, notée $\mathbf{V}_{MC}^{rotationnelle}$, est donnée par :

$$\mathbf{V}_{MC}^{rotationnelle} = R_{MI}\mathbf{V}_{IC} + \omega_{MI} \times R_{MI}\mathbf{T}_{IC} \quad (4.4)$$

On peut obtenir ω_{MI} en effectuant la rotation de ω_I (ω_I : vitesse angulaire de la centrale inertielle) par rapport à R_M :

$$\omega_{MI} = R_{MI}\omega_I \quad (4.5)$$

$\mathbf{V}_{IC} = 0$ car les deux capteurs sont rigidement liés l'un à l'autre. L'expression de la vitesse totale de R_C par rapport à R_M est la somme de la vitesse linéaire et rotationnelle :

$$\mathbf{V}_{MC} = \mathbf{V}_{MC}^{linéaire} + \mathbf{V}_{MC}^{rotationnelle} = \mathbf{V}_{MI} + \omega_{MI} \times R_{MI}\mathbf{T}_{IC} \quad (4.6)$$

4.4.2 Accélération

Pour avoir l'accélération de la caméra dans le repère monde, il suffit de dériver l'équation 4.6. On obtient alors :

$$\mathbf{A}_{MC} = \mathbf{A}_{MI} + \dot{\omega}_{MI} \times R_{MI} \mathbf{T}_{IC} + \omega_{MI} \times \frac{d}{dt} (R_{MI} \mathbf{T}_{IC}) \quad (4.7)$$

On considère que la vitesse angulaire est constante, l'accélération angulaire représentée par le terme $\dot{\omega}_{MI}$ dans l'équation 4.7 est donc nulle. L'équation 4.7 devient :

$$\mathbf{A}_{MC} = \mathbf{A}_{MI} + \omega_{MI} \times \frac{d}{dt} (R_{MI} \mathbf{T}_{IC}) \quad (4.8)$$

Dans les systèmes physiques, l'accélération due à l'effet de gravitation existe. Ce terme est exprimé dans le repère monde R_M en un vecteur dirigé vers le bas. Il est défini par :

$$G = (0 \ 0 \ 9.81)^T (m/sec^2) \quad (4.9)$$

4.4.3 Vitesse angulaire

Les données gyroscopiques de la centrale inertielle sont exprimées dans leur propre repère R_I , nous avons besoin d'exprimer la rotation de la caméra par rapport au repère monde, $\dot{\alpha}_{MC}$, à partir de ω_I , la vitesse angulaire de la centrale inertielle. On définit la matrice symétrique de biais qui utilise la matrice de rotation et sa dérivée, cette matrice est définie par S :

$$S = \dot{R}_{MC} R_{MC}^{-1} \quad (4.10)$$

La matrice de biais peut être exprimée par le vecteur de vitesse angulaire. Si nous définissons la vitesse angulaire par le vecteur :

$$\omega_I = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (4.11)$$

alors la matrice S peut être écrite :

$$S = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (4.12)$$

En utilisant la propriété de la matrice de rotation décrite dans l'équation 4.10, on peut trouver la relation entre les termes du vecteur de la vitesse angulaire et les termes de la matrice de rotation. Se basant sur cette relation et la convention des angles d'Euler $Z - Y - X$, on peut trouver aussi la relation entre les angles de vitesse et le vecteur des vitesses angulaires. Le résultat est donné par :

$$\omega_I = J(\alpha_{MC}) \dot{\alpha}_{MC} \quad (4.13)$$

où la matrice J est le Jacobien qui relie les angles et le vecteur de vitesse angulaire. Le vecteur des angles de vitesse est défini par :

$$\dot{\alpha}_{MC} = \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (4.14)$$

α_{MC} est une fonction du temps, elle détermine l'orientation de R_C par rapport à R_M en respectant la convention $Z - Y - X$ des angles d'Euler. La matrice du Jacobien obtenue est exprimée par :

$$J(\alpha_{MC}) = \begin{pmatrix} \cos \phi \cos \theta & -\sin \phi & 0 \\ \sin \phi \cos \theta & \cos \phi & 0 \\ -\sin \theta & 0 & 1 \end{pmatrix} \quad (4.15)$$

avec : ϕ angle de rotation autour de X , θ angle de rotation autour de Y , ψ angle de rotation autour de Z .

On peut exprimer le vecteur des angles de vitesse en fonction du vecteur de vitesse angulaire (les données du gyroscope) avec l'équation :

$$\dot{\alpha}_{MC} = W(\alpha_{MC})\omega_I \quad (4.16)$$

La matrice W est définie par l'inverse de la matrice du Jacobien J :

$$W(\alpha_{MC}) = J^{-1}(\alpha_{MC}) = \begin{pmatrix} \cos \phi \sec \theta & \sin \phi \sec \theta & 0 \\ -\sin \phi & \cos \phi & 0 \\ \cos \phi \tan \theta & \sin \phi \tan \theta & 1 \end{pmatrix} \quad (4.17)$$

4.4.4 Prédiction de l'orientation

La relation entre l'angle de rotation et la vitesse angulaire (donnée par la centrale inertielle) est donnée par l'équation 4.18, on peut approximer avec la série de Taylor, l'orientation par :

$$\alpha_{MC}(t + \Delta T) = \alpha_{MC}(t) + \dot{\alpha}_{MC}(t) \Delta T + \ddot{\alpha}_{MC}(t) \frac{\Delta T^2}{2} + \dots \quad (4.18)$$

Le nombre de termes à retenir dans l'équation 4.18 dépend de ΔT . Les trois premiers termes sont suffisants pour assurer une bonne précision pour cette application.

Le terme de l'accélération de l'équation 4.18 est donné par :

$$\ddot{\alpha}_{MC}(t) = \frac{\partial}{\partial \omega} [W(\alpha_{MC})\omega] \dot{\omega}(t) + \frac{\partial}{\partial \alpha_{MC}} [W(\alpha_{MC})\omega] \dot{\alpha}_{MC}(t) \quad (4.19)$$

On considère que l'accélération angulaire est nulle. Par conséquent, l'équation 4.19 est simplifiée :

$$\ddot{\alpha}_{MC}(t) = \frac{\partial}{\partial \alpha_{MC}} [W(\alpha_{MC})\omega] \dot{\alpha}_{MC}(t) \quad (4.20)$$

On définit, F , la matrice dérivée de $W(\alpha_{MC})\omega$ par rapport à α_{MC} :

$$F(\alpha_{MC}, \omega) = \frac{\partial}{\partial \alpha_{MC}} [W(\alpha_{MC})\omega] = \begin{bmatrix} 0 & \frac{\cos \phi \sin \theta}{\cos^2 \theta} \omega_x + \frac{\sin \phi \sin \theta}{\cos^2 \theta} \omega_y & \frac{\cos \phi}{\cos \theta} \omega_y - \frac{\sin \phi}{\cos \theta} \omega_x \\ 0 & 0 & -(\cos \phi) \omega_x - (\sin \phi) \omega_y \\ 0 & \frac{\cos \phi}{\cos^2 \theta} \omega_x + \frac{\sin \phi}{\cos^2 \theta} \omega_y & \frac{\cos \phi \sin \theta}{\cos \theta} \omega_y - \frac{\sin \phi \sin \theta}{\cos \theta} \omega_x \end{bmatrix} \quad (4.21)$$

en remplaçant l'équation 4.16, 4.20 et 4.21 dans 4.18, on obtient :

$$\alpha_{MC}(t + \Delta T) = \alpha_{MC}(t) + W(\alpha_{MC}(t))\omega(t)\Delta T + F(\alpha_{MC}(t), \omega(t))W(\alpha_{MC}(t))\omega(t) \left(\frac{\Delta T^2}{2} \right) \quad (4.22)$$

L'équation 4.22 représente l'orientation de la caméra au cours du temps à partir de l'orientation $\alpha_{MC}(t)$ et de la vitesse angulaire issue de la centrale inertielle $\omega(t)$.

4.4.5 Prédiction de la translation

Pour prédire la translation de R_C par rapport à R_M , nous avons besoin d'écrire le modèle cinématique du mouvement uniforme. Le développement en séries de Taylor, nous permet d'écrire l'équation qui prédit la translation de R_C par rapport à R_M :

$$\mathbf{T}_{MC}(t + \Delta T) = \mathbf{T}_{MC}(t) + \mathbf{V}_{MC}(t) \Delta T + \mathbf{A}_{MC}(t) \left(\frac{\Delta T^2}{2} \right) + \dots \quad (4.23)$$

Nous utilisons les trois premiers termes de l'équation 4.23 pour représenter l'équation de la translation car ils sont suffisants pour notre application.

Le terme de la vitesse est prédit par l'équation suivante :

$$\mathbf{V}_{MC}(t + \Delta T) = \mathbf{V}_{MC}(t) + \mathbf{A}_{MC}(t) \Delta T \quad (4.24)$$

L'accélération est prédite par :

$$\mathbf{A}_{MC}(t + \Delta T) = \mathbf{A}_{MC}(t) \quad (4.25)$$

Les deux équations de prédiction de l'orientation et de la translation sont établies, nous allons maintenant décrire le module de suppléance de données permettant de gérer les données des deux capteurs pour calculer la pose de la caméra.

4.5 Suppléance de données

Nous allons décrire l'algorithme hybride utilisé pour la gestion d'occultations et la suppléance de données de la centrale inertielle et la caméra. Nous présentons le module de suppléance de données et nous expliquons les étapes de fonctionnement de l'algorithme, ses propriétés et aussi ses paramètres. Concernant l'orientation, les deux capteurs fonctionnent d'une manière parallèle. La caméra estime la rotation lorsque les cibles sont visibles et la MTi détermine cette même orientation si les cibles sont occultées. La suppléance de données est effectuée pour la translation, les deux capteurs calculent la translation au même temps car la MTi nécessite les mesures de la caméra pour corriger sa dérive.

Le système hybride est composé de 3 modules (figure 4.7) :

- Module d'estimation de la pose de la caméra.
- Module d'estimation de l'orientation à partir de la centrale inertielle.
- Module d'estimation de la position à partir des accélérations de la centrale inertielle.

Le diagramme temporel de suppléance de données de translation de la caméra et de la centrale inertielle est représenté sur la figure 4.8. Si la cible est visible et identifiée, l'algorithme de suivi basé vision estime la pose de la caméra et réalise le recalage du modèle 3D sur l'image. Par ailleurs, lorsque la cible n'est pas détectée à cause d'une occultation ou d'un flou de mouvement, la pose dans ce cas, est estimée par la centrale inertielle. Ce schéma montre aussi la collaboration entre les deux capteurs, d'une part, la suppléance de données de la MTi et de la caméra permet de corriger la dérive dans le calcul de position par la MTi, et d'autre part, si le système de vision devient défaillant, la centrale inertielle prend le relais du système de vision pour poursuivre le tracking.

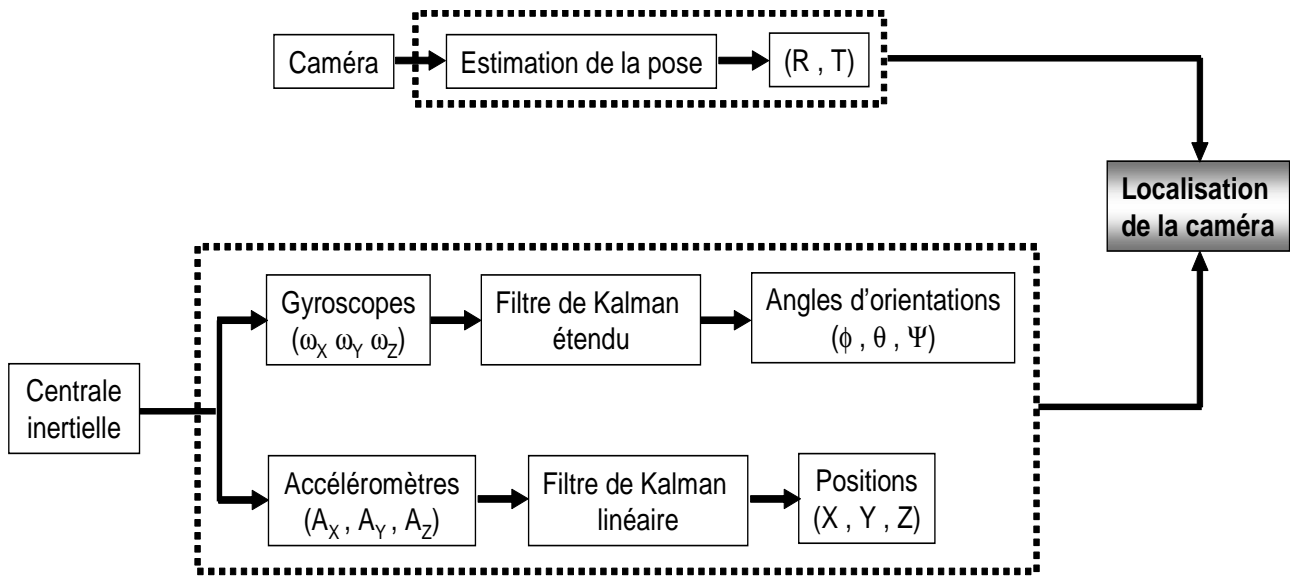


FIG. 4.7 – Suppléance de données du capteur hybride.

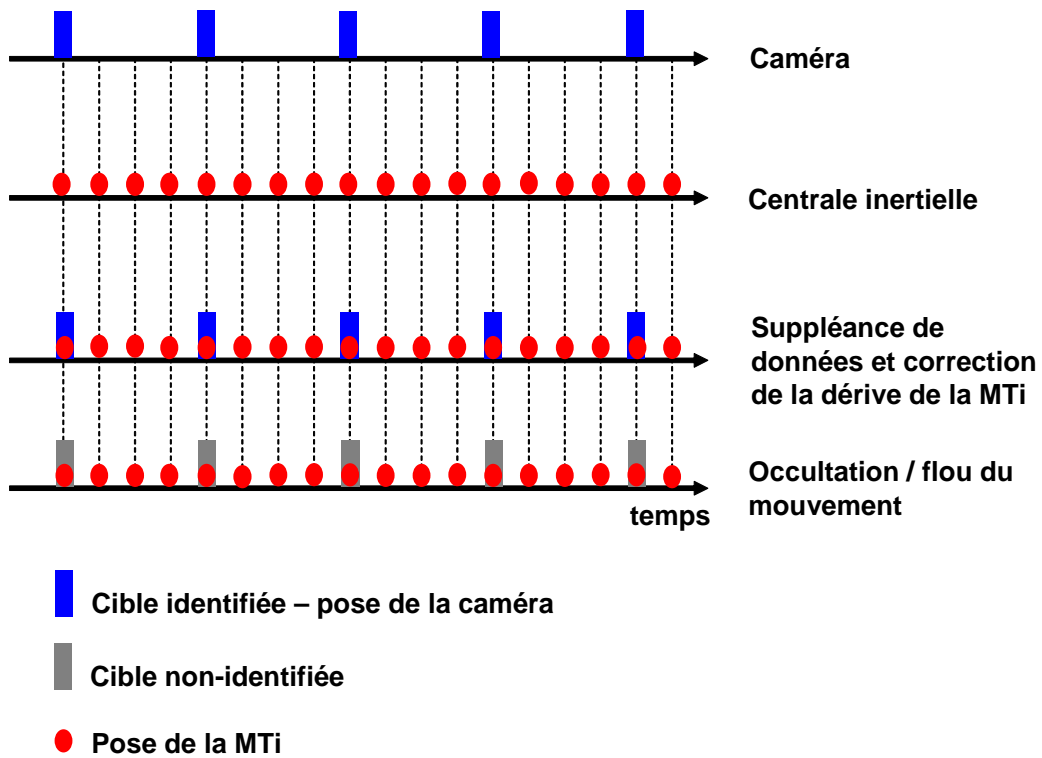


FIG. 4.8 – Diagramme temporel de suppléance de données.

4.5.1 Module d'orientation

Les caractéristiques de la MTi par rapport aux orientations sont :

- Angles de rotations en 3D.
- Résolution angulaire : 0.05° *RMS*.
- Précision statique (roulis/tangage) : $< 0.5^\circ$.
- Précision statique (lacet) : $< 1.0^\circ$.
- Précision dynamique : 2° *RMS*.
- Fréquence d'échantillonnage : 25, 50, 100, 120, 200, 256, 320, 400, 512 Hz.

La MTi fonctionne avec précision dans des conditions variables de l'environnement. Les performances de l'orientation sont améliorées en faisant des réglages sur le Filtre de Fusion de Données du capteur. Trois paramètres sont importants pour le réglage du filtre : le facteur de pondération, le gain du filtre et le facteur d'adaptation du champ magnétique.

4.5.1.1 Facteur de pondération

Ce facteur permet d'indiquer l'influence des magnétomètres sur les données des accéléromètres. Si ce facteur est égal à 1, cela indique que les données du champ magnétique sont égales à celles des accélérations. Si cette valeur est égale à 0, alors seuls les accéléromètres fonctionnent sans l'influences des champs magnétiques ce qui permet une meilleure localisation en orientation de la centrale inertielle.

4.5.1.2 Gain du filtre

Le gain du filtre est un facteur très important pour le réglage de la fréquence de fusion de données du filtre. Par exemple si le gain du filtre est égal à 1, cela veut dire que les composantes des fréquences du vecteur d'orientation dépassant 1Hz sont déterminées par les gyroscopes et les composantes inférieures à 1Hz sont calculées à partir des accéléromètres et magnétomètres. Les valeurs valides des gains sont comprises entre 0.01 et 50, cependant, quelques valeurs peuvent créer des instabilités de l'algorithme de fusion dans certaines conditions, la valeur recommandée du gain est 1.

4.5.1.3 Facteur d'adaptation du champ magnétique

La MTi peut être utilisée pour estimer l'orientation d'un objet 3D en temps réel, cependant, lorsqu'elle est montée sur un objet qui contient des matériaux ferromagnétiques, comme une caméra, le champ magnétique mesuré est distordu. Ceci cause des erreurs de mesure sur l'orientation. Les perturbations du champ magnétique peuvent être divisées en deux types :

- Les perturbations causées par les objets ferromagnétiques présents dans l'environnement : ce type de perturbations est non-déterministe et ne peut pas être complètement compensé. Toutefois, on peut réduire l'effet de ces perturbations en modifiant les paramètres du filtre de la MTi.
- Les perturbations dues à un objet sur lequel la MTi est montée : dans ce cas l'effet des distorsions magnétiques affecte son orientation. Ce type d'altération est déterministe et peut être prédit et éliminé par une calibration faite par un logiciel spécialisé, appelé AMD (Adapt to Magnetic Disturbances), ou bien en modifiant le facteur de pondération qui réduit l'influence des magnétomètres.

Un filtre de Kalman étendu est implémenté dans la MTi pour calculer les orientations de la centrale inertielle à partir des vitesses angulaires, l'équation du filtre est donnée par :

$$\begin{pmatrix} \alpha_k \\ \omega_k \end{pmatrix} = \begin{pmatrix} 1 & \left((\Delta T)W(\alpha_{k-1}) + \frac{(\Delta T)^2}{2}F(\alpha_{k-1}, \omega_{k-1})W(\alpha_{k-1}) \right) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{k-1} \\ \omega_{k-1} \end{pmatrix} \quad (4.26)$$

L'orientation peut être exprimée par les trois représentations suivantes (voir annexe D) :

- Angles d'Euler.
- Quaternions.
- Matrices de rotation.

4.5.2 Module des données calibrées

4.5.2.1 Modèle physique du capteur

Les capteurs à l'intérieur de la MTi (accéléromètres, gyroscopes, magnétomètres) sont calibrés par rapport aux modèles physiques des données. La modélisation du capteur est élaborée après l'identification et l'évaluation de ses paramètres.

Le modèle basique du capteur est linéaire et donné par :

$$s = K_T^{-1}(u - b_T) \quad (4.27)$$

Ce modèle simplifié, comporte une matrice de gain unique K_T et un vecteur de biais b_T . Les données calibrées sont utilisées pour relier la sources de tension u (entier non-signé) avec des grandeurs physiques s .

La centrale inertielle fournit des accélérations dans son repère local. Les accéléromètres mesurent les accélérations de la MTi y compris la gravité. Donc pour avoir des accélérations propres, il est nécessaire de soustraire l'accélération de la gravité. L'accélération propre de la MTi, A_I , est donnée par :

$$\mathbf{A}_I = \mathbf{A}_{brute} - \begin{pmatrix} 0 & 0 & 9.81 \end{pmatrix}^T (m/s^2) \quad (4.28)$$

Pour estimer les positions à partir des accélérations, nous utilisons le filtre de Kalman linéaire qui réalise une double intégration de l'accélération, les étapes du filtre sont expliquées dans la section suivante.

4.5.3 Le filtre de Kalman pour la translation

Nous allons maintenant faire une représentation d'état de la modélisation du mouvement développée dans la section 4.4.5. Le vecteur d'état du processus représente les paramètres de translation de la caméra que l'on cherche à déterminer. Nous avons utilisé un filtre de Kalman pour l'estimation de la translation de la caméra à partir des accélérations de la MTi.

En se basant sur le modèle cinématique développé précédemment, nous sommes en mesure de prédire la translation avec le filtre de Kalman. Nous utilisons les accéléromètres de la centrale inertielle pour estimer les positions avec le vecteur d'état suivant :

$$\hat{x} = \left(\mathbf{X} \ \mathbf{Y} \ \mathbf{Z} \ \mathbf{V}_X \ \mathbf{V}_Y \ \mathbf{V}_Z \ \mathbf{A}_X \ \mathbf{A}_Y \ \mathbf{A}_Z \right)^T \quad (4.29)$$

Le modèle d'état du système est donné par :

$$\begin{pmatrix} \mathbf{X}_k \\ \mathbf{V}_k \\ \mathbf{A}_k \end{pmatrix} = \begin{pmatrix} \mathbf{X}_{k-1} + \Delta T \mathbf{V}_{k-1} + \frac{(\Delta T)^2}{2} \mathbf{A}_{k-1} \\ \mathbf{V}_{k-1} + \Delta T \mathbf{A}_{k-1} \\ \mathbf{A}_{k-1} \end{pmatrix} + \begin{pmatrix} W_{k-1}^1 \\ W_{k-1}^2 \\ W_{k-1}^3 \end{pmatrix} \quad (4.30)$$

où W_{k-1}^i représente le bruit de mesure et $\mathbf{X}_k, \mathbf{V}_k, \mathbf{A}_k$ sont des vecteurs de dimension 3×1 qui désignent respectivement la position, la vitesse et l'accélération dans l'espace.

Nous allons maintenant appliquer le filtre de Kalman à la modélisation du mouvement que nous avons établie précédemment. Les mesures du système proviennent de la centrale inertielle, le filtre récupère des mesures d'accélération et réalise une double intégration afin d'estimer les positions. Lorsque les données du capteur sont disponibles, le filtre effectue une prédiction et une correction des états. La phase de prédiction utilise l'état estimé à l'instant précédent pour produire une estimation à l'état courant. Dans l'étape de correction ou de mise à jour, les observations à l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise (annexe B).

4.5.3.1 Prédiction

La prédiction des états et de la covariance d'erreur est réalisée par les équations de projections suivantes :

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q\end{aligned}\quad (4.31)$$

où : A est la matrice de transition, \hat{x}_k^- est l'estimation du vecteur d'état a priori à l'instant k , \hat{x}_{k-1} est l'estimation du vecteur d'état à l'instant $k - 1$. P représente la covariance d'erreur et Q est la covariance du bruit de processus.

Nous appliquons maintenant le filtre de Kalman linéaire à la modélisation de la translation :

$$\begin{pmatrix} \mathbf{X}_k \\ \mathbf{V}_k \\ \mathbf{A}_k \end{pmatrix} = \begin{pmatrix} 1 & \Delta T & \frac{(\Delta T)^2}{2} \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{X}_{k-1} \\ \mathbf{V}_{k-1} \\ \mathbf{A}_{k-1} \end{pmatrix}\quad (4.32)$$

où ΔT est le temps d'échantillonnage. Les états du système sont des vecteurs de dimension 3×1 .

4.5.3.2 Correction

Dans cette étape, le filtre met à jour les états du système avec les données des accélérations de la centrale inertielle, tout d'abord, le gain du filtre de Kalman est calculé par :

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}\quad (4.33)$$

où K est le gain du Kalman, H est la matrice de mesure et R est la covariance du bruit de mesure.

Les états sont ensuite mis à jour avec les mesures des accélérations par l'équation suivante :

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)\quad (4.34)$$

avec z_k est le vecteur de mesure représentant les accélérations de la caméra A_C .

Enfin la covariance d'erreur est mise à jour par :

$$P_k = (I - K_k H) P_k^- \quad (4.35)$$

où :

$$H = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1)^T \quad (4.36)$$

Le filtre de Kalman que nous avons employé permet de calculer la position du repère de la centrale inertielle par rapport à un point de référence. Or, dans notre étude, nous voulons estimer la position

de la caméra par rapport au repère monde. Les deux capteurs sont liés rigidement entre eux, l'accélération de la caméra est la même que celle de la centrale inertielle lorsque le capteur hybride est en mouvement de translation, l'accélération de la caméra est donnée par :

$$\mathbf{A}_C = R_{CI}\mathbf{A}_I \quad (4.37)$$

telle que R_{CI} est la matrice de rotation de la centrale inertielle par rapport à la caméra qu'on a déterminée lors de la calibration du capteur hybride.

Toutefois, lorsqu'il s'agit d'un mouvement en rotation, cette accélération est donnée par l'équation :

$$\mathbf{A}_C = \mathbf{A}_I + \dot{\omega}_I \times R_{CI}\mathbf{T}_{IC} + \omega_I \times \frac{d}{dt}(R_{CI}\mathbf{T}_{IC}) \quad (4.38)$$

4.6 Résultats expérimentaux

Nous présentons maintenant les résultats expérimentaux que nous avons obtenus par notre système hybride de tracking. Nous parlons de l'architecture proposée pour la gestion des deux capteurs : caméra et centrale inertielle et nous proposons une solution globale pour la gestion d'occultations des cibles en utilisant la localisation de la MTi.

Nous avons consacré une partie considérable de cette étude expérimentale à l'évaluation de la translation avec le filtre de Kalman car cette estimation repose sur une double intégration des accélérations issues de la centrale inertielle, ce qui provoque des erreurs et des dérives qui sont accumulées en fonction du temps.

4.6.1 Protocole expérimental

Le protocole expérimental que nous avons adopté pour nos expérimentations est divisé en deux parties : la première consiste à caractériser la centrale inertielle, les types de signaux obtenus, le filtrage et la soustraction de la gravité. La deuxième partie permet l'évaluation du suivi hybride avec le système multi-capteurs, la calibration de l'orientation et de la translation de la MTi par rapport à la caméra, le paramétrage et l'évaluation du module d'estimation de la translation en utilisant le filtre de Kalman et enfin, le test du système hybride en présence d'occultations et le flou de mouvement.

4.6.2 Caractérisation de la centrale inertielle : MTi

4.6.2.1 Données de la MTi : rotations et données calibrées

La MTi est une centrale inertielle miniature contenant des accéléromètres, des gyroscopes et des magnétomètres. Elle contient un processeur permettant le calcul des orientations et des données calibrées. La MTi est une unité de mesure permettant le contrôle, la navigation et la capture des mouvements.

Les rotations sont fournies directement par la composante logicielle de la MTi, toutefois, ces rotations ont été obtenues à partir des vitesses angulaires calculées par les gyroscopes (figure 4.9-b). Un filtre de Kalman étendu est implémenté dans le processeur de la MTi afin de déterminer les rotations sous forme de matrices de rotations, angles de rotation (figure 4.10-a) ou quaternions (figure 4.10-b). Le réglage du facteur de pondération, du gain de filtre et le facteur d'adaptation du champ magnétique permet d'avoir des orientations très précises.

Les accéléromètres fournissent des accélérations de la MTi dans son repère local. Les signaux d'accélérations mesurent aussi la gravité terrestre. Les allures des signaux d'accélérations sont représentées sur la figure 4.9-a.

Les magnétomètres mesurent les champs magnétiques, ces capteurs sont utilisés pour améliorer les mesures des rotations. Les signaux issus des magnétomètres sont illustrés sur la figure 4.9-c.

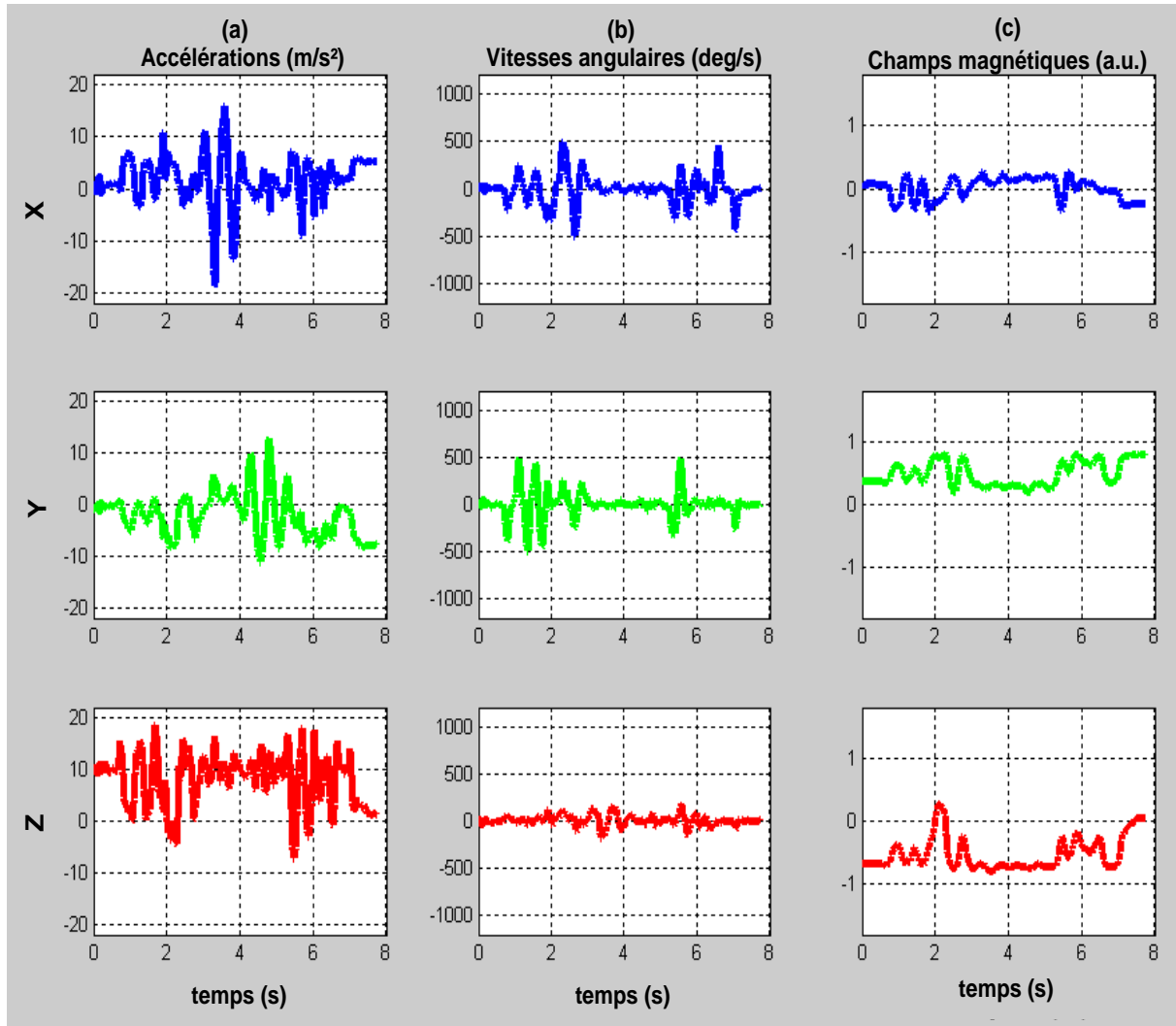


FIG. 4.9 – Données calibrées de la MTi.

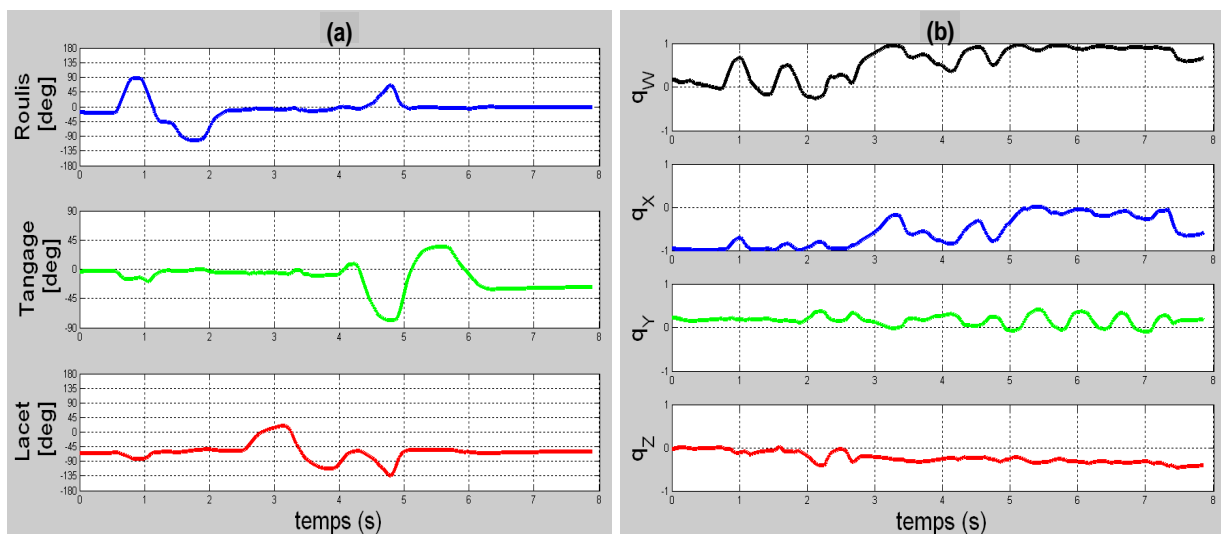


FIG. 4.10 – Représentation des rotations de la MTi. (a) Angles d'Euler. (b) Quaternions.

4.6.2.2 Filtrage des données

Nous nous intéressons particulièrement aux signaux d'accélération qui nous servent au calcul des positions. Ces signaux sont entachés de bruit et doivent être filtrés.

Nous avons appliqué un filtre numérique de Tchebychev. Ce filtre a une meilleure sélectivité et monotonie en bande passante. La courbe de réponse en bande passante présente un certain nombre d'ondulations. La valeur maximale de ces ondulations est un paramètre de conception du filtre. Plus elle est importante, à ordre constant, plus le filtre est sélectif (sa pente est plus raide hors bande passante). Il est réalisable par un procédé classique de synthèse en échelle. Ce filtre est assez largement utilisé là où l'ondulation ne pose pas de problèmes. Matthaei [106] a donné des formules récursives simples qui permettent de calculer facilement la valeur des composants constituant ce filtre.

Le filtre numérique de Tchebychev d'ordre N est défini par l'équation aux différences suivante :

$$y(n) = \sum_{k=0}^N b_k x(n-k) - \sum_{k=1}^M a_k y(n-k) \quad (4.39)$$

Sa fonction de transfert, dans le domaine fréquentiel (Transformée en Z), est définie par :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k z^{-k}}{\sum_{k=0}^M a_k z^{-k}} \quad (4.40)$$

Dans notre cas, nous avons considéré un filtre de Tchebychev d'ordre 3. Ce filtre numérique passe-bas permet d'éliminer les bruits hautes-fréquences, les résultats du filtrage des signaux d'accélération sont illustrés sur la figure 4.11. Les coefficients du filtre sont donnés dans la tableau 4.2.

a_0	a_1	a_2
1.0000	-1.1605	0.4344
b_0	b_1	b_2
0.1334	0.0072	0.1334

TAB. 4.2 – Coefficients du filtre passe-bas.

4.6.2.3 Soustraction de la gravité

Comme nous l'avons déjà mentionné, les accéléromètres de la MTi calculent des accélérations brutes, la gravité de la terre est aussi mesurée. Afin de récupérer les accélérations propres de la MTi sans la gravité, nous avons soustrait sa valeur du vecteur de mesure de l'accéléromètre comme suit :

$$\mathbf{A}_I = \mathbf{A}_{brute} - R_{IM}G \quad (4.41)$$

où : $G = (0 \ 0 \ 9.81)^T$ (m/s^2) et R_{IM} est la matrice de rotation du repère monde par rapport au repère local de la centrale inertielle.

La soustraction de l'accélération de la gravité de la terre des accéléromètres de la MTi est représentée par la figure 4.12.

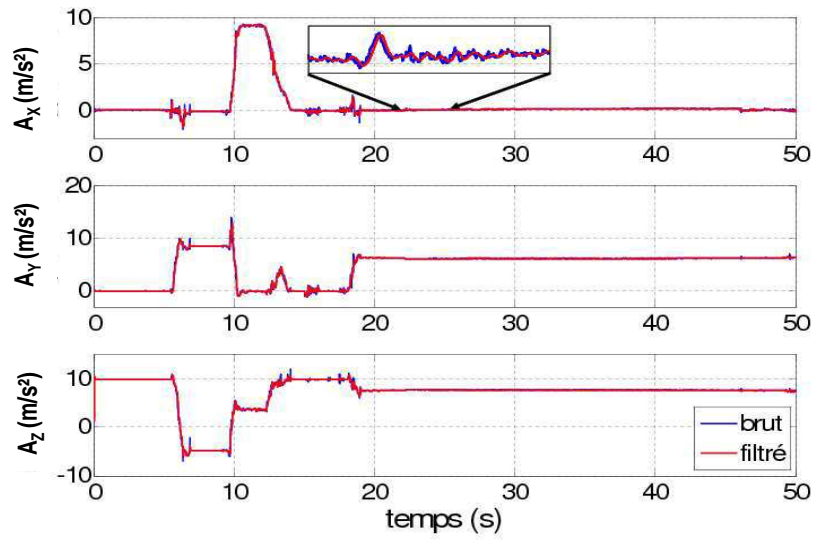


FIG. 4.11 – Filtrage des signaux d'accélérations.

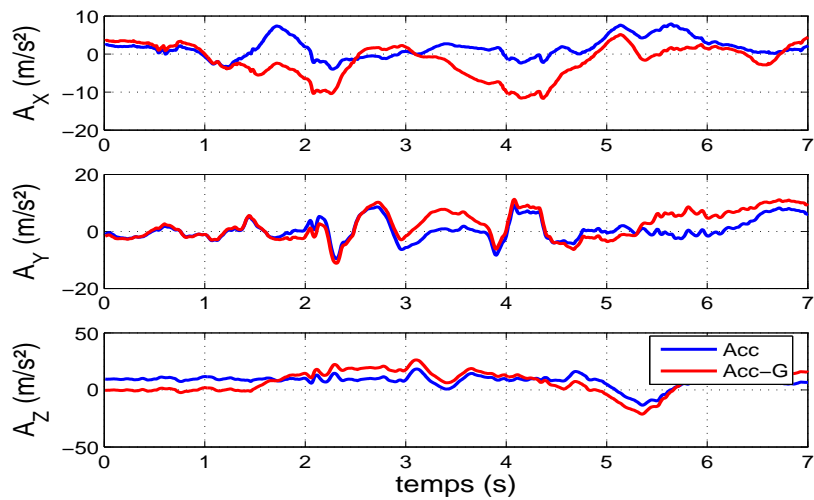


FIG. 4.12 – Représentation de l'accélération de la MTi avant et après soustraction de la gravité.

Nous avons subdivisé les expérimentations réalisées en deux parties. Dans un premier temps, nous nous intéressons à l'évaluation de la translation de la MTi pour le suivi hybride (section 4.6.3). Nous menons une étude exhaustive pour caractériser les erreurs de dérives de la centrale inertielle lors de l'estimation de la translation. Les tests effectués pour la translation portent sur une :

- calibration de l'orientation et de la translation,
- paramétrisation du filtre de Kalman pour la translation,
- évaluation du filtre de Kalman avec les données synthétiques,
- évaluation du filtre de Kalman avec les données réelles.

La deuxième partie des évaluations concerne la pose de la MTi (translation et rotation), les tests réalisés sont présentés dans la section 4.6.4 et se rapportent aux :

- suivi et recalage,
- occultations,
- flou de mouvement.

4.6.3 Suivi hybride

Dans cette deuxième partie expérimentale, nous allons présenter les étapes de calibration du capteur hybride, les paramètres du filtre de Kalman pour la translation et nous testons le filtre avec les données synthétiques et réelles. Un intérêt particulier est donné à l'estimation des positions avec la MTi car celles-ci sont calculées par double intégration des accélérations. L'accumulation des constantes et l'intégration du bruit influent considérablement sur les résultats obtenus. Plusieurs évaluations sont menés pour caractériser la dérive de la MTi et déterminer le temps minimal de correction de la translation.

Concernant l'orientation, elle est estimée correctement car les erreurs de rotation sont prises en charge par les composantes logicielles de la MTi. En outre, nous avons effectué un réglage optimal du filtre de fusion de données de l'orientation pour compenser les perturbations dues aux champs magnétiques et aux interférences causées par les objets métalliques.

4.6.3.1 Calibration de l'orientation

La calibration de l'orientation détermine la matrice de rotation entre la centrale inertielle et la caméra, cela est réalisé en calculant la transformation suivante :

$$R_{CI} = R_{CM}R_{MI} \quad (4.42)$$

où R_{CM} est calculée en estimant la pose de la caméra et R_{MI} est la rotation de la MTi par rapport au repère monde, fournie par la MTi.

Nous avons calculé 335 valeurs de R_{CI} qui sont représentées sur la figure 4.13, nous avons exprimé la rotation avec les quaternions car ils ne présentent pas de singularités. Puisque les composantes des quaternions sont proches les unes des autres, nous les avons linéarisées autour de leurs valeurs moyennes. La rotation est donc, donnée par les expressions suivantes :

$$\begin{aligned} Q_X &= \frac{1}{N} \sum_{i=1}^N Q_{Xi} \\ Q_Y &= \frac{1}{N} \sum_{i=1}^N Q_{Yi} \\ Q_Z &= \frac{1}{N} \sum_{i=1}^N Q_{Zi} \\ Q_W &= \frac{1}{N} \sum_{i=1}^N Q_{Wi} \end{aligned} \quad (4.43)$$

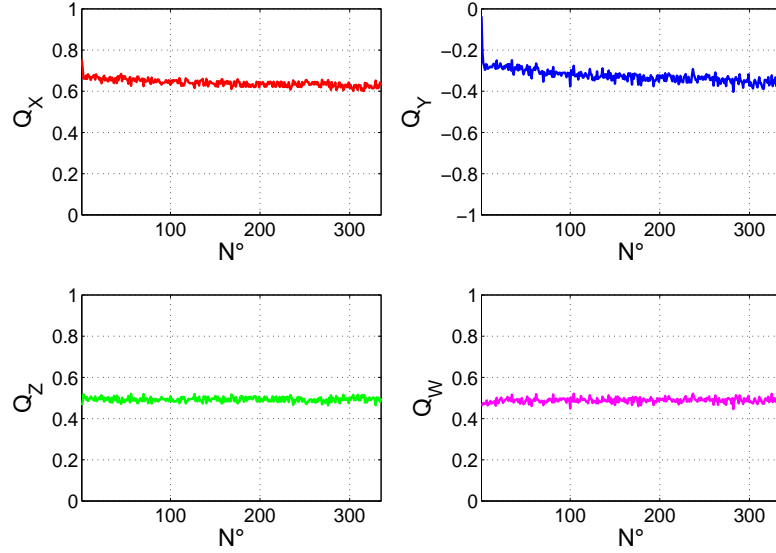


FIG. 4.13 – Les composantes du quaternion représentant la rotation R_{CI} .

Les valeurs numériques de la rotation R_{CI} sont données par :

$$Q_{CI} = \begin{pmatrix} 0.6414 \\ -0.3252 \\ 0.4927 \\ 0.4883 \end{pmatrix} \quad (4.44)$$

Ce quaternion représente les angles d'Euler suivants :

$$\alpha_{CI} = \begin{pmatrix} 96.3977 \\ -71.7319 \\ 11.9268 \end{pmatrix} (deg) \quad (4.45)$$

4.6.3.2 Calibration de la translation

La translation du repère de la MTi par rapport au repère de la caméra est déterminée par la relation suivante :

$$\mathbf{T}_{CI} = R_{CM}\mathbf{T}_{MI} + \mathbf{T}_{CM} \quad (4.46)$$

où :

- \mathbf{T}_{MI} : translation du repère de la centrale par rapport au repère monde.
- \mathbf{T}_{CM} : translation du repère monde par rapport au repère de la caméra.

\mathbf{T}_{MI} est déterminée par simple mesure de l'origine du repère local de la MTi représenté physiquement sur le boîtier du capteur par rapport au repère monde R_M . Par ailleurs, un travail antérieur a été réalisé pour déterminer avec précision la translation de la caméra par rapport à la centrale inertielle. Le système utilisé est composé du capteur hybride monté sur le bras d'un robot. Le robot permet le déplacement de son repère outil dans l'espace de travail et calcule la position et l'orientation de ce repère par rapport à un repère de référence. La calibration de la caméra et les coordonnées fournies par le robot permettent de déterminer la transformation entre la centrale inertielle et la caméra [98].

La caméra et la MTi sont rigidement liées, le repère global de la MTi est superposé sur le repère monde de la caméra. La MTi mesure alors les orientations de son repère local par rapport au repère monde de la caméra, soit R_{MI} . L'estimation de la pose détermine la rotation, R_{CM} , et la translation, \mathbf{T}_{CM} , de la caméra par rapport au repère monde. Enfin, la translation du repère local de la MTi par rapport au repère de la caméra est déterminée par l'équation 4.46.

Enfin, les valeurs de la translation, T_{CI} sont données par :

$$\mathbf{T}_{CI} = \begin{pmatrix} 46.496 \\ -33.407 \\ -61.116 \end{pmatrix} (mm) \quad (4.47)$$

4.6.3.3 Paramètres du filtre de Kalman pour la translation

Nous présentons maintenant les paramètres du filtre de Kalman utilisé pour l'estimation de la translation. Nous avons prêté une attention particulière à cette étape car elle représente le verrou principal de notre démarche de suivi hybride. Un réglage optimal du filtre est primordial pour assurer une estimation correcte des positions. Les choix des paramètres du filtre ont été adaptés de manière à minimiser les erreurs et permettre un meilleur compromis entre la précision et le temps de calcul.

Le vecteur de mesure du filtre de Kalman contient les accélérations selon les trois directions :

$$z = (\mathbf{A}_X \ \mathbf{A}_Y \ \mathbf{A}_Z)^T \quad (4.48)$$

Le vecteur d'état est représenté par :

$$x = (\mathbf{X} \ \mathbf{Y} \ \mathbf{Z} \ \mathbf{V}_X \ \mathbf{V}_Y \ \mathbf{V}_Z \ \mathbf{A}_X \ \mathbf{A}_Y \ \mathbf{A}_Z)^T \quad (4.49)$$

Le filtre permet la double intégration des accélérations pour estimer les positions de la MTi, la matrice de transition du filtre est linéaire et donnée par :

$$A = \begin{pmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 & \frac{(\Delta T)^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & \frac{(\Delta T)^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & \frac{(\Delta T)^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.50)$$

La matrice de mesure est exprimée par :

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.51)$$

La matrice de covariance d'erreur d'état est une matrice définie positive. Elle traduit l'erreur sur le modèle du processus. Les valeurs de la matrice représentent la confiance accordée au modèle d'état pour représenter le système modélisé. Dans notre cas, nous avons affecté la valeur de 10^{-4} à la matrice de covariance d'erreur, car notre modélisation mathématique établie à partir des équations du mouvement est précise. En outre, la matrice est diagonale car nous considérons que les erreurs sont indépendantes.

La matrice de covariance d'erreur du processus, est donnée par :

$$Q = 10^{-4}I_9 \quad (4.52)$$

où I_9 est matrice identité de dimension 9×9 . La matrice de covariance des erreurs de mesure exprime l'incertitude sur les données issues des capteurs utilisés. Cette matrice est diagonale définie positive. En s'appuyant sur la documentation technique de la MTi [166], nous avons considéré la matrice de covariance des erreurs de mesure suivante :

$$R = 10^{-4}I_3 \quad (4.53)$$

La matrice de covariance d'erreur exprime les erreurs d'estimation des états avec le filtre de Kalman. Des valeurs très grandes signifient que le filtre n'estime pas correctement les états. Pour notre expérimentation, nous avons choisi la matrice de covariance d'erreur suivante :

$$P = I_9 \quad (4.54)$$

Pour représenter une évolution réaliste de notre processus, nous avons rajouté le bruit de modèle. C'est un bruit blanc gaussien, modélisé par une distribution normale de covariance Q :

$$W = \mathcal{N}(0, Q) \quad (4.55)$$

Dans l'implémentation, le bruit de modèle est exprimé par :

$$W = \sqrt{Q} \cdot rand(9, N) \quad (4.56)$$

telle que $rand(9, N)$ est une matrice de valeurs aléatoires de dimension $9 \times N$. 9 représente le nombre d'états du modèle et N le nombre d'itérations du filtre.

De la même manière, nous avons modélisé le bruit de mesure. Ce bruit blanc gaussien est représenté par une distribution normale de covariance R :

$$V = \mathcal{N}(0, R) \quad (4.57)$$

Le bruit de mesure est simulé par le fonction suivante :

$$V = \sqrt{R} \cdot rand(3, N) \quad (4.58)$$

Le filtre de Kalman utilisé permet d'estimer, à partir d'une série de mesures, les accélérations, les vitesses et les positions de la MTi, en tenant compte des erreurs de mesure. Le filtre se décompose en deux étapes, prédiction à partir des informations passées et mise à jour des estimations à l'arrivée de nouvelles mesures. Le filtre calcule la meilleure combinaison de toutes ces sources d'information. Le calcul inclut l'estimation de la moyenne et la précision de la quantité estimée.

4.6.3.4 Évaluation du filtre avec des données synthétiques

Nous commençons l'évaluation du filtre avec des mesures synthétiques. Nous allons tester plusieurs types de signaux théoriques afin de vérifier le fonctionnement du filtre et le valider pour une utilisation avec des données réelles.

Nous rappelons que l'objectif du filtre est l'estimation des positions à partir des accélérations fournies par la MTi. Les équations de mesures du filtre, représentent la relation entre les accélérations selon les trois axes \mathbf{X} , \mathbf{Y} et \mathbf{Z} , et les états du système. Ces états expriment la modélisation cinématique du mouvement en translation de la MTi.

Test 1 :

Notre premier test consiste à considérer les accélérations constantes suivantes :

$$\begin{aligned}\mathbf{A}_X &= 1 \text{ (m/s}^2\text{)} \\ \mathbf{A}_Y &= -0.2 \text{ (m/s}^2\text{)} \\ \mathbf{A}_Z &= 0.5 \text{ (m/s}^2\text{)}\end{aligned}\tag{4.59}$$

Ces accélérations théoriques sont représentées sur la figure 4.14-a. Les estimations des accélérations, des vitesses et des positions sont représentées respectivement sur les figures 4.14-b, 4.14-c et 4.14-d.

Dans la figure 4.14-b, on remarque que le filtre de Kalman parvient à estimer avec grande précision les accélérations du modèle d'état. En effet, les courbes des estimations a posteriori des accélérations sont confondues avec les mesures réelles. Le bon paramétrage du filtre a conduit aussi à des estimations très correctes des accélérations a priori dont les courbes sont superposées sur les accélérations réelles. Les figures 4.14-c et 4.14-d, représentent respectivement l'intégrale simple et double de l'accélération. Les vitesses sont désignées par des rampes avec différentes pentes. Ceci est le résultat de l'intégrale simple du système d'équations 4.59 :

$$\begin{aligned}\mathbf{V}_X &= t \\ \mathbf{V}_Y &= -0.2 t \\ \mathbf{V}_Z &= 0.5 t\end{aligned}\tag{4.60}$$

De même les positions illustrées dans la figure 4.14-d sont représentées par des paraboles car le filtre réalise une intégration des vitesses. L'intégrale est donnée par :

$$\begin{aligned}\mathbf{X} &= 0.50 t^2 \\ \mathbf{Y} &= -0.10 t^2 \\ \mathbf{Z} &= 0.25 t^2\end{aligned}\tag{4.61}$$

Après ce premier test de simulation du filtre avec des accélérations constantes, nous concluons que le filtre de Kalman utilisé, permet une estimation correcte du vecteur d'état. En effet, tous les états du modèle sont bien estimés, les positions, les vitesses, les accélérations a priori et a posteriori. De plus, les paramètres que nous avons fixés (les covariances des erreurs et les bruits) sont adaptés au fonctionnement du filtre.

Test 2 :

Nous réalisons un deuxième test avec les mêmes accélérations que nous avons employées précédemment, sauf que cette fois-ci nous avons ajouté un bruit blanc gaussien aux mesures pour simuler des mesures réelles (figure 4.15-a). Ce bruit est représenté dans le filtre de Kalman par la matrice du bruit de mesure V , donnée par :

$$V = \mathcal{N}(0, R)\tag{4.62}$$

Nous avons fixé la covariance de l'erreur de mesure à :

$$R = 10^{-2} I_3\tag{4.63}$$

Les estimations obtenues des valeurs du vecteur d'état sont correctes. Le filtre de Kalman a permis le filtrage du bruit additionnel des mesures. L'intégration simple et double des accélérations désignent respectivement les vitesses et les positions de la MTi (figure 4.15-b, c, d).

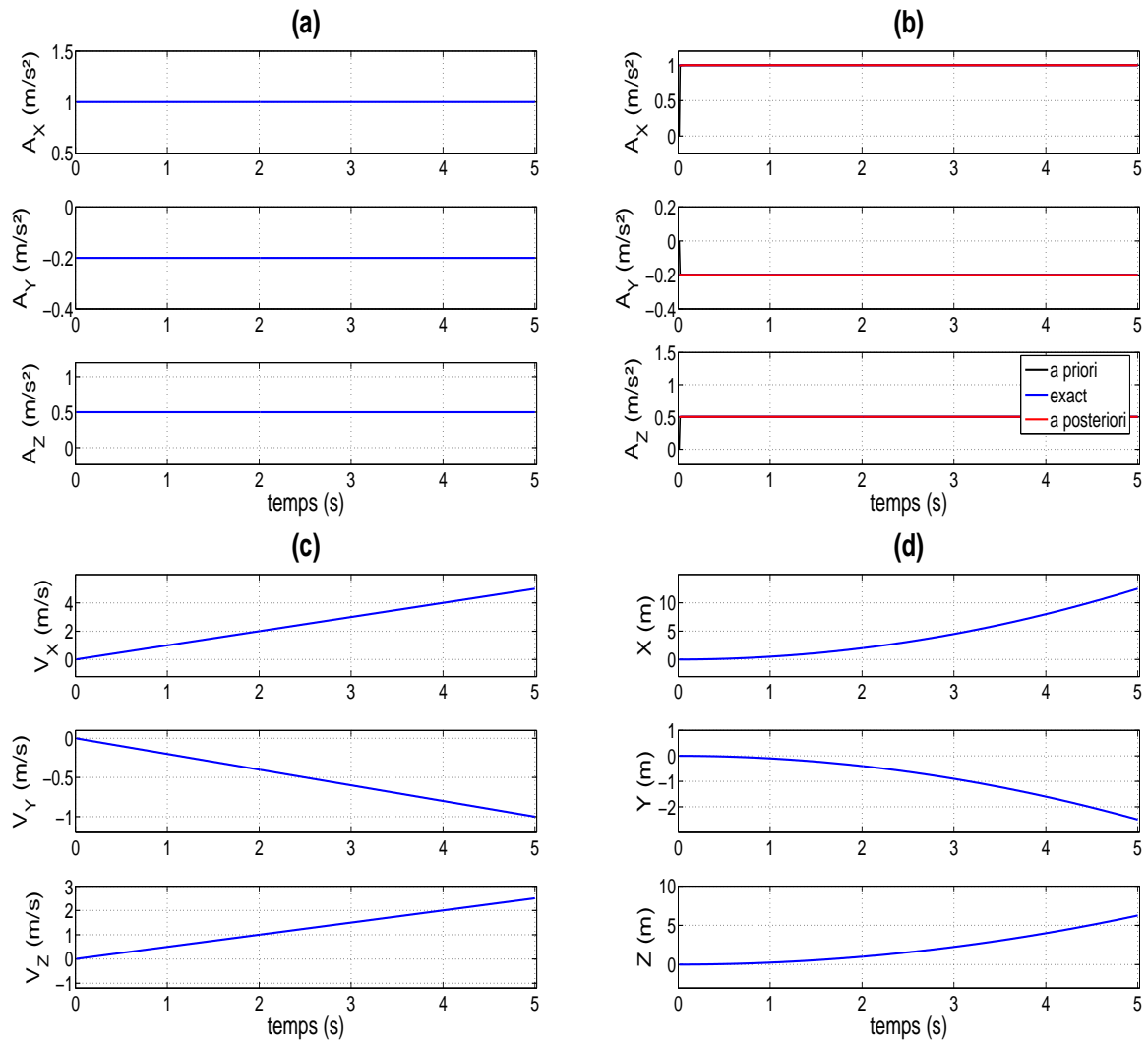


FIG. 4.14 – Simulation des accélérations constantes. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.

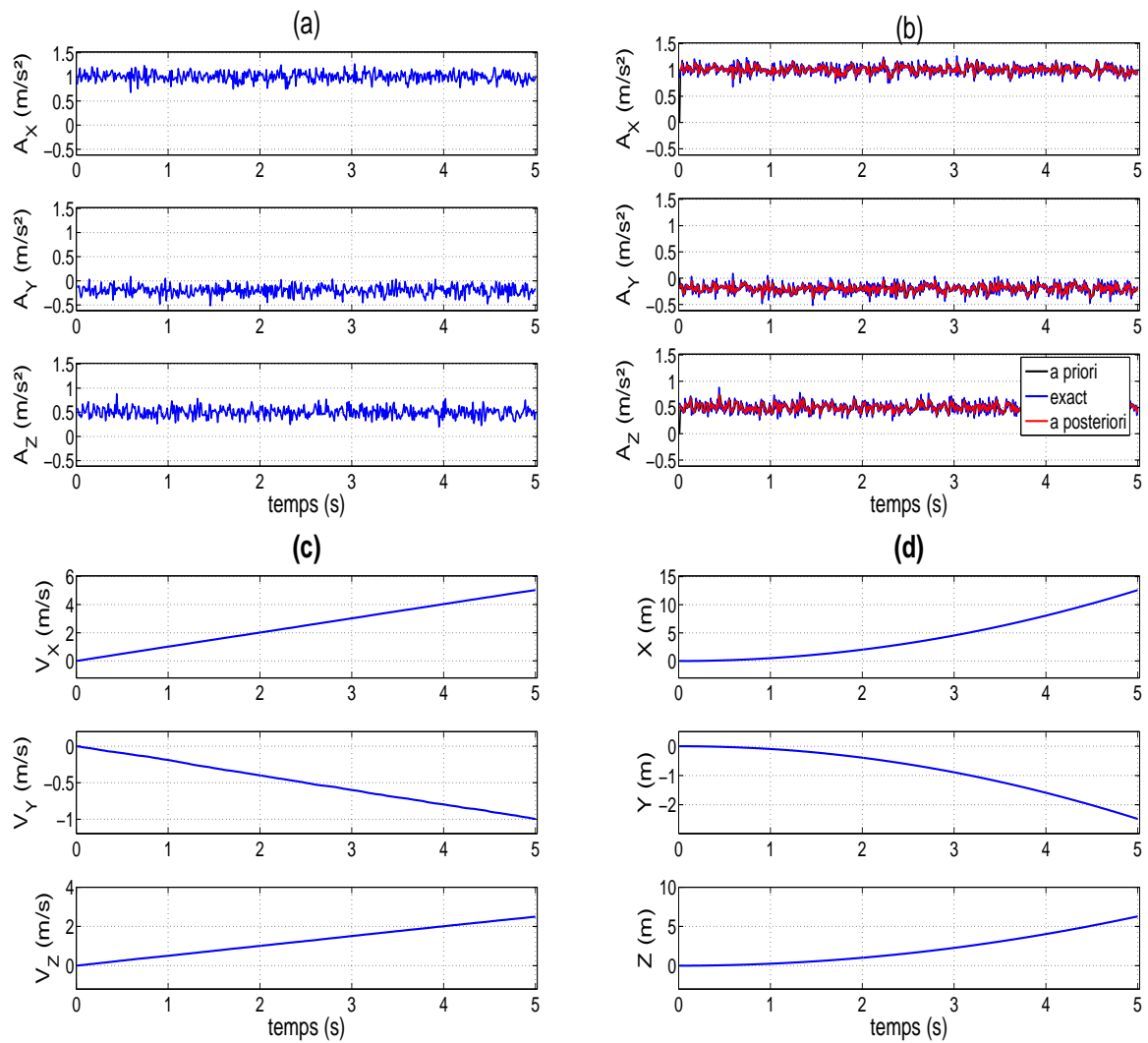


FIG. 4.15 – Simulation des accélérations constantes bruitées. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.

Test 3 :

Le troisième test avec les données synthétiques consiste à simuler des accélérations sous forme de signaux carrés (figure 4.16-a). Ces signaux sont exprimés par les équations suivantes :

$$\mathbf{A}_X = \begin{cases} 1 & \text{si } 0 < t \leq T/2 \\ 0 & \text{si } T/2 < t \leq T \end{cases} \quad (4.64)$$

$$\mathbf{A}_Y = \begin{cases} 0.5 & \text{si } 0 < t \leq T/2 \\ 0 & \text{si } T/2 < t \leq T \end{cases} \quad (4.65)$$

$$\mathbf{A}_Z = \begin{cases} -0.6 & \text{si } 0 < t \leq T/2 \\ 0 & \text{si } T/2 < t \leq T \end{cases} \quad (4.66)$$

où T représente la période du signal carré, $T = 50ms$.

Un bruit additif est ajouté aux mesures, ce bruit blanc gaussien représenté par une distribution normale de valeurs aléatoires, est défini à l'aide de la matrice V du le filtre de Kalman. Les valeurs de la matrice du bruit de mesure V , sont données par l'équation 4.62 et la covariance de l'erreur de mesure est la même que celle fournie dans l'équation 4.63.

La figure 4.16-b montre l'estimation des accélérations du vecteur d'état. Le filtre de Kalman a effectué une estimation très précise des signaux a priori et a posteriori, ce qui permet de conclure que les paramètres du filtre sont bien ajustés. Les vitesses représentant l'intégrale simple des signaux d'accélérations sont illustrées sur la figure 4.16-c. Enfin, les positions sont obtenues à partir de l'intégration des vitesses et elles sont représentées sur la figure 4.16-d.

Test 4 :

Le dernier test concernant les données synthétiques est la simulation des accélérations sous forme de signaux sinusoïdaux (figure 4.17-a). Ces signaux sont exprimés par les équations suivantes :

$$\begin{aligned} \mathbf{A}_X &= \sin\left(\frac{\pi}{8}t\right) \\ \mathbf{A}_Y &= 0.5 \sin\left(\frac{\pi}{8}t\right) \\ \mathbf{A}_Z &= 0.25 \sin\left(\frac{\pi}{8}t\right) \end{aligned} \quad (4.67)$$

La période du signal sinusoïdal est $T = 250ms$. Les accélérations sont entachées de bruit. Les valeurs de la matrice du bruit de mesure et sa covariance sont données par les équations 4.62 et 4.63. La figure 4.17-b illustre l'estimation des accélérations a priori et a posteriori du vecteur d'état. Les vitesses représentant l'intégrale simple des signaux d'accélérations sont illustrées sur la figure 4.17-c. Enfin, les positions sont obtenues à partir de l'intégration des vitesses et elles sont représentées sur la figure 4.17-d. Ce dernier test avec les accélérations sinusoïdales, confirme la validité et l'efficacité du filtre à estimer les positions par double intégrations. De plus, les paramètres du filtre sont correctement ajustés pour estimer le vecteur d'état et cela pour diverses formes de signaux de mesures.

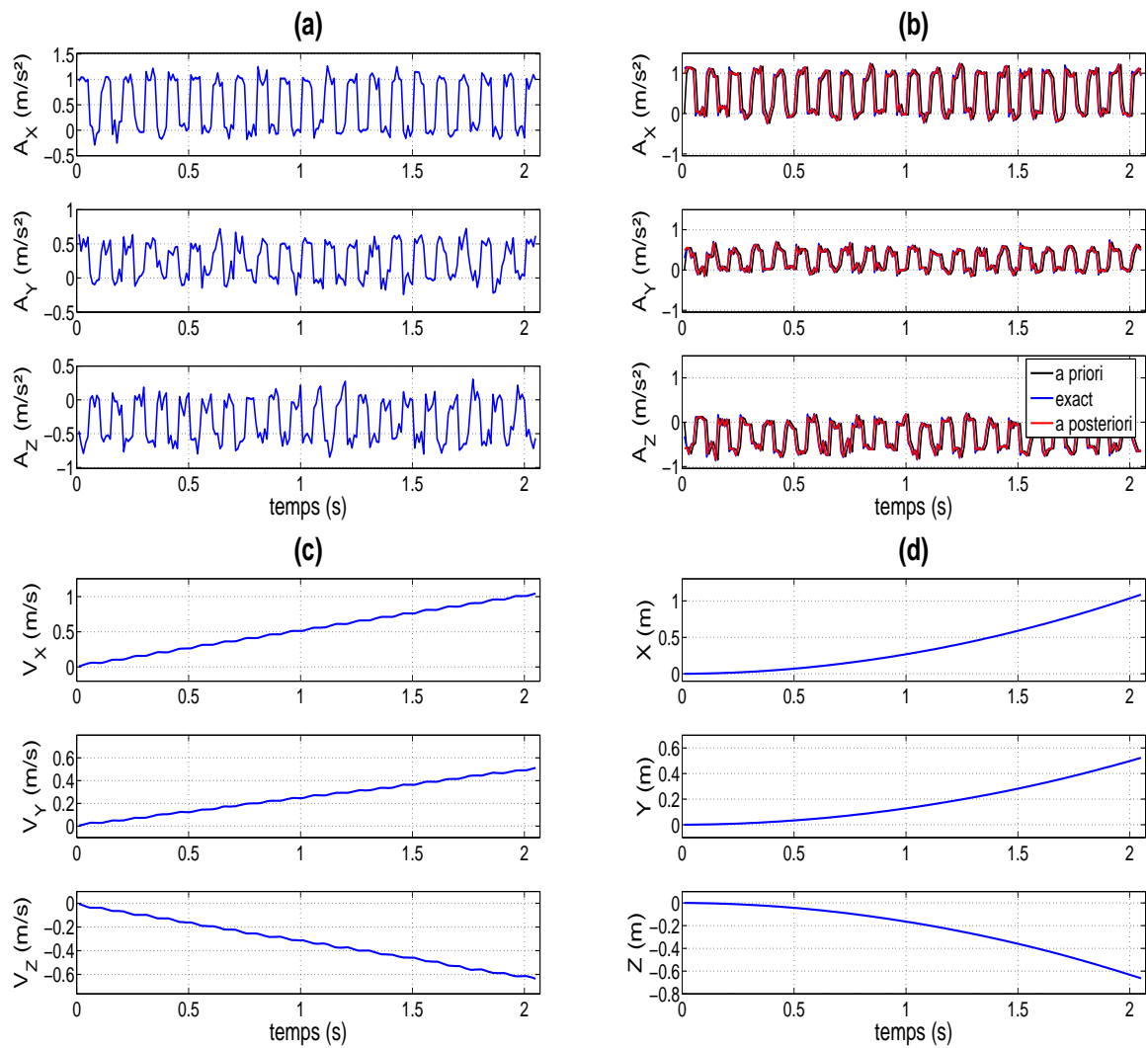


FIG. 4.16 – Simulation des accélérations sous forme d'un signal carré bruité. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.

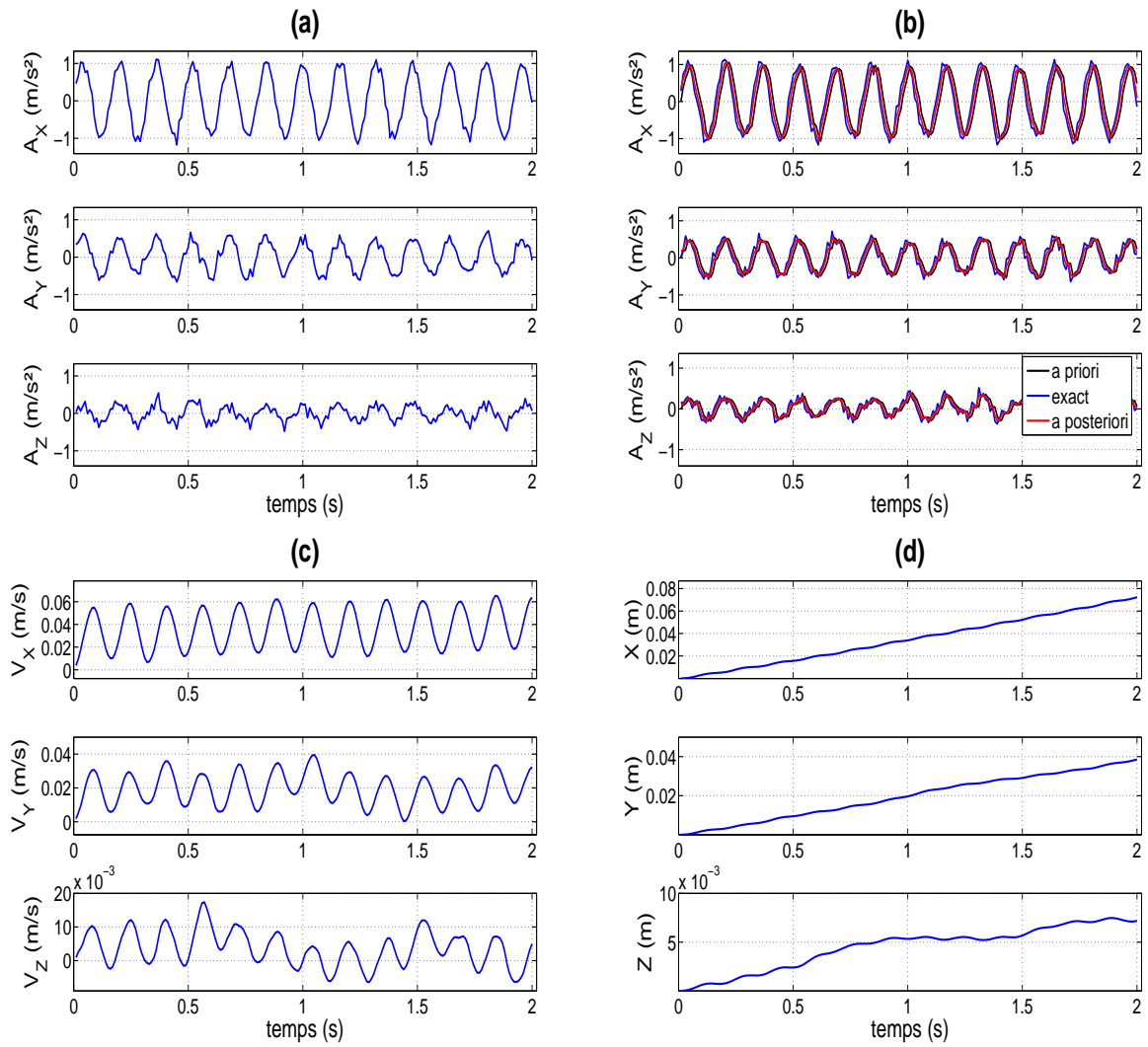


FIG. 4.17 – Simulation des accélérations sinusoïdales bruitées. (a) Accélérations. (b) Estimation des accélérations. (c) Estimation des vitesses. (d) Estimation des positions.

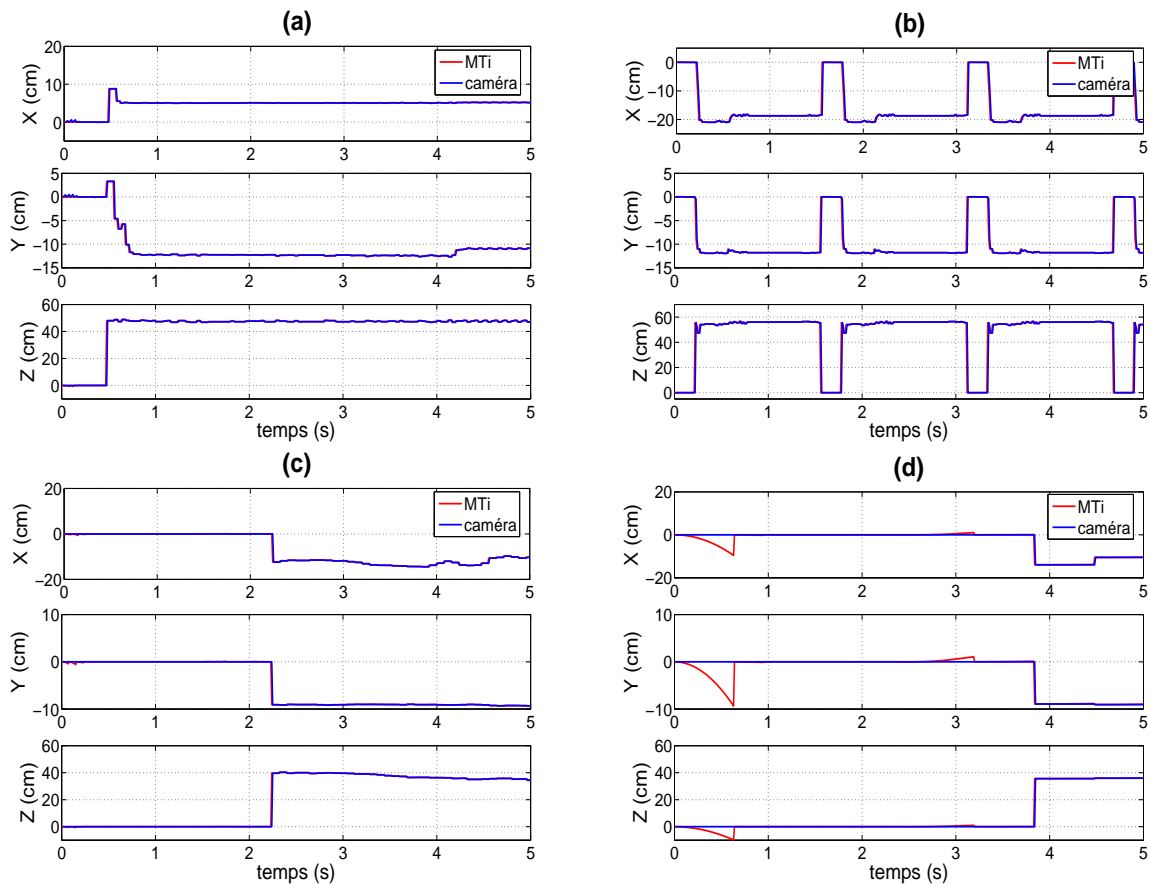
4.6.3.5 Évaluation du filtre avec des données réelles

Changement de fréquence de la MTi

Afin de déterminer la fréquence maximale pour laquelle la centrale inertielle ne présente pas de dérives, nous avons effectué des tests de tracking avec le capteur hybride, la MTi fournit des mesures pour l'estimation de la pose et la caméra permet de corriger la position de la MTi. Les tests sont réalisés avec les fréquences suivantes de la MTi : $25Hz$, $100Hz$, $200Hz$ et $400Hz$, la fréquence de la caméra étant de $25Hz$. Nous remarquons dans l'ensemble des figures 4.18-a,b,c,d, que malgré le changement de fréquence de la MTi, un suivi correct est réalisé. Cela s'explique par le fait que la caméra permet de corriger la position de la MTi même lorsque celle-ci est utilisée à la fréquence maximale de $400Hz$. Toutefois, sur la figure 4.18-d, on observe quelques dérives de la MTi, ceci est dû à la fréquence employée qui fournit des données aberrantes ce qui se traduit par des instabilités au niveau de l'estimation de la translation. Le tableau 4.3 traduit le nombre de données fournies par la MTi pour chaque donnée de la caméra.

Fréq. caméra	Nbr de données	Fréq. de la MTi	Nbr de données
25Hz	1	25Hz	1
25Hz	1	100Hz	4
25Hz	1	200Hz	8
25Hz	1	400Hz	16

TAB. 4.3 – Nombre de données en fonction des fréquences.

FIG. 4.18 – Changement de fréquence de la MTi. (a) $25Hz$. (b) $100Hz$. (c) $200Hz$. (d) $400Hz$.

Temps minimal de correction de données de la MTi

Nous avons vu dans le test précédent que quelque soit la fréquence de la MTi, et si la fréquence d'acquisition de la caméra est égale à $25Hz$, le suivi hybride est toujours maintenu sans dérive de la MTi. En effet, la caméra corrige la position de la MTi par une estimation correcte chaque $40ms$.

Le but de l'expérimentation suivante est d'estimer le temps minimal pour lequel la MTi dérive. Nous allons déterminer cet instant afin que la caméra puisse ré-initialiser correctement les positions calculées par la centrale inertielle. Les figures 4.19-a, b, c montrent que la MTi calcule les positions sans dérive jusqu'à $480ms$. Les courbes représentant l'estimation de la position par la MTi et la caméra sont superposées, ce qui explique que la MTi est corrigée constamment à un intervalle régulier. Nous remarquons aussi que la MTi a besoin d'un temps pour se stabiliser au départ et l'occurrence de quelques perturbations peut surgir lors du démarrage de la MTi. Ce phénomène est causé par un régime transitoire durant lequel la MTi restaure les paramètres de communication.

La figure 4.19-d illustre le cas de dérive de la MTi. Les positions calculées par la MTi sont représentées par un signal en dent de scie. L'accumulation des constantes d'intégration est la cause principale de la dérive. En effet, les positions calculées par la MTi sont le résultat de l'intégrale double des accélérations. Cette erreur de dérive est représentée par l'écart entre les positions de la MTi et celles de la caméra (figure 4.20).

Enfin, le temps pour lequel l'erreur de dérive devient prépondérante est estimé à $600ms$. Ce temps est déterminé après plusieurs tests de suivi et de recalage. Le temps minimal de correction étant $480ms$, donc si la centrale inertielle fonctionne à une fréquence de $100Hz$, la caméra doit corriger la position après 48 acquisitions de données d'accélération de la MTi.

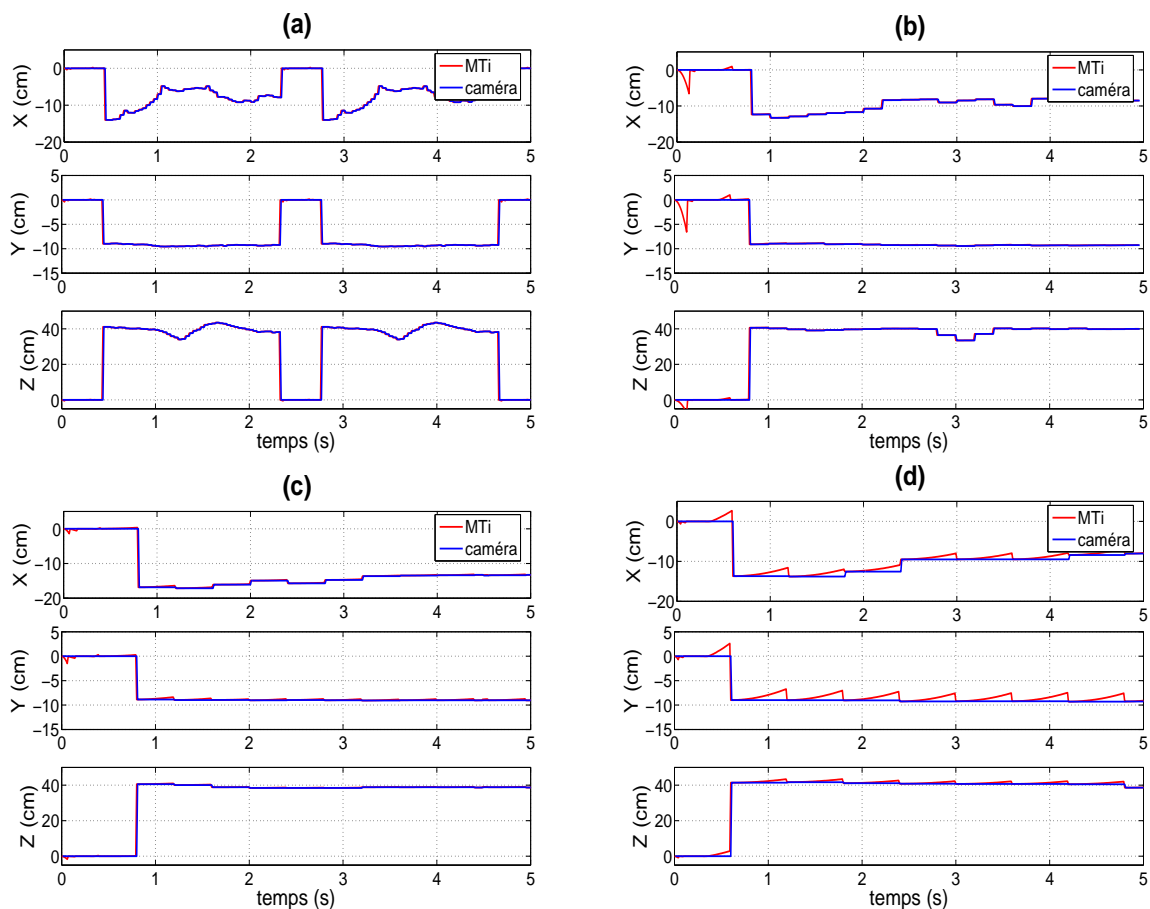


FIG. 4.19 – Changement de temps de correction des positions calculées par la MTi. (a) Correction après $40ms$. (b) Correction après $200ms$. (c) Correction après $480ms$. (d) Correction après $600ms$.

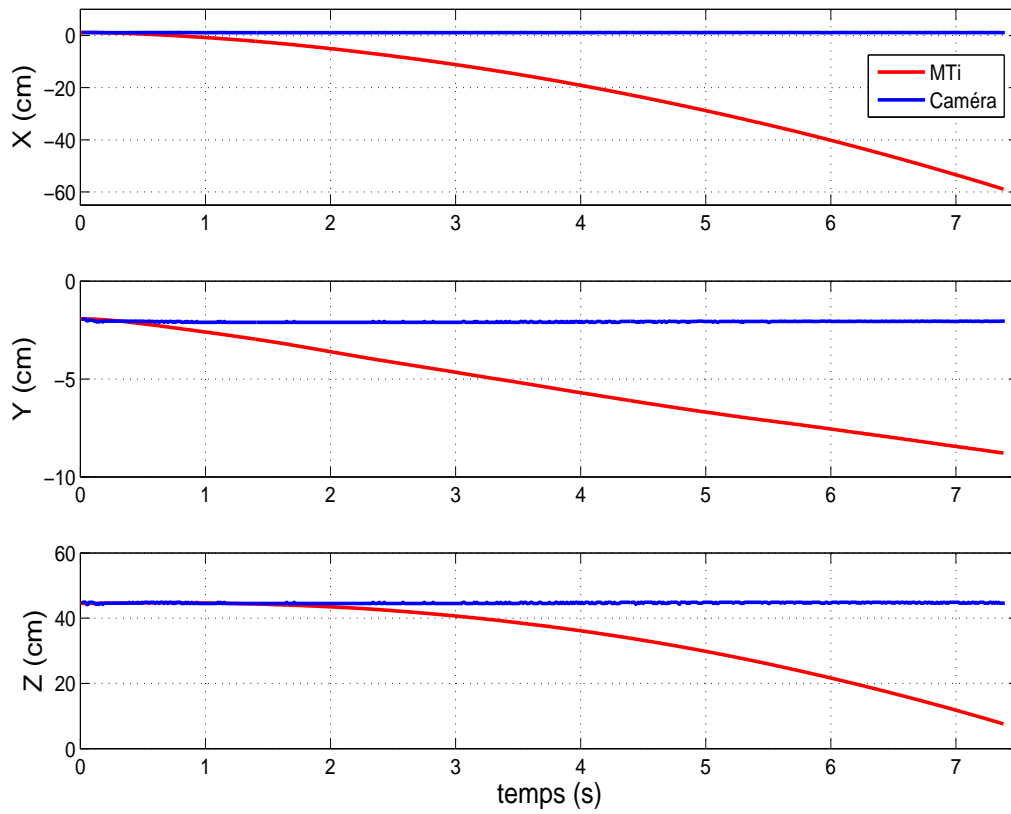


FIG. 4.20 – Dérive de la MTi en fonction du temps.

Suivi lors d'un mouvement uniforme

Après avoir déterminé le temps minimal pour lequel la MTi ne dérive pas, nous avons testé l'aptitude de la centrale inertielle à effectuer un suivi avec une petite vitesse. Nous avons fixé la fréquence de la MTi à $100Hz$ et celle de la caméra à $25Hz$. Cela signifie que le système de vision corrige la position de la MTi chaque $40ms$. Un mouvement quelconque avec cette basse vitesse du capteur hybride est réalisé. Nous remarquons que la MTi suit parfaitement le mouvement en translation de la caméra (figure 4.21). Les courbes d'estimation de position de la MTi et de la caméra sont superposées l'une sur l'autre ce qui permet de déduire que ce test représente les conditions optimales pour le suivi hybride. En effet, lorsque la dérive en translation de la MTi est corrigée, le système de tracking devient robuste et précis. Nous rappelons que la dérive dans notre cas, est due uniquement à la translation de la centrale inertielle car l'orientation est déterminée correctement et ne pose pas de problème particulier. Toutefois, un temps de correction de $40ms$ est peu exploitable dans le cas d'une occultation totale de la cible pour une durée plus longue.

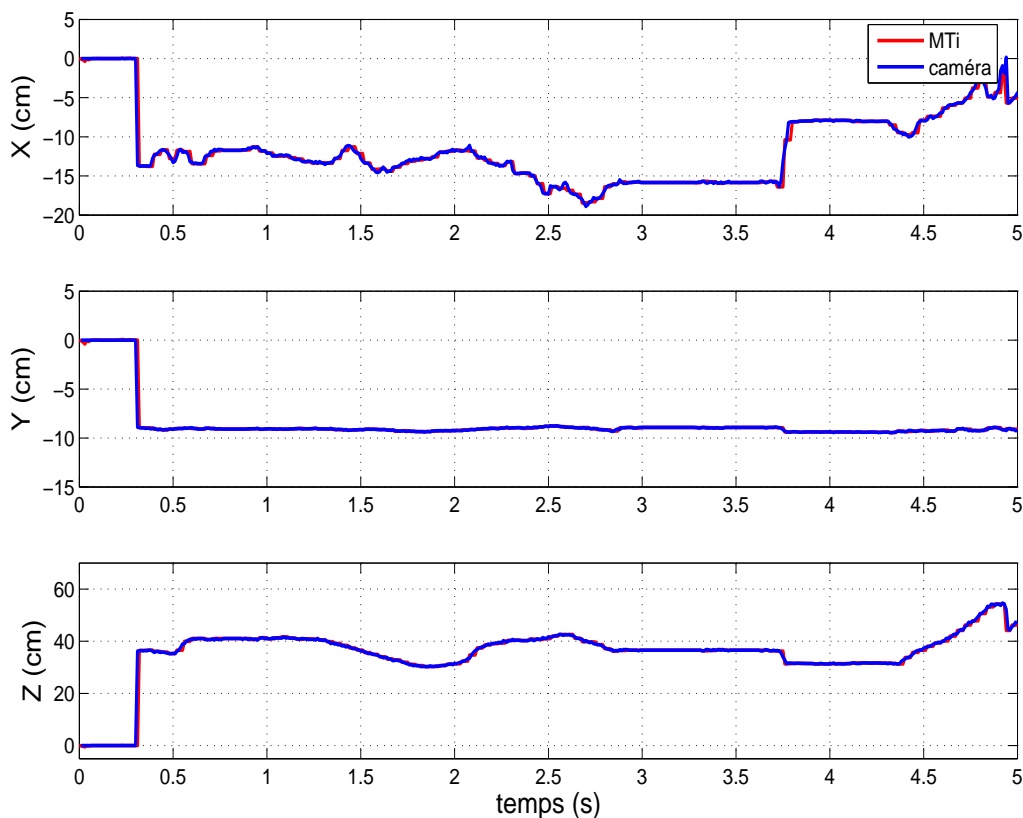


FIG. 4.21 – Suivi lors d'un mouvement uniforme.

Suivi lors d'un mouvement brusque

Dans cette expérimentation, un mouvement brusque du capteur hybride est réalisé. Nous avons utilisé deux fréquences de la MTi, $25Hz$ et $100Hz$, la fréquence de la caméra est égale à $25Hz$. Un mouvement aléatoire rapide est effectué dans cette expérience pour tester la capacité de tracking de la MTi dans le cas d'un mouvement brusque. Nous observons sur la figure 4.22-a que les deux courbes sont pratiquement superposées, nous pouvons dire, dans ce cas, que la MTi est adaptée aux mouvements brusques lorsque sa fréquence d'échantillonnage est de $25Hz$. En effet, à cette fréquence, la caméra corrige la translation de la MTi pour chaque acquisition de données d'accélération et le vecteur d'état du filtre de Kalman est initialisé avec les bonnes mesures de translation fournies par la caméra. Une seule intégration est calculée par le filtre pour déterminer les positions de la MTi ce qui explique le résultat obtenu.

Par contre, si on change la fréquence de la MTi à $100Hz$ (figure 4.22-b), on remarque que les deux courbes sont décalées l'une par rapport à l'autre. La MTi n'arrive pas à suivre correctement le mouvement de la caméra et un temps de latence est généré entre les deux capteurs. On observe aussi que cet écart entre les deux courbes est notable lorsque le mouvement change de direction, ce qui indique que le filtre ne parvient pas à bien estimer les positions si le mouvement n'est pas uniforme. Puisque le mouvement est très rapide, les mesures d'accélération de la MTi sont très variables et ceci entraîne une mauvaise estimation de la translation. Les accéléromètres à l'intérieur de la MTi sont sensibles aux mouvements ce qui se traduit par des valeurs disparates des accélérations et ceci génère des altérations dans l'estimation des positions par le filtre de Kalman.

Le filtre effectue une double intégration pour estimer les positions et parmi les hypothèses qui permettent l'estimation correcte de ces positions il y a le type de mouvement réalisé. Pratiquement, des mouvements lents et de petite durée sont les mieux adaptés pour déterminer les positions avec précisions [165].

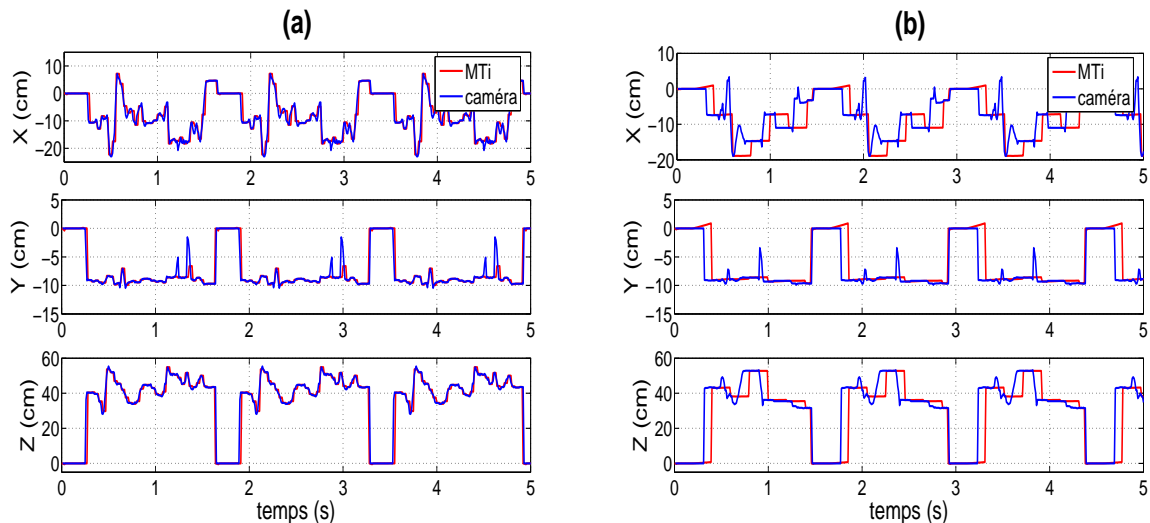


FIG. 4.22 – Suivi lors d'un mouvement brusque. (a) Fréquence de la MTi : $25Hz$. (b) Fréquence de la MTi : $100Hz$.

4.6.4 Résultats de suivi et de recalage

4.6.4.1 Cible visible

Nous présentons maintenant les résultats de recalage obtenus par l'estimation de la pose de la MTi. Dans cette expérience, la cible est visible par la caméra pour que cette dernière puisse estimer la pose au même temps que la MTi. La fréquence d'acquisition de la caméra est de $25Hz$ et celle de la MTi est fixée à $100Hz$. Ce choix permet d'avoir des conditions favorables quant au temps de correction de la translation de la MTi. Le but de cette expérience est de vérifier la précision de l'alignement du cube virtuel en utilisant les données de la centrale inertielle.

Sur la figure 4.23 un cube virtuel est superposé sur la cible en utilisant la pose calculée par la MTi. La caméra est utilisée aussi pour déterminer la pose, cependant, les valeurs affichées nous permettent de comparer les paramètres de la pose estimée par les deux capteurs pour évaluer la précision du recalage. Une première analyse a permis de vérifier que la pose estimée par la MTi est précise car l'objet virtuel est correctement aligné sur la cible quelque soit son positionnement dans l'espace.

Dans un second temps, nous nous sommes intéressés aux valeurs numériques des paramètres de la pose estimée par les deux capteurs. On remarque que les angles de rotations de la MTi sont légèrement différents de ceux de la caméra. Un écart de quelques degrés apparaît entre les paramètres de rotation de la MTi et de la caméra. Cet écart est dû aux erreurs de calibration du repère global de la centrale inertielle (section 4.3.3.1). Par ailleurs, les valeurs de translation de la MTi sont égales à celles de la caméra. En effet, les images de la figure 4.23 ont été acquises lorsque le capteur hybride est immobile, ce qui implique que l'accélération de la MTi est nulle. Par conséquent, les positions estimées par la centrale inertielle sont égales aux positions initiales ($\mathbf{X} = \mathbf{A}t^2 + \mathbf{X}_0$, si $\mathbf{A} = 0$ alors $\mathbf{X} = \mathbf{X}_0$). Ces valeurs initiales de positions sont les composantes de translation déterminées par la caméra, d'où l'égalité entre les paramètres de translation des deux capteurs.

Pour quantifier les erreurs de translation et de rotation, nous avons calculé 40 pose du capteur hybride lorsque la cible est visible et identifiée par la caméra. Les écarts entre les composantes de translations sont illustrés sur la figure 4.24-a. Cette courbe représente l'erreur en translation, elle exprime la différence entre les positions de la caméra et celles de la MTi par les relations suivantes :

$$\begin{aligned} E_X &= \mathbf{X}_{cam} - \mathbf{X}_{MTi} \\ E_Y &= \mathbf{Y}_{cam} - \mathbf{Y}_{MTi} \\ E_Z &= \mathbf{Z}_{cam} - \mathbf{Z}_{MTi} \end{aligned} \quad (4.68)$$

Nous avons ensuite calculé les erreurs quadratiques moyennes des composantes de la translation déterminées par les formules suivantes :

$$\begin{aligned} EQM_X &= \frac{1}{N} \sqrt{\sum_{i=1}^N (\mathbf{X}_{cam_i} - \mathbf{X}_{MTi_i})^2} \\ EQM_Y &= \frac{1}{N} \sqrt{\sum_{i=1}^N (\mathbf{Y}_{cam_i} - \mathbf{Y}_{MTi_i})^2} \\ EQM_Z &= \frac{1}{N} \sqrt{\sum_{i=1}^N (\mathbf{Z}_{cam_i} - \mathbf{Z}_{MTi_i})^2} \end{aligned} \quad (4.69)$$

Les valeurs numériques des erreurs quadratiques moyennes sont :

$$\begin{aligned} EQM_X &= 1.7299e - 004(cm) \\ EQM_Y &= 1.7343e - 004(cm) \\ EQM_Z &= 1.4622e - 004(cm) \end{aligned} \quad (4.70)$$

Les erreurs obtenues sont relativement faibles car la fréquence de la MTi est égale à $100Hz$ ce qui signifie qu'une correction de position est apportée par la caméra pour chaque $40ms$.

Les erreurs de rotation ont été estimées de la même manière que les erreurs de translation, nous avons calculé la différence entre les angles de rotations de la MTi avec ceux de la caméra en utilisant les relations suivantes :

$$\begin{aligned} E_{\phi} &= \phi_{cam} - \phi_{MTi} \\ E_{\theta} &= \theta_{cam} - \theta_{MTi} \\ E_{\psi} &= \psi_{cam} - \psi_{MTi} \end{aligned} \quad (4.71)$$

ϕ , θ , ψ représentent respectivement les angles de roulis, tangage et lacet. La figure 4.24-b illustre ces erreurs de rotation. Dans cette figure, la variance de l'erreur est approximativement égale à 5° pour les angles du roulis et du tangage alors qu'elle est $< 1^{\circ}$ pour le lacet. Cette différence est causée par le type de mouvement effectué, en l'occurrence le capteur hybride n'a pas subi une grande rotation autour du lacet ce qui explique la faible variance de l'angle du lacet. Les erreurs quadratiques moyennes des angles de rotation sont déterminées par les expressions suivantes :

$$\begin{aligned} EQM_{\phi} &= \frac{1}{N} \sqrt{\sum_{i=1}^N (\phi_{cam_i} - \phi_{MTi_i})^2} \\ EQM_{\theta} &= \frac{1}{N} \sqrt{\sum_{i=1}^N (\theta_{cam_i} - \theta_{MTi_i})^2} \\ EQM_{\psi} &= \frac{1}{N} \sqrt{\sum_{i=1}^N (\psi_{cam_i} - \psi_{MTi_i})^2} \end{aligned} \quad (4.72)$$

Les valeurs numériques de ces erreurs sont :

$$\begin{aligned} EQM_{\phi} &= 0.3674(deg) \\ EQM_{\theta} &= 0.1274(deg) \\ EQM_{\psi} &= 0.0775(deg) \end{aligned} \quad (4.73)$$

Ces erreurs sont dues à la calibration du repère global de la MTi. En effet, la définition de ce repère se fait par superposition du repère local de la centrale inertielle sur le repère monde de la cible, un mauvais alignement entre ces deux repères génère des erreurs dans l'estimation de l'orientation de la MTi.

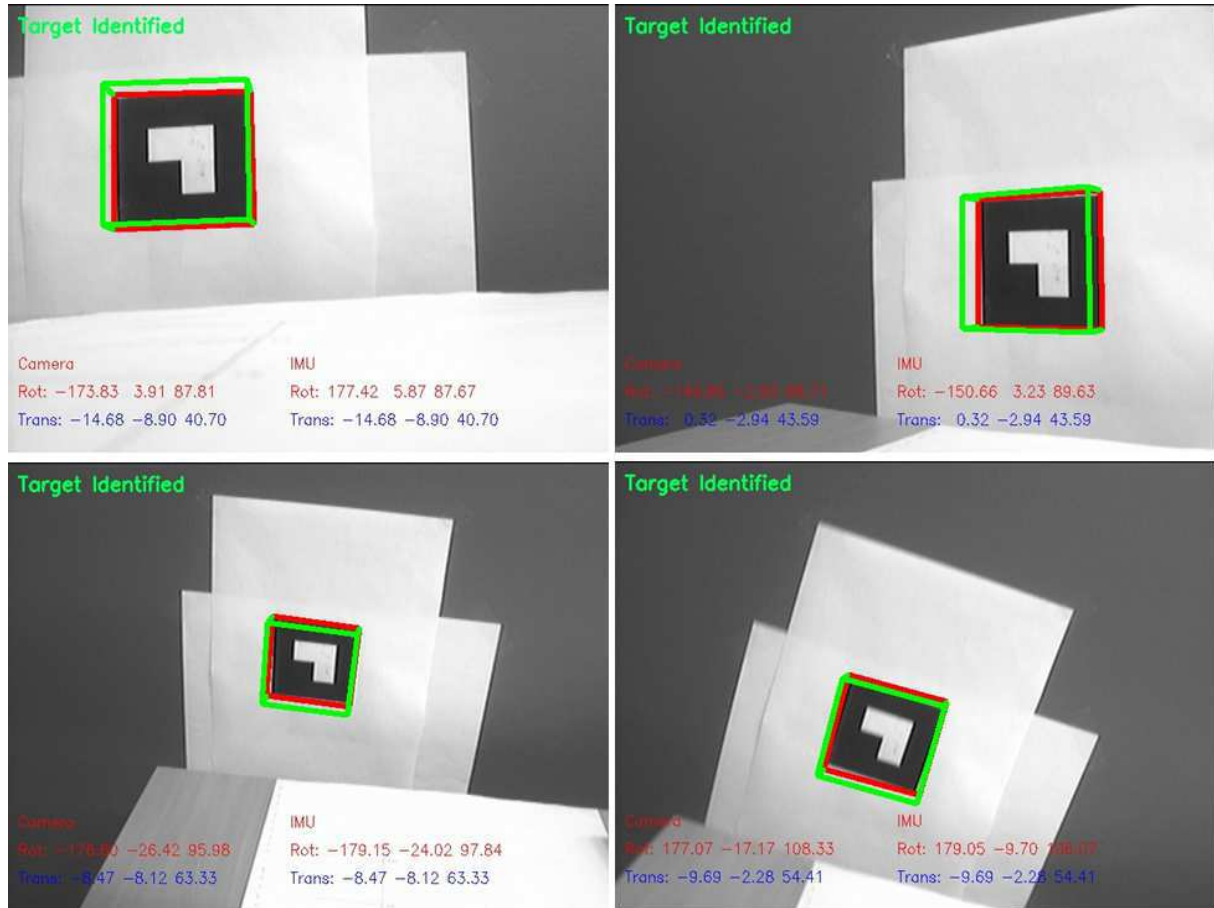


FIG. 4.23 – Résultats de suivi et de recalage avec le capteur hybride lorsque la cible est visible.

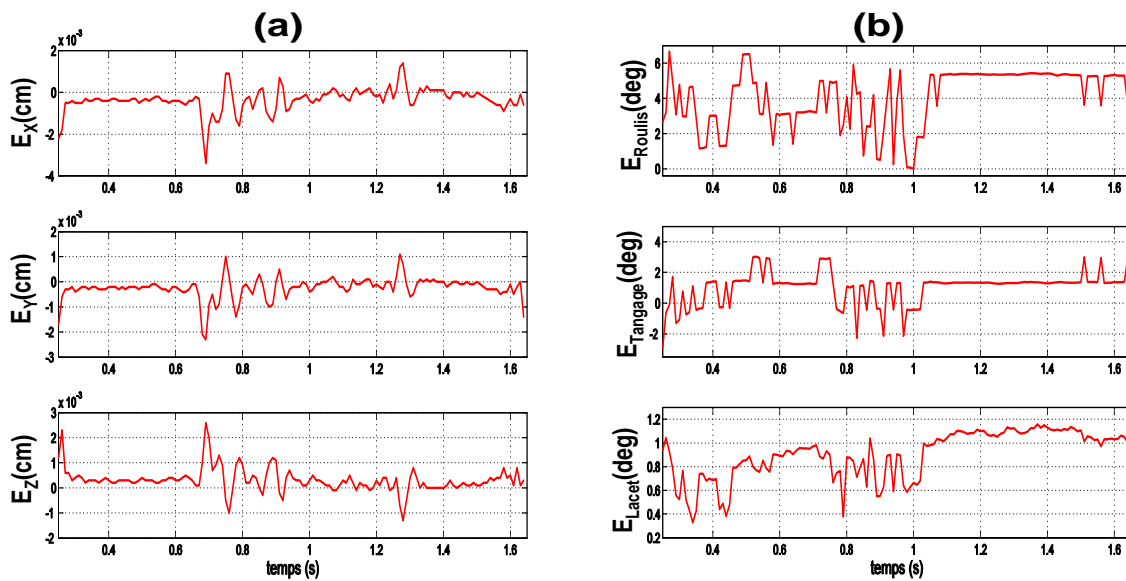


FIG. 4.24 – Erreurs d'estimation de la pose entre la caméra et la MTi. (a) Erreurs de translation. (b) Erreurs de rotation.

4.6.4.2 Cible occultée

Nous avons testé le filtre hybride dans le cas d'occultation de la cible. Nous avons occulté partiellement et totalement la cible pour vérifier l'aptitude de la centrale inertielle à estimer la pose. Les résultats illustrés sur la figure 4.25 montre que la MTi calcule correctement la position et l'orientation de la cible et permet un recalage correct du cube virtuel sur celle-ci, alors que la caméra ne parvient pas à déterminer la pose car l'objet d'intérêt n'est pas identifié.

Dans cette expérience le premier module de suivi de cible basé sur le système de localisation par la vision est mis en échec car la cible n'est pas détectée. Ce module de tracking est déclenché si la cible est entièrement visible par la caméra, toutefois, si un point caractéristique de la cible n'est pas visible alors notre algorithme d'identification et de suivi ne fonctionnera plus. Le module de tracking hybride prend le relais et permet le recalage du cube virtuel sur la cible. En effet, la centrale inertielle estime la pose de son repère local par rapport au repère du monde, et puisque le système hybride est calibré, alors la pose de la MTi est transcrite dans le repère de la caméra et un recalage du cube virtuel est réalisé.

Le suivi hybride permet de gérer les occultations des cibles lorsque celles-ci ne sont pas reconnaissables par le système de vision. La condition de visibilité totale ou partielle de la cible n'est pas requise pour ce module de tracking. Le système multi-capteurs apporte une solution intéressante quant à la gestion d'occultations, cependant, les erreurs de dérives de la translation ne permettent pas une utilisation de longue durée surtout lorsque le système hybride est en mouvement.

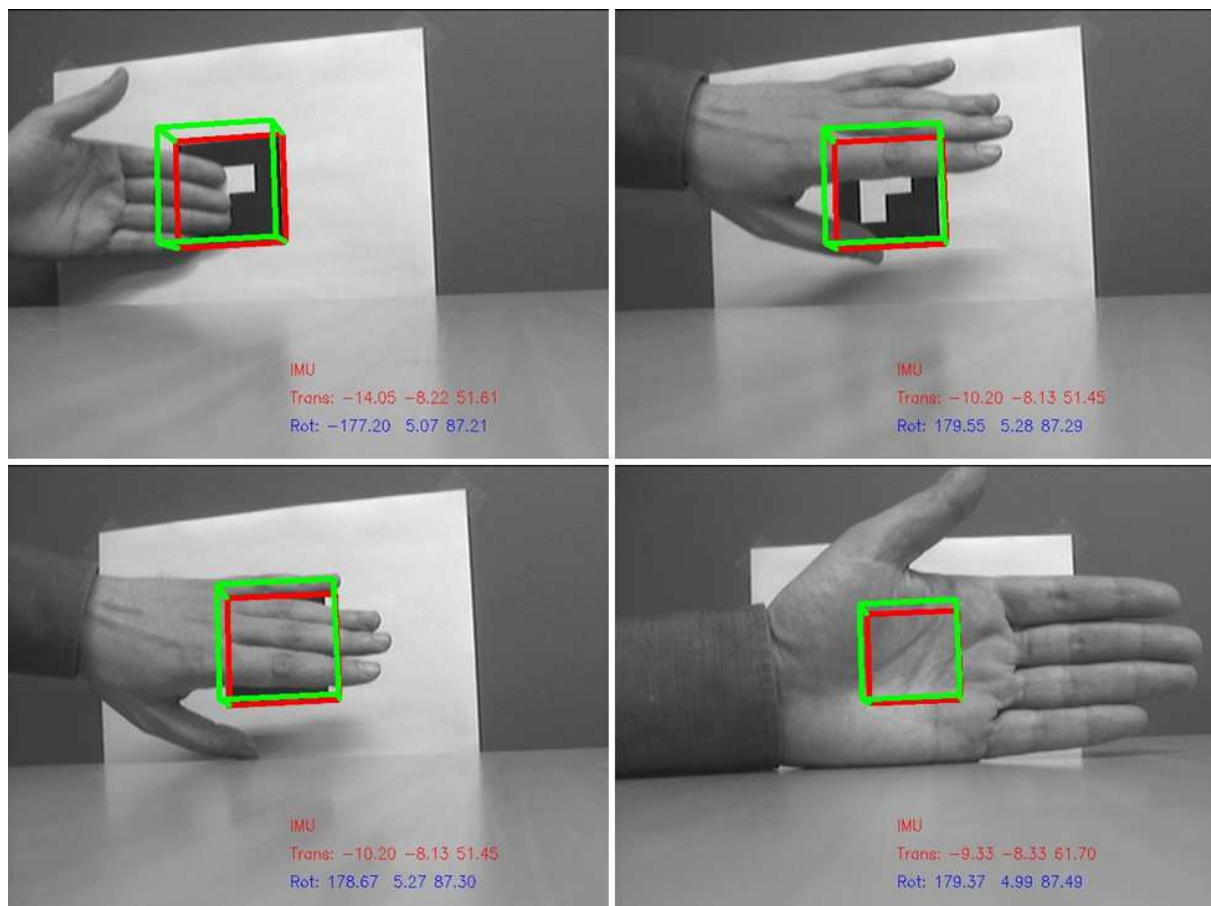


FIG. 4.25 – Résultats de suivi et de recalage avec le capteur hybride lorsque la cible est occultée.

4.6.4.3 Flou de mouvement

Afin de tester la robustesse de notre système face au flou d'image généré par les mouvements brusques du capteur hybride, nous avons réalisé une expérimentation qui consiste à déplacer le dispositif de tracking très rapidement autour de la cible. Nous remarquons sur la figure 4.26 que la cible est suivie malgré qu'elle n'est pas détectée par la caméra car la qualité de l'image est dégradée. Toutefois, la MTi prend le relais de la caméra et permet l'estimation de la pose et le recalage du cube virtuel.

Cette expérience montre un autre avantage de l'utilisation de la centrale inertielle. Lors d'un flou de mouvement, le système de vision est peu robuste aux acquisitions non-instantanées des images. Si la fréquence de déplacement du capteur hybride est supérieure à celle de la fréquence de la caméra, ce phénomène apparaît. Puisque la centrale inertielle ne requiert pas la visibilité de la cible pour se localiser dans l'espace, le problème du flou de mouvement ne constitue pas un problème pour l'estimation de la pose par la MTi.

Toutefois, si le mouvement de déplacement du dispositif est effectué à une fréquence supérieure à celle de la MTi, une latence est générée et cela se traduit par un décalage du positionnement du cube virtuel sur la cible. Les deux images d'en haut de la figure 4.26 illustrent ce problème. On aperçoit un certain écart entre l'entité virtuelle et l'objet d'intérêt. Par contre, sur les deux autres images (celles d'en bas de la figure 4.26), les modèles virtuels 3D sont bien alignés sur la cible car la fréquence de déplacement est inférieure à celle de la MTi.



FIG. 4.26 – Résultats de suivi et de recalage avec le capteur hybride en présence de flou de mouvement.

4.6.4.4 Gestion d'occultations partielles et totales de la cible

Afin de tester le système global de gestion d'occultations, nous avons intégré le programme de suivi robuste basé sur l'algorithme RANSAC développé dans le chapitre 3 avec l'algorithme de suivi hybride que nous avons développé dans ce chapitre (figure 4.27). En effet, lorsque la cible est partiellement occultée l'algorithme RANSAC permet de suivre les points d'intérêts de la cible et un calcul de pose est effectué pour recalculer les cubes virtuels sur les cibles. Toutefois, si les deux cibles sont occultées totalement, l'algorithme robuste ne parvient plus à faire le suivi et par conséquent le module de tracking hybride avec la MTi démarre et permet la localisation de la caméra et ainsi le recalage d'objets virtuels.

La figure 4.28 résume les différents modules de notre système développé tout au long de cette thèse. Le diagramme synoptique de cette figure représente une architecture multimodale de suivi et de gestion d'occultations. Notre système de tracking permet de surmonter les problèmes de suivi de marqueurs dans des conditions variées d'un environnement de réalité augmentée. Cette architecture est constituée d'un module de tracking de cibles codées basé sur un algorithme hybride d'estimation de pose. Pour pallier les occultations partielles des cibles, un autre module représentant une extension de notre méthode de localisation spatio-temporelle est élaboré. Ce module réalise le suivi de points caractéristiques des cibles dans l'image. Cette étape se fonde sur l'étude des propriétés du suivi (détection de points, mise en correspondance 2D-2D et 2D-3D) et sur l'adaptation et la gestion des différentes conditions expérimentales. Enfin, une partie de suivi multi-capteurs composé d'un dispositif hybride de tracking permet de gérer les occultations totales des cibles.

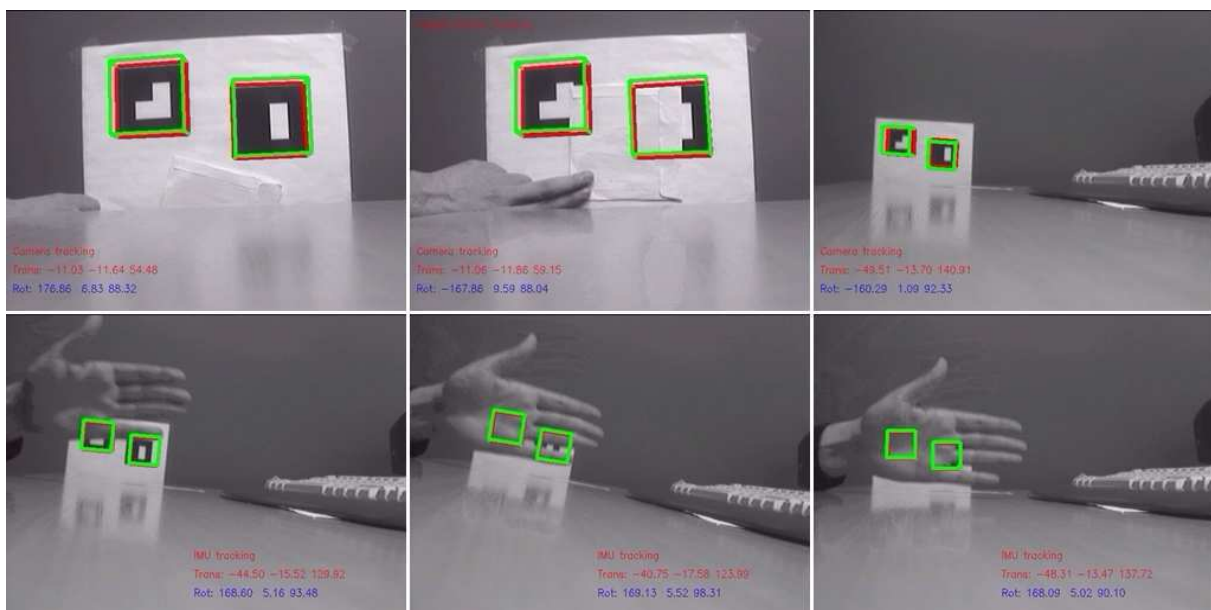


FIG. 4.27 – Suivi avec le capteur hybride et gestion d'occultations partielles et totales.

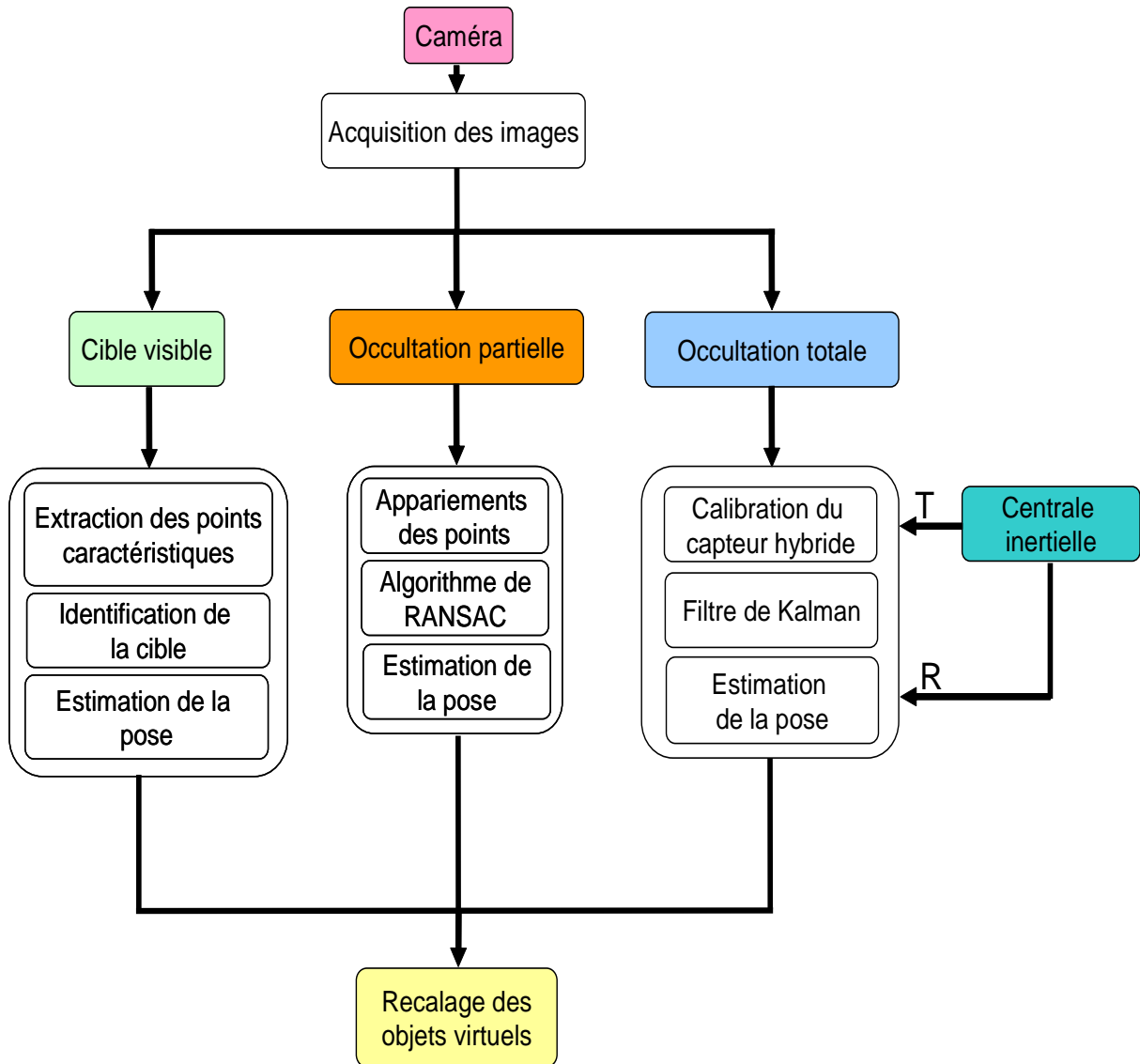


FIG. 4.28 – Diagramme de suivi et de gestion d’occultations avec le système global de tracking.

4.7 Conclusion

Dans ce chapitre, nous avons présenté un système multi-capteurs pour l'estimation de la pose de la caméra et le recalage virtuel en réalité augmentée. Notre approche hybride est basée sur la collaboration entre une centrale inertielle et un système de vision composé d'une caméra, pour réaliser le suivi de cibles et la gestion d'occultations en temps réel.

Nous avons commencé par détailler la procédure de calibration du capteur hybride, ensuite, nous avons décrit le réglage des paramètres du filtre de Kalman pour l'estimation de la translation. Ce module d'estimation de la translation a été étudié et testé avec des données synthétiques et réelles de la MTi. La dérive de la centrale inertielle a constitué notre principal problème lors des expérimentations. Cette erreur est due à l'accumulation des constantes d'intégration du signal d'accélération pour estimer la position. Des évaluations ont été effectuées pour caractériser cette dérive de la MTi et déterminer le temps minimal de correction par la caméra. Par ailleurs, un réglage optimal et adaptatif du filtre de rotation de la MTi a permis d'avoir des orientations précises et ceci a dispensé de mener une étude approfondie sur ce module.

La méthode hybride proposée contenait deux canaux complémentaires de suppléance de données, ce module combine à la fois des données de tracking hautes fréquence issues de la centrale inertielle et des données basses fréquences fournies par la caméra. Ce système hybride exploite les avantages des deux capteurs pour améliorer la stabilité, la robustesse et la précision du suivi. La centrale inertielle est rapide mais peu précise à cause de la dérive dans le calcul des positions à partir des accélérations, et la caméra est précise mais fonctionne à une basse fréquence ajouter à cela le temps de traitement qu'exigent les algorithmes implémentés pour exploiter les images.

L'approche hybride présente une originalité quant à l'utilisation de la MTi pour estimer la translation de la caméra. Toutefois, notre méthode ne peut être appliquée que pour des occultations totales de très courtes durée. En effet, les dérives en translation de la centrale inertielle empêche une estimation correcte des positions, la caméra doit garder le contrôle sur la MTi afin de ré-initialiser ses positions à un intervalle régulier.

Un protocole expérimental a été élaboré pour caractériser la centrale inertielle et paramétrer le module de suivi hybride. Des expériences dans un environnement réel de l'utilisateur ont été effectuées, les résultats obtenus ont montrés que notre dispositif hybride de tracking apporte des améliorations significatives pour les systèmes de réalité augmentée basés vision en donnant à l'utilisateur plus de libertés du mouvement (vitesse de déplacement) et sans se soucier de la visibilité ou l'occultation de la cible. Notre système hybride a été testé dans divers scénarios de suivi, quand les cibles sont visibles, occultées, lors d'un mouvement brusque.

Enfin le module de tracking hybride a été intégré dans une architecture globale de suivi et de gestion d'occultations en insérant la partie de suivi robuste avec RANSAC. Le système multimodal utilise différents capteurs et techniques pour permettre le suivi de cibles lorsqu'elles sont visibles, partiellement ou totalement occultées dans des conditions réelles de l'environnement.

Conclusion générale

Dans cette thèse, nous nous sommes intéressés au suivi robuste et multi-capteurs des cibles dans les systèmes de réalité augmentée. Ce domaine de recherche comporte plusieurs verrous technologiques et scientifiques que nous avons identifiés et qui se rapportent souvent à la précision, la robustesse et la rapidité du dispositif de tracking. Nous avons passé en revue plusieurs systèmes de réalité augmentée réalisés ces dernières années. L'étude de ces systèmes a permis d'en extraire les principales fonctionnalités et les architectures les plus pertinentes pour le suivi robuste en temps réel. De plus, nous avons analysé les solutions techniques et scientifiques proposées dans la littérature pour pallier les différents verrous abordés dans cette thèse.

L'objectif de la thèse est de développer une architecture multimodale de suivi de cibles et de gestion d'occultations pour une application de réalité augmentée. L'approche multimodale proposée consiste à combiner différents capteurs et techniques, alliés en fonction des conditions réelles de l'environnement. Nous nous sommes intéressés aux problèmes d'alignement visuel des objets virtuels sur la scène réelle et au suivi robuste et multi-capteurs des cibles codées dans les séquences d'images. Dans ce contexte, nous avons proposé des solutions originales pour résoudre les problèmes liés à l'utilisation du système réalisé d'une part et d'autre part, analyser son fonctionnement pour une mise en oeuvre efficace de ce dispositif de tracking.

Le travail réalisé a permis de lever le verrou sur certaines limitations liées à la réalisation d'un système de suivi en réalité augmentée. Nous avons pu surmonter dans cette thèse, les problèmes de tracking de marqueurs en élaborant une architecture de suivi en présence d'occultations. L'architecture proposée contient un module de tracking de cibles codées basé sur un algorithme hybride d'estimation de pose. Pour résoudre le problème d'occultations partielles des cibles, un deuxième module de suivi de points caractéristiques a été proposé. Cette étape est basée sur l'étude des propriétés du suivi (détection de points, mise en correspondance 2D-2D et 2D-3D) dans différentes conditions expérimentales. Enfin, nous avons inclus dans notre architecture de suivi une partie de suivi multi-capteurs. Ce dernier module composé d'une centrale inertielle et d'une caméra, permet le suivi et la gestion d'occultations totales des cibles. La suppléance des données cinématiques de la centrale inertielle et des images de la caméra ouvre de nouvelles perspectives pour l'amélioration des systèmes de localisation basés vision en réalité augmentée. Cependant, de tels systèmes multi-capteurs présentent des limites qui sont parfois mal cernées. Dans ce contexte, il est important d'ajuster les paramètres intrinsèques des capteurs utilisés afin de caractériser plus finement la dualité et la collaboration des données hétérogènes issues des capteurs.

Dans la première partie de notre travail, nous avons présenté un système de localisation par la vision qui repose sur l'utilisation d'un algorithme d'échantillonnage de zones rectangulaires de l'image que nous avons développé. Cette méthode basée sur les marqueurs visuels permet de localiser en temps réel les objets d'intérêts dans les images. Pour le suivi de cibles dans les images, nous avons introduit des algorithmes spécifiques d'estimation de pose pour améliorer la précision et la rapidité de la localisation en combinant une méthode analytique directe développée dans notre laboratoire et une méthode itérative de calcul de la pose qui est le filtre de Kalman. L'approche proposée peut être

améliorée en compensant les distortions radiales de l'image qui représentent une source d'erreur pour l'estimation des distances faites par le système. Ensuite, nous avons effectué une étude comparative pour évaluer les performances de plusieurs systèmes de localisation. Cette étude a porté sur divers critères de performances et les tests expérimentaux obtenus ont prouvés que notre approche hybride de calcul de pose fournit une solution intéressante quant à la localisation de la caméra en utilisant des cibles codées.

Nous avons développé par la suite une méthode de tracking robuste permettant de maintenir les augmentations sur les objets réels malgré les changements de conditions de détection de cibles. Dans le cadre d'une application de suivi en environnement réel de l'utilisateur, l'identification des cibles est conditionnée par la visibilité des points caractéristiques constituant les marqueurs. Nous avons donc proposé une méthode de gestion d'occultations qui permet le suivi des points d'intérêt même si les cibles ne sont pas identifiées. Cette méthode repose sur un suivi de points en utilisant le détecteur de Harris et la mise en correspondance des points dans la séquence d'images est effectuée par corrélation. Dans un tel processus de suivi, des faux appariements de points peuvent surgir à tout instant, ceci nous a incité à employer l'algorithme robuste RANSAC pour déterminer les bonnes mises en correspondances des indices visuels. Toutefois, notre méthode de suivi robuste présente des inconvénients, comme le temps de calcul et le nombre de points visibles au cours de la séquence. Pour pallier le problème, nous avons alors introduit un autre capteur qui est la centrale inertielle afin de localiser le système de vision lorsque le nombre d'indices visuels ne permet pas d'estimer la pose de la caméra.

Enfin, nous avons présenté une approche hybride de tracking basée sur la collaboration d'un capteur inertielle et d'une caméra pour permettre le suivi de cible et la gestion d'occultations. Le dispositif hybride a été calibré pour transcrire l'ensemble des données dans le repère de référence qui est celui de la caméra. Ces données sont, ensuite, rassemblées dans un module de suppléance d'informations cinématiques et de points caractéristiques 2D de l'image. Des expériences dans un environnement réel ont été menées, les résultats obtenus ont démontrés que notre dispositif de tracking hybride apporte des améliorations significatives aux systèmes de suivi basé vision. Notre système donne à l'utilisateur plus de libertés de mouvement, en outre la condition de visibilité des cibles n'est pas requise pour un tel système. Nous avons présenté aussi les problèmes liés aux deux capteurs et nous avons montré que notre approche collaborative permet de pallier leurs inconvénients. La centrale inertielle présente des dérives dans le calcul des positions à partir de l'intégration des accélérations et la caméra souffre de la lenteur des traitements. Néanmoins, la rapidité du capteur inertielle et la précision de la caméra font un bon compromis pour réaliser un système de tracking effectif.

En résumé, le travail réalisé a présenté des originalités tout au long de cette thèse. En commençant par le module de localisation par la vision où un nouvel algorithme d'identification de cibles codées et une méthode hybride d'estimation de pose ont été développés. Ensuite, l'approche robuste proposée a permis de résoudre le problème d'occultations partielles des cibles en se basant sur une utilisation judicieuse d'un algorithme robuste. De plus, le système de mise en correspondance des points 3D du modèle d'objet et ses projections 2D dans l'image a constitué une contribution pour l'estimation de la pose de la caméra et le maintien du recalage virtuel. Enfin, le module de suivi hybride a apporté des améliorations notables pour le suivi basé vision. Les occultations totales et le flou de mouvement ont été gérés par ce module. En outre, la calibration du capteur hybride, l'estimation de la translation avec la centrale inertielle et le fonctionnement global du processus de suivi en présence d'occultations, ont présentés des aspects originaux de cette dernière partie de la thèse.

Perspectives

Cette thèse présente un système de tracking de cibles codées en réalité augmentée. Plusieurs problématiques ont été abordées et requièrent une attention particulière. Les évolutions de notre système les plus immédiates concernent essentiellement le système de localisation par la vision dont le fonctionnement principal repose sur l'utilisation de marqueurs codés. En effet, nous avons imposé des contraintes sur la forme et le motif des objets cibles pour les détecter dans l'image. Ces marqueurs doivent être positionnés dans divers endroits de l'environnement, ce qui restreint l'utilisation de notre système uniquement dans un environnement instrumenté. Le système d'extraction de primitives étant le premier verrou pour notre système de localisation. Une solution à ce problème sera l'utilisation des amers naturels ou des méthodes hybrides fonctionnant à l'aide d'un marqueur qui servira à initialiser la pose de l'objet.

Notre méthode robuste de gestion d'occultations partielles peut être optimisée pour un fonctionnement en temps réel. Le paramétrage de l'algorithme constitue l'un des verrous pour un fonctionnement correct de la méthode. En ce qui concerne les mouvements particuliers ou bien le changement de luminance et d'échelle, cela étant réalisé dans des limites du paramétrage de l'algorithme et il est nécessaire d'effectuer un réglage adaptatif si l'on veut réaliser le tracking dans des conditions variables de l'environnement.

Au niveau du système de tracking multi-capteurs, les améliorations concernent la détermination de la translation avec précision. Le calcul de positions à partir de la centrale inertielle, génère des erreurs considérables au cours du temps, ce qui met en échec le suivi après une période d'utilisation. La méthode que nous avons proposée pour résoudre le problème de la dérive, apporte une solution sous la contrainte du temps minimal de correction par la caméra. Ce temps représentant l'instant de mis à jour de la pose est très petit et empêche le suivi en cas de défaillance du système de vision pour une longue durée. Pour surmonter le problème, l'utilisation d'un autre capteur de position plus précis est nécessaire.

Notre système pourrait bien s'apprêter à une utilisation en vision directe (dans ce cas le monde réel est directement vu par l'opérateur par le biais de casques de vision semi-transparents). Divers problèmes spécifiques concernant la localisation et le suivi ont été résolus dans ce travail et une architecture de prédiction du point de vue, apportera des éléments de solution aux verrous technologiques des systèmes de réalité augmentée en vision directe.

Enfin, le système proposé est adapté à des environnements instrumentés où le volume de travail est restreint. Toutefois, cette thèse a permis de définir une base de critères nécessaires pour mettre en oeuvre un système multi-capteurs de suivi dans des applications de réalité augmentée en extérieur. En effet, dans le cadre du projet RAXENV (Réalité Augmentée en eXtérieur appliquée à l'ENVironnement, ANR Technologies logicielles 2006), mené en partenariat avec le Bureau des Ressources Géologiques et Minières et la Lyonnaise des eaux Suez, un système hybride composé d'une centrale inertielle (MTi), d'une caméra et d'un GPS sont utilisés pour la localisation dans un environnement de réalité augmentée en extérieur. L'objectif est de proposer à l'utilisateur des fonctionnalités à l'aide d'un dispositif portable où les aspects de l'environnement sont fournis. L'intérêt scientifique du projet réside dans le fait que le système doit pouvoir se localiser dans un environnement sur lequel il ne possède pas de connaissance a priori.

Bibliographie

- [1] F. Ababsa. *Recalage automatique d'objets 3D de forme libre en utilisant la clinométrie*. PhD thesis, Université d'Évry Val d'Essonne, Évry, France, December 2002.
- [2] F. Ababsa, M. Maldi, J. Y. Didier, and M. Mallem. Vision-based tracking for mobile augmented reality. In *Multimedia Services in Intelligent Environments*, Springer, 2007.
- [3] F. Ababsa and M. Mallem. Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. In *Proceedings of ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI'04)*, pages 431–435, Nanyang, Singapore, December 16-18, 2004.
- [4] K. Agusanto, L. Li, Z. Chuangui, and N. W. Sing. Photorealistic rendering for augmented reality using environment illumination. In *ISMAR'03*, pages 208–216, Tokyo, Japan, 2003.
- [5] K. Ahlers, D. Breen, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker, and D. Greer. An augmented vision system for industrial applications. *Telem manipulator and Telepresence Technologies, SPIE Proceedings*, 2351 :345–359, October 1994.
- [6] J. Alves, J. Lobo, and J. Dias. Camera-inertial sensor modelling and alignment for visual navigation. In *Proceedings of ICAR 2003, The 11th International Conference on Advanced Robotics*, pages 1693–1698, Coimbra, Portugal, June 2003.
- [7] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5) :578–589, 2003.
- [8] H. Araujo, R. Carceroni, and C. Brown. *A Fully Projective Formulation for Lowe's Tracking Algorithm*. Technical Report 641, University of Rochester, USA, 1996.
- [9] M. Aron, G. Simon, and M. O. Berger. Utilisation d'un capteur inertiel comme aide au suivi basé vision. In *15ème congrès francophone Reconnaissance des Formes et Intelligence Artificielle - RFIA'2006*, Tours, France, 2006.
- [10] Arvika : augmented reality for development, production and servicing. <http://www.arvika.de>.
- [11] H. Asada and J. Brady. The curvature primal sketch. In *IEEE Trans. Pattern Anal. Mach. Intelligence (PAMI)*, 8(1) :2–14, 1986.
- [12] T. Auer, M. Gervautz, and A. Pinz. Tracking in a multi-user augmented reality system. In *Proceedings of the First IASTED International Conference on Computer Graphics and Imaging*, pages 249–253, Halifax, Canada, June 1998.
- [13] R. Azuma. A survey of augmented reality. In *Presence : Teleoperators and Virtual Environments*, 6(4) :355–385, August 1997.
- [14] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, and B. Macintyre. Recent advances in augmented reality. In *IEEE Computer Graphics and Applications*, pages 34–47, 2001.
- [15] M. Bajura, H. Fuch, and R. Ohbuchi. Merging virtual objects with the real world : Seeing ultrasound imaginary within the patient. In *Proceedings of SIGGRAPH'92*, pages 203–210, Chicago, USA, July 1992.
- [16] M. Bajura and U. Neumann. Dynamic registration correction in video-based augmented reality systems. In *IEEE Computer Graphics and Applications*, 15(5) :52–60, September 1995.

- [17] M. O. Berger. Resolving occlusion in augmented reality : a contour based approach without 3d reconstruction. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CV-PR'97)*, pages 91–96, Puerto Rico, USA, 1997.
- [18] V. Berzins. Accuracy of laplacian edge detector. In *Computer Vision, Graphics and Image Processing*, 27(2) :195–210, April 1984.
- [19] P. J. Besl, J. B. Birch, and L. T. Watson. Robust window operators. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 591–600, Tampa, Florida, USA, 1988.
- [20] M. Billinghurst, H. Kato, and I. Poupyrev. The magic book : Moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3) :6–8, 2001.
- [21] A. Blake and A. Zisserman. Visual reconstruction. In *The MIT Press*, Massachusetts, USA, 1987.
- [22] D. Breen, R. Whitaker, E. Rose, and M. Tuceryan. Interactive occlusion and automatic object placement for augmented reality. In *Proc. of Eurographics*, pages 11–22, Poitiers, France, 1996.
- [23] W. Broll, M. Wittkamper, and E. Meier. Illuminating the mixed reality stage : Applying complex lighting conditions ar. In *International Symposium on Augmented Reality (ISAR'01)*, New York, NY, USA, October 2001.
- [24] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6) :679–698, 1986.
- [25] T. P. Caudell and D. W. Mizell. Augmented reality : An application of heads-up display technology to manual manufacturing processes. In *Proceedings of Hawaii International Conference on System Sciences*, pages 659–669, Hawaii, USA, January 1992.
- [26] L. Chai, W. A. Hoff, and T. Vincent. Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. In *Presence : Teleoperators and Virtual Environments*, volume 11, pages 474–492, Cambridge, MA, USA, 2002.
- [27] Y. Cho and U. Neumann. Multi-ring color fiducial systems for scalable fiducial tracking augmented reality. In *VRAIS'98 : Proceedings of the Virtual Reality Annual International Symposium*, page 212, Atlanta, GA, USA, 1998.
- [28] A. I. Comport, D. Kragic, E. Marchand, and F. Chaumette. Robust real-time visual tracking : Comparison, theoretical analysis and performance evaluation. In *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, pages 2852–2857, Barcelona, Spain, 2005.
- [29] A. I. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR'03)*, pages 36–45, Tokyo, Japan, October 2003.
- [30] A. I. Comport, E. Marchand, and F. Chaumette. Suivi robuste d'objets complexes en temps réel par asservissement visuel virtuel. In *Congrès francophone de Vision par Ordinateur, ORA-SIS'03*, pages 591–600, Gerardmer, France, May 2003.
- [31] A. I. Comport, E. Marchand, and F. Chaumette. Robust model based tracking for robot vision. In *IEEE/RSJ Int. Conf. on IROS'04*, volume 1, pages 692–697, Sendai, Japan, 2004.
- [32] P. Debevec. Rendering synthetic objects into real scenes : Bridging traditional and image based graphics with global illumination and high dynamic range photography. In *Proc. SIG-GRAPH'98*, pages 189–198, Orlando, Florida, USA, 1998.
- [33] D. DeMenthon and L. S. Davis. Exact and approximate solutions of the perspective-three-point problem. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(11) :1100–1105, 1992.
- [34] M. Dhome, M. Richetin, J. T. Lapreste, and G. Rives. Determination of the attitude of 3d objects from a single perspective view. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(12) :1265–1278, 1989.

- [35] J. Y. Didier. *Contributions à la dextérité d'un système de réalité augmentée mobile appliqué à la maintenance industrielle*. PhD thesis, Université d'Évry Val d'Essonne, Évry, France, December 2005.
- [36] L. Douadi. *Contribution à l'étude du recalage de données 3D/couleur*. PhD thesis, Université Montpellier II, France, 2006.
- [37] D. Drascic. Stereoscopic vision and augmented reality. *Scientific Computing and Automation*, 9(7), June 1993.
- [38] D. Drascic, J. J. Grodski, P. Milgram, K. Ruffo, P. Wong, and S. Zhai. Argos : A display system for augmenting reality. In *Proceedings of INTERCHI'93 : Human Factors in Computing Systems*, volume 88, pages 24–29, Amsterdam, Netherlands, April 1993.
- [39] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on PAMI*, 27(7) :932–946, 2002.
- [40] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1) :11–15, 1972.
- [41] N. I. Durlach and A. S. Mavor. *Virtual Reality : Scientific and Technological Challenges*. Report of the Committee on Virtual Reality Research and Development to the National Research Council, National Academy Press, University of North Carolina, USA, 1995.
- [42] E. K. Edwards, J. P. Rolland, and K. P. Keller. Video see-through design for merging of real and virtual environments. In *Proc. IEEE VR'93*, pages 223–233, Seattle, WA, USA, 1993.
- [43] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. In *IEEE Transactions on Robotics and Automation*, 8 :313–326, 1992.
- [44] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world : 2d windows for 3d augmented reality. In *UIST'93 : Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 145–155, Atlanta, Georgia, USA, 1993.
- [45] S. Feiner, B. Macintyre, and D. Seligmann. Knowledge-based augmented reality. *Commun. ACM*, 36(7) :53–62, 1993.
- [46] M. Fiala. Artag, a fiducial marker system using digital techniques. In *CVPR'05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 590–596, San Diego, CA, USA, 2005.
- [47] P. D. Fiore. Efficient linear solution of exterior orientation. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23 :140–148, 2001.
- [48] J. Fischer and G. Baratoff. Detecting dynamic occlusion in front of static backgrounds for ar scenes. In *EGVE'03 : Proceedings of the workshop on Virtual environments*, pages 153–161, Zurich, Switzerland, 2003.
- [49] M. A. Fischler and R. C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395, June 1981.
- [50] A. Fournier. *Illumination problems in computer augmented reality*. TR 95-35, Department of Computer Science at the University of British Columbia, Canada, 1994.
- [51] E. Foxlin, M. Harrington, and Y. Altsuler. Miniature 6-dof inertial system for tracking hmds. In *SPIE Helmet and Head Mounted Display III*, 3362, April 1998.
- [52] E. Foxlin and L. Naimark. Vis-tracker : A wearable vision-inertial self-tracker. In *VR'03 : Proceedings of the IEEE Virtual Reality 2003*, pages 199–206, Los Angeles, California, USA, March 2003.

- [53] A. Fuhrmann, G. Hesina, F. Faure, and M. Gervautz. Occlusion in collaborative augmented environments. In *Proc. 5th EUROGRAPHICS Workshop on Virtual Environments*, Vienna, Austria, 1999.
- [54] T. A. Funkhouser and C. H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Computer Graphics (SIGGRAPH'93 Proceedings)*, pages 247–254, Anaheim, California, USA, August 1993.
- [55] D. B. Gennery. Tracking known three-dimensional objects. In *Proc. of American Association of Artificial Intelligence (AAAI'82)*, pages 13–17, Pittsburgh, PA, USA, August 1982.
- [56] D. B. Gennery. Visual tracking of known three-dimensional objects. In *International Journal of Computer Vision*, 7(3) :243–270, 1992.
- [57] S. Gibson and A. Murta. Interactive rendering with real-world illumination. In *Eurographics Workshop on Rendering Techniques*, pages 365–376, London, UK, 2000.
- [58] B. L. Gottesfeld. A survey of image registration techniques. In *ACM, Computing Surveys*, 24(4) :326–376, 1992.
- [59] R. Grasset and J. D. Gascuel. Environnement de réalité augmentée collaboratif : Manipulation d'objets réels et virtuels. In *AFIG'01 (Actes des 14èmes journées de l'AFIG)*, pages 101–112, November 2001.
- [60] M. Haller, S. Drab, and W. Hartmann. A real-time shadow approach for an augmented reality application using shadow volumes. In *VRST'03 : Proceedings of the ACM symposium on Virtual reality software and technology*, pages 56–65, Osaka, Japan, 2003.
- [61] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. Robust statistics : the approach based on influence functions. In *Wiley series in probability and mathematical*, New York, NY, USA, 1986.
- [62] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6) :1426–1446, December 1989.
- [63] C. Harris. Tracking with rigid models. In *A. Blake, ed., Active Vision, chap. 4, MIT Press*, pages 59–73, 1993.
- [64] C. Harris and C. Stennett. Rapid – a video rate object tracker. In *Proc. 1st British Machine Vision Conf.*, pages 73–78, Oxford, UK, 1990.
- [65] C. Harris and M. Stephens. Combined corner and edge detector. In *Proceedings of the Alvey Conference*, pages 147–151, 1988.
- [66] W. Hartmann, J. Zauner, M. Haller, T. Luckeneder, and W. Woess. Shadow catcher : A vision based illumination condition sensor using artoolkit. In *IEEE International Augmented Reality Toolkit Workshop*, pages 44–45, Tokyo, Japan, October 2003.
- [67] S. Heiler. Robust estimates in linear regression - a simulation approach. *Computational Statistics*, pages 115–136, 1981.
- [68] E. C. Hildreth and N. M. Grzywacs. Incremental rigidity scheme for recovering structure from motion : Position-based versus velocity based formulations. In *J. Opt. Soc. Am. A*, 4(3) :503–516, March 1987.
- [69] W. Hoff, K. Nguyen, and T. Lyon. Computer vision-based registration techniques for augmented reality. In *Proc. of Intelligent Robots and Computer Vision XV, in Intelligent Systems Advanced Manufacturing, SPIE*, volume 2904, pages 538–548, Boston, Massachusetts, USA, 1996.
- [70] R. L. Holloway. *Registration Errors in Augmented Reality Systems*. PhD thesis, UNC Chapel Hill, Department of Computer Science, USA, 1995.

- [71] D. Homgren. *Design And Construction of a 30-Degree See-Through Mounted Display*. Technical Reports 92-030, University of North Carolina, USA, 1992.
- [72] B. Horn and B. G. Schunck. Determining optical flow. In *Artificial Intelligence*, 17 :185–203, 1981.
- [73] P. V. C. Hough. Machine analysis of bubble chamber pictures. In *Int. Conf. on High Energy Accelerators and Instrumentation*, pages 554–556, CERN, 1959.
- [74] P. J. Huber. Robust statistics. In *volume IX of Wiley*, New York, NY, USA, 1981.
- [75] Intersense. <http://www.isense.com>.
- [76] A. L. Janin, D. W. Mizell, and T. P. Caudell. Calibration of head-mounted displays for augmented reality applications. In *Proceedings of IEEE VRAIS'93*, pages 246–255, Seattle, USA, 1993.
- [77] J. M. Jolion, P. Meer, and S. Bataouche. Robust clustering with applications in computer vision. *PAMI*, 13(8) :791–802, 1991.
- [78] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya. A stereoscopic video see-through augmented reality system based on real-time vision-based registration. In *Proc. IEEE Virtual Reality 2000 (VR'00)*, pages 255–262, New Brunswick, NJ, USA, 2000.
- [79] M. Kanbara and N. Yokoya. Geometric and photometric registration for real-time augmented reality. In *ISMAR'02 : Proceedings of the International Symposium on Mixed and Augmented Reality*, pages 279–280, Darmstadt, Germany, 2002.
- [80] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, pages 321–331, 1987.
- [81] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR'2000)*, pages 111–119, Munich, Germany, October 2000.
- [82] E. Kerrien. *Outils d'imagerie multimodalité pour la neuroradiologie interventionnelle*. PhD thesis, Institut National Polytechnique de Lorraine, Nancy, France, 2000.
- [83] W. S. Kim. Advanced teleoperation, graphics aids and application to time delay environments. In *Proceedings of the 1st Industrial Virtual Reality Show and Conference (IVR'93)*, pages 202–207, Makuhari Meese, Japan, June 1993.
- [84] W. S. Kim. Virtual reality calibration and preview / predictive displays for telerobotics. In *Presence : Teleoperators and Virtual Environments*, 5(2) :173–190, 1996.
- [85] K. Kiyokawa, M. Billinghurst, B. Campbell, and E. Woods. An occlusion capable optical see-through head mount display for supporting co-located collaboration. In *Proc. 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, pages 133–142, Tokyo, Japan, 2003.
- [86] G. Klein. *Visual Tracking for Augmented Reality*. PhD thesis, University of Cambridge, UK, 2006.
- [87] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. Real-time vision-based camera tracking for augmented reality applications. In *VRST'97 : Proceedings of the ACM symposium on Virtual reality software and technology*, pages 87–94, Lausanne, Switzerland, 1997.
- [88] Z. D. Lan, R. Mohr, and P. Remagnino. Robust matching by partial correlation. In *Proceedings of the sixth British Machine Vision Conference*, pages 651–660, Birmingham, UK, September 1995.
- [89] K. M. Lee, P. Meer, and R. H. Park. *Robust adaptive segmentation of range images*. Technical report : Sogang University, Korea and Rutgers University, Korea, June 1996.

- [90] V. Lepetit and M. O. Berger. Handling occlusions in augmented reality systems : A semi-automatic method. In *Proc. IEEE and ACM International Symposium on Augmented Reality (ISAR'00)*, pages 146–197, Munich, Germany, 2000.
- [91] S. Z. Li. Robustizing robust m estimation using deterministic annealing. *Pattern Recognition*, 29(1) :159–166, 1996.
- [92] C. Loscos, G. Drettakis, and L. Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(3), September 2000.
- [93] D. G. Lowe. Three-dimensional object recognition from single two-dimensional image. In *Artificial Intelligence*, 31 :355–395, 1987.
- [94] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. In *Int. J. Comput. Vision*, 8(2) :113–122, 1992.
- [95] C. P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6) :610–622, 2000.
- [96] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, 1981.
- [97] P. Maes. Artificial life meets entertainment : Lifelike autonomous agents. *Commun. ACM*, 38(11) :108–114, 1995.
- [98] M. Maidi, F. Ababsa, and M. Mallem. Vision-inertial system calibration for tracking in augmented reality. In *2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO'05)*, pages 156–162, Barcelona, Spain, 2005.
- [99] M. Maidi, F. Ababsa, and M. Mallem. Active contours motion based on optical flow for tracking in augmented reality. In *8th International Conference on Virtual Reality (VRIC'06)*, pages 215–222, Laval, France, 2006.
- [100] M. Maidi, F. Ababsa, and M. Mallem. Robust augmented reality tracking based visual pose estimation. In *3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO'06)*, pages 346–35, Setúbal, Portugal, 2006.
- [101] M. Maidi, F. Ababsa, and M. Mallem. Robust fiducials tracking in augmented reality. In *The 13th International Conference on Systems, Signals and Image Processing (IWSSIP'2006)*, pages 423–42, Budapest, Hungary, 2006.
- [102] S. Malik, G. Roth, and C. McDonald. Robust 2d tracking for real-time augmented reality. In *Proceedings of Vision Interface (VI'2002)*, pages 399–406, Calgary, Alberta, Canada, 2002.
- [103] E. Malis and E. Marchand. Méthodes robustes d'estimation pour la vision robotique. In *Journées nationales de la recherche en robotique, JNRR'05*, Guidel, France, October 2005.
- [104] E. Marchand and F. Chaumette. Virtual visual servoing : a framework for realtime augmented reality. *Computer Graphics Forum*, 21(3) :289–298, September 2002.
- [105] H. Matsuka, A. Onosawa, and E. Osaya. Environment mapping for objects in the real world. In *International ARToolkit Workshop (ART)*, October 2002.
- [106] G. L. Matthaei, L. Young, and E. M. T. Jones. Impedance-matching networks and coupling structures. In *Microwave Filters*, 1980.
- [107] G. Medioni and Y. Yasumoto. Corner detection and curve representation using cubic b-splines. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 764–769, April 1986.
- [108] D. Merad. *Reconnaissance 2D/2D et 2D/3D d'objets à partir de leurs squelettes*. PhD thesis, Université d'Évry Val d'Essonne, Évry, France, December 2004.

- [109] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality : A class of displays on the reality-virtuality continuum. In *SPIE : Telem manipulator and Telepresence Technologies*, 2351 :282–292, 1994.
- [110] P. Milgram, S. Zhai, D. Drascic, and J. J. Grodski. Applications of augmented reality for human-robot. In *Proceedings of International Conference on Intelligent Robotics and Systems*, pages 1467–1472, Yokohama, Japan, July 1993.
- [111] J. V. Miller and C. V. Stewart. Muse : Robust surface fitting using unbiased scale estimates. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 300–306, San Francisco, California, USA, June 1996.
- [112] D. Mintz. Robust consensus based edge detection. *Computer Vision, Graphics and Image Processing : Image Understanding*, 59(2) :137–153, 1994.
- [113] D. Mintz, P. Meer, and A. Rosenfeld. Consensus by decomposition : A paradigm for fast high breakdown point robust estimation. In *DARPA '92*, pages 345–362, 1992.
- [114] D. Mogilev, K. Kiyokawa, M. Billinghamurst, and J. Pair. Ar pad : An interface for face-to-face ar collaboration. In *Computer Human Interaction (CHI'02) ACM*, pages 654–655, Minneapolis, Minnesota, USA, 2002.
- [115] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. 5th Int. Joint Conf. Artificial Intell.*, volume 2, page 584, Cambridge, Massachusetts, USA, August 1977.
- [116] J. J. More. The levenberg-marquardt algorithm, implementation and theory. *Numerical Analysis*, 1977.
- [117] J. D. Mulder. Realistic occlusion effects in mirror-based co-located augmented reality systems. In *Proc. IEEE Virtual Reality Conference (VR'2005)*, pages 203–208, Bonn, Germany, 2005.
- [118] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR'02)*, pages 27–36, Darmstadt, Germany, 2002.
- [119] N. S. Netanyahu and I. Weiss. Analytic outlier removal in line fitting. In *Proceedings of IAPR Workshop on Computer Vision*, volume 2, pages 406–408, October 1994.
- [120] T. Ohshima, K. Satou, H. Yamamoto, and H. Tamura. Rv-border guards : A multi-player entertainment in mixed reality space. In *Proceedings of the second International Workshop on Augmented Reality*, pages 1345–1352, San Fransisco, USA, 1999.
- [121] M. Olano, J. Cohen, M. Mine, and G. Bishop. Combatting rendering latency. In *SI3D'95 : Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 19–24, Monterey, California, USA, 1995.
- [122] S. Otmane. *Télétravail Robotisé et Réalité Augmentée : Application à la Téléopération via Internet*. PhD thesis, Université d'Évry Val d'Essonne, Évry, France, December 2000.
- [123] E. Oyama, N. Tsunemoto, S. Tachi, and Y. Inoue. Experimental study on remote manipulation using virtual reality. In *Presence : Teleoperators and Virtual Environments*, 2(2) :112–124, 1993.
- [124] C. Pinhanez. Augmenting reality with projected interactive displays. In *International Symposium on Virtual and Augmented Architecture*, Dublin, Ireland, 2001.
- [125] B. Piper, C. Ratti, and H. Ishii. Illuminating clay : A 3-d tangible interface for landscape analysis. In *CHI'02 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 355–362, Minneapolis, Minnesota, USA, 2002.
- [126] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proc. British Machine Vision Conference (BMVC'05)*, pages 519–528, Oxford, UK, 2005.

- [127] L. Quan and Z. D. Lan. Linear n-point camera pose determination. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8) :774–780, 1999.
- [128] R. Raskar, G. Welch, and W. C. Chen. Table-top spatially-augmented reality : Bringing physical models to life with projected imagery. In *International Workshop on Augmented Reality (IWAR'99)*, pages 64–71, San Francisco, CA, USA, 1999.
- [129] R. Raskar, G. Welch, K. L. Low, and D. Bandyopadhyay. Shader lamps : Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 89–102, London, UK, 2001.
- [130] M. Rauterberg, M. Fjeld, H. Krueger, M. Bichsel, U. Leonhardt, and M. Meier. Build-it : a video-based interaction technique of a planning tool for construction and design. *Work With Display Units (WWDU)*, pages 175–176, 1997.
- [131] J. Rekimoto. Pick-and-drop : a direct manipulation technique for multiple computer environments. In *Symposium on User Interface Software and Technology (UIST'97)*, pages 31–39, Banff, Alberta, Canada, 1997.
- [132] J. Rekimoto. Matrix : A realtime object identification and registration method for augmented reality. In *APCHI'98 : Proceedings of the Third Asian Pacific Computer and Human Interaction*, pages 63–68, Washington, DC, USA, 1998.
- [133] J. Rekimoto and Y. Ayatsuka. Cybercode : designing augmented reality environments with visual tags. In *DARE'00 : Proceedings of DARE 2000 on Designing augmented reality environments*, pages 1–10, Elsinore, Denmark, 2000.
- [134] J. Rekimoto and K. Nagao. The world through the computer : Computer augmented interaction with real world environments. In *Symposium on User Interface Software and Technology*, pages 29–36, Pittsburgh, Pennsylvania, USA, 1995.
- [135] J. Rolland, L. Davis, and Y. Baillet. A survey of tracking technologies for virtual environments. *W. Barfield T. Caudell, eds., Fundamentals of Wearable Computers and Augmented Reality chap. 3, Lawrence Erlbaum Assoc., 2000.*
- [136] P. J. Rousseeuw. Least median of squares regression. In *Journal of American Statistics Association*, 79 :871–880, 1984.
- [137] C. Sandor, A. MacWilliams, M. Wagner, M. Bauer, and G. Klinker. Sheep : The shared environment entertainment pasture. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, Darmstadt, Germany, 2002.
- [138] I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1) :1–12, 1999.
- [139] B. G. Schunck. The image flow constraint equation. *Graphics and Image Processing*, 35(1) :20–46, 1986.
- [140] B. G. Schunck. Image flow segmentation and estimation by constraint line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :1010–1027, 1989.
- [141] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. In *IEEE International Conference on Computer Vision*, volume 30, pages 222–229, Bombay, India, 1998.
- [142] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30 :1–24, 1999.
- [143] D. Sims. New realities in aircraft design and manufacture. *IEEE Computer Graphics and Applications*, 14(2), March 1994.

- [144] S. S. Sinha and B. G. Schunck. A robust method for surface reconstruction. In *Proceedings of the International Workshop on Robust Computer Vision*, pages 183–200, Washington, USA, 1990.
- [145] A. State, G. Hirota, D. T. Chen, W. F. Garrett, and M. A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *SIGGRAPH'96 Conference Proceedings*, pages 429–438, New Orleans, USA, 1996.
- [146] A. State, M. A. Livingston, G. Hirota, W. F. Garrett, M. C. Whitton, H. Fuchs, and E. D. Pisano. Techniques for augmented-reality systems : Realizing ultrasound-guided needle biopsies. In *Proceedings of SIGGRAPH'96*, pages 439–446, New Orleans, USA, August 1996.
- [147] J. Stauder. Augmented reality with automatic illumination control incorporating ellipsoidal models. *Transactions on Multi Media*, 1(12), June 1999.
- [148] C. V. Stewart. *A new robust operator for computer vision : Theoretical analysis*. Technical report, Department of Computer Science, Rensselaer Polytechnic Institute, NY, USA, August 1993.
- [149] D. Stricker, G. Klinker, and D. Reiners. A fast and robust line-based optical tracker for augmented reality applications. In *Proc. First International Workshop on Augmented Reality (IWAR'98)*, pages 129–145, San Francisco, USA, 1998.
- [150] N. Sugano, H. Kato, and K. Tachibana. The effects of shadow representation of virtual objects in augmented reality. In *ISMAR'03 : Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 76, Tokyo, Japan, 2003.
- [151] I. E. Sutherland. The ultimate display. In *IFIPS Congress*, volume 2, pages 506–508, New York, NY, USA, May 1965.
- [152] I. E. Sutherland. A head-mounted three-dimensional display. In *IFIPS Conference*, volume 33, pages 757–764, 1968.
- [153] G. Tharp, S. Hayati, and L. Phan. Telepresence system for telerobotic inspection. In *SPIE Proceedings : Telemanipulator and Telepresence*, volume 2351, pages 366–373, Boston, USA, 1994.
- [154] P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In *Proceedings of the seventh British Machine Vision Conference*, volume 15, pages 655–664, Edinburgh, Scotland, September 1996.
- [155] S. Ullman. Maximizing rigidity : The incremental recovery of 3-d structure from rigid and nonrigid motion. In *perception*, volume 13, pages 255–274, 1984.
- [156] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *Proc. 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 48–57, Arlington, USA, 2004.
- [157] J. Wang, R. Azuma, G. Bishop, V. Chi, J. Eyles, and H. Fuchs. Tracking a head-mounted display in a room-sized environment with head-mounted cameras. In *Proc. SPIE Helmet-Mounted Displays II*, volume 1290, pages 47–57, Orlando, USA, 1990.
- [158] B. Wanstall. Hud on the head for combat pilots. In *Interavia*, volume 44, pages 334–338, April 1989.
- [159] M. Ward, R. Azuma, R. Bennett, S. Gottschalk, and H. Fuchs. A demonstrated optical tracker with scalable work area for head-mounted display systems. In *Proc. 1992 Symposium on Interactive 3D graphics (SI3D'92)*, pages 43–52, Cambridge, Massachusetts, USA, 1992.
- [160] G. Welch and G. Bishop. Scaat : incremental tracking with incomplete information. In *Proc. 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH'97)*, Los Angeles, California, USA, 1997.

- [161] G. Welch and G. Bishop. *An introduction to the Kalman filter*. Technical Report N. TR 95-041, Department of Computer Science, University of North Carolina, USA, 2004.
- [162] M. Wloka. *Dissertation proposal : Time-critical graphics*. Tech. Rep. TR-CS-93-50, Computer Science Department, Brown University, USA, 1993.
- [163] M. Wloka. Lag in multiprocessor virtual reality. In *Presence : Teleoperators and Virtual Environments*, 4(1) :50–63, 1995.
- [164] M. Wloka and B. Anderson. Resolving occlusion in augmented reality. In *Symposium on Interactive 3D Graphics Proceedings*, pages 5–12, New York, NY, USA, April 1995.
- [165] Xsens. <http://www.xsens.com>.
- [166] Xsens. *Xsens Technologies B. V. : MTi and MTx User Manual and Technical Documentation*. Enschede, The Netherlands, 2007.
- [167] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *VR'01*, pages 71–78, Yokohama, Japan, March 2001.
- [168] S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *VR'99 : Proceedings of the IEEE Virtual Reality*, pages 260–267, Houston, USA, 1999.
- [169] X. Yu, T. D. Bui, and A. Krzyzak. Robust estimation for range image segmentation and reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5) :530–538, May 1994.
- [170] Z. Zhang. *A Flexible New Technique for Camera Calibration*. Technical Report MSR-TR-98-71, Microsoft Research, Microsoft Corporation, Redmond, WA, USA, 1998.
- [171] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1–2) :87–119, October 1995.
- [172] X. Zhuang, T. Wang, and P. Zhang. A highly robust estimator through partial likelihood function modeling and its application in computer vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 14 :19–35, 1992.

Annexe A

Calibration de la caméra par la méthode de Zhang

A.1 Introduction

La technique de calibration de caméra de Zhang [170] est une méthode très populaire au sein de la communauté de vision par ordinateur. La caméra observe un modèle d'objet planaire dans différentes orientations, ces orientations sont obtenues soit en déplaçant la caméra ou le modèle planaire. Les paramètres intrinsèques et extrinsèques de la caméra sont résolus en utilisant une solution analytique, suivie d'une optimisation non-linéaire basée sur le critère de maximum de vraisemblance. Les distorsions radiales et tangentielles de la lentille sont également modélisées. Les résultats obtenus par cette technique sont intéressants comparés aux techniques classiques qui emploient plusieurs plans orthogonaux.

A.2 Équations de base

On commence tout d'abord par définir les notations utilisées pour les développements mathématiques ultérieurs.

A.2.1 Notations

Un point 2D est désigné par $p = [u, v]^T$. Un point 3D est défini par $P = [X, Y, Z]^T$, on utilise \tilde{x} pour définir un vecteur homogène en ajoutant 1 dans sa dernière composante : $\tilde{p} = [u, v, 1]^T$ et $\tilde{P} = [X, Y, Z, 1]^T$. La caméra est modélisée par un modèle de sténopé : la relation entre un point 3D P et sa projection p est donnée par :

$$s\tilde{p} = M_{int} \begin{bmatrix} R & T \end{bmatrix} \tilde{P} \quad (\text{A.1})$$

où s est un facteur d'échelle arbitraire, (R, T) , appelé les paramètres extrinsèques, sont la rotation et la translation qui relie le repère monde au repère de la caméra. M_{int} est la matrice intrinsèque de la caméra donnée par :

$$M_{int} = \begin{pmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.2})$$

avec (u_0, v_0) les coordonnées du point principal, α_u et α_v les facteurs d'échelle selon les axes u et v de l'image, et γ est un paramètre qui décrit la non-orthogonalité des axes u et v de l'image. On utilise aussi l'abréviation : M_{int}^{-T} pour $(M_{int}^{-1})^T$ ou $(M_{int}^T)^{-1}$.

A.2.2 L'homographie entre le modèle planaire et son image

En supposant que le plan contenant les points employés pour la calibration a pour équation $Z = 0$ dans le repère monde, alors à partir de l'équation A.1, on obtient :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = M_{int} \begin{bmatrix} r_1 & r_2 & r_3 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = M_{int} \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (\text{A.3})$$

Par abus de notation, on utilise P pour désigner un point dans le modèle planaire, avec $P = [X, Y]^T$ puisque Z est toujours égal à 0, par conséquent, $\tilde{P} = [X, Y, 1]^T$. Donc, un point du modèle P et son image p sont reliés par une homographie H de dimension 3×3 définie à un facteur d'échelle près :

$$s\tilde{p} = H\tilde{P} \quad (\text{A.4})$$

avec :

$$H = \lambda M_{int} \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \quad (\text{A.5})$$

où r_1 , r_2 et r_3 désignent les colonnes de la matrice de rotation R_{CO} , rotation du repère objet par rapport au repère de la caméra.

A.2.3 Contraintes des paramètres intrinsèques

Étant donné une image du modèle planaire, l'homographie peut être décrite par :

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} \quad (\text{A.6})$$

À partir de l'équation A.5, on peut écrire :

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda M_{int} \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \quad (\text{A.7})$$

où λ est un facteur d'échelle arbitraire. Sachant que r_1 et r_2 sont orthonormaux, nous avons :

$$h_1^T M_{int}^{-T} M_{int}^{-1} h_2 = 0 \quad (\text{A.8})$$

$$h_1^T M_{int}^{-T} M_{int}^{-1} h_1 = h_2^T M_{int}^{-T} M_{int}^{-1} h_2 \quad (\text{A.9})$$

Il y a deux contraintes sur les paramètres intrinsèques car l'homographie a 8 degrés de libertés et il y a 6 paramètres extrinsèques (3 pour la rotation et 3 pour la translation), nous avons donc deux contraintes seulement sur les paramètres intrinsèques. Cependant, $M_{int}^{-T} M_{int}^{-1}$ décrit l'image d'un conique absolu. Dans la section suivante, nous en donnons l'interprétation géométrique.

A.2.4 Interprétation géométrique

Il est facile de vérifier que le modèle planaire est décrit dans le repère de la caméra par l'équation suivante :

$$\begin{bmatrix} r_3 \\ r_3^T t \end{bmatrix}^T \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = 0 \quad (\text{A.10})$$

où $w = 0$ pour les points à l'infini et $w = 1$ sinon. Ce plan intersecte le plan à l'infini par une ligne, on peut facilement voir que $[r_1 \ 0]^T$ et $[r_2 \ 0]^T$ sont des points particuliers de cette ligne. N'importe quel point est une combinaison linéaire de ces deux points, i.e. :

$$x_\infty = a \begin{bmatrix} r_1 \\ 0 \end{bmatrix} + b \begin{bmatrix} r_2 \\ 0 \end{bmatrix} = \begin{bmatrix} ar_1 + br_2 \\ 0 \end{bmatrix} \quad (\text{A.11})$$

Nous calculons maintenant l'intersection de cette ligne avec le conique absolu. Par définition, le point x_∞ , connu comme un point circulaire, satisfait : $x_\infty^T x_\infty = 0$, i.e. :

$$(ar_1 + br_2)^T (ar_1 + br_2) = 0 \quad (\text{A.12})$$

avec :

$$a^2 + b^2 = 0 \quad (\text{A.13})$$

La solution est donnée par $b = \pm ai$, où $i^2 = -1$. Les deux points d'intersection sont alors :

$$x_\infty = a \begin{bmatrix} r_1 \pm ir_2 \\ 0 \end{bmatrix} \quad (\text{A.14})$$

Leurs projections sur le plan image à un facteur d'échelle près, sont données par :

$$\tilde{p}_\infty = M_{int} (r_1 \pm ir_2) = h_1 \pm ih_2 \quad (\text{A.15})$$

Le point \tilde{p}_∞ est sur l'image du conique absolu, décrit par $M_{int}^{-T} M_{int}^{-1}$. Cela donne :

$$(h_1 \pm ih_2)^T M_{int}^{-T} M_{int}^{-1} (h_1 \pm ih_2) = 0 \quad (\text{A.16})$$

En exigeant que les parties réelles et imaginaires sont égales à zéro, on retrouve les équations A.8 et A.9.

A.3 Résolution des paramètres de la calibration

Cette section donne les détails sur la résolution des paramètres de la calibration. On commence par la solution analytique, suivie par une optimisation non-linéaire basée sur un critère de maximum de vraisemblance. Enfin, nous incluons les distorsions de la lentille pour tenir en compte les défauts de la caméra.

A.3.1 Solution analytique

Soit :

$$B = M_{int}^{-T} M_{int}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \quad (\text{A.17})$$

$$= \begin{bmatrix} \frac{1}{\alpha_u^2} & \frac{-\gamma}{\alpha_u^2 \alpha_v} & \frac{v_0 \gamma - u_0 \alpha_v}{\alpha_u^2 \alpha_v} \\ \frac{-\gamma}{\alpha_u^2 \alpha_v} & \frac{\gamma^2}{\alpha_u^2 \alpha_v^2} + \frac{1}{\alpha_v^2} & -\frac{\gamma(v_0 \gamma - u_0 \alpha_v)}{\alpha_u^2 \alpha_v^2} - \frac{v_0}{\alpha_v^2} \\ \frac{v_0 \gamma - u_0 \alpha_v}{\alpha_u^2 \alpha_v} & -\frac{\gamma(v_0 \gamma - u_0 \alpha_v)}{\alpha_u^2 \alpha_v^2} - \frac{v_0}{\alpha_v^2} & \frac{(v_0 \gamma - u_0 \alpha_v)^2}{\alpha_u^2 \alpha_v^2} + \frac{v_0^2}{\alpha_v^2} + 1 \end{bmatrix} \quad (\text{A.18})$$

B étant symétrique et définie par le vecteur suivant :

$$b = [B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33}]^T \quad (\text{A.19})$$

Soit la i^{ime} colonne du vecteur H , $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Nous avons donc :

$$h_i^T B h_j = v_{ij}^T b \quad (\text{A.20})$$

avec :

$$v_{ij} = [h_{i1} h_{j1} \quad h_{i1} h_{j2} + h_{i2} h_{j1} \quad h_{i2} h_{j2} \quad h_{i3} h_{j1} + h_{i1} h_{j3} \quad h_{i3} h_{j2} + h_{i2} h_{j3} \quad h_{i3} h_{j3}]^T \quad (\text{A.21})$$

Par conséquent, les deux contraintes fondamentales des équations A.8 et A.9, peuvent être écrites :

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

Si n images du modèle planaire sont observées, on a :

$$Vb = 0 \quad (\text{A.23})$$

où V est une matrice $2n \times 6$. Si $n \geq 3$, on a une solution unique de b définie à un facteur d'échelle près. Si $n = 2$, on peut imposer la contrainte $\gamma = 0$, i.e., $[0, 1, 0, 0, 0, 0]b = 0$, qui est ajouté comme une équation supplémentaire à l'équation A.23. Si $n = 1$, on peut seulement résoudre deux paramètres intrinsèques de la caméra, par exemple α_u et α_v en supposant que u_0 et v_0 sont connus et $\gamma = 0$. La solution pour l'équation A.23 est connue comme le vecteur propre de $V^T V$ associé à la plus petite valeur propre. Une fois b est estimée, on peut calculer tous les paramètres intrinsèques de la caméra M_{int} .

La matrice B est estimée à un facteur d'échelle près, $B = \lambda M_{int}^{-T} M_{int}$, λ est un facteur d'échelle arbitraire. Les paramètres intrinsèques à partir de la matrice B sont déterminés par :

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

M_{int} étant déterminée, les paramètres extrinsèques de chaque image sont calculés. À partir de l'équation A.5, on a :

$$\begin{aligned} r_1 &= \lambda M_{int}^{-1} h_1 \\ r_2 &= \lambda M_{int}^{-1} h_2 \\ r_3 &= r_1 \times r_2 \\ T &= \lambda M_{int}^{-1} h_3 \end{aligned} \quad (\text{A.24})$$

avec $\lambda = 1 / \|M_{int}^{-1} h_1\| = 1 / \|M_{int}^{-1} h_2\|$. Bien sûr à cause des bruits sur les mesures, la matrice de rotation ainsi calculée, ne satisfait pas les propriétés d'une matrice de rotation. Pour résoudre ce problème, on approxime la matrice R par une matrice Q de dimension 3×3 . Ceci revient à minimiser le critère suivant :

$$\min_R \|R - Q\|_F^2 \quad (\text{A.25})$$

avec $R^T R = I$ et puisque :

$$\|R - Q\|_F^2 = \text{trace} \left((R - Q)^T (R - Q) \right) = 3 + \text{trace} (Q^T Q) - 2\text{trace} (R^T Q) \quad (\text{A.26})$$

l'équation A.25 est équivalente à la maximisation de $\text{trace}(R^T Q)$. La décomposition en valeurs singulières de Q donne USV^T , où $S = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$. Si on définit une matrice orthogonale Z , telle que $Z = V^T R^T U$, alors :

$$\text{trace} (R^T Q) = \text{trace} (R^T USV^T) = \text{trace} (V^T R^T US) = \text{trace} (ZS) = \sum_{i=1}^3 z_{ii}\sigma_i \leq \sum_{i=1}^3 \sigma_i \quad (\text{A.27})$$

Le maximum est obtenu en mettant $R = UV^T$ car $Z = I$, ce qui représente la solution à l'équation A.25.

A.3.2 Estimation par maximum de vraisemblance

La solution présentée avant est obtenue à travers une minimisation d'une distance algébrique qui n'a pas de sens physique. Étant donné n images d'un modèle planaire et soit m le nombre de points appartenant au modèle, on suppose que les points image sont corrompus par une distribution de bruit identique et indépendant. Les estimations par maximum de vraisemblance sont obtenues par la minimisation de la fonction suivante :

$$\sum_{i=1}^n \sum_{j=1}^m \|p_{ij} - \hat{p}(M_{int}, R_i, T_i, P_j)\|^2 \quad (\text{A.28})$$

où $\hat{p}(M_{int}, R_i, T_i, P_j)$ est la projection du point P_j dans l'image i , selon l'équation A.5.

A.3.3 Les distortions radiales

Soit (u, v) les coordonnées idéales d'un pixel de l'image et (\tilde{u}, \tilde{v}) les coordonnées réelles observées. Les points idéaux sont la projection des points du modèle par la méthode de sténopé. De la même manière, (x, y) et (\tilde{x}, \tilde{y}) sont les coordonnées idéales et normalisées de l'image :

$$\begin{aligned} \tilde{x} &= x + x \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] \\ \tilde{y} &= y + y \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] \end{aligned} \quad (\text{A.29})$$

où k_1 et k_2 sont les coefficients de la distorsion radiale. Le centre de la distorsion radiale est le point principal. À partir de $\tilde{u} = u_0 + \alpha_u \tilde{x} + c \tilde{y}$ et $\tilde{v} = v_0 + \alpha_v \tilde{y}$, on a :

$$\begin{aligned} \tilde{u} &= u + (u - u_0) \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] \\ \tilde{v} &= v + (v - v_0) \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] \end{aligned} \quad (\text{A.30})$$

A.3.3.1 Estimation des distortions radiales par alternance

Les distortions radiales sont supposées être de faibles valeurs, alors on peut estimer les cinq autres paramètres intrinsèques avec la technique décrite dans la section A.3.2. On estime k_1 et k_2 après avoir estimé les autres paramètres, ce qui donne les coordonnées idéales d'un point (u, v) . À partir de l'équation A.30, on a deux équations pour chaque point dans chaque image :

$$\begin{bmatrix} (u - u_0)(x^2 + y^2) & (u - u_0)(x^2 + y^2)^2 \\ (v - v_0)(x^2 + y^2) & (v - v_0)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \tilde{u} - u \\ \tilde{v} - v \end{bmatrix} \quad (\text{A.31})$$

Soient m points dans n images, on peut rassembler les équations et avoir $2 \times m \times n$ équations, néanmoins on a : $Dk = d$, où $k = [k_1, k_2]^T$. La solution des moindres carrés linéaires est donnée par :

$$k = (D^T D)^{-1} D^T d \quad (\text{A.32})$$

Une fois k_1 k_2 sont estimés, on peut ajuster les autres paramètres en résolvant l'équation A.28 où $\hat{p}(M_{int}, R_i, T_i, P_j)$ est remplacée par l'équation A.30.

A.3.3.2 Estimation par maximum de vraisemblance globale

L'estimation complète des paramètres est réalisée par la minimisation de la fonction suivante :

$$\sum_{i=1}^n \sum_{j=1}^m \|p_{ij} - \tilde{p}(M_{int}, k_1, k_2, R_i, T_i, P_j)\|^2 \quad (\text{A.33})$$

où $\tilde{p}(M_{int}, k_1, k_2, R_i, T_i, P_j)$ est la projection du point M_j dans l'image i selon l'équation A.5 et A.30. C'est un problème de minimisation non-linéaire, qui est résolu par l'algorithme de Levenberg-Marquardt.

A.4 Résumé

En résumant, la procédure de calibration de Zhang est constituée des étapes suivantes :

- Soit modèle d'objet planaire,
- prendre plusieurs images du modèle planaire dans différentes orientations,
- détecter les points caractéristiques dans les images,
- estimer les cinq paramètres intrinsèques et tous les paramètres extrinsèques,
- estimer les coefficients de distortions radiales,
- ajuster tous les paramètres avec une estimation par maximum de vraisemblance.

Annexe B

Filtre de Kalman

En 1960, R.E. Kalman a publié un fameux article qui décrit une solution récursive pour résoudre le problème de filtrage des données discrètes. Depuis ce temps et grâce à la grande avancée dans le traitement numérique, le filtre de Kalman est devenu un outil indispensable de recherche [161].

Le filtre de Kalman est un ensemble d'équations mathématiques qui estiment les états d'un processus, pour minimiser l'erreur quadratique moyenne. Le filtre est très efficace dans plusieurs aspects : estimation des états passés, présents et futurs et peut modéliser même des systèmes dont le modèle est inconnu.

B.1 Le filtre de Kalman discret

B.1.1 Le processus à estimer

Le filtre de Kalman s'adresse au problème général d'estimation des états, $x \in R^n$, d'un processus en temps discret, régit par l'équation aux différences stochastiques suivantes :

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (\text{B.1})$$

avec la mesure $z \in R^n$ définie par :

$$z_k = Hx_k + v_k \quad (\text{B.2})$$

Les variables aléatoires w_k et v_k représentent respectivement les bruits de processus et de mesures. Ce sont des bruits blancs, indépendants, avec des distributions normales de probabilités :

$$p(w) \approx \mathcal{N}(0, Q) \quad (\text{B.3})$$

$$p(v) \approx \mathcal{N}(0, R) \quad (\text{B.4})$$

En pratique, la matrice de covariance du bruit de processus Q et la matrice de covariance du bruit de mesure R peuvent changer à chaque instant de mesure, cependant, on les considère ici comme étant constantes.

Les différentes matrices utilisées sont : la matrice A de dimension $n \times n$, qui relie l'état de l'instant $k - 1$ à l'instant k . La matrice B de dimension $n \times l$ relie la commande $u \in R^l$ à l'état x . La matrice H de dimension $m \times n$ relie l'état à la mesure z_k .

B.1.2 Les origines calculatoires du filtre

On définit $\hat{x}_k^- \in R^n$ l'état a priori à estimer à l'instant k et $\hat{x}_k \in R^n$ l'état a posteriori à estimer à l'instant k en présence de la mesure z_k . On peut définir les erreurs a priori et a posteriori avec :

$$\begin{aligned} e_k^- &\equiv x_k - \hat{x}_k^- \\ e_k &\equiv x_k - \hat{x}_k \end{aligned} \quad (\text{B.5})$$

La covariance de l'erreur a priori est alors :

$$P_k^- = E [e_k^- e_k^{-T}] \quad (\text{B.6})$$

et la covariance de l'erreur a posteriori est donnée par :

$$P_k = E [e_k e_k^T] \quad (\text{B.7})$$

On commence par retrouver l'équation qui calcule l'estimation de l'état a posteriori \hat{x}_k , cette estimation est une combinaison linéaire de l'estimation a priori \hat{x}_k^- et la différence pondérée entre une mesure actuelle z_k et une mesure de prédiction $H\hat{x}_k^-$:

$$\hat{x}_k = \hat{x}_k^- + K (z_k - H\hat{x}_k^-) \quad (\text{B.8})$$

La différence $z_k - H\hat{x}_k^-$ est appelée l'innovation ou le résidu. Elle traduit l'écart entre la mesure prédite $H\hat{x}_k^-$ et la mesure actuelle z_k . Un résidu égal à zero signifie que le filtre a convergé.

La matrice K est de dimension $n \times m$, appelée la matrice du gain du filtre de Kalman et définie par :

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (\text{B.9})$$

B.1.3 Algorithme du filtre de Kalman discret

Le filtre de Kalman estime les états d'un processus en utilisant une boucle de commande, le filtre estime ces états à chaque instants, et reçoit les mesures par un retour de boucle. Les équations du filtre de Kalman sont divisées en deux parties : mise à jour des états et mise à jour des mesures. La mise à jour des états permet la projection des états et la covariance de l'erreur. La mise à jour des mesures introduit de nouvelles mesures dans la boucle de correction pour avoir une meilleure estimation des états a posteriori.

Les équations de mise à jour des états sont aussi connues sous le nom de prédiction et les équations de mise à jour des mesures par les équations de corrections.

Les équations de prédictions sont données par :

$$\hat{x}_k^- = A\hat{x}_{k-1} + B u_{k-1} \quad (\text{B.10})$$

$$P_k^- = A P_{k-1} A^T + Q \quad (\text{B.11})$$

Les équations de corrections sont exprimées par :

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (\text{B.12})$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (\text{B.13})$$

$$P_k = (I - K_k H) P_k^- \quad (\text{B.14})$$

La première tâche dans la partie correction, est le calcul du gain de Kalman K_k (équation B.12). L'étape suivante est de déterminer l'estimation des états a posteriori en ajoutant les nouvelles mesures z_k (équation B.13). L'étape finale est d'estimer la nouvelle covariance de l'erreur (équation B.14).

Le filtre de Kalman est un estimateur récursif. Cela signifie que pour estimer l'état courant, seuls l'état précédent et les mesures actuelles sont nécessaires. L'historique des observations et des estimations n'est ainsi pas requis.

Le filtre de Kalman a deux phases distinctes : Prédiction et Mise à jour. La phase de prédiction utilise l'état estimé de l'instant précédent pour produire une estimation de l'état courant. Dans l'étape de mise à jour, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise (figure B.1).

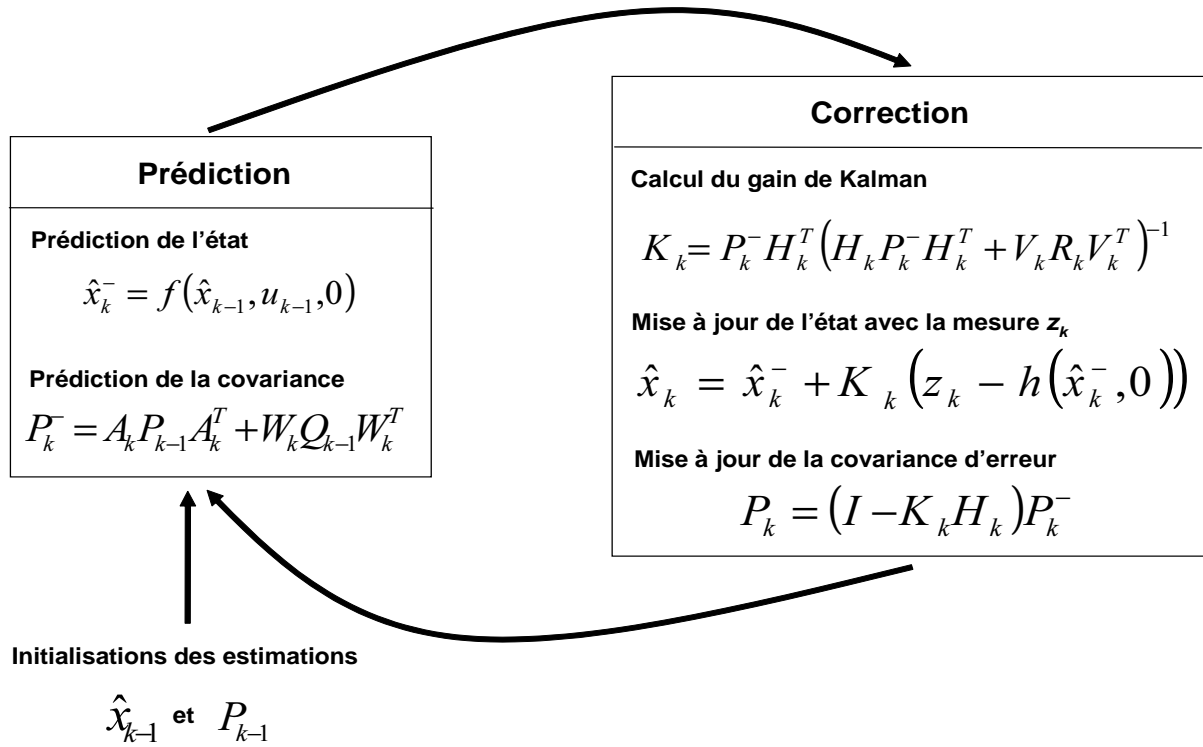


FIG. B.1 – Filtre de Kalman linéaire.

B.2 Filtre de Kalman étendu

Le filtre de Kalman est limité aux systèmes linéaires. Cependant, la plupart des systèmes physiques sont non linéaires. La non-linéarité peut être associée au modèle du processus, au modèle d'observation ou bien à tous les deux.

Dans le filtre de Kalman étendu, les modèles d'évolution et d'observation n'ont pas besoin d'être des fonctions linéaires de l'état mais peuvent à la place être des fonctions différentiables décrites par :

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (\text{B.15})$$

$$z_k = h(x_k, v_k) \quad (\text{B.16})$$

La fonction f peut être utilisée pour calculer l'état prédit à partir de l'état estimé précédent et, semblablement, la fonction h peut être employée pour calculer l'observation prédite de l'état prédit. Cependant, f et h ne peuvent pas être appliqués directement au calcul de la covariance : une matrice des dérivées partielles, qui est la jacobienne, est calculée. À chaque instant, la jacobienne est évaluée avec les états estimés courants. Ces matrices peuvent être employées dans les équations du filtre de Kalman. Ce processus linéarise essentiellement la fonction non-linéaire autour de l'estimation courante.

Pour estimer un processus non-linéaire, on commence par linéariser les équations (B.15 et B.16) qui régissent le système :

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \tilde{x}_{k-1}) + W w_{k-1} \quad (\text{B.17})$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + V v_k \quad (\text{B.18})$$

où :

- x_k et z_k sont les vecteurs d'états et de mesure.
- \tilde{x}_k et \tilde{z}_k sont les approximations des vecteurs d'états et de mesures.

- \hat{x}_k est une estimation a posteriori de l'état à l'instant k .
- Les variables aléatoires w_k et v_k représentent les bruits du processus et de mesure.
- A est la matrice jacobienne des dérivées partielles de f par rapport à x , donnée par :

$$A_{ij} = \frac{\partial f_i}{\partial x_j} (\hat{x}_{k-1}, u_{k-1}, 0) \quad (\text{B.19})$$

- W est la matrice jacobienne des dérivées partielles de f par rapport à w :

$$W_{ij} = \frac{\partial f_i}{\partial w_j} (\hat{x}_{k-1}, u_{k-1}, 0) \quad (\text{B.20})$$

- H est la matrice jacobienne des dérivées partielles de h par rapport à x :

$$H_{ij} = \frac{\partial h_i}{\partial x_j} (\hat{x}_{k-1}, 0) \quad (\text{B.21})$$

- V est la matrice jacobienne des dérivées partielles de h par rapport à v :

$$V_{ij} = \frac{\partial h_i}{\partial v_j} (\hat{x}_k, 0) \quad (\text{B.22})$$

On définit une nouvelle notation pour la prédiction de l'erreur :

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k \quad (\text{B.23})$$

et le résidu du mesure par :

$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k \quad (\text{B.24})$$

En utilisant les équations B.23 et B.24, on peut écrire les équations d'erreurs suivantes :

$$\tilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k \quad (\text{B.25})$$

$$\tilde{e}_{z_k} \approx H e_{x_k} - \eta_k \quad (\text{B.26})$$

où ε et η représentent des nouvelles variables aléatoires qui ont une moyenne nulle.

On remarque que les équations B.25 et B.26 sont linéaires et rassemblent la différence et l'équation de mesure B.1 et B.2 du filtre de Kalman discret. L'erreur \hat{e}_k est utilisée pour obtenir une estimation a posteriori du processus :

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \quad (\text{B.27})$$

Les variables aléatoires des équations B.25 et B.26 ont les distributions de probabilité suivantes :

$$\begin{aligned} p(\tilde{e}_{x_k}) &\approx \mathcal{N}(0, E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T]) \\ p(\varepsilon_k) &\approx \mathcal{N}(0, W Q_k W^T) \\ p(\eta_k) &\approx \mathcal{N}(0, V R_k V^T) \end{aligned} \quad (\text{B.28})$$

Étant donnés ces approximations et admettant que la valeur prédite de \hat{e}_k est égale à 0, l'équation du filtre de Kalman utilisée pour estimer \hat{e}_k est donnée par :

$$\hat{e}_k = K_k \tilde{e}_{z_k} \quad (\text{B.29})$$

En substituant l'équation B.29 dans B.27 et en utilisant l'équation B.24, on obtient :

$$\hat{x}_k = \tilde{x}_k + K_k \tilde{e}_{z_k} = \tilde{x}_k + K_k (z_k - \tilde{z}_k) \quad (\text{B.30})$$

L'équation B.30 peut être utilisée pour la mise à jour des mesures dans le filtre de Kalman étendu, avec les équations B.15 et B.16, le gain du filtre K_k calculé dans l'équation B.12 et la substitution de la covariance de l'erreur de mesure.

Les équations complètes du filtre de Kalman étendu sont données ci-dessous, on a remplacé \tilde{x}_k par \hat{x}_k^- pour désigner une valeur a priori, on a aussi mis l'indice k dans les jacobiniennes A , W , H et V pour dire que ces matrices sont différentes à chaque instant. Les équations de prédiction du filtre sont :

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{B.31}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{B.32}$$

Comme pour le filtre de Kalman discret, les équations de prédictions projettent les états et la covariance de l'erreur de l'instant $k - 1$ à l'instant k . A_k et W_k sont les matrices jacobiniennes du processus à l'instant k et Q_k est la covariance du bruit de processus à l'instant k . Les équations de mises à jour des mesures sont :

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{B.33}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \tag{B.34}$$

$$P_k = (I - K_k H_k) P_k^- \tag{B.35}$$

Les mises à jour des mesures corrigent l'estimation des états et la covariance avec z_k . H_k et V sont les matrices jacobiniennes de mesures à l'instant k et R_k est la covariance du bruit de mesure à l'instant k . Le filtre de Kalman étendu fonctionne de la même manière que le filtre de Kalman linéaire (figure B.2)

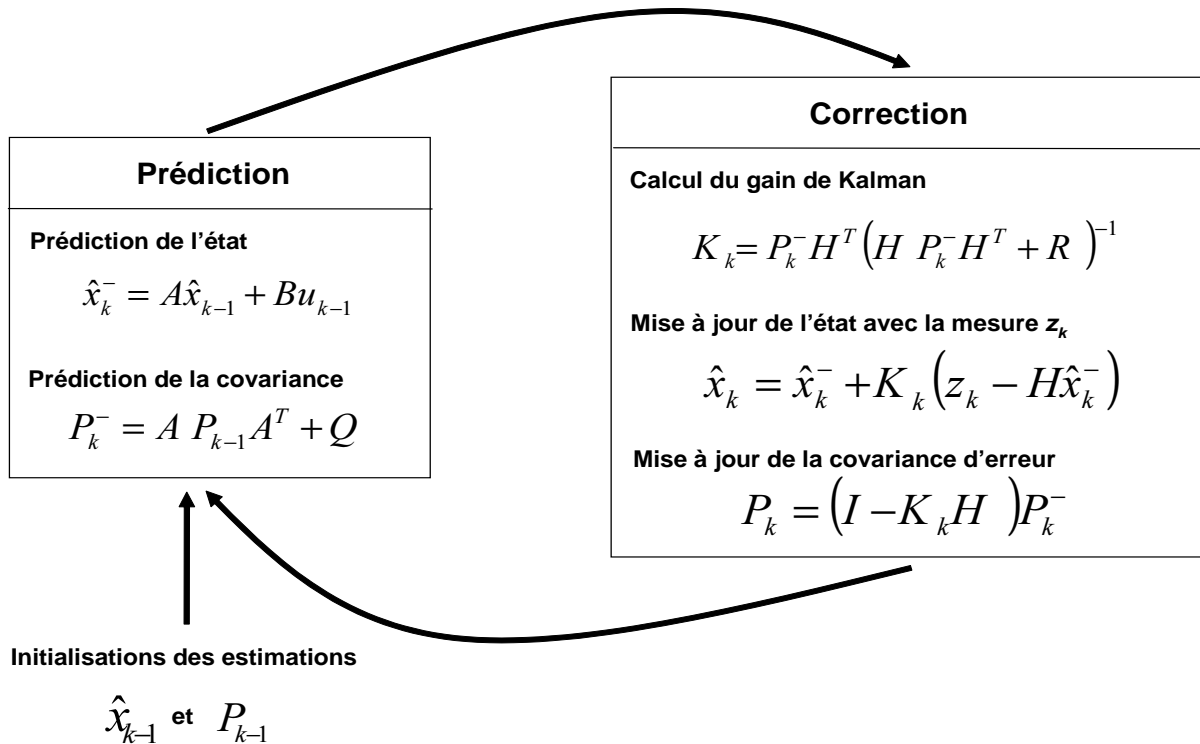


FIG. B.2 – Filtre de Kalman étendu.

B.3 Conclusion

Le filtre de Kalman est un filtre prédicteur-correcteur. Il prend en entrée un vecteur de mesure, puis il agit sur des paramètres internes du filtre pour effectuer une mise à jour des états et des mesures. Le principe de fonctionnement du filtre de Kalman est tout d'abord la prédiction des états et de la covariance de l'erreur du filtre à l'aide des estimations à l'étape précédente, ensuite une correction est apportée à cette prédiction en utilisant de nouvelles mesures.

Le filtre de Kalman est un observateur permettant l'estimation du vecteur d'état. Dans un environnement stochastique, lorsque les bruits de mesures et de processus sont blancs et gaussiens, le filtre de Kalman est optimal et l'erreur d'observation est la plus faible comparée à d'autres observateurs.

Annexe C

Le filtre de Harris

Le filtre de Harris est utilisé pour détection des points d'intérêts. Le détecteur de points de Harris est une version modifiée du détecteur de Moravec [115]. En effet, le détecteur de Moravec fonctionne dans un contexte limité. Harris et Stephens [65] ont identifié certaines limitations et les ont corrigées en se basant sur le calcul d'une matrice liée à la fonction d'auto-corrélation du signal calculée sur une fenêtre.

Harris considère le développement de Taylor de la fonction d'intensité I du pixel (x, y) au voisinage du pixel (u, v) :

$$I(x + u, y + v) = I(u, v) + x \frac{\partial I}{\partial x} + y \frac{\partial I}{\partial y} + o(x^2, y^2) \quad (\text{C.1})$$

Le détecteur de Harris réalise une auto-corrélation d'un point avec un voisinage en calculant la matrice d'auto-corrélation qui détermine la structure du voisinage :

$$E(x, y) = \sum_{u,v} w(u, v) \left[x \frac{\partial I}{\partial x} + y \frac{\partial I}{\partial y} + o(x^2, y^2) \right]^2 \quad (\text{C.2})$$

où $w(u, v)$ représente une fonction de la fenêtre de voisinage.

En négligeant le terme $o(x^2, y^2)$ (pour les petits déplacements), on obtient l'expression analytique suivante :

$$E(x, y) = Ax^2 + 2Cxy + By^2 \quad (\text{C.3})$$

avec :

$$\begin{aligned} A &= \frac{\partial^2 I}{\partial x^2} \otimes w \\ B &= \frac{\partial^2 I}{\partial y^2} \otimes w \\ C &= \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) \otimes w \end{aligned} \quad (\text{C.4})$$

où \otimes représente le signe de convolution.

Généralement, le filtre de voisinage, w , est un filtre binaire appliqué sur un voisinage rectangulaire. Harris et Stephens ont proposé d'utiliser un filtre gaussien afin de faire une modélisation réaliste du voisinage (figure C.1) :

$$w(u, v) = e^{-\frac{(u^2+v^2)}{2\sigma^2}} \quad (\text{C.5})$$

Pour prendre en compte le comportement général de la fonction, E , localement, on écrit :

$$E(x, y) = (x, y) \cdot M(x, y)^t \quad (\text{C.6})$$

où :

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (\text{C.7})$$

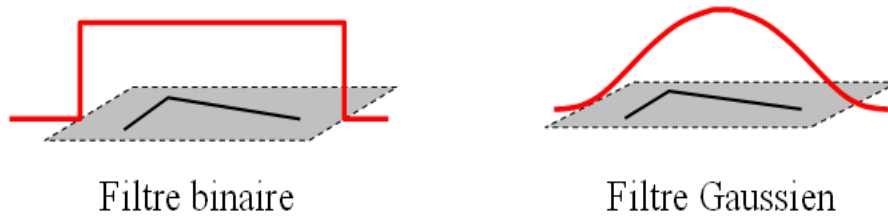
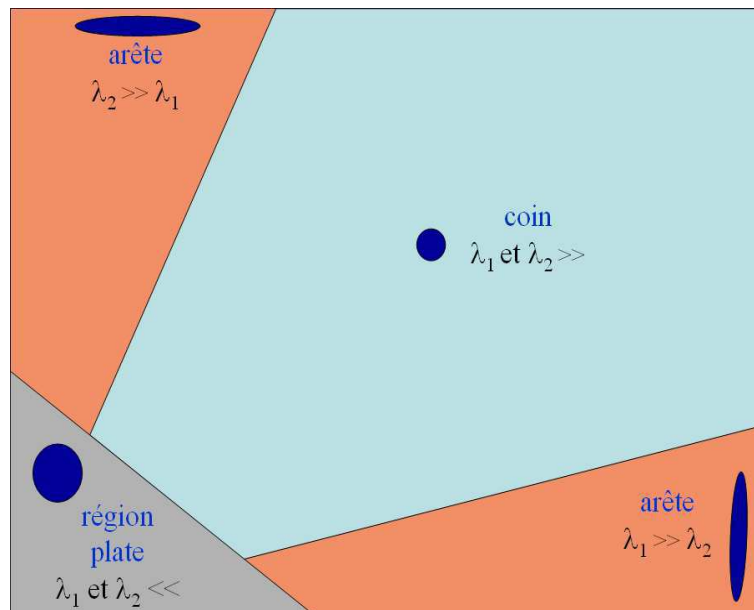


FIG. C.1 – Filtre de voisinage utilisé.

La matrice M caractérise le comportement local de la fonction E , les valeurs propres de cette matrice correspondent, en effet, aux courbures principales associées à E :

- Si les deux valeurs propres sont de faibles valeurs, alors la région considérée a une intensité approximativement constante.
- Si l'une des valeurs propres est de forte valeur alors que l'autre est de faible valeur alors la région contient un contour.
- Si les deux valeurs propres sont de fortes valeurs alors l'intensité varie fortement dans toutes les directions, ce qui caractérise un coin.

Le changement d'intensité dans la fenêtre est exprimée par les valeurs propres, λ_1 et λ_2 , de la matrice M et selon leurs valeurs : des coins, des contours ou des régions plates sont déterminés (figure C.2).

FIG. C.2 – Classification des courbures suivant les valeurs propres de M .

Annexe D

Représentation des rotations

D.1 Matrice de rotation

Une matrice de rotation est utilisée pour tourner un ensemble de points dans un système de coordonnées. Chaque point a de nouvelles coordonnées, mais sa distance relative reste inchangée. La matrice de rotation est définie par le produit des 3 matrices de rotations R_Z , R_Y et R_X respectivement autour de l'axe Z , Y et X . Toutes les rotations sont définies à l'aide des fonctions trigonométriques *sinus* et *cosinus* par l'expression suivante :

$$R = R_Z R_Y R_X \quad (\text{D.1})$$

$$R = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \quad (\text{D.2})$$

D.2 Angles d'Euler

Les angles d'Euler représentent un triplet d'angles de rotation (ϕ, θ, ψ) décrivant respectivement les rotations autour des axes X , Y et Z . On passe du référentiel fixe $Oxyz$ au référentiel lié au solide $Ox'y'z'$ par trois rotations successives comme illustré sur la figure D.1. Les 3 angles d'Euler sont définis par :

1. précession ψ , rotation autour de l'axe Oz , fait passer de $Oxyz$ au référentiel $Ouvw$.
2. nutation θ , rotation autour de l'axe Ou , fait passer de $Ouvw$ à $Ouwz'$.
3. rotation propre ϕ , rotation autour de l'axe Oz' , fait passer de $Ouwz'$ au référentiel lié au solide $Ox'y'z'$.

Le vecteur de rotation instantané du solide est donné par la simple somme :

$$\vec{\Omega} = \dot{\psi} \vec{z} + \dot{\theta} \vec{u} + \dot{\phi} \vec{z}' \quad (\text{D.3})$$

où les vecteurs apparaissant dans le membre de droite sont les vecteurs unitaires des axes correspondants.

Pour les angles d'Euler, il existe plusieurs conventions, la description de rotation de l'équation D.2 représente une rotation autour de $Z - X - Z$. Des conventions similaires sont obtenues avec d'autres configurations, ZYZ, XYX, XZX, YZY, YXY . Ces 6 combinaisons ont des expressions légèrement différentes et ont le même principe de fonctionnement.

Un deuxième type de convention très employé est connu sous le nom des angles de "roulis", "tangage", "lacet", parfois et par confusion, ces angles sont appelés aussi, les angles d'Euler. Cependant,

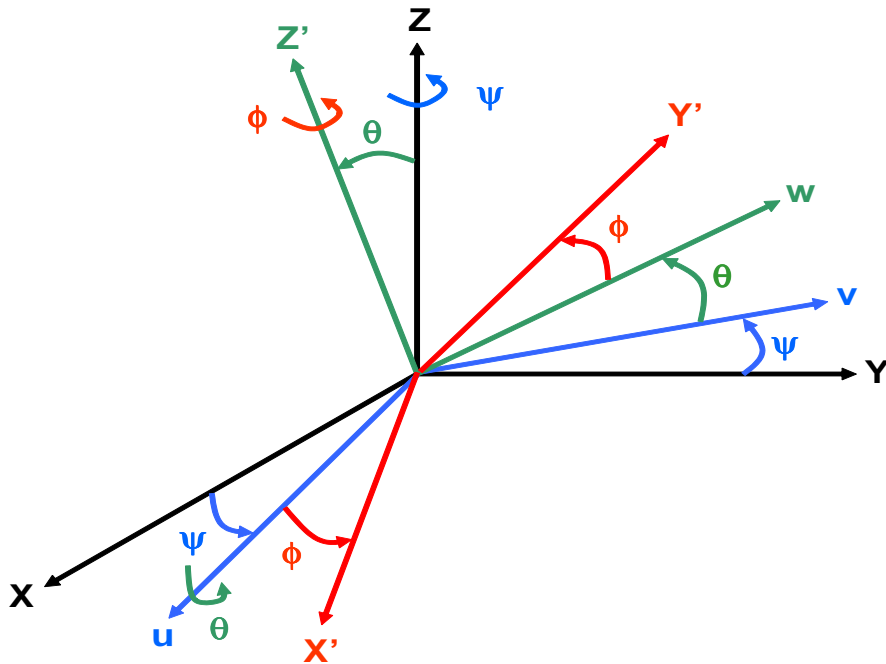


FIG. D.1 – Représentation des angles d'Euler.

le vrai nom de ces angles est les angles de Bryan, du Cadran ou bien des angles nautiques (figure D.2).

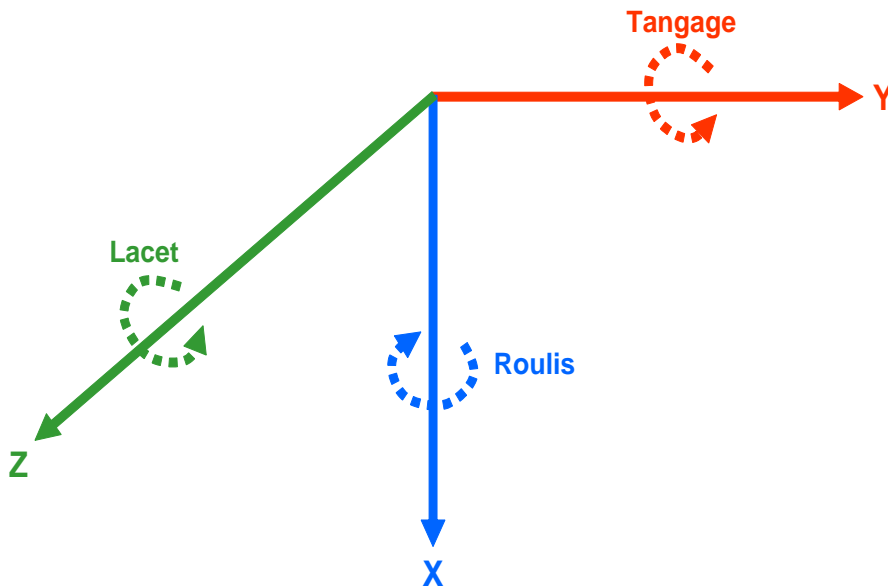


FIG. D.2 – Représentation des angles de Bryan.

Le problème avec les angles de rotation en général est la perte d'un degré de liberté. Comme la matrice de rotation finale dépend de l'ordre des multiplications, il est possible que parfois, l'une des rotations autour d'un axe soit confondue avec un autre axe de rotation. Il devient parfois impossible de tourner l'objet suivant un certain axe, ce qui provoque une singularité. La seule solution à ce problème est d'utiliser les matrices de rotations ou les quaternions.

D.3 Quaternions

Le principe de représentation des rotations utilisé avec les quaternions est opposé totalement avec les angles d'Euler. Pour stocker une rotation dans un quaternion, on utilise non plus une série de rotation successives autour de chaque axe mais un vecteur arbitraire et un angle quelconque de rotation autour de ce vecteur. La partie réelle du quaternion contient l'angle et la partie imaginaire représente le vecteur 3D. Le quaternion étant obligatoirement unitaire, c'est à dire de norme 1.

Le quaternion Q est représenté par : $Q = (X, Y, Z, W)$ (figure D.3). Ce quaternion peut se convertir en une matrice de rotation, R , de dimension 3×3 comme suit :

$$R = \begin{pmatrix} 1 - 2Y^2 - 2Z^2 & 2XY - 2ZW & 2XZ + 2YW \\ 2XY + 2ZW & 1 - 2X^2 - 2Z^2 & 2YZ - 2XW \\ 2XZ - 2YW & 2YZ + 2XW & 1 - 2X^2 - 2Y^2 \end{pmatrix} \quad (\text{D.4})$$

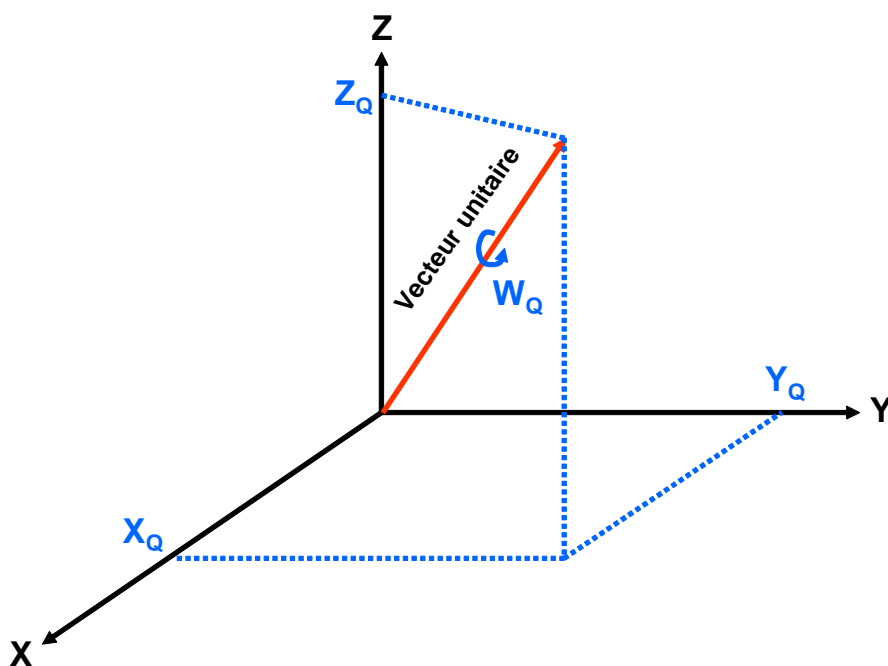


FIG. D.3 – Représentation de la rotation avec un quaternion.

Les quaternions offrent plusieurs avantages, le premier est la place mémoire relativement faible qu'ils occupent à comparer avec celle que nécessitent les matrices de rotations. Leur second avantage est la facilité qu'ils offrent pour la combinaison de rotations. La représentation angle-plus-axe des quaternions ne souffre pas du problème de singularité de rotation et l'utilisation des quaternions permet ainsi d'en venir à bout.

Suivi Hybride en présence d'Occultations pour la Réalité Augmentée

Résumé

Un des défis majeurs de la réalité augmentée est celui de l'alignement spatial des objets virtuels sur le monde réel. Plusieurs techniques de localisation par la vision ont été proposées dans la littérature pour améliorer le recalage virtuel. Le suivi hybride qui combine différentes technologies et méthodes semble ouvrir une nouvelle voie d'utilisation pour combler les divers problèmes liés à la précision et à la robustesse des algorithmes. Dans ce travail, nous avons développé une architecture multimodale de suivi de cibles et de gestion d'occultations en réalité augmentée. L'approche multimodale consiste à combiner différents capteurs et techniques, associés en fonction des conditions réelles de l'environnement. Cette architecture est constituée d'un module de tracking de cibles codées basé sur un algorithme hybride d'estimation de pose. Toutefois, cet algorithme de suivi de cibles est mis en échec si les marqueurs visuels ne sont pas entièrement visibles. Nous avons étendu donc, cette méthode de localisation spatio-temporelle par un module de suivi de points caractéristiques en présence d'occultations partielles des cibles. Cette étape se fonde sur l'étude des propriétés du suivi robuste des points d'intérêts et sur l'adaptation et la gestion des différentes conditions expérimentales. Enfin, une partie de suivi multi-capteurs est incluse dans notre démarche de conception logicielle. Ce dernier module de l'architecture est composé d'un dispositif hybride de tracking permettant de pallier les occultations totales des cibles. Des expérimentations avec le système global et de nombreuses évaluations ont servi à montrer l'intérêt de l'approche multimodale proposée pour le suivi et la gestion d'occultations en réalité augmentée. Le travail réalisé a permis de définir une base de critères nécessaires pour une mise en oeuvre efficace d'une architecture de suivi robuste et multi-capteurs dans un environnement de réalité augmentée.

Mots clés : réalité augmentée, vision par ordinateur, suivi temps réel, systèmes multi-capteurs.

Hybrid Tracking in presence of Occlusions for Augmented Reality

Abstract

One of the major challenges of augmented reality is the registration of virtual objects on real scene. Several localization techniques were proposed in literature to improve registration. The hybrid tracking which combines various technologies and methods, seems to open a new way to overcome different problems related to accuracy and robustness. In this work, we developed a multimodal tracking architecture to handle targets occlusions in augmented reality applications. The multimodal approach combines different sensors and techniques which are associated according to the real environment conditions. This architecture consists of a registration module of coded targets based on a hybrid algorithm of pose estimation. However, this tracking algorithm based coded fiducials fails if the visual markers are not completely visible. Therefore, we extended this space-time method of localization by a module of feature points tracking and occlusions management. This step is based on the study of the tracking properties and on the adaptation and the management of various experimental conditions. Finally, a multi-sensors tracking part is included into our software design approach. This module of architecture is composed of a hybrid device which overcomes total targets occlusion problem. Experiments with the overall system and many evaluations showed the interest of the proposed multimodal approach for tracking and occlusion management in augmented reality. This work allowed to define a base of necessary criteria for an effective implementation of a robust and multi-sensors tracking architecture in augmented reality environments.

Keywords : augmented reality, computer vision, real-time tracking, multi-sensors systems.