



HAL
open science

Vers un système interactif de réalité augmentée mobile pour la supervision de scénarios de maintenance industrielle

Alia Rukubayihunga

► **To cite this version:**

Alia Rukubayihunga. Vers un système interactif de réalité augmentée mobile pour la supervision de scénarios de maintenance industrielle. Interface homme-machine [cs.HC]. Université d'Evry Val-d'Essonne; Université Paris Saclay, 2016. Français. NNT : 2016SACLE051 . tel-01442986v1

HAL Id: tel-01442986

<https://hal.science/tel-01442986v1>

Submitted on 21 Jan 2017 (v1), last revised 1 Mar 2017 (v3)

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.

NNT : 2016SACLE051

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PARIS-SACLAY
PRÉPARÉE À L'UNIVERSITÉ D'EVRY-VAL-D'ESSONNE

Ecole doctorale n°580

ED STIC (Sciences et technologies de l'information et de la communication)

Spécialité de doctorat : Informatique

par

MME ALIA RUKUBAYIHUNGA

Vers un système interactif de réalité augmentée mobile pour la
supervision de scénarios de maintenance industrielle

Thèse présentée et soutenue à Evry, le 15 décembre 2016.

M. FRÉDÉRIC MERIENNE	Professeur ENSAM, INSTITUT IMAGE	(Président)
M. JEAN-MARC CIEUTAT	Chercheur associé (HDR) ESTIA, Bidart	(Rapporteur)
M. GUILLAUME MOREAU	Professeur Ecole Centrale de Nantes	(Rapporteur)
M. FRÉDÉRIC SOMMERLAT	Directeur du Développement WASSA SAS	(Examineur)
M. JEAN-YVES DIDIER	Maître de Conférence Laboratoire IBISC	(Co-encadrant)
M. SAMIR OTMANE	Professeur Laboratoire IBISC	(Directeur de thèse)

мои родители

Remerciements

Après ces trois années d'une expérience aussi passionnante que palpitante qu'est une thèse de doctorat, je tiens à remercier chaleureusement toutes les personnes qui ont été des instigateurs et des catalyseurs de cette aventure.

Dans un premier temps, je voudrais exprimer mes plus vifs remerciements à mon directeur de thèse Samir OTMANE pour sa disponibilité, sa réactivité ainsi que ses nombreux conseils tout le long de la thèse malgré ses nombreuses charges.

J'adresse également mes plus sincères remerciements mon encadrant Jean-Yves DIDIER avec qui j'ai beaucoup appris et qui s'est montré à l'écoute, disponible et clairvoyant malgré toutes ses nombreuses missions pédagogiques et de recherche.

Je tiens à remercier Frédéric SOMMERLAT, le directeur de développement de la société WASSA pour sa confiance et sa compréhension des enjeux de la thèse.

Je remercie Mehdi CHOUITEN qui a initié cette thèse au sein de la société WASSA et qui m'a partagé son expérience en tant qu'ancien doctorant.

Je remercie également tous les membres du jury, notamment Jean-Marc CIEUTAT et Guillaume MOREAU qui ont accepté de rapporter ces travaux de recherche et d'y apporter des remarques constructives, ainsi que Frédéric MERIENNE qui a fait l'honneur d'être président de mon jury.

Je souhaite également remercier l'ANRT à travers ses subventions de recherches qui m'a permis de me consacrer à ma thèse ainsi que pour l'organisation des formations pour les futurs docteurs.

Je remercie avec grand plaisir mes stagiaires Sylvain DEGEORGES, Kevin GOILARD, Mathieu MOUTTAPA qui ont été d'une grande aide dans les développements de la solution mobile.

Je voudrais remercier en outre mes collègues du laboratoire IBISC, notamment l'équipe IRA2 et particulièrement Samia BOUCHAFA, Frédéric DAVESNE, Amine CHELLALI, Guillaume BOUYER, David ROUSSEL, Etienne COLLE qui m'ont accueillie chaleureusement et qui se sont montrés disponibles pour quelques éclaircissements. Ma reconnaissance va également vers Sabine SEGALA qui a facilité les procédures administratives. Je n'oublie pas non plus mes co-doctorants du laboratoire IBISC, notamment Junior, Khouloud, Faouzi, Aylen, Ludovic, Hakim, Housseem, Ryma, M'Hamed, Omar,... qui ont apporté une bouffée d'oxygène dans cet environnement.

Je remercie également mes collègues de la société WASSA, notamment les membres et ex-membres de l'équipe R&D et particulièrement Flora, Adrien, Myriam, Erwann, Aurélie, Samantha pour ses discussions très passionnées, Peter pour son côté un peu déjantée qui apporte un grand bol de bonne humeur, Dorothée, Antony, Mithra, Elsa, Jessica,... Un merci spécial à Dorothée et Alexandre qui ont bien voulu faire don de leur image pour la science.

J'en profite pour remercier mes amis que j'ai quelque peu délaissée dernièrement, qui m'ont encouragée et apportée leur support moral et matériel tout le long de ces trois années, notamment M&M (qui se reconnaîtront, je l'espère), Anges, Pierre, Arlaine, Florent,...

Enfin, le meilleur pour la fin, ma gratitude et ma reconnaissance vont à des êtres qui me sont très chers que sont ma famille, notamment mes parents, mes frères et ma soeur pour leur soutien indéfectible, leur support, leur attention et leurs encouragements qui m'ont accompagnés tout le long de cette aventure.

Table des matières

Introduction générale	15
I Etat de l’art	21
1 Tour d’horizon de la réalité augmentée	23
1.1 Réalité Augmentée : définition et balayage du domaine	23
1.1.1 Taxonomie fonctionnelle de la réalité augmentée	24
a) Taxonomie fonctionnelle de [Fuchs et al., 2003]	24
i) Fonctionnalité « Superposition réel/virtuel »	24
ii) Fonctionnalité « Réalité documentée » ou « Virtualité do- cumentée »	25
iii) Fonctionnalité « Réalité à compréhension ou visibilité augmentée »	25
iv) Fonctionnalité « Association visuelle du réel et du virtuel »	26
v) Fonctionnalité « Association comportementale du réel et du virtuel »	26
vi) Fonctionnalité « Substitution du réel par le virtuel » ou « Réalité virtualisée »	26
b) Taxonomie fonctionnelle d’interaction / d’interactivité	27
i) Interaction entre l’homme et le monde réel	27
ii) Interaction humaine sur le monde virtuel	27
iii) Interaction humaine sur le monde virtuel et sur le monde réel	28
iv) Interactivité entre le monde réel et le monde virtuel	29
1.1.2 Technologies de visualisation pour la réalité augmentée	29
a) Les visiocasques	29
i) Visiocasques à vision directe	29
ii) Visiocasques à vision indirecte	31
b) Projecteur rétinien	33
c) Technologies de visualisation fixe	33

d)	Dispositifs de projection portatifs	35
e)	Dispositifs de visualisation de poche	35
1.2	Suivi et estimation de la pose	38
1.2.1	Suivi basé capteurs de position ou d'orientation	39
1.2.2	Suivi basé vision	40
a)	Suivi à l'aide de caméras à infrarouge	41
b)	Suivi à l'aide de capteurs à lumière visible	42
i)	Suivi à l'aide de cibles codées	42
ii)	Suivi sans cibles codées	45
c)	Suivi à l'aide de capteurs de profondeur	47
d)	Algorithmes PnP	48
1.2.3	Méthodes hybrides d'estimation de la pose ou fusion de capteurs	50
1.3	Bilan	51
2	Réalité augmentée mobile et industrielle	53
2.1	Analyse de caractéristiques de quelques systèmes de RA mobile	53
2.2	RAM : applications et limitations	57
2.2.1	Applications de réalité augmentée mobile	57
2.2.2	Limitations	62
2.3	Réalité augmentée dans l'industrie	64
2.3.1	RA mobile dans le processus industriel	65
a)	Conception et développement	65
b)	Fabrication et production	67
c)	Mise en service et maintenance	68
2.3.2	Limitations	70
2.4	Scénario de réalité augmentée pour la maintenance	72
2.4.1	Conception du scénario de maintenance	72
2.4.2	Suivi du scénario de maintenance	74
2.5	Bilan	76
II	Contributions :	
	Détection des tâches d'assemblage et de désassemblage	79
3	L'assemblage en tant que structure rigide	83
3.1	L'erreur de reprojexion, une métrique pour la détection d'actions d'assemblage ou de désassemblage	83
3.1.1	Pipeline basé vision par ordinateur	84
a)	Pièces d'assemblage sous forme d'objets indépendants	86
b)	Pièces d'assemblage sous forme d'un objet unique	87
3.1.2	Expérimentation et résultats	88

a)	Protocole expérimental	88
b)	Résultats	89
3.1.3	Discussion	93
3.2	Détermination de l'instant d'assemblage ou de désassemblage	96
3.2.1	Algorithmes de détection de ruptures	96
a)	Algorithmes hors ligne	97
i)	Carte de contrôle de Shewhart	98
ii)	Carte de contrôle avec moyenne géométrique glissante	99
iii)	Carte de contrôle avec moyenne "finie" glissante	99
iv)	Algorithme des dérivées filtrées	100
b)	Adaptation des algorithmes hors ligne pour une détection en ligne	100
i)	Carte de contrôle de Shewhart	101
ii)	Carte de contrôle avec moyenne géométrique glissante	101
iii)	Carte de contrôle avec moyenne "finie" glissante	102
c)	Algorithmes en ligne avec connaissance <i>a priori</i> des données avant et après rupture	102
i)	Algorithme du CUSUM (<i>Cumulative Sum</i>)	103
ii)	Algorithme à taille d'échantillon fixe (TEF)	104
d)	Algorithmes en ligne avec connaissance <i>a priori</i> uniquement avant la rupture	104
i)	Algorithme du CUSUM + GLR (<i>Generalized Likelihood Ratio</i>)	104
ii)	Algorithme du CUSUM + MLR (<i>Mixture Likelihood Ratio</i>)	105
3.2.2	KSUM et KSUMratio	106
3.2.3	Expérimentation	108
3.2.4	Discussion	110
3.3	Bilan	112
4	L'assemblage en tant que structure dynamique	113
4.1	Détermination de la matrice de transformation	114
4.1.1	Gestion du <i>gimbal lock</i> et des instabilités numériques	116
4.1.2	Mouvement par rapport à un objet de référence statique	119
4.1.3	Mouvement par rapport à la position initiale de l'objet en mouvement	121
4.1.4	Discussion	122
4.2	Défauts liés à la mesure	127
4.2.1	Paramètres influençant l'estimation du mouvement	127
4.2.2	Utilisation de la régression linéaire	133
4.2.3	Utilisation du SVM	135
4.2.4	Correction des mesures de mouvement en temps réel	136

4.2.5	Expérimentations	137
4.2.6	Discussion	139
4.3	Défauts liés à l'exécution de l'assemblage	141
4.3.1	Méthode pour la reconnaissance d'actions	142
4.3.2	Expérimentations	144
4.3.3	Discussion	145
4.4	Bilan	146
5	Intégration au framework ARCS (<i>Augmented Reality Component System</i>)	148
5.1	Établissement de la base de modèles et des scénarios	149
5.2	Mise en place du système de reconnaissance d'actions	149
5.3	Mise en place d'un système de reconnaissance des pièces peu texturées	154
5.4	Bilan	158
	Conclusion et perspectives	160
	Annexe A Résultats : Analyse de l'évolution de l'erreur de reprojection lors de tâches d'assemblage	184
A.1	Vidéos synthétiques	184
A.2	Vidéos dans des conditions réelles	184
A.3	Scénarios non concluants	187
	Annexe B Résultats : Caractérisation du mouvement à l'aide de la matrice de transformation	190
B.1	Mouvement par rapport à un objet de référence statique	190
B.2	Mouvement par rapport à la position initiale de l'objet en mouvement	191
	Annexe C Travaux en entreprise	194
C.1	SDK "What's That Face"	194
C.1.1	Etat de l'art	194
C.1.2	Algorithmes choisis et implémentés	195
C.1.3	Conclusion et perspectives	196
C.2	PoC GDF Suez	196
C.2.1	Etat de l'art	198
a)	Suivi à l'aide de marqueurs	198
b)	Suivi sans marqueurs à l'aide de capteurs	199
c)	Visualisation de divers sites industriels	200
i)	Smart Vidente	200
ii)	SiteLens	201
C.2.2	Algorithme choisi et implémenté	202
C.2.3	Conclusion et perspectives	202

Table des figures

1.1	Continuum de Milgram	24
1.2	Réalité documentée. Projet AMRA (Assistance à la Maintenance en RA). [Didier, 2005]	25
1.3	Réalité à compréhension ou visibilité augmentée. Projet AMRA. [Didier, 2005]	25
1.4	Association visuelle du réel et du virtuel. Projet AMRA. [Didier, 2005]	26
1.5	Interaction avec l'utilisateur	27
1.6	test	30
1.7	Modèles de visiocasques à vision indirecte. ¹	31
1.8	Principes des visiocasques. [Azuma, 1997]	32
1.9	Projecteur rétinien	33
1.10	Projecteurs rétiniens Avegant Glyph. ²	33
1.11	Pico projecteurs ³	34
1.12	Dispositif de projecteur portable.	35
1.13	Dispositif de visualisation de poche. ⁴	35
1.14	Schématisation du processus d'un système de réalité augmentée. [Didier, 2014]	38
1.15	Pipeline général pour l'estimation de la pose basée vision.	41
1.16	Exemples de cibles codées utilisées par ARToolKit.[Didier et al., 2008]	42
1.17	Exemples de cibles codées utilisées par ARTag.[Didier et al., 2008]	43
1.18	Exemples de cibles codées utilisées par ArUco.[Didier et al., 2008]	43
1.19	Etapes d'identification des cibles codées à l'aide de Cybercode.[Didier et al., 2008]	44
1.20	Exemples de cibles codées utilisées par InterSense.[Didier et al., 2008]	45
1.21	Kinects de Microsoft. ⁵	48
2.1	Tourisme et patrimoine culturel. [Liestol and Morrison, 2013]	58
2.2	Navigation dans un campus d'université à l'aide de CAViAR. [Delail et al., 2012] 58	
2.3	Aide à la navigation avec WikitudeDrive . [Carmigniani and Furht, 2011] . . .	59
2.4	Réalité augmentée dans l'industrie du jeu vidéo.	60
2.5	Visualisation de modèle 3D à partir de formes planaires. [Hagbi et al., 2011]	61
2.6	Surveillance de l'environnement. [Veas et al., 2012]	61
2.7	Smart Vidente pour l'ingénierie civile. [Schall et al., 2013]	62

2.8	Cycle de vie d'un produit.	64
2.9	Réalité augmentée pour le design de produit. [Park, 2008]	66
2.10	ClonAR pour le remodelage d'objet réel. [Csongei et al., 2012]	66
2.11	SpaceDesign pour la modélisation en temps réel. [Fiorentino et al., 2002]	67
2.12	Pistolet de soudure intelligent de ARVIKA . [Friedrich et al., 2002]	67
2.13	Système KARMA pour l'assistance à la maintenance. [Feiner et al., 1993]	68
2.14	Assistance pour l'assemblage d'une pièce mécanique. [Henderson and Feiner, 2009] 69	
2.15	ARVIKA dans le cadre d'un test de collision d'une automobile. [Friedrich et al., 2002] 69	
2.16	Représentation du « stress » (contraintes) d'un pont. [Re, 2013]	70
2.17	Exemples de structure hiérarchique pour l'assemblage de pièces.	73
3.1	Pipeline général pour l'estimation de la pose et le calcul de l'erreur de reprojection suivant les approches basées modèles.	84
3.2	Exemple de configuration des modèles 3D enregistrés.	85
3.3	Vue locale de l'évolution de l'erreur de reprojection pendant la phase initiale d'assemblage (cas du suivi multiple sur la vidéo synthétique représentant une rotation autour de l'axe Z de l'objet 1). (cf graphe entier 3.4e)	89
3.4	Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une rotation autour de l'axe Z de l'objet 0.	90
3.5	Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une translation selon l'axe X de l'objet 0.	91
3.6	Désassemblage suivant une rotation autour de l'axe Z de l'objet 0 dans des conditions réelles.	92
3.7	Désassemblage suivant une translation selon l'axe X de l'objet 0 dans des conditions réelles.	93
3.8	Superposition des courbes d'évolution de l'erreur de reprojection déterminée avec des vidéos synthétiques et réelles.	94
3.9	Evolution de l'erreur de reprojection suivant la résolution de l'image (cas de la rotation autour de l'axe Z de l'objet 0 dans des conditions réelles avec le suivi multiple).	94
3.10	Application du KSUM et KSUMration sur les valeurs de l'erreur de reprojection enregistrées suivant différents mouvements des pièces.	108
4.1	Schéma illustrant la détermination de la matrice de transformation en général.	115
4.2	Graphe montrant les zones de perturbations avec la fonction cos.	118
4.3	Schéma représentant les relations entre chaque pièce d'un objet à assembler ou à désassembler au cours du temps.	119

4.4	Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une translation selon l'axe X de l'objet 0 en considérant le mouvement relatif entre l'objet en mouvement et l'objet de référence.	123
4.5	Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 en considérant le mouvement relatif entre l'objet mobile et l'objet de référence. (pièces posées sur un plan)	124
4.6	Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une translation selon l'axe X de l'objet 0 en considérant le mouvement relatif de l'objet mobile à l'instant t par rapport à sa position initiale.	125
4.7	Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 en considérant le mouvement relatif de l'objet mobile à l'instant t par rapport à sa position initiale. (pièces posées sur un plan)	126
4.8	Assemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 dans des conditions réelles. (pièces non posées sur une surface plane)	138
4.9	Evolution des paramètres de mouvement obtenues dans le cadre d'un de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 dans des conditions réelles. (pièces non posées sur une surface plane).	138
4.10	Correction des valeurs de mouvements (angle et trajectoire).	139
4.11	Correction des courbes de mouvement (vecteur de rotation et de translation).	140
4.12	Nomenclature des liaisons mécaniques entre deux solides. ⁶	141
4.13	Changement de repères pour la reconnaissance des liaisons entre les pièces (cas d'une liaison pivot d'axe une des arêtes du cube).	142
4.14	Identification d'une liaison pivot d'axe Y de l'objet de référence en tenant compte du nombre de degrés de liberté.	144
5.1	Connexion entre les composants ARCS pour le tracé de l'erreur de reprojec- tion et des paramètres de mouvement au cours du temps.	150
5.2	Capture d'écran du navigateur.	151
5.3	Points retenus d'un élément 3D de l'imprimante.	156
A.1	Vidéo synthétique représentant l'assemblage et le désassemblage de deux ob- jets suivant une rotation autour de l'axe Z de l'objet 0.	185
A.2	Vidéo synthétique représentant l'assemblage et le désassemblage de deux ob- jets suivant une rotation autour de l'axe X de l'objet 0.	185
A.3	Vidéo synthétique représentant l'assemblage et le désassemblage de deux ob- jets suivant une translation selon l'axe X de l'objet 0.	185
A.4	Vidéo synthétique représentant le désassemblage de deux objets suivant un mouvement de vissage selon l'axe X de l'objet 0.	186

A.5	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une translation selon l'axe Y de l'objet 0.	186
A.6	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une translation selon l'axe Z de l'objet 0.	187
A.7	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant un pivot selon l'axe X de l'objet 0.	187
A.8	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une rotation autour de l'axe Z de l'objet 0.	188
A.9	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une rotation autour de l'axe Y de l'objet 0.	188
A.10	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une translation selon l'axe X de l'objet 0.	188
A.11	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant un vissage selon l'axe Y de l'objet 0.	189
A.12	Vidéo synthétique représentant le désassemblage de deux objets suivant une rotation autour de l'axe Y de l'objet 0.	189
A.13	Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant un vissage selon l'axe Z de l'objet 0.	189
B.1	Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Y de l'objet 0.	190
B.2	Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Z de l'objet 0.	191
B.3	Vidéo synthétique représentant le désassemblage de deux pièces suivant un pivot autour de l'axe Y de l'objet 0.	191
B.4	Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Z de l'objet 0.	191
B.5	Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0.	192
B.6	Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Y de l'objet 0.	192
B.7	Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Z de l'objet 0.	192
B.8	Vidéo synthétique représentant le désassemblage de deux pièces suivant un pivot autour de l'axe Y de l'objet 0.	193
B.9	Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Z de l'objet 0.	193
B.10	Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0.	193

C.1	Résultats de la classification des genres à l'aide du Fisherface.	197
C.2	Insertion d'une éolienne dans un environnement extérieur. [Ota et al., 2010] .	199
C.3	MARWind. [Dekker et al., 2013]	200
C.4	Smart Vidente. [Schall et al., 2013]	201
C.5	Utilisation de SiteLens pour l'analyse spatiale des données. Ici, il s'agit de l'analyse de la production de CO sur une rue donnée. [White and Feiner, 2009] 201	
C.6	Étapes pour la détection de la ligne d'horizon.	203
C.7	Rendu final de la superposition.	203

Liste des tableaux

1.1	Propriétés des différentes technologies de visualisation	36
2.1	Quelques systèmes de RA mobile.	56
3.1	Délai de détection par rapport à l’instant de changement effectif. (<i>Algo</i> = <i>algorithme utilisé, Mvt</i> = <i>type de mouvement se déroulant dans la scène</i>) . . .	111
4.1	Tableau récapitulant les corrélations entre l’erreur dans l’estimation de la ma- trice de transformation et différents paramètres pouvant l’influencer.	130
4.2	Tableau récapitulant les corrélations entre la variation de la matrice de trans- formation image par image et les métriques que nous avons retenues.	133

Introduction générale

Etant enfants, nous nous imaginions des univers imaginaires avec nos compagnons de jeu afin d'endosser des rôles définis entre nous. Ce fleuve que nous avons inclus dans notre scénario et localisé spatialement, nous savions que nous devions le sauter dans nos courses-poursuites. Plus âgés, nous sommes devenus plus pragmatiques, mais nous sollicitons toujours notre imagination pour se représenter une maison en construction ou à aménager, des pièces à assembler, le processus pour réparer des objets ou pour réaliser une recette de cuisine. Cependant, l'imagination est difficile à communiquer et à transmettre. Cela est d'autant plus difficile et plus critique lorsqu'il s'agit de concepts ou d'instructions abstraites. Chaque personne perçoit les éléments décrits suivant son expérience et son point de vue. Une même description peut donc générer autant de représentations que d'individus impliqués. Ainsi la réalité augmentée, en superposant des entités virtuelles pouvant représenter notre imaginaire sur un environnement réel, permettrait de partager les mêmes informations abstraites de manière efficace.

Il s'agit d'un domaine qui a été exploité assez rapidement dans divers secteurs du fait de son utilité et de ses possibles applications. Le marketing, la publicité, le tourisme, l'architecture y ont trouvé un moyen pour attirer de plus en plus de monde en suscitant la curiosité du public afin de vendre ou de relancer leurs produits ou leurs sites [[Lee and Billingham, 2012](#), [Liestol and Morrison, 2013](#), [Hagbi et al., 2011](#), [Zendjebil, 2010](#)]. Cependant, c'est dans les jeux vidéos que la réalité augmentée connaît le plus de succès. On l'observe notamment avec le jeu Pokémon Go de Nintendo, Niantic et Pokémon Company sorti récemment. Malheureusement, dans le domaine industriel, elle reste encore balbutiante car jugée, à ses débuts, trop intrusive et pas toujours adéquate à un environnement industriel. Avec la sortie d'appareils de visualisation de réalité augmentée beaucoup plus compacts, plus performants et plus légers, des industriels réinvestissent dans ce domaine et explorent l'adéquation de ce domaine au contexte industriel.

En effet, dans une optique de réduction des coûts et des temps de production et de recherche de solutions pouvant répondre aux inconvénients liés à l'arrêt des machines de production pour des raisons de maintenance ou pour des spécifications de certains produits, les industriels explorent des moyens leur permettant de satisfaire à leurs attentes. Une des parties du processus

de production dans laquelle la réalité augmentée peut être intégrée et où les industriels peuvent faire des économies est celui de l'apprentissage des opérateurs. Une formation appropriée et adéquate des agents industriels peut être moins coûteuse qu'une reconfiguration de tout le processus de production. De plus, du fait de leur flexibilité, ces derniers peuvent répondre assez facilement et rapidement à des besoins spéciaux de l'industriel. Cette formation nécessite souvent un opérateur suffisamment expérimenté et pédagogue pour transmettre les compétences requises. Malheureusement, celui-ci peut ne pas être disponible. Alors des manuels d'instructions peuvent y pallier. Mais ces derniers peuvent être parfois plus encombrants qu'un système de réalité augmentée, désuets, insuffisamment informatifs et assez indigestes. Ces manuels peuvent ne pas répondre aux spécificités d'un apprentissage, ni au niveau de technicité des apprentis. Pour qu'il soit effectif, complet et durable, un apprentissage sollicite nos différents sens notamment la vue, l'ouïe et le toucher si nous restons dans un contexte industriel. Pour qu'une connaissance soit effectivement acquise, le meilleur moyen d'apprendre et de se l'approprier est de passer par la visualisation de la tâche à effectuer et par l'expérience. La réalité augmentée peut répondre à ces besoins en affichant des informations pertinentes au bon moment au bon endroit.

De nombreuses applications de réalité augmentée ont d'ailleurs vu le jour pour servir d'aide de maintenance aux opérateurs [Feiner et al., 1993, Friedrich et al., 2002, Billingham et al., 2008, Hakkarainen et al., 2008]. Toutefois, elles sont souvent restées dans le domaine de la recherche. Ces applications avaient montré qu'il était possible pour l'opérateur d'améliorer ses performances et sa productivité [Sääski et al., 2008, Leu et al., 2013]. Cependant, les méthodes d'interaction avec le système afin de visualiser les instructions successives ruinaient l'efficacité apportée par la réalité augmentée. Elles nécessitaient d'interrompre sa tâche de maintenance afin d'appuyer sur un menu pour indiquer que celle-ci était terminée et qu'il faudrait afficher la suite du processus. Ce type de système de suivi du scénario de maintenance peut facilement déconcentrer l'utilisateur, créer une fatigue visuelle, cognitive et physique et induire ainsi des erreurs dans l'exécution de la tâche de maintenance. De plus, il ne permet pas au système d'identifier et de signaler à l'utilisateur les erreurs qu'il aurait pu commettre. Celui-ci reste seul juge de la bonne exécution des tâches qu'il a accompli.

C'est dans ce contexte que la société Wassa, spécialisée dans le web et les solutions mobiles, et l'équipe IRA2 du laboratoire IBISC, spécialisée dans la réalité augmentée, la réalité virtuelle et la robotique, ont initié ce sujet de thèse. L'objectif était de se positionner dans l'aide non seulement à l'apprentissage dans un contexte industriel mais également au contrôle du processus en s'aidant d'appareils de visualisation mobiles. Il s'agissait ainsi de concevoir un système de RA permettant :

- de fournir à l'utilisateur les instructions de maintenance de manière automatique et en

temps réel. Il s'agissait de proposer une méthode permettant de maintenir une certaine fluidité dans les scénarios de maintenance afin de préserver la concentration de l'utilisateur le long de la simulation.

- de contrôler la bonne réalisation de la tâche et d'identifier les erreurs commises par l'opérateur à travers des dispositifs mobiles. Une notification doit être affichée afin de signaler à l'utilisateur qu'une tâche a été mal exécutée ou qu'une pièce mécanique n'a pas été correctement placée. Le système doit pouvoir enregistrer chacune de ces erreurs.
- de fonctionner sur n'importe quel appareil mobile en tenant compte des limitations liées à l'utilisation de ces appareils, notamment concernant leur capacité de calcul et de stockage.

Afin de répondre à ces spécificités, nous nous sommes intéressés aux données qui étaient calculées lors du processus de réalité augmentée et qui pouvaient contenir des informations pertinentes sur ce qui se passe dans l'espace de travail de l'opérateur. Ces informations peuvent être très utiles à exploiter sans introduire des procédures coûteuses supplémentaires dans le système de réalité augmentée. Comme les tâches de maintenance sont essentiellement constituées de phase d'assemblage et de désassemblage, il s'agit ainsi d'identifier ces dernières. En définissant, dans un premier temps, l'assemblage comme une transformation rigide entre des pièces mécaniques invariante au cours du temps, il s'avère qu'une métrique comme l'erreur de reprojction correctement définie et astucieusement associée à une configuration de modèles 3D permet d'identifier et de localiser dans le temps ces deux phases. Nous avons ainsi, par la suite, proposé deux méthodes que nous avons nommées KSUM et KSUMratio permettant de localiser en temps réel le changement de phase. Ces derniers présentent dans nos expérimentations des résultats plutôt satisfaisants en terme de délais de détection et de temps de calcul par rapport à d'autres algorithmes de détection de ruptures. Détecter et identifier dans quelle phase se trouve l'utilisateur permettrait d'automatiser l'affichage des instructions de l'utilisateur sans qu'il n'ait besoin d'intervenir ou d'interrompre de manière incessante son activité.

Cependant, comprendre et reconnaître ce qui se déroule dans l'espace de travail est un challenge dans la communauté de vision par ordinateur. Cela peut être une aide pour contrôler les actions de l'utilisation, identifier les erreurs qu'il pourrait commettre et les corriger durant la tâche de maintenance. C'est dans cette optique que nous avons explorée, dans un deuxième temps, les transformations relatives existantes entre les différentes pièces des objets à maintenir. Si nous définissons, cette fois-ci, l'assemblage comme une liaison mécanique avec un certain nombre de degrés de liberté fixe et le désassemblage comme l'apparition d'un degré de

liberté supplémentaire, nous pouvons à l'aide des transformations relatives identifier quand il y a un assemblage et si celui-ci est correctement effectué. Cette étape permet ainsi de notifier au système réalité augmentée les erreurs commises par l'utilisateur et d'effectuer un contrôle de la tâche de maintenance. Comme, dans un contexte réel, les données de mouvement enregistrées sont énormément bruitées, nous avons exploré un certain nombre de paramètres en amont de la détermination et de la caractérisation du mouvement qui peuvent influencer les données bruitées. Nous avons proposé une méthode permettant de les corriger en utilisant une méthode d'apprentissage notamment des SVM (support à vecteurs de machines). Les résultats obtenus apparaissent beaucoup plus lisses et peuvent être exploités pour réaliser une comparaison avec les données de mouvement enregistrées dans notre banque de modèles.

Nous avons implémenté l'ensemble des modules de notre système de réalité augmentée grâce au framework ARCS.js (Augmented Reality Component System) [Didier et al., 2009, Didier et al., 2012] développé au laboratoire IBISC. Celui-ci nous permet de tirer parti des capacités du Javascript et du HTML5 liées notamment à la portabilité sur n'importe quel appareil mobile dont un navigateur web est installé.

Ainsi le manuscrit est organisé de la manière suivante :

- Dans un premier temps, dans le chapitre 1, nous ferons un tour d'horizon de la réalité augmentée. Nous reverrons les définitions de la réalité augmentée et sa taxonomie. Nous y proposerons également une taxonomie relative à l'interaction de l'utilisateur avec l'environnement réel et virtuel et à l'interactivité du l'environnement réel et virtuel. Nous y présenterons les différents types d'appareils de visualisation des augmentations en indiquant les modèles les plus récents selon leur type, ainsi que leurs avantages et inconvénients. Nous nous intéresserons également aux méthodes d'estimation de la pose de la caméra et de suivi.
- Dans le chapitre 2, nous explorerons les tendances des applications de la réalité augmentée mobile de manière générale puis de manière spécifique dans le domaine industriel. Nous y présenterons également le processus de suivi des scénarios en réalité augmentée.
- Dans un troisième temps, nous présenterons les contributions que nous avons apportées, nos hypothèses, nos démarches, nos expérimentations et nos résultats. Notamment, dans le chapitre 3, nous montrerons comment la métrique choisie, *ie* l'erreur de de reprojection, permet d'identifier les phases d'assemblage et de désassemblage. Nous y présenterons notre approche pour la détection de changement de contexte. Cette

approche a fait l'objet d'une publication (cf. [[Rukubayihunga et al., 2015](#)]).

- Dans le chapitre 4, nous analyserons également les transformations relatives afin de réaliser un contrôle des processus de maintenance. Nous y identifierons les paramètres pouvant influencer le comportement des données de mouvement calculées et nous proposerons une méthode permettant de lisser les courbes de mouvement en prenant en compte cette influence. Nous y introduirons également notre approche nous permettant de comparer les mouvements relatifs à celui établi hors ligne dans notre banque de modèles. Cette partie a fait l'objet de publications [[Rukubayihunga et al., 2016b](#), [Rukubayihunga et al., 2016a](#)].
- Dans le chapitre 5, nous spécifierons le système implémenté en utilisant le framework ARCS (Augmented Reality Component System) [[Didier et al., 2009](#), [Didier et al., 2012](#)], développé au laboratoire IBISC, qui permet une implémentation rapide de systèmes de réalité augmentée. Nous y présenterons les différentes méthodes utilisées en partant de la reconnaissance des objets impliqués dans notre scénario de maintenance à l'évaluation du système.
- Nous terminerons avec des propositions pour aller plus loin dans le cadre de l'apprentissage et du contrôle des tâches de maintenance à l'aide des systèmes de réalité augmentée mobiles.

Première partie

Etat de l'art

Chapitre 1

Tour d'horizon de la réalité augmentée

1.1 Réalité Augmentée : définition et balayage du domaine

Le principal objectif de la réalité augmentée (RA) est, comme l'expression l'indique, de rehausser la perception de la réalité. Il s'agit, dans la majorité des cas, d'introduire, dans l'environnement réel perçu par l'utilisateur, des éléments virtuels, tels que des images de synthèse, des objets virtuels, du texte, des symboles, des schémas, des graphiques, etc.

[Azuma, 1997] répertorie les trois principales caractéristiques des applications de réalité augmentée. Premièrement, celles-ci doivent combiner le réel et le virtuel c'est-à-dire elles doivent pouvoir incruster des éléments virtuels dans un environnement réel en acquérant des informations du monde réel et en ajoutant, par la suite, des objets virtuels. Deuxièmement, elles doivent être interactives et fonctionner en temps réel. Enfin, il doit exister une cohérence dans l'espace 3D donc une homogénéité spatiale qui doit être assurée par le système de RA en temps réel également. Autrement dit, le système de RA doit pouvoir assurer un recalage correct, *ie* un bon alignement, des objets virtuels sur le monde réel. L'utilisateur doit pouvoir percevoir la profondeur et la perspective lors du rendu. Ces différents critères formulés par Azuma excluent les films à effets spéciaux qui ne comprennent pas l'interactivité dans leur processus et le caractère "*temps réel*" des systèmes de RA. En effet, ces derniers incluent des images de synthèse dans leur processus après l'acquisition des données de l'environnement réel *ie* en post-production. Donc ils incluent une étape de post-traitement des données visuelles et sonores dans le processus de production du film excluant, de ce fait, le facteur temps comme caractéristique importante.

Le continuum de Milgram [Milgram and Kishino, 1994] permet de mieux situer le concept de réalité augmentée. Ce dernier introduit, par la même occasion, la notion de réalité mixte qui permet d'inclure toutes les variantes du réel au virtuel donc tous les degrés de mixage des éléments virtuels ou réels. La figure 1.1, permet d'illustrer cette représentation de Milgram.

Par ailleurs, [Fuchs et al., 2003] présentent la réalité augmentée comme étant un ensemble de techniques permettant d'associer un monde réel avec un monde virtuel, notamment en incrus-

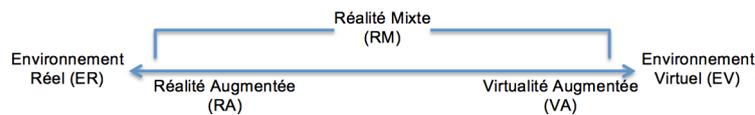


FIGURE 1.1 – Continuum de Milgram

tant des entités virtuelles dans des images réelles. Il est à noter que les associations entre monde réel et éléments virtuels ne se réalisent pas uniquement de manière visuelle, elles peuvent correspondre également à une intégration d'éléments sonores ou tactiles avec des retours d'effort. La définition de Fuchs n'écarte pas, ainsi, les films à effets spéciaux comme c'est le cas avec la définition d'Azuma.

[Fuchs et al., 2003] proposent, par la suite, une classification fonctionnelle des applications de réalité augmentée selon le niveau d'incrustation des éléments virtuels dans la scène réelle.

1.1.1 Taxonomie fonctionnelle de la réalité augmentée

Dans « *le traité de la réalité virtuelle* », [Fuchs et al., 2003] présentent la taxonomie fonctionnelle de la réalité augmentée. Ils distinguent cinq fonctionnalités selon le niveau d'incrustation des objets virtuels dans l'environnement réel. Cette taxonomie concerne principalement le degré de lien entre les entités réelles et virtuelles. L'utilisateur est exclu de cette classification que nous allons présenter dans un premier temps.

En tenant compte de la définition de [Azuma, 1997], la réalité augmentée inclut l'utilisateur comme acteur à part entier dans les systèmes de RA. Nous introduirons, dans la suite de cette section, une taxonomie impliquant non seulement les entités réelles et virtuelles mais également l'utilisateur.

a) Taxonomie fonctionnelle de [Fuchs et al., 2003]

Comme nous le précisons précédemment, la taxonomie de [Fuchs et al., 2003] tient compte du lien entre les éléments virtuels et réels.

i) Fonctionnalité « Superposition réel/virtuel »

Cette fonctionnalité correspond à une simple superposition d'éléments virtuels sur des images réelles sans aucune relation entre les entités réelles et virtuelles. Il peut s'agir, ici, d'un affichage sur un même écran de ces dernières sans aucune cohérence spatiale. Elles peuvent être visualisées soit simultanément sur l'écran total, soit sur deux fenêtres différentes de l'écran, soit sur une fenêtre et en fond d'écran.

ii) Fonctionnalité « Réalité documentée » ou « Virtualité documentée »

L'environnement réel est documenté par des éléments textuels, en général, affichés à l'écran. Un simple rapport d'information existe entre les éléments virtuels (des textes) et ceux réels mais sans raccord spatial effectif entre ces différents éléments (cf. fig. 1.2). Dans cette fonctionnalité, l'environnement réel peut être documenté à l'aide d'éléments virtuels et vice-versa.

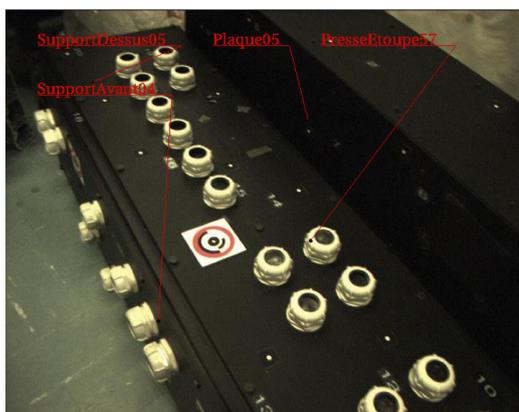


FIGURE 1.2 – Réalité documentée. Projet AMRA (Assistance à la Maintenance en RA). [Didier, 2005]

iii) Fonctionnalité « Réalité à compréhension ou visibilité augmentée »

Deux sous-niveaux peuvent exister selon l'apport de l'augmentation :

- La « Réalité à compréhension augmentée » correspond à une introduction, près des objets réels associés, d'informations sémantiques passives tels que des titres, des légendes ou des symboles (cf. fig. 1.3a). Ces derniers permettent d'obtenir des informations plus précises et plus localisées sur les objets réels.



(a) Réalité à compréhension augmentée



(b) Réalité à visibilité augmentée

FIGURE 1.3 – Réalité à compréhension ou visibilité augmentée. Projet AMRA. [Didier, 2005]

- La « *Réalité à visibilité augmentée* » correspond à une augmentation visuelle des objets réels (cf. fig. 1.3b). Les éléments virtuels (assimilés au modèle « fil de fer ») permettent une vue géométrique précise des contours des objets réels. Ceux-ci sont parfaitement alignés donc recalés aux objets réels.

iv) Fonctionnalité « Association visuelle du réel et du virtuel »

Dans cette fonctionnalité, des objets virtuels sont ajoutés à l'environnement réel. Deux cas sont possibles :

- soit les objets virtuels sont superposés aux images réelles sans occultations par des objets réels. On parle alors « *d'association par superposition* » (cf. fig. 1.4).
- soit les objets virtuels sont intégrés dans la scène réelle. Ainsi, il est possible aux objets réels d'occulter des éléments virtuels. On parle « *d'association tridimensionnelle* ».

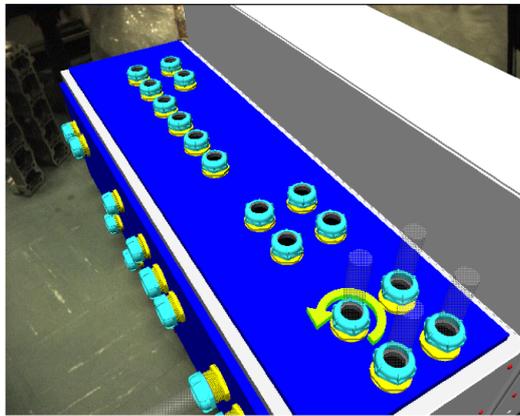


FIGURE 1.4 – Association visuelle du réel et du virtuel. Projet AMRA. [Didier, 2005]

v) Fonctionnalité « Association comportementale du réel et du virtuel »

Dans cette fonctionnalité, la modélisation des objets virtuels tient compte de leurs propriétés physiques suivant différentes lois physiques telles la gravité, les frottements, l'élasticité, la fluidité, etc. Des interactions géométriques et physiques sont établies entre les objets virtuels et réels.

vi) Fonctionnalité « Substitution du réel par le virtuel » ou « Réalité virtualisée »

Dans cette fonctionnalité, l'environnement réel est substitué par son modèle virtuel si celui-ci est connu selon le même point de vue. Celle-ci correspond à une immersion dans un environnement entièrement virtuel où l'utilisateur est complètement déconnecté de la réalité.

b) Taxonomie fonctionnelle d’interaction / d’interactivité

La taxonomie de [Fuchs et al., 2003] nous a paru incomplète du fait de l’absence de l’utilisateur et de la notion d’interaction. Selon la définition introduite par [Azuma, 1997], trois éléments interviennent dans le système (cf. fig. 1.5). Il s’agit des objets réels, des objets virtuels et des acteurs humains. Nous proposons, ainsi, de définir une taxonomie fonctionnelle, non pas en fonction du niveau d’incrustation des éléments virtuels dans la scène réelle, mais plutôt suivant le sens d’implication de chaque élément par rapport aux autres dans le système de RA. Quatre fonctionnalités semblent le mieux la représenter.

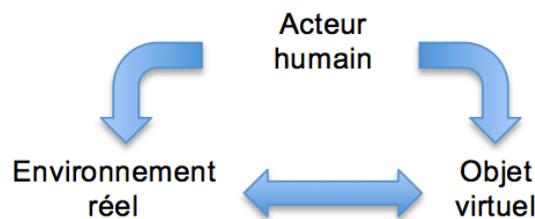


FIGURE 1.5 – Interaction avec l’utilisateur

i) Interaction entre l’homme et le monde réel

L’interaction de l’homme sur le monde réel est celui que nous connaissons habituellement sans aucun artifice ou matériel informatique. Nous pouvons agir sur les objets de notre environnement et ces derniers peuvent nous influencer à travers nos différents sens (vue, odorat, toucher, ouïe et goût).

ii) Interaction humaine sur le monde virtuel

L’interaction de l’homme sur le monde virtuel correspond à l’ensemble des techniques d’interactions établies en réalité virtuelle. [Bowman et al., 1999a] les classifie suivant le type de tâches d’interaction :

- la navigation regroupant les techniques permettant de se déplacer visuellement à l’intérieur du monde virtuel et de l’explorer,
- la sélection regroupant les techniques permettant de choisir les éléments virtuels que l’utilisateur désire manipuler, déplacer ou modifier,
- la manipulation afin de modifier structurellement, de déplacer ou de faire tourner les éléments virtuels dans l’environnement virtuel,
- le contrôle de menus qui permet de prendre en main l’application de réalité virtuelle et de la contrôler.

D’autres comme [Coomans and Timmermans, 1997] les classifient suivant leur modalité *ie*

linguistique / non linguistique, analogique / non analogique,... ou comme [Djelil, 2013] suivant la modalité "naturelle" (main, voix, regard, mouvement du corps, mouvement des lèvres, activité cérébrale, expression faciale) utilisée pour interagir avec l'environnement virtuel.

iii) Interaction humaine sur le monde virtuel et sur le monde réel

Dans un contexte de réalité augmentée, l'homme peut interagir aussi bien dans l'environnement réel que dans l'environnement virtuel. Le rendu affiché à l'utilisateur doit ainsi tenir compte des changements opérés dans la structure de l'environnement réel. Nous proposons, dans ce contexte, de classer ces interactions suivant l'implication de l'utilisateur dans le système RA.

Utilisateur spectateur : Cette partie regroupe l'ensemble des techniques laissant l'utilisateur spectateur de la scène virtuelle. Il ne peut et ne doit ni agir sur les éléments réels, ni sur les éléments virtuels. Le système de réalité augmentée lui affiche une simulation prédéfinie dans un environnement connu ne tolérant pas de modifications dans sa structure globale. Les séquences du scénario sont affichées sans aucune intervention de l'utilisateur.

Utilisateur semi-passif du virtuel : Dans cette partie, l'utilisateur ne peut agir que sur les objets réels, les déplacer, les faire tourner, les faire apparaître ou disparaître. Ces objets peuvent être utilisés comme interfaces pour commander le système. L'utilisateur peut également, dans cette fonctionnalité, s'aider de sa gestuelle pour contrôler le système de RA.

Utilisateur semi-passif du réel : Dans cette partie, l'environnement réel est fixé et immuable. L'application est calibrée sur cet environnement. L'utilisateur, lui, ne peut agir que sur les objets virtuels. Comme, pour le cas précédent, il peut les déplacer, les faire tourner, les faire apparaître ou disparaître, les modifier structurellement. Cependant, l'immuabilité de l'environnement réel n'est effectif qu'avec l'utilisateur. Celui-ci n'a aucun effet sur le réel de manière directe. Les objets virtuels peuvent, en revanche, avoir un effet sur les objets réels afin de conserver la cohérence spatiale entre les entités réelles et virtuelles. L'utilisateur contrôle bien l'environnement virtuel mais ne peut contrôler le monde réel qu'à travers ces éléments virtuels. Ceci nous ramène ainsi à la réalité virtualisée.

Utilisateur acteur à part entière : Dans cette partie, l'utilisateur peut interagir aussi bien avec les objets réels que les objets virtuels. Il peut manipuler ces derniers à sa guise tel qu'il le ferait avec de vrais objets *ie* les déplacer, les tourner, les modifier, les faire disparaître ou apparaître dans le champ de vision. Par ailleurs, il peut se déplacer dans un environnement réel relativement large. Cette fonctionnalité permet de donner l'illusion à l'utilisateur qu'il a une

prise effective sur les objets virtuels et qu'il n'est pas limité à un espace de travail restreint.

iv) **Interactivité entre le monde réel et le monde virtuel**

Dans fonctionnalité, les objets réels peuvent agir sur l'environnement virtuel et les objets virtuels sur l'environnement réel soit à travers l'action humaine soit à travers les données recueillies des capteurs intégrés aux objets. Grâce aux systèmes de communication mis en place entre les objets réels entre eux et les objets réels et virtuels *ie* grâce à l'internet de objets, les changements de contexte opérés soit dans l'environnement réel, soit dans l'environnement virtuel vont entraîner une adaptation automatique respectivement des objets virtuels ou/et des objets réels. Les objets réels mis en avant ne sont pas uniquement ceux connectés ou ceux munis d'un capteur récupérant et analysant les données recueillies, ils peuvent également se mouvoir suivant les modifications dans l'environnement virtuel sans que l'utilisateur n'intervienne véritablement.

Cette fonctionnalité inclut non seulement la gestion visuelle des interactions *ie* des occultations réel/virtuel et des interpénétrations mais également la gestion des lois physiques (forces, collisions, objets déformables,...) entre les objets réels et virtuels.

1.1.2 **Technologies de visualisation pour la réalité augmentée**

Les technologies de visualisation pour la RA peuvent être classées selon la procédure utilisées pour superposer les objets virtuels et réels [Billingham et al., 2015] ou selon le dispositif matériel utilisé. Dans cette section, nous avons préféré présenter la seconde classification selon le dispositif matériel.

a) **Les visiocasques**

Les visiocasques sont des technologies héritées de la réalité virtuelle. Il y a encore quelques années, ils ressemblaient à d'énormes casques à "moto". Aujourd'hui, ils se sont "miniaturisés" pour avoir la taille de lunettes. On distingue les visiocasques selon l'acquisition et la visualisation des informations de l'environnement réel. En effet, le processus de traitement des données de l'environnement réel sera réalisé différemment si l'on doit voir ces dernières directement ou indirectement.

i) **Visiocasques à vision directe**

Ce type de visiocasques est très populaire ces dernières années auprès des fabricants tels que Daqri, Epson, Google ou Laster. Quelques modèles sont présentés à la figure 1.6.

Les visiocasques à vision directe (correspondant aux *optical see-through head mounted displays* en anglais) comprennent dans leur système de visualisation des miroirs ou des prismes

semi-transparents (cf. fig. 1.8a page 32). Ces derniers sont placés en face des yeux de l'utilisateur et permettent à l'utilisateur de percevoir l'environnement réel à travers ce matériel. Un dispositif traitant des informations concernant les objets virtuels projette ceux-ci sur le miroir semi-transparent.

L'avantage le plus important des visiocasques à vision directe est celui de voir directement ce qui se passe dans la réalité, d'avoir une perception réaliste des distances entre les objets et d'obtenir une vision nette de l'environnement réel sans aucune distorsion due au matériel d'acquisition. De plus, le champ de vision du monde réel reste le même qu'en vision normale. Cet accès direct à l'information réelle convient aux applications de RA qui requièrent une certaine sécurité. Cela peut être le cas pour les applications dans les domaines industriel, médical ou militaire.



FIGURE 1.6 – Modèles de visiocasques à vision directe.¹

De plus, ce dispositif peut supporter des applications mobiles et multi-utilisateurs. Les utilisateurs peuvent se déplacer dans leur environnement de travail et collaborer entre eux en implémentant un système de communication entre les différents visiocasques [Bimber and Raskar, 2006].

Cependant, les visiocasques à vision directe présentent également quelques inconvénients qui limitent son utilisation par le public. Même s'ils ne pèsent plus aussi lourds qu'il y a

1. Les images des différents casques sont prises sur divers sites mais principalement sur les sites officiels des marques en question.

quelques années, ils restent encore disparates. Par exemple, le visiocasque de Daqri pèse 1kg (fig.1.6a), d’autres beaucoup moins comme celui Google (fig.1.6b) avec 43g ou celui de Senth (fig.1.6g) avec 45g. En moyenne, la plupart des visiocasques à vision directe pèsent une centaine de grammes, ce qui est relativement assez élevé lorsque l’on compare cette masse à celle des lunettes de vue qui pèsent en moyenne 20g et que l’on sait que la majorité du poids des lunettes notamment est supporté par le nez.

Par ailleurs, des problèmes d’accommodation de l’œil existent. Dans le cas du visiocasque à vision directe, l’œil adapte facilement sa vergence aux objets réels. En revanche, avec les objets virtuels qui sont affichés sur un écran en face de son œil, ce processus peut être assez perturbant.

Le fait que les objets virtuels soient ”plaqués” directement sur un écran semi-transparent donne un effet fantôme. En effet, le contraste et la transparence requise pour ce type de dispositif ne permettent pas de s’affranchir complètement de cet effet. De plus, les occultations mutuelles sont souvent incohérentes et assez mal gérées également.

ii) Visiocasques à vision indirecte

Très peu d’entreprises comme Canon ou Vuzix continuent à explorer ce type de visiocasques ces dernières années. Quelques visiocasques à vision indirecte sont présentées à la figure 1.7.



FIGURE 1.7 – Modèles de visiocasques à vision indirecte.²

Ce dispositif utilise un système de caméras (cf. fig. 1.8b). Selon la méthode d’estimation de la pose implémentée dans le système RA, une ou plusieurs caméras récupèrent les données visuelles de l’environnement réel. L’estimation de la pose de la caméra et des calculs graphiques sont effectués. Puis le résultat de l’incrustation des objets virtuels dans cet environnement est retransmis sur un petit écran placé en face des yeux de l’utilisateur.

Comme pour les visiocasques à vision directe, les visiocasques à vision indirecte supportent des applications mobiles et multi-utilisateurs. De plus, ils peuvent correctement gérer les occultations mutuelles c-à-d occultations réel sur virtuel et virtuel sur réel. En effet, du fait que les

2. Les images des différents casques sont prises sur divers sites mais principalement sur les sites officiels des marques en question.

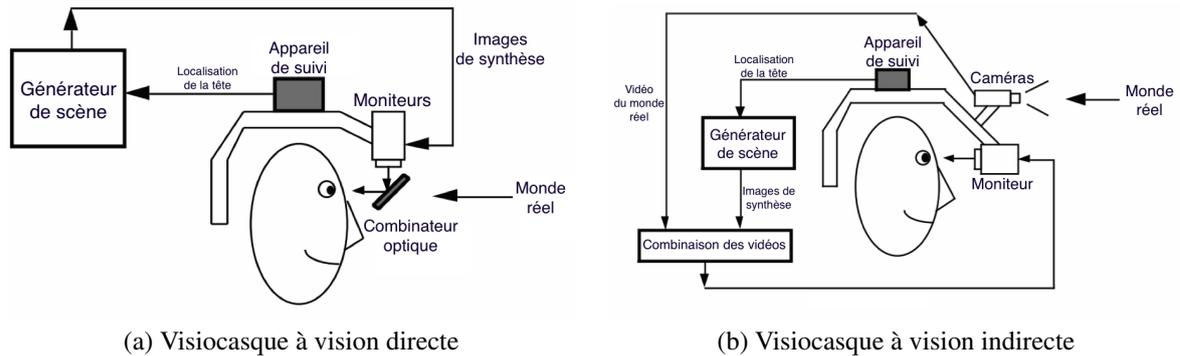


FIGURE 1.8 – Principes des visiocasques. [Azuma, 1997]

données réelles sont acquises, cela permet de les manipuler et d'afficher ensuite des séquences d'images corrigées et augmentées de l'environnement réel [Bimber and Raskar, 2006].

En revanche, ils sont assez lourds, beaucoup plus que les visiocasques à vision directe. Ils peuvent être encombrants et intrusifs [Papagiannakis et al., 2008].

Par ailleurs, la résolution du rendu final est limitée par le matériel utilisé. Avec la miniaturisation des composants afin d'obtenir des visiocasques de plus en plus légers et peu encombrants, reproduire les capacités visuelles humaines, notamment sa résolution correspondant à 12 000 sur 7 200 pixels, devient difficile [Kiyokawa, 2012, Re, 2013]. Même si, avec les visiocasques à vision directe, les données de l'environnement réel se trouvent peu affectées, les incrustations des éléments virtuels, en revanche, le sont. Tandis qu'avec les visiocasques à vision indirecte, aussi bien les données réelles que virtuelles sont limitées par la résolution du dispositif.

D'autre part, le champ de vision est également restreint. En général, à cause des limitations optiques et des techniques de transmission des images, la vision frontale est souvent privilégiée lors de la conception des systèmes de visualisation par rapport à la vision périphérique qui n'est pratiquement pas prise en compte [Re, 2013]. Ainsi, un champ de vision étroit est souvent privilégié par rapport à un champ de vision large.

Des problèmes d'accommodation peuvent apparaître lorsque les objets sont proches. L'image visualisée par l'utilisateur peut lui sembler très floue car le focus de ce dispositif reste fixe. De plus, le point de vue affiché sur l'écran est celui des caméras et pas forcément celui de l'utilisateur. Il apparaît ainsi un décalage entre ce qui est vu et ce qui est attendu par l'utilisateur entraînant quelques problèmes cognitifs. Afin de pallier cela, la mise en place de miroirs dans le dispositif est privilégiée afin de corriger le point de vue.

En outre, les traitements effectués pour le suivi entraînent une certaine latence entre les informations récupérées et celles retransmises, ce qui peut gêner l'utilisateur surtout dans le cadre d'une interaction et donc un inconfort lors des mouvements rapides de la tête. En revanche, il est possible avec ce dispositif de synchroniser correctement l'affichage des objets virtuels sur l'environnement réel.

b) Projecteur rétinien

Le dispositif permet de projeter l’image résultante directement sur la rétine de l’utilisateur à l’aide d’un laser à faible voltage (cf. fig. 1.9).

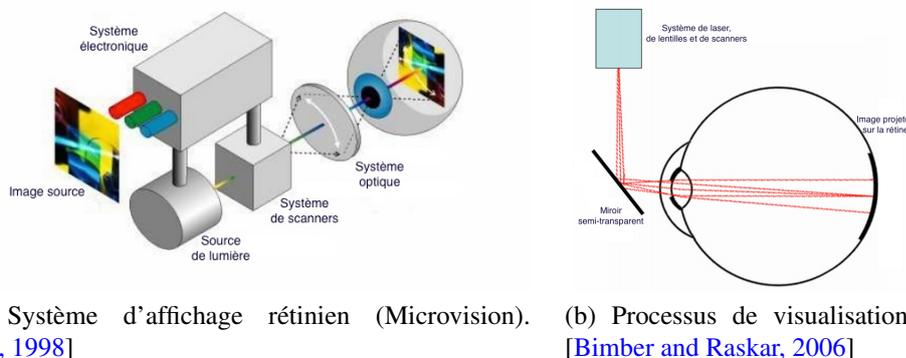


FIGURE 1.9 – Projecteur rétinien

Ce dispositif présente un contraste et une luminosité élevés [Bimber and Raskar, 2006]. Il assure une grande résolution, un champ de vision large, consomme peu d’énergie et permet des applications mobiles et multi-utilisateurs. Cependant il ne permet pas une accommodation oculaire car la distance focale reste fixe. De plus, son coût reste élevé. Des prototypes binoculaires tels que l’Avegant Glyph ont été introduits sur le marché (cf. fig. 1.10).



FIGURE 1.10 – Projecteurs rétiniens Avegant Glyph.³

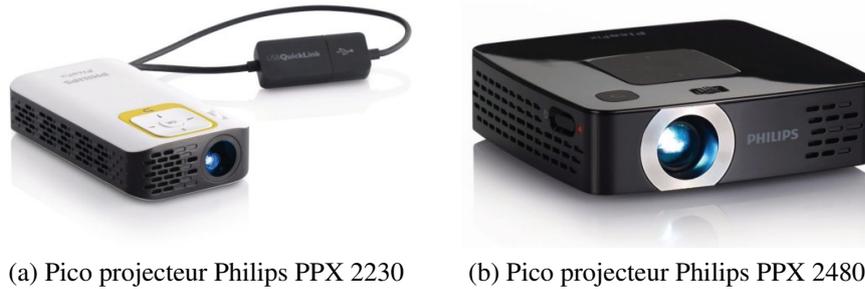
Malgré les avantages présentés par ce dispositif, les utilisateurs sont encore réticents à l’adopter du fait de la projection de l’augmentation directement sur la rétine.

c) Technologies de visualisation fixe

D’autres dispositifs ne requièrent pas d’être portés ou tenus par l’utilisateur. Ils sont généralement fixés dans l’environnement de travail.

3. <https://www.avegant.com/product>

4. <http://www.philips.fr/c-m-so/projecteurs-mobiles>



(a) Pico projecteur Philips PPX 2230 (b) Pico projecteur Philips PPX 2480

FIGURE 1.11 – Pico projecteurs⁴

Les interfaces basées sur la vidéo (ou *screen based video see-through* en anglais) sont composées d'un ordinateur et d'une caméra. La caméra récupère les informations de l'environnement réel puis les transmet à l'ordinateur. L'écran associé à l'ordinateur sert d'interface de visualisation de l'augmentation réalisée. Il s'agit de la solution la moins coûteuse parmi les dispositifs de visualisation. Cependant, l'immersion et l'interaction ne sont pas effectives vu que la visualisation est entièrement indirecte.

Les interfaces d'affichage optique (ou *spatial optical see-through display*) correspondent à une projection sur des miroirs ou des écrans semi-transparents plus grands que ceux utilisés par les visiocasques à vision directe. L'accommodation oculaire est gérée par ce type de dispositif. Le champ de vision est plus large et la résolution des images plus élevée. De plus, la calibration est plus facile et plus stable. Cependant quelques artéfacts persistent notamment les problèmes d'occultations et le « découpage » des objets virtuels qui n'est pas naturelle.

Les interfaces de projection spatiale (ou *projections based spatial display*) consistent en des projections directes dans l'environnement réel. Ces dispositifs présentent un champ de vision moins restreint que les dispositifs présentés précédemment, une résolution élevée et une accommodation oculaire qui est gérée. Cependant, les ombres ne sont pas gérées. Ils nécessitent une projection sur une surface plane au risque d'obtenir des images résultantes floutées. De plus, la plupart des projecteurs ne réalisent un focus que sur une unique surface réfléchissante. Par ailleurs, plus le nombre de projecteurs utilisés est élevé, plus la complexité devient grande. Ceci est dû à l'étape de synchronisation des projecteurs, aux réglages des superpositions, à la détermination de l'alignement géométrique des augmentations et des objets réels ou à la calibration des couleurs.

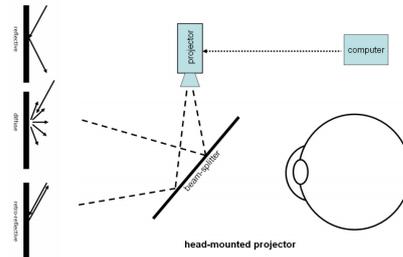
Les pico projecteurs sont des projecteurs miniatures qui peuvent afficher des images de même qualité que les projecteurs conventionnels (cf. fig. 1.11). Ce sont des appareils facilement transportables dans une poche, par exemple.

Ces technologies sont appropriées pour des démonstrations grandeur nature avec des interactions avec les utilisateurs limitées.

d) Dispositifs de projection portatifs



(a) Prototype de projecteur portatif. [Azuma et al., 2001]



(b) Système de projecteur portatif. [Bimber and Raskar, 2006]

FIGURE 1.12 – Dispositif de projecteur portatif.

Sur ce dispositif (cf. fig 1.12), l’image résultante est réfléchiée devant les yeux de l’utilisateur à l’aide d’une surface rétro-réflexive placée dans l’environnement et d’un couple de projecteurs située sur la tête de l’utilisateur. [Bimber and Raskar, 2006] indiquent que ce dispositif permet des applications mobiles et multi-utilisateurs. Son champ de vision est large. Il permet également de résoudre des problèmes liés à la distorsion et aux parallaxes. Cependant, la luminosité est très dépendante de l’environnement. Ainsi son utilisation reste effective que dans un environnement contrôlé, donc en intérieur. Il peut être lourd et encombrant. De plus, la résolution des images est limitée et ce dispositif nécessite l’installation d’une surface rétro-réflexive.

e) Dispositifs de visualisation de poche



(a) iPhone



(b) Tablette Android



(c) Tablette Tango

FIGURE 1.13 – Dispositif de visualisation de poche.⁵

Il s’agit de dispositifs portables tels que les smartphones, les tablettes PC, les assistants numériques personnels (PDA), les PC ultra mobiles, etc (cf. fig. 1.13). Il s’agit d’une solution peu coûteuse pour la réalité augmentée mobile. Cependant, leur champ de vision est limité et la résolution de l’écran de l’appareil dépend des caractéristiques optiques de la caméra. La perception et l’expérience des augmentations à l’aide de ces dispositifs peuvent être parfois

5. Les images des différents casques sont prises sur divers sites mais principalement sur les sites officiels des marques en question.

	VVD	VVI	PR	PP	DP	IV	IAO	IPS
Portabilité	+	+	+	+	+	-	-	-
Résolution	-	-	+	-	+/-	+/-	+	+
Rendu	-	+/-	+	NA	+/-	+/-	+/-	+/-
Latence	+	-	NA	NA	NA	+/-	+/-	+/-
Champ de vision	-	-	+	+	-	-	+	+
Occultations	-	+	NA	NA	NA	+/-	-	NA
Poids	-	-	+	-	+	-	-	+
Confort	-	-	+/-	+/-	+/-	-	+/-	+/-

VVD : Visiocasques à vision directe VVI : Visiocasques à vision indirecte PR : Projecteur rétinien PP : Projecteur portatif DP : Dispositifs de poche IV : Interface basée vidéo
 IAO : Interface d'affichage optique IPS : Interface de projection spatiale

TABLE 1.1 – Propriétés des différentes technologies de visualisation

assez déstabilisantes pour l'utilisateur. En effet, la plupart des applications de RA mobile réalisent un rendu en se basant sur la perspective de l'appareil et non sur celle de l'utilisateur [Čopič Pucihar et al., 2013]. Ce dernier se retrouve à observer deux vues qui ne sont pas complémentaires, celle que l'écran du mobile lui renvoie et celle qu'il perçoit de l'environnement réel. Par ailleurs, le fait de devoir tenir le matériel afin de voir l'augmentation à travers l'écran peut limiter les interactions possibles entre le système de réalité augmentée et l'utilisateur. Ce fait peut également demander au système d'estimer en permanence la position et l'orientation de la caméra puisque l'utilisateur sera constamment en mouvement.

Même si, maintenant, les smartphones possèdent des processeurs plus puissants, des processeurs dédiés au graphisme, une résolution élevée pour certains et des divers capteurs tels que le GPS (Global Positioning System), des accéléromètres, des magnétomètres ou un capteur de profondeur, de nombreux problèmes persistent concernant l'augmentation de la scène réelle. Du fait des traitements effectués afin d'incruster les éléments virtuels et de la puissance de calcul de ces dispositifs, il peut exister une certaine latence entre la capture de l'environnement réel et l'affichage de l'image résultante. Cet obstacle lié aux performances de calcul de ces appareils nécessite l'optimisation des algorithmes utilisés. Par ailleurs, vu leur capacité de stockage ainsi que de calcul, il est souvent nécessaire d'instaurer une stratégie de communication avec un serveur qui gardera en mémoire la base de connaissance de l'application de réalité augmentée installée sur les mobiles et/ou qui effectuera une partie des opérations à réaliser si cela est nécessaire.

Le tableau 1.1 récapitule les caractéristiques des technologies de visualisation du contenu de réalité augmentée, les plus et les moins de chaque type de dispositif. Il s'avère que les interfaces de projection spatiale présentent de nombreux avantages aussi bien pour le rendu des augmentations, le champ de vision ou le confort. Cependant, ils nécessitent une configuration préalable

dans l'environnement de travail. Notre objectif étant laisser la possibilité à l'utilisateur de pouvoir bouger d'un environnement de travail à un autre, nous nous intéresserons, dans un premier temps, aux dispositifs de poche et testerons, par la suite, notre système sur des visiocasques qui permettront à l'utilisateur d'avoir les mains libres lors des interactions potentielles.

1.2 Suivi et estimation de la pose

Comme indiqué dans les sections précédentes, l’objectif d’un système de RA est de rehausser la perception de la réalité et de ”tromper” les sens de l’utilisateur en introduisant des objets virtuels dans l’environnement réel. Afin d’atteindre cet objectif, le système suit le processus indiqué sur la figure 1.14. Il effectue trois phases importantes :

- une étape de construction de la base d’éléments virtuels à introduire,
- une phase de capture des informations réelles
- et une phase d’alignement des caméras réelles et virtuelles et de rendu.

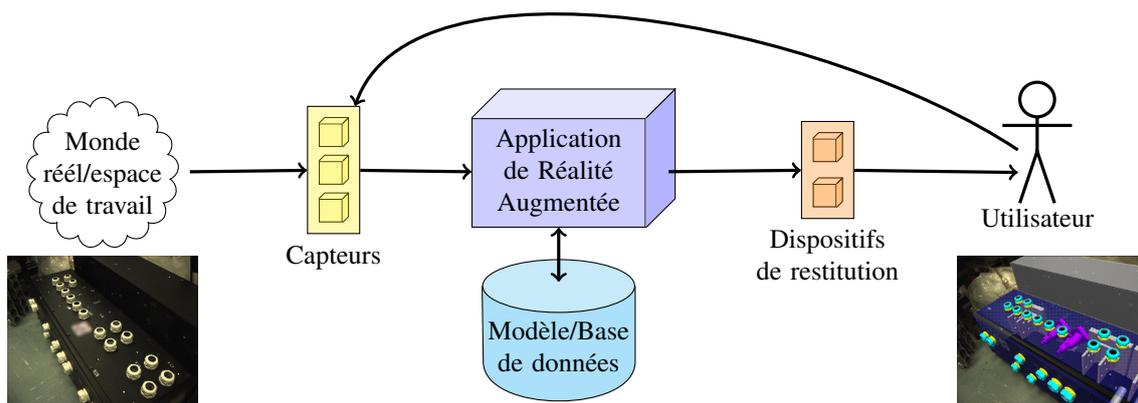


FIGURE 1.14 – Schématisation du processus d’un système de réalité augmentée. [Didier, 2014]

Ainsi, dans un premier temps, les éléments virtuels que l’on souhaite ajouter sont stockés dans une base de connaissance. Celle-ci est constituée au préalable, lors d’une phase hors ligne. Elle peut contenir :

- des données des augmentations. Il peut s’agir d’éléments textuels, sonores, visuels comme des images, des vidéos ou d’objets synthétiques. Le recueil de ces données peut devenir exhaustif lorsque le spectre d’application du système de RA devient large.
- des données sur la scène réelle obtenues *a priori* et exploitables pour le recalage. Elles peuvent être issues des résultats de la reconstruction préalable de la scène à travers des algorithmes de type SLAM (*Simultaneous Localization And Mapping*) [Smith and Cheeseman, 1986, Leonard and Durrant-Whyte, 1991] ou SfM (*Structure from Motion*) [Dellaert et al., 2000, Nilsson et al., 2010].

Dans une phase en amont, les informations de l’environnement réel sont relevées par des capteurs (capteurs de position et d’orientation ou de vision). Suivant le type de capteurs, l’estimation de la pose de la caméra réelle ou suivi, donc de sa position et de son orientation, diffère. La pose peut être déterminée et utilisée soit quasiment directement à l’aide des capteurs de position et d’orientation, soit à l’aide des algorithmes d’optimisation avec les capteurs de vision. Il est à rappeler que cette étape est importante et cruciale afin d’assurer un recalage correct des caméras réelles et virtuelles et un rendu cohérent.

1.2.1 Suivi basé capteurs de position ou d’orientation

Cette catégorie de suivi se base sur les capteurs intégrés au dispositif de réalité augmentée, en particulier dans des smartphones ou des visiocasques. Ils peuvent permettre de déterminer quasiment directement la position ou l’orientation de la caméra réelle.

La position de l’utilisateur est estimée à l’aide d’un réseau d’émetteurs-récepteurs. Par exemple, le GPS (*Global Positioning System*) qui est intégré dans la majorité des appareils mobiles, se base sur l’émission et la réception de signaux horodatés provenant de quatre satellites. La différence entre les moments d’émission et de réception ainsi que la trilatération de ces données permettent d’estimer les coordonnées de l’appareil dans le repère terrestre. Ce type de capteur est souvent utilisé pour la navigation à cause de sa « disponibilité » dans des espaces assez larges ainsi que pour son efficacité. En effet, son utilisation dans des espaces ouverts notamment en extérieur permet de recevoir des signaux satellitaires qui déterminent la position de l’utilisateur et donc de la caméra réelle. Par exemple, [Schmeil and Broll, 2006] utilisent un GPS standard pour la localisation en extérieur et le suivi de l’orientation dans le cadre d’une application de guidage en extérieur à l’aide d’un assistant virtuel. Tandis que [Azuma et al., 2006] utilisent une méthode hybride incluant le GPS afin d’améliorer le suivi de la position et le recilage dans le cadre d’opérations militaires.

La précision de suivi à l’aide du GPS varie entre 3 et 10 m selon la communication avec le satellite. En revanche, avec le D-GPS (*Differential GPS*) qui utilise des antennes relais terrestres, la précision est de 10 cm à 5 m. Cependant, il est à noter que dans les milieux urbains, la réception des signaux satellitaires peut être perturbée par les ombrages et les réflexions causées par les bâtiments aux alentours. Cela peut ainsi détériorer l’efficacité du suivi à l’aide de ce capteur.

Par ailleurs, comme le suivi à l’aide du GPS est inefficace en intérieur, il est possible d’utiliser des capteurs tels que le HSCPS (*High Sensitivity GPS*) ou l’UWB (*Ultra-Wide Band*). Cette dernière technologie est également un système de positionnement local pour des environnements assez larges en intérieur. Il a été commercialisé par Ubisense avec [Steggles and Gschwind, 2005]. Un réseau de capteurs est placé à des positions précises dans l’espace à recouvrir. Des tags communiquent avec ce réseau de capteurs afin de déterminer leur position 3D avec une précision d’environ 15 cm. Ces tags sont portés par les utilisateurs et fixés aux objets et sont constitués d’un receveur RF (radio-fréquence) et d’un transmetteur UWB.

Une autre catégorie de suivi à l’aide de capteurs de position utilise les antennes GSM (*Global System Mobile*) pour déterminer la position et l’orientation de l’utilisateur d’un mobile. Ces calculs sont réalisés en triangulant les signaux reçus par l’appareil. Cependant, ils sont soumis à de grandes variations, donc il y a de fortes probabilités que l’alignement des objets virtuels dans l’environnement réel soit inadéquat.

Comme le wifi est accessible dans la plupart des smartphones, des méthodes de suivi à l’aide

de cette technologie de communication sans fil ont été proposées dans la littérature. Les RSSI (*Received Signal Strength Indication*) des points d’accès permettent d’estimer la position du dispositif de réalité augmentée de manière plus efficace. Par exemple, [Peternier et al., 2006] utilisent une méthode de localisation basée wifi pour un système de réalité mixte sur PDA afin de visualiser des personnages virtuels. Tandis que [Liu et al., 2008] propose un algorithme de localisation basée wifi à l’aide d’un unique point d’accès comme aide à la navigation. L’avantage de ce type de suivi est qu’il n’est pas nécessaire d’inclure d’autres capteurs et il utilise des protocoles de communication déjà disponibles sur les mobiles. Cependant, comme la puissance du signal en intérieur n’est pas linéaire, les mesures utilisées pour le suivi ne seront pas précises [Pu et al., 2011].

Les capteurs inertiels tels que les accéléromètres, les gyroscopes, les boussoles électroniques ou d’autres capteurs rotationnels permettent d’estimer le point de vue de l’utilisateur, donc l’orientation de l’appareil mobile. Ils mesurent de hautes fréquences, donc des mouvements rapides du dispositif de réalité augmentée. Les mouvements lents de ce dernier peuvent entraîner du bruit et de la dérive provenant de la double intégration des données (*ie* à l’intégration des vitesses angulaires) et ainsi à l’accumulation des erreurs de mesure. De plus, il est à signaler que les capteurs magnétiques utilisés pour la détermination de l’orientation sont sensibles en présence d’objets ferreux dans l’environnement et aux perturbations électromagnétiques.

Même si l’estimation de la pose de la caméra à l’aide de capteurs de position et/ou d’orientation donne des résultats acceptables, il s’avère que la précision de ce type de suivi n’est pas encore effective.

Ainsi un autre capteur est le plus couramment utilisé. Il s’agit de la caméra qui permet d’exploiter des informations visuelles de l’environnement réel lorsqu’elles sont disponibles.

1.2.2 Suivi basé vision

Dans le cadre d’une estimation de la pose à l’aide de capteurs de vision, le système de RA capture l’image issue de l’environnement réel, puis extrait les informations visuelles pertinentes, *ie* des primitives pouvant correspondre à des points, des lignes ou des régions d’intérêt. Ces primitives sont, ensuite, mises en correspondance avec les primitives 3D stockées dans la base de connaissance. Ces correspondances permettront d’estimer la pose de la caméra à l’aide d’algorithmes consistant à résoudre des problèmes PnP (*Perspective-n-Point*). Ce processus est indiqué sur la figure 1.15.

Plusieurs méthodes ont été proposées dans la littérature pour estimer la pose de la caméra virtuelle à l’aide de capteurs optiques. [Billinghurst et al., 2015] suggèrent de les classer en trois catégories suivant le type de capteurs optiques *ie* les méthodes utilisant les capteurs à infrarouge, celles utilisant les capteurs de lumière visible et celles utilisant les capteurs de profondeur.

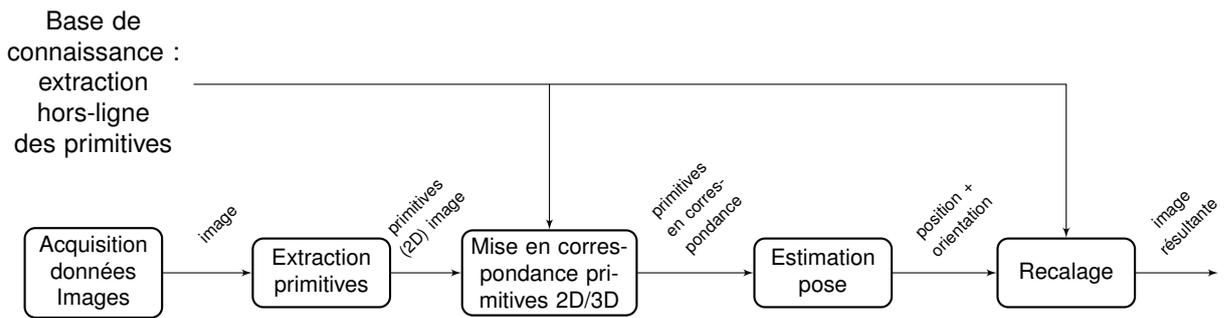


FIGURE 1.15 – Pipeline général pour l'estimation de la pose basée vision.

a) Suivi à l'aide de caméras à infrarouge

Ces méthodes sont similaires aux méthodes utilisant le RFID (*Radio Frequency Identification*). Elles fonctionnent sur le principe d'émetteur-récepteur. Des marqueurs sont installés dans l'environnement de travail afin d'émettre ou de réfléchir de la lumière infrarouge qui sera invisible à l'œil. En effet, la longueur d'onde dans le spectre infrarouge qui est dans l'intervalle de 780nm à 1mm est supérieure à celui de la lumière blanche, donc au spectre visible. Ainsi le système de suivi peut être constitué de deux types de "marqueurs" [Siltanen, 2012] :

- des dispositifs émettant de l'infrarouge telles que des diodes électroluminescentes infrarouges [Olwal, 2006]. Une caméra à infrarouge les capte et les filtre en éliminant la lumière ambiante. Les marqueurs correspondront aux points détectés visibles dans l'image acquise. Même si d'un premier abord, il semble que l'on récupère uniquement le positionnement des marqueurs de manière indifférente, il est possible de les identifier en instaurant un système de diodes clignotantes. Cependant il est alors nécessaire d'observer la scène pendant un certain temps pour l'identification. De plus, ce type de dispositifs nécessite une source d'énergie pour l'émission et leur mise en place est coûteuse en temps et complexe.
- des dispositifs rétro-réfléchissants qui vont renvoyer la lumière infrarouge reçue. Plusieurs sources de lumière infrarouge dirigées dans des directions différentes sont placées près de la caméra et émettent des ondes lumineuses qui sont réfléchies par ces marqueurs avec une faible diffusion. Ces marqueurs peuvent avoir des motifs différents permettant ainsi non seulement leur détection mais également leur identification. Ce type de dispositifs ne nécessite aucune source d'énergie sauf pour les sources de lumière.

Les méthodes d'estimation de la pose à l'aide de caméras infrarouges couvrent un environnement jusqu'à 6m et ont une précision de 0,03 à 0.1m [Huang et al., 2013]. Elles sont également précises suivant l'orientation angulaire, robustes à l'illumination et aux ombres et rapides. Cependant, la mise en place des marqueurs peut être important, complexe, très coûteuse et intrusive. Elles nécessitent une source d'énergie et un système de synchronisation des sources lumineuses. De plus, elles ne peuvent fonctionner que dans un environnement intérieur

contrôlé. En effet, des interférences peuvent apparaître en présence d’un rayonnement solaire ou de lumière fluorescente.

b) Suivi à l’aide de capteurs à lumière visible

Le capteur photographique de type CCD (*Charged Couple Device*) ou CMOS (*Complementary Metal-Oxide-Semiconductor*) est un des capteurs à lumière visible le plus populaire dans le domaine de la réalité augmentée. Du fait qu’il récupère les images de couleur ou en niveau de gris de l’environnement réel qui peuvent être analysées et utilisées pour gérer les problèmes de contraste et d’occultations mutuelles lors des augmentations, il constitue un outil le plus utilisé dans la communauté. Les méthodes s’aidant ce type de capteur peuvent être classée en deux catégories : celles utilisant des cibles codées, celles extrayant des descripteurs dits ”naturels”.

i) Suivi à l’aide de cibles codées

Il s’agit d’une méthode de suivi fréquemment utilisée dans des applications de réalité augmentée et qui, en général, surpasse la méthodes extrayant des descripteurs dits ”naturels”. Il est facile à implémenter et de nombreuses bibliothèques telles que ARToolKit [[Kato and Billingham, 1999](#), [Kato et al., 2000](#)] qui est très populaire dans la littérature scientifique, ARTag [[Fiala, 2005](#)], Cybercode [[Rekimoto and Ayatsuka, 2000](#)], InterSense [[Naimark and Foxlin, 2002](#)], ALVAR [[Siltanen, 2012](#)] ou ArUco [[Garrido-Jurado et al., 2014](#)] offrent la possibilité de détecter ces cibles, de les identifier et d’estimer facilement la pose (position et orientation) de la caméra ainsi que son suivi.

Ces marqueurs sont, généralement, de forme carrée et avec des bords noirs et un fond blanc. Leur partie centrale contient un code spécifique qui permet de les reconnaître (cf. fig. 1.16, 1.18, 1.17). Il s’avère que ce type de marqueurs est détectable facilement et de manière fiable si les paramètres de détection sont définis de manière adéquate. En effet, le contraste noir/blanc est plus facile à détecter que la différence de couleur. Ceci rend optimal les marqueurs de couleurs noire et blanche. De plus, comme pour estimer la pose la caméra de manière unique, il suffit de localiser quatre points connus, le carré constitue la forme la plus simple à identifier dans une image en localisant ses quatre sommets.

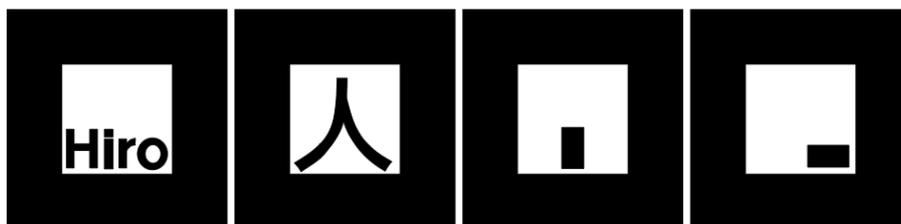


FIGURE 1.16 – Exemples de cibles codées utilisées par ARToolKit.[[Didier et al., 2008](#)]

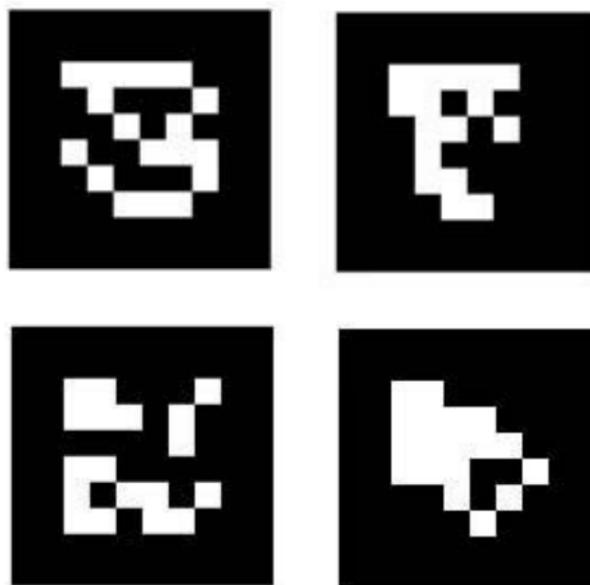


FIGURE 1.17 – Exemples de cibles codées utilisées par ARTag.[Didier et al., 2008]

Afin d’estimer la pose de la caméra en se basant des marqueurs, plusieurs étapes sont nécessaires :

- une phase d’acquisition de l’image de l’environnement réel.
- une phase de prétraitement.
- une phase de détection des potentiels cibles.
- une phase d’identification et de décodage des cibles codées.
- et enfin une phase d’estimation de la pose de la caméra.

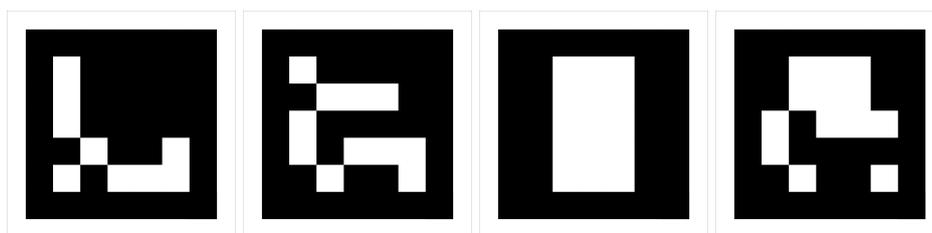


FIGURE 1.18 – Exemples de cibles codées utilisées par ArUco.[Didier et al., 2008]

La phase de prétraitement consiste à réaliser une série de traitement d’image de bas niveau. L’image acquise est convertie en une image en niveau de gris. Un seuillage est effectué afin d’obtenir un image binaire. Ce seuillage peut être adaptatif pour régler les problèmes dus à l’illumination locale. Des méthodes de détection de contours ou de coins sont ensuite appliquées. Il est à noter qu’un mauvais échantillonnage, un mauvais seuil de binarisation, un flou lié au mouvement ou du bruit peuvent entraîner de petites erreurs de localisation des sommets ou des segments dans l’image acquise. Ces petites erreurs influent énormément sur le résultat de l’estimation de la pose qui peut être finalement complètement fausse.

La phase de détection des potentielles cibles codées peut correspondre à une analyse en com-

posantes connexes. Il s’agit de tester si la zone d’intérêt est un quadrilatère ou pas en comptant le nombre de sommets connexes ou d’arêtes. C’est le cas avec l’ARToolkit qui applique cette analyse sur une image normalisée afin de détecter des quadrilatères de même que Cybercode qui identifie des zones noires et repère ainsi la barre noire située sous le marqueur comme le montre la fig. 1.19. D’autres méthodes peuvent être utilisées afin de supprimer des zones sans intérêt. Par exemple, le critère de la taille du potentiel marqueur peut être considéré. En effet, si celui-ci est trop petit, il est éliminé puisque cela signifierait qu’il est beaucoup trop éloigné de la caméra et que s’il est retenu, la pose calculée peut être incertaine et donc peu fiable. De plus, son identification serait difficile. Le nombre de pixels de la régions d’intérêt peut supprimer les régions trop petite ou trop grande. L’histogramme du potentiel marqueur ou son nombre de "trous" donc de zones blanches ou également le nombre de changement d’intensité dans deux directions orthogonales peut permettre d’éliminer certaines zones de l’image puisque ces caractéristiques sont spécifiques pour chaque marqueur. Cette étape de tri et d’élimination des mauvais marqueurs se réalise rapidement.

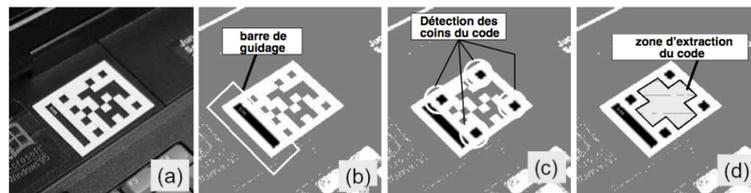


FIGURE 1.19 – Étapes d’identification des cibles codées à l’aide de Cybercode.[Didier et al., 2008]

La phase d’identification ou de décodage des marqueurs peut consister en une mise en correspondance des *templates* comme avec ARToolkit ou à identifier le code associé au contenu du marqueur extrait comme avec Cybercode, ARTag ou ArUco. Pour le cas d’une mise en correspondance d’un *template*, une homographie peut être réalisée entre le plan de la cible et le plan image. Ensuite, un coefficient de corrélation est calculé entre l’image normalisée et une banque d’images afin de reconnaître la cible. Cependant, cette approche peut être coûteuse en terme de temps si la base d’images est énorme. De plus, elle peut reconnaître des *templates* similaires alors qu’ils sont visuellement très différents. Pour l’approche consistant à décoder le contenu du marqueur, il s’agit principalement de récupérer le contenu et d’identifier le code binaire associé. Par exemple, avec Cybercode ou avec ARTag, les paramètres liés à la distorsion de la caméra et/ou l’inclinaison du marqueur dans l’image en tenant compte des quatre sommets identifiés peuvent être estimés afin d’extraire le code binaire.

Contrairement aux bibliothèques citées, InterSense utilise des cibles codées de forme circulaire (cf. fig.1.20). La procédure pour l’estimation de la pose reste la même que pour les cibles de forme carrée. Cependant, pour la phase de prétraitement, un rehaussement de contraste est d’abord effectué, suivi d’une détection d’arêtes à l’aide un filtre de Sobel. Afin de conserver les

bons marqueurs, une binarisation et une érosion permettent de filtrer les zones blanches à l’aide de forme spécifique. Le code est ensuite identifié par un ensemble de trois points, un blanc et deux noirs.



FIGURE 1.20 – Exemples de cibles codées utilisées par InterSense.[Didier et al., 2008]

Les cibles identifiées sont, ensuite, mises en correspondance avec le modèle 3D stocké dans la base de connaissance. Celles-ci permettent, ainsi, d’estimer la pose de la caméra. Le suivi des points est effectué le long du flux vidéo reçu par le système. Le fait de connaître, au préalable, la position des cibles codées dans l’environnement permet un calcul fiable et précis du point de vue pour un alignement correct des objets virtuels dans l’environnement réel. Cependant, le fait de disposer des cibles codées dans l’environnement réel peut être assez contraignant en intérieur et inadéquat en extérieur. De plus, une occultation de ces marqueurs peut facilement entraîner une perte de suivi. Par ailleurs, il est à signaler que les marqueurs vus de face sont susceptibles de fournir un pose plus instable et incertaine que ceux apparaissant de manière oblique sur l’image. En effet, les grands déplacements suivant l’axe de profondeur, généralement identifié par l’axe Z peut correspondre à de faibles déplacements suivant l’axe des x et y de l’image. De faibles erreurs dans la détection des marqueurs dans l’image peut influencer la pose suivant l’axe Z . De plus, lors d’une séquence vidéo, l’estimation de la pose est combinée à une méthode de suivi comme le filtre de Kalman ou de Kalman étendu afin de stabiliser la pose obtenue qui a tendance à osciller au cours de temps. Un autre inconvénient des marqueurs est lié au champ de vision de la caméra nécessaire pour les localiser. Ce champ de vision est souvent étroit, ce qui peut entraîner une perte de vue des marqueurs par l’utilisateur lorsque celui-ci est en mouvement. Afin de pallier ce problème d’occultations et de suivi, des bibliothèques comme ARToolkit, ARTag ou ArUco propose d’utiliser plusieurs marqueurs qui constituent finalement un marqueur un peu plus grand. Cette approche est beaucoup plus robuste et adéquate.

ii) Suivi sans cibles codées

Lorsque l’environnement est complexe, comme c’est le cas en extérieur, l’utilisation des techniques de suivi à l’aide de marqueurs impliquerait de disposer d’un très grand nombre de cibles codées dans l’environnement du fait d’un champ de vision beaucoup plus large acquis par les caméras. Pour y remédier, des méthodes de suivi sans cibles codées ont été proposées.

Ces méthodes se basent sur l’extraction dans la scène réelle de primitives dites ”naturelles”.

Celles-ci peuvent correspondre à des contours, des segments, des surfaces planaires, des points d’intérêt ou des imagerie. Dans un environnement bien texturé, les points d’intérêt sont souvent les plus populaires et sont associés à des descripteurs. En revanche, les segments sont les plus usités dans le cas où l’environnement est moins marqué par des textures et/ou lorsqu’un modèle 3D comme le modèle CAO (conception assistée par ordinateur) est disponible comme c’est souvent le cas dans l’industrie. Les surfaces planaires sont moins populaires mais ont déjà été utilisées dans le cadre d’une application de RA pour l’architecture [Santos et al., 2007]. Les façades des bâtiments qui correspondaient aux surfaces planaires étaient extraites afin de réaliser l’estimation de la pose et le suivi de la caméra. Le choix des primitives à utiliser dans le système de RA dépend de l’application, des objets à identifier et de l’espace de travail. Il est admis que les bonnes primitives sont celles qui sont considérées comme bien définies dans l’image, dont les régions locales sont suffisamment riches pour en extraire des données pertinentes, notamment des descripteurs invariants aux transformations affines, aux rotations, à l’échelle et à l’illumination globale et locale.

Les descripteurs tels que SIFT (*Scale-Invariant Feature Transform*) [Lowe, 1999], SURF (*Speeded Up Robust Features*) [Bay et al., 2008], BRIEF (*Binary Robust Independent Elementary Features*) [Calonder et al., 2010], ORB (*Oriented FAST and Rotated BRIEF*) [Rublee et al., 2011], BRISK (*Binary Robust Invariant Scalable Keypoints*) [Leutenegger et al., 2011], ou FREAK (*Fast Retina Keypoint*) [Ortiz, 2012] sont souvent cités dans des applications de RA. Bien qu’invariant à l’échelle, aux transformations affines, à la rotation ou à la luminosité, le descripteur SIFT est très coûteux aussi bien lors de la détection des points d’intérêt que lors de la description et à la mise en correspondance. Des alternatives ont été proposées afin de le rendre plus rapide. [Grabner et al., 2006] ont proposé le Fast approximated SIFT qui consiste à approximer les différences de gaussiennes et les histogrammes d’orientations lors de la détection et de la description. Cela leur permettait de réduire le temps de calcul lors de ces étapes. En revanche, [Ke and Sukthankar, 2004] ont proposé le PCA-SIFT qui utilisait la décomposition en composantes principales (PCA) afin de réduire la taille du descripteur SIFT. Celui passait ainsi de 128 éléments à 36. Cela permettait de réduire le temps de calcul de la phase de mise en correspondance. Cependant, malgré ces améliorations qui ont été apportées, ce type de descripteur reste encore trop coûteux pour des applications de RA en temps réel sur mobile.

Le descripteur SURF a été introduit afin de pallier ce défaut du SIFT. Il est plus rapide et est très fréquemment utilisé en RA. En revanche, il n’est invariant qu’à l’échelle et à la rotation. De plus, même s’il est trois fois plus rapide que le SIFT [Bay et al., 2008], il est encore trop complexe pour le mobile dont la capacité de calcul est limitée.

Afin de s’adapter aux caractéristiques des appareils mobiles, des descripteurs comme BRIEF, BRISK ou ORB sont introduits. Ce sont des descripteurs qui permettent de réduire le temps de

calcul et la taille de stockage nécessaire en réduisant leur dimension et en utilisant un système binaire au lieu de celui des nombres rationnels. La phase de mise en correspondance est également plus rapide en calculant la distance de Hamming.

Lorsque des modèles sont disponibles, la structure générale des objets est extraite des images à l'aide d'une combinaison de lignes, de cercles, de cylindres ou de cercles. La pose de la caméra peut être estimée à l'aide des descripteurs lorsque l'environnement est texturé. Lorsqu'aucun modèle n'est disponible ou lorsqu'il est partiel, des techniques permettant de recréer l'espace de travail tel le SLAM (*Simultaneous Localization And Mapping*) [Durrant-Whyte and Bailey, 2006], le PTAM (*Parallel Tracking and Mapping*) [Klein and Murray, 2007] qui est fiable dans un espace étroit ou utilisant la notion d'image-clé [Vacchetti et al., 2004] peuvent permettre d'apprendre l'environnement et de créer ou de suivre un modèle.

Plusieurs approches peuvent être choisies afin d'estimer la pose. Après une phase d'initialisation estimant la pose de la caméra, une méthode de suivi des primitives dans une séquence vidéo permettrait de retrouver cette pose. Par exemple, [Vacchetti et al., 2004] utilisent, dans le cadre d'une application de guidage et de conception dans un environnement industriel, une méthode consistant d'abord à une mise en correspondance 2D des points d'intérêt, puis en déduisent la position 3D des points sur la surface de l'objet. Ainsi, leur méthode détecte et extrait, dans chaque frame, des points d'intérêt de Harris. A l'instant t , ils supposent que la position 2D et 3D des points d'intérêts à l'instant $t-1$, est connue, ainsi que les paramètres de translation et de rotation. Une méthode de rétro-projection leur permet de réaliser des correspondances entre les points à l'instant t et ceux à l'instant $t-1$. Ainsi sont réalisées les correspondances 2D/3D. L'algorithme de [Dementhon and Davis, 1995] leur permet de déterminer la position de la caméra et l'algorithme du RANSAC [Fischler and Bolles, 1981] d'éliminer les mauvaises correspondances. Une seconde approche consiste à effectuer une détection, une mise en correspondance et une estimation de la pose à chaque instant. La troisième consiste à allier un suivi et une détection locale proche de la région d'intérêt à l'instant précédent. Les méthodes de suivi sont souvent privilégiées par rapport à des méthodes de détection et de mise en correspondance du fait de leur rapidité. Cependant, celles-ci supposent souvent que les déplacements entre deux images sont petits. Ainsi, si le mouvement est trop rapide donc le déplacement est large, ce suivi devient peu fiable.

c) Suivi à l'aide de capteurs de profondeur

Avec la sortie de capteurs de profondeur tels les Kinects de Microsoft, il est possible de récupérer directement les données de profondeur de la scène. Deux méthodes peuvent être configurées sur ces dispositifs :

- soit un système à lumière structurée. Il consiste à projeter sur la scène des motifs lumi-

neux qui peuvent correspondre soit à des franges d’interférence, des grilles, des lignes ou des ellipses à l’aide d’un projecteur infrarouge. Une caméra à infrarouge légèrement décalée par rapport au projecteur *ie* d’un autre point de vue récupère l’image obtenue. Les déformations enregistrées vont être utilisées pour le calcul des distances de profondeur à l’aide d’une méthode de triangulation. C’est le même système utilisé sur la 1ère version de la Kinect de Microsoft 1.21a. Même si ces systèmes permettent de déterminer la profondeur réelle des objets, ils ne peuvent pas, comme, pour les caméras à infrarouge, fonctionner en présence d’une source lumineuse plus puissante telle que le soleil. En effet, le signal devient, dans ce cas, indiscernable. De plus, si les objets visés sont très éloignés de la source lumineuse, il sera difficile pour le système de calculer la profondeur du fait de l’atténuation du signal avec la distance.

- soit un système à temps de vol. Il consiste à envoyer un signal lumineux (laser ou infrarouge) et à mesurer son temps de parcours jusqu’à ce que le rayon réfléchi soit récupéré par le capteur. Cette donnée permet de déterminer la profondeur du point de réflexion à l’aide de la valeur connue de la vitesse lumineuse. Ce système a été adopté sur la Kinect v2 de Microsoft 1.21b. Cette technique génère ainsi un nuage avec un grand nombre de points. Cela peut nécessiter un large espace de stockage sur les matériels utilisés et donc de tirer parti des capacités de calcul du CPU ou du GPU. Par ailleurs, des limitations physiques liées à la taille du capteur peuvent engendrer des images de profondeur de faible résolution, à une faible portée ou à un bruit de mesure important. De plus, si des capteurs à infrarouge sont adoptés, les mêmes limitations présentées précédemment apparaissent.



FIGURE 1.21 – Kinects de Microsoft.⁶

d) Algorithmes PnP

Les parties a), b), c) consistaient à montrer les techniques utilisées pour l’extraction des primitives sur l’image 2D. Après leur mise en correspondance avec les primitives issues de la base de connaissance, des algorithmes sont appliqués afin d’estimer la pose d’une caméra calibrée. Ils permettent de résoudre des problèmes PnP (*Perspective-n-Point*) *ie* d’estimer l’orientation et

6. <https://www.microsoft.com/en-us/devices>

la position de la caméra à partir d’un ensemble n de correspondances 2D/3D. Ils peuvent être classés en deux catégories [Didier et al., 2008] :

- les techniques analytiques.
- et les techniques d’optimisation.

Les méthodes analytiques telles que celles de [Quan and Lan, 1999, Dhome et al., 1989] sont basées sur une analyse mathématique des relations géométriques existant entre les points 3D et leur projeté image. Ils fournissent ainsi un système d’équations issues des mises en correspondance 2D/3D d’un nombre réduit de primitives. Elles fournissent ainsi également un nombre réduit de solutions. Par exemple, pour une recherche de pose avec un ensemble de 3 primitives, quatre solutions sont théoriquement possibles, tandis qu’avec un ensemble de 4 primitives, une solution unique en ressort tant que les 4 points ne sont pas alignés. La complexité de ces méthodes est faible et le temps de calcul rapide. En revanche, elles sont très dépendantes de la qualité de la mise en correspondance des primitives 2D/3D. Si cette dernière est correcte et certaine, la pose obtenue sera précise. Cependant si elle est incorrecte, la pose sera peu fiable et engendrera un mauvais alignement des objets virtuels dans l’environnement réel.

Les méthodes d’optimisation peuvent permettre d’affiner le calcul de la pose de la caméra en présence d’un grand nombre de primitives mises en correspondance. Elles consistent, en général, à minimiser un critère d’erreur. Elles sont ainsi formulées comme un problème non linéaire de moindres carrés d’un système d’équations polynomiales. Des solutions itératives [Dementhon and Davis, 1995, Oberkampf et al., 1996, Lu et al., 2000, Kumar and Hanson, 1994] sont souvent proposées afin de résoudre ce type de problème. Du fait de ces itérations, elles deviennent coûteuse en temps mais fournissent une solution précise lorsqu’elles convergent. Cette convergence peut dépendre de l’initialisation effectuée pour démarrer le processus d’optimisation. Elles peuvent gérer un nombre arbitraire de correspondant permettant ainsi de ne pas dépendre fortement à la qualité de ces correspondances tels que c’est le cas avec les méthodes analytiques. D’autres solutions non itératives comme celle de [Lepetit et al., 2009, Ferraz et al., 2014] peuvent permettre de déterminer la pose de la caméra avec une complexité linéaire en préservant la précision du résultat.

Des cas dégénérés peuvent exister dans lesquels même un nombre infini de points ne fournisse pas une solution unique [Lepetit and Fua, 2005, Ameller et al., 2002, Wrobel, 2001]. C’est le cas :

- lorsque le nuage de points dans l’espace 3D forme un cube twisté *ie* un horoptère avec la caméra incluse dans cette région. Dans cette situation, le système d’estimation de pose peut fournir une infinité de solutions.
- lorsque les points sont à l’infini.
- lorsque le nuage de points forme un cercle et une droite orthogonale au plan du cercle.

En revanche, comme le rappellent [Lepetit and Fua, 2005], même si la caméra et le nuage de

points sont loin de ces cas limites, il se peut que les solutions obtenues peuvent ne pas être fiables du fait du processus général d'estimation de la pose.

Ainsi, le choix de l'algorithme d'estimation de la pose doit prendre en compte le matériel utilisé, les primitives extraites, la précision et la rapidité souhaitée pour l'application de réalité augmentée. Par ailleurs, afin d'augmenter la précision de la pose et de pallier la défaillance de certains capteurs, une fusion des données est souvent adoptée.

1.2.3 Méthodes hybrides d'estimation de la pose ou fusion de capteurs

L'utilisation d'une seule méthode et d'un seul capteur pour le suivi et l'estimation de la pose n'est pas toujours adéquate. Combiner diverses méthodes de suivi et la provenance des données permettrait de tirer profit des points forts de chaque capteur et de compenser leurs points négatifs afin d'obtenir des augmentations correctes et une cohérence spatiale entre les éléments virtuels et les éléments réels.

En général, il s'agit d'une combinaison de techniques de suivi basées capteur et de techniques basées vision par ordinateur qui permettent un suivi robuste. Un suivi à l'aide d'un capteur magnétique ou inertiel et d'une caméra tire avantage de la rapidité de mise à jour des données de position et de rotation des capteurs magnétiques ou inertiels et de la stabilité et de la faible dérive de la caméra.

Il est également possible de combiner un GPS, des capteurs inertiels et/ou une caméra comme [Azuma et al., 2006]. Ils utilisent une méthode hybride incluant le GPS afin d'améliorer le suivi de la position et le recalage dans le cadre d'opérations militaires. Les casques utilisés dans l'expérience comportent 3 gyroscopes orthogonaux, 2 capteurs d'inclinaison, un GPS et une caméra infrarouge pour des expérimentations dans des conditions nocturnes. Le GPS permet, ainsi, de déterminer la position des éléments en mouvement, notamment le soldat et le matériel mobile. Les gyroscopes et les capteurs d'inclinaison calculent l'orientation du point de vue. La caméra infrarouge détecte des balises infrarouges installées un peu partout dans l'environnement extérieur afin de corriger les erreurs relevées par les différents capteurs. Ce type de fusion permet ainsi de retrouver les six degrés de liberté nécessaire à la pose de la caméra (position et orientation). De plus, il permet de pallier à une potentielle perte de signal due à une occultation ou au fait, que l'utilisateur se retrouve à l'intérieur d'un bâtiment. De même, [Zendjebil, 2010] propose également de fusionner les données issus de la caméra, du GPS et d'un capteur inertiel pour le projet RAXENV (Réalité Augmentée en eXtérieur appliquée aux sciences de l'ENVironnement). L'objectif était de prévenir les défaillances dans l'estimation de la pose basée vision en reléguant le calcul au couple GPS/capteur inertiel. Comme l'application devait fonctionner dans un environnement extérieur, les primitives extraites pouvaient être influencées par les changements de luminosité et/ou de contraste ou par les occultations imprévisibles pouvant s'y produire. Les capteurs inertiels et le GPS semblaient ainsi être les plus

adéquats pour prendre le relai dans la détermination la position et l’orientation de la caméra dans un environnement extérieur.

Les appareils mobiles, avec tous les capteurs qui y sont intégrés tels la caméra, l’accéléromètre, le gyroscope, le GPS ou le wifi, sont des outils qui favorisent l’utilisation de méthodes de suivi et d’estimation de pose hybride. Par exemple, [J. Newman, 2006] développent un système de réalité augmentée mobile exploitant les caractéristiques d’un certain nombre de capteurs. Ils utilisent l’UWB (*Ultra-Wide Band*) pour déterminer la position en intérieur et des accéléromètres pour l’orientation combinés aux cibles codées.

Le choix d’une méthode de suivi dépend énormément de l’environnement dans lequel le système de réalité augmentée sera introduit ainsi que du système lui-même. En effet, l’environnement peut être intérieur ou extérieur ou encore une combinaison des deux et le système peut être mobile ou statique. Il s’agit de déterminer quelle fusion de capteurs serait la plus adéquate et qui permettrait de déterminer la pose la plus fiable suivant l’application de RA.

1.3 Bilan

Ce chapitre consistait à faire un tour d’horizon général de la réalité augmentée. Nous sommes revenu sur le concept de la réalité augmentée et sa taxonomie fonctionnelle. Nous y avons proposé une taxonomie relative à l’interaction et l’interactivité entre l’utilisateur, les objets réels et les entités virtuelles. Nous avons également présenté les différents dispositifs de visualisation pour la RA existants. Nous y abordons les avantages et les limitations de chacun des types de dispositifs. Bien que leur port, notamment des casques, reste encore très peu populaires, des travaux sur leur ergonomie et leur performances permettront certainement un boom de leur vente et celui des applications de RA dans divers domaines tel qu’a connu Pokemon Go. On relève, tout de même, que les dispositifs de visualisation de poche sont un matériel qui répond, parmi les mieux, à plusieurs caractéristiques pouvant améliorer l’utilisation des applications de RA par une majorité de monde à court terme. Il s’agit du dispositif de visualisation que nous avons privilégié malgré la contrainte de la prise en main de ce matériel.

De plus, nous ne pouvions aborder la RA globalement sans mentionner les méthodes pour le suivi et l’estimation de la pose qui permettent de maintenir une cohérence spatiale et temporelle entre le monde réel et le monde virtuel. Le choix de l’algorithme de suivi dépend énormément de l’environnement dans lequel le système de RA doit s’appliquer ainsi que des moyens matériels disponibles, de la précision et de la vitesse d’exécution souhaitée. La tendance, ces dernières années, est de ne plus se contenter et se fier à un seul capteur mais d’établir des méthodes hybrides afin de tirer avantage de tous les capteurs en jeu et les relayer en cas de défaillance d’un d’entre eux. Dans nos expérimentations, nous avons le choix des capteurs tels l’accéléromètre,

la centrale inertielle ou la caméra. Nous avons surtout opté pour des méthodes de suivi basée sur la caméra car, comme notre dispositif est destiné au monde industriel, nous supposons que des éléments ferreux pouvant être présent dans l'environnement de travail de l'utilisateur peuvent fausser les mesures avec les autres capteurs.

Le chapitre suivant permettra d'illustrer le potentiel de la RA mobile dans des domaines variés et, notamment dans l'industrie. Nous y aborderons les points faibles et forts de ces systèmes. Nous évoquerons également les approches proposées dans la littérature afin de suivre et contrôler les actions effectuées par l'utilisateur.

Chapitre 2

Réalité augmentée mobile et industrielle

Dans ce chapitre, nous présenterons des applications de réalité augmentée mobile dans des domaines aussi variées que le tourisme, les jeux vidéos ou le marketing et plus particulièrement dans l'industrie ainsi que leur limitations. La réalité augmentée mobile (RAM) pourrait être définie comme des systèmes de RA permettant d'assurer la mobilité de l'utilisateur [Höllerer and Feiner, 2004]. Les allées et venues de l'utilisateur ne sont pas limitées à un endroit précis équipé des dispositifs de RA. Cette définition n'implique pas uniquement des dispositifs portatifs comme des smartphones, des tablettes ou des visiocasques mais également des réseaux des dispositifs fixés dans l'environnement préservant la mobilité de(s) utilisateur(s). Dans le cadre de nos travaux, nous nous restreignons à considérer la réalité augmentée mobile par rapport aux dispositifs mobiles [Arth et al., 2015] ie aux systèmes portables par l'utilisateur.

2.1 Analyse de caractéristiques de quelques systèmes de RA mobile

Nous avons, dans un premier temps, réparti dans un tableau 2.1 un certain nombre d'applications que nous citerons dans ce chapitre. Nous y avons relevé les caractéristiques telles que les dispositifs de visualisation, les techniques de suivi et d'estimation de la pose, les domaines d'applications et s'il est possible d'en déduire un intérêt pour le domaine industriel. Nous y avons également noté les bibliothèques utilisées pour l'implémentation ainsi que les points forts et les points faibles. Ces caractéristiques nous permettent de décrire ces applications et d'identifier les groupements possibles. Ce tableau permettra ainsi d'identifier les limitations liées à l'utilisation des dispositifs mobiles dans le cas général et plus spécifiquement dans le domaine industriel.

Système RA (Nom et référence)	Technologie de visualisation	Techniques de suivi	Industrie (Oui/Non)	Domaine d'applications	Bibliothèques	Points forts	Points faibles
ClonAR [Csongei et al., 2012]	VVI Canon VH2007	Marqueurs	Oui, dans la concep- tion des produits	Domaine artistique, Chirurgie plastique, Architecture inté- rieure, Entraînement chirurgical	- Kinect Fusion pour la reconstruction de la scène réelle - MR Platform - Phantom Omni	- Déformation en temps réel des structures réelles recons- truites - Rendu rapide des surfaces - Raymarcher permettant de prédire les effets de lumière	- Détails des captures - Résolution « faible »
BurnAR [Weir et al., 2012]	VVI Canon VH- 2007		Non	Psychologie	MR Platform	Sensation de chaleur	
Nestor [Hagbi et al., 2011]	- DP Nokia N95 - IV Dell Latitude D630		Possible	Publicité, marketing		- Reconnaissance et aligne- ment corrects Maintenance d'un suivi effi- cace	- Reconnaissance uniquement des formes non convexes - Précision des contours ex- traits → possibilité d'erreurs ? - Pas de gestion de la recon- naissance des formes en cas d'occultation partielle
PhoneGuide [Bruns et al., 2007]		Communication wi-fi	non	Tourisme, musée		- Reconnaissance d'objets (à l'aide du réseau de neurones) - Localisation - Utilisation capteur intégré à la base au mobile	Précision du suivi et de la loca- lisation (Bluetooth → bcp d'er- reurs (absorption et réflexion d'ondes), de nombreux émet- teurs → incertitude)
Time Machine [Dan et al., 2013]	DP Apple iPad mini tablette iOS 6.1.3			Tourisme	Game Engine Unity 3D 4.0 - Vuforia SDK pour la reconnaissance des points d'intérêt (POI)	cloud database	- Images panoramiques classi- fiées manuellement par leur po- sition GPS - Nécessité d'une large base d'images mosaïques d'un lieu précis

Système RA (Nom et référence)	Technologie de visualisation	Techniques de suivi	Industrie (Oui/Non)	Domaine d'applications	Bibliothèques	Points forts	Points faibles
CityViewAR [Lee and Billinghurst, 2012]	DP tablette + téléphone Android 2.2	-GPS, accéléromètres, magnétomètre - GPS pour la position géographique du device - boussole électronique et accéléromètre pour l'orientation de la vue du device	non	Tourisme, design, BTP	- Google Map API Service - Google Sketch-up pour la modélisation des bâtiments - Info POI stockées à l'aide SQLite database - Android SDK - OpenGL ES API		GPS erreurs de localisation : utilisation de points précis sur une carte. L'utilisateur est prié de les visiter pour éviter des dérives
CAViAR [Delail et al., 2012]	iPhone	- Cartes du campus superposé au Google Map - Initialisation à l'aide d'un marqueur - Navigation inertielle pour la mise à jour de la localisation	oui	- navigation - Localisation	- Qualcomm's Vuforia SDK - OpenGL ES 2.0	- Localisation précise d'un bureau à l'aide de technique de vision par ordinateur pour la reconnaissance (PCA) - Bonne gestion des occultations à l'aide de Vuforia SDK	Objets planaires pour la détection statique
ARCHEOGUIDE [Gleue and Dähne, 2001]	Casques HMD - Sony Glasstrom - Glasstrom-based n Vision VB-30	- GPS différentiel Garmin GPS 35 LVS - Boussole PNI TCM2-50	non	-Tourisme - Héritage culturel - Archéologie		Localisation correcte en extérieur	- Dispositifs encombrants et lourds - Interaction pas naturelle
MARS [Höllner et al., 1999]	HWD Sony LDI 100B / LDI-D 100B	- GPS - Magnétomètre		Navigation	Coterie	Localisation correcte en extérieur	Dispositifs encombrants
Touring Machine [Feiner et al., 1997]	Optical see-through display Virtual I/O i-glasses	- Trimble DSM GPS - Magnétomètre		Navigation	Coterie	Localisation correcte en extérieur	- Faible contraste et faible résolution du HMD - Suivi pas assez robuste
[Veas et al., 2012]	Panasonic CFU1 tablet PC	- Centrale inertielle - GPS	oui	- Environnement - Géologie	- KAOS framework - OpenGL	- Meilleure compréhension des paramètres environnementaux à travers une visualisation plus adaptée	

Système RA (Nom et référence)	Technologie de visualisation	Techniques de suivi	Industrie (Oui/Non)	Domaine d'applications	Bibliothèques	Points forts	Points faibles
RAXENV [Zendjebil, 2010, Ababsa et al., 2012]	Tablette PC	- Centrale inertielle Xsens MTi - Trimble GPS Pathfinder ProXT - Caméra USB uEye UI-2220RE	oui	Géologie	ARCS	Localisation et recalage corrects en milieu extérieur	
Smart Vidente [Schall et al., 2013]	Tablette PC (Motion J3400)	- GPS Novatel OEMV-2 L1/L2 Real-Time Kinematic - Magnétomètre - Caméra	oui	- BTP - Génie civil		- Bonne mesure de l'orientation de l'appareil - Visualisation de structures cachées	- Perception de la profondeur
Fata Morgana [Klinker et al., 2002]	Optical see-through HMD Sony Glasstron	- Caméra - Cibles codées (LED)	oui	Conception	- ARToolKit - Système DWARF	- Visualisation du modèle en taille réelle	- Limitations liées au HMD (résolution, champ de vision) - Manque de réalisme pour les designers - Qualité du rendu - Temps de latence
COGNITO [Gorecky et al., 2011]	Optical see-through HMD	- Caméra - Centrale inertielle	oui	- Assemblage - Maintenance		- Aide à l'opérateur pour des tâches complexes	- Pas de données concernant le suivi et la précision
AMRA [Didier, 2005]	- Tablette - Smartphone - Optical see-through HMD	- caméra	oui	- Assemblage - Maintenance industrielle	ARCS	- Aide contextualisée à l'opérateur pour des tâches complexes - Portabilité, lisibilité et réutilisabilité du code simple et rapide à l'aide ARCS	
TAC [Bottecchia, 2010]	- Optical see-through HMD Video see-through HMD	- centrale inertielle	oui	- Maintenance industrielle		- Télé-assistance d'un opérateur par un expert distant	Symboles de communication pas toujours communes entre l'opérateur et l'expert distant
MARTS [Ghouaïel, 2014]	- Smartphone	- caméra	non	- Tourisme - Apprentissage historique		- Coprésence d'un guide virtuel captant l'attention de l'utilisateur	Gestion des occultations

TABLE 2.1 – Quelques systèmes de RA mobile.

2.2 RAM : applications et limitations

2.2.1 Applications de réalité augmentée mobile

Avec la popularisation et l'accélération des calculs des appareils mobiles tels que les smartphones ou les tablettes et dans une volonté de démocratiser la réalité augmentée auprès du public, de nombreuses entreprises et laboratoires tels que le laboratoire IBISC avec le projet AMRA (Assistance à la Maintenance en Réalité Augmentée) [Didier et al., 2005] pour la maintenance ou RAXENV [Zendjebil, 2010] pour les sciences de la terre ont montré son utilité à travers diverses applications. En 2014, IKEA, à travers son catalogue, démontrait au grand public comment la réalité augmentée permettait de visualiser et de disposer des meubles dans une pièce. De même, les opticiens ATOL proposaient une application pour observer l'effet des choix de lunettes sur l'utilisateur et l'Oréal une application pour réaliser un maquillage virtuel. D'autres domaines tels que le tourisme, la décoration intérieure, le marketing, les jeux vidéo... y ont vu également un intérêt immédiat afin d'attirer de nouveaux visiteurs, clients ou utilisateurs.

Par exemple, on note une forte demande et une utilité flagrante de la réalité augmentée dans l'industrie du tourisme. Les visiteurs d'un lieu historique ou autres peuvent, à travers ces applications, obtenir des informations sur des événements passés et marquants qui s'y sont déroulés ou encore avoir une vue de l'architecture antérieure du lieu visité. Ces applications permettent ainsi de donner une seconde vie à ces derniers.

Un des premiers systèmes de réalité augmentée dédié à l'héritage culturel est le projet ARCHEOGUIDE (Augmented Reality-based Cultural HERitage On-site GUIDE) [Gleue and Dähne, 2001]. Il fournissait aux visiteurs non seulement des informations sur des sites archéologiques tels que l'ancienne Olympia en Grèce mais présentait également une reconstruction des bâtiments tels qu'ils étaient à l'époque de leur construction. CityViewAR [Lee and Billingham, 2012] offre les mêmes fonctionnalités. Il permet de visualiser les bâtiments et d'obtenir des informations sur les monuments historiques de la ville de Christchurch en Nouvelle-Zélande qui a été dévastée en 2010 et en 2011 par de nombreux tremblements de terre. Avec Time Machine [Dan et al., 2013], par contre, une visualisation panoramique des lieux historiques à des époques différentes et des moments différents de la journée est possible.

Cependant, la réalité augmentée dans le tourisme ne concerne pas que la découverte de l'histoire des monuments ou des lieux précis. Elle s'applique également dans l'apprentissage de l'histoire à travers la mise à disposition au visiteur d'informations relatives à des objets historiques. Par exemple, [Bruns et al., 2007] proposait un système de réalité augmentée mobile pour les musées nommé PhoneGuide. Celui-ci devait afficher des informations concernant les objets exposés dans le musée après sa reconnaissance par le mobile. Il permettait également de localiser les objets à travers le bâtiment. De plus, dans la même lancée concernant l'apprentissage de l'histoire à travers des systèmes de réalité augmentée, [Liestol and Morrison, 2013] ont étudié

l'incidence de différentes vues possibles (*ie* vue normale, zoom, vue d'oiseau, vue cartographiée, vue « souterraine », vue « mouvante », vue libre) lors de l'acquisition d'informations par l'utilisateur. Ce système a été appliqué sur les modèles reconstruits du Forum Augustus et du Forum Républicain à Rome. Ces derniers étaient superposés, lors de la visualisation, aux restes de ces monuments (cf. fig.2.1).



FIGURE 2.1 – Tourisme et patrimoine culturel. [Liestol and Morrison, 2013]

Par ailleurs, le tourisme ne concerne pas que la découverte de l'histoire d'un lieu ou d'objets regroupés dans un musée, il implique également l'exploration de paysages particuliers par les touristes.

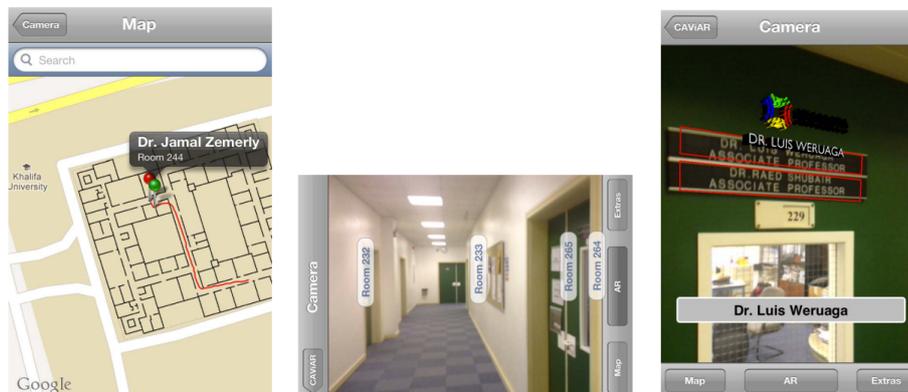


FIGURE 2.2 – Navigation dans un campus d'université à l'aide de CAViAR. [Delail et al., 2012]

Lorsqu'on vit dans une grande ville ou lorsqu'on travaille dans de grands buildings, il arrive souvent que l'on se perde. Les applications de navigation et de guidage dans des environnements intérieurs ou en extérieur ont été largement développées dans la littérature scientifique. Ainsi, on peut citer le projet MARS (Mobile Augmented Reality System) [Höllerer et al., 1999] qui permettait de se guider dans un environnement extérieur et laissait la possibilité à l'utilisateur de l'annoter. Celui-ci était précédé de l'application Touring Machine [Feiner et al., 1997] développé par la même équipe de l'université de Columbia. Celle-ci présentait des informations sur les bâtiments en tenant compte de la localisation de l'utilisateur. Dans la même optique d'un navigation plutôt en intérieur, CAViAR proposé par [Delail et al., 2012] permettait à un visiteur

d'un campus d'une université de se localiser dans les bâtiments de cette dernière et de retrouver l'emplacement d'un bureau précis (cf. fig.2.2).

D'autres applications de réalité augmentée pour guider les nouveaux étudiants, ne connaissant pas encore l'université, ont été mises en place dans de nombreuses universités américaines telles que Purdue University, les universités de Californie, de Santa Barbara, du Wisconsin, et bien d'autres. Ce type d'applications comme le souligne [Kurkovsky et al., 2012] doit pouvoir acquérir la position et l'orientation de l'utilisateur, identifier les bâtiments apparaissant dans le champ de vision, les points d'entrée de ces bâtiments ainsi que les départements ou les facultés et les bureaux qu'ils y abritent.

Cependant, les applications de RA pour la navigation n'impliquent pas uniquement le piéton. Elles peuvent également concerner les automobilistes. Ainsi des applications dédiées à l'assistance à la conduite automobile ont vu le jour. Par exemple, à l'aide de WikitudeDrive [Carmigniani and Furht, 2011] qui fonctionne comme un GPS, les utilisateurs peuvent avoir une vue sur la route ainsi qu'être informés en temps réel sur la direction à prendre (cf. fig.2.3).



FIGURE 2.3 – Aide à la navigation avec WikitudeDrive . [Carmigniani and Furht, 2011]

Un autre domaine a également vu assez rapidement l'intérêt de la réalité augmentée. Il s'agit de l'industrie du jeu vidéo. Cela permet de donner une autre dimension à un scénario de jeu vidéo. Par exemple, FireFighter 360 [Carmigniani and Furht, 2011] propose au joueur de lutter, tel que le ferait un vrai pompier, contre un feu virtuel apparaissant dans l'environnement réel de manière aléatoire (cf. fig.2.4a).

D'autres jeux vidéos tels que GeoBoids favorisent l'exercice d'une activité physique. Dans GeoBoids, des créatures sont disséminées un peu partout dans l'environnement réel (cf. fig.2.4b). Celles sont visibles et audibles. Le but du jeu est d'atteindre l'emplacement d'un essaim de créatures avant une durée bien déterminée et de tous les capturer avant de passer à la prochaine étape. Dans la même optique, ARQuake [Thomas et al., 2000] qui est une transposition dans le domaine de la RA du jeu Quake, offrait la possibilité au joueur de parcourir un environnement aussi bien intérieur qu'extérieur et de tirer sur ses ennemis virtuels qui apparaissaient dans la scène. Par ailleurs, un jeu de réalité augmentée mobile a eu énormément de succès ces derniers

mois et a le plus popularisé la RA. Il s'agit de Pokémon Go (cf fig. 2.4c) développé par Niantic, Nintendo et Pokémon Company. Dans le cadre de ce jeu, les joueurs doivent capturer une série de personnages virtuels géolocalisés cachés dans l'environnement réel et peuvent interagir à travers des batailles sur des arènes géolocalisées également.



FIGURE 2.4 – Réalité augmentée dans l'industrie du jeu vidéo.

D'autres applications restent dans les interactions à travers des réseaux sociaux. Elles consistent à offrir une autre expérience à l'utilisateur à travers ce mode de communication. Par exemple, avec TweetGlue, [Higuchi et al., 2015] affichent les tweets des utilisateurs à l'endroit où ceux-ci ont été émis et permettent à d'autres utilisateurs d'y réagir lorsqu'ils se situent aux endroits alentours.

Les entreprises portent un intérêt grandissant pour des applications de réalité augmentée afin d'attirer de plus en plus de nouveaux clients. Ainsi dans le domaine du marketing ou de la publicité, ces applications permettent de mettre en avant et de promouvoir un produit.

[Hagbi et al., 2011] ont présenté Nestor, un système de réalité augmentée permettant de reconnaître des modèles 3D et de les afficher en temps-réel à partir de formes planaires (cf. fig.2.5). La technique utilisée dans cette étude permet la visualisation des entités virtuelles sans l'utilisation de marqueurs qui est souvent contraignante dans un environnement inconnu. Cette technique consistant à faire apparaître des modèles 3D à partir d'images planaires est maintenant largement utilisée dans le marketing. Elle permet au futur client de visualiser les produits sous toutes ses coutures et tous ses angles.

D'autres applications dans le marketing sont possibles. Par exemple, des cabines d'essayage virtuel présentées par la société Ayotle au Laval Virtual 2014 permettent aux clients de tester assez rapidement des collections de vêtements dans un magasin. Le client se place devant une borne ou un écran muni d'une Kinect et choisit interactivement les modèles de vêtements qu'il

1. <https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=fr>



FIGURE 2.5 – Visualisation de modèle 3D à partir de formes planaires. [Hagbi et al., 2011]

désire essayer. Cependant ce type d'applications utilise des modèles de vêtements assez figés. Or ces derniers sont des objets déformables. Ainsi le réalisme des effets des interactions avec ces objets reste encore à améliorer.

L'ingénierie ne reste pas en marge dans l'émergence de la réalité augmentée. En effet, dans le domaine des sciences de l'environnement, on pourrait citer RAXENV (Réalité Augmentée en eXtérieur appliquée à l'ENVironnement) [Zendjebil, 2010, Ababsa et al., 2012] qui étudiait la faisabilité d'un système de RA pour les sciences de la terre. Le Château de Saumur (France) est un des sites où des tests ont été effectués. Il devait permettre notamment une assistance pour des problématiques liées à la géotechnique et à la gestion des risques en cas de reconstruction d'un rempart du bâtiment. [Veas et al., 2012] ont proposé un système permettant de réaliser la surveillance des changements dans l'environnement précis (cf. fig.2.6). Deux sites avaient été choisis pour l'implémentation de leur système, Dorfberg à Davos et Kylmäoja (Finlande). Sur le site de Dorfberg qui est une zone où le risque d'avalanches est significatif, des capteurs y ont été disséminés. Ceux-ci doivent permettre de recueillir des informations concernant les conditions climatiques (températures, vitesse du vent, radiation du soleil, épaisseur de la neige) et concernant les caractéristiques du sol et de la neige. Ces différentes données sont ensuite visualisées spatialement par les chercheurs à l'aide de leur dispositif de réalité augmentée et enregistrées dans le système.

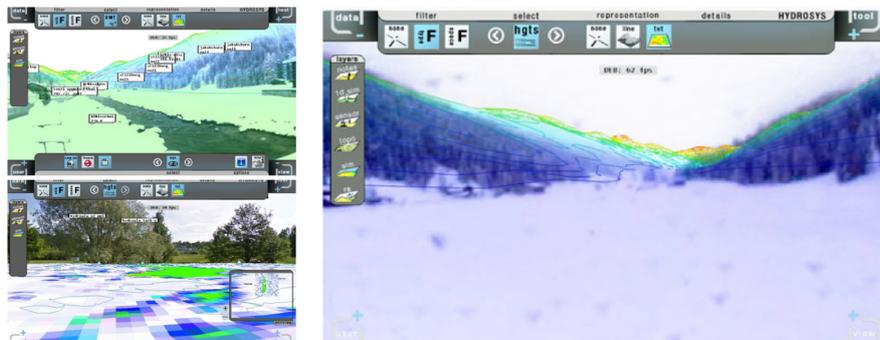


FIGURE 2.6 – Surveillance de l'environnement. [Veas et al., 2012]

Par ailleurs, pour le BTP ou l'ingénierie civile, Smart Vidente proposé par [Schall et al., 2013] permet la visualisation de structures devant se situer sous le sol (cf. fig.2.7). Ainsi les ingénieurs peuvent planifier, comparer les emplacements prévus de leurs modèles et la réalité et réaliser les corrections nécessaires avant les travaux effectifs.



FIGURE 2.7 – Smart Vidente pour l'ingénierie civile. [Schall et al., 2013]

Dans un cadre un peu formalisé, [Hartl et al., 2016] ont proposé un système de RA mobile permettant de vérifier efficacement les hologrammes présents sur des documents administratifs biométriques notamment sur des passeports ou des visas. Ils ont montré que leur système surpassait la vérification effectuée visuellement par des agents.

2.2.2 Limitations

Pour rappel, les technologies de visualisation impliquées dans la réalité augmentée mobile correspondent aux visiocasques et aux appareils mobiles. Il est possible d'y inclure les projecteurs rétiniens ou les pico-projecteurs qui préservent la mobilité de l'utilisateur sans d'autres contraintes que celles liées aux matériels eux-mêmes.

Malgré les progrès réalisés pour améliorer les capacités de calculs des appareils mobiles, ces derniers sont encore limités et ne sont pas encore comparables à celles des ordinateurs. C'est pourquoi il est nécessaire d'optimiser les algorithmes pour des applications de réalité augmentée.

Il faudrait également tenir compte de la consommation d'énergie de l'application du fait de l'utilisation des batteries par ces appareils. Toutes ces considérations sont importantes afin d'obtenir un « réalisme » graphique 3D et des interactions en temps réel.

Par ailleurs, la plupart des applications de réalité augmentée ont des contraintes liées à la taille de l'écran des appareils mobiles. Ainsi ces derniers ne permettent pas d'obtenir un champ de vision large.

De plus, dans un contexte d'environnement inconnu notamment en extérieur, il n'est pas possible matériellement de disséminer dans ce type d'environnement des cibles codées. Le suivi indoor / outdoor est encore difficilement applicable non seulement du fait du changement d'échelle d'un environnement à l'autre mais également du changement de luminosité et

de contraste, entre autres. Donc l'utilisation d'une seule méthode de suivi pour les deux environnements est encore irréalisable et inapplicable.

Comme une base de connaissance contenant les entités virtuelles est indispensable dans un système de réalité augmentée, un dispositif de communication efficace entre le point névralgique de l'application et cette base doit être correctement implanté. Le choix est souvent porté sur une communication wifi entre un serveur où la base de connaissance est stockée et le mobile qui correspond au client. Ainsi les objets virtuels sont importés et enregistré sur l'appareil mobile lorsque le système de RA en requiert l'utilisation. Lorsque leur affichage n'est plus nécessaire, ceux-ci sont alors supprimés. Ainsi ce dispositif supposerait d'avoir, tout le long de l'application, une bonne connexion internet.

Mise à part le côté inesthétique des dispositifs de visualisation et d'interaction, la RA mobile pose également des problèmes d'acceptation auprès du grand public notamment concernant la sécurité et le secret des données. En effet, une des fonctionnalités revendiquées par la RA mobile dans la plupart des publicités la concernant est celui de pouvoir fournir des informations n'importe où à n'importe quel moment. Il peut s'agir d'informations concernant un lieu, une station de métro, un produit ou une personne. Dans le dernier cas, grâce à la reconnaissance faciale, un utilisateur pourrait avoir accès à toutes les informations web disponibles sur un inconnu qu'il pourrait rencontrer par hasard. Même si les données fournies sur le web sont souvent délivrées délibérément par les internautes, la question qui se pose, alors, est si l'on désire que l'on soit immédiatement identifié dans la rue et que des inconnus aient déjà des informations nous concernant, nos opinions avant la moindre interaction.

Dans cette section, nous nous sommes intéressés à la réalité augmentée mobile dans des domaines allant du tourisme à la navigation en passant par les jeux vidéos. Cette partie ne se voulait pas exhaustive mais plutôt démonstrative des applications de RA existantes ou ayant existé. Dans la section suivante, nous nous concentrerons sur des applications de RA mobiles destinées au domaine industriel.

2.3 Réalité augmentée dans l'industrie

L'impact de la réalité augmentée dans le processus industriel est évident et indéniable. En effet, des applications de réalité augmentée dédiées à l'industrie permettent d'une part d'améliorer l'efficacité des agents et d'autre part de réaliser des gains en terme de temps et de finances.

L'étude de [Sääski et al., 2008] permet d'illustrer cette évolution. Cette étude consistait à comparer les performances des agents de maintenance utilisant un manuel et de ceux utilisant une application de réalité augmentée lors d'une tâche d'assemblage. Elle montrait que ceux qui étaient guidés par le système de RA étaient 13% plus rapides dans l'exécution de la tâche par rapport à ceux qui s'aidaient du manuel. De plus, ils commettaient six fois moins d'erreurs dans le choix des outils et deux fois moins d'erreurs dans le positionnement du matériel à assembler. D'autres études telles que celles de [Baird and Barfield, 1999] ou de [Tang et al., 2003] ou de [Henderson and Feiner, 2011] ont également démontré cet apport, en terme de rapidité et d'efficacité, de la réalité augmentée dans l'industrie comparé à des manuels ou à des instructions affichées sur ordinateur. Cette rapidité vient du fait que :

- l'opérateur peut facilement localiser spatialement l'endroit où la tâche doit être effectuée.
- les troubles musculo-squelettiques et les tensions liées aux mouvements de la tête durant la tâche de maintenance sont réduits.

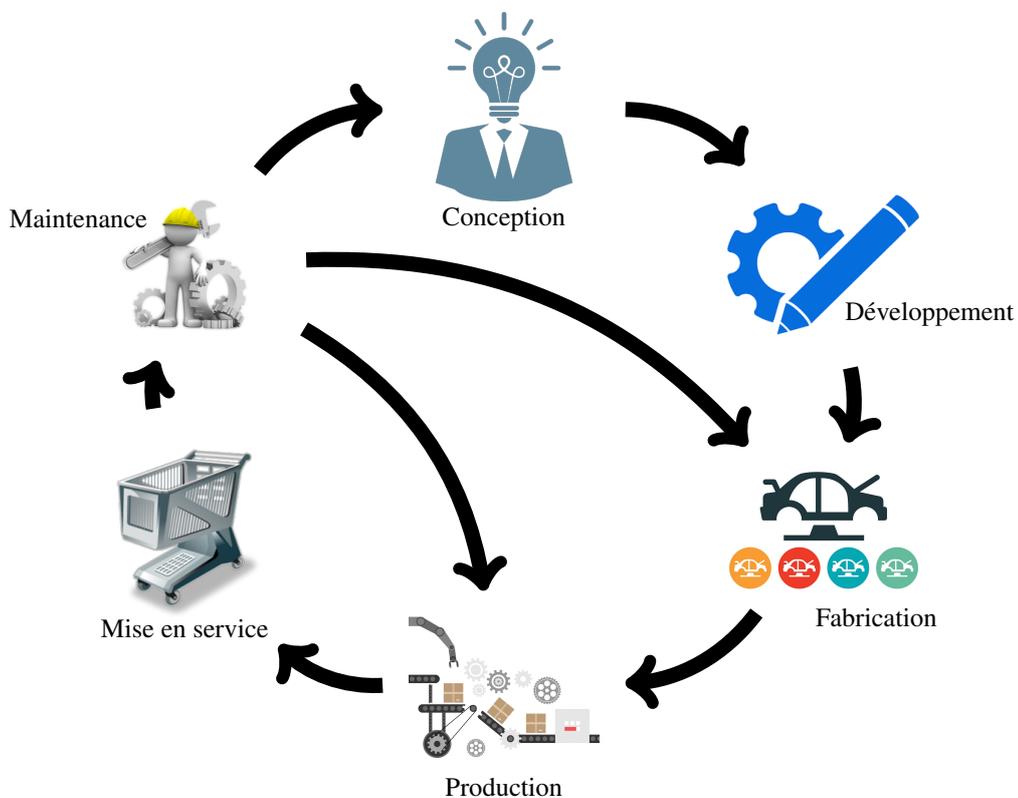


FIGURE 2.8 – Cycle de vie d'un produit.

Ce gain de temps peut être notable également tout le long du cycle de vie d'un produit *ie* de la conception à la maintenance (cf. fig. 2.8). Un produit, notamment nouveau, germe d'une idée ou d'un besoin. Cette idée doit être conceptualisée et développée. Des maquettes ou des modèles CAO (conception assistée par ordinateur) sont construits et des prototypes fabriqués avant d'être testés par un panel d'utilisateurs et modifiés suivant les erreurs notifiées ou les améliorations demandées. Si le produit séduit le panel et obtient l'accord des industriels, sa phase de commercialisation peut alors être lancée. Ainsi il est produit puis mis en service dans différents points de vente. En cas de souci, une phase de maintenance est prévue. La phase de maintenance n'intervient pas uniquement après la mise en service du produit, mais également lors de la phase de fabrication ou de production. En effet, les appareils ou les robots utilisés lors de ces phases peuvent nécessiter une intervention humaine soit pour des pannes imprévues, des opérations de maintenance préventive ou pour la personnalisation du produit.

Dans cette section, nous nous intéressons aux applications de RA mobile destinées au monde industriel ainsi qu'aux limitations qui peuvent en ressortir.

2.3.1 RA mobile dans le processus industriel

De la conception à la maintenance, des applications de RA mobile ont été proposées afin de répondre à des besoins des industriels à chaque étape du cycle de vie d'un produit.

a) Conception et développement

Dans l'industrie, la conception et le développement sont des étapes primordiales dans le processus. Celle-ci permet de conceptualiser les idées, de modéliser les produits et de planifier les étapes de construction de ces derniers. Habituellement, des prototypes sont conçus soit physiquement, soit virtuellement à l'aide de logiciels de conception afin de produire un modèle CAO. Cependant, même si les modèles réels peuvent correspondre à ce qui est attendu comme produits finaux, ils peuvent devenir rapidement coûteux à réaliser en terme de temps et de finances. Leurs structures et les matériaux utilisés sont potentiellement très difficiles à modifier. La collaboration entre les différents acteurs dans le processus de conception est également très limitée. Les modèles créés entièrement virtuellement ne favorisent pas non plus la collaboration. De plus, la visualisation de ces derniers en dimension réelle dans un environnement réel n'est souvent pas accessible, ce qui ne permet pas forcément d'évaluer l'adéquation du modèle au contexte réel. Ainsi la réalité augmentée pourrait être un plus ou une alternative. Elle permettrait d'améliorer non seulement la perception du produit final mais également la collaboration entre les différents acteurs (experts ou non) et les modifications sur le modèle virtuel.

Ainsi, dans le cas d'une visualisation simple des modèles virtuels dans un contexte réel selon des points de vue différents, on pourrait citer Fata Morgana [[Klinker et al., 2002](#)]. Celui-ci

devait permettre aux développeurs de visualiser sous tous les angles le modèle créé. En revanche, [Regenbrecht et al., 2002] favorisait la collaboration avec MagicMeeting à l'aide d'une interface utilisateur tangible. Cependant, donner la possibilité à l'utilisateur du système de RA de réaliser des changements directement sur les entités virtuelles lorsque celui-ci détecte une anomalie est une caractéristique du système qui peut être attrayante pour les concepteurs. Ainsi, afin d'évaluer un modèle, [Park, 2008] a proposé un système de RA nommé "Augmented Reality based Re-formable Mock-up" qui permettait de modifier, de manière interactive, la forme et l'apparence du produit (cf. fig.2.9).

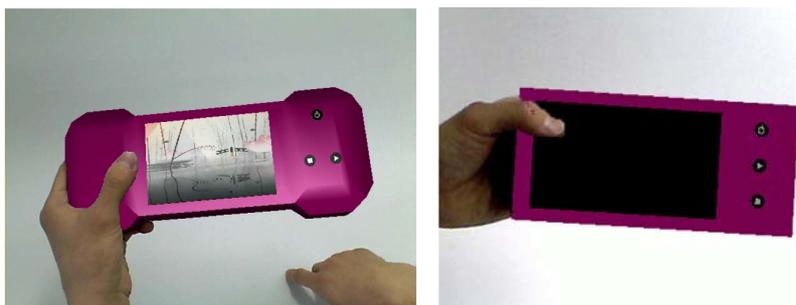


FIGURE 2.9 – Réalité augmentée pour le design de produit. [Park, 2008]

D'autres applications dédiées à la modélisation en temps réel ont été également développées. C'est le cas, par exemple, de ClonAR [Csongei et al., 2012] qui est un système permettant de remodeler des objets réels (cf. fig.2.10). Les objets réels sont scannés à l'aide de KinectFusion. Puis l'utilisateur édite l'objet qui l'intéresse et réalise des modifications dessus à l'aide d'un système haptique.

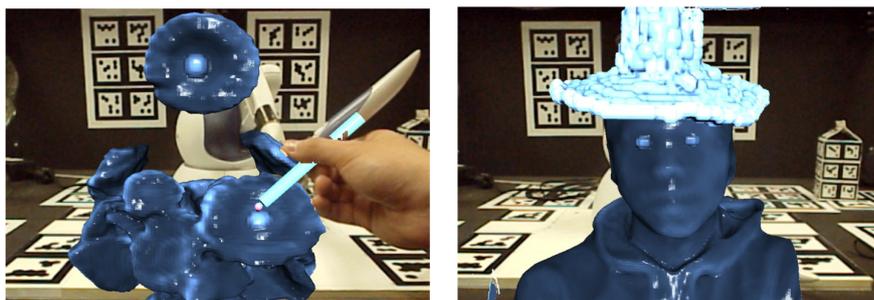


FIGURE 2.10 – ClonAR pour le remodelage d'objet réel. [Csongei et al., 2012]

SpaceDesign proposé par [Fiorentino et al., 2002] donnait la possibilité à l'utilisateur de réaliser le design d'un modèle d'automobile (cf. fig. 2.11 page suivante). Ils utilisaient des formes libres de courbes et de surfaces pour une modélisation moins contraignante. Cependant, aucun prérequis mathématique n'était nécessaire pour appréhender et prendre en main l'application. Ainsi une visualisation et une navigation 3D, une édition en temps réel et une interaction intuitive étaient fournies par SpaceDesign.

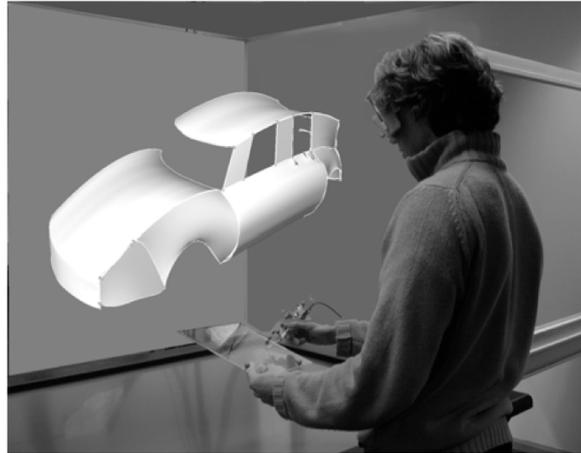


FIGURE 2.11 – SpaceDesign pour la modélisation en temps réel. [Fiorentino et al., 2002]

b) Fabrication et production

Dans le processus de fabrication et de production d'un produit spécifique, il peut être nécessaire qu'un agent intervienne. En effet, même si dans la chaîne de production, la majorité des étapes est effectuée par des automates, il reste certaines tâches qui doivent être accomplies par un opérateur. Ces tâches peuvent être délicates à réaliser et nécessitent une expertise spécifique et un apprentissage au préalable pour un novice. Ainsi afin de donner de directives en temps réel et assez explicites à l'utilisateur, des applications de réalité augmentée servant de guide ou d'instructeur ont été développées. La plupart d'entre elles [Tang et al., 2003, Billingham et al., 2008, Hakkarainen et al., 2008, Gorecky et al., 2011] se sont surtout axées sur l'assemblage et le désassemblage des pièces par un opérateur.

[Friedrich et al., 2002] ont introduit ARVIKA un système de réalité augmentée mobile pour le design et la production dans les industries automobiles et aérospatiales. Il s'agissait d'un système basé sur l'utilisateur permettant d'améliorer l'ergonomie des applications et l'utilisabilité de ces dernières par les ingénieurs.

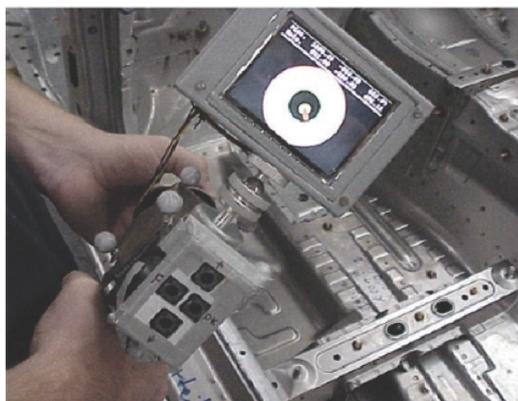


FIGURE 2.12 – Pistolet de soudure intelligent de ARVIKA . [Friedrich et al., 2002]

c) Mise en service et maintenance

Dans le monde industriel, il était habituel pour un agent de maintenance d'avoir en sa possession un ou un ensemble de manuels. Ce dernier lui permet d'obtenir des informations sur les étapes à effectuer en cas de soucis matériels ou autres bien particuliers. Il arrivait souvent que les manuels ne soient pas assez explicites pour la compréhension, ni pédagogiques ou ne contiennent pas suffisamment d'informations techniques pour accomplir la tâche de maintenance. Ils supposent souvent que l'opérateur a déjà de l'expérience et toutes les compétences requises et nécessaires pour comprendre et déchiffrer les instructions écrites. Ceci n'est pas souvent le cas. Un débutant ou une personne qui n'a jamais rencontré un problème décrit par le manuel ne peut pas toujours appréhender la manière dont les explications sont présentées. Le mainteneur peut alors se trouver dans une situation de stress ou d'incertitude liée au fait qu'il ne sache pas quels sont les bons gestes afin de réaliser correctement son boulot ou à des situations d'urgence comme, par exemple, dans l'aviation et l'aéronautique où un agent peut avoir à opérer sur un avion alors que des passagers sont en train d'embarquer. Ces situations peuvent induire des erreurs de la part des opérateurs de maintenance [Rashid et al., 2013]. La RA, en superposant les informations adéquates ou des modèles 3D manipulables dans l'environnement de travail de l'utilisateur, peut réduire considérablement ce type d'erreurs.

Ainsi, c'est pour pallier ce type de problèmes que des applications de réalité augmentée dédiées spécialement à la maintenance en milieu industriel ont vu le jour. Par exemple, KARMA (*Knowledge based Augmented Reality for Maintenance Assistance*) proposé par [Feiner et al., 1993] constituait une aide pour la maintenance et un guide pour des tâches de réparation. Il avait été testé dans le cadre d'une assistance à la maintenance et à la réparation d'une imprimante laser, notamment pour des tâches telles que le remplissage du papier à bac et le changement de la cartouche de toner (cf. fig2.13). Ils construisirent leur base de connaissance constituée d'objets virtuels à partir d'IBIS (*Intent-Based Illustration System*) qui est un moteur graphique permettant la modélisation d'objets et l'animation. Il permettait ainsi la représentation virtuelle des éléments de l'imprimante. Il affichait les instructions à l'utilisateur porteur d'un visiocasque en représentant les objets virtuels en mode fil de fer.

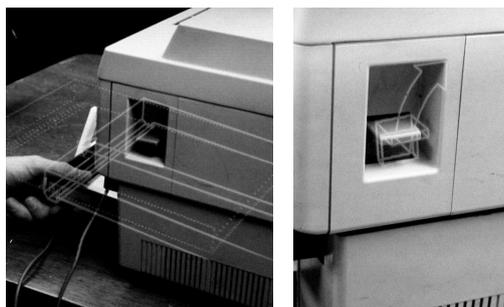


FIGURE 2.13 – Système KARMA pour l'assistance à la maintenance. [Feiner et al., 1993]

[Henderson and Feiner, 2009, Henderson and Feiner, 2011] ont également présenté une application dédiée au mécanicien militaire. Elle leur permettait de réaliser des tâches de maintenance sur des véhicules blindés de type tourelles (cf. fig.2.14). Elle avait été conçue pour faciliter la compréhension des instructions à suivre afin d'atteindre assez rapidement l'objectif de réparation visée.

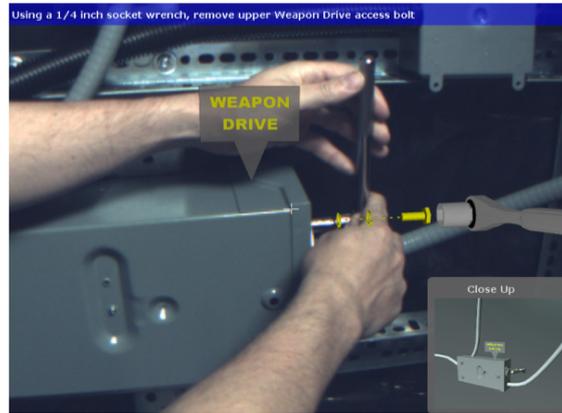


FIGURE 2.14 – Assistance pour l'assemblage d'une pièce mécanique. [Henderson and Feiner, 2009]

Par ailleurs, [Friedrich et al., 2002] introduisent ARVIKA pour des applications dans l'industrie, notamment automobile et aérospatiale. Ils l'utilisent, par exemple, pour réaliser des comparaisons directes entre les résultats issus des tests et ceux issus de la modélisation du produit (cf. fig.2.15). Ainsi lors d'un test de collision d'une automobile, ils affichent le modèle de déformation prédit par les calculs sur le véhicule accidenté dans des conditions réelles afin que l'ingénieur de développement puisse comparer les deux résultats et mettent ainsi, en avant, les différences pouvant exister. En plus de l'information graphique qui est déterminée en utilisant une méthode des éléments finis sur une grille et dont les résultats sont déterminés à l'aide des données telles que les contraintes, la tension, l'énergie interne, ils ajoutent d'autres informations



FIGURE 2.15 – ARVIKA dans le cadre d'un test de collision d'une automobile. [Friedrich et al., 2002]

textuelles telles que la vitesse.

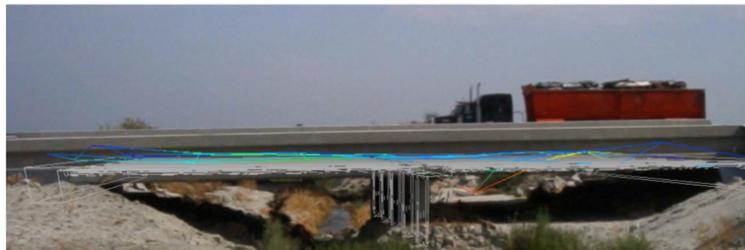


FIGURE 2.16 – Représentation du « stress » (contraintes) d'un pont. [Re, 2013]

Afin de déterminer si la structure d'un pont peut être considéré comme un succès, [Clothier and Bailey, 2004] ont proposé une solution de réalité augmentée pour représenter et visualiser les contraintes et les déformations subies par celui-ci. Cette application a été testée sur le pont Kings Stormwater situé dans le désert de Californie. Plusieurs capteurs, notamment des jauges de déformations, des potentiomètres linéaires, des accéléromètres, un thermomètre, avaient été placés sur le pont lors de sa construction. Ces derniers récupèrent des données liées au climat, au trafic, aux vibrations subies par le pont à cause du passage des véhicules et à cause du mouvement du terrain, etc. Les informations recueillies sont ensuite analysées selon l'intérêt de l'ingénieur et affichées directement en superposition au pont.

2.3.2 Limitations

Pour des applications de réalité augmentée dans le milieu industriel, des compromis sont nécessaires. Selon l'application, un choix doit être à faire entre la rapidité, la précision et donc la tolérance aux erreurs. En effet, le caractère temps réel des applications est important puisqu'un des objectifs de ces systèmes est de gagner du temps en fournissant des informations utiles et adéquates au moment précis où elles sont nécessaires. De plus, l'interaction avec l'opérateur requiert cette caractéristique afin de traiter correctement les instructions de l'utilisateur. Une latence entre les données entrantes, donc celles fournies par l'utilisateur, et les données sortantes c'est-à-dire le rendu doit être suffisamment imperceptible afin de réduire non seulement la gêne mais également des erreurs. La robustesse du système face à des conditions contrôlées ou inconnues suggère des stratégies différentes à adopter.

Par ailleurs, un système de RA industriel doit pouvoir assurer une cohérence spatio-temporelle des augmentations. Le suivi et l'estimation de la pose de la caméra dans ces systèmes sont des éléments centraux afin de maintenir cette cohérence. Or les objets intervenant dans un environnement industriel sont souvent très peu texturés et leur forme reste réglementée et standardisée. Ainsi les algorithmes se basant sur la détection des points d'intérêt peuvent souvent ne pas être adéquat dans cet environnement. De même, la reconnaissance à l'aide des segments ou de *templates* ne permettent pas toujours d'identifier les objets mécaniques suivant n'importe

quel point de vue.

Une autre difficulté récurrente est celle liée à l'acceptabilité des systèmes de RA auprès des utilisateurs finaux [Pentenrieder et al., 2007]. L'utilisabilité et l'ergonomie du système doivent être prises en considération lors de l'implémentation des applications afin qu'elles puissent être effectives et efficaces dans le domaine industriel. Ces interfaces doivent être suffisamment intuitives, complètes et facilement compréhensibles par les agents industriels afin que le système de RA reste attractif pour les opérateurs. De plus, le fait que les systèmes de RA dans ce domaine répondent à des besoins spécifiques pour une tâche précise, ne permet pas de les rendre généralisables et évolutifs [Fite-Georgel, 2011]. Cela nécessite donc l'implication des différents acteurs intervenants dans la chaîne de production pour faciliter son acceptation. Par ailleurs, afin de favoriser l'acceptabilité du système de RA auprès des opérateurs, il est recommandé de maintenir une certaine continuité dans la méthodologie utilisée lors du processus industriel [Re, 2013].

En outre, la sécurité des agents et du matériel doit être préservée. Si le système est encombrant et inconfortable pour l'opérateur, de telle sorte qu'il gêne son environnement de travail, des erreurs fatales peuvent survenir pouvant le mettre en danger ou compromettre l'équipement industriel. Les occultations doivent être correctement gérées non seulement afin de conserver la continuité entre le monde réel et virtuel mais également afin de ne pas gêner la vue de l'utilisateur durant la tâche à réaliser. De plus, le système doit être suffisamment fiable et fonctionner de manière robuste et stable dans un environnement industriel.

Comme le souligne [Regenbrecht et al., 2007], les systèmes de RA pour l'industrie manquent encore de maturité en terme de suivi, de dispositifs de visualisation, de gestion du contenu, de gestion des contraintes telles que les variations de luminosité, les changements d'échelle, les occultations ou les mouvements brusques, etc. Même si des avancées technologiques (puissance de calcul des appareils mobiles, performances des systèmes de communication,...) ont été effectuées, ces derniers restent des enjeux majeurs pour un véritable déploiement industriel.

Dans cette section, nous avons présenté un certain nombre d'applications de réalité augmentée mobile présents dans des domaines aussi variés que le tourisme, le marketing ou les jeux vidéos et notamment dans l'industrie. Nous y avons également relevé les limitations de ces solutions ne permettant pas souvent leur "extraction" (ou popularisation) du monde de la recherche. Dans la suite, nous nous intéresserons plus particulièrement à ce qui touche l'aide à la maintenance et au contrôle des actions de l'utilisateur.

2.4 Scénario de réalité augmentée pour la maintenance

Dans cette partie, nous intéressons essentiellement aux méthodes utilisées pour réaliser un scénario pour l'aide à la maintenance et au contrôle. Nous discuterons des principes d'organisation des instructions afin de les fournir à l'opérateur ainsi que des solutions afin de les afficher automatiquement suivant le changement de contexte dans l'environnement de travail du mainteneur.

2.4.1 Conception du scénario de maintenance

Concevoir un scénario de maintenance adapté aux utilisateurs et répondant aux besoins des industriels est une étape cruciale dans l'élaboration d'un système de RA pour l'industrie. Celle-ci doit être pensée avant tout développement et établit hors ligne avant tout début de simulation.

Comme les tâches de maintenance sont constituées principalement d'actions d'assemblage et de désassemblage de pièces entre elles, deux structures s'en extraient afin d'établir un scénario de maintenance :

- une **structure hiérarchique** qui correspond à un diagramme fonctionnel ou structurel *ie* de relations entre les composants du produit à assembler (cf fig. 2.17). Ce type de fiche montre à quoi devrait ressembler un objet à un moment précis de l'assemblage. Elle s'inspire de la manière dont la plupart des gens pensent le fonctionnement des tâches d'assemblage. Comme un objet peut être segmenté en de petites parties structurées hiérarchiquement, chaque élément peut être placé spatialement dans l'environnement avec des lignes directrices indiquant comment l'attacher à l'élément précédent [Agrawala et al., 2003] ou avec une simulation montrant les interactions avec une autre structure [Zauner et al., 2003, Pang et al., 2005]. Par exemple, [Zauner et al., 2003] identifie à chaque étape l'élément de base, puis affiche une petite simulation du placement des éléments avec cet élément de sa position courante à celle que devrait avoir l'ensemble assemblé. Cette structure est facilement formalisable puisque les contraintes géométriques et les degrés de liberté entre deux composantes mécaniques sont clairement identifiables et définissables.
- et une **structure linéaire** ou **procédurale** qui correspond à un diagramme ou une séquence d'actions à réaliser afin d'obtenir l'objet à assembler. Il s'agit de montrer comment joindre deux parties d'un objet à travers une simulation 3D, une vidéo, des icônes ou du texte. C'est une structure qui est souvent préférée dans la littérature ([Engelke et al., 2015, Goto et al., 2010, Raghavan et al., 1999, Feiner et al., 1993, Tang et al., 2003, Didier et al., 2005]) car elle est plus facile à suivre par les utilisateurs. En général, la tâche d'assemblage est fragmentée en de petites actions regroupées en

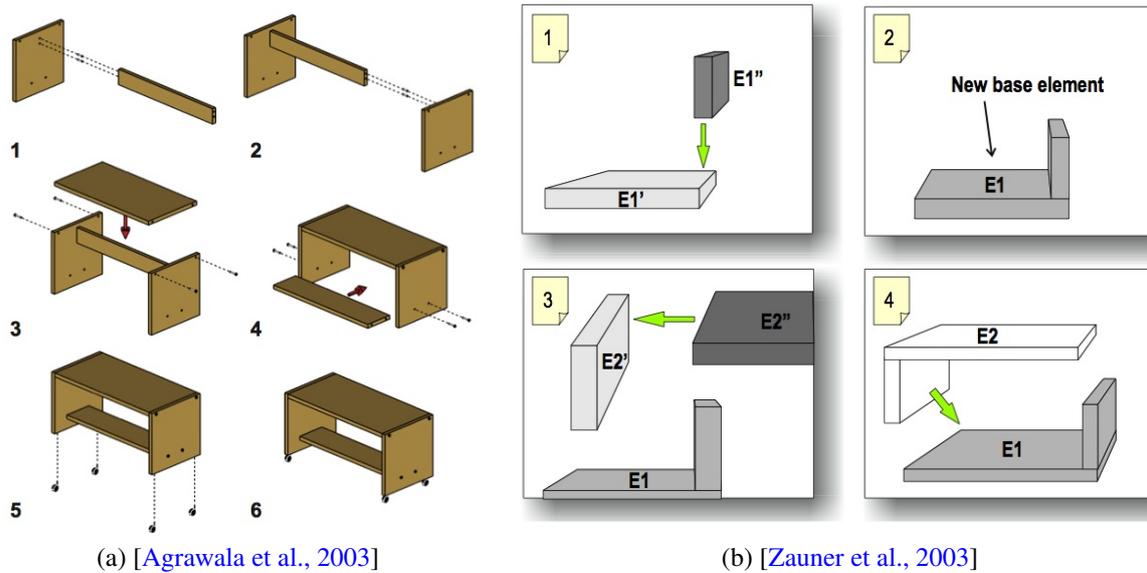


FIGURE 2.17 – Exemples de structure hiérarchique pour l’assemblage de pièces.

des étapes principales. Celles-ci indiquent l’état du système au début de la petite action et son état final. Les instructions successives sont organisées en graphe d’états. De multiples (mais limitées) hypothèses sont testées et vérifiées afin d’établir l’évolution de l’action. C’est le cas, par exemple, avec qui le système AMRA, qui est une aide dans l’apprentissage des mécanismes liés à la maintenance, génère les procédures de maintenance sous forme d’un arbre structuré en utilisant des technologies basées XML [Didier et al., 2005]. Il n’affiche pas uniquement des animations des actions à réaliser mais également des symboles pertinents donnant des précisions sur les mouvements et permet à l’utilisateur d’interagir avec les éléments virtuels. D’autres systèmes de RA tels que [Weck et al., 2004, Schwald and De Laval, 2003] ont également une structuration du scénario de maintenance grâce au langage XML permettant de définir ses propres balises et ainsi de créer les liens entre la base de données des composantes et les procédures de maintenance. L’étape d’identification et de classification d’actions peut parfois être fastidieuse, des stratégies ont été adoptées afin de découper automatiquement des vidéos d’un tâche de maintenance en de petites actions distinctes. Des approches d’apprentissage non supervisées ont été proposées afin de détecter des changements de contexte dans des vidéos d’aide à la maintenance et ainsi fragmenter la séquence [Damen et al., 2014, Petersen and Stricker, 2012].

L’organisation des instructions apportées à l’utilisateur est très dépendante de la manière de penser du concepteur, de sa perception visuelle, de son expérience et/ou des explications de ”donneur d’ordre”. Le guide, ainsi constitué, peut parfois se révéler incomplète et ne peut s’adapter aux capacités et aux compétences des utilisateurs finaux. Si les instructions affichées sont peu détaillées et trop complexes à comprendre et à suivre, un débutant ne pourra pas les

suivre. Tandis que si celles-ci sont trop simples et doivent absolument suivre un ordre bien précis, l'expert les trouvera très ennuyantes, restrictives et inutiles. Un compromis doit être élaboré afin de tenir compte des différents cas extrêmes. C'est ainsi que [Heiser et al., 2004] se sont questionnés sur les bonnes pratiques à adopter afin d'établir de bons ensembles de fiches d'instructions. Il en ressort de leur étude que ces instructions devraient être présentées à l'utilisateur de la manière suivante :

- étape par étape avec un diagramme pour chaque étape importante ou complexe,
- clair avec un ordre suffisamment explicite,
- une composante à ajouter à une étape spécifique doit être bien visible,
- la manière dont les composantes sont à assembler et l'endroit où elles doivent être placées doit être également visibles avec des indications à l'aide de flèches ou autres indications claires,
- une structure linéaire *ie* un diagramme d'actions est préférable à une structure hiérarchique *ie* à un diagramme structurel puisqu'elle est plus facile à suivre,
- il faudrait éviter un changement de point de vue et garder ainsi une orientation stable le long de la tâche.

Les instructions établies hors ligne par les systèmes de RA ne sont pas figées. Des systèmes tels que celui de [Pang et al., 2005, Zhu et al., 2013] proposent d'apporter une flexibilité dans l'établissement des scénarios de maintenance et de corriger des instructions enregistrées au préalable. Par exemple, [Pang et al., 2005] permettent au planificateur de jouer la séquence et de vérifier si la séquence est réalisable dans un environnement de RA. Si ce n'est pas le cas, celui-ci peut re-identifier et redéfinir les nouvelles contraintes, effectuer les changements nécessaires et, ainsi, replanifier la séquence d'instructions jusqu'à ce qu'elle soit satisfaisante. D'autres proposent des menus afin d'ajouter des annotations (symboles, textes, animations,...) dans les simulations [Engelke et al., 2015, Zhu et al., 2011], des marqueurs spécifiques afin de modifier les relations géométriques entre les pièces [Zauner et al., 2003] ou des *templates* [Knopfle et al., 2005, Weck et al., 2004].

2.4.2 Suivi du scénario de maintenance

Que ce soit dans le cadre d'une structure linéaire ou hiérarchique de la tâche de maintenance, des vidéos ou des simulations indiquant l'état initial de la petite action et son état final sont joués jusqu'à ce que l'utilisateur intervienne sur le système de RA à travers un menu de contrôle [Hakkarainen et al., 2008, Billinghamurst et al., 2008, Henderson and Feiner, 2009, Zauner et al., 2003, Goto et al., 2010]. Des commandes sont proposées à l'opérateur afin de passer à l'étape suivante ou précédente, afin de stopper la simulation, de la ralentir ou de l'accélérer. L'utilisateur peut ainsi sélectionner un élément du menu lui fournissant une nouvelle instruction ou lui permettant de sauter une étape, de revenir à l'étape précédente ou de rejouer celle cou-

rante. L'avantage principal de cette stratégie est sa flexibilité et son adaptabilité assez simple vis-à-vis de l'utilisateur. Cependant, même si ces méthodes fournissent les instructions étape par étape, c'est toujours l'utilisateur qui doit informer le système et confirmer la fin de l'action. De plus, si à une étape spécifique, ce dernier commet une erreur, celle-ci peut se répercuter et se répandre tout le long de la tâche d'assemblage et le système ne peut identifier de lui-même la méprise.

L'analyse des gestes [Ward et al., 2006, Radkowski and Stritzke, 2012] peut être également un moyen de reconnaître les actions liées aux tâches de maintenance. Dans [Ward et al., 2006], les actions basiques comme ouvrir un tiroir, serrer un boulon ou tourner un tournevis sont enregistrées en se basant sur les données fournies par un microphone et un accéléromètre. Une analyse linéaire discriminante (LDA) est appliquée sur les informations sonores et des modèles de Markov cachés (HMM) sur les informations provenant de l'accéléromètre afin de classifier chacune des actions. En revanche, en utilisant la Kinect de Microsoft, [Radkowski and Stritzke, 2012] parvient avec une précision relative à suivre les mains de l'utilisateur et à reconnaître les gestes de la main. Cette méthode peut permettre non seulement de déterminer si un opérateur a correctement exécuté une procédure et de fixer des gestes pour les transitions entre les différentes étapes. Cependant il ne permet pas d'identifier le bon positionnement des objets dans la scène et si les mouvements entre les pièces mécaniques ont été correctement effectués. [Bleser et al., 2015] utilise également une Kinect et des centrales inertiels attachées au corps de l'utilisateur. Ils définissent des descripteurs spatiotemporels basés sur les relations entre les objets et les poignets de l'utilisateur et du haut du corps. Chaque événement est appris et modélisé à l'aide des HMM et des machines à vecteurs de support (SVM). Ces modèles permettent ainsi par la suite de reconnaître les actions de l'utilisateur et de proposer une séquence vidéo relative à l'événement courant.

Des métriques comme la distance entre les objets peuvent être exploitées afin de préciser si les objets sont assemblés ou désassemblés. Bien que cette stratégie soit simple à mettre en place et peut sembler pertinente, plusieurs inconvénients se présentent. Tout d'abord, elle ne peut détecter la rupture de liens entre les objets que s'il y a une translation entre eux. Si c'est une rotation qui est réalisée, elle échouera facilement. Par ailleurs, pour estimer le moment où l'action commence, le seuil devra s'adapter à de nombreux paramètres telles que la résolution des images ou la distance entre les objets et la caméra.

Des estimateurs de mouvement comme le flot optique [Horn and Schunck, 1981] ou la mise en correspondance de bloc [Hussain et al., 2014] peuvent être adoptés. Cependant, ils permettent seulement de détecter quand un mouvement est induit soit par un objet, soit par la caméra sans aucune différenciation. De plus, ils ne sont pas efficaces dans toutes les situations, notamment lorsque les objets d'une même structure sont en mouvement au même moment. Ce qui peut fausser l'identification d'une situation d'assemblage ou désassemblage.

Des méthodes basées sur l'apprentissage supervisé ou non supervisé ou sur l'identification d'un changement d'une séquence peuvent permettre d'afficher automatiquement des instructions à l'utilisateur. Cependant, des questions restent encore en suspens, notamment sur la synchronisation de ces notifications et l'action de l'opérateur. En effet, si l'utilisateur reçoit l'information beaucoup trop tôt, celle-ci peut le rendre confus et s'il la reçoit trop tard, le système de RA devient ennuyeux et inutile, ses performances s'amenuisent ainsi. Il faudrait fournir l'instruction au moment le plus adéquat. De plus, des questions liées à l'affichage se posent également puisque l'enregistrement des instructions se réalise souvent suivant le point de vue du planificateur. Si des variations de point de vue, d'échelle ou de vitesse d'exécution qui peuvent survenir du fait des appareils utilisés ou du physique de l'utilisateur, ce type de reconnaissance peut parfois faillir et proposer des instructions erronées ou inadéquates. Par ailleurs, ces méthodes permettent un suivi du scénario enregistré dans la base de données, mais ne spécifient pas à l'utilisateur s'il a correctement effectué une tâche ou placé les pièces mécaniques.

2.5 Bilan

Dans ce chapitre, nous avons, dans un premier temps, analysé les caractéristiques de certains systèmes et applications de réalité augmentée mobiles afin de les classer et en déduire les limitations possibles dans chaque groupe. Nous relevons, notamment, les impératifs liés aux appareils mobiles (capacités de calcul et batterie notamment) qui nécessitent de penser à des stratégies d'optimisation du processus de RA afin de maintenir une fluidité dans le rendu et les interactions avec l'application. Des approches de suivi autre que celles avec des marqueurs sont à privilégier. En effet, du fait de la mobilité du système, celui doit être prêt à fonctionner dans n'importe quel environnement sans une dissémination des cibles qui peut être fastidieux et pas toujours possible.

Malgré l'utilité que pourrait avoir ces systèmes particulièrement dans le domaine industriel, ils restent souvent cantonnés dans la recherche. À part les problèmes liés à son acceptabilité et à la reconnaissance des objets mécaniques dans un environnement industriel, nous identifions également un manque d'ergonomie dans les interactions avec les méthodes mises en place pour le suivi des scénarios enregistrés pouvant réduire les avantages de RA dans ce domaine. De plus, des questions relatives à la précision, la rapidité, la tolérance aux erreurs et à la sécurité sont également à prendre en compte lors de la conception du système mobiles.

Afin de répondre à cette problématique d'optimisation de calcul des systèmes de RA mobile, nous nous sommes demandés s'il n'y avait pas un certain nombre de variables qui sont calculées dans le *pipeline* de RA qui pouvaient être "détournées" afin de faciliter l'identification

par le système des événements se déroulant dans l'environnement de travail de l'opérateur et de contrôler les actions de ce dernier. Ce sont ces questions que nous abordons dans les chapitres suivants. L'objectif est surtout de permettre une supervision automatique des scénarios de maintenance industrielle *ie* de reconnaître les tâches effectuées par l'utilisateur, d'afficher les instructions de manière automatique sans son intervention sur le système et de lui notifier les erreurs qu'il pourrait commettre. Sachant que les dispositifs de visualisation mobiles choisis (smartphones, tablettes, lunettes de RA) ont une capacité de calcul limitée et une faible puissance embarquée, il s'agit d'atteindre cet objectif sans passer par des traitements lourds et longs. En utilisant de manière appropriée les données déjà existantes, il est possible d'obtenir les résultats souhaités sans un ralentissement du système de RA.

Deuxième partie

Contributions :

Détection des tâches d'assemblage et de désassemblage

Dans un contexte de constante compétitivité, les industriels recherchent de plus en plus de technologies leur permettant de réduire les coûts de production ainsi que leurs durées. L'automatisation de la plupart des procédures permet de répondre à leurs impératifs. Cependant, même si la grande majorité des tâches de production ou de maintenance est réalisée par des robots, certaines nécessitent une intervention humaine. En effet, les robots ne sont pas toujours disponibles. Durant le processus de fabrication, des pannes imprévues peuvent survenir. Des opérations préventives de maintenance, qui requièrent un arrêt total des machines, peuvent être planifiées. Des procédures très particulières et minutieuses peuvent être requises notamment dans le cadre d'une personnalisation d'un produit spécifique. Reprogrammer ou reconfigurer l'automation industrielle, déjà mise en place, peut être très coûteux par rapport à une formation appropriée et adéquate des agents industriels. Ces derniers peuvent s'adapter assez facilement et rapidement au contexte pour accomplir des tâches spéciales de maintenance, de réparation ou de vérification.

L'utilisation des systèmes de réalité augmentée dans les industries automobiles et aérospace [Fite-Georgel, 2011, Regenbrecht et al., 2005, Tang et al., 2003] a déjà démontré l'efficacité de la RA pour l'apprentissage, la réparation, la maintenance ou le contrôle. En particulier, la performance des opérateurs, leur attention et, par conséquent, leur productivité pendant des tâches d'assemblage ont été accrues à l'aide des technologies de RA en comparaison avec l'exploitation d'un manuel d'instruction énorme et encombrant [Sääski et al., 2008]. En effet, en affichant en temps réel les informations adéquates au bon moment, au bon endroit, ces applications de RA permettraient ainsi de réaliser des actions délicates sans aucune interruption.

Malheureusement, malgré son potentiel théorique, la plupart des applications de RA existantes affichent de manière répétitive chacune des étapes de la procédure [Hakkarainen et al., 2008, Billingham et al., 2008, Henderson and Feiner, 2009] sous forme d'une animation montrant le début et la fin de la séquence à effectuer. Le passage à l'étape suivante est notifié par l'utilisateur manuellement. Ainsi ce dernier est obligé d'interrompre son action pour indiquer au système qu'il souhaite consulter la suite des instructions parce que le système n'est pas capable de savoir automatiquement et en temps réel ce qui s'est déroulé exactement dans l'environnement de travail de l'utilisateur. Non seulement, le manque de concentration et la fatigue visuelle, psychologique et physique peuvent apparaître de manière insidieuse, mais encore l'opérateur est le seul juge quant à la qualité de l'exécution de la tâche suivant son

intuition ou son expérience. Par ailleurs, le système n'identifie pas les potentielles erreurs qui ont pu survenir, ou encore, ne peut déterminer à quel niveau de la réalisation de la séquence d'actions l'utilisateur se situe précisément.

Dans cette étape de notre recherche, nous proposons d'utiliser des techniques de vision par ordinateur pour donner la capacité au système de détecter et de suivre automatiquement chacune des étapes de la procédure de maintenance que l'opérateur doit accomplir. Afin de construire ce système, nous utilisons les informations contextuelles de la procédure de maintenance et les données associées enregistrées dans le système comme, par exemple, des modèles CAO (conception assistée par ordinateur).

Dans un premier temps, en considérant l'assemblage comme une structure rigide, nous présenterons notre approche afin d'identifier les phases d'assemblage et de désassemblage. Dans un deuxième temps, en définissant l'assemblage comme une structure dynamique, nous discuterons des approches afin de reconnaître les mouvements entre les objets effectués par l'utilisateur pouvant permettre d'identifier les potentielles fausses manoeuvres de ce dernier.

Chapitre 3

L'assemblage en tant que structure rigide

Dans cette partie, nous définissons l'assemblage de la manière suivante :

Définition 1. *Un assemblage est une transformation rigide existante entre nos pièces mécaniques. Si un paramètre de la matrice de transformation varie au cours du temps, lors d'un changement dans l'environnement de travail de l'utilisateur, un désassemblage se produit.*

Suivant cette définition, on considèrera que deux pièces, liées de manière rigide, sont assemblées. Par contre, dès que ces pièces sont séparées, donc lorsqu'un paramètre de la matrice les liant de manière rigide change, on estime qu'elles sont désassemblées.

Afin de déterminer dans quelle phase (assemblage ou désassemblage) se trouve l'utilisateur, nous introduisons, dans un premier temps, une métrique et un cadre défini pour son exploitation. Nous montrerons que cette donnée peut permettre une automatisation d'un système de RA, un suivi du scénario de maintenance et une identification des mauvaises manœuvres de l'opérateur selon le scénario prédéfini. Il s'agit d'une approche qui diffère de celles présentées dans la section [2.4.2](#).

Dans un deuxième temps, nous présenterons également notre approche afin d'identifier le moment du changement de phase (assemblage ou désassemblage) et la comparerons à d'autres approches permettant d'identifier des points de rupture dans un signal.

3.1 L'erreur de reprojction, une métrique pour la détection d'actions d'assemblage ou de désassemblage

Comme notre application finale de RA est censée fonctionner sur des dispositifs mobiles tels que des smartphones ou des tablettes dont les capacités de calcul sont limitées, nous nous sommes intéressés à une métrique qui permettrait d'identifier les tâches d'assemblage et de désassemblage sans ajouter un surplus de calcul dans le processus. Celle-ci doit donc être simple et rapide à calculer. De plus, comme nous sommes dans un contexte de maintenance industrielle

où chaque élément peut être décrit par un modèle CAO, notre approche d'estimation de pose est basée modèle [Dementhon and Davis, 1995].

De ces faits, nous nous sommes intéressés à l'erreur de reprojection qui est souvent déterminée lors du processus d'optimisation et d'estimation de la pose de la caméra. C'est une donnée qui est donc généralement directement disponible sans aucun surcoût. Dans cette partie, nous nous efforçons de démontrer notre hypothèse suivante : *"l'erreur de reprojection combinée à un modèle adéquat permet de caractériser les tâches d'assemblage et de désassemblage"*.

Nous avons analysé et exploré son évolution durant des séquences de réalisation de ces actions suivant deux configurations initiales différentes de nos pièces :

- nous considérons nos pièces comme appartenant à un seul et même modèle,
- nous estimons que nos pièces sont des entités 3D indépendantes.

Dans cette section, nous introduirons notre métrique suivant la configuration considérée et nous discuterons de résultats obtenus.

3.1.1 Pipeline basé vision par ordinateur

Le processus général pour estimer la pose de la caméra et pour déterminer l'erreur de reprojection est illustré sur la figure 3.1. Les primitives de la scène réelle sont extraites, comparées et mises en correspondance avec les primitives des modèles du monde réel enregistrés et associés à des éléments virtuels à afficher. Ces primitives peuvent correspondre à des points d'intérêt, des lignes, des textures, des ellipses, ... En utilisant les mises en correspondance 2D-3D, la pose de la caméra est estimée. Les résultats obtenus à ce stade peuvent être employés pour régler les paramètres de la caméra virtuelle afin d'assurer une cohérence spatiale entre le réel et le virtuel. Sur ce *pipeline* (fig. 3.1), la dernière étape consiste à déterminer l'erreur entre les primitives extraites et celles projetées en se basant sur la nouvelle pose trouvée.

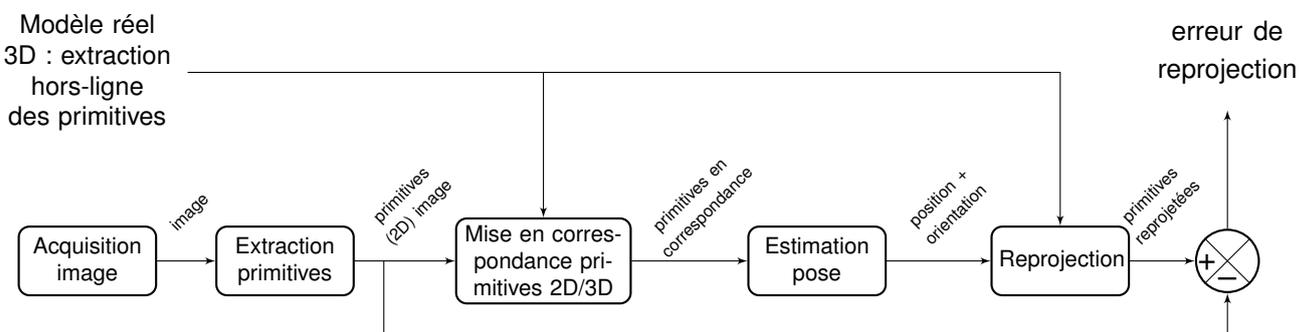


FIGURE 3.1 – Pipeline général pour l'estimation de la pose et le calcul de l'erreur de reprojection suivant les approches basées modèles.

Notre objectif est de prouver l'hypothèse selon laquelle l'analyse de l'erreur de reprojection permettrait d'identifier quand deux objets sont assemblés ou désassemblés. Dans cette partie,

nous détaillerons le calcul théorique de l'erreur de reprojection suivant deux hypothèses de travail différentes. La première considère que les différents éléments 3D constituant un assemblage sont représentés par un seul et unique objet et la seconde considère qu'ils sont indépendants les uns aux autres. Nous supposons que si l'erreur de reprojection est associée astucieusement à une configuration préalable adéquate de l'environnement, celui-ci enregistrerait des changements potentiels dans la scène. Il permettrait ainsi de les détecter au cours du temps. Par exemple, si on considère que des pièces assemblées sont rigidement liées, nous nous attendrions à une erreur de reprojection proche de 0 et stationnaire ou, du moins, stationnaire uniquement tant qu'aucun mouvement n'est effectué. A partir du moment où le désassemblage commence, cette erreur devrait varier au cours du temps.

Nous supposons dans nos notations que les primitives que nous utilisons sont des points représentés sous forme de coordonnées 2D pour ceux issus des images et sous forme de coordonnées 3D pour ceux issus des modèles enregistrés.

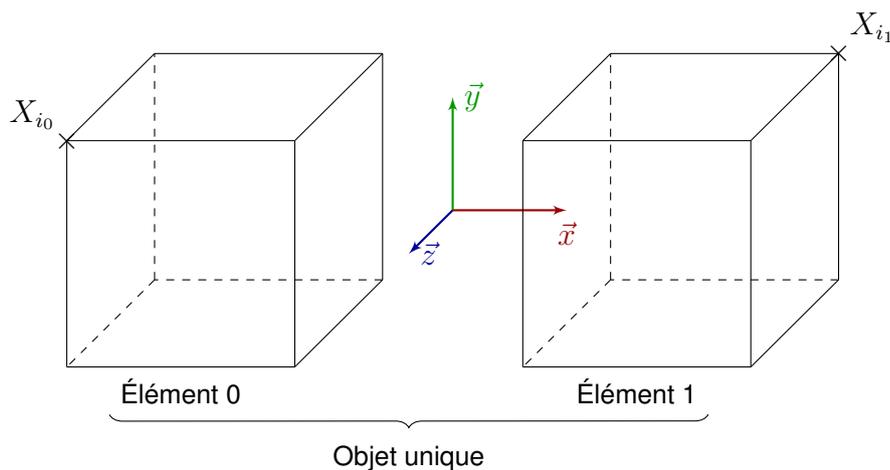
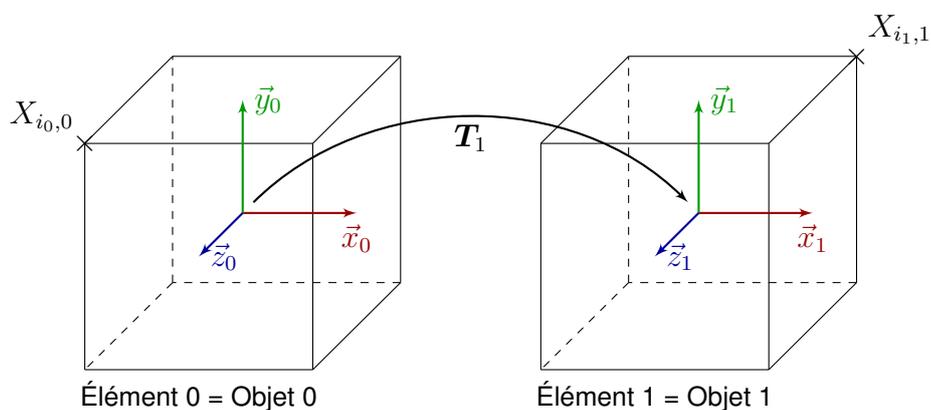


FIGURE 3.2 – Exemple de configuration des modèles 3D enregistrés.

a) Pièces d'assemblage sous forme d'objets indépendants

Les pièces d'assemblage sont décrites comme étant des modèles 3D indépendants les uns des autres. Les primitives des modèles sont extraites hors ligne et leur localisation 3D relativement au repère local auquel elles appartiennent. La figure 3.2a illustre les notations utilisées.

Dans une tâche d'assemblage, les éléments sont reliés entre eux suivant une transformation rigide. Ainsi les repères locaux liés au centre de objets sont liés par cette même transformation. Elle peut être déterminée initialement hors ligne ou durant une phase où les objets sont assemblés.

Soient $\mathbf{x}_{i,j,t}$ les coordonnées 2D de la primitive i extraites de l'image t et $\mathbf{X}_{i,j}$ son correspondant 3D exprimé dans le référentiel de l'objet j . La matrice de projection $\mathbf{P}_{j,t}$, suivant le modèle de sténopé linéaire, reliant les points 3D et 2D est déterminée à l'aide de l'équation suivante :

$$\mathbf{x}_{i,j,t} \cong \mathbf{P}_{j,t} \mathbf{X}_{i,j} \quad (3.1)$$

$\mathbf{P}_{j,t}$ peut également s'exprimer de la manière suivante :

$$\mathbf{P}_{j,t} = \mathbf{K} [\mathbf{R} | \mathbf{t}]_{j,t} \quad (3.2)$$

Où

- \mathbf{K} est la matrice intrinsèque de la caméra qui est établie hors ligne lors d'une phase de calibration [Zhang, 1999] et décrivant les capacités intrinsèques de projection d'une caméra, transformant ainsi un point 3D exprimé dans le repère de la caméra en un point 2D exprimé dans le repère de l'image.
- $[\mathbf{R} | \mathbf{t}]_{j,t}$ est la matrice reliant un point 3D exprimé dans le repère global ou de référence en un point 3D exprimé dans le système de la caméra. Cette matrice est la combinaison d'une rotation et d'une translation et représente les paramètres extrinsèques de la caméra. Elle constitue la pose de la caméra.

La combinaison de la calibration hors ligne et de l'estimation en ligne de la pose de la caméra permet d'obtenir la matrice de projection de chaque objet modélisé.

Comme nous considérons que les pièces d'assemblage sont liées par une transformation rigide, les relations entre les repères liés à chacune des pièces sont donc définies par cette même transformation. Celle-ci peut être déterminée soit dans une phase hors ligne, soit en se basant sur la première image acquise lors d'une phase où les pièces sont assemblées. Si la matrice de transformation est disponible dans le fichier issu de la CAO (ce qui est peut-être le cas dans l'industrie), celle-ci sera privilégiée.

Ainsi la matrice de transformation \mathbf{T}_j entre l'objet j et l'objet de référence 0 peut être déduite en se basant sur la relation suivante :

$$[\mathbf{R}|\mathbf{t}]_{j,t} = \mathbf{T}_j[\mathbf{R}|\mathbf{t}]_{0,t}, \forall j \in \{1..n\} \quad (3.3)$$

L'erreur de reprojection e_t suivant notre modèle correspond à la valeur moyenne de la distance entre les points 2D extraits à l'image t et le projeté de son correspondant 3D est exprimée ainsi :

$$e_t = \sqrt{\frac{1}{m} \sum_l \sum_k \|\mathbf{y}_{k,l,t} - \mathbf{x}_{k,l,t}\|^2} \quad (3.4)$$

où

- m nombre total de primitives mises en correspondance utilisées pour l'estimation de la matrice de projection,
- l l'index de l'objet 3D,
- k l'index de la primitive 3D mise en correspondance dans la "liste" des primitives d'un objet 3D,
- $\mathbf{x}_{k,l,t}$ les coordonnées 2D de la primitive image mise en correspondance et
- $\mathbf{y}_{k,l,t}$ les coordonnées 2D de la primitive 3D projetée obtenue en combinant les équations 3.1, 3.2 et 3.3. Ainsi,

$$\mathbf{y}_{k,l,t} \cong \mathbf{KT}_l[\mathbf{R}|\mathbf{t}]_{0,t}\mathbf{X}_{k,l} \quad (3.5)$$

b) Pièces d'assemblage sous forme d'un objet unique

La seconde configuration modélise les pièces d'assemblage comme un seul et unique objet. Les primitives sont extraites et décrites suivant le repère lié à l'objet global. La figure 3.2b page 85 illustre cette configuration dans le cas où nous n'avons que deux pièces d'assemblage.

De petites différences existent avec le modèle précédent. Comme les différents éléments 3D sont regroupés sous forme d'un seul objet, une seule matrice de projection est calculée à chaque image.

Soient $\mathbf{x}_{i,t}$ les coordonnées 2D de la primitive i extraite à l'image t et \mathbf{X}_i son correspondant 3D exprimé dans le repère de l'objet global. On obtient ainsi la matrice de projection de la manière suivante :

$$\mathbf{x}_{i,t} \cong \mathbf{P}_t\mathbf{X}_i \quad (3.6)$$

Nous en déduisons l'erreur de reprojection :

$$e_t = \sqrt{\frac{1}{m} \sum_k \|\mathbf{y}_{k,t} - \mathbf{x}_{k,t}\|^2} \quad (3.7)$$

Où :

- m est le nombre total de primitives mises en correspondance utilisé pour l'estimation de la pose,
- k l'index d'une primitive,
- $\mathbf{x}_{k,t}$ ses coordonnées image et
- $\mathbf{y}_{k,t}$ les coordonnées de son projeté déterminées de la manière suivante :

$$\mathbf{y}_{k,t} \cong \mathbf{P}_t \mathbf{X}_k \quad (3.8)$$

L'idée derrière ces modèles est que si deux pièces se désassemblent, les relations entre les primitives de chacune des pièces se modifient. Si nous considérons que cette relation est constante au cours du temps, l'erreur de reprojection telle nous l'avons exprimée se modifie également. Ainsi en évaluant et en analysant l'évolution de l'erreur de reprojection au cours d'une séquence d'assemblage/désassemblage, nous pouvons l'utiliser comme métrique pour déterminer quand des pièces assemblées sont désassemblées et vice-versa.

3.1.2 Expérimentation et résultats

a) Protocole expérimental

Dans nos expériences, nous analysons l'évolution de l'erreur de reprojection dans des séquences vidéos créées synthétiquement et dans des vidéos acquises dans des conditions réelles d'utilisation. Nous avons choisi de travailler dans un premier temps avec deux cubes dont chaque face comporte une image bien texturée (fig. 3.4a page 90). Ainsi, ces faces ont une texture distincte les unes des autres. Afin de prouver notre hypothèse, nous avons voulu nous affranchir des problèmes de détection de primitives liés au fait que les pièces mécaniques industrielles sont très peu texturées. Chaque séquence représente un mouvement bien spécifique entre les objets. Il peut s'agir de rotations (fig. 3.4d page 90, 3.6d page 92), de translations (fig. 3.5d page 91, 3.7d page 93), ou d'une combinaison des deux.

Dans les vidéos synthétiques (fig. 3.4 page 90, 3.5), les objets sont assemblés au début de la séquence. Ils sont séparés suivant une transformation canonique (ici, rotation ou/et translation), puis réassemblés suivant une transformation inverse.

En revanche, avec les vidéos acquises dans des conditions réelles (fig. 3.6 page 92, 3.7 page 93), les objets sont uniquement désassemblés. Dans tous les cas, le repère d'un objet est toujours choisi comme référentiel monde. Il est à noter que les mouvements des objets aussi bien dans les vidéos synthétiques que dans les vidéos réelles sont les mêmes.

Dans notre protocole expérimental, il faudrait également préciser les méthodes d'extraction de primitives et de description. Les primitives que nous détectons dans nos images et sur nos modèles sont des points d'intérêt. Elles sont extraites en se basant sur l'algorithme des différences de gaussiennes (DoG). Les descripteurs SIFT (*Scale-Invariant Feature Trans-*

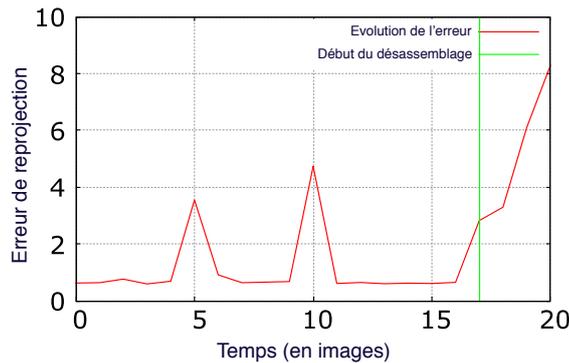


FIGURE 3.3 – Vue locale de l'évolution de l'erreur de reprojection pendant la phase initiale d'assemblage (cas du suivi multiple sur la vidéo synthétique représentant une rotation autour de l'axe Z de l'objet 1). (cf graphe entier 3.4e)

form) [Lowe, 2004, Lowe, 1999] ont été privilégiés par rapport aux descripteurs SURF (*Speeded Up Robust Features*) [Bay et al., 2008] qui sont le plus souvent utilisés en RA. En effet, les SIFT sont plus robustes en présence de flou lié au mouvement par rapport aux SURF [Gauglitz et al., 2011]. Ils présentent une meilleure précision dans la mise en correspondance entre des primitives d'images statiques et celles d'images où des objets sont en mouvement.

Une mise en correspondance symétrique en s'aidant la bibliothèque FLANN (*Fast Library for Approximate Nearest Neighbors*) [Muja and Lowe, 2009] permet des mises en correspondance fortes entre les primitives extraites des séquences d'images et celles de notre base de données. Cette méthode réalise une recherche rapide des plus proches voisins dans des espaces de grande dimension. Il permet de supprimer des *outliers* qui peuvent induire une mauvaise estimation de la matrice de projection. Dans la phase de mise en correspondance, les descripteurs extraits des images de nos vidéos sont réordonnés directement suivant leur appartenance à un modèle 3D spécifique.

Pour l'estimation de la pose de la caméra, l'approche proposée par Lepetit and al. [Lepetit et al., 2009] a été utilisée. Elle a été combinée au RANSAC (*Random sample consensus*) afin de supprimer davantage d'*outliers*.

b) Résultats

Quelque soit la méthode utilisée, nous notons une augmentation de l'erreur de reprojection directement après le début du désassemblage dans le cas des vidéos synthétiques et après un bref laps de temps dans des conditions réelles.

Dans le cadre de la rotation dans les vidéos synthétiques, pour les deux configurations, l'erreur de reprojection augmente de manière asymptotique quand les deux objets sont séparés suivant une rotation spécifique (fig.3.4e). Une stabilisation se produit lorsque les objets sont statiques, puis l'erreur de reprojection décroît dès que les objets retournent à leur état initial,

c'est-à-dire lorsqu'on les ré-assemble suivant une rotation inverse autour du même axe. Il est à noter que l'erreur n'est nulle ni avant le désassemblage (la valeur moyenne est de 1.06) ni après le ré-assemblage (fig.3.3).

De plus, elle varie d'une image à une autre suivant les *inliers* extraits durant l'estimation de la pose. Cette non nullité et cette variation de l'erreur dans un contexte quasi similaire découlent du processus d'optimisation lié à l'estimation de la pose de la caméra. En effet, l'erreur mesure l'exactitude et la précision du processus d'optimisation à un instant précis. Celle-ci peut donc varier suivant les paramètres choisis en entrée de la méthode d'optimisation (dans notre cas, il s'agit des correspondances 2D-3D utilisées à l'instant t).

Avec le suivi global, il existe un décalage par rapport aux valeurs de l'erreur de reprojection enregistrées avec la méthode considérant les objets indépendamment les uns des autres. Cela est dû au fait que les points utilisés pour le calcul de l'erreur sont ceux obtenus après la mise en correspondance et non après la détermination de la matrice de projection. Les *inliers* fournis par le RANSAC peuvent n'appartenir qu'à un élément de l'objet global. Dans notre expérience avec deux éléments constituant l'objet global, les *inliers* peuvent correspondre qu'à des points de l'élément de droite ou qu'à ceux de l'élément de gauche. Et cela peut varier d'une image à une autre. Ainsi ces *inliers* ne permettent pas relever des informations pertinentes quand à l'évolution de l'erreur de reprojection pendant l'exécution de la tâche d'assemblage. Il faut donc tenir compte également des *outliers* du RANSAC.

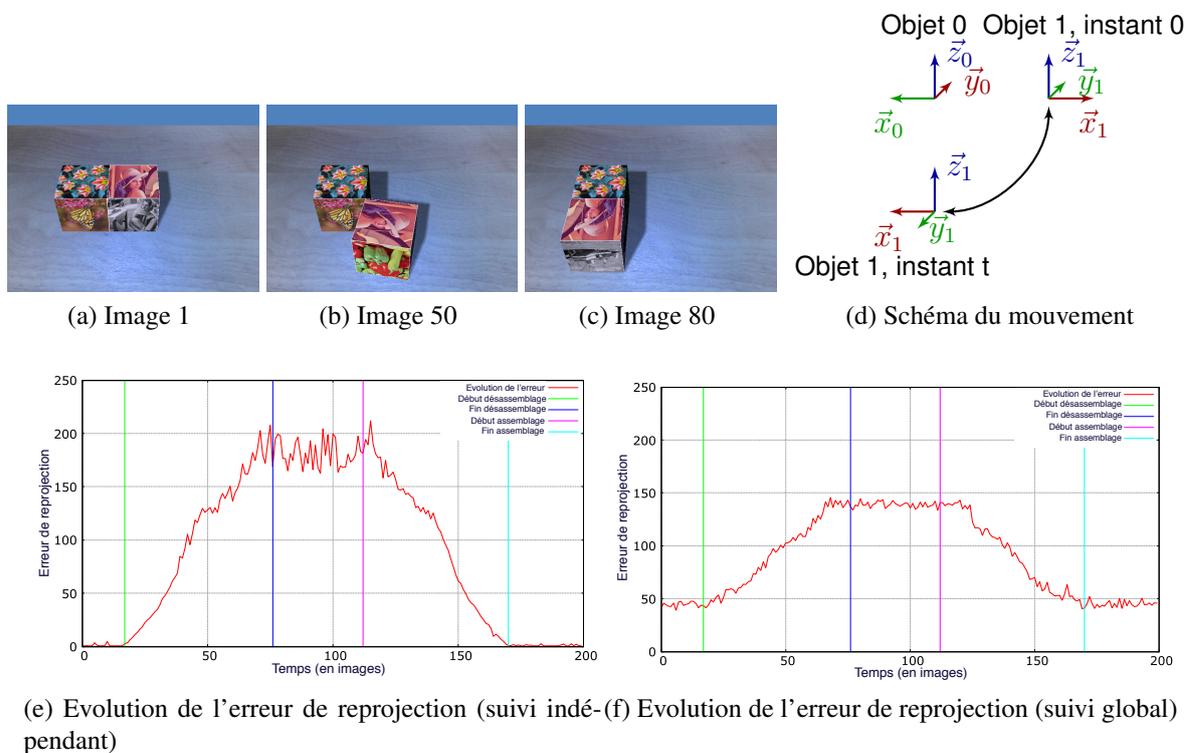


FIGURE 3.4 – Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une rotation autour de l'axe Z de l'objet 0.

Par ailleurs, les valeurs maximales dans le cadre du suivi global sont moins importantes que celles dans le cadre du suivi indépendant des éléments. Il y a un aplatissement de la courbe du fait que l'on tienne compte des *outliers* qui se répercutent sur l'ensemble de la courbe.

La même rotation a été enregistrée dans des conditions réelles mais uniquement dans le cas d'un désassemblage des pièces. Nous avons gardé la même configuration des cubes que dans les vidéos synthétiques. Avec le suivi indépendant des éléments, nous observons également un accroissement de l'erreur de reprojection durant la phase de désassemblage et une relative stabilisation quand les objets s'arrêtent de bouger (fig.3.6e) à l'instar des vidéos synthétiques (fig. 3.8a page 94). Cependant, la méthode globale montre une augmentation de l'erreur après un court laps de temps (fig.3.6f). Ceci est toujours dû à la prise en compte des *outliers* qui influencent la forme de la courbure et propagent l'erreur initiale. En revanche, les changements dans la scène sont davantage notables directement les vidéos synthétiques (fig. 3.8c page 94) puisque celles-ci sont moins bruitées.

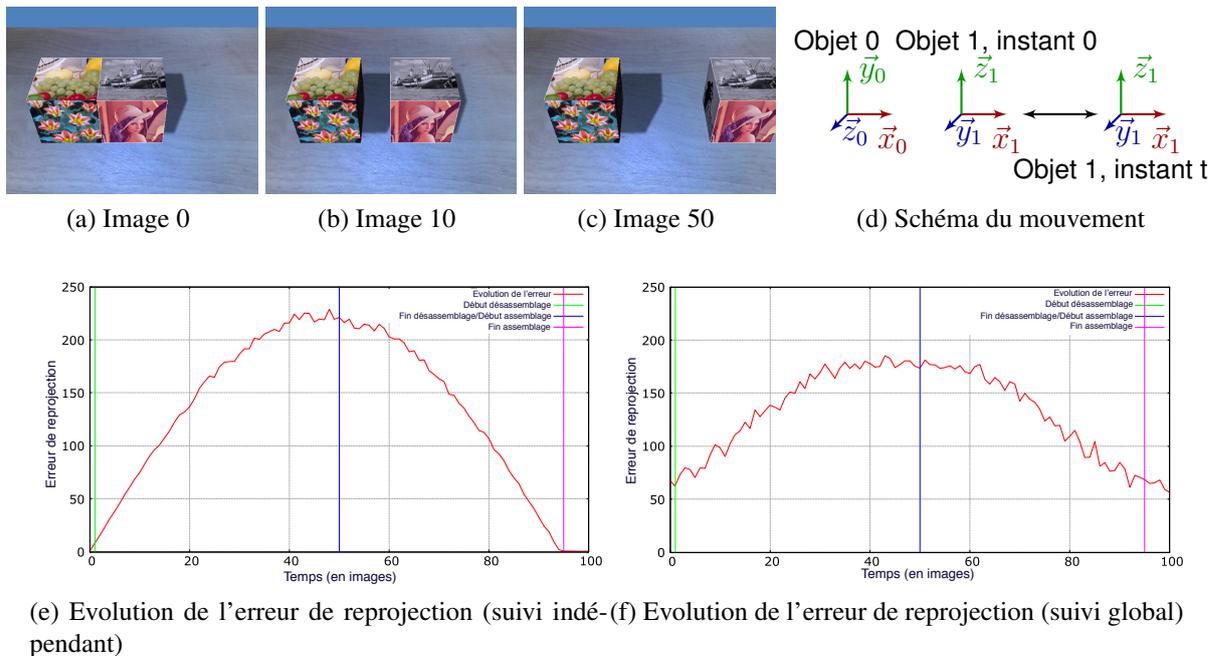


FIGURE 3.5 – Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une translation selon l'axe X de l'objet 0.

Même si, avec le suivi multiple, la courbe de l'erreur de reprojection dans les conditions réelles se superpose, au début du désassemblage, à celle des vidéos synthétiques celle-ci reste toujours au-dessus de la dernière (fig. 3.8a page 94, 3.8c page 94). Les mouvements de la caméra qui peuvent avoir lieu dans les conditions réelles et le flou du mouvement existant dans chaque image peuvent potentiellement induire ces valeurs relativement plus grandes.

Quant au temps d'exécution, il reste pratiquement le même dans les deux configurations différentes.

Les mêmes observations peuvent être faites dans le cadre d'une translation. L'erreur de reprojection augmente de manière asymptotique au début du désassemblage non seulement dans le cas des vidéos synthétiques (fig.3.5e, 3.5f) mais également dans celles acquises dans des conditions réelles (fig.3.7e, 3.7f). Cet accroissement est notable quelque soit la méthode utilisée, aussi bien avec le tracking multiple (fig.3.5e, 3.7e) que le suivi global (fig.3.5f, 3.7f).

Comme dans les vidéos synthétiques, les objets sont séparés dans un premier temps puis regroupés immédiatement, aucune stabilisation de la courbe n'est observée. Ainsi on remarque une décroissance de la courbe d'évolution durant l'assemblage immédiatement après la croissance de celle-ci.

Comme dans le cas de la rotation, les valeurs de l'erreur de reprojection dans les conditions réelles sont relativement plus grandes que dans les vidéos synthétiques (fig. 3.8b page 94, 3.8d page 94) toujours à cause du flou du mouvement et du mouvement de la caméra.

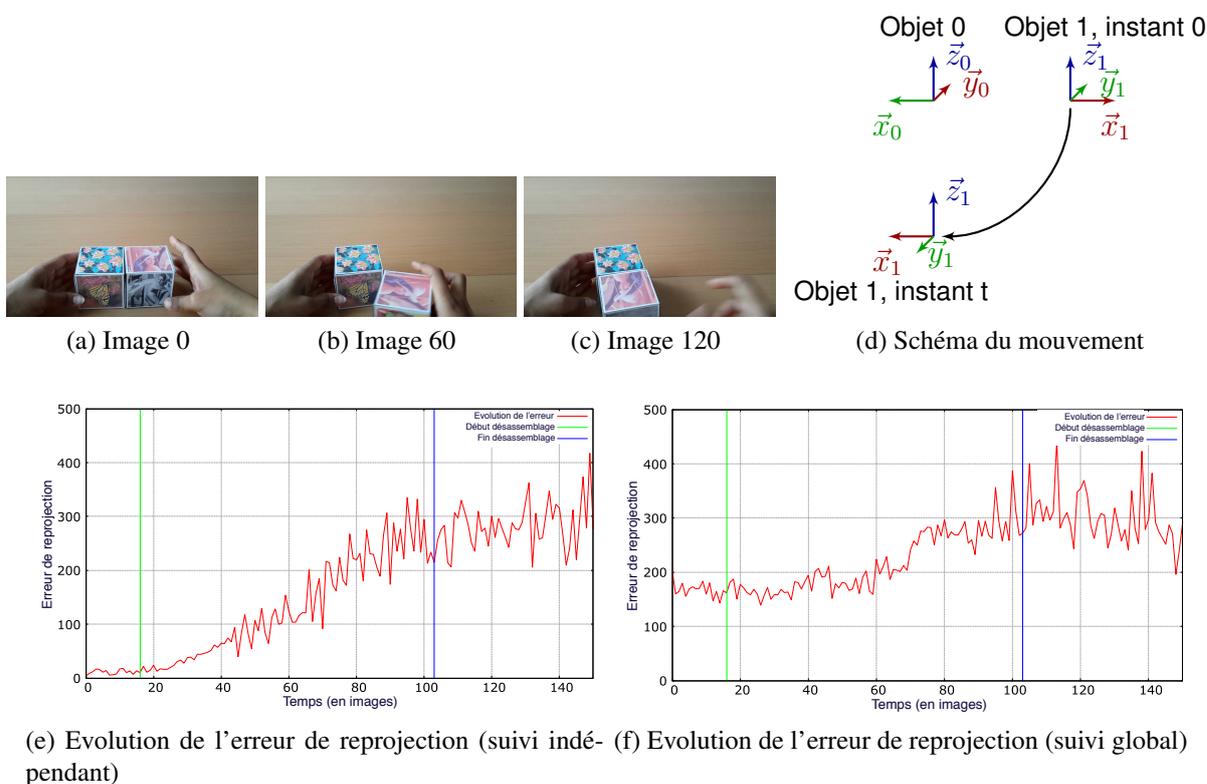


FIGURE 3.6 – Désassemblage suivant une rotation autour de l'axe Z de l'objet 0 dans des conditions réelles.

Nous avons également analysé l'effet de la taille et de la résolution de l'image sur l'évolution de l'erreur de reprojection (fig.3.9). En général, la courbe garde la même allure quelque soit la taille de l'image. Elle croît au début du désassemblage. Cependant, si nous tenons compte des valeurs lissées (fig.3.9b) à l'aide d'un filtre gaussien, il peut être noté que l'image avec la meilleure résolution (1920x1080) fournit des valeurs les plus grandes.

De plus amples résultats sont présentés dans l'annexe A pour le cas du suivi multiple suivant

des scénarios différents. Des mouvements où l'hypothèse n'est pas concluante y sont également regroupés. Ceci est sûrement dû à une mauvaise estimation de la configuration initiale des objets lors de la détection des points d'intérêt.

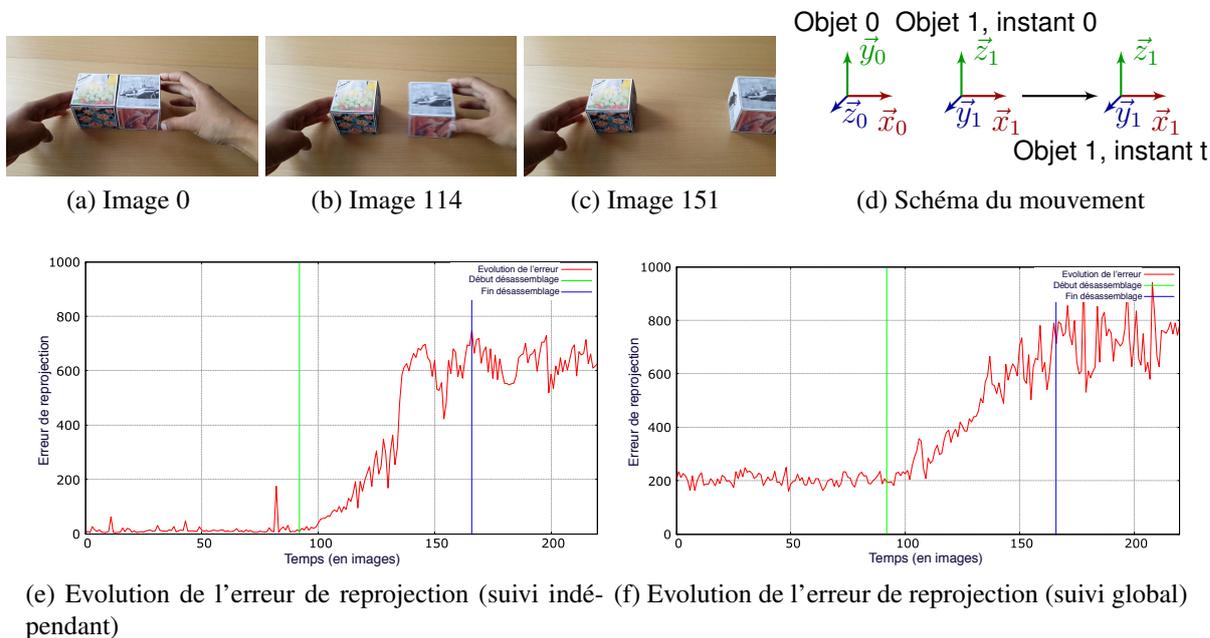


FIGURE 3.7 – Désassemblage suivant une translation selon l'axe X de l'objet 0 dans des conditions réelles.

3.1.3 Discussion

Notre hypothèse selon laquelle l'erreur de reprojection peut caractériser des tâches d'assemblage et de désassemblage a été démontrée dans cette partie. En effet, en choisissant de manière adéquate les modèles 3D (indépendant ou global) sur lesquels les primitives 3D sont extraites, en sélectionnant des méthodes robustes à chacune des étapes du *pipeline* pour l'estimation de la pose et en définissant de manière astucieuse l'erreur de reprojection suivant les modèles 3D choisis, cette dernière permet d'identifier les phases d'assemblage et de désassemblage. La combinaison des méthodes sélectionnée peut influencer les courbes d'évolution de cette erreur au cours d'une séquence. Dans notre cas, par exemple, les courbes issues du modèle global diffèrent de celles issues du modèle indépendant du fait du RANSAC qui peut limiter l'efficacité de ce modèle.

Comme la mise en place des modèles de configuration pour le suivi multiple est plus simple à mettre en place et à paramétrer durant une phase hors ligne que pour le suivi global, ce modèle est privilégié par la suite dans nos expérimentations. Par ailleurs, comme il n'y a pas un gros décalage sur les courbes d'évolution de l'erreur de reprojection dans le suivi indépendant, il serait

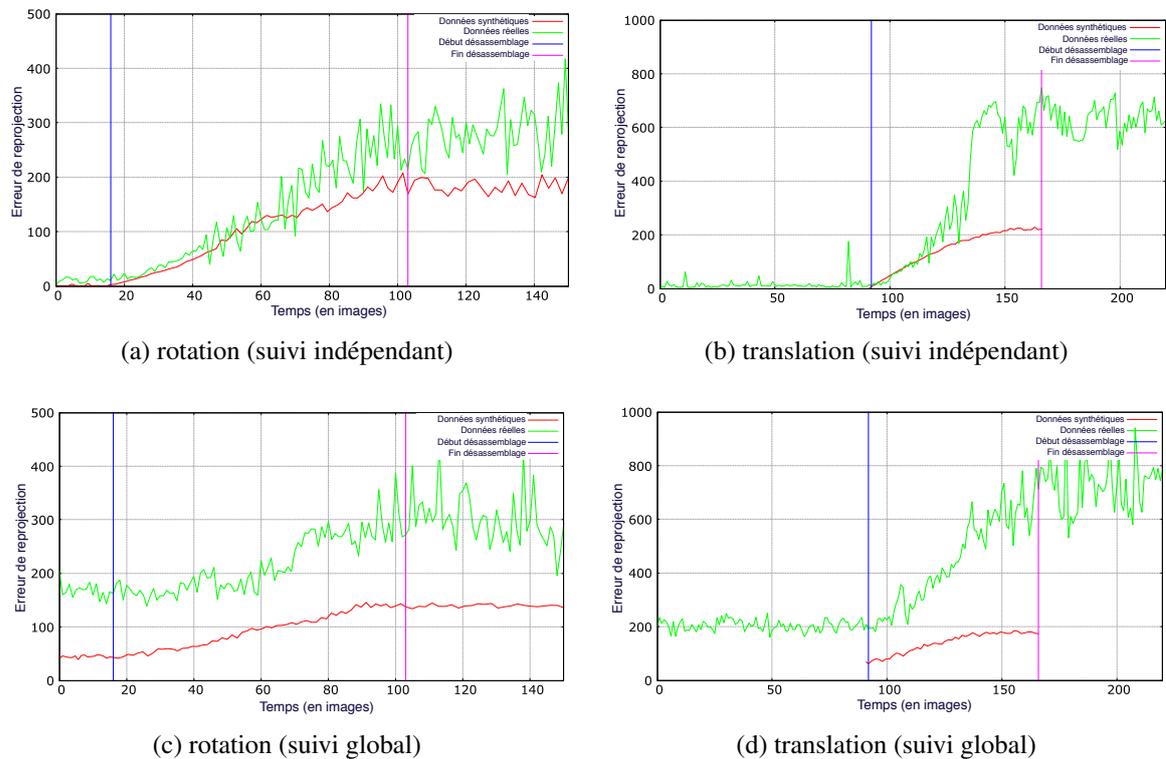


FIGURE 3.8 – Superposition des courbes d'évolution de l'erreur de reprojection déterminée avec des vidéos synthétiques et réelles.

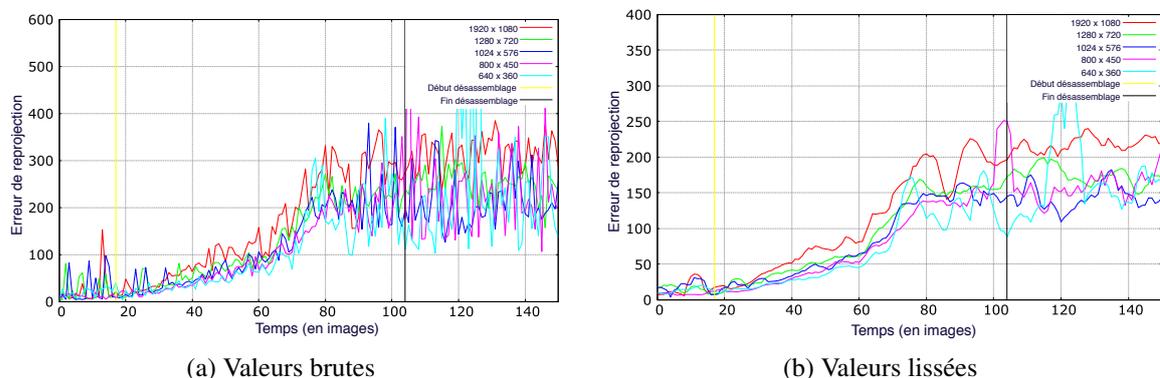


FIGURE 3.9 – Evolution de l'erreur de reprojection suivant la résolution de l'image (cas de la rotation autour de l'axe Z de l'objet 0 dans des conditions réelles avec le suivi multiple).

possible d'établir une approche permettant d'identifier quand on passe d'un état d'assemblage à un état de désassemblage.

Il est à rappeler que, malgré que les expérimentations aient été effectuées sur des objets cubiques texturés, l'hypothèse selon laquelle l'erreur de reprojction permet de caractériser les phases d'assemblage et de désassemblage peut être généralisée avec d'autres formes d'objets. En effet, cette donnée est extraite et détournée après la phase d'estimation de la pose et, par conséquent, après l'extraction des primitives et leur mise en correspondance. En l'associant avec la configuration indépendante des objets intervenant dans l'assemblage, on s'attendra ainsi à observer une variation progressive de l'erreur de reprojction lors de la tâche d'assemblage.

Par la suite, nous établirons une stratégie afin de déterminer à quel moment nous passons d'une phase d'assemblage à une phase de désassemblage et inversement à l'aide de cette métrique.

3.2 Détermination de l'instant d'assemblage ou de désassemblage

Dans cette partie, nous tirerons parti des informations fournies par la courbe de l'erreur de reprojection établie dans la section précédente afin d'établir une approche permettant de détecter automatiquement un changement de contexte dans l'espace de travail de l'utilisateur. Dans cette dernière, nous avons constaté que l'erreur de reprojection telle que nous l'avons définie, augmente lorsque la configuration des pièces à assembler ou à désassembler diffère de sa configuration enregistrée initialement. Ce comportement différent de l'erreur de reprojection entre une phase où les pièces sont assemblées ou désassemblées, s'il est détecté, permettrait de déterminer l'instant où l'on bascule d'une phase à une autre. Ainsi, nous avons exploré un certain nombre d'algorithmes de détection de rupture avant d'établir la nôtre et de comparer ses performances aux autres méthodes. Les algorithmes, généralement hors ligne, seront adaptés afin qu'ils puissent fonctionner en ligne pour nos expérimentations.

3.2.1 Algorithmes de détection de ruptures

Au cours du temps, un signal ou une série chronologique peut se modifier en terme d'amplitude ou de fréquence. Ils peuvent ainsi présenter des points de rupture. Divers algorithmes permettent de les détecter en se basant sur une fonction de décision dépendante des caractéristiques du signal et d'un seuil choisi. Ces algorithmes sont identifiés comme des algorithmes de détection de ruptures et sont souvent utilisés pour la détection de défauts, le diagnostic, la maintenance conditionnelle des procédés industriels, la sécurité des systèmes complexes, le contrôle de qualité ou la prédiction des catastrophes naturelles.

Plusieurs approches peuvent être rencontrées dans la littérature concernant la détection de ruptures :

- celles sans modèle analytique [[Darkhovski, 1994](#), [Steland, 2002](#), [Pawlak et al., 2003](#)]. Il peut s'agir, ici, de s'aider d'une redondance de matérielle pour détecter l'anomalie ou encore de comparer des seuils avec la moyenne ou la mesure des observations ou de réaliser une analyse spectrale des signaux enregistrés [[Verdier, 2007](#)];
- celles basées sur un modèle analytique (ou paramétrique) [[Chen and Patton, 1999](#), [Maquin and Ragot, 2000](#), [Frank, 1996](#), [Isermann, 1997](#)]. Dans cette approche, on suppose que les observations suivent un modèle défini par une ou plusieurs équations ou modèles paramétriques qui dépendent de un ou plusieurs paramètres inconnus. Cette approche est souvent choisie en robotique ou en automatique où des modèles théoriques peuvent être extraits facilement des lois de la physique et des dépendances observées entre les différents signaux entrées-sorties enregistrés ;

- celles relevant de l'intelligence artificielle en se basant sur les réseaux de neurones [Fancourt and Principe, 2000], sur les machines à vecteurs de support (SVM) [Davy and Godsill, 2002], les systèmes d'inférences flous [Zadeh, 1965].

Nous étudierons, dans cette section, les approches avec ou sans modèles analytiques utilisant des données statistiques. Il s'agit d'approches les plus usitées et dont nous nous en inspirerons. Il s'agit d'approches pouvant être réinterprétées assez facilement à des contextes inconnues et dont on peut établir les statistiques au fur et à mesure que les données sont recueillies. Ainsi nous présenterons également l'adaptation des algorithmes hors ligne pour des cas en ligne. Cela nous permettra de les comparer à notre approche.

Par ailleurs, il est à noter que nous ne nous intéresserons qu'à des tests basés sur deux hypothèses uniquement. Dans notre cas, il s'agit des hypothèses que nous définissons de la manière suivante :

$$\begin{cases} H_0 : \text{les pièces sont en phase d'assemblage} \\ H_1 : \text{les pièces sont en phase de désassemblage} \end{cases} \quad (3.9)$$

Chacune des hypothèses est caractérisée par un ou plusieurs paramètres qui peuvent représenter le modèle décrivant des données recueillies avant et après le point de rupture. Celles-ci peuvent correspondre, par exemple, à une moyenne des données et/ou à un écart-type. Ces paramètres seront notées de manière générale θ_0 et θ_1 respectivement pour les hypothèses H_0 et H_1 .

Une extrapolation des théories suivantes pour des hypothèses multiples peut être trouvée dans [Basseville and Nikiforov, 1993, Basseville and Nikiforov, 2002, Tartakovsky and Alexander, 2008, Fillatre, 2011].

a) Algorithmes hors ligne

Les algorithmes de détection de ruptures hors ligne sont utilisés dans une optique de détection et d'isolation de fautes. Il s'agit d'identifier dans un signal quand un système passe d'un état normal à un état anormal [Fillatre, 2011]. Ces algorithmes sont basés sur la théorie des tests d'hypothèse. Les données observées sont mesurées et enregistrées préalablement. Il s'agira ainsi de déterminer à quel moment une rupture a pu se produire dans la succession des données en réalisant des tests d'hypothèse sur chacune des données.

Dans l'ensemble des méthodes définies ci-dessous, nous considérons des échantillons de variables aléatoires indépendantes y_i avec $i \in [1; N]$ suivant une loi de probabilité $p_\theta(y)$ dépendant uniquement du paramètre θ . De manière générale, on considère que ces échantillons suivent une loi normale $\mathcal{N}(\mu, \sigma)$. Les paramètres qui peuvent ainsi changer lors d'une détection de rupture sont soit la moyenne μ , soit l'écart-type σ . Ainsi le paramètre θ correspondra à une des variables précitées.

Par ailleurs, il est supposé qu'avant le changement à l'instant t_0 inconnu, $\theta = \theta_0$ et qu'après t_0 , $\theta = \theta_1 \neq \theta_0$. Les paramètres θ_0 et θ_1 sont connus, par contre t_0 est inconnue.

Ainsi, les hypothèses sont définies de la manière suivante :

$$\begin{cases} H_0 : \theta = \theta_0 \\ H_1 : \theta = \theta_1 \end{cases} \quad (3.10)$$

θ est le paramètre dont dépend la densité de probabilité de la séquence étudiée, c'est-à-dire θ_0 peut représenter soit la moyenne des données avant la rupture t_0 soit son écart-type et θ_1 soit la moyenne des données après la rupture t_0 soit son écart-type. Ces paramètres sont supposés être connus. Dans notre cas de figure, l'hypothèse H_0 correspondra à l'état où nos objets sont liés par une transformation rigide fixe donc au moment où l'erreur de reprojexion reste faible et les objets ne sont pas en mouvement. Par contre, l'hypothèse H_1 correspondra au moment où les paramètres de la transformation vont varier donc à la phase où nos objets seront en mouvement et se désolidariseront.

L'objectif des algorithmes de détection de changement est donc de déterminer avec précision la variable t_0 , instant où le système passe de l'hypothèse H_0 à l'hypothèse H_1 .

i) Carte de contrôle de Shewhart

Dans le cadre de la carte de contrôle de Shewhart [Shewhart, 1930], les échantillons sur lesquels elle est appliquée, sont d'une taille fixée N .

Pour chaque élément y_k de l'échantillon sera testé et classé sous une des hypothèses définies dans l'équation 3.10. La règle de décision est définie comme suit :

$$d = \begin{cases} 0 & \text{si } S_1^k < h \text{ donc } H_0 \text{ est choisi} \\ 1 & \text{si } S_1^k \geq h \text{ donc } H_1 \text{ est choisi} \end{cases} \quad (3.11)$$

avec $S_j^k = \sum_{i=j}^k s_i$ et $s_i = \ln\left(\frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)}\right)$ correspondant au logarithme de vraisemblance.

L'alarme sera déclenchée lorsque

$$t_a = N.\min(k : d_k = 1) \quad (3.12)$$

Si l'on suppose que les échantillons suivent une loi normale $\mathcal{N}(\mu, \sigma)$, une simplification peut être apportée à la carte de contrôle de Shewhart. Cette simplification se base sur l'hypothèse que la rupture dans le signal correspond uniquement à un changement de moyenne avec une même

variance de part et d'autre de la rupture. Ainsi le paramètre θ dont dépendra le test d'hypothèse sera la moyenne μ .

L'alarme est déclenchée, dans ce cas, lorsque :

$$|\bar{y}(K) - \mu_0| \geq k \frac{\sigma}{\sqrt{N}} \quad (3.13)$$

avec $\bar{y}(K) = \frac{1}{N} \sum_{i=N(K-1)+1}^{NK} y_i$, moyenne des y_i sur une fenêtre de taille N .

Dans nos expérimentations, nous avons trouvé que les valeurs suivantes $N = 10$ et $k = 3$ étaient les plus pertinentes.

Pour le cas où la rupture correspond à un changement de variance avec une moyenne constante, il est possible de consulter [Basseville and Nikiforov, 1993] pour obtenir de plus amples détails.

Cette méthode convient très bien lorsque les paramètres θ_0 et θ_1 des tests d'hypothèse sont connus et où seul le moment du changement de caractéristique du signal est inconnu.

ii) Carte de contrôle avec moyenne géométrique glissante

Dans le cas de la carte de contrôle avec moyenne géométrique glissante (*geometric mobile average (GMA)*) [Basseville and Nikiforov, 1993, Roberts, 1959], l'idée est d'attribuer un poids plus important aux données les plus récentes et un poids plus faible aux données les plus anciennes. La fonction de décision est alors définie comme suit :

$$g_k = \sum_{i=0}^{\infty} \gamma_i s_{k-i} \quad (3.14)$$

avec $\gamma_i = \alpha(1 - \alpha)^i$ où $\alpha \in [0 : 1]$ est un facteur d'oubli et $s_i = \ln\left(\frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)}\right)$.

Ainsi, de manière récursive, on aurait :

$$g_k = (1 - \alpha)g_{k-1} + \alpha s_k \text{ avec } g_0 = 0 \quad (3.15)$$

L'alarme est déclenchée lorsque :

$$t_a = \min(k : g_k \geq h) \quad (3.16)$$

avec h un seuil déterminé empiriquement.

iii) Carte de contrôle avec moyenne "finie" glissante

Dans le cadre de la carte de contrôle avec moyenne "finie" glissante (*finite mobile average (FMA)*) [Basseville and Nikiforov, 1993], l'idée est la même que la carte de contrôle avec

moyenne géométrique glissante. Il s'agit d'attribuer un poids plus important aux données les plus récentes. La différence entre ces deux cartes est qu'avec la présente méthode, on tient compte des données incluses dans une fenêtre bien définie alors qu'avec la méthode précédente, on considèrerait toutes les valeurs du signal enregistrées.

Ainsi, la fonction de décision sera définie comme suit :

$$g_k = \sum_{i=0}^{N-1} \gamma_i s_{k-i} \quad (3.17)$$

avec $\gamma_{i \in [1; N-1]}$ un ensemble fini de poids.

L'alarme est déclenchée lorsque :

$$t_a = \min(k : g_k \geq h) \quad (3.18)$$

avec h un seuil déterminé empiriquement.

iv) Algorithme des dérivées filtrées

L'algorithme des dérivées filtrées [Basseville and Nikiforov, 1993] consiste à prendre une décision sur la présence d'une rupture dans le signal en prenant en compte des variations entre deux valeurs successives du signal. Ces variations sont pondérées suivant le même principe que les méthodes précédentes en attribuant un poids plus important aux données les plus récentes. La fonction de décision est définie comme suit :

$$g_k = \sum_{i=0}^{N-1} \gamma_i s_{k-i} \quad (3.19)$$

avec $\gamma_{i \in [1; N-1]}$ un ensemble fini de poids et $s_k = \nabla g_k = g_k - g_{k-1}$

L'alarme est déclenchée lorsque :

$$t_a = \min(k : \sum_{i=0}^{N-1} \mathbb{1}_{(\nabla_{k-i} \geq h)} \geq \eta) \quad (3.20)$$

Cette approche semble intéressante à explorer si le signal reçu est plutôt très lisse le long de l'expérimentation. Son comportement est, cependant, très imprévisible en présence de bruit.

b) Adaptation des algorithmes hors ligne pour une détection en ligne

Dans la section précédente, nous présentions un certain nombre d'algorithmes permettant de détecter les moments de ruptures dans les cas hors ligne. L'intérêt de ces méthodes est qu'elles

peuvent être facilement adaptées à des cas en ligne en choisissant des fenêtrages et des mises-à-jour adéquates. Ainsi, dans cette section, nous reformulons ces algorithmes de les comparer à notre approche.

i) Carte de contrôle de Shewhart

Pour le cas de la carte de contrôle de Shewhart, une décision est prise à chaque instant t à l'aide la formule suivante :

$$g_t = \sum_{i=1}^t s_i \quad (3.21)$$

Même si la fonction de décision pourrait permettre de déterminer le moment de changement de contexte chaque instant, le problème est de déterminer les paramètres θ_0 et θ_1 . Ces derniers tels que formulés dans la section précédente ne sont pas forcément disponibles dans des cas en ligne. Il est possible de considérer trois approches différentes si l'on considère qu'il n'y a que la moyenne qui change entre deux états et que, donc, l'écart-type reste le même. Celles-ci se différencient essentiellement par le fenêtrage considéré. Nous désignons par :

- *Shewart1* : la moyenne μ_0 est déterminée pour les dix premières observations et est conservée comme telle jusqu'à la fin de la séquence. La moyenne μ_1 est déterminée au fur à mesure sur une fenêtre de taille 10 et glisse d'un pas de 1. La décision est prise sur la dernière donnée ajoutée pour le calcul de μ_1 .
- *Shewart2* : la moyenne μ_0 est initialisée sur les dix premières observations et mise à jour en tenant compte en plus de la donnée sortante de la fenêtre de calcul de μ_1 .
- *Shewart3* : la moyenne μ_0 est initialisée sur les dix premières observations et mise à jour en tenant compte en plus de la donnée sortante de la fenêtre de calcul de μ_1 et des 9 valeurs la précédant.

Nous appliquons la règle de décision 3.13 page 99 avec σ l'écart-type des dix premières données (non mis à jour), $k = 3$ et $N = 10$.

ii) Carte de contrôle avec moyenne géométrique glissante

Pour le cas de la carte de contrôle avec moyenne géométrique glissante, la fonction de décision devient :

$$g_t = \sum_{i=0}^t \gamma_i s_{t-i} \quad (3.22)$$

Comme pour la carte de contrôle de Shewhart, les paramètres θ_0 et θ_1 sont déterminés en fonction du fenêtrage et de la mise à jour effectuée. Ainsi nous désignons de la même manière *GMA1*, *GMA2* et *GMA3* suivant respectivement les mêmes conditions que *Shewart1*, *Shewart2* et *Shewart3*.

Dans nos expérimentations, nous appliquons ainsi la fonction de décision 3.22 et la règle de

décision 3.16 page 99 avec $\alpha = 0.7$ et $h = 0.1$.

iii) Carte de contrôle avec moyenne "finie" glissante

Dans le cas de la carte de contrôle avec moyenne "finie" glissante, la fonction de décision est réécrite de la manière suivante :

$$g_t = \sum_{i=0}^{N-1} \gamma_i s_{t-i} \tag{3.23}$$

Comme pour la carte de contrôle de Shewhart, les paramètres θ_0 et θ_1 sont déterminés en fonction du fenêtrage et de la mise à jour effectuée. Ainsi nous désignons de la même manière *FMA1* et *FMA2* suivant respectivement les mêmes conditions que *Shewart1* et *Shewart2*.

Dans nos expérimentations, nous nous appuyons sur la fonction de décision 3.23 et la règle de décision 3.18 page 100 avec $N = 10$, $h = 0.3$ et $\gamma = (\frac{N}{N!}, \dots, \frac{1}{N!})$ (le poids est défini suivant le principe des moyennes pondérées).

c) Algorithmes en ligne avec connaissance a priori des données avant et après rupture

Les algorithmes de détection de rupture en ligne sont basés sur la théorie des tests séquentiels. Une des différences avec les algorithmes hors ligne est que les observations ne sont pas d'une taille fixe et l'objectif dans ces approches est de détecter au plus tôt le moment de rupture dans la séquence $(y_i)_{i \geq 1}$ étudiée.

Comme une observation est collectée à chaque instant t et incluse dans l'ensemble des données, les approches en ligne doivent décider si celle-ci suit la distribution de probabilité P_{θ_0} ou P_{θ_1} . Il s'agit donc de réaliser une "suite de tests d'hypothèse" :

$$(y_1) \mapsto \begin{cases} H_0 & t_0 > 1 \\ H_1 & t_0 \leq 1 \end{cases} \quad (y_1, y_2) \mapsto \begin{cases} H_0 & t_0 > 2 \\ H_1 & t_0 \leq 2 \end{cases} \quad \dots \quad (y_1, y_2, \dots, y_t) \mapsto \begin{cases} H_0 & t_0 > t \\ H_1 & t_0 \leq t \end{cases}$$

avec t_0 l'instant de rupture effectif.

Les approches en ligne consistent à détecter et à isoler le plus rapidement possible les changements après le moment d'apparition du défaut. Il s'agit d'un problème complexe car il s'agit de prendre une décision en ayant le plus petit nombre possible de données vérifiant potentiellement l'hypothèse H_1 afin de réduire le délai de détection.

NB : Les algorithmes séquentiels traitent les données une à une au fur à mesure qu'elles sont

enregistrées. Ils sont théoriquement optimaux. Les algorithmes de type TEF (à taille d'échantillon fixe) regroupent un nombre fixe d'observations et traite le signal en blocs de données successifs. Les avantages de ce type d'algorithmes sont liés au fait qu'ils sont plus adaptés dans la pratique lors de l'enregistrement des données, qu'ils sont faciles à implémenter et ré-adaptables pour des systèmes à structure variable [Fillatre, 2011].

i) Algorithme du CUSUM (Cumulative Sum)

L'algorithme de la somme cumulée CUSUM (Cumulative Sum) [Page, 1954, Wald, 1947] est un algorithme de référence dans la détection de rupture. En général, il est considéré comme optimal dans le sens où il minimise le retard moyen à la détection suivant le critère d'optimalité défini par Lorden [Moustakides, 2002, Fillatre, 2011]. L'idée, dans cette méthode, est de tirer parti de l'évolution de la variable S_k . En effet, cette dernière a, en général, une déviation négative avant le point de changement et une déviation positive après le changement. Elle tient compte de toutes les observations à chaque instant t .

La fonction de décision sera définie comme suit :

$$g_k = S_k - m_k = M_k \tag{3.24}$$

avec $S_k = \sum_1^k s_i$, $m_k = \min_{j \in [1;k]} S_j$ et $M_k = \max_{j \in [k;t]} S_j$.

Elle consiste à calculer la différence entre la valeur de la somme des logarithmes du rapport de vraisemblance et la valeur minimale de l'ensemble des sommes des logarithmes du rapport de vraisemblance calculée jusqu'à l'instant t . Le test est effectué sur l'ensemble des observations enregistrées à chaque instant jusqu'à ce que l'algorithme détecte une rupture dans le signal reçu.

L'alarme est déclenchée lorsque :

$$t_a = \min(k : M_k \geq h) \tag{3.25}$$

avec h un seuil déterminé empiriquement.

Lorsque t tend vers l'infini, la quantité d'information à traiter dans l'algorithme du CUSUM augmente considérablement. Afin de limiter cet accroissement, [Lai, 1998] a proposé l'algorithme WLCUSUM (*window limited CUSUM*). Il s'agit d'appliquer l'algorithme du CUSUM uniquement sur chaque observation extraite d'une fenêtre de temps limité.

Comme, pour les méthodes hors ligne présentées précédemment, les paramètres des distributions des observations avant et après le changement sont connus. Afin de s'affranchir de cette connaissance *a priori* de la forme de la distribution des observations, [Wang et al., 2002] propose d'associer les méthodes non paramétriques de [Darkhovski, 1994] et le CUSUM dans le cadre d'une détection d'anomalies dans un réseau de communication. Tandis que, toujours

dans le même cadre, [Soule et al., 2005] estiment les paramètres de distribution sur une fenêtre de données avant d'appliquer le CUSUM. Cette dernière méthode présente un inconvénient lié au temps de détection des changements qui est relativement "sacrifié" [Lung-Yut-Fong, 2011].

ii) Algorithme à taille d'échantillon fixe (TEF)

L'algorithme à taille d'échantillon fixe [Fillatre, 2011] est souvent utilisé dans les réseaux de télécommunication pour détecter et localiser des anomalies volumiques. Il prend en compte une fenêtre de valeurs du signal à chaque instant t avant de prendre une décision entre deux hypothèses H_0 et H_1 . Cette règle de décision est basée sur le lemme de Neyman-Pearson qui est appliqué sur chaque échantillon de taille N . Elle est définie de manière suivante :

$$d_t = \begin{cases} 0 & \text{si } S_{N(t-1)+1}^{Nt} < h \text{ donc } H_0 \text{ est choisi} \\ 1 & \text{si } S_{N(t-1)+1}^{Nt} \geq h \text{ donc } H_1 \text{ est choisi} \end{cases} \quad (3.26)$$

L'alarme est déclenchée dès que $d_t = 1$.

$$t_a = \min(tN : d_t = 1) \quad (3.27)$$

NB : L'algorithme est théoriquement deux fois plus lent que l'algorithme du CUSUM [Fillatre, 2011]. Par ailleurs, il correspond à la carte de contrôle de Shewhart appliquée sur une fenêtre de taille fixe.

d) Algorithmes en ligne avec connaissance *a priori* uniquement avant la rupture

Les algorithmes présentés précédemment supposent que les paramètres θ_0 et θ_1 sont connus au préalable. Ceci n'est pas toujours le cas. Même s'il n'y a aucune connaissance *a priori* concernant ces deux paramètres, il est parfois possible d'estimer le paramètre θ_0 avec une phase d'initialisation. En revanche, il est alors nécessaire d'introduire des algorithmes prenant en compte le fait que le paramètre θ_1 soit inconnu dans la formulation de la fonction de décision.

i) Algorithme du CUSUM + GLR (*Generalized Likelihood Ratio*)

La règle du GRL (*Generalized Likelihood Ratio*) est introduit par [Lorden, 1971] pour des données indépendantes. Il s'agit d'une variante du CUSUM dans laquelle on suppose que la valeur de la variable θ_1 est inconnue. Cependant celle-ci appartient à un intervalle Θ_1 qui, lui, est connu. Il s'agira, ainsi, d'estimer le maximum de vraisemblance sur Θ_1 et non simplement la vraisemblance θ_1 comme dans le CUSUM classique.

La règle de décision sera définie de la manière suivante :

$$g_k = \max_{j \in [1;k]} (\sup_{\theta_1 \in \Theta_1} S_j^k(\theta_1)) \quad (3.28)$$

L'alarme se déclenchera lorsque :

$$t_a = \min(k : g_k \geq h) \quad (3.29)$$

avec h un seuil.

Un des inconvénients de cette méthode est qu'elle est coûteuse en temps. Cela est lié à la recherche de la valeur maximale du rapport de vraisemblance. Une stratégie pour limiter ce temps est de réaliser cette recherche sur une fenêtre de temps fixée [Verdier, 2007]. Par ailleurs, elle nécessite d'une certaine manière, d'avoir une connaissance *a priori* de l'intervalle dans lequel évolue le paramètre θ_1 , ce qui, dans nos expérimentations, n'est pas forcément accessible. En effet, cette donnée peut énormément varier d'une simulation à une autre. Or choisir un intervalle important pour inclure tous les cas possibles peut ruiner l'effet temps réel souhaité.

ii) Algorithme du CUSUM + MLR (*Mixture Likelihood Ratio*)

La règle du MLR (*Mixture Likelihood Ratio*) [Pollak and Siegmund, 1975] est une variante de la règle du GRL présentée précédemment. Au lieu de chercher à maximiser le rapport de vraisemblance sur un intervalle Θ_1 , il s'agit, dans cette approche, de considérer que l'on a une connaissance *a priori* de la distribution de la variable θ_1 .

La règle de décision devient, dans ce cas :

$$g_k = \max_{j \in [1;k]} \left(\ln \frac{\int_{\Theta_1} \prod_{i=j}^k p_{\theta}(y_i) dG(\theta)}{\prod_{i=j}^k p_{\theta_0}(y_i)} \right) \quad (3.30)$$

avec G la distribution de probabilité de la variable θ_1 .

NBI : Un des paramètres permettant de déterminer si un algorithme en ligne de détection de ruptures est meilleur qu'un autre est de calculer le retard de détection. Celui-ci est défini de manière suivante :

$$\tau = t_a - t_0 + 1 \quad (3.31)$$

avec t_a l'instant de rupture proposé par l'algorithme et t_0 l'instant effectif de rupture de la séquence.

Si l'alarme est déclenchée avant l'instant t_0 , il s'agit d'une fausse alarme. Le deuxième paramètre à prendre en compte est également la durée avant une fausse alarme, c'est-à-dire

entre une détection (quelle qu'elle soit) et une fausse alarme. Un algorithme de détection de rupture en ligne sera d'autant plus performant qu'il arrivera à obtenir un retard de détection très faible et une durée avant une fausse alarme suffisamment grande.

NB2 : Dans le cas où θ_0 est inconnu, on suppose qu'il appartient à un intervalle Θ_0 [Mei, 2006].

3.2.2 KSUM et KSUMratio

Dans les signaux que nous recueillons relatifs à l'évolution de l'erreur de reprojction, si nous considérons que ceux-ci suivent une loi normale $\mathcal{N}(\mu, \sigma)$, alors on observe que la moyenne μ et l'écart-type σ varient de part et d'autre du point de rupture. Or dans les algorithmes de détection présentés précédemment, seul un paramètre est pris en compte dans le changement de variables. Il pouvait s'agir de la moyenne ou bien de l'écart-type du signal. Par ailleurs, la plupart de ces approches, en particulier, celles hors ligne, suppose que ces paramètres sont déjà connus. Cependant, ces hypothèses ne reflètent pas la réalité où nous n'avons pas forcément une connaissance *a priori* de la distribution et de la magnitude des données recueillies. Par exemple, dans nos expériences, chaque signal peut être différent suivant le type de déplacement de nos pièces, la rapidité de l'action effectuée par l'utilisateur, le contexte enregistré au début de la simulation. Ainsi, nous ne pouvons pas définir avec certitude les paramètres avant et après l'assemblage ou le désassemblage des pièces. De plus, nous ne pouvons pas simplement définir un seuil pour tous les cas du fait de la variabilité du signal étudié. Par ailleurs, il est à noter que, théoriquement, les approches citées précédemment détectent plutôt bien un changement lorsque celui-ci apparaît de manière brusque. En revanche, dans nos expérimentations, le changement d'état dans le contexte réel peut se faire de manière graduelle ou abrupte suivant le type de mouvement effectué dans l'environnement réel.

Il s'agit, dans cette partie, de spécifier une stratégie de détection de rupture qui puissent déterminer le moment de l'assemblage ou du désassemblage avec un délai assez court dans n'importe quel scénario où l'erreur de reprojction est enregistrée et sans avoir une connaissance *a priori* des paramètres avant et après le changement.

Nous proposons d'appliquer une fonction, notamment un noyau. Cette procédure permettrait de transformer les données enregistrées de telle sorte que les valeurs obtenues puissent être exploitées directement. En effet, celles-ci deviennent moins bruitées.

Nous présentons deux méthodes aboutissant à des résultats satisfaisant. Celles-ci tiennent compte des valeurs passées sous forme d'une somme à laquelle un noyau gaussien sera appliquée.

La première que nous appellerons KSUM est formulée de la manière suivante :

$$g_k = K\left(\sum_{i=1}^k y_i\right) \quad (3.32)$$

avec $K(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$ un noyau gaussien.

La deuxième formulation consiste à évaluer le rapport entre le KSUM à l'instant t et celui à l'instant $t - 1$. Nous l'appellerons KSUMratio et est formulée de la manière suivante :

$$g_k = \frac{K\left(\sum_{i=1}^k y_i\right)}{K\left(\sum_{i=1}^{k-1} y_i\right)} \quad (3.33)$$

avec $K(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$ un noyau gaussien.

Dans les deux cas, l'alarme est déclenchée lorsque :

$$t_a = \min(k : g_k \leq h) \quad (3.34)$$

avec h un seuil.

En exploitant ces formulations, les courbes résultantes sont un peu plus lisses que les données brutes (cf. fig. 3.10a, 3.10b, 3.10c, 3.10d). De plus, les valeurs avant et après le changement évoluent dans des intervalles disjoints similaires dans la majorité des mouvements de nos expérimentations. Avant le changement, dans le cas du KSUM, les valeurs obtenues sont proches de 0.4 et, dans le cas du KSUMratio, celles-ci sont proches de 1. En revanche, après le changement, dans le cas du KSUM et du KSUMratio, les valeurs obtenues sont proches de 0. Nous pouvons le constater dans les figures 3.10a, 3.10b, 3.10c, 3.10d.

Ainsi comme indiqué par l'équation 3.32, le KSUM peut être interprété comme la probabilité que la somme des erreurs de reprojction jusqu'à l'instant t suit une distribution normale $\mathcal{N}(0, 1)$. Cette probabilité sera grande (proche de 0.4 dans expériences) aussi longtemps que l'erreur de reprojction restera proche de 0. Ceci est le cas tant que les pièces resteront jointes suivant une transformation rigide fixe, donc tant que les pièces ne bougeront pas l'une par rapport à l'autre. Le KSUM va décroître brusquement jusqu'à 0 dès lors que les données y_i seront suffisantes pour perturber de manière considérable leur somme cumulée. Cela a lieu lorsque l'erreur de reprojction augmente et est suffisamment différente de 0. Cette augmentation correspond au moment où la phase de désassemblage apparaît. Cela représente également une modification dans la distribution des données.

Pour le cas du KSUMratio, nous évaluons à quel point cette distribution change entre deux instants successifs t et $t - 1$. Le KSUMratio sera élevé et proche de 1 lorsque les variations pour le KSUM sont faibles. En revanche, il va décroître jusqu'à 0 si le KSUM varie significativement. Ce qui arrive lorsque les pièces sont désassemblées. Toutefois, on peut souligner que la courbe du KSUMratio est un peu plus bruitée que celle du KSUM, mais beaucoup moins que

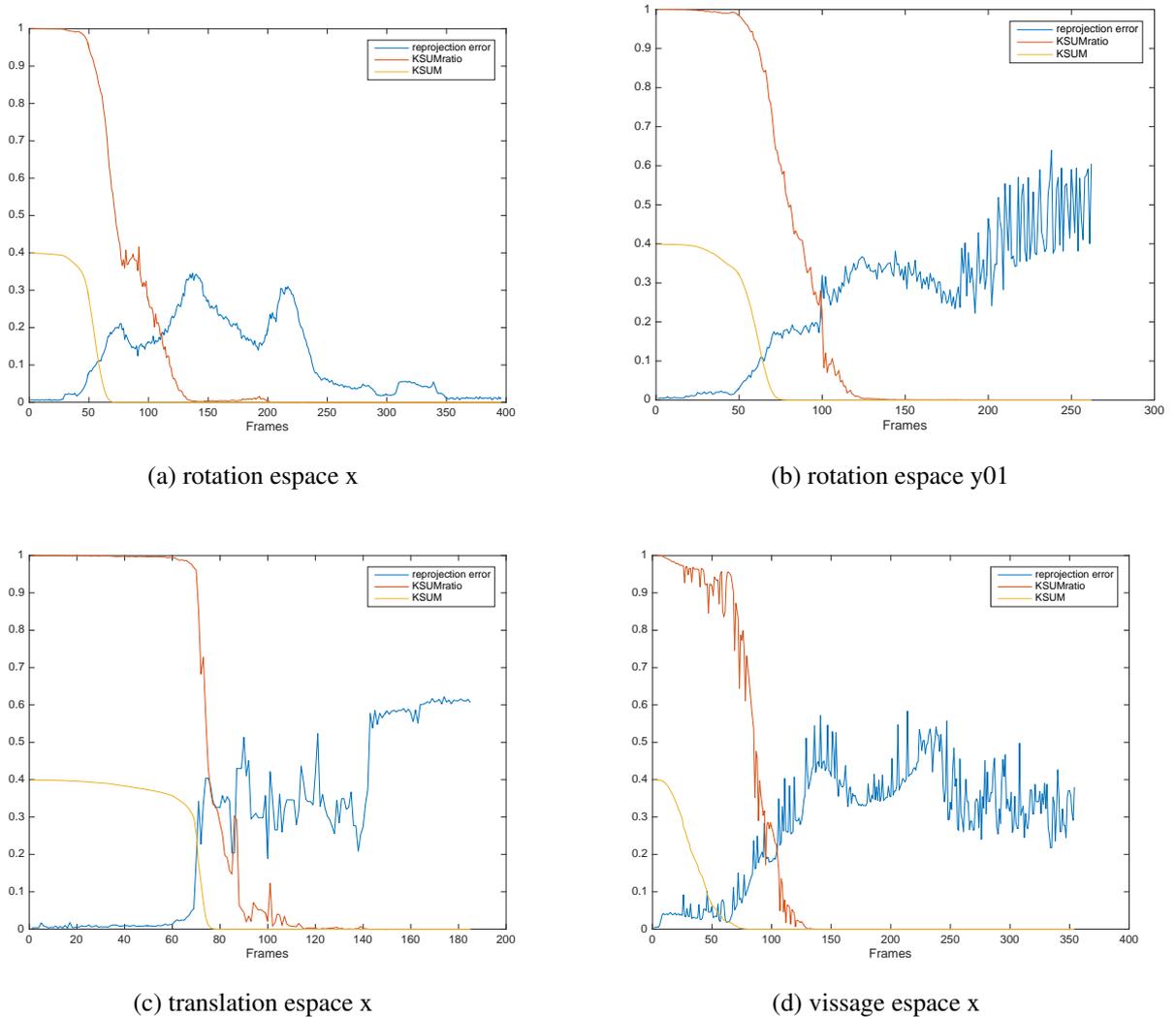


FIGURE 3.10 – Application du KSUM et KSUMratio sur les valeurs de l’erreur de reprojection enregistrées suivant différents mouvements des pièces.

celle de l’erreur de reprojection. Cependant, la présence de bruit dans les valeurs du KSUMratio influence peu la prise de décision effectuée pour signaler le changement de contexte dans l’environnement de travail.

3.2.3 Expérimentation

La donnée que nous étudions est l’erreur de reprojection. Celle-ci est collectée suivant le protocole expérimental défini dans la section 3.1 page 83. Dans le cadre de cette expérimentation, l’erreur de reprojection a été normalisée par rapport aux dimensions du champ de vue afin d’obtenir des valeurs variant entre 0 et 1. Les scénarios de déplacement peuvent correspondre à des rotations ou à des translations simples ou à des combinaisons des deux types de mouvement (voir l’annexe A page 184). Dans nos expérimentations, nous utilisons des vidéos prises dans

des conditions réelles.

Nous avons reformulé les approches hors ligne afin qu'elles puissent fonctionner en ligne dans la section [b \) page 100](#).

Le paramètre θ_0 est défini en se basant sur les $n = 10$ premières valeurs du signal recueillies. Par ailleurs, nous supposons que les données suivent une loi normale $\mathcal{N}(\mu, \sigma)$ de part et d'autre du changement et que l'écart-type σ reste le même.

Comme l'algorithme des dérivées filtrées était très sensible au bruit, les résultats obtenus n'ont pas été pertinents dans la détection du changement de contexte.

Pour l'algorithme du CUSUM, dans les trois cas de fenêtrage, nous avons choisi un seuil $h = 9$.

Pour le cas des algorithmes de CUSUM fusionné avec le GLR ou le MLR (*GLR1*, *GLR2*, *GLR3*, *MLR1*, *MLR2*, *MLR3* suivent le même fenêtrage et mise à jour que les *Shewart1*, *Shewart2*, *Shewart3* définis dans la section [b \) page 100](#)), nous supposons que θ_1 suit une distribution normale $\mathcal{N}(0.32, 0.21)$ et donc que $\Theta_1 = [0.11; 0.53]$. De plus, $h = 8$ pour le GLR et $h = 400$ pour le MLR.

Pour le KSUM, $h = 0.2$ et pour le KSUMratio, $h = 0.9$.

NB : Il est à noter que nous vérifions s'il y a une certaine persistance de la décision dès que l'alarme est déclenchée avant de fixer la décision.

Nous avons ainsi comparé les performances de notre approche à celles des algorithmes présentés précédemment. Ainsi le tableau [3.1 page 111](#) indique les délais enregistrés entre le moment de détection du changement et celui effectif ainsi que le temps d'exécution de ces algorithmes sur ces mêmes données.

NB : Le délai de détection exprimé en secondes est estimé sur la base d'une fréquence moyenne de 24 images par seconde.

Il apparaît que l'algorithme du CUSUM qui est largement populaire est plus précis en termes de délai de détection avec une moyenne de 0.066s et un écart-type de 0.89s. Le FMA qui attribue des poids plus forts aux données les plus récentes tend à établir un instant de détection beaucoup plus tôt que les autres algorithmes. En effet, le spectre de sa gaussienne en termes de délai de détection reste négatif. Le GLR et le MLR sont des algorithmes un peu plus coûteux en temps. Notamment, ils sont à peu près 1000 fois plus lents que KSUM. Ceci s'explique par le fait que ces approches nécessitent d'estimer le paramètre θ_1 qui maximiserait la vraisemblance dans un intervalle bien spécifique ou suivant une distribution bien spécifique. Par ailleurs, ces algorithmes ne fournissent pas des moments de détection de rupture plus précis malgré le temps nécessaire pour la détection. Notre approche KSUM permet de prendre une décision plus rapidement (6.2×10^{-4} s) pour chaque observation acquise. Tandis que le KSUMratio fournit un bon compromis entre la rapidité (6.4×10^{-4} s) et la précision de la détection.

3.2.4 Discussion

Dans cette partie, nous avons introduit le KSUM et le KSUMratio qui permettent d'estimer le moment où nous passons de la phase d'assemblage à la phase de désassemblage. Ces méthodes répondent à une contrainte concernant la méconnaissance que nous pouvons avoir quant aux valeurs de l'erreur de reprojction initialement et à son évolution au cours du temps. Ainsi elles semblent être adéquates expérimentalement pour détecter automatiquement et sans *a priori* les changements de contexte dans l'environnement de travail de l'opérateur.

Nous les avons comparées à un certain nombre d'algorithmes de détection de ruptures assez populaires. Il en ressort que même si le CUSUM fournit des résultats pertinents en termes de délai de détection et même si son temps d'exécution n'est pas si élevé, le KSUMratio apporte une certaine amélioration en termes de temps de calcul et peut être un bon compromis entre le délai de détection et la vitesse d'exécution. Le temps d'exécution est un paramètre important dans notre cadre de travail du fait que l'application de RA, que nous souhaitons implémenter, doit être déployée sur des appareils mobiles dont les capacités de calcul et de stockage sont limitées.

Algo \ Mvt	rotation espace x	rotation espace y01	rotation espace y02	rotation espace z01	rotation espace z02	rotation plan y01	rotation plan y02	rotation plan z	translation espace x	translation espace y	translation espace z	translation plan x	translation plan y	translation plan z	vissage espace x	vissage espace y	vissage espace z	délai moyen (+/- écart-type) (en nb frames)	délai moyen (+/- écart-type) (en s)	temps d'exécution moyen (en 10^{-4} ms)
Shewart1	-5	-22	-22	-1	-5	85	-28	-10	0	20	5	7	23	33	-51	18	4	3 ± 29.59	0.125 ± 1.23	8.5
Shewart2	-5	-22	-22	-2	3	91	-28	17	2	20	5	8	23	-32	-55	16	4	1.35 ± 31.30	0.056 ± 1.30	13.3
Shewart3	-5	-21	-20	-2	3		-18	17	3	20	5	7	24	10	7	16	3	16.59 ± 57.32	0.691 ± 2.38	12.9
GMA1	-5	-24	-1	-2	-2	26	47	-4	-3	18	3	6	26	53	-54	18	1	6.06 ± 24.99	0.252 ± 1.04	26.8
GMA2	-5	-24	-12	-3	2	84	-17	-3	-2	18	3	7	26	53	-55	14	1	5.12 ± 30.37	0.213 ± 1.27	32.6
GMA3	6	-7	-9	46	33		37	22	46	34	29	91	96	32		30	77.88 ± 105.55	3.245 ± 4.40	30.8	
FMA1	-5	-24	-27	-2	-25	29	-29	-14	-2	18	3	7	22	30	-54	18	2	-34.47 ± 12.07	-1.436 ± 0.50	16.0
FMA2	-5	-23	-25	-2	2	-17	-29	-10	1	19	4	7	22	-32	-55	13	1	-36.59 ± 12.00	-1.524 ± 0.5	22.6
CUSUM1	6	-7	-12	42	33	22	11	4	24	30	23	21	32	38	-40	42	24	17.24 ± 21.66	0.718 ± 0.90	16.6
CUSUM2	6	0	-12	42	33	22	15	7	25	30	23	21	32	38	-10	42	24	19.88 ± 16.72	0.828 ± 0.70	19.9
CUSUM3	6	-7	-12	41	28	-5	-23	0	24	30	23	21	14	21	-30	42	24	1.59 ± 21.34	0.066 ± 0.89	18.9
GLR1	15	14	-1	11	-24	29	18	17	11	20	5	10	22	56	2	21	11	113.94 ± 16.01	4.748 ± 0.67	7977.9
GLR2	15	16	23	11	3	97	-29	17	11	20	5	10	22	56	6	18	11	18.35 ± 25.86	0.765 ± 1.08	8033.2
GLR3	26	30	33	24	13	93	0	27	21	29	15	20	32	66	28	38	24	30.53 ± 20.97	1.27 ± 0.87	7719.5
MLR1	-4	-14	-22	0	14	182	115	5	-1	20	6	2	74	24	-9	2	11	23.82 ± 52.63	0.992 ± 2.19	8312.5
MLR2	-4	-13	-22	0	16		119	6	0	20	6	2	91	24	-2	3	11	28.82 ± 63.60	1.201 ± 2.65	8359.0
MLR3	6	-5	-12	9	17		116	15	9	30	16	11	39	33	7	13	16	32.53 ± 58.63	1.355 ± 2.44	7721.2
KSUM	20	18	13	18	-47	-38	-44	-7	12	-44	2	9	-45	-16	-32	18	3	-9.41 ± 26.40	-0.392 ± 1.1	6.2
KSUMratio	20	19	23	17	-47	-34	-47	17	11	-1	5	10	-48	22	2	25	11	0.294 ± 26.49	0.012 ± 1.10	6.4

TABLE 3.1 – Délai de détection par rapport à l’instant de changement effectif. (Algo = *algorithme utilisé*, Mvt = *type de mouvement se déroulant dans la scène*)

3.3 Bilan

Dans ce chapitre, nous nous sommes intéressés à une métrique, notamment l'erreur de reprojection, nous permettant de caractériser des tâches d'assemblage ou de désassemblage suivant la définition que nous avons établie de cette tâche. En l'associant à une configuration adéquate des modèles 3D (indépendant ou global), à des méthodes robustes d'estimation de la pose et d'extraction de points d'intérêt et en définissant cette erreur de manière astucieuse, nous avons montré qu'il était possible d'identifier dans quelle phase nous nous trouvions. Cependant, il s'avère que considérer les objets de manière indépendante favorise une identification plus rapide et potentiellement plus générique des différentes phases.

C'est cette configuration qui a été privilégiée et qui nous a permis d'établir nos méthodes pour identifier le moment où se produit les changements de phases. Nous les avons appelées KSUM et KSUMratio qui consistent principalement à changer l'apparence des courbes afin de déterminer les moments de rupture de façon générique, c'est-à-dire sur n'importe quel type de mouvement en conservant les mêmes paramètres. Nous les avons comparées à d'autres algorithmes de détection de ruptures. Il en ressort que notre approche effectue le processus de décision plus rapidement que les autres algorithmes avec des délais similaires à celui du CUSUM. Ainsi, en associant l'erreur de reprojection et nos approches de détection de l'instant d'assemblage, nous pouvons identifier les phases de maintenance en cours.

Par la suite, nous nous sommes demandés s'il était possible de reconnaître l'action de l'utilisateur et ses erreurs en extrayant des données calculées pour l'estimation de la pose. Pour cela, nous avons défini le terme *assemblage* d'une autre manière afin d'établir notre approche pour identifier le type de liaison en cours et les erreurs se produisant.

Chapitre 4

L'assemblage en tant que structure dynamique

Dans un futur où l'on souhaite être efficace à tout moment et à toute occasion, où l'on souhaite obtenir les informations pertinentes rapidement et automatiquement, les systèmes de réalité augmentée constituent une solution appropriée à cette optique. Cependant, afin de répondre à ce besoin en toute circonstance, il est nécessaire de comprendre ce qui se déroule dans la scène réelle. Il s'agit d'identifier non seulement les objets présents dans l'environnement mais également les mouvements s'y produisant. Cette dernière étape reste encore un défi pour la communauté scientifique.

La plupart des systèmes sont conçus pour des scénarios spécifiques sans défaut. Ceux qui intègrent une reconnaissance d'actions peuvent identifier une action si le mouvement effectué par l'utilisateur correspond au modèle appris par le système. Si cette action n'est pas identifiée, l'utilisateur devra simplement recommencer l'action jusqu'à ce que l'identification soit celle attendue. Ce type d'approche se base sur un principe de reconnaissance d'actions de manière globale centrée sur l'utilisateur. L'analyse et l'identification se réalise une fois que l'utilisateur aura effectué le geste complètement. De ce fait, il ne permet pas de savoir si localement, c'est-à-dire durant l'exécution de la tâche, cette dernière intègre des erreurs et si les pièces manipulées ont été correctement placées dans l'environnement de travail de l'opérateur.

Notre objectif est d'établir un système avec tolérance de défauts non pas centré sur l'utilisateur mais sur les objets manipulés. Ce type de système se déroule en deux temps [Blanke et al., 2006] :

- une phase de détection, d'identification et d'enregistrement des défauts,
- une phase de reconfiguration du système de contrôle pour s'adapter au contexte présentant un défaut afin que le système puisse continuer effectuer son rôle, c'est-à-dire celui de contrôler la bonne exécution de l'ensemble du processus.

La première partie du processus consiste à mesurer les sorties, c'est-à-dire les paramètres

de mouvement relatifs des pièces et de vérifier leur consistance par rapport au modèle prévu. Cette étape permettra d'identifier les défauts et de rectifier les données utilisées pour le reste du processus de contrôle. Cette correction est ainsi effectuée lors de la deuxième phase du système.

Dans cette partie, nous considérons deux types de défauts :

- les erreurs liées à l'enregistrement ou à la détermination des paramètres de mouvement.
- les erreurs liées à la mauvaise exécution de la tâche d'assemblage.

Ainsi, contrairement à la section précédente, dans cette partie, nous définissons une tâche d'assemblage de la manière suivante :

Définition 2. *Un assemblage se produit lorsque la liaison mécanique existante entre deux pièces conserve le même nombre de degrés de liberté suivant les mêmes axes.*

L'utilisateur sera toujours en phase d'assemblage si les pièces qu'il manipule vérifient les paramètres de la liaison attendue. En revanche, un désassemblage aura lieu si un degré de liberté supplémentaire apparaît dans la liaison considérée.

De plus, dans le domaine industriel, les modèles CAO des pièces mécaniques manipulés par l'opérateur sont souvent disponibles. Ces modèles représentent non seulement la forme 3D des objets, mais également les relations mécaniques entre les différentes pièces. On y retrouve donc les contraintes de mouvement présentes lors d'un assemblage. Ces relations nous permettront de vérifier la bonne exécution des actions.

Toutefois, pour identifier les actions en cours dans la scène réelle, nous utilisons la matrice de transformation qui représente la relation existante entre deux pièces, par conséquent le mouvement relatif des pièces. Cependant cette matrice n'est pas exploitable et interprétable directement, il est nécessaire de la convertir sous une forme exploitable par notre système, notamment sous forme d'une représentation axe-angle. Ainsi nous spécifierons dans un premier temps quelles sont les corrections effectuées afin de récupérer les données concernant le mouvement. Nous analyserons leur évolution au cours du temps suivant deux configurations différentes pour lesquelles nous indiquerons celle qui répond au mieux à une bonne analyse du mouvement. Nous proposerons une méthode afin de rectifier les défauts liés à l'enregistrement de ces données et enfin la métrique pour reconnaître les défauts liés à la mauvaise exécution d'une action par l'utilisateur.

4.1 Détermination de la matrice de transformation

La matrice de transformation entre deux objets contient des informations liées à la rotation et à la translation d'un objet par rapport à un autre. Nous nous attendons en analysant l'évolution du vecteur de rotation, de l'angle de rotation ainsi que du vecteur de translation à retrouver les mouvements effectués entre les objets. Cette matrice de transformation relative se base sur

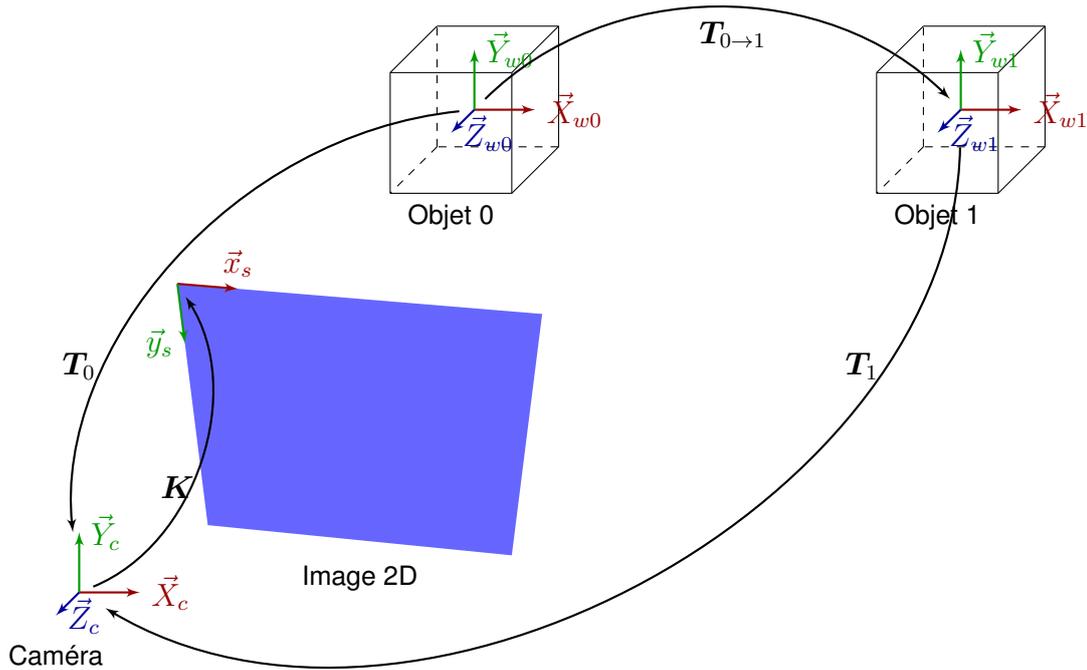


FIGURE 4.1 – Schéma illustrant la détermination de la matrice de transformation en général.

les matrices de projection de chaque pièce comme indiqué par l'équation 3.3 page 87. Nous reformulons ainsi de la manière suivante pour deux objets i et j distincts.

$$\mathbf{T}_i = \mathbf{T}_{i \rightarrow j} \mathbf{T}_j \quad (4.1)$$

L'évolution des paramètres de translation et de rotation entre les objets inclus dans la matrice de transformation sera analysée afin d'identifier les caractéristiques des mouvements ayant lieu dans la scène. La décomposition de la matrice de rotation sera sous forme d'une représentation axe-angle. Il s'agit d'un format qui est souvent utilisé en mécanique pour déterminer l'axe de rotation instantanée. De plus, cette représentation permet une meilleure interprétation des mouvements entre les pièces. Elle peut permettre un passage plus direct à la partie rotationnelle des torseurs cinématiques représentant les liaisons mécaniques entre deux pièces.

Afin de pouvoir exploiter correctement les données concernant la rotation, nous présentons notre méthode pour rectifier ces paramètres afin d'éviter le phénomène de *gimbal lock* ainsi que les instabilités numériques qui surviennent lorsque les angles ont des valeurs proches de $0[\pi]$ lorsque l'on ne prend pas en compte la continuité du mouvement.

4.1.1 Gestion du *gimbal lock* et des instabilités numériques

La matrice de rotation est représentée sous forme d'un vecteur de rotation $\vec{\omega}$ et d'un angle θ .

Cependant, comme il n'y a, en théorie, aucune discontinuité entre les angles et les vecteurs de rotation au cours du temps et comme les données traitées sont bruitées, nous proposons d'introduire des seuils permettant de garder une cohérence dans les mouvements enregistrés. Ainsi, le vecteur de rotation et l'angle seront déterminés de la manière suivante.

A l'instant t , nous supposons que l'angle θ est dans l'intervalle $[0; \pi]$. Dans cet intervalle, la fonction \cos est bijective et nous sommes assurés quant à l'unicité de l'angle avec sa fonction inverse.

Ainsi sachant que, la matrice de rotation

On a alors :

$$\theta = \arccos \left(\frac{(r_{11} + r_{22} + r_{33}) - 1}{2} \right) \quad (4.2)$$

avec la matrice de rotation

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (4.3)$$

Les fragments peuvent faire autant de demi-cycles de rotation que nécessaire pour une tâche d'assemblage spécifique. Donc, en principe, l'angle de rotation α_t peut varier entre 0 et ∞ . Ainsi,

$$\alpha_t = s \theta + 2 k \pi \quad (4.4)$$

Avec

- $k = 0$ à l'instant $t = 0$. Il est incrémenté à chaque fois qu'un objet fait un cycle complet de rotation.
- le paramètre s est défini comme suit,

$$s = \begin{cases} -1 & \text{si } \alpha_t[2\pi] \in [-\pi; 0[\\ 1 & \text{si } \alpha_t[2\pi] \in [0; \pi[\end{cases} \quad (4.5)$$

Comme des instabilités numériques et un *gimbal lock* peuvent survenir aux abords des angles 0 et π , nous introduisons un seuil ϵ afin de déterminer le vecteur de rotation. Ainsi,

si $\cos(\alpha_t) \in]-1 + \epsilon; 1 - \epsilon[$ (on rappelle que dans cet intervalle $\sin(\alpha_t) \neq 0$),

$$\begin{cases} \omega_{x_t} = \frac{r_{32_t} - r_{23_t}}{2 \sin(\alpha_t)} \\ \omega_{y_t} = \frac{r_{13_t} - r_{31_t}}{2 \sin(\alpha_t)} \\ \omega_{z_t} = \frac{r_{21_t} - r_{12_t}}{2 \sin(\alpha_t)} \end{cases} \quad (4.6)$$

Dans nos expérimentations, ce seuil a été fixé à 0.02 autrement dit ces perturbations interviennent lorsque l'angle de rotation est dans les intervalles suivants $[0, 0.2]$ et $[2.94; \pi]$.

Si $\cos(\alpha_t) \in [-1; -1 + \epsilon] \cup [1 - \epsilon; 1]$, on tient compte des paramètres de rotation aux instants précédents notamment à $t - 1$ et à $t - 2$. En effet, dans cet intervalle une variation du sens du vecteur de rotation et de l'angle peut se produire. Il est alors nécessaire de déterminer le paramètre s . Celui-ci nous permettra d'identifier le cadran dans lequel devrait se retrouver l'angle de rotation. (cf fig. 4.2 page suivante)

Concernant le vecteur de rotation, afin de garder une continuité dans le mouvement potentiel de ce vecteur, nous décidons de garder les mêmes paramètres que le vecteur précédent s'il existe.

$$\begin{cases} \vec{\omega}_t = \vec{\omega}_{t-1} \text{ si } \vec{\omega}_{t-1} \exists \\ \vec{\omega}_t = \vec{0} \text{ sinon} \end{cases} \quad (4.7)$$

Si $\cos(\alpha_{t-1}) \in [-1; -1 + \epsilon]$ et $\cos(\alpha_{t-2}) \in [-1; 0]$ (zone en rouge dans la figure 4.2 page suivante), nous assurons que α_{t-1} est un minimum local pour la fonction \cos entre les valeurs α_t et α_{t-2} , autrement dit

$$\begin{cases} \frac{\partial \cos(\alpha_t)}{t} \Big|_{t-1} > 0 \\ \frac{\partial \cos(\alpha_t)}{t} \Big|_{t-2} < 0 \end{cases} \quad (4.8)$$

Par ailleurs, pour tenir compte du demi-cycle de rotation, il faudrait incrémenter le paramètre k suivant les conditions suivantes :

$$\begin{cases} \frac{\partial \alpha_t[2\pi]}{t} \Big|_{t-1} < 0 \\ \frac{\partial \alpha_t[2\pi]}{t} \Big|_{t-2} > 0 \end{cases} \quad (4.9)$$

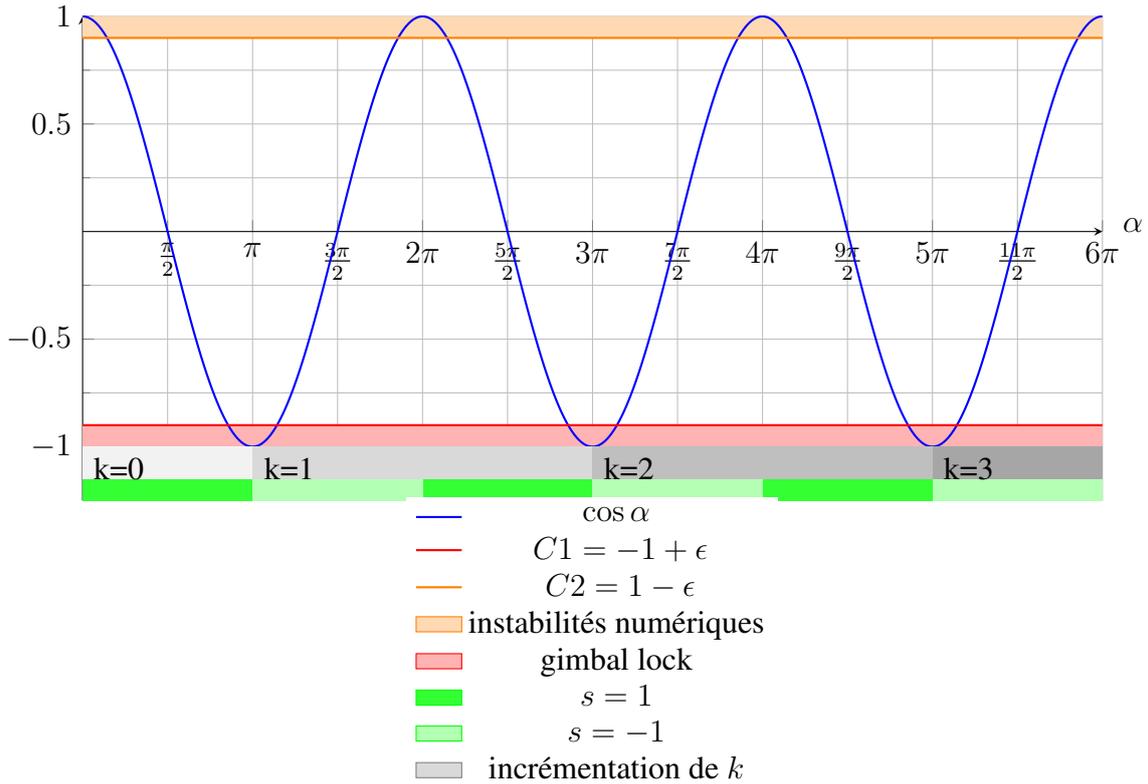


FIGURE 4.2 – Graphe montrant les zones de perturbations avec la fonction cos.

Ainsi si les conditions 4.8 et 4.9 page précédente sont réunies, alors $s = -1$ et k est incrémenté.

En revanche, si $\cos(\alpha_{t-1}) \in [1 - \epsilon; 1]$ et $\cos(\alpha_{t-2}) \in [0; 1]$ (zone en vert dans la figure 4.2), nous nous assurons que α_{t-1} est un maximum local pour la fonction \cos entre les valeurs α_t et α_{t-2} , autrement dit

$$\begin{cases} \frac{\partial \cos(\alpha_t)}{\partial t} \Big|_{t-1} < 0 \\ \frac{\partial \cos(\alpha_t)}{\partial t} \Big|_{t-2} > 0 \end{cases} \quad (4.10)$$

Ainsi si la condition 4.10 est remplie, le paramètre $s = 1$.

La méthode présentée ci-dessous nous permet ainsi de gérer le *gimbal lock* et certaines instabilités numériques pouvant survenir lors de la simulation.

Par la suite, nous avons analysé les courbes obtenues suivant le mouvement relatif considéré :

- soit le mouvement relatif considéré correspond à celui d'une pièce mobile par rapport à la position d'un objet de référence statique,
- soit le mouvement relatif considéré correspond à celui de la pièce mobile à un instant t

par rapport à sa position à l'instant initial.

Il s'agit, ici, dans cette analyse d'identifier la meilleure configuration afin de reconnaître non seulement le mouvement et la trajectoire de la pièce mobile mais également le mouvement relatif de cette pièce par rapport à celle de référence.

4.1.2 Mouvement par rapport à un objet de référence statique

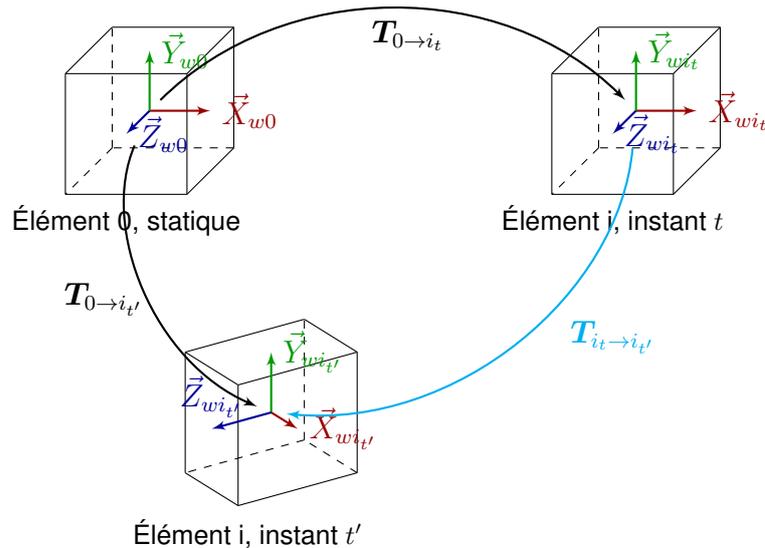


FIGURE 4.3 – Schéma représentant les relations entre chaque pièce d'un objet à assembler ou à désassembler au cours du temps.

Dans cette configuration, nous supposons qu'une pièce de l'objet à assembler ou à désassembler est statique au cours du temps (cf fig. 4.3). Cette partie de l'objet correspondra à notre référence. Il sera lié au repère monde. Ainsi, à chaque instant t , on détermine les relations $T_{0 \rightarrow i_t}$ (cf équations 4.1 page 115) qui est décomposée en matrice de rotation R et en vecteur de translation t .

Les expérimentations sont effectuées, dans un premier temps sur des vidéos synthétiques représentant des mouvements canoniques comme une translation (fig. 4.4a page 123) ou une rotation (fig. 4.5a page 124). Dans un deuxième temps, nous avons analysé nos rectifications sur des vidéos de ces mêmes mouvements pris dans des conditions réelles afin de vérifier leur adéquation avec les mouvements pris dans des conditions idéales. Nous effectuons, comme dans la section 3.1 page 83, des tests en suivant le même protocole expérimental. On trouvera dans l'annexe B page 190 des mouvements plus variés et de plus amples résultats.

L'analyse des variations des paramètres de rotation et de translation montrent une cohérence avec le mouvement en cours dans la scène aussi bien avec les mouvements pris dans des conditions idéales (donc des vidéos synthétiques) et ceux pris dans des conditions réelles. En effet, pour le cas d'une translation suivant l'axe X de l'objet de référence (fig. 4.4a page 123), on

observe bien, durant la phase d'assemblage, une absence de rotation (fig. 4.4d page 123, 4.4e, 4.4f, 4.5g). Dans le cas idéal (fig. 4.4d page 123, 4.4f page 123), l'angle de rotation et le vecteur de rotation reste pratiquement nulle et les courbes obtenues ne sont pas très bruitées. Ceci permet d'interpréter les mouvements assez rapidement. En revanche, dans le cas réel, les courbes représentant la rotation (fig. 4.4e page 123, 4.4f page 123), sont assez bruitées. Une analyse des données de translation et de rotation à un instant t précis, ne permettrait pas toujours d'identifier de manière certaine le type d'assemblage, c'est-à-dire de liaison mécanique détectée à ce moment. Une vue globale de ces courbes permet de nous conforter quant à cette absence de rotation durant la phase d'assemblage. Par ailleurs, concernant le vecteur de translation, théoriquement, on s'attend à ce qu'il y ait une augmentation des valeurs de ce vecteur que suivant l'axe X vu que l'objet en mouvement s'éloigne de l'objet de référence suivant l'axe X . Les figures 4.4h page 123 et 4.4i page 123 montrent l'évolution du vecteur de translation représentent le sens de variations de ce vecteur attendu. De plus, une représentation des vecteurs de translation sous forme de trajectoire (fig. 4.4b page 123, 4.4c page 123) permet de vérifier que le mouvement suit bien une trajectoire rectiligne dans l'espace 3D durant la phase d'assemblage quelles que soient les conditions d'enregistrement de nos vidéos. Une représentation sous forme de trajectoire ne permet pas de vérifier si la pièce subit une rotation.

Nous pouvons faire les mêmes observations dans le cas d'une rotation suivant l'axe Y de l'objet de référence (fig. 4.5a page 124). Théoriquement, on devrait s'attendre à :

- une augmentation de l'angle de rotation,
- à ce que l'axe Y soit reconnu comme vecteur de rotation,
- à une légère augmentation des valeurs du vecteur de translation suivant l'axe X , puis à une diminution de celles-ci jusqu'à atteindre 0 puisque la trajectoire de l'objet en mouvement forme un arc de cercle sur le plan XZ ,
- à une augmentation des valeurs du vecteur de translation suivant l'axe Z et
- à ce que les valeurs du vecteur de translation suivant l'axe Y soit stables durant la phase d'assemblage,
- à ce que la trajectoire corresponde à un arc de cercle sur le plan XZ .

Les différents points précités sont observés aussi bien dans le cas idéal que dans le cas réel (fig. 4.5 page 124). Cependant, comme pour le cas précédent, les courbes dans le cas réel sont assez bruitées. La présence de bruit peut potentiellement induire une mauvaise interprétation de la liaison en cours.

4.1.3 Mouvement par rapport à la position initiale de l'objet en mouvement

Dans cette section, il s'agit d'explorer les paramètres de rotation et de translation de la matrice de transformation $T_{i_t \rightarrow i_{t'}}$. Nous avons analysé l'évolution des mouvements de l'objet en mouvement par rapport à sa configuration initiale. Cette matrice est déterminée de la manière suivante :

$$T_{0 \rightarrow i_{t'}} = T_{0 \rightarrow i_t} T_{i_t \rightarrow i_{t'}} \Rightarrow T_{i_t \rightarrow i_{t'}} = T_{0 \rightarrow i_t}^{-1} T_{0 \rightarrow i_{t'}} \quad (4.11)$$

Contrairement à la section précédente, ici, l'objet de référence est l'objet en mouvement à l'instant initial. On analyse les variations des vecteurs de rotation et de translation selon la position initiale de l'objet.

Les expérimentations sont toujours effectuées suivant le même protocole expérimental qu'à la section précédente [3.1 page 83](#). Nous avons effectué, dans un premier temps, des tests sur les mêmes vidéos synthétiques avec de mouvements canoniques entre les objets (fig. [4.6a page 125](#), [4.7a page 126](#)).

Nous observons toujours une cohérence entre l'évolution de courbes de mouvement (angle, vecteur de rotation et vecteur de translation) et les mouvements se déroulant sur la scène. On observe bien une trajectoire rectiligne dans le cas d'une translation (fig. [4.6e page 125](#)) et un arc de cercle dans le cas de notre rotation (fig. [4.7e page 126](#)). Cette observation est plus détaillée avec les courbes des vecteurs de translation (fig. [4.6d page 125](#), [4.7d page 126](#)). Les rotations correspondent bien à une absence de rotation pour la translation (fig. [4.6b page 125](#), [4.6c page 125](#)) et à une rotation suivant l'axe Y de l'objet en mouvement dans le cas de la rotation (fig. [4.7b page 126](#), [4.7c page 126](#)).

Nous pouvons également observer des similitudes entre les rotations obtenues dans le cas où l'on récupère les données de mouvement par rapport à un objet de référence. Cependant il y a un décalage dans le cas du vecteur de translation dû à la position de l'objet en mouvement par rapport à l'objet de référence. Les similitudes dans les paramètres de rotation et la trajectoire sont liées au fait que les repères des deux objets soient les mêmes mais avec un décalage dans les positions. Si les repères étaient dirigés différemment les paramètres de rotation et de translation seraient différents.

L'idée dans l'exploration de cette configuration était d'analyser le mouvement de l'objet en mouvement dans un même repère, notamment le repère attaché à la position initiale de l'objet en mouvement et de le reconnaître dans l'espace de travail. Cependant, les courbes obtenues ainsi que la trajectoire peuvent devenir difficile à représenter et à interpréter, lorsque le mouvement résultant devient complexe.

4.1.4 Discussion

Nous avons exploré deux configurations différentes afin d'identifier ce qui se passe dans l'espace de travail de l'utilisateur. La première consiste à analyser les paramètres de mouvement de l'objet en mouvement par rapport à un objet de référence statique. Elle nous permettrait non seulement de reconnaître le mouvement de l'objet en mouvement mais également la liaison mécanique devant apparaître entre les objets lors de l'assemblage. La deuxième permettra de reconnaître uniquement le mouvement de l'objet en mouvement pouvant avoir lieu lors de l'assemblage sans avoir besoin d'avoir à récupérer les matrices de transformation d'un objet de référence statique le long de l'action.

Cependant cette deuxième configuration peut être difficile à interpréter si le mouvement devient complexe. De plus, elle suppose que la matrice de transformation initiale extraite de l'objet en mouvement est suffisamment fiable pour la définir comme référence lors de l'exécution de la tâche d'assemblage. Or en prenant en compte la première configuration, nous pouvons soit nous baser sur des matrices de transformations extraites à l'instant initial en vérifiant leur adéquation avec notre base de connaissance lorsque les pièces sont toutes statiques, soit nous baser directement sur la liaison mécanique définie dans notre base de connaissance. De ce fait, nous avons opté pour la première configuration afin d'identifier les liaisons mécaniques entre les objets et afin de vérifier la bonne exécution de l'action.

Nous avons, dans un premier temps, réalisé des tests sur des vidéos synthétiques, puis sur des vidéos prises dans des conditions réelles. L'analyse effectuée reste visuelle. Celle-ci nous permet de vérifier si les courbes obtenues seront suffisamment exploitable pour identifier les liaisons mécaniques entre les pièces. Or il s'avère que les courbes obtenues dans les conditions réelles sont beaucoup plus bruitées que dans des conditions idéales (vidéos synthétiques). Cela peut être dû non seulement à la présence de mouvement de la caméra ou au flou du mouvement présents dans les conditions réelles, mais également à la détection des points d'intérêts nécessaires pour l'estimation de pose, à la mise en correspondance, au processus d'estimation de la pose incluant le RANSAC, en bref, aux méthodes utilisées lors du pipeline de l'estimation de la pose. Le bruit présent peut induire une mauvaise interprétation des mouvements relatifs des pièces. Nous nous sommes alors intéressés à une procédure permettant de correctement reconnaître ces mouvements et identifier les potentielles erreurs commises par l'utilisateur lors de la tâche d'assemblage sans que ce bruit n'influence trop cette reconnaissance. Vu que plusieurs éléments interviennent dans la procédure d'estimation de la pose, nous nous sommes demandé lesquels pouvaient avoir une plus grande influence sur la pose.

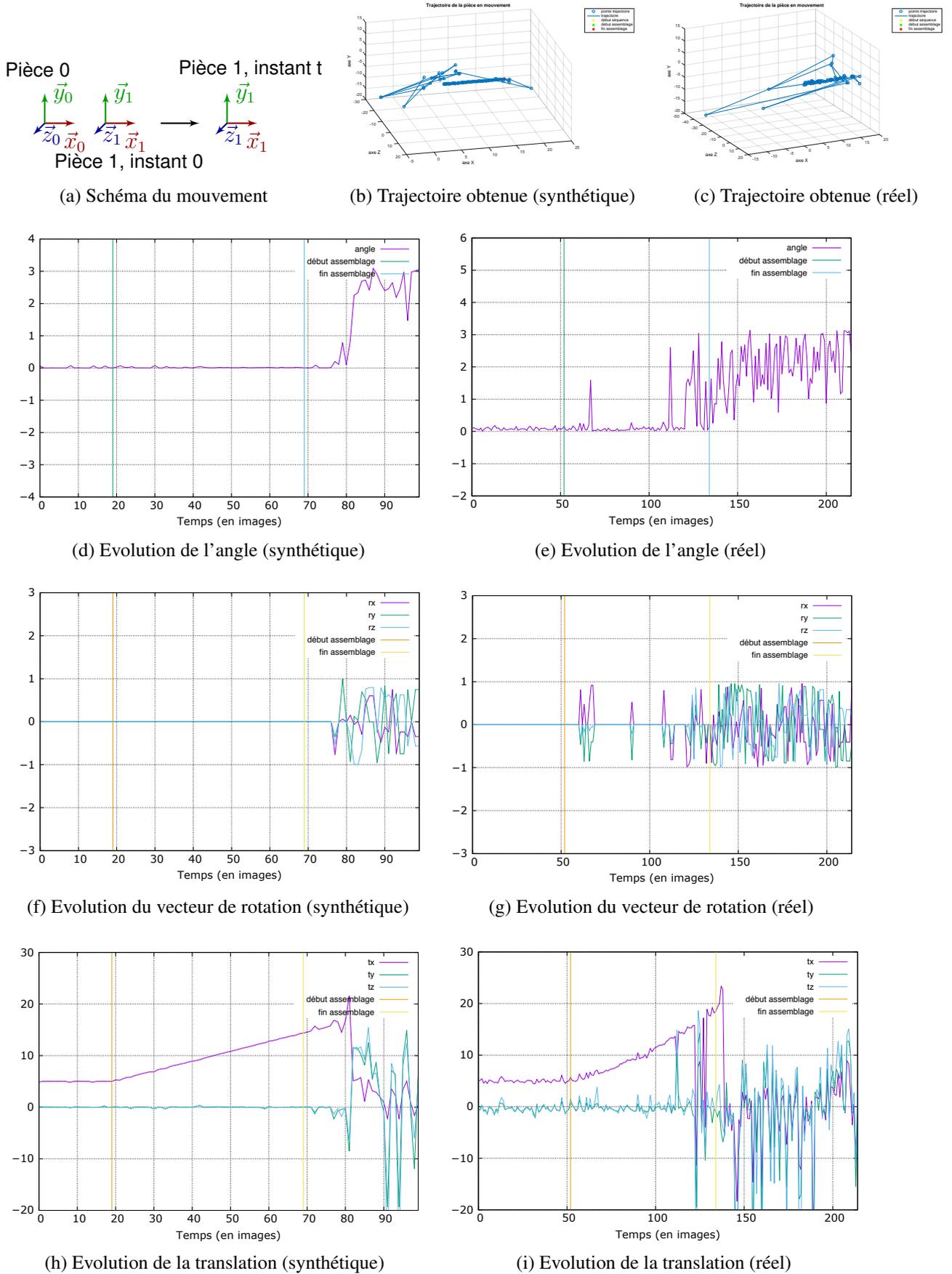


FIGURE 4.4 – Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une translation selon l'axe X de l'objet 0 en considérant le mouvement relatif entre l'objet en mouvement et l'objet de référence.

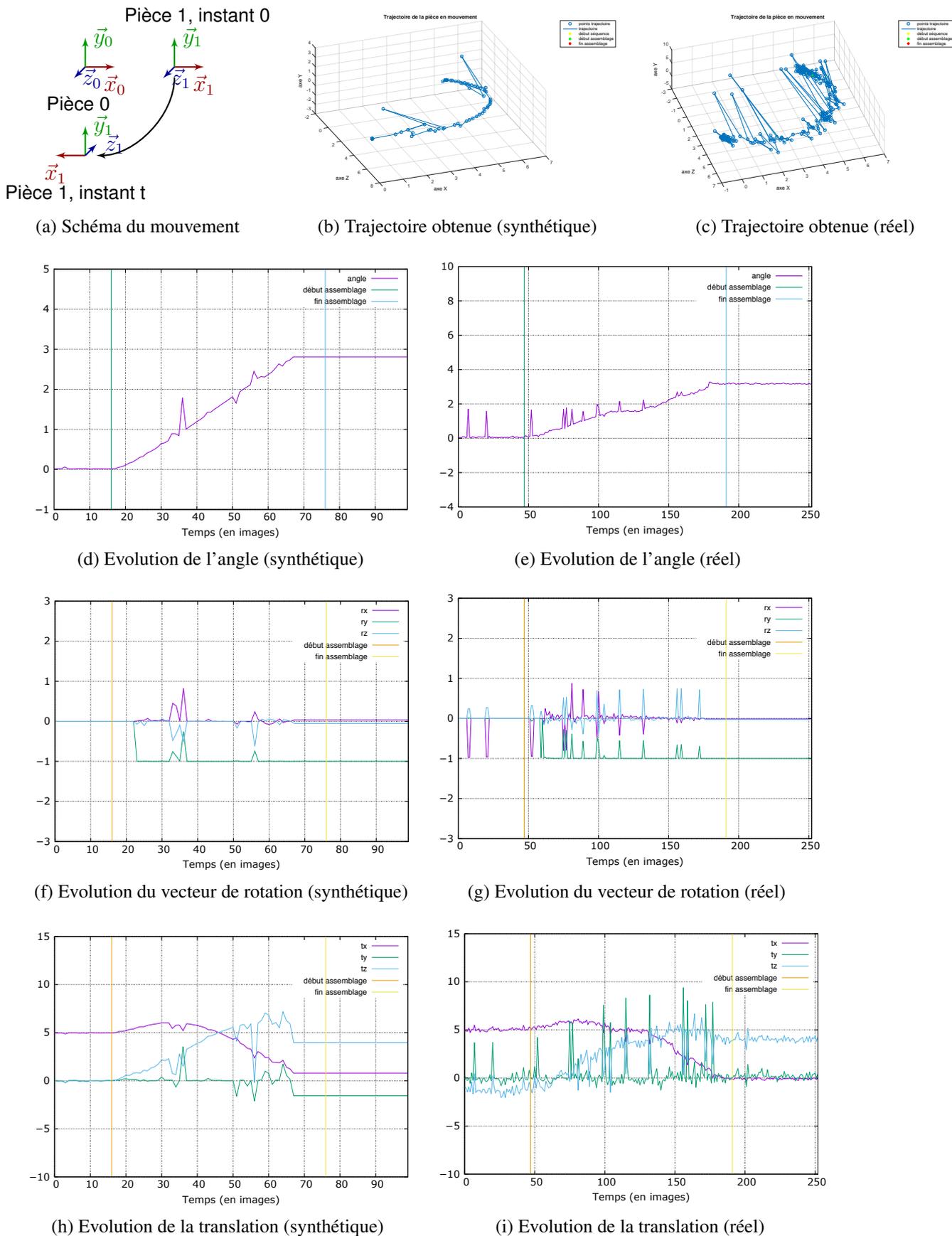


FIGURE 4.5 – Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 en considérant le mouvement relatif entre l'objet mobile et l'objet de référence. (pièces posées sur un plan)

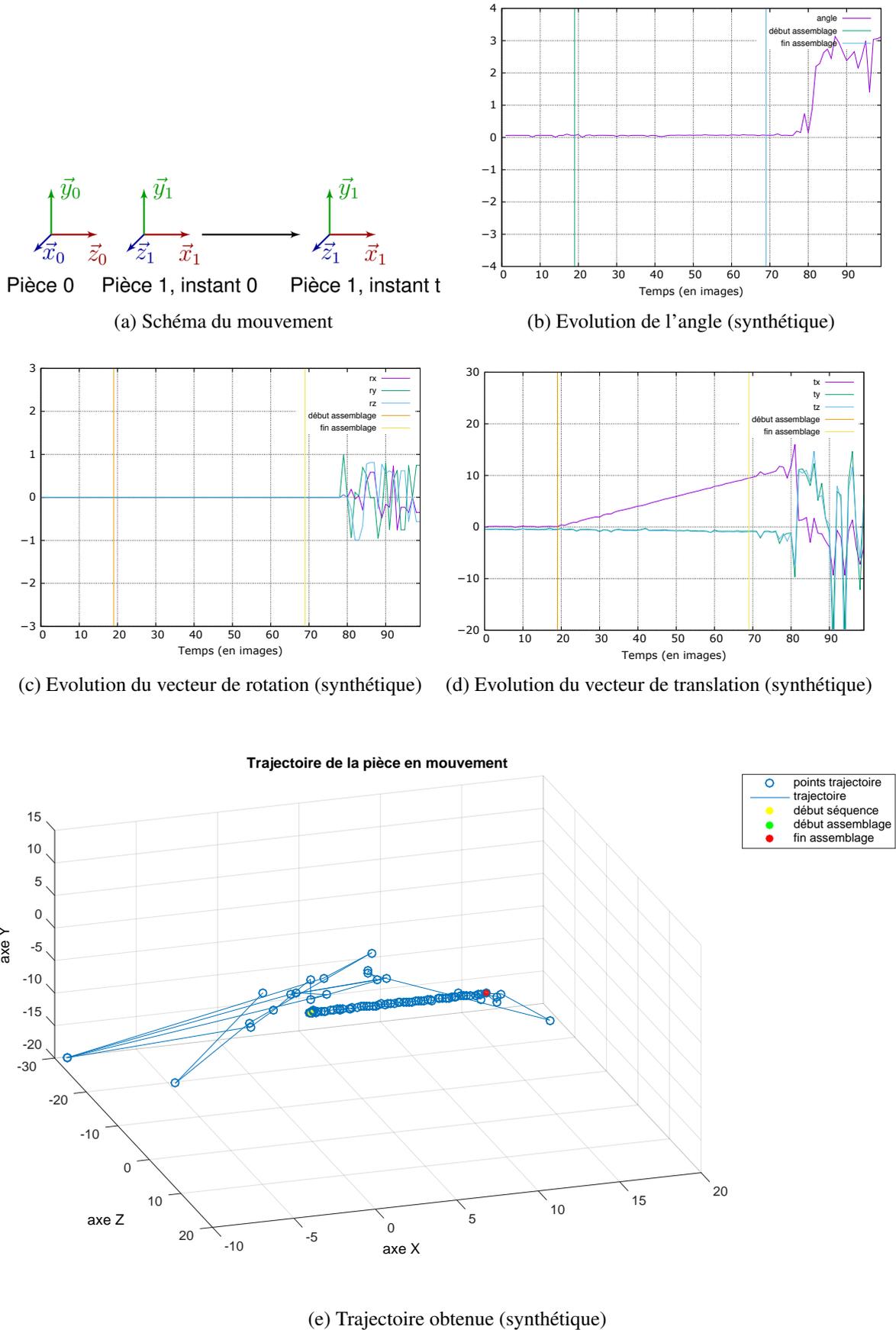


FIGURE 4.6 – Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une translation selon l'axe X de l'objet 0 en considérant le mouvement relatif de l'objet mobile à l'instant t par rapport à sa position initiale.

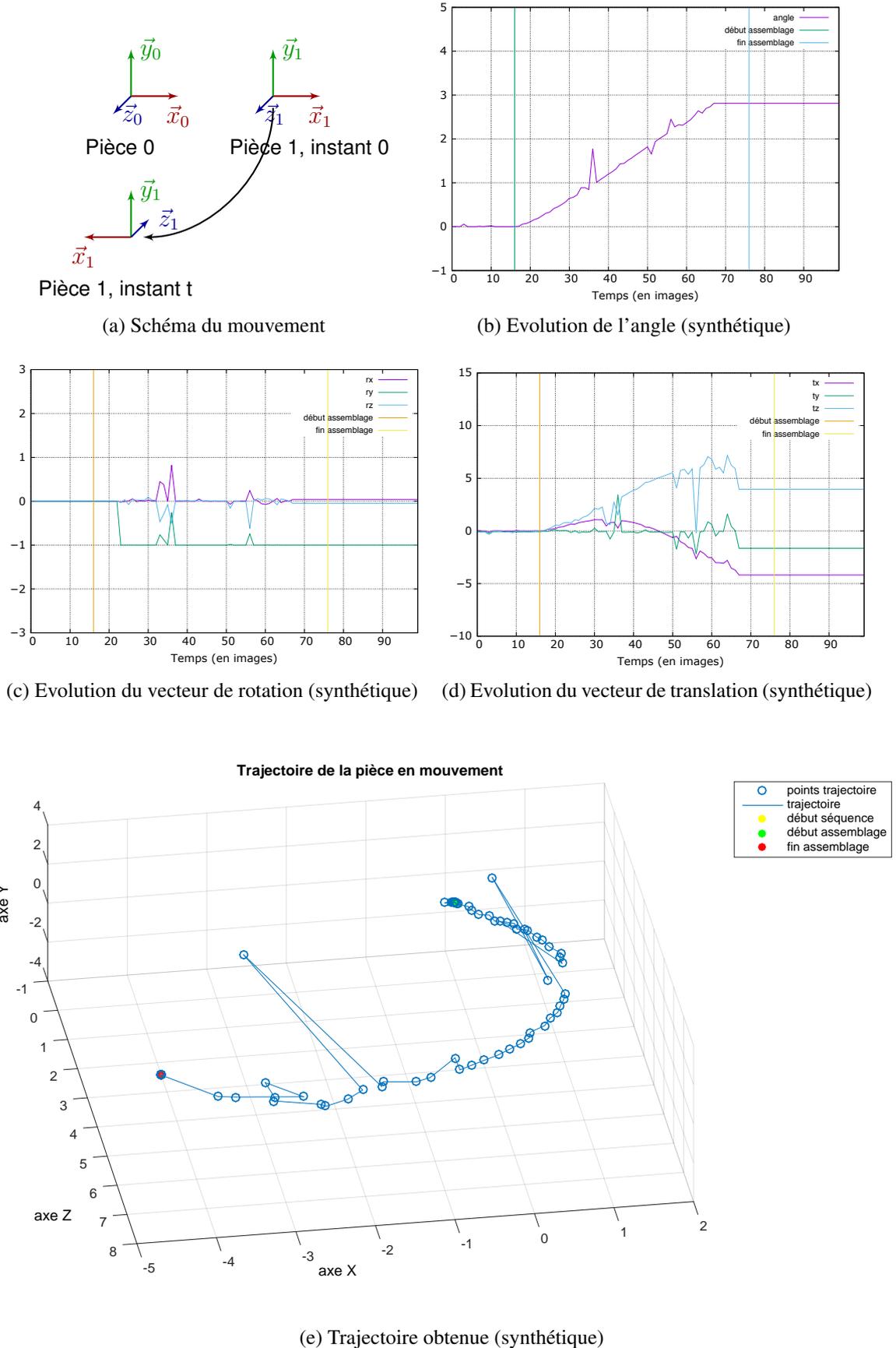


FIGURE 4.7 – Evolution des paramètres de mouvement lors l'assemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 en considérant le mouvement relatif de l'objet mobile à l'instant t par rapport à sa position initiale. (pièces posées sur un plan)

4.2 Défauts liés à la mesure

Dans la section précédente, nous avons montré que la matrice de transformation pouvait transmettre globalement la bonne information concernant le mouvement entre nos pièces. Cependant, à chaque instant, les courbes que nous obtenions pour certaines simulations (fig. 4.8 page 138) dans le cadre réel étaient très bruitées (fig. 4.9 page 138). Nous ne pouvions pas être certain de la qualité de la mesure recueillie pour des actions enregistrées dans des conditions réelles. S'agit-il d'une mesure exacte ou d'une autre bruitée ? Si elle est bruitée, à quel point peut-on lui accorder confiance et comment la corriger ? C'est dans l'optique de répondre à ces interrogations que, dans cette partie, nous nous sommes intéressés aux paramètres en entrée qui pouvaient influencer les paramètres de mouvement extraits de la matrice de transformation (voir section 4 page 113).

Pour cela, nous avons, dans un premier temps, identifié les éléments dans la chaîne de calcul pour l'estimation du mouvement qui pouvait avoir une influence. Nous utilisons les mesures de corrélation (Pearson et Spearman) et l'information mutuelle afin de sélectionner ceux qui pouvaient influencer nos données de mouvement en sortie. Dans un premier temps, nous avons établi une fonction liant certains de ces éléments avec l'erreur dans le mouvement à l'aide la régression linéaire. Dans un deuxième temps, nous avons testé une méthode de classification afin de prédire le niveau de confiance que nous pouvons avoir sur les données de mouvement enregistrées. Nous avons notamment appliqué les machines à vecteurs de support (SVM) que nous avons déjà exploré dans des travaux liés à la reconnaissance des genres lors de travaux en entreprise (cf. annexe C page 194). Nous avons ainsi établi une méthode afin de réduire le bruit présent dans les signaux représentant le mouvement relatif des pièces et obtenir ainsi des valeurs facilement exploitables pour des comparaisons avec les liaisons enregistrées dans notre base de connaissance.

4.2.1 Paramètres influençant l'estimation du mouvement

Lors du processus d'estimation du mouvement des pièces mécaniques, plusieurs données peuvent influencer la qualité de cette estimation. Nous avons analysé la corrélation entre l'erreur dans l'estimation de cette matrice et les paramètres suivants :

- le nombre d'*inliers* de chaque objet utilisés pour l'estimation de la pose ;
- le déplacement 2D et 3D du centre du nuage de points de chaque objet entre deux images de notre séquence vidéo qui peut être interprété comme la "perturbation" de ce centre de gravité au cours du temps ;
- les informations géométriques des nuages de points 3D telles que la linéarité, la planarité

et la sphéricité qui sont déterminées de la manière suivante :

$$\left\{ \begin{array}{l} \text{linéarité} = \frac{\lambda_1 - \lambda_2}{\lambda_1} \\ \text{planarité} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \\ \text{sphéricité} = \frac{\lambda_3}{\lambda_1} \end{array} \right. \quad (4.12)$$

avec $\lambda_{1,2,3}$ correspondent aux valeurs propres issues de la matrice de covariance du nuage de points. Les valeurs propres sont ordonnées comme suit $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

- le volume et la densité des nuages de points de chaque objet,
 - la qualité moyenne des mises en correspondance qui représente la valeur moyenne des distances enregistrées pour les mises en correspondance,
 - le pourcentage d'*inliers* pouvant être considérés comme des correspondances fortes.
- Nous avons défini une correspondance forte de la manière suivante :

$$c \in S_{forte} \Leftrightarrow \text{dist}(c) \leq \mu + \frac{3\sigma}{\sqrt{N}} \quad (4.13)$$

avec c un *inlier*, S_{forte} l'ensemble des correspondances fortes, $\text{dist}(c)$ la distance euclidienne associée à la correspondance c , μ et σ respectivement la valeur moyenne et l'écart-type des scores de similarité des correspondances établies pour un objet spécifique à une image t de notre séquence vidéo et N le nombre d'*inliers*. Nous nous inspirons des paramètres d'un intervalle de confiance à 99,7 % pour établir le seuil que nous utilisons. Ce pourcentage peut indiquer que les scores de similarité des correspondances mises en place ont 99,7 % de chances de se trouver dans l'intervalle choisi. Nous notons alors le pourcentage effectif d'*inliers* se trouvant dans cet intervalle.

Comme la matrice de transformation est basée sur les poses des objets mis en jeu dans les séquences vidéos, nous avons également pris en compte certains paramètres caractérisant les deux objets. Nous avons sélectionné, dans ce cas précis, que les paramètres présentant des corrélations fortes telles que

- le rapport entre le nombre d'*inliers* de l'objet de référence et de l'objet en mouvement ;
- le rapport entre le nombre d'*inliers* de l'objet de référence et celui de l'objet en mouvement et le rapport entre le nombre d'*inliers* de l'objet en mouvement et celui de l'objet de référence suivant deux formulations différentes définies de la manière suivante :

$$\left\{ \begin{array}{l} r_{inliers}(O_r/O_m) = \frac{N_{Or}}{N_{Om}} + \frac{N_{Om}}{N_{Or}} \\ r_{expo_{inliers}}(O_r/O_m) = e^{\frac{-N_{Or}}{N_{Om}}} + e^{\frac{-N_{Om}}{N_{Or}}} \end{array} \right. \quad (4.14)$$

- le rapport entre le pourcentage de correspondances fortes de l'objet de référence et celui de l'objet en mouvement et le rapport entre le pourcentage de correspondances fortes entre l'objet en mouvement et celui de référence suivant la formulation suivante :

$$r_{pourcentage}(O_r/O_m) = \frac{P_{O_r}}{P_{O_m}} + \frac{P_{O_m}}{P_{O_r}} \quad (4.15)$$

L'erreur dans l'estimation de la matrice de transformation est calculée de la manière suivante :

$$erreur = \|T_{exp}T_{théorique}^{-1}\|_{frobenius} \quad (4.16)$$

avec T_{exp} la matrice de transformation à l'instant t et $T_{théorique}$ la matrice de transformation théoriquement appliquée aux objets à l'instant t .

Nous avons établi nos corrélations en nous basant sur les données recueillies des vidéos synthétiques. Nous avons effectué cette analyse sur 9 vidéos de 100 images chacune représentant des mouvements distincts. Comme le mouvement théorique entre les objets est connu dans ces vidéos, nous avons donc une vérité terrain sur les transformations entre les objets qui sont manipulés.

Afin d'identifier les liens entre les paramètres en entrée et l'erreur entre la théorie et l'expérimentation, nous avons utilisé trois métriques :

- **Coefficient de corrélation** : le coefficient de corrélation linéaire ou coefficient de Pearson indique le degré de liaison linéaire entre deux variables. La valeur p associée permet de tester la significativité statistique des valeurs du coefficient de corrélation calculée.
- **Coefficient de Spearman** : contrairement au coefficient de Pearson, le coefficient de Spearman permet d'identifier les liens monotones entre deux variables.
- **Information mutuelle** : elle permet d'identifier le degré de dépendance statistique existant entre deux variables.

Déterminer ces trois métriques permettrait d'identifier assez rapidement le type de lien pouvant exister entre deux variables et ainsi de déterminer la forme de la fonction qui pourrait les lier.

Nous obtenons ainsi les valeurs suivantes pour les métriques présentées précédemment que nous regroupons dans le tableau 4.1. L'objet 0 correspond à l'objet de référence, statique dans nos séquences vidéos et l'objet 1 à l'objet en mouvement.

		Erreur modèle / expérience				
id objet	paramètres	Pearson	valeur p	Spearman	valeur p	IM
0	nb inliers	0.0842	0.0115	-0.3585	0	1.3070
	déplacement 2D	0.1604	0	0.0184	0.5816	0.2173
	déplacement 3D	0.0269	0.4204	0.1029	0.0020	0.0331
	linéarité	0.0282	0.3980	-0.0689	0.0389	0.1241
	planarité	0.0480	0.1498	0.1071	0.0013	0.1111
	sphéricité	-0.0769	0.0210	-0.0019	0.9544	0.0690
	volume	0.0734	0.0277	-0.2203	0	1.0081
	densité	-0.0532	0.1106	-0.0133	0.6907	0.2338
	qualité moyenne correspondance	-0.4184	0	-0.0938	0.0049	0.7058
	pourcentage correspondance forte	0.1979	0	0.2920	0	3.2034e-16
1	nb inliers	-0.4844	0	-0.2465	0	1.1266
	déplacement 2D	0.5103	0	0.1476	0	0.7344
	déplacement 3D	0.5782	0	0.1503	0	0.2468
	linéarité	0.2611	0	0.1253	0.0002	0.1404
	planarité	0.1068	0.0013	-0.1370	0	0.1039
	sphéricité	-0.3043	0	-0.1397	0	0.0389
	volume	-0.3329	0	-0.0959	0.0040	1.0338
	densité	0.0086	0.7962	-0.1835	0	0.4276
	qualité moyenne correspondance	0.1578	0	-0.1049	0.0016	1.2838
	pourcentage correspondance forte	0.6592	0	0.4022	0	3.2034e-16
0 et 1	rapport inliers objet 0 / objet 1	0.6499	0	-0.1972	0	0.8414
	$r_{inliers}(O_r/O_m)$	0.6582	0	0.3969	0	0.8241
	$r_{expo_{inliers}}(O_r/O_m)$	0.7313	0	0.4145	0	3.2034e-16
	$r_{pourcentage}(O_r/O_m)$	0.7331	0	0.3695	0	3.2034e-16

TABLE 4.1 – Tableau récapitulant les corrélations entre l’erreur dans l’estimation de la matrice de transformation et différents paramètres pouvant l’influencer.

Si l'on ne tenait compte que des objets de manière individuelle et uniquement du coefficient de corrélation, il s'avère que les paramètres de l'objet de référence, excepté la qualité moyenne des correspondances, ont une influence linéaire quasiment nulle sur la matrice de transformation. En revanche, la qualité moyenne des correspondances a une corrélation linéaire moyenne avec l'erreur dans l'estimation de cette matrice. En effet, cela peut être dû au fait que ce paramètre garantit, d'une certaine manière, la qualité de la pose associée à l'objet de référence. Cependant, comme cette valeur n'est pas assez forte (en général, on estime que la corrélation linéaire est forte si sa valeur est supérieure à 0.7), nous ne pouvons pas en déduire une potentielle existence d'une fonction linéaire entre la qualité des correspondances et l'erreur. Toutefois, l'information mutuelle nous révèle une potentielle relation entre l'erreur et le nombre d'*inliers* de l'objet de référence, le volume de son nuage de points et la qualité moyenne de ses correspondances.

Pour l'objet en mouvement, le pourcentage de correspondances fortes possède une corrélation linéaire presque forte avec l'erreur commise sur la matrice de transformation. En revanche, le nombre d'*inliers* de l'objet de référence, leur déplacement 2D ou 3D et la dispersion 2D de son nuage de points ont une corrélation linéaire moyenne. Le coefficient de Spearman permet d'éliminer ces dernières variables dans le cadre d'une recherche d'une potentielle fonction linéaire avec l'erreur. En effet, leur coefficient de Spearman est relativement faible, cela indique que s'il y a une fonction reliant l'erreur et ces variables, celle-ci n'est pas monotone et donc non linéaire. L'information linéaire nous révèle, toutefois, que l'erreur et les variables suivantes : le nombre d'*inliers* de l'objet en mouvement, son déplacement 2D, le volume de son nuage de points et la qualité moyenne des correspondances, ont statistiquement une forte dépendance entre eux.

De plus, il apparaît ainsi que, dans la structure géométrique des nuages de points représentée, la linéarité, la planarité ou la sphéricité n'ont pas de corrélation significative avec l'erreur dans l'estimation de la matrice de transformation.

Si nous combinons les informations concernant le nombre d'*inliers* et le pourcentage de correspondances fortes provenant de chaque objet, c'est-à-dire, en calculant les rapports de ces informations entre un objet en mouvement et l'objet de référence, une corrélation linéaire forte peut être relevée. Les rapports $r_{\text{expo}_{\text{inliers}}}(O_r/O_m)$ et $r_{\text{pourcentage}}(O_r/O_m)$ affichent les valeurs du coefficient de Pearson les plus fortes dans nos expérimentations. Nous testerons ces métriques pour établir, à l'aide de la régression linéaire, une fonction linéaire entre l'erreur et celles-ci afin de prédire la qualité de la pose estimée ainsi que le degré de confiance à attribuer aux données relevant du mouvement entre nos objets. Nous avons également tenté d'utiliser ces métriques pour établir des niveaux de confiance en la matrice de transformation à l'aide de la logique floue.

Il est à noter qu'en général pour estimer la pose de la caméra, trois à cinq points au minimum

sont requis selon l'algorithme PnP choisi. Cependant, pour ne se contenter que de ce nombre de points, il faudrait être certain que les points extraits de l'image 2D correspondent parfaitement aux primitives 3D du modèle présent dans notre base de données, ce qui n'est pas forcément le cas. Cela signifie donc que si un des points correspond à un *outlier*, la pose calculée pour un des objets peut être totalement aberrante. Cela induit ainsi une mauvaise estimation de la matrice de transformation. Si un nombre de correspondances assez important est disponible pour estimer la pose, une sélection peut être effectuée afin d'éliminer un certain nombre d'*outliers*. De plus, la qualité de la mise en correspondance ou, du moins, le pourcentage de points pour lesquels les correspondances peuvent être considérées comme fortes ont une influence sur l'estimation de la pose induisant ainsi une erreur sur l'estimation de la matrice de transformation. Ainsi, nous avons exploré l'utilisation d'une méthode d'apprentissage, notamment les machines à vecteurs support à l'aide des variables concernant le nombre d'*inliers* de chaque objet ainsi que la qualité moyenne des correspondances afin de prédire le niveau de confiance que l'on peut avoir sur la matrice de transformation calculée à l'instant t .

Jusqu'à présent, les analyses que nous réalisons sur des données recueillies dans des conditions idéales peuvent ne pas refléter correctement la réalité. Mais, comme nous n'avons pas dans notre base de connaissance la valeur de la matrice de transformation théorique à chaque instant t dans les conditions réelles, nous nous sommes également demandé s'il existait une corrélation entre la variation de la matrice de transformation entre l'instant t et l'instant $t + 1$ et l'erreur d'estimation entre la transformation théorique et celle expérimentale. La variation de la matrice de transformation est calculée de la manière suivante :

$$variation = \|T_t T_{t+1}^{-1}\|_{frobenius} \quad (4.17)$$

A partir de nos données, nous avons observé une forte corrélation linéaire entre ces deux informations (cf tableau 4.2). Cela nous permettrait de nous affranchir de la nécessité d'avoir le modèle de mouvement théorique (donc le positionnement des pièces images par images sur nos séquences réelles) pour la caractérisation de la qualité des matrices de transformation calculées dans des conditions réelles. En effet, établir le mouvement théorique des pièces dans des conditions réelles peut être difficile et variable suivant la rapidité de l'utilisateur.

De plus, nous avons également vérifié que les corrélations entre les deux métriques que nous avons retenues pour l'estimation de la confiance à attribuer aux mesures que nous calculons et les variations de la matrice de transformation restaient fortes (cf tableau 4.2). Cette vérification nous permettait de garder ces deux métriques dans le cadre d'une estimation d'une fonction linéaire entre ces variables et l'erreur associée dans le cadre d'une simulation réelle.

id objet	paramètres	variation matrice de transformation
0 et 1	erreur modèle / expérience	0.7454
	$r_{expo_inliers} (O_r/O_m)$	0.6934
	$r_{pourcentage} (O_r/O_m)$	0.6718

TABLE 4.2 – Tableau récapitulant les corrélations entre la variation de la matrice de transformation image par image et les métriques que nous avons retenues.

4.2.2 Utilisation de la régression linéaire

La régression linéaire est une méthode permettant d'établir une fonction linéaire entre deux variables Y et \mathbf{X} . Un modèle linéaire expliquera la donnée en sortie Y en fonction de la donnée en entrée \mathbf{X} . Il s'exprime de la manière suivante :

$$Y = \beta_0 + \beta_1 \mathbf{X} + \varepsilon \quad (4.18)$$

avec β_0 et β_1 des coefficients à déterminer et ε un bruit suivant en général une loi normale.

Les coefficients sont établis à l'aide de la méthode des moindres carrés, c'est-à-dire en minimisant la somme des carrés des écarts entre les n données enregistrées et celles calculées à partir du modèle de l'équation 4.18 :

$$\min_{\beta_0, \beta_1} \sum_{i=1}^n (\beta_0 + \beta_1 \mathbf{x}_i + \varepsilon - y_i)^2 \quad (4.19)$$

Ainsi, d'une part, nous avons modélisé le lien pouvant exister entre les variables montrant les plus grands coefficients de corrélations (cf. tableau 4.2), c'est-à-dire $r_{expo_inliers} (O_r/O_m)$ et $r_{pourcentage} (O_r/O_m)$ représentées par la variable \mathbf{X} et l'erreur dans l'estimation de la matrice de transformation (cf. équation 4.16 page 129) représentée par la variable Y . Cette fonction nous permettrait de prédire si la matrice de transformation que nous calculons à l'instant t est correct ou pas et ainsi quelle confiance attribuée aux valeurs de mouvement extraites à partir de cette matrice.

Nous avons entraîné un modèle de régression sur des données générées synthétiquement. Celles-ci sont recueillies à partir de 9 vidéos synthétiques de 100 images représentant différents types de mouvement entre les pièces mécaniques. Une centaine de données choisies aléatoires sont destinées au test et les autres pour l'apprentissage. Nous obtenons un modèle de régression avec un résidu de $-3.5807e-10$. Cette valeur permet de supposer que les valeurs prédites seront proches de celle attendue théoriquement.

Afin de déterminer la précision de modèle de régression linéaire appris, c'est-à-dire le pourcentage de bonne classification, nous considérons que

- les vrais positifs sont détectés lorsque les matrices de transformation théoriques et expérimentales sont similaires, c'est-à-dire lorsque l'erreur calculée à partir de l'équation 4.16 et l'erreur prédite à partir du modèle de régression 4.18 sont inférieures à un seuil s (fixé à 4 dans nos expérimentations).
- les vrais négatifs sont détectés lorsque ces matrices sont différentes, c'est-à-dire lorsque les erreurs sont supérieures à s .

Nous obtenons ainsi une précision de 79%. Il s'agit du pourcentage de bonne classification en se basant sur le modèle appris.

D'autre part, nous établissons également le modèle linéaire reliant l'erreur entre la matrice de transformation théorique et expérimentale (cf. eq. 4.16 page 129) représentée par la variable Y et la variation de la matrice de transformation entre les instants t et $t-1$ (cf. eq. 4.17 page 132) représentée par la variable X . Cette fonction permettra de "retrouver" l'erreur entre la matrice de transformation théorique et expérimentale dans des situations réelles où la matrice de transformation théorique n'est pas disponible. L'objectif est de réaliser un apprentissage directement sur les données réelles et de faire des corrections adéquates sur les données de rotation et de translation afin de les exploiter directement pour la reconnaissance du mouvement et la détection d'erreur d'exécution.

Comme pour l'apprentissage précédent, nous nous basons sur les données synthétiques qui nous permet de vérifier la justesse du modèle. Il s'avère que le modèle de régression appris nous donne un résidu de 477.3804 sur les données d'apprentissage. Cela suppose que le modèle reste très éloigné des valeurs attendues. Ainsi, malgré la corrélation forte que nous avons obtenu entre l'erreur sur les matrices de transformation théoriques et expérimentales et la variation sur les matrices de transformation entre l'instant t et $t-1$ (cf. tableau 4.2), la régression linéaire ne nous permet pas d'obtenir un lien fiable entre ces deux valeurs.

NB : Par ailleurs, si une classification des matrices de transformation en se basant sur la variation est effectuée, il est probable que celle-ci ne correspondra pas à celle se basant sur l'erreur entre les matrices de transformation théorique et expérimentale. Par exemple, si on suppose un séquence de classification suivante avec l'erreur : S-S-S-D-S-D-D-S-D-S-D-S-S (avec S : les matrices sont similaires et D : les matrices sont différentes). Nous obtiendrons la séquence suivante avec la variation des matrices : S-S-S-D-D-D-I-D-D-D-D-D-S (avec N : classification indéterminée, c'est-à-dire si la séquence avec l'erreur est D-D, S obtenue lorsque nous avons une séquence S-S et D lorsque la séquence est S-D ou D-S). Ainsi, nous pourrions considérer que la plupart des matrices de transformation extraites sont incorrectes si les données sont très bruitées.

Vu que le lien ne peut être fiable et donc que nous ne pouvons estimer l'erreur entre l'expérience et la théorique, nous nous sommes intéressés à une autre méthode d'apprentissage, notamment au SVM afin de classifier nos données réelles et corriger les valeurs de rotation et de translation.

4.2.3 Utilisation du SVM

Afin de créer un hyperplan de séparation de nos données, nous appliquons un SVM sur celles-ci afin d'estimer le niveau de confiance que nous pouvons avoir sur les matrices de transformation extraites. Ce niveau de confiance nous permettra, par la suite, d'établir une méthode pour réduire le bruit existant liés à la mesure des différents paramètres de mouvement.

Les paramètres en entrée utilisés pour l'apprentissage ont été choisis suivant la valeur de leur information mutuelle. Parmi les variables dont l'information mutuelle est grande, nous avons sélectionné les paramètres suivants pour effectuer notre apprentissage à l'aide du SVM :

- le nombre d'*inliers* de chaque objet (de référence et en mouvement),
- la qualité moyenne des correspondances de chaque objet.

Nous les avons concaténés afin de constituer notre descripteur pour la classification. Le vecteur ainsi obtenu sera de taille 4.

La labellisation des données est réalisée en fonction de la valeur à chaque instant t de l'erreur entre la matrice de transformation théorique et celle expérimentale. Comme seules les données issues du cadre idéal, donc des vidéos synthétiques, contiennent des informations sur la matrice de transformation théorique à chaque instant, celles-ci constituent notre base d'entraînement. Le label 0 indiquera que les matrices de transformation sont similaires et donc sont correctes. En revanche, le label 1 signifiera que la matrice de transformation expérimentale est très différente de celle qui est attendue, donc que celle-ci est incorrecte.

$$Tag_t = \begin{cases} 0 & \text{if } e_t \leq \varepsilon \\ 1 & \text{sinon} \end{cases} \quad (4.20)$$

Avec e_t spécifié par l'équation 4.16 page 129 calculé à chaque instant t et ε un seuil (fixé à 2.5 dans nos expérimentations).

Nous avons entraîné sur les données sélectionnées un modèle SVM avec un noyau gaussien [Vapnik, 1998] après la normalisation de ces dernières. Nous obtenions ainsi un hyperplan avec une précision de 86.4%.

A l'aide ce modèle, nous pouvons obtenir un score de confiance [Platt, 1999] de nos données réelles. Ce score indique la probabilité qu'une donnée appartienne à un groupe donné.

4.2.4 Correction des mesures de mouvement en temps réel

Afin de corriger le bruit existant dans les mesures du mouvement, c'est-à-dire les angles et vecteurs de rotation et les vecteurs de translation, il était possible d'y appliquer des filtres de lissage de signaux 1D. Il aurait pu s'agir des filtres tels que les filtres médian, de moyenne, passe-bas, celui de Savitzky Golay, Chebyshev, Butterworth, Wiener, ... [Chen et al., 2006, Hargittai, 2005] ou de passer par une transformation en ondelettes [Xu et al., 1994, Pizurica et al., 2003]. Cependant, certaines méthodes telles que la transformation en ondelettes sont des approches hors-ligne ou nécessitent d'être appliquées sur une fenêtre de données. Or le fenêtrage dans le cadre d'une application en temps réel induit la présence d'un délai dans la prise de décision. Par ailleurs, les données corrigées à un instant t à l'aide de ces méthodes peuvent être impactées si les données précédentes ou suivantes correspondent à des données aberrantes avec une magnitude élevée. Le filtre de Kalman aurait pu également être choisi. Cependant celui-ci suppose de prédire l'évolution des données quelle que soit la liaison en jeu. Établir un *apriori* sur les relations des données au cours du temps supposerait d'établir un modèle pour tout type de liaison mécanique présente dans notre base de connaissance. Cela peut devenir rapidement un travail fastidieux si nous devons les définir en prenant en compte les vitesses d'exécution des tâches de chaque utilisateur qui définissent la progression des courbes de mouvement que nous analysons.

Afin de pallier les inconvénients des approches précédentes, nous avons proposé d'utiliser les scores de confiance pouvant être déterminés avec le modèle SVM établi précédemment. Ce score nous indique à quel point nous pouvons avoir confiance en la matrice de transformation que nous avons estimée. De plus, comme ces matrices sont dépendantes des matrices de projection, il nous précise également à quel point nous pouvons avoir confiance en la précision de l'estimation de la pose.

En se basant sur ce score, nous avons introduit les formules suivantes pour corriger les valeurs calculées pour l'angle de rotation α , le vecteur de rotation $\vec{\omega}$ et le vecteur de translation \vec{t} . Notre objectif, avec ces formules, est de réduire la valeur des mesures que nous suspectons d'être erronées. Dans notre formulation, non seulement nous prenons en compte le score de confiance mais également nous vérifions qu'il existe une cohérence temporelle entre les données successives. Nous nous inspirons des hypothèses habituelles du filtre de Kalman utilisées pour le suivi d'objet dans des séquences vidéo qui spécifient qu'entre deux images successives, les objets restent quasiment à la même position, c'est-à-dire leur déplacement est pratiquement nul. Ainsi nous supposons qu'entre deux instants successifs $t - 1$ et t , l'angle de rotation et le vecteur de translation sont pratiquement similaires et donc qu'ils ne varient pas énormément. En revanche, contrairement au filtre de Kalman, nous n'introduisons pas de paramètre représentant le bruit pouvant exister dans le modèle. Nous corrigeons automatiquement les mesures enregistrées à l'instant t .

Ainsi, pour réduire le bruit sur les paramètres angulaires, nous appliquons la formule sui-

vante :

$$\alpha_{t|g=1}^{corr} = \begin{cases} \alpha_{t-1|g=0} + (1-c)(\alpha_{t|g=1} - \alpha_{t-1|g=0}) \\ \alpha_{t-1|g=1}^{corr} + (1-c)(\alpha_{t|g=1} - \alpha_{t-1|g=1}^{corr}) \end{cases} \quad (4.21)$$

Avec g le label prédit par le modèle SVM, c le score de confiance obtenue pour le label prédit à l'instant t et $corr$ signifie "corrigé".

Si nous suspectons une mauvaise classification d'une mesure en se basant sur son score de confiance faible ou sur la grande variation soudaine de cette mesure par rapport à la mesure précédente, nous appliquons alors la formule suivante :

$$\alpha_{t|g=0}^{corr} = \alpha_{t|g=0} - k(1-c)(\alpha_{t|g=0} - \alpha_{t-1}^{corr}) \quad (4.22)$$

Avec k un coefficient déterminé empiriquement.

Nous appliquons les mêmes formules spécifiées précédemment sur chacune des coordonnées du vecteur de translation.

En revanche, contrairement à l'angle de rotation et au vecteur de translation, nous ne pouvons pas considérer que les coordonnées du vecteur de rotation sont pratiquement similaires à des instants successifs. Ces coordonnées peuvent apparaître ou disparaître de manière soudaine dès le moment où l'angle de rotation augmente ou devient nul. Et elles resteront stables durant la procédure d'assemblage. Ainsi, il est nécessaire de spécifier une correction qui soit adaptée à ce cas. La formule suivante est appliquée lorsque nous n'avons aucune confiance en la matrice de transformation relative, c'est-à-dire nous suspectons que celle-ci est erronée :

$$\vec{\omega}_{t|g=0}^{corr} = \begin{cases} 0 & \text{if } \alpha_{t|g=0}^{corr} \leq \varepsilon_\alpha \\ \vec{\omega}_{t|g=0} & \text{si } \alpha_{t|g=0}^{corr} \geq \varepsilon_\alpha \text{ et } c \geq \varepsilon_{confiance} \\ \vec{\omega}_{t-1}^{corr} & \text{sinon} \end{cases} \quad (4.23)$$

et lorsque cette matrice est correcte :

$$\vec{\omega}_{t|g=1}^{corr} = \begin{cases} 0 & \text{if } \alpha_{t|g=1}^{corr} \leq \varepsilon_\alpha \\ \vec{\omega}_{t-1}^{corr} & \text{sinon} \end{cases} \quad (4.24)$$

avec ε_α un seuil angulaire et $\varepsilon_{confiance}$ un seuil pour le score de confiance.

4.2.5 Expérimentations

Le protocole expérimental est le même que dans la section 3.1 page 83. Dans le cadre d'une vérification de nos formulations, nous l'avons testé sur les données recueillies pour un assemblage suivant une rotation autour de l'axe Y de l'objet de référence avec nos pièces qui ne sont

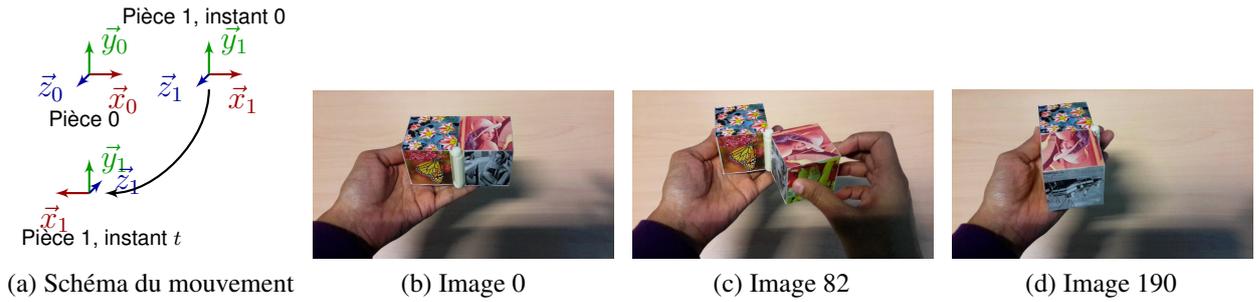


FIGURE 4.8 – Assemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 dans des conditions réelles. (pièces non posées sur une surface plane)

pas posées sur une surface plane.

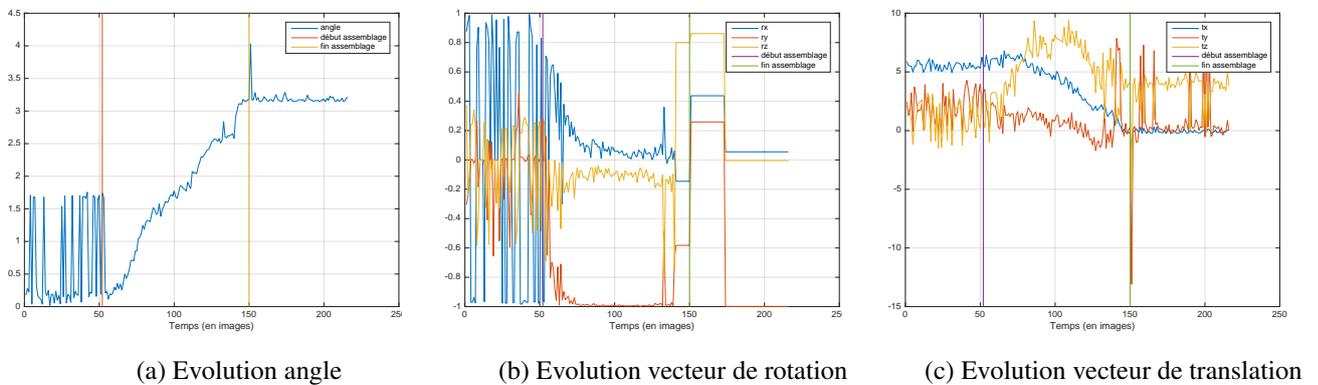


FIGURE 4.9 – Evolution des paramètres de mouvement obtenues dans le cadre d'un de deux pièces suivant une rotation autour de l'axe Y de l'objet 0 dans des conditions réelles. (pièces non posées sur une surface plane).

A chaque instant, nous prédisons la qualité de la matrice de transformation en se basant sur le modèle SVM que nous avons appris. Le score de confiance pour chaque mesure est enregistré et utilisé pour la correction en temps réel des paramètres de mouvement extraits des matrices de transformation.

Ainsi, pour le cas de la rotation autour de l'axe Y de l'objet de référence (fig. 4.8), nous observons qu'avant la correction du signal, les données brutes pouvaient induire le système de reconnaissance d'actions en erreur. En effet, le système aurait pu identifier à certains instants un mouvement avec potentiellement un *gimbal lock* avant le début de l'assemblage (fig. 4.9) du fait que l'angle de rotation varie énormément et que le vecteur de rotation n'est pas stable. En prenant en compte ces variations et en identifiant un axe dominant malgré le *gimbal lock*, le système aurait considéré qu'il se déroulait un pivot autour de l'axe X avant la phase effective d'assemblage. Il aurait ainsi transmis une notification à l'utilisateur lui indiquant qu'il n'effectuait pas la bonne action alors qu'aucune action de la part de l'utilisateur n'était en cours en réalité. Ce type d'erreurs peut perturber l'opérateur et annihiler complètement la confiance qu'il

pourrait avoir dans le système.

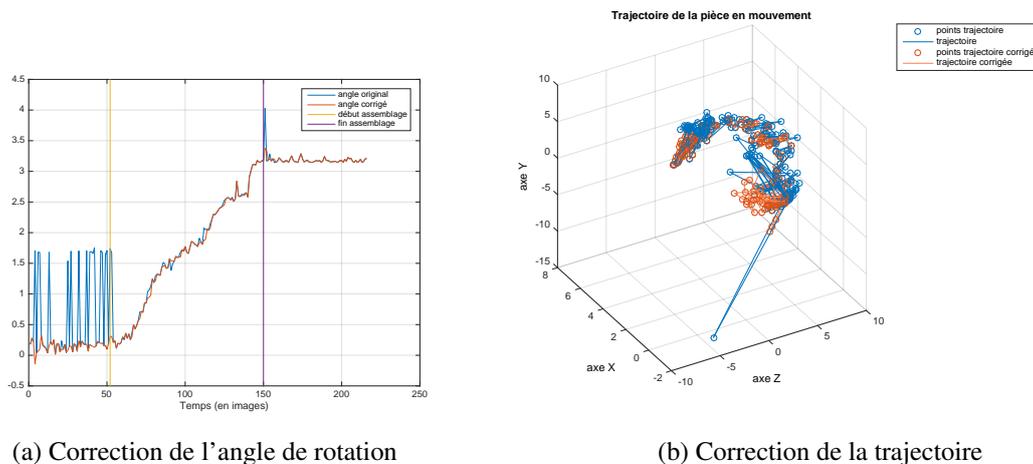


FIGURE 4.10 – Correction des valeurs de mouvements (angle et trajectoire).

Les corrections effectuées sur les paramètres angulaires comme l'illustre la figure 4.10a permettent d'éviter une mauvaise interprétation avant le début effectif de la phase d'assemblage. L'influence des mesures que nous considérons comme erronées est réduite à l'aide de la méthode que nous avons spécifiée qui permet de prédire à quel point une matrice de transformation relative est correcte. Les coordonnées du vecteur de rotation sont également globalement corrigées (fig. 4.11a, 4.11b, 4.11c). Dans notre exemple, lorsqu'il n'y a pas de mouvement, le système, après la réduction du bruit, pourra reconnaître les mouvements associés à la liaison mécanique lors de la phase effective de l'assemblage. Durant cette phase, la correction que nous avons effectuée sur les courbes des paramètres angulaires et rotationnels sont conformes à la courbe globale initialement enregistrée sans une application d'une méthode de réduction du bruit. Les pics sont supprimés et un léger lissage est perceptible. Nous pouvons noter les mêmes observations dans le cas des paramètres de translation (fig. 4.10b, 4.11a, 4.11b, 4.11c). Cependant, les mesures du vecteur de translation suivant l'axe Z restent encore assez bruitées (fig. 4.11c).

En général, notre approche pour réduire le bruit dans nos mesures n'introduit pas de valeurs fictives aberrantes ni de degré de liberté supplémentaire dans les mouvements enregistrés. Ceci est un point important dans le processus de contrôle des tâches d'assemblage.

4.2.6 Discussion

Dans cette section, nous avons, dans un premier temps, identifié les paramètres en entrée de notre pipeline pour l'estimation de la pose pouvant avoir une influence sur les mesures de mouvement relatif. Différents paramètres peuvent être spécifiés dans le but soit d'établir une fonction entre ces variables et l'erreur induite dans le calcul de la matrice de transformation à l'aide d'une régression linéaire, soit de prédire à quel point nous pouvons avoir confiance en

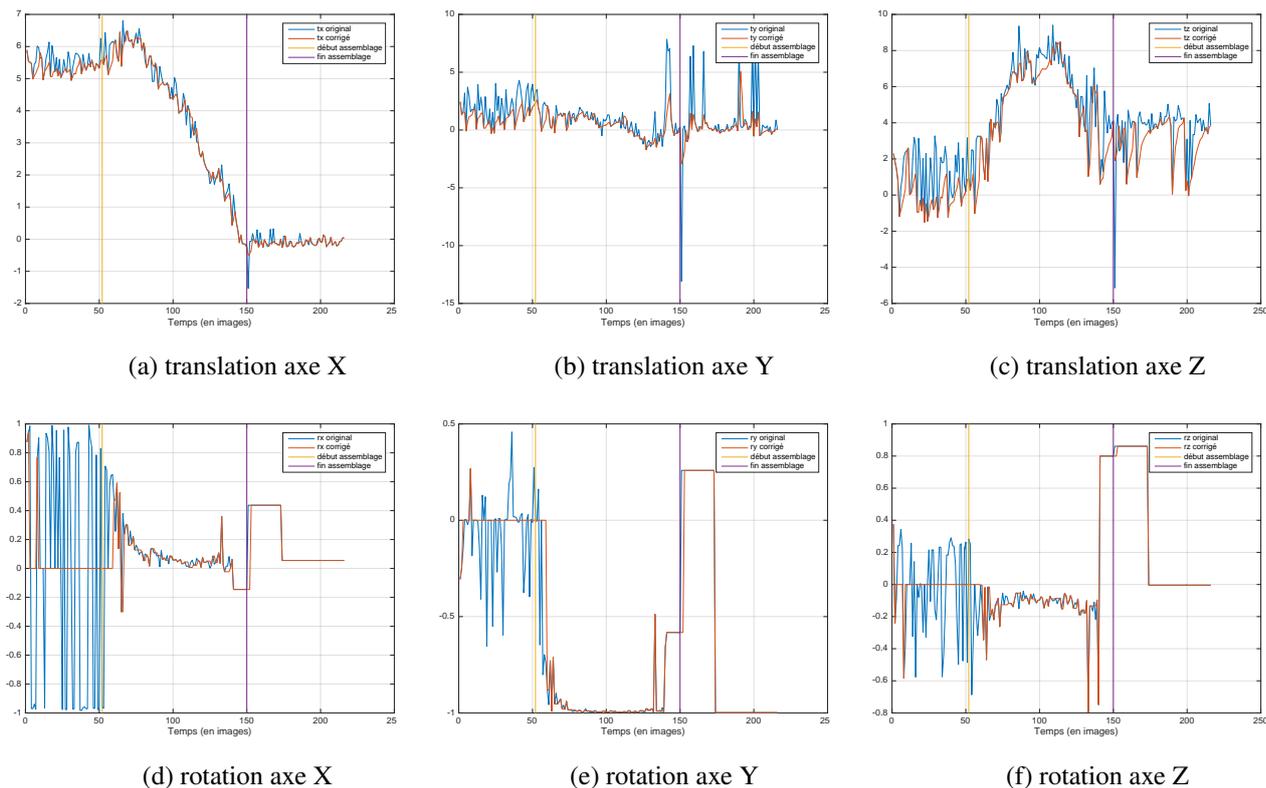


FIGURE 4.11 – Correction des courbes de mouvement (vecteur de rotation et de translation).

la matrice de transformation que nous extrayons. Ainsi nous avons établi un modèle SVM qui nous fournit un score de confiance à chaque instant. Ce score a été, ensuite, introduit dans nos formulations afin de réduire en temps réel le bruit présent dans les mesures du mouvement. Nos tests illustrent la réduction du bruit que nous avons pu obtenir. Nos formules permettent de ne pas ajouter de degré de liberté supplémentaires ni d'introduire de dérives et elle nous permet de réaliser une correction en temps réel de ces mesures.

Même si les résultats sont encourageants et que l'on a pu éliminer des valeurs très aberrantes, nos courbes sont encore assez bruitées. Un filtrage supplémentaire peut encore être effectué afin d'obtenir des courbes plus lisses. Après avoir rectifié les paramètres de mouvement, la prochaine étape consiste à établir une métrique permettant de reconnaître les actions de l'utilisateur et les liaisons mécaniques entre les pièces et ainsi de réaliser un contrôle automatique du processus d'assemblage.

4.3 Défauts liés à l'exécution de l'assemblage

Un assemblage entre deux pièces suit un mécanisme bien spécifique qui permet de définir de quelle manière ces pièces sont solidaires le long de son exécution. En mécanique, une nomenclature classe douze liaisons normalisées possibles en incluant la liaison nulle qui est le complémentaire de la liaison encastrement (cf. tableau 4.12).

NB : La liaison encastrement correspond à la première définition que nous avons établie précédemment dans le chapitre 3.

Nom	point d'application	ddl	Représentation plane	Représentation 3D	Torseur cinématique	Torseur des AM
Encastrement	tout point de l'espace	0			$\begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} F_x & M_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_A \mathcal{R}$
Pivot d'axe (A, \vec{x})	tout point de l'axe	1			$\begin{Bmatrix} \omega_x & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} F_x & 0 \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_A \mathcal{R}$
Glissière de direction \vec{x}	tout point de l'espace	1			$\begin{Bmatrix} 0 & V_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} 0 & M_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_A \mathcal{R}$
Hélicoïdale d'axe (A, \vec{x})	tout point de l'axe	1			$\begin{Bmatrix} \omega_x & p \cdot \omega_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} F_x & p \cdot F_x \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_A \mathcal{R}$
Pivot glissant d'axe (A, \vec{x})	tout point de l'axe	2			$\begin{Bmatrix} \omega_x & V_x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} 0 & 0 \\ F_y & M_y \\ F_z & M_z \end{Bmatrix}_A \mathcal{R}$
Rotule à doigt de centre A bloquée en \vec{x}	centre de la liaison	2			$\begin{Bmatrix} 0 & 0 \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} F_x & M_x \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_A \mathcal{R}$
Rotule de centre A	centre de la liaison	3			$\begin{Bmatrix} \omega_x & 0 \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} F_x & 0 \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_A \mathcal{R}$
Plane de normale \vec{y}	tout point de l'espace	3			$\begin{Bmatrix} 0 & V_x \\ \omega_y & 0 \\ 0 & V_z \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} 0 & M_x \\ F_y & 0 \\ 0 & M_z \end{Bmatrix}_A \mathcal{R}$
Annulaire en A d'axe (A, \vec{x})	au centre de la liaison	4			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ \omega_z & 0 \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ F_z & 0 \end{Bmatrix}_A \mathcal{R}$
Rectiligne de ligne (A, \vec{x}) et de normale \vec{y}	tout point du plan (A, \vec{x} , \vec{y})	4			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ 0 & V_z \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ 0 & M_z \end{Bmatrix}_A \mathcal{R}$
Ponctuelle en A de normale \vec{y}	tout point de la normale au contact	5			$\begin{Bmatrix} \omega_x & V_x \\ \omega_y & 0 \\ \omega_z & V_z \end{Bmatrix}_A \mathcal{R}$	$\begin{Bmatrix} 0 & 0 \\ F_y & 0 \\ 0 & 0 \end{Bmatrix}_A \mathcal{R}$

FIGURE 4.12 – Nomenclature des liaisons mécaniques entre deux solides. ¹

Ces liaisons mécaniques se différencient les unes des autres par le nombre de degrés de liberté, le nombre de rotations et de translations possibles. Si nous supposons que les liens entre nos pièces sont enregistrés suivant les torseurs cinématiques associées à la liaison mécanique, nous spécifierons une métrique permettant de reconnaître le mouvement associé et de notifier à l'utilisateur qu'une erreur a été commise. Suivant la définition 2, cette erreur apparaîtra si un degré de liberté s'ajoute dans les mesures de mouvement extraites de la matrice de transformation.

4.3.1 Méthode pour la reconnaissance d'actions

Les torseurs cinématiques différencient les liaisons mécaniques entre elles. Cependant, ceux-ci sont exprimés en fonction du point central de la liaison. Les mesures de mouvement que nous avons extraites sont exprimées en fonction des repères des objets (les origines des repères sont le centre de chacun des objets). Les paramètres de rotation et de translation correspondent donc aux changements d'orientation et au déplacement du repère lié à l'objet mobile par rapport au repère lié à l'objet statique de référence. En supposant que cette position centrale et le repère associé sont correctement enregistrés dans la base de connaissance, il est alors nécessaire, dans un premier temps, d'effectuer un changement de repères afin de faire correspondre les torseurs cinématiques (cf. fig. 4.13). Ainsi, les paramètres de mouvement relatif qui nous permettront d'identifier la liaison entre les pièces à chaque instant t correspondent à ceux extraits de la matrice de transformation $\mathbf{T}_{L \rightarrow i_t}$. Celui-ci est déterminé de la manière suivante :

$$\mathbf{T}_{0 \rightarrow i_t} = \mathbf{T}_{0 \rightarrow L} \mathbf{T}_{L \rightarrow i_t} \quad (4.25)$$

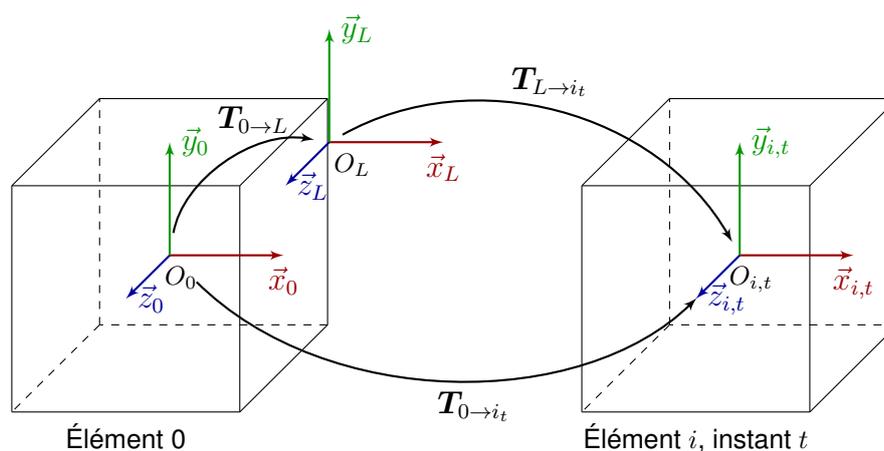


FIGURE 4.13 – Changement de repères pour la reconnaissance des liaisons entre les pièces (cas d'une liaison pivot d'axe une des arêtes du cube).

1. https://tiplanet.org/forum/archives_voir.php?id=179333

Le torseur cinématique d'une liaison en un point central O_L est exprimé et enregistré dans la base de connaissance de la manière suivante :

$$\mathcal{V}_{O_L \in R_L} = \begin{cases} \Omega_{R_L} \\ \mathbf{V}_{O_L \in R_L} \end{cases} \quad (4.26)$$

avec Ω_{R_L} la vitesse angulaire de l'objet dans le repère R_L et $\mathbf{V}_{O_L \in R_L}$ la vitesse linéaire du point O_L dans le repère R_L .

Comme nous étudions le mouvement au point $O_{i,t}$, point central de l'objet mobile, le torseur cinématique en ce point sera :

$$\mathcal{V}_{O_{i,t} \in R_L} = \begin{cases} \Omega_{R_L} \\ \mathbf{V}_{O_L \in R_L} + \mathbf{O}_{i,t} \mathbf{O}_L \wedge \Omega_{R_L} \end{cases} \quad (4.27)$$

Or comme nous exprimons la matrice de transformation $\mathbf{T}_{L \rightarrow i_t}$ sous forme d'une rotation de type axe-angle et d'un vecteur de translation telle que spécifiée dans la section 4.1, nous pouvons déterminer la vitesse angulaire théorique Ω_{R_L} et la vitesse linéaire $\mathbf{V}_{O_{i,t} \in R_L}$ du point $O_{i,t}$ dans le repère R_L de la manière suivante :

$$\begin{cases} \Omega_{R_L} = \frac{\omega_t - \omega_{t_0}}{t - t_0} \\ \mathbf{V}_{O_{i,t} \in R_L} = \frac{\mathbf{t}_{i,t} - \mathbf{t}_{i,t_0}}{t - t_0} = \frac{\mathbf{O}_{i,t} \mathbf{O}_L - \mathbf{O}_{i,t_0} \mathbf{O}_L}{t - t_0} \end{cases} \quad (4.28)$$

avec t_0 l'instant où l'on détecte le début de l'assemblage que nous déterminons à l'aide du KSUM que nous avons présenté dans la section 3.2.2.

Comme les liaisons, à l'exception de la liaison hélicoïdale, peuvent être identifiées de manière unique grâce au nombre de degrés de liberté possible suivant la rotation et suivant la translation et comme les mesures de mouvement extraites sont toujours bruitées malgré la réduction que nous avons établie dans la section 4.2.4, nous déterminons le nombre de degrés de liberté à l'instant t à l'aide d'un seuillage sur les écarts entre les torseurs cinématiques aux points O_L et $O_{i,t}$. Il y a un degré de liberté à l'instant t si :

$$|p_t| \geq \varepsilon \quad (4.29)$$

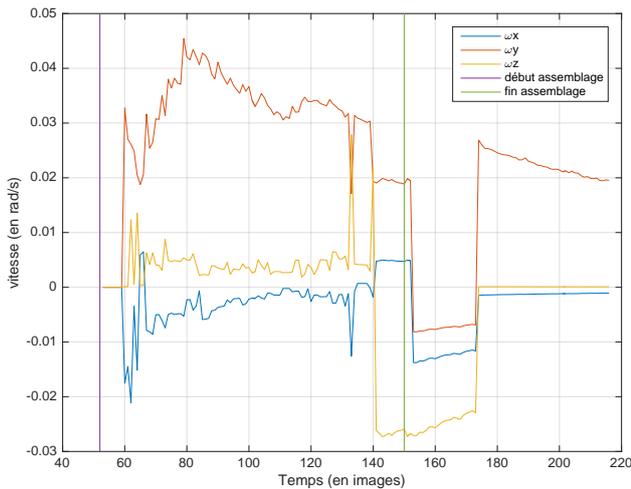
avec p pouvant correspondre aux coordonnées de la vitesse angulaire Ω_{R_L} ou celles de la vitesse linéaire de $\mathbf{V}_{O_L \in R_L}$ qui est calculée à partir des équations 4.27 et 4.28 à l'instant t .

Une erreur dans l'action effectuée par l'utilisateur intervient si ce nombre n'est pas constant le long de la simulation ou si les axes considérés diffèrent, c'est-à-dire si un pivot d'axe \vec{x} ne

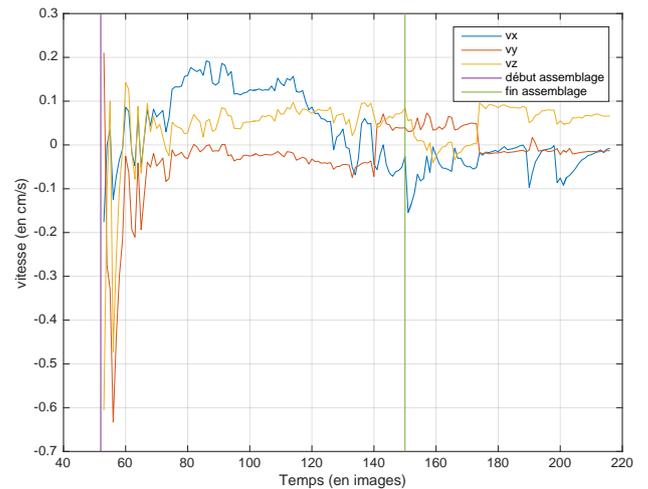
change pas en un pivot d'axe \vec{y} par exemple.

NB : Il est à signaler que nous testons une certaine persistance du nombre de degrés de liberté durant 5 images. Ceci nous permet de ne pas considérer qu'il y a une erreur à chaque fois qu'un outlier est mesuré.

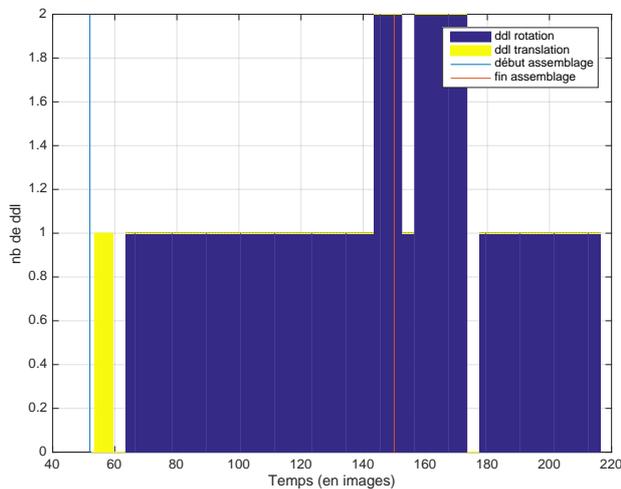
4.3.2 Expérimentations



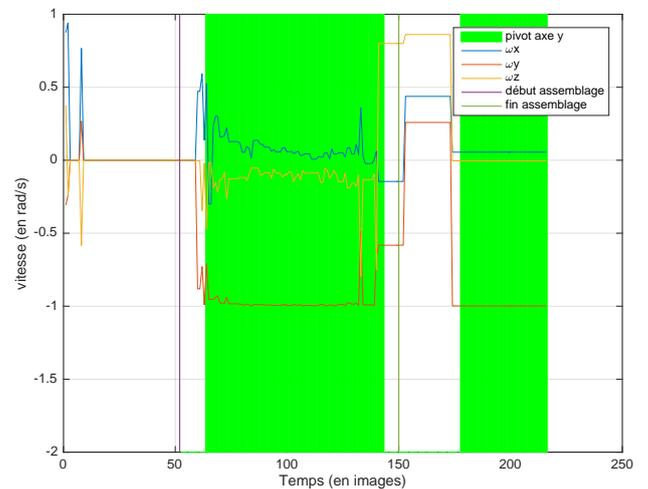
(a) vitesse angulaire



(b) vitesse linéaire



(c) nombre de degrés de liberté



(d) identification liaison

FIGURE 4.14 – Identification d'une liaison pivot d'axe Y de l'objet de référence en tenant compte du nombre de degrés de liberté.

Le protocole expérimental reste le même que dans la section 3. Nous souhaitons, dans un premier temps, identifier une liaison pivot le long de la simulation (cf. fig. 4.8 page 138) et déterminer à quel moment la condition choisie 4.29 ne la reconnaît plus. Ce mouvement est

choisi du fait que les paramètres de mouvement extraits, notamment de translation, sont plus complexes que ceux d'une liaison glissière. En effet, quatre paramètres notables varient au cours du temps (l'angle de rotation, le vecteur de rotation suivant l'axe Y , le vecteur de translation suivant les axes X et Z) (cf. fig. 4.9 page 138).

Nos torseurs sont enregistrés de la manière suivante :

- une valeur 1 ou 0 est notée si un degré de liberté suivant un certain axe est possible,
- le nombre de degrés de liberté pour la rotation et pour la translation, correspondant à la somme de chacune des colonnes du torseur est également noté.

Afin d'identifier la liaison en cours et en se basant sur les courbes obtenues dans les figures 4.14a et 4.14b, les seuils choisis pour l'équation 4.29 page 143 est de 0.01 pour la vitesse angulaire et de 0.2 pour la vitesse linéaire. Ceci nous permet de déterminer le nombre de degrés de liberté à chaque instant t (cf. fig. 4.14c). Ce nombre restreint les liaisons de la base de connaissance à comparer. Les axes mis en jeu dans la liaison durant la simulation sont identifiées et comparé aux liaisons potentielles enregistrées. Nous pouvons ainsi reconnaître une liaison pivot d'axe Y de l'objet de référence 4.14d sur une grande partie de l'action. Cependant, celle-ci n'a pu être reconnue au début de l'assemblage du fait que des degrés de liberté suivant les translations ont été comptabilisé. Ceci n'est, en revanche, pas très pénalisant dans ce cas puisque le mouvement relatif des pièces n'a pas encore été véritablement extrait et que la reconnaissance de la liaison pivot est maintenu jusqu'à la "saturation" des paramètres.

En revanche, des ajustements sont nécessaires afin d'éviter une reconnaissance d'une action lorsqu'elle n'a plus lieu comme c'est le cas après la phase d'assemblage, dans la figure 4.14d.

4.3.3 Discussion

Dans cette section, nous nous sommes intéressés à la question de la reconnaissance des actions de l'utilisateur et à celle des erreurs pour survenir. Pour l'instant, notre méthode permet d'identifier des liaisons simples (pivot et glissière) en cours et enregistrée dans notre base de connaissance. En se basant sur le nombre de degrés de liberté possibles à un instant t , nous pouvons, dans un premier temps, réduire rapidement le champ recherche des liaisons entre les pièces. En tenant compte des axes mis en jeu dans le mouvement, il est ainsi possible de reconnaître la liaison mécanique.

Cependant cette méthode nécessite encore des ajustements afin d'interrompre la recherche lorsque les mouvements relatifs n'ont plus lieu. Par ailleurs, il faudrait également la rendre plus générique pour reconnaître de manière fiable n'importe quelle liaison, y compris la liaison hélicoïdale. De plus, une reconnaissance plus globale et plus variée, sur des liaisons plus complexes et en temps réel, sont à envisager.

4.4 Bilan

Dans ce chapitre, nous avons exploré une autre définition de l'assemblage afin de caractériser les mouvements relatifs entre les pièces. En considérant une configuration adéquate et en l'associant à une bonne gestion du *gimbal lock*, nous pouvons identifier correctement les paramètres de mouvement. Celle-ci consiste extraire les vecteurs de translation et de rotation et l'angle de rotation de l'objet mobile par rapport à l'objet de référence qui est statique. Elle est plus facilement interprétable et à transposer que la configuration consistant à relever les mouvements de l'objet mobile à chaque instant par rapport à sa position initiale. Cette dernière correspondra plus à une visualisation de la vitesse de cet objet au cours du temps.

Comme les mesures enregistrées dans les conditions réelles sont très bruitées, nous nous sommes demandés s'il n'y avait pas de variables du *pipeline* de RA qui peuvent les influencer. Il en ressort que le nombre de primitives de chaque objet ainsi que la qualité moyenne des mises en correspondances pouvaient les faire varier de manière importante. Ceci est explicable par le fait que ces variables interviennent directement dans l'estimation de la pose si l'algorithme du RANSAC ne les supprime pas. Nous avons, par la suite, appris un modèle permettant de prédire à quel point ces variables vont influencer nos mesures. Nous avons proposé une méthode afin de réduire le bruit dans les valeurs des paramètres de mouvement en se basant sur l'influence des variables retenues en entrée et de certaines données de l'image précédente. Nous notons une amélioration surtout importante lorsque la véritable liaison n'a pas encore eu lieu et une rectification de mesures au cours du temps. Ceci nous permet ainsi d'exploiter ces données pour identifier les liaisons en cours.

En effectuant un changement d'origine du repère sur le point central de la liaison mécanique enregistré dans la base de connaissance, il est alors possible de reconnaître cette liaison à chaque instant en conservant la première configuration. Nous y avons présenté notre approche pour réaliser cette reconnaissance. Pour le moment, seules les liaisons simples comme la glissière et le pivot sont correctement identifiées au cours de l'exécution de la tâche d'assemblage. Il est nécessaire de vérifier notre approche sur d'autres liaisons afin de la valider complètement.

Dans le chapitre suivant, nous présentons l'intégration sous *ARCS (Augmented Reality Component System)* des composants présentées dans ce chapitre et le chapitre précédent afin de faire fonctionner le système sur n'importe quel appareil portable (tablette et smartphone) et pour des simulations dans des environnements plus variées et par d'autres utilisateurs.

Chapitre 5

Intégration au framework ARCS (*Augmented Reality Component System*)

ARCS (*Augmented Reality Component System*) est framework libre sous licence publique générale GNU, développé par Jean-Yves DIDIER du laboratoire IBISC [Didier et al., 2009, Didier et al., 2012]. Il permet le prototypage rapide d'applications de réalité augmentée. Cet outil se base sur le principe de la programmation orientée composants. L'objectif est de favoriser, de manière simple et rapide, une réutilisation des applications, des méthodes et des algorithmes implémentés, l'intégration de nouveaux algorithmes, la réalisation des tests et le déploiement des applications. Cette programmation modulaire améliore ainsi la lisibilité, la maintenance et la ré-utilisabilité du code. Celui-ci fonctionne suivant un principe de signal-slot. Une composante, contenant des slots (des fonctions recevant des signaux), exécute une tâche spécifique puis envoie un signal à un slot d'une autre composante, et ainsi de suite, constituant ainsi un réseau de signaux-slots, donc de communication entre les composantes, et représentant une chaîne de traitement.

Il a d'abord été développé en C++ puis reprogrammé en JavaScript sous le nom de *ARCS.js*. Cette version de ARCS permet de tirer parti des dernières évolutions du standard HTML et des API JavaScript associées. Il permet ainsi de faire fonctionner des applications de réalité augmentée sous *node.js* et sur n'importe quel navigateur. Il permet ainsi d'exécuter une application sous *ARCS.js* sur n'importe quel support de RA, aussi bien sur un ordinateur qu'un smartphone ou une tablette de n'importe quelle marque. De cette façon, il n'est plus nécessaire de passer par une phase de ré-implémentation ou d'encapsulation du code d'une application de RA. Nous nous affranchissons de toutes contraintes liées au type de matériel à utiliser pour nos simulations.

Dans cette partie, il s'agit d'intégrer le processus complet pour la reconnaissance des objets 3D, des actions de l'utilisateur et de ses erreurs dans le *framework* ARCS afin de réaliser, par la

suite, des tests en temps réel sur des tablettes, smartphones ou lunettes de RA.

5.1 Établissement de la base de modèles et des scénarios

Dans une de nos expérimentations (cf. section 5.2), nous gardons toujours les cubes utilisées lors de l'établissement des concepts, c'est-à-dire ceux décrits dans le chapitre 3. Dans une autre (cf. section 5.3 page 154), notre base de connaissance est constitué des modèles 3D de chaque partie constituant l'objet global (ici d'une imprimante).

Les liaisons entre les différentes pièces sont enregistrées comme dans la section 4.3.2.

Nous enregistrons également la séquence des mouvements devant être reconnu associée aux simulations 3D entre les pièces à afficher à l'utilisateur.

5.2 Mise en place du système de reconnaissance d'actions

Nous avons, d'abord, implémenté sous ARCS la phase d'estimation de l'erreur de reprojexion et d'extraction des paramètres de mouvement. Les composantes suivantes ont été intégrées et connectés suivant la figure 5.1 :

- *Animator* : un signal est envoyé au composant *Video* indiquant la fin de traitement de l'image précédente.
- *Video* : il reçoit un signal du composant *Animator* permettant ainsi de récupérer un image de la caméra et envoie un signal au composant *FeatureSceneExtractor*.
- *FeatureSceneExtractor* : il reçoit un signal du composant *Video* et réalise l'extraction et la description des points d'intérêt SURF de l'image envoyée par le composant *Video*. A la fin de ce processus, il envoie un signal au composant *Matching*.
- *FeatureCubeExtractor* : il est appelé une seule fois à l'initialisation du système. Il récupère les images des cubes contenues dans notre base de connaissance. Il en effectue l'extraction et la description des points d'intérêt SURF ainsi que leur position 3D. Puis, il envoie un signal au composant *Matching*.
- *Matching* : il reçoit les signaux des composants *FeatureCubeExtractor* et *FeatureSceneExtractor* ainsi que l'ensemble des points d'intérêt et des descripteurs associés provenant de ces composants. Il met en correspondance les points d'intérêt 2D de l'image en cours et ceux 3D de la base de connaissance. Puis, il envoie un signal au composant *poseEstimate*.

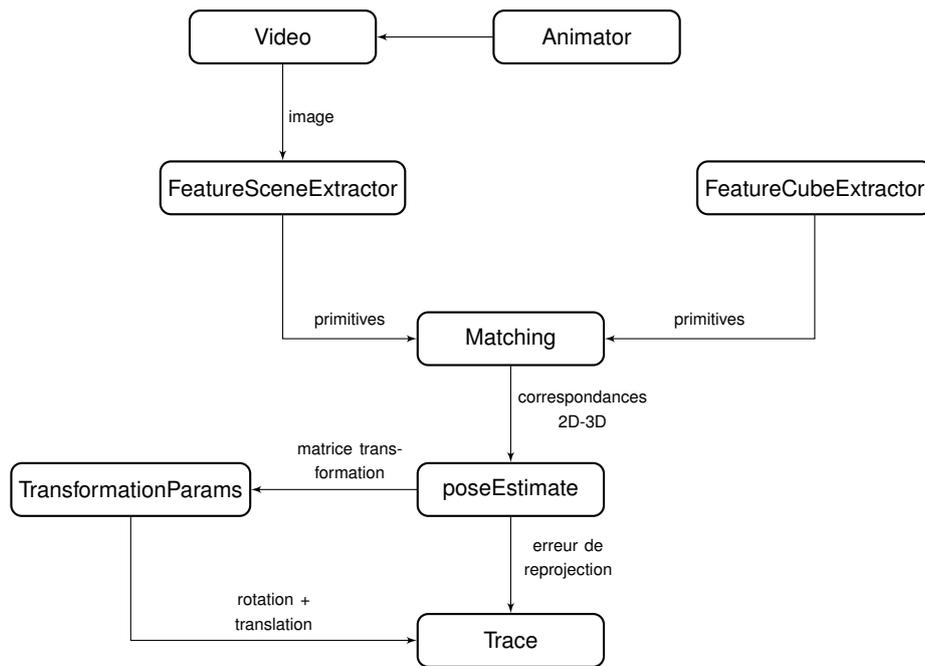


FIGURE 5.1 – Connexion entre les composants ARCS pour le tracé de l’erreur de reprojection et des paramètres de mouvement au cours du temps.

- *poseEstimate* : il reçoit un signal du composant *Matching* ainsi que les mises en correspondance 2D-3D. Il estime la pose de la caméra, identifie la matrice de transformation entre les deux pièces (cf. section 4 page 113) et enregistre l’erreur de reprojection (cf. section 3 page 83). Puis il envoie un signal aux composants *Trace* d’une part et *TransformationParams* d’autre part.
- *TransformationParams* : il reçoit un signal du composant *poseEstimate* ainsi que la matrice de transformation entre les deux pièces. Il en extrait les paramètres de mouvement, c’est-à-dire ceux de la rotation et ceux de la translation (cf. section 4 page 113). Il envoie, ensuite, un signal au composant *Trace*.
- *Trace* : il reçoit des signaux du composant *poseEstimate* avec la valeur de l’erreur de reprojection à l’instant t et du composant *TransformationParams* avec les paramètres de mouvement relatif à l’instant t . Il affiche ainsi sur le navigateur l’évolution de l’erreur de reprojection et du mouvement à chaque instant t (cf. fig.5.2).

Le fait d’implémenter le système sous JavaScript qui est un langage interprété a entraîné un ralentissement de ses performances. La vitesse d’exécution était de 2 s/image, ce qui est extrêmement lent pour obtenir une fluidité dans une vidéo de 24 images/s. La lenteur observée ne permettait pas à ce moment de tester, en temps réel, la reconnaissance et le contrôle des

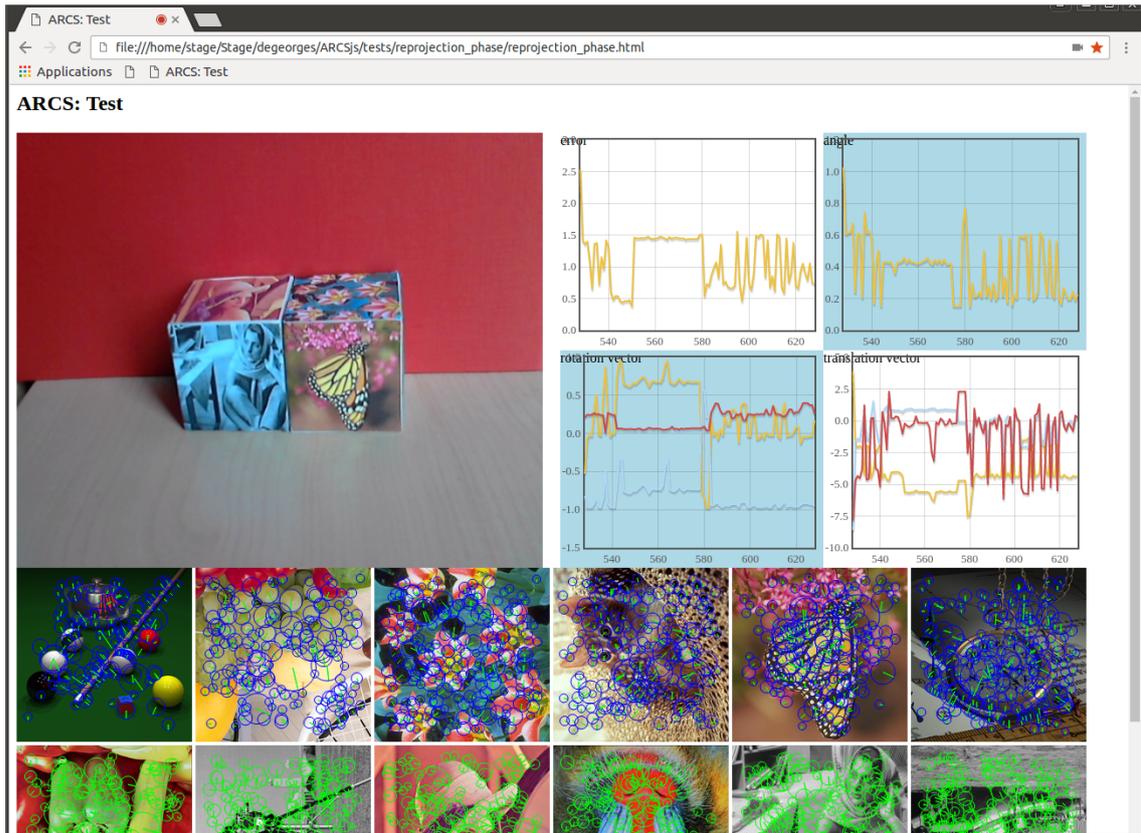


FIGURE 5.2 – Capture d’écran du navigateur.

actions de l’utilisateur. En effet, afin d’obtenir des valeurs de mouvement cohérentes, il fallait effectuer un mouvement très lent de la pièce mobile. Si la vitesse de déplacement était trop rapide, le système n’a pas le temps d’effectuer le processus d’estimation des paramètres de mouvement pour chacune des images acquises et renvoyait ainsi parfois des valeurs erronées (cf. fig. 5.2). Il était donc nécessaire d’effectuer une optimisation du processus implémenté. L’objectif serait de s’approcher d’une vitesse moyenne d’exécution de 0.04 s/images afin de maintenir la fluidité de l’action qui est un impératif pour une application dans l’industrie.

Dans un premier temps, nous avons observé que le composant *Matching* prenait 82% du temps d’exécution. La première stratégie a ainsi été de rechercher des méthodes permettant d’accélérer le processus de mise en correspondance.

Optimisation du composant *Matching* : Nous avons testé les approches suivantes afin d’optimiser le composant *Matching* :

- *Suppression de la fonction Math.sqrt* : Comme, lors de la mise en correspondance il s’agit de calculer la distance euclidienne entre deux descripteurs, cette fonction est appelée à chaque couple possible. Or la fonction *Math.sqrt* appartient à la bibliothèque mathématique de JavaScript qui est une bibliothèque externe dont l’appel est très coûteux en JavaScript. Par ailleurs, le calcul de la racine carrée d’un nombre lui-même est

toujours plus coûteux que le calcul de son carré, peu importe le langage. Ainsi, nous optons pour une estimation de la distance euclidienne au carré des couples de descripteurs qui conserve la monotonie de la fonction racine carrée. Cette suppression permet d'obtenir un gain de temps de 0.7 en moyenne avec ainsi un temps d'exécution de 1.3 s/image.

- *Utilisation de kd-tree* : Les arbres à k -dimension ou *kd-tree* sont des structures de données permettant une recherche plus rapide des plus proches voisins pour des vecteurs de dimension k comparé à celle linéaire qui est utilisé pour les correspondances. Il s'agit d'un arbre binaire dans lequel les vecteurs sont stockés dans les nœuds. Lors de l'ajout d'un nouvel élément, nous partons de la racine puis comparons la $i^{\text{ème}}$ coordonnée du vecteur à insérer à celle du nœud, avec i la profondeur du nœud. Selon que la valeur soit plus petite ou plus grande, nous descendons respectivement dans le fils gauche ou droit et nous réitérons jusqu'à arriver à une feuille. Nous remplaçons alors la feuille par un nœud contenant le vecteur à insérer et une feuille pour chaque fils. Si la profondeur dépasse la dimension k du vecteur, nous la calculons modulo k .

La recherche de plus proche voisin revient à descendre dans l'arbre comme pour insérer un nouvel élément, puis lors de la remontée, tester pour chaque nœud s'il est nécessaire de descendre dans l'autre branche. Comme tous les points construisant le même arbre sont générés au même moment, nous peut nous arranger pour que l'arbre construit soit aussi équilibré que possible. Dans le meilleur des cas, la recherche dans les points de l'image sera à présent logarithmique au lieu de linéaire. Cependant avec cette modification, aucun gain de temps n'est observé. Pour chaque test, l'arbre est entièrement parcouru. Cela s'explique en partie par le fait que le nombre de dimensions des descripteurs est très élevé et parce que, comme les vecteurs sont normés, leurs composantes ont toutes des valeurs proches. Les vecteurs n'ont donc pas de composantes principales pouvant permettant de les séparer efficacement. Ainsi les *kd-tree* n'ont pas permis de réduire le temps de calcul du processus de recherche des correspondances.

- *Optimisation des mises en correspondance symétriques* : Vu que la recherche des correspondances est linéaire, un point 2D de l'image sera associé à un dernier point 3D de la base de connaissance qui sera le plus similaire. Ceci peut être problématique s'il y a d'autres correspondances avec des distances égales ou très proches et si le système conserve de mauvaises correspondances. Ces *outliers* peuvent avoir une influence sur l'estimation de la pose qui doit être suffisamment fiable afin d'estimer les paramètres de mouvement relatif entre les pièces. Nous avons alors opté pour une mise en correspondance symétrique, c'est-à-dire que nous conservions uniquement l'ensemble commun

des correspondances issues de la recherche de l'ensemble des points de l'image dans l'ensemble des points de la base de données et de la recherche inverse, c'est-à-dire de la base de données à l'image. Cependant ces recherches sont également coûteuses. La stratégie adoptée serait de vérifier si le premier ensemble de correspondances de la base de donnée vers l'image contient au moins 4 points pour chaque image représentant les cubes dans nos expérimentations. Ce nombre correspond au minimum de points nécessaires pour estimer la pose de la caméra. Si ce n'est pas le cas, il n'est pas nécessaire d'effectuer la recherche inverse sur ces images. En revanche, si c'est le cas, la recherche inverse ne concernera que les couples sélectionnés lors de la première recherche. La première condition permet de gagner 0.5, tandis que la seconde 0.1 s. La première condition fait gagner beaucoup plus de temps que la seconde du fait que sur les douze faces du cube enregistrées dans notre base, six au grand maximum peuvent être visibles. Ceci permet ainsi de diviser le temps de calcul pour cette partie par deux puisque la deuxième boucle n'est pas effectuée pour l'ensemble des images de référence.

- *Calcul d'une distance partielle* : La distance euclidienne entre les descripteurs réalise une somme de l'ensemble des 64 éléments constituant les descripteurs. De plus, le composant fait appel à cette fonction un grand nombre de fois, proportionnellement au nombre de couples possibles entre les points de la base de données et les points de l'image. Il est possible de subdiviser, de manière adéquate, cette somme afin de comparer la somme partielle avec la distance minimale enregistrée. Il s'avère que réaliser ce test au milieu de l'itération sur la somme permet de gagner 0.25 s.
- *Réduction du nombre d'images de référence à tester* : Si nous supposons que les mouvements entre deux images sont suffisamment lent pour considérer que celles-ci sont similaires, nous pouvons alors également supposer qu'entre deux images récupérées par la caméra, les images de référence identifiées sont les mêmes. Ainsi, lors de la mise en correspondance, nous ne tenons compte que des images de références reconnues à l'image précédente. Une mise en jour est effectuée toutes les dix images afin de tenir compte des changements pouvant survenir et si aucune image de référence n'a été identifiée à un instant t . Nous notons, alors, un ralentissement pratiquement toutes les dix images correspondant à la mise à jour.

En tenant compte de toutes les modifications apportées, le temps moyen d'exécution descend de 2 s à 0.5 s. Le composant *Matching* occupe 43.6% du temps de calcul. Même si les améliorations effectuées ont permis de réduire de près d'un quart le temps de calcul, celui-ci ne permet toujours pas de réaliser une simulation en temps réel. Nous avons alors opté pour

l'utilisation d'*asm.js*.

Utilisation de *asm.js* : *asm.js* est un langage intermédiaire constituant un sous-ensemble du JavaScript. Il permet d'obtenir des performances similaires aux langages avec gestion de la mémoire comme le C, ce au prix de certaines limitations et restrictions (mise en place d'une pile, calcul de l'adressage ...). Ces limitations font en sorte qu'il est plus confortable d'utiliser un *transpiler* du C vers le JavaScript pour écrire les fonctions. Ainsi, certaines fonctions du code JavaScript ont été réécrites en C puis compilées avec *emscripten* qui génère un fichier *.js*. C'est ce fichier qui sera ajouté au script permettant ainsi une meilleure gestion de la mémoire par rapport au *garbage collector* de JavaScript.

Un certain nombre de fonctions ont été ainsi réécrites, notamment la fonction permettant le calcul des descripteurs qui prenait 39.4% du temps de calcul après les améliorations sur le composant *Matching*. Ce passage à *asm.js* de cette fonction uniquement, en particulier, permet de gagner 0.15 s réduisant ainsi le pourcentage d'occupation de cette fonction à 10% sur l'ensemble du processus. La réécriture d'autres fonctions du composant *Matching*, y compris celle calculant les descripteurs permet de gagner 0.2 s globalement. Ainsi, le temps d'exécution passe à 0.3 s.

Même si cette progression est intéressante, passer chaque fonction sous *asm.js* peut être fastidieux, notamment lorsque les données manipulées sont des tableaux ou des objets, ce qui est le cas dans le système implémenté. Ceci est essentiellement dû au passage des fonctions en C qui ne peuvent avoir comme attribut des tableaux ou des objets, ni les modifier ou les renvoyer. Afin de contourner cet obstacle, il est possible de jouer sur la gestion des allocations mémoire de ces données. Cependant, nous risquons de perdre du temps dans cette étape s'il faut allouer et désallouer de la mémoire pour chacune des fonctions au détriment de celle que nous avons gagné dans leur exécution. Il s'agira donc, par la suite, d'identifier les phases du *pipeline* permettant d'éviter les problèmes liés à l'utilisation des tableaux et des objets et de le réécrire en C.

5.3 Mise en place d'un système de reconnaissance des pièces peu texturées

D'un autre côté, nous avons également mis en place sous *ARCS.js* un système pour reconnaître des pièces peu texturées notamment celles des pièces d'une imprimante. L'objectif était de pouvoir effectuer des tests des différentes approches proposées sur une situation de maintenance réelle sur une imprimante en affichant automatiquement les instructions à l'utilisateur et de le notifier en cas d'erreurs.

Nous avons choisi, dans un premier temps, d'implémenter la méthode de [Weiss and Ray, 2001] afin de reconnaître des objets peu texturés à partir des modèles 3D des objets en se basant sur les invariants projectifs. Celle-ci permet non seulement d'identifier les objets mais également de déterminer le point de vue associé en extrayant des primitives pour l'estimation de la pose.

Rappel bref de la méthode :

L'approche proposée par [Weiss and Ray, 2001] est constituée de trois étapes principales :

- La première consiste à extraire des primitives 3D de l'objet 3D, notamment ses sommets et à calculer ses invariants, c'est-à-dire à déterminer les points qui sont invariants aux transformations affines. Ce processus sera effectué une seule fois, à l'initialisation du système.
- La deuxième consiste à extraire les primitives 2D de l'image récupérée par la caméra et à calculer ses invariants à chaque instant t .
- La troisième correspond à la mise en correspondance des invariants 2D et 3D permettant, par la suite, l'estimation de la pose de la caméra.

Extraction des invariants du modèle 3D : Un composant a été implémenté afin d'extraire les invariants de l'objet 3D. Le modèle 3D est contenu dans un fichier *.obj* qui regroupe l'ensemble des points de l'objet et des triangles les reliant. Même si l'objet peut paraître simple, les points extraits peuvent être nombreux (1000 points avec certains de nos modèles) si des arrondis existent dans le modèle considéré et ainsi augmenter le temps de traitement. Même si c'est une étape qui est effectuée à l'initialisation, il est important que ce temps ne soit pas trop lent afin de ne pas décourager l'utilisateur. Nous avons ainsi, dans un premier temps, réduit ce nombre en éliminant, d'abord, tous les triangles pouvant appartenir à une même face et en identifiant les sommets potentiels selon l'appartenance d'un point à trois faces ou plus différentes (cf. fig. 5.3 page suivante). Comme l'objectif est uniquement de réduire le nombre de points à considérer pour le traitement, il n'est pas nécessaire d'effectuer des tests supplémentaires afin de vérifier la cohérence spatiale des "sommets" extraits.

Par la suite, il s'agit d'identifier les invariants 3D de l'objet. Pour cela, les bases possibles de l'ensemble des points 3D sont identifiées et inclus dans des groupements de cinq points. Ces bases permettent de reconnaître un objet suivant n'importe quel point de vue. Chaque groupe-

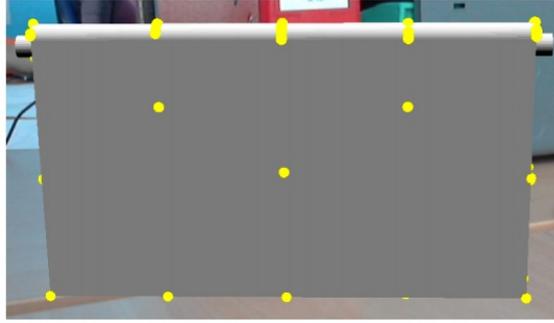


FIGURE 5.3 – Points retenus d'un élément 3D de l'imprimante.

ment i de cinq points $(x_j, y_j, z_j)_{j \in \{1..5\}}$ constitue la matrice A_i :

$$A_i = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ y_1 & y_2 & y_3 & y_4 & y_5 \\ z_1 & z_2 & z_3 & z_4 & z_5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (5.1)$$

Ces derniers sont ensuite associés à des vecteurs I_i considérés comme invariants :

$$I_i = \begin{pmatrix} M_1 & M_2 & M_3 \\ M_5 & M_5 & M_5 \end{pmatrix} \quad (5.2)$$

avec M_j le déterminant de la matrice A_i en supprimant la $j^{\text{ème}}$ colonne. Cette métrique est introduite parce qu'elle est relativement invariante suivant les transformations affines et projectives.

Extraction des invariants 2D de l'image : Les points d'intérêt de l'image à l'aide du détecteur de Harris [Harris and Stephens, 1988]. Le détecteur de Canny [Canny, 1986] et la transformée de Hough [Duda and Hart, 1972] sont utilisés afin de réduire le nombre de points d'intérêt à considérer pour l'extraction des invariants 2D.

Comme pour le cas 3D, des groupements de 5 points sont effectués. Chaque groupement est représenté par la matrice B_k :

$$B_k = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ y_1 & y_2 & y_3 & y_4 & y_5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (5.3)$$

Ils sont ensuite associés au vecteur v suivant :

$$v = \begin{pmatrix} i_{12} & i_{13} & i_{25} & i_{35} \end{pmatrix} = \begin{pmatrix} \frac{m_{12}}{m_{15}} & \frac{m_{13}}{m_{15}} & \frac{m_{25}}{m_{15}} & \frac{m_{35}}{m_{15}} \end{pmatrix} \quad (5.4)$$

avec m_{lp} le déterminant de la matrice B_k en éliminant les $l^{\text{ème}}$ et $p^{\text{ème}}$ colonnes.

Mise en correspondance des invariants :

Les invariants 2D et 3D sont liés par les équations suivantes :

$$\begin{cases} v_0 + I_x v_2 - I_y = 0 \\ v_2 + I_x v_3 - I_z = 0 \end{cases} \quad (5.5)$$

avec v le vecteur invariant 2D et I le vecteur invariant 3D.

Comme pour la section 5.2 page 149, le système de reconnaissance est composé des composants suivantes :

- *Animator*
- *Video* qui envoie un signal au composant *FeatureSceneExtractor*.
- *FeatureSceneExtractor* : il reçoit un signal du composant *Video* et réalise l'extraction des points d'intérêt de Harris de l'image envoyée et détermine les invariants 2D par le composant *Video*. A la fin de ce processus, il envoie un signal au composant *Matching*.
- *FeatureModelExtractor* : il est appelé une seule fois à l'initialisation du système. Il récupère les données du modèle 3D et en extrait les invariants 3D. Puis, il envoie un signal au composant *Matching*.
- *Matching* : il reçoit les signaux des composants *FeatureCubeExtractor* et *FeatureSceneExtractor* ainsi que l'ensemble des points d'intérêt et des invariants associés provenant de ces composants. Il met en correspondance les invariants 2D de l'image en cours et ceux 3D de la base de connaissance.

Le système ne permettait pas une reconnaissance en temps réel. La vitesse d'exécution était de 6 images par seconde. Il nécessitait ainsi une optimisation du code.

Comme la section précédente, nous avons testé nos améliorations en utilisant *asm.js* pour certaines fonctions. Cela a permis de gagner 4 images/s et de passer ainsi à 12 images/s. Cependant cette vitesse reste encore faible pour une simulation en temps réel.

5.4 Bilan

Dans cette partie, nous avons intégré sous le *framework* ARCS une partie du système consistant à reconnaître les actions de l'utilisateur, à identifier ses erreurs et à afficher les instructions pertinentes au moment adéquat et une partie consistant à reconnaître des objets très peu texturés à partir de leur modèle 3D. Cependant les temps d'exécution de chacune des parties restent encore très longs pour des simulations en temps réel. Ceci est surtout dû au langage utilisé qu'est le JavaScript et également à la manière de coder un certain nombre de fonctions. Des améliorations ont été apportées afin de réduire cette durée en optimisant ces fonctions. Mais les gains de temps obtenus ne sont pas suffisants pour du temps réel. Des ajustements sont encore nécessaires.

Des approches se basant sur l'utilisation de *asm.js* ont été testées. Certaines fonctions posant problème ont été passées sous C pour pouvoir être interprétées sous *asm.js* tout en profitant des avantages liés au C. Les résultats obtenus sont honorables mais des améliorations sont possibles afin d'éviter de perdre du temps dans des allocations et désallocations de mémoire du fait du recours aux tableaux et aux objets. Des stratégies sont possibles afin d'en réduire ces allers-retours notamment en reprenant les parties du code manipulant les tableaux de les objets.

Conclusion et perspectives

Conclusion

Dans ce manuscrit, nous avons proposé dans un premier temps, notre taxonomie des interactions et des interactivités entre l'utilisateur, les objets réels et les objets virtuels. Celle-ci est basée sur le degré d'implication de chaque élément par rapport à l'autre.

Dans un deuxième temps, nous avons défini l'assemblage comme une structure rigide. Cette définition nous a permis de mettre les bases théoriques permettant d'utiliser une métrique telle que l'erreur de reprojection afin d'identifier les phases d'un assemblage. Nous avons étudié deux configurations différentes (globale et indépendante) des modèles 3D enregistrés à associer à cette métrique. En exploitant cette métrique avec une configuration globale, nous avons montré que les périodes d'assemblage ou de désassemblage pouvaient être caractérisé par l'erreur de reprojection.

Nous avons, ensuite, proposé deux méthodes que nous avons appelées KSUM et KSUMratio permettant de détecter l'instant où le changement de phases se produisait. Nous les avons comparées avec d'autres algorithmes de détection de ruptures. Il en ressortait que notre approche produit des résultats presque similaires à celles obtenus avec le CUSUM (qui est un algorithme très utilisé pour la détection de ruptures) en termes de délai de détection. De plus, en termes de temps d'exécution, KSUM présentait une meilleure performance. C'est celle que nous avons retenue pour la suite. En identifiant quand un désassemblage débute, il est alors possible de fournir à l'utilisateur l'instruction suivante de manière automatique sans son intervention sur le système.

Afin d'identifier le mouvement relatif entre les pièces, il a fallu redéfinir l'assemblage comme un structure dynamique dont les pièces conservent le même nombre de degrés de liberté dans le temps et suivant les mêmes axes. En étudiant les mouvements relatifs des objets mobiles par rapport à l'objet de référence qui est statique et en réécrivant la matrice de transformation sous forme d'un vecteur de translation et d'un couple axe-angle, les mesures recueillies

permettaient d'observer les variations des paramètres de mouvement telles qu'elles étaient attendues. Cependant, celles-ci étaient très bruitées dans les conditions réelles du fait des potentielles variations de luminosité, du flou de mouvement produit par le déplacement des pièces ou de la caméra. Nous nous sommes alors intéressés aux variables en entrée qui pouvaient considérablement les influencer. Le nombre de primitives de chaque objet et la qualité moyenne des correspondances pour chaque objet apparaissaient comme celle pouvant induire le bruit dans nos valeurs. Nous avons ainsi proposé une méthode de réduction basée sur une méthode d'apprentissage sur ces données, notamment le SVM. Cette dernière nous indiquait à quel point les données enregistrées en entrée pouvaient introduire du bruit. Nous avons ainsi formulé des équations permettant de corriger les paramètres de mouvement. Nous avons obtenu une nette amélioration lorsque l'assemblage n'avait pas débuté et un léger lissage des mesures extraites. Cependant, nous pouvions noter que les valeurs les plus aberrantes, donc celle avec une valeur très au-dessus des autres valeurs précédentes étaient corrigées sur nos séquences.

En récupérant ainsi les paramètres de mouvement relatif extraits après la phase de réduction de bruit, nous pouvions dans une certaine mesure identifier la liaison en cours. Nous avons proposé une approche basée sur la détermination du nombre de degrés de liberté sur chacun. Celle-ci nous a permis, pour l'instant de reconnaître des liaisons simples (telles que le pivot et la glissière) dans la majorité de la séquence ainsi que certaines erreurs se produisant. Cependant, des améliorations sont encore nécessaires afin que l'approche soit fiable et robuste.

D'un autre côté, nous avons intégré une partie de notre *pipeline* dans le *framework* ARCS.js. Ce dernier nous permettait de nous affranchir des contraintes de développement liées au matériel utilisé, c'est-à-dire de permettre une portabilité du code sur n'importe quels tablettes ou smartphones et de réaliser un prototypage rapide de notre système. Cependant, comme le langage JavaScript est interprété, il était nécessaire d'établir un certain nombre de stratégies afin d'obtenir une application en temps réel. Les modifications apportées ont considérablement réduit le temps d'exécution. En revanche, d'autres améliorations peuvent encore être effectuées notamment en passant la majorité du code sous *asm.js*.

Les approches que nous avons établi le long du manuscrit peuvent être généralisées et testées sur des objets de forme différente que celle utilisée lors de nos expérimentations. Comme l'erreur de reprojexion et les matrices de transformation extraites sont basées sur les matrices de projection déduites de l'estimation de la pose, la reconnaissance des objets, l'extraction des primitives et leur mise en correspondance sont donc supposée être suffisamment robuste et fiable pour obtenir une pose correcte.

Perspectives

L'objectif final de ce système est de pouvoir réaliser la certification de la procédure de maintenance par des opérateurs. Il s'agit d'obtenir un système de RA capable de suivre les tâches de l'utilisateur et de certifier si les actions que ce dernier effectue sont en adéquation avec ce qui est prévu par la procédure, de notifier et enregistrer non seulement la présence d'une erreur mais également le type d'erreur précisément de manière automatique. C'est une problématique très importante notamment en aéronautique actuellement.

Dans le but, d'atteindre cet objectif et d'implémenter un système robuste, plusieurs perspectives peuvent être envisagées. La première consiste à concevoir le système complet de détection de l'assemblage et de reconnaissance des liaisons mécaniques sous *ARCS.js* en tenant compte de l'utilisation de *asm.js*. Dans un premier temps, il est possible de réaliser des tests sur des objets simples afin de vérifier le bon déroulement du système et de tester s'il reconnaît la séquence du scénario établi, s'il affiche les bonnes instructions au bon moment, s'il identifie correctement les erreurs et les notifie à l'utilisateur.

Une expérimentation sur des pièces mécaniques réelles doit être effectuée afin de valider le champ d'application des méthodes proposées pour le contexte industriel. Une série de tests avec des utilisateurs variés peut être également effectuée afin d'ajuster correctement les paramètres spécifiés dans nos méthodes pour des applications en temps réel.

Notre approche de reconnaissance des liaisons mécaniques peut être améliorée afin de la rendre générique, plus robuste et fiable quelque soit les conditions d'expérimentations.

D'autres problématiques telles que celles liées à l'acceptabilité, à l'ergonomie et aux occultations peuvent être introduit afin d'en faire un outil utilisé par des opérateurs de maintenance. De plus, même si globalement, l'approche que nous avons proposée afin de détecter quand un assemblage a lieu dans le but de fournir des instructions de maintenance au moment adéquat, et afin de reconnaître le type de liaisons en cours et les erreurs de l'utilisateur n'est pas remise en question si d'autres méthodes de suivi sont utilisées, il peut être intéressant d'explorer d'autres dispositifs de visualisation tels que la tablette Tango qui permet de récupérer également les informations de profondeur.

Bibliographie

- [Ret, 1998] (1998). Virtual retinal display technology. In *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, volume 1, pages D35/1–D35/6 vol.1.
- [Ababsa et al., 2012] Ababsa, F., Zendjebil, I., Didier, J.-Y., Pouderoux, J., and Vairon, J. (2012). Outdoor augmented reality system for geological applications. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pages 416–421.
- [Abdi et al., 1995] Abdi, H., Valentin, D., Edelman, B., and O’Toole, A. J. (1995). More about the difference between men and women : Evidence from linear neural networks and the principal-component approach.
- [Agrawala et al., 2003] Agrawala, M., Phan, D., Heiser, J., Haymaker, J., Klingner, J., Hanrahan, P., and Tversky, B. (2003). Designing effective step-by-step assembly instructions. *ACM Trans. Graph.*, 22(3) :828–837.
- [Akman et al., 2012] Akman, O., Jonker, P. P., TU Delft : Mechanical, Maritime and Materials Engineering : BioMechanical Engineering, and TU Delft, Delft University of Technology (2012). *Robust Augmented Reality*. PhD thesis.
- [Alexandre, 2010] Alexandre, L. A. (2010). Gender recognition : A multiscale decision fusion approach. *Pattern Recognition Letters*, 31(11) :1422 – 1427.
- [Ameller et al., 2002] Ameller, M.-A., Quan, L., and Triggs, B. (2002). Le Calcul de Pose : de nouvelles méthodes matricielles. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA ’02)*, volume 1, pages 39–47, Angers, France.
- [Ardakany and Joula, 2012] Ardakany, A. R. and Joula, A. M. (2012). Gender recognition based on edge histogram. *International Journal of Computer Theory and Engineering*, 4(2) :127–130.
- [Arth et al., 2015] Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, R., Schmalstieg, D., and Wagner, D. (2015). The history of mobile augmented reality developments in mobile ar over the last almost 50 years.
- [Azuma et al., 2001] Azuma, R., Bailiot, Y., and Feiner, S. (2001). Recent advances in augmented reality.

- [Azuma et al., 2006] Azuma, R., Neely, H., Daily, M., and Leonard, J. (2006). Performance analysis of an outdoor augmented reality tracking system that relies upon a few mobile beacons. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 101–104. IEEE Computer Society.
- [Azuma, 1997] Azuma, R. T. (1997). A survey of augmented reality. *Presence : Teleoperators and Virtual Environments*, 6(4) :355–385.
- [Baird and Barfield, 1999] Baird, K. M. and Barfield, W. (1999). Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality*, 4(4) :250–259.
- [Basseville and Nikiforov, 1993] Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes : Theory and Application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [Basseville and Nikiforov, 2002] Basseville, M. and Nikiforov, I. V. (2002). Fault isolation for diagnosis : nuisance rejection and multiple hypotheses testing. Research report, INRIA.
- [Bauer et al., 2001] Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., Sandor, C., and Wagner, M. (2001). Design of a component-based augmented reality framework. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 45–54.
- [Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3) :346 – 359. Similarity Matching in Computer Vision and Multimedia.
- [Billinghurst, 2005] Billinghurst, M. (2005). Augmented reality interfaces for visual analytics.
- [Billinghurst et al., 2015] Billinghurst, M., Clark, A., and Lee, G. (2015). A survey of augmented reality. *Found. Trends Hum.-Comput. Interact.*, 8(2-3) :73–272.
- [Billinghurst et al., 2008] Billinghurst, M., Hakkarainen, M., and Woodward, C. (2008). Augmented assembly using a mobile phone. In *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, MUM '08, pages 84–87, New York, NY, USA. ACM.
- [Bimber and Raskar, 2006] Bimber, O. and Raskar, R. (2006). Modern approaches to augmented reality. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Courses*, page 1, New York, NY, USA. ACM.
- [Blanke et al., 2006] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., and Schröder, J. (2006). *Diagnosis and Fault-Tolerant Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Bleser et al., 2015] Bleser, G., Damen, D., Behera, A., Hendeby, G., Mura, K., Miezal, M., Gee, A., Petersen, N., Mações, G., Domingues, H., et al. (2015). Cognitive learning, monitoring and assistance of industrial workflows using egocentric sensor networks. *PloS one*, 10(6) :e0127769.

- [Bolt, 1980] Bolt, R. A. (1980). "Put-that-there" : Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, pages 262–270, New York, NY, USA. ACM.
- [Bottecchia, 2010] Bottecchia, S. (2010). *Système TAC : Télé-Assistance Collaborative. Réalité augmentée et NTIC au service des opérateurs et des experts dans le cadre d'une tâche de maintenance industrielle supervisée*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier.
- [Bowman et al., 1999a] Bowman, D. A., Advisor, H., Larry, D., Hodges, F., Mine, M., Stoakley, R., Pierce, J., Forsberg, A., Hinckley, K., Conway, M., Peterson, B., Poupyrev, I., Kruijff, E., and Laviola, J. (1999a). Interaction techniques for common tasks in immersive virtual environments - design, evaluation, and application.
- [Bowman and Hodges, 1997] Bowman, D. A. and Hodges, L. F. (1997). An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics, I3D '97*, pages 35–ff., New York, NY, USA. ACM.
- [Bowman et al., 1999b] Bowman, D. A., Johnson, D. B., and Hodges, L. F. (1999b). Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '99*, pages 26–33, New York, NY, USA. ACM.
- [Bowman et al., 2004] Bowman, D. A., Kruijff, E., LaViola, J. J., and Poupyrev, I. (2004). *3D User Interfaces : Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- [Broll et al., 2000] Broll, W., Meier, E., and Schardt, T. (2000). The virtual round table - a collaborative augmented multi-user environment. In *Proceedings of the Third International Conference on Collaborative Virtual Environments, CVE '00*, pages 39–45, New York, NY, USA. ACM.
- [Brunelli and Poggio, 1993] Brunelli, R. and Poggio, T. (1993). Face recognition : features versus templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(10) :1042–1052.
- [Bruns et al., 2007] Bruns, E., Brombach, B., Zeidler, T., and Bimber, O. (2007). Enabling mobile phones to support large-scale museum guidance. *IEEE Multimedia*, 14(2) :16–25.
- [Calonder et al., 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF : Binary Robust Independent Elementary Features. In *Proceedings of the 11th European Conference on Computer Vision : Part IV, ECCV'10*, pages 778–792, Berlin, Heidelberg. Springer-Verlag.

- [Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6) :679–698.
- [Carmigniani and Furht, 2011] Carmigniani, J. and Furht, B. (2011). Augmented reality : an overview. In *Handbook of Augmented Reality*, page 3–46. Springer.
- [Chen et al., 2006] Chen, J., Benesty, J., Huang, Y., and Doclo, S. (2006). New insights into the noise reduction wiener filter. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4) :1218–1234.
- [Chen and Patton, 1999] Chen, J. and Patton, R. J. (1999). *Robust Model-based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, Norwell, MA, USA.
- [Chen et al., 2011] Chen, Y., Abushakra, A., and Lee, J. (2011). Vision-Based Horizon Detection and Target Tracking for UAVs. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Wang, S., Kyungnam, K., Benes, B., Moreland, K., Borst, C., DiVerdi, S., Yi-Jen, C., and Ming, J., editors, *Advances in Visual Computing*, volume 6939 of *Lecture Notes in Computer Science*, pages 310–319. Springer Berlin Heidelberg.
- [Clothier and Bailey, 2004] Clothier, M. and Bailey, M. (2004). Augmented reality visualization tool for kings stormwater bridge. In *IASTED International Conference on Visualization, Imaging and Image Processing (VIIP)*.
- [Coomans and Timmermans, 1997] Coomans, M. K. D. and Timmermans, H. J. P. (1997). Towards a taxonomy of virtual reality user interfaces. In *Information Visualization, 1997. Proceedings., 1997 IEEE Conference on*, pages 279–284.
- [Csongei et al., 2012] Csongei, M., Hoang, L., Eck, U., and Sandor, C. (2012). ClonAR : rapid redesign of real-world objects. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, page 277–278.
- [Damen et al., 2014] Damen, D., Haines, O., Leelasawassuk, T., Calway, A., and Mayol-Cuevas, W. (2014). Multi-user egocentric online system for unsupervised assistance on object usage. In *European Conference on Computer Vision*, pages 481–492. Springer.
- [Dan et al., 2013] Dan, F., Dandan, M., Yupeng, Z., and Dongdong, W. (2013). Time machine : A mobile augmented reality system for tourism based on coded-aperture camera. In *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, UIC/ATC 2013, Vietri sul Mare, Sorrento Peninsula, Italy, December 18-21, 2013*. IEEE.
- [Darkhovski, 1994] Darkhovski, B. S. (1994). *Nonparametric methods in change-point problems : a general approach and some concrete algorithms*, volume Volume 23 of *Lecture Notes–Monograph Series*, pages 99–107. Institute of Mathematical Statistics, Hayward, CA.
- [Davy and Godsill, 2002] Davy, M. and Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation.

- [Dekker et al., 2013] Dekker, G., Zhang, Q., Moreland, J., and Zhou, C. (2013). MARWind : Mobile augmented reality wind farm visualization. In *Proceedings of WorldComp 2013 International Conference on Modeling, Simulation and Visualization Methods. Las Vegas, NV*.
- [Delail et al., 2012] Delail, B. A., Weruaga, L., and Zemerly, M. J. (2012). CAViAR : Context Aware Visual Indoor Augmented Reality for a University Campus. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03, WI-IAT '12*, pages 286–290, Washington, DC, USA. IEEE Computer Society.
- [Dellaert et al., 2000] Dellaert, F., Seitz, S. M., Thorpe, C. E., and Thrun, S. (2000). Structure from motion without correspondence. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 557–564 vol.2.
- [Dementhon and Davis, 1995] Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *International journal of computer vision*, 15(1-2) :123–141.
- [Demirkus et al., 2010] Demirkus, M., Toews, M., Clark, J., and Arbel, T. (2010). Gender classification from unconstrained video sequences. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 55–62.
- [Dhome et al., 1989] Dhome, M., Richetin, M., Lapreste, J. T., and Rives, G. (1989). Determination of the attitude of 3d objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12) :1265–1278.
- [Didier, 2005] Didier, J.-Y. (2005). *Contributions à la dextérité d'un système de réalité augmentée mobile appliqué à la maintenance industrielle*. PhD thesis, Université d'Evry Val d'Essonne, Evry.
- [Didier, 2014] Didier, J.-Y. (2014). *Réalité augmentée - tour d'horizon et techniques*.
- [Didier et al., 2008] Didier, j.-Y., Ababsa, F.-E., and Mallem, M. (2008). Hybrid camera pose estimation combining square fiducials localization technique and orthogonal iteration algorithm. *International Journal of Image and Graphics*, 08(01) :169–188.
- [Didier et al., 2012] Didier, J.-Y., Chouiten, M., Mallem, M., and Otmane, S. (2012). ARCS : A framework with extended software integration capabilities to build augmented reality applications. In *2012 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, pages 60–67. IEEE.
- [Didier et al., 2009] Didier, J.-Y., Otmane, S., and Mallem, M. (2009). ARCS, une architecture logicielle reconfigurable pour la conception des applications de réalité augmentée. *TSI. Technique et science informatiques*, 28(6-7) :891–919.
- [Didier et al., 2005] Didier, J.-Y., Roussel, D., Mallem, M., Otmane, S., Naudet, S., Pham, Q.-C., Bourgeois, S., Mégard, C., Leroux, C., and Hocquard, A. (2005). AMRA : augmented

- reality assistance for train maintenance tasks. In *Workshop Industrial Augmented Reality, 4th ACM/IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2005)*, pages Elect-Proc, Vienna (Austria).
- [DiVerdi and Hollerer, 2007] DiVerdi, S. and Hollerer, T. (2007). GroundCam : a tracking modality for mobile mixed reality. In *IEEE Virtual Reality Conference, 2007. VR '07*, pages 75–82.
- [Djelil, 2013] Djelil, F. (2013). Etude et évaluation de l'apport des nuis pour des applications de réalité virtuelle et augmentée.
- [Duan et al., 2004] Duan, L.-Y., Xu, M., Tian, Q., and Xu, C.-S. (2004). Nonparametric motion model with applications to camera motion pattern classification. In *Proceedings of the 12th Annual ACM International Conference on Multimedia, MULTIMEDIA '04*, pages 328–331, New York, NY, USA. ACM.
- [Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1) :11–15.
- [Dunn and Parberry, 2002] Dunn, F. and Parberry, I. (2002). *3D Math Primer for Graphics and Game Development*. Wordware game math library. Wordware Pub.
- [Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous Localisation and Mapping (SLAM) : Part I The Essential Algorithms. *IEEE Robotics and Automation Magazine*, 2 :2006.
- [Engelke et al., 2015] Engelke, T., Keil, J., Rojtberg, P., Wientapper, F., Schmitt, M., and Bockholt, U. (2015). Content first : a concept for industrial augmented reality maintenance applications using mobile devices. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 105–111. ACM.
- [Ettinger et al., 2002a] Ettinger, S., Nechyba, M., Ifju, P., and Waszak, M. (2002a). Vision-guided flight stability and control for micro air vehicles. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2134–2140 vol.3.
- [Ettinger et al., 2002b] Ettinger, S. M., Nechyba, M. C., Ifju, P. G., and Waszak, M. (2002b). Towards flight autonomy : Vision-based horizon detection for micro air vehicles.
- [Fancourt and Principe, 2000] Fancourt, C. L. and Principe, J. C. (2000). On the use of neural networks in the generalized likelihood ratio test for detecting abrupt changes in signals.
- [Feiner et al., 1997] Feiner, S., MacIntyre, B., Hollerer, T., and Webster, A. (1997). A touring machine : prototyping 3D mobile augmented reality systems for exploring the urban environment. In *First International Symposium on Wearable Computers, 1997. Digest of Papers*, pages 74–81.
- [Feiner et al., 1993] Feiner, S., Macintyre, B., and Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM*, 36(7) :53–62.

- [Fellous, 1997] Fellous, J.-M. (1997). Gender discrimination and prediction on the basis of facial metric information. *Vision Research*, 37(14) :1961 – 1973.
- [Ferraz et al., 2014] Ferraz, L., Binefa, X., and Moreno-Noguer, F. (2014). Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection. In *Conference in Computer Vision and Pattern Recognition (CVPR)*.
- [Fiala, 2005] Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 590–596 vol. 2.
- [Fillatre, 2011] Fillatre, L. (2011). Contributions en détection et classification statistique paramétrique.
- [Fiorentino et al., 2002] Fiorentino, M., de Amicis, R., Monno, G., and Stork, A. (2002). Spacedesign : A mixed reality workspace for aesthetic industrial design. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, page 86–318.
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395.
- [Fite-Georgel, 2011] Fite-Georgel, P. (2011). Is there a reality in industrial augmented reality? In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 201–210.
- [Förstner, 2001] Förstner, W. (2001). Generic estimation procedures for orientation with minimum and redundant information. In Gruen, A. and Huang, T., editors, *Calibration and Orientation of Cameras in Computer Vision*, volume 34 of *Springer Series in Information Sciences*, pages 63–94. Springer Berlin Heidelberg.
- [Frank, 1996] Frank, P. (1996). Analytical and qualitative model-based fault diagnosis – a survey and some new results. *European Journal of Control*, 2(1) :6 – 28.
- [Friedrich et al., 2002] Friedrich, W., Jahn, D., and Schmidt, L. (2002). ARVIKA-Augmented reality for development, production and service. In *ISMAR*, volume 2002, page 3–4. Citeseer.
- [Fuchs et al., 2003] Fuchs, P., Moreau, G., and Berthoz, A., editors (2003). *Le traité de la réalité virtuelle. Volume 1. , Fondements et interfaces comportementales*. Sciences mathématiques et informatique. Les Presses de l'École des Mines, Paris.
- [Gao et al., 2003] Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8) :930–943.
- [Garrido-Jurado et al., 2014] Garrido-Jurado, S., noz Salinas, R. M., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6) :2280 – 2292.

- [Gauglitz et al., 2011] Gauglitz, S., Höllerer, T., and Turk, M. (2011). Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3) :335–360.
- [Georgilas et al., 2014] Georgilas, I., Gale, E., Adamatzky, A., and Melhuish, C. (2014). UAV Horizon Tracking Using Memristors and Cellular Automata Visual Processing. In Natraj, A., Cameron, S., Melhuish, C., and Witkowski, M., editors, *Towards Autonomous Robotic Systems*, Lecture Notes in Computer Science, pages 64–75. Springer Berlin Heidelberg.
- [Gershikov et al., 2013] Gershikov, E., Libe, T., and Kosolapov, S. (2013). Horizon line detection in marine images : Which method to choose? *International Journal On Advances in Intelligent Systems*, 6(1 and 2) :79–88.
- [Ghouaiel, 2014] Ghouaiel, N. (2014). *Ingénierie de la conception de systèmes de réalité augmentée mobile. Applications de la réalité augmentée mobile au tourisme culturel*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France.
- [Gleue and Dähne, 2001] Gleue, T. and Dähne, P. (2001). Design and implementation of a mobile device for outdoor augmented reality in the archeoguide project. In *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage*, VAST '01, pages 161–168, New York, NY, USA. ACM.
- [Gorecky et al., 2011] Gorecky, D., Worgan, S. F., and Meixner, G. (2011). COGNITO : A Cognitive Assistance and Training System for Manual Tasks in Industry. In *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics*, ECCE '11, pages 53–56, New York, NY, USA. ACM.
- [Goto et al., 2010] Goto, M., Uematsu, Y., Saito, H., Senda, S., and Iketani, A. (2010). Task support system by displaying instructional video onto ar workspace. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 83–90.
- [Grabner et al., 2006] Grabner, M., Grabner, H., and Bischof, H. (2006). Fast approximated SIFT. In *7th Asian Conference of Computer Vision*, pages 918–927.
- [Grasset and Gascuel, 2001] Grasset, R. and Gascuel, J.-D. (2001). Environnement de Réalité Augmentée Collaboratif : Manipulation d'Objets Réels et Virtuels. In *AFIG 01, actes des 14èmes Journées de l'Association Française d'Informatique Graphique*, pages 101–112, Limoges, France.
- [Gupta et al., 2012] Gupta, A., Fox, D., Curless, B., and Cohen, M. (2012). DuploTrack : A Real-time System for Authoring and Guiding Duplo Block Assembly. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 389–402, New York, NY, USA. ACM.

- [Gutta et al., 2000] Gutta, S., Huang, J., Jonathon, P., and Wechsler, H. (2000). Mixture of experts for classification of gender, ethnic origin, and pose of human faces. *Neural Networks, IEEE Transactions on*, 11(4) :948–960.
- [Hagbi et al., 2011] Hagbi, N., Bergig, O., El-Sana, J., and Billinghamurst, M. (2011). Shape recognition and pose estimation for mobile augmented reality. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10) :1369–1379.
- [Hakkarainen et al., 2008] Hakkarainen, M., Woodward, C., and Billinghamurst, M. (2008). Augmented assembly using a mobile phone. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '08*, pages 167–168, Washington, DC, USA. IEEE Computer Society.
- [Haller et al., 2007] Haller, M., Billinghamurst, M., and Thomas, B. (2007). *Emerging Technologies of Augmented Reality : Interfaces and Design*. Idea Group Publishing.
- [Hargittai, 2005] Hargittai, S. (2005). Savitzky-golay least-squares polynomial filters in ecg signal processing. In *Computers in Cardiology, 2005*, pages 763–766.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer.
- [Hartl et al., 2016] Hartl, A. D., Arth, C., Grubert, J., and Schmalstieg, D. (2016). Efficient verification of holograms using mobile augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 22(7) :1843–1851.
- [Heiser et al., 2004] Heiser, J., Phan, D., Agrawala, M., Tversky, B., and Hanrahan, P. (2004). Identification and validation of cognitive design principles for automated generation of assembly instructions. In *Proceedings of the working conference on Advanced visual interfaces*, pages 311–319. ACM.
- [Henderson and Feiner, 2011] Henderson, S. and Feiner, S. (2011). Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE transactions on visualization and computer graphics*, 17(10) :1355–1368.
- [Henderson and Feiner, 2009] Henderson, S. J. and Feiner, S. (2009). Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, page 135–144.
- [Higuchi et al., 2015] Higuchi, T., Iwahashi, H., Yamaguchi, H., and Higashino, T. (2015). TweetGlue : Leveraging a crowd tracking infrastructure for mobile social augmented reality. In *IWCMC*, pages 1030–1035. IEEE.
- [Hinterstoisser et al., 2010] Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., and Navab, N. (2010). Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2257–2264.

- [hired CloudFlare, 2014] hired CloudFlare, S. (2014). Experts detect biggest distributed denial-of-service attack ever.
- [Höllerer and Feiner, 2004] Höllerer, T. and Feiner, S. (2004). Mobile augmented reality. *Telegeoinformatics : Location-Based Computing and Services*. Taylor and Francis Books Ltd., London, UK.
- [Horn and Schunck, 1981] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1–3) :185 – 203.
- [Hoult, 2012] Hoult, C. (2012). Giality – geospatial data for augmented reality.
- [Huang et al., 2013] Huang, Z., Hui, P., Peylo, C., and Chatzopoulos, D. (2013). Mobile augmented reality survey : a bottom-up approach. *CoRR*, abs/1309.4413.
- [Hussain et al., 2014] Hussain, A., Knight, L., Al-Jumeily, D., Fergus, P., and Hamdan, H. (2014). Block matching algorithms for motion estimation – a comparison study. In Thampi, S. M., Gelbukh, A., and Mukhopadhyay, J., editors, *Advances in Signal Processing and Intelligent Recognition Systems*, volume 264 of *Advances in Intelligent Systems and Computing*, pages 359–369. Springer International Publishing.
- [Höllerer et al., 1999] Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., and Hallaway, D. (1999). Exploring MARS : Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. *Computers and Graphics*, 23 :779–785.
- [Irawati et al., 2008] Irawati, S., Ahn, S., Kim, J., and Ko, H. (2008). VARU Framework : Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications. In *Virtual Reality Conference, 2008. VR '08. IEEE*, pages 201–208.
- [Isermann, 1997] Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods — an introduction. *Control Engineering Practice*, 5(5) :639 – 652.
- [J. Newman, 2006] J. Newman, G. S. (2006). Wide-area tracking tools for augmented reality. *Advances in Pervasive Computing*, 207 :3–6.
- [Jacoby et al., 1994] Jacoby, R. H., Ferneau, M., and Humphries, J. (1994). Gestural interaction in a virtual environment.
- [Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D) :35–45.
- [Kato and Billinghurst, 1999] Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94.
- [Kato et al., 2000] Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. (2000). Virtual object manipulation on a table-top ar environment. In *Augmented Reality*,

2000. (ISAR 2000). *Proceedings. IEEE and ACM International Symposium on*, pages 111–119.
- [Ke and Sukthankar, 2004] Ke, Y. and Sukthankar, R. (2004). PCA-SIFT : A More Distinctive Representation for Local Image Descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'04*, pages 506–513, Washington, DC, USA. IEEE Computer Society.
- [Khan et al., 2011] Khan, S. A., Nazir, M., Riaz, N., and Naveed, N. (2011). Computationally intelligent gender classification techniques : An analytical study. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 4(4).
- [Kiyokawa, 2012] Kiyokawa, K. (2012). Trends and vision of head mounted display in augmented reality. In *Ubiquitous Virtual Reality (ISUVR), 2012 International Symposium on*, pages 14–17.
- [Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '07*, pages 1–10, Washington, DC, USA. IEEE Computer Society.
- [Klinker et al., 2002] Klinker, G., Dutoit, A. H., Bauer, M., Bayer, J., Novak, V., and Matzke, D. (2002). Fata morgana - a presentation system for product design. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pages 76–85.
- [Knopfle et al., 2005] Knopfle, C., Weidenhausen, J., Chauvigne, L., and Stock, I. (2005). Template based authoring for ar based service scenarios. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pages 237–240.
- [Kumar and Hanson, 1994] Kumar, R. and Hanson, A. (1994). Robust methods for estimating pose and a sensitivity analysis. *CVGIP : Image Understanding*, 60(3) :313 – 342.
- [Kurkovsky et al., 2012] Kurkovsky, S., Koshy, R., Novak, V., and Szul, P. (2012). Current issues in handheld augmented reality. In *Communications and Information Technology (IC-CIT), 2012 International Conference on*, page 68–72.
- [Lai, 1998] Lai, T. L. (1998). Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Transactions on Information Theory*, 44(7) :2917–2929.
- [Langlotz, 2013] Langlotz, T. (2013). *AR 2.0 : Social Media in Mobile Augmented Reality*. PhD thesis.
- [Lee and Billingham, 2012] Lee, G. and Billingham, M. (2012). CityViewAR Outdoor AR Visualization. In *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction, CHINZ '12*, pages 97–97, New York, NY, USA. ACM.

- [Leonard and Durrant-Whyte, 1991] Leonard, J. J. and Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3.
- [Lepetit and Fua, 2005] Lepetit, V. and Fua, P. (2005). Monocular model-based 3d tracking of rigid objects : A survey. In *Foundations and Trends in Computer Graphics and Vision*, pages 1–89.
- [Lepetit et al., 2009] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP : An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2) :155–166.
- [Leu et al., 2013] Leu, M. C., ElMaraghy, H. A., Nee, A. Y., Ong, S. K., Lanzetta, M., Putz, M., Zhu, W., and Bernard, A. (2013). CAD model based virtual assembly simulation, planning and training. *CIRP Annals - Manufacturing Technology*, 62(2) :799–822.
- [Leutenegger et al., 2011] Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK : Binary Robust Invariant Scalable Keypoints. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 2548–2555, Washington, DC, USA. IEEE Computer Society.
- [Lian and Lu, 2006] Lian, H.-C. and Lu, B.-L. (2006). Multi-view gender classification using local binary patterns and support vector machines. In *Proceedings of the Third International Conference on Advances in Neural Networks - Volume Part II, ISNN'06*, pages 202–209, Berlin, Heidelberg. Springer-Verlag.
- [Liestol and Morrison, 2013] Liestol, G. and Morrison, A. (2013). Views, alignment and incongruity in indirect augmented reality. In *Mixed and Augmented Reality-Arts, Media, and Humanities (ISMAR-AMH), 2013 IEEE International Symposium on*, page 23–28.
- [Lindeman et al., 2012] Lindeman, R. W., Lee, G., Beattie, L., Gamper, H., Pathinarupothi, R., and Akhilesh, A. (2012). Geoboids : A mobile ar application for exergaming. In *Mixed and Augmented Reality (ISMAR-AMH), 2012 IEEE International Symposium on*, pages 93–94.
- [Liu et al., 2008] Liu, A. L., Hile, H., Kautz, H., Borriello, G., Brown, P. A., Harniss, M., and Johnson, K. (2008). Indoor wayfinding : developing a functional interface for individuals with cognitive impairments. *Disability and rehabilitation. Assistive technology*, 3(1) :69–81.
- [Lorden, 1971] Lorden, G. (1971). Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42(6) :1897–1908.
- [Lowe, 1999] Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2.

- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110.
- [Lu et al., 2000] Lu, C.-P., Hager, G. D., and Mjolsness, E. (2000). Fast and globally convergent pose estimation from video images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6) :610–622.
- [Lung-Yut-Fong, 2011] Lung-Yut-Fong, A. (2011). *Robust change-point detection for multi-dimensional signals. Application to computer network anomaly detection*. Theses, Télécom ParisTech.
- [MacWilliams et al., 2004] MacWilliams, A., Reicher, T., Klinker, G., and Brüggel, B. (2004). Design patterns for augmented reality systems. In 0001, E. D., Gray, P. D., Trevisan, D., and Vanderdonck, J., editors, *MIXER*, volume 91 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Maquin and Ragot, 2000] Maquin, D. and Ragot, J. (2000). *Diagnostic des systèmes linéaires*. Collection Pédagogique d’Automatique. Hermès science publications.
- [Martinez et al., 2013] Martinez, H., Laukkanen, S., and Mattil, J. (2013). A new hybrid approach for augmented reality maintenance in scientific facilities. *International Journal of Advanced Robotic Systems*, page 1.
- [Mei, 2006] Mei, Y. (2006). Sequential change-point detection when unknown parameters are present in the pre-change distribution. *The Annals of Statistics*, 34(1) :92–122.
- [Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, E77-D(12) :1321–1329.
- [Mine et al., 1997] Mine, M. R., Brooks, Jr., F. P., and Sequin, C. H. (1997). Moving objects in space : Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’97*, pages 19–26, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Moghaddam and Yang, 2000] Moghaddam, B. and Yang, M.-H. (2000). Gender classification with support vector machines. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 306–311.
- [Moustakides, 2002] Moustakides, G. V. (2002). Optimality of the CUSUM Procedure. Research report, INRIA. Projet IRIS.
- [Mozaffari et al., 2010] Mozaffari, S., Behravan, H., and Akbari, R. (2010). Gender classification using single frontal image per person : Combination of appearance and geometric based features. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1192–1195.
- [Muja and Lowe, 2009] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP’09*, pages 331–340. INSTICC Press.

- [Naimark and Foxlin, 2002] Naimark, L. and Foxlin, E. (2002). Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pages 27–36.
- [Nater et al., 2011] Nater, F., Grabner, H., and Gool, L. J. V. (2011). Unsupervised workflow discovery in industrial environments. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, pages 1912–1919.
- [Ng et al., 2012] Ng, C. B., Tay, Y. H., and Goi, B. M. (2012). Vision-based human gender recognition : a survey. *arXiv preprint arXiv :1204.1611*.
- [Nilsson et al., 2010] Nilsson, J., Odblom, A., Fredriksson, J., Zafar, A., and Ahmed, F. (2010). Performance evaluation method for mobile computer vision systems using augmented reality. In *2010 IEEE Virtual Reality Conference (VR)*, pages 19–22.
- [Oberkampff et al., 1996] Oberkampff, D., DeMenthon, D. F., and Davis, L. S. (1996). Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3) :495 – 511.
- [Olwal, 2006] Olwal, A. (2006). LightSense : enabling spatially aware handheld interaction devices. In *IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006*, pages 119–122.
- [Ortiz, 2012] Ortiz, R. (2012). FREAK : Fast Retina Keypoint. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12*, pages 510–517, Washington, DC, USA. IEEE Computer Society.
- [Ota et al., 2010] Ota, N., Yabuki, N., and Fukuda, T. (2010). Development of an accurate positioning method for augmented reality using multiple markers. In *Proceedings of the International Conference on Computing in Civil and Building Engineering-2010*.
- [Otmane, 2010] Otmane, S. (2010). *Software models and techniques for assistance to interaction and collaboration in mixed reality*. Habilitation à diriger des recherches, Université d'Evry-Val d'Essonne.
- [Ouramdane et al., 2006] Ouramdane, N., Otmane, S., Davesne, F., and Mallem, M. (2006). Follow-me : A new 3d interaction technique based on virtual guides and granularity of interaction. In *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications, VRCIA '06*, pages 137–144, New York, NY, USA. ACM.
- [Page, 1954] Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1-2) :100–115.
- [Pang et al., 2005] Pang, Y., Nee, A. Y., Youcef-Toumi, K., Ong, S., and Yuan, M. (2005). Assembly design and evaluation in an augmented reality environment.

- [Papagiannakis et al., 2008] Papagiannakis, G., Singh, G., and Magnenat-Thalmann, N. (2008). A survey of mobile and wireless technologies for augmented reality systems. *Journal of Visualization and Computer Animation*, 19(1) :3–22.
- [Park, 2008] Park, J. (2008). Augmented reality based re-formable mock-up for design evaluation. In *Ubiquitous Virtual Reality, 2008. ISUVR 2008. International Symposium on*, pages 17–20.
- [Pawlak et al., 2003] Pawlak, M., Rafajlowicz, E., and Steland, A. (2003). On detecting jumps in time series : Nonparametric setting. (2003,28).
- [Pentenrieder et al., 2007] Pentenrieder, K., Bade, C., Doil, F., and Meier, P. (2007). Augmented reality-based factory planning - an application tailored to industrial needs. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '07*, pages 1–9. IEEE Computer Society.
- [Peternier et al., 2006] Peternier, A., Vexo, F., and Thalmann, D. (2006). Wearable mixed reality system in less than 1 pound. In Lin, M. and Hubbold, R. J., editors, *EGVE*, pages 35–44. Eurographics Association.
- [Petersen and Stricker, 2012] Petersen, N. and Stricker, D. (2012). Learning task structure from video examples for workflow tracking and authoring. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 237–246.
- [Pierce et al., 1997] Pierce, J. S., Forsberg, A. S., Conway, M. J., Hong, S., Zeleznik, R. C., and Mine, M. R. (1997). Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics, I3D '97*, pages 39–ff., New York, NY, USA. ACM.
- [Pizurica et al., 2003] Pizurica, A., Philips, W., Lemahieu, I., and Acheroy, M. (2003). A versatile wavelet domain noise filtration technique for medical imaging. *IEEE Transactions on Medical Imaging*, 22(3) :323–331.
- [Platt, 1999] Platt, J. C. (1999). Probabilities for SV Machines. MIT Press.
- [Pollak and Siegmund, 1975] Pollak, M. and Siegmund, D. (1975). Approximations to the expected sample size of certain sequential tests. *Ann. Statist.*, 3(6) :1267–1282.
- [Pu et al., 2011] Pu, C.-C., Pu, C.-H., and Lee, H.-J. (2011). Indoor location tracking using received signal strength indicator. In Foerster, A. and Foerster, A., editors, *Emerging Communications for Wireless Sensor Networks*. InTech.
- [Quan and Lan, 1999] Quan, L. and Lan, Z. (1999). Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8) :774–780.
- [Radkowski and Stritzke, 2012] Radkowski, R. and Stritzke, C. (2012). Interactive hand gesture-based assembly for augmented reality applications. In *ACHI 2012, The Fifth International Conference on Advances in Computer-Human Interactions*, pages 303–308.

- [Raghavan et al., 1999] Raghavan, V., Molineros, J., and Sharma, R. (1999). Interactive evaluation of assembly sequences using augmented reality. *Robotics and Automation, IEEE Transactions on*, 15(3) :435–449.
- [Rashid et al., 2013] Rashid, H. S., Place, C. S., and Braithwaite, G. R. (2013). Investigating the investigations : A retrospective study in the aviation maintenance error causation. *Cogn. Technol. Work*, 15(2) :171–188.
- [Rashid et al., 2006] Rashid, O., Bamford, W., Coulton, P., Edwards, R., and Scheible, J. (2006). PAC-LAN : mixed-reality gaming with RFID-enabled mobile phones. *Comput. Entertain.*, 4(4).
- [Re, 2013] Re, G. (2013). *Low cost augmented reality for industrial problems*. PhD thesis, Italy.
- [Regenbrecht et al., 2005] Regenbrecht, H., Baratoff, G., and Wilke, W. (2005). Augmented reality projects in the automotive and aerospace industries. *Computer Graphics and Applications, IEEE*, 25(6) :48–56.
- [Regenbrecht et al., 2007] Regenbrecht, H., Haller, M., Billingham, M., and Thomas, B. (2007). Industrial augmented reality applications. In *Emerging technologies of augmented reality : Interfaces and design*, pages 283–304. Idea Group Publishing.
- [Regenbrecht et al., 2002] Regenbrecht, H., Wagner, M., and Baratoff, G. (2002). Magicmeeting : A collaborative tangible augmented reality system. *Virtual Reality*, 6(3) :151–166.
- [Rekimoto and Ayatsuka, 2000] Rekimoto, J. and Ayatsuka, Y. (2000). Cybercode : Designing augmented reality environments with visual tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, pages 1–10, New York, NY, USA. ACM.
- [Roberts, 1959] Roberts, S. W. (1959). Control chart tests based on geometric moving averages. *Technometrics*, 1(3) :239–250.
- [Ruble et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb : An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 2564–2571, Washington, DC, USA. IEEE Computer Society.
- [Rukubayihunga et al., 2014] Rukubayihunga, A., Didier, J.-Y., Chouiten, M., and Otmane, S. (2014). An overview of occlusion issues handling in augmented reality. In *Journées de l'Association française de réalité virtuelle, augmentée, mixte et d'interaction 3D*, Reims, France.
- [Rukubayihunga et al., 2015] Rukubayihunga, A., Didier, J. Y., and Otmane, S. (2015). Reprojection error as a new metric to detect assembly/disassembly maintenance tasks. In *Image*

- Processing Theory, Tools and Applications (IPTA)*, 2015 International Conference on, pages 513–518, Orléans, France.
- [Rukubayihunga et al., 2016a] Rukubayihunga, A., Didier, J.-Y., and Otmane, S. (2016a). Real time noise reduction to identify motion parameters in ar maintenance scenario. In *Mixed and Augmented Reality (ISMAR), 2016 15th IEEE International Symposium on*, Merida, Mexique.
- [Rukubayihunga et al., 2016b] Rukubayihunga, A., Didier, J.-Y., and Otmane, S. (2016b). Towards assembly steps recognition in augmented reality. In *Proceedings of the 2016 Virtual Reality International Conference, VRIC '16*, Laval, France. ACM.
- [Salonen et al., 2007] Salonen, T., Sääski, J., Hakkarainen, M., Kannetis, T., Perakakis, M., Siltanen, S., Potamianos, A., Korkalo, O., and Woodward, C. (2007). Demonstration of assembly work using augmented reality. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 120–123. ACM.
- [Santos et al., 2007] Santos, P., Stork, A., Gierlinger, T., Pagani, A., Paloc, C., Barandarian, I., Conti, G., Amicis, R. d., Witzel, M., Machui, O., Jiménez, J. M., Araujo, B., Jorge, J., and Bodammer, G. (2007). IMPROVE : an innovative application for collaborative mobile mixed reality design review. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 1(2) :115–126.
- [Savitzky and Golay, 1964] Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8) :1627–1639.
- [Schall et al., 2013] Schall, G., Zollmann, S., and Reitmayr, G. (2013). Smart Vidente : Advances in mobile augmented reality for interactive visualization of underground infrastructure. *Personal Ubiquitous Computing*, 17(7) :1533–1549.
- [Schmalstieg et al., 2002] Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavári, Z., Encarnação, L. M., Gervautz, M., and Purgathofer, W. (2002). The studierstube augmented reality project. *Presence : Teleoperators and Virtual Environments*, 11 :33–54.
- [Schmeil and Broll, 2006] Schmeil, A. and Broll, W. (2006). Mara : an augmented personal assistant and companion. In *ACM SIGGRAPH 2006 Sketches*, page 141. ACM.
- [Schwald and De Laval, 2003] Schwald, B. and De Laval, B. (2003). An augmented reality system for training and assistance to maintenance in the industrial context.
- [Shewhart, 1930] Shewhart, W. A. (1930). Economic quality control of manufactured product1. *Bell System Technical Journal*, 9(2) :364–389.
- [Siltanen, 2012] Siltanen, S. (2012). *Theory and applications of marker-based augmented reality*. Licentiate thesis, Aalto University School of Science. Licentiate thesis.
- [Smith and Cheeseman, 1986] Smith, R. C. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *Int. J. Rob. Res.*, 5(4) :56–68.

- [Soule et al., 2005] Soule, A., Salamatian, K., and Taft, N. (2005). Combining filtering and statistical methods for anomaly detection. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, pages 31–31, Berkeley, CA, USA. USENIX Association.
- [Steggles and Gschwind, 2005] Steggles, P. and Gschwind, S. (2005). The ubisense smart space platform.
- [Steland, 2002] Steland, A. (2002). Nonparametric monitoring of financial time series by jump-preserving control charts. *Statistical Papers*, 43(3) :401–422.
- [Stoakley et al., 1995] Stoakley, R., Conway, M. J., and Pausch, R. (1995). Virtual reality on a wim : Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 265–272, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Sturman et al., 1989] Sturman, D. J., Zeltzer, D., and Pieper, S. (1989). Hands-on interaction with virtual environments. In *Proceedings of the 2Nd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology*, UIST '89, pages 19–24, New York, NY, USA. ACM.
- [Sääski et al., 2008] Sääski, J., Salonen, T., Hakkarainen, M., Siltanen, S., Woodward, C., and Lempiäinen, J. (2008). Integration of design and assembly using augmented reality. In *Micro-Assembly Technologies and Applications*, page 395–404. Springer.
- [Tang et al., 2003] Tang, A., Owen, C., Biocca, F., and Mou, W. (2003). Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 73–80, New York, NY, USA. ACM.
- [Tartakovsky and Alexander, 2008] Tartakovsky and Alexander (2008). Multidecision Quickest Change-Point Detection : Previous Achievements and Open Problems. *Sequential Analysis*, 27(2) :201–231.
- [Thomas et al., 2000] Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W. (2000). ARQuake : an outdoor/indoor augmented reality first person application. In *Wearable Computers, The Fourth International Symposium on*, pages 139–146.
- [Timm and Barth, 2011] Timm, F. and Barth, E. (2011). Accurate eye centre localisation by means of gradients. In Mestetskiy, L. and Braz, J., editors, *VISAPP*, pages 125–130. SciTePress.
- [Vacchetti et al., 2004] Vacchetti, L., Lepetit, V., Ponder, M., Papagiannakis, G., Fua, P., Thalmann, D., and Thalmann, N. M. (2004). A stable real-time AR framework for training and planning in industrial environments. In Ong, S. K. and DEng, A. Y. C. N., editors, *Virtual and Augmented Reality Applications in Manufacturing*, pages 129–145. Springer London.

- [Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.
- [Čopić Pucihar et al., 2013] Čopić Pucihar, K., Coulton, P., and Alexander, J. (2013). Evaluating dual-view perceptual issues in handheld augmented reality : Device vs. user perspective rendering. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction, ICMI '13*, pages 381–388, New York, NY, USA. ACM.
- [Veas et al., 2012] Veas, E., Grasset, R., Ferencik, I., Grünewald, T., and Schmalstieg, D. (2012). Mobile augmented reality for environmental monitoring. *Personal and Ubiquitous Computing*, 17(7) :1515–1531.
- [Verdier, 2007] Verdier, G. (2007). *Statistical detection of model change in complex dynamical systems - Application to the supervision of wastewater treatment processes*. Theses, Université Montpellier II - Sciences et Techniques du Languedoc.
- [Veres et al., 2010] Veres, G. V., Grabner, H., Middleton, L., and Gool, L. J. V. (2010). Automatic workflow monitoring in industrial environments. In *Computer Vision - ACCV 2010 - 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part I*, pages 200–213.
- [Viola and Jones, 2004] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int. J. Comput. Vision*, 57(2) :137–154.
- [Wald, 1947] Wald, A. (1947). *Sequential Analysis*. John Wiley and Sons, 1st edition.
- [Wang et al., 2002] Wang, H., Zhang, D., and Shin, K. G. (2002). Detecting syn flooding attacks.
- [Wang et al., 2010] Wang, J.-G., Li, J., Yau, W.-Y., and Sung, E. (2010). Boosting dense sift descriptors and shape contexts of face images for gender recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 96–102.
- [Ward et al., 2006] Ward, J., Lukowicz, P., Troster, G., and Starner, T. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10) :1553–1567.
- [Weck et al., 2004] Weck, M., Hamadou, M., Hoymann, H., Jahn, D., and Lescher, M. (2004). Mobile service applications for machine tools. In *Virtual and Augmented Reality Applications in Manufacturing*, pages 237–255. Springer.
- [Weir et al., 2012] Weir, P., Sandor, C., Swoboda, M., Nguyen, T., Eck, U., Reitmayr, G., and Dey, A. (2012). BurnAR : feel the heat. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, page 331–332.
- [Weiss and Ray, 2001] Weiss, I. and Ray, M. (2001). Model-based recognition of 3d objects from single images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2) :116–128.

- [White and Feiner, 2009] White, S. and Feiner, S. (2009). SiteLens : Situated visualization techniques for urban site visits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1117–1120, New York, NY, USA. ACM.
- [Wiener, 1964] Wiener, N. (1964). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press.
- [Wrobel, 2001] Wrobel, B. P. (2001). *Minimum Solutions for Orientation*, pages 7–62. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Xu et al., 1994] Xu, Y., Weaver, J. B., Healy, D. M., and Lu, J. (1994). Wavelet transform domain filters : a spatially selective noise filtration technique. *IEEE Transactions on Image Processing*, 3(6) :747–758.
- [Yang and Ai, 2007] Yang, Z. and Ai, H. (2007). Demographic classification with local binary patterns. In Lee, S.-W. and Li, S., editors, *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 464–473. Springer Berlin Heidelberg.
- [Zachary G. Stoumbos, 2000] Zachary G. Stoumbos, Marion R. Reynolds, T. P. R. W. H. W. (2000). The state of statistical process control as we proceed into the 21st century. *Journal of the American Statistical Association*, 95(451) :992–998.
- [Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3) :338 – 353.
- [Zauner et al., 2003] Zauner, J., Haller, M., Brandl, A., and Hartman, W. (2003). Authoring of a mixed reality assembly instructor for hierarchical structures. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 237–246. IEEE.
- [Zendjebil, 2010] Zendjebil, I. M. (2010). *Localisation 3D basée sur une approche de suppléance multi-capteurs pour la Réalité Augmentée Mobile en Milieu Extérieur*. PhD thesis, Université d'Evry Val d'Essonne, Evry.
- [Zhang, 1999] Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. IEEE.
- [Zhu et al., 2011] Zhu, J., Nee, A.-C., and Ong, S.-K. (2011). Online authoring for augmented reality remote maintenance.
- [Zhu et al., 2013] Zhu, J., Ong, S., and Nee, A. (2013). An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology*, 66(9-12) :1699–1714.

Annexe A

Résultats : Analyse de l'évolution de l'erreur de reprojection lors de tâches d'assemblage

De plus amples résultats confirmant notre hypothèse selon laquelle l'erreur de reprojection permet de caractériser les tâches d'assemblage et de désassemblage (fig. [A.1a](#), [A.2a](#), [A.3a](#), [A.4a](#), [A.7a](#), [A.8a](#), [A.9a](#), [A.10a](#), [A.5a](#), [A.6a](#), [A.11a](#)). Nous avons vérifié notre hypothèse sur des vidéos synthétiques et des vidéos dans des conditions réelles. Celles-ci représentent des mouvements simples tels qu'une rotation simple autour d'un vecteur correspondant à un des axes du repère d'un des objets (fig. [A.1a](#), [A.2a](#), [A.7a](#), [A.8a](#), [A.9a](#)) ou tels qu'une translation suivant également un des axes du repère d'un des objets (fig. [A.3a](#), [A.10a](#), [A.5a](#), [A.6a](#)). Des mouvements un peu plus complexes tels qu'un vissage ont également été utilisés (fig. [A.4a](#), [A.7a](#), [A.11a](#)).

Dans cette série de résultats, l'évolution de l'erreur de reprojection a été normalisée suivant la résolution de l'image. Cette précaution permet de s'affranchir des potentielles variations existant lors d'un effet de zoom ou de résolution.

A.1 Vidéos synthétiques

Quelques exemples obtenus avec des vidéos synthétiques (fig. [A.1a](#), [A.2a](#), [A.3a](#), [A.4a](#)) confirmant notre hypothèse.

A.2 Vidéos dans des conditions réelles

Quelques exemples obtenus avec des vidéos réelles (fig. [A.7a](#), [A.8a](#), [A.9a](#), [A.10a](#), [A.5a](#), [A.6a](#), [A.11a](#)) confirmant notre hypothèse.

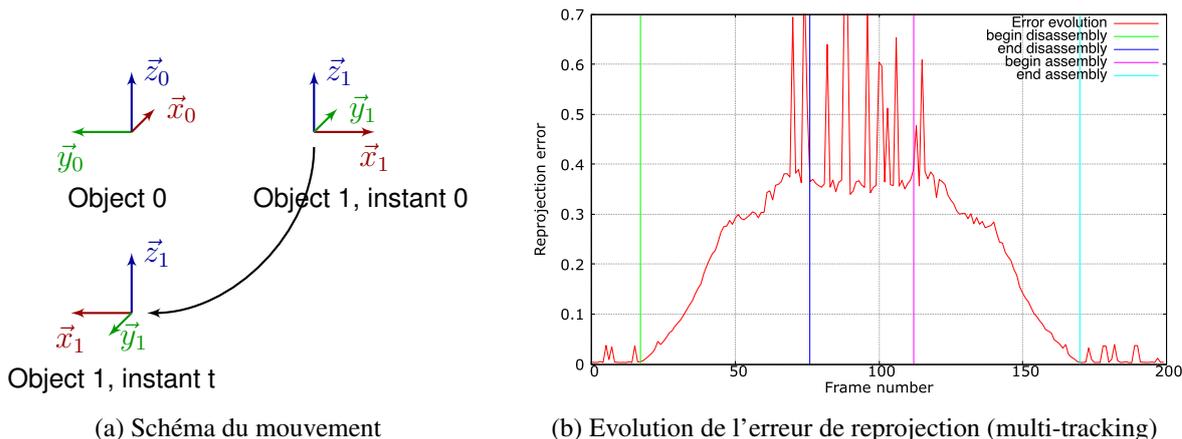


FIGURE A.1 – Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une rotation autour de l'axe Z de l'objet 0.

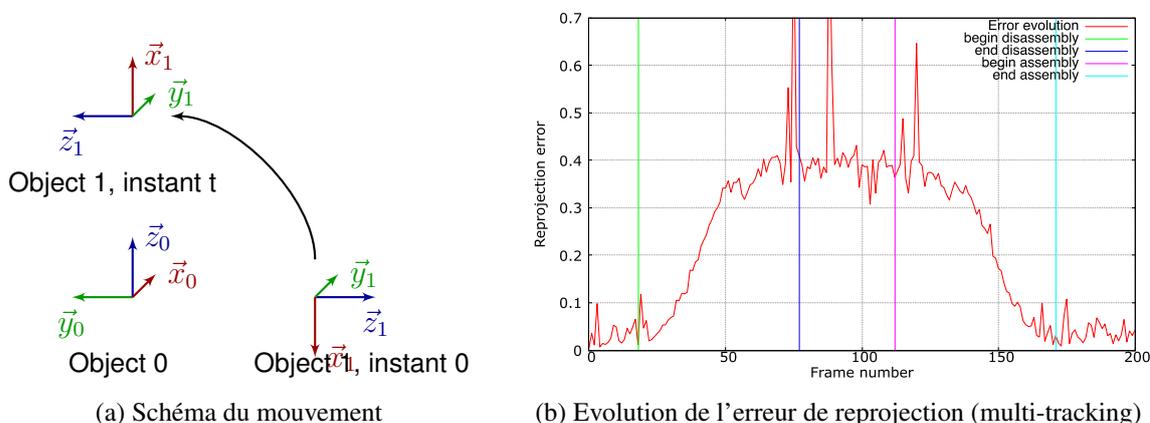


FIGURE A.2 – Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une rotation autour de l'axe X de l'objet 0.

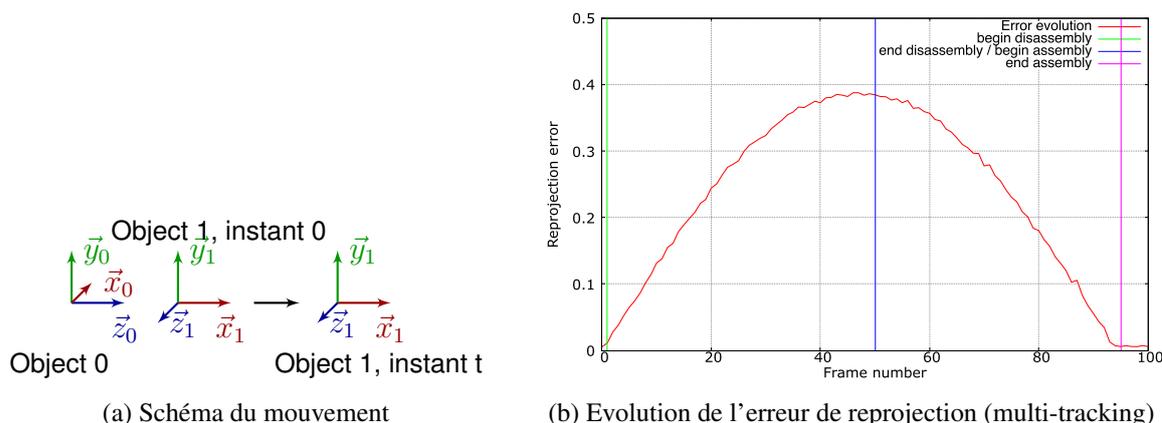


FIGURE A.3 – Vidéo synthétique représentant l'assemblage et le désassemblage de deux objets suivant une translation selon l'axe X de l'objet 0.

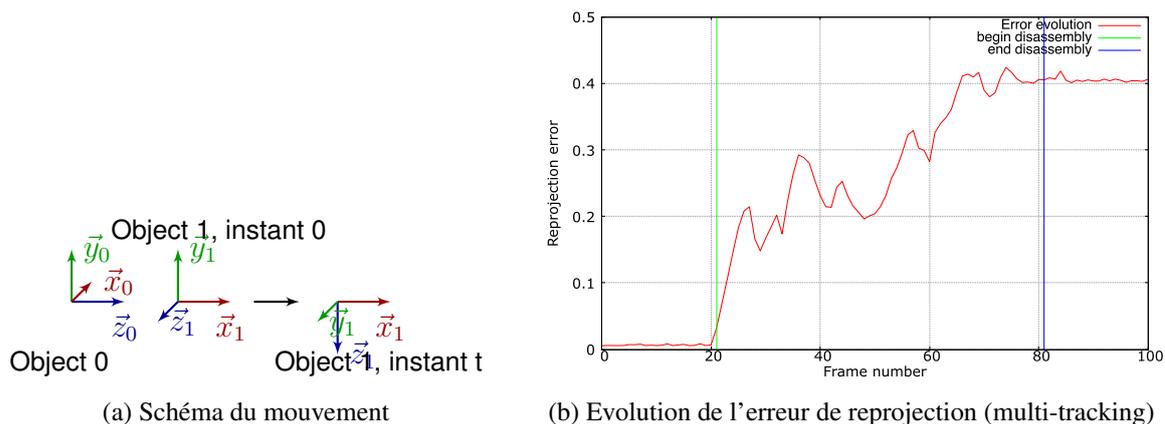


FIGURE A.4 – Vidéo synthétique représentant le désassemblage de deux objets suivant un mouvement de vissage selon l'axe X de l'objet 0.

Certains résultats font une distinction entre des mouvements sur le plan et dans l'espace. En effet, pour des scénarios se déroulant sur le plan, les deux objets manipulés sont posés sur une surface plane indiquant une certaine contrainte ou une perte de liberté suivant certains axes. Les mouvements dans l'espace supposent un positionnement des objets sans contrainte spécifique. Chaque couple de scénarios suit le même "comportement" et donc les mêmes caractéristiques visuelles du mouvement.

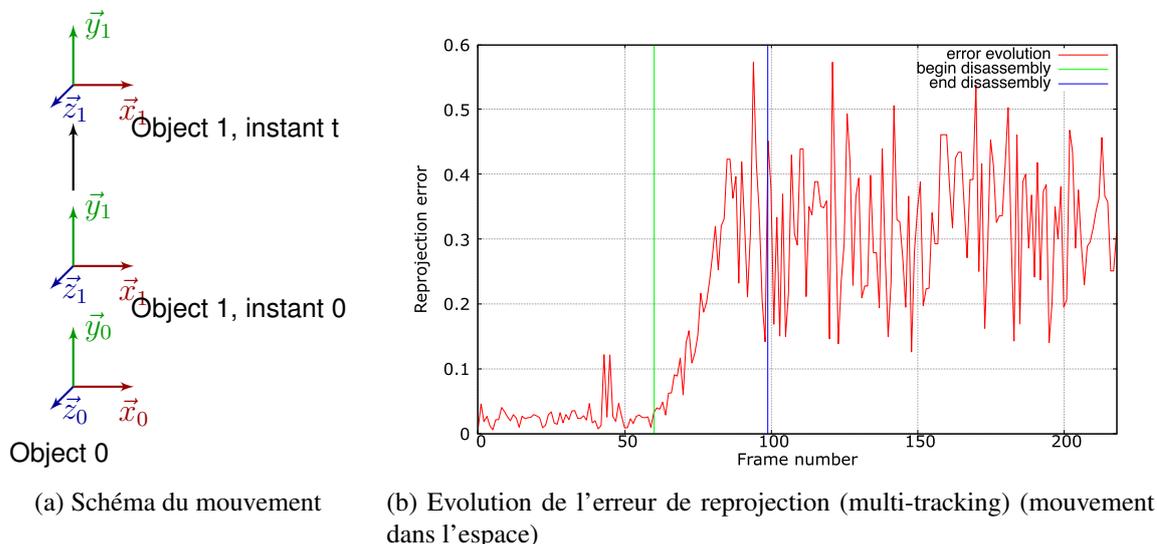


FIGURE A.5 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une translation selon l'axe Y de l'objet 0.

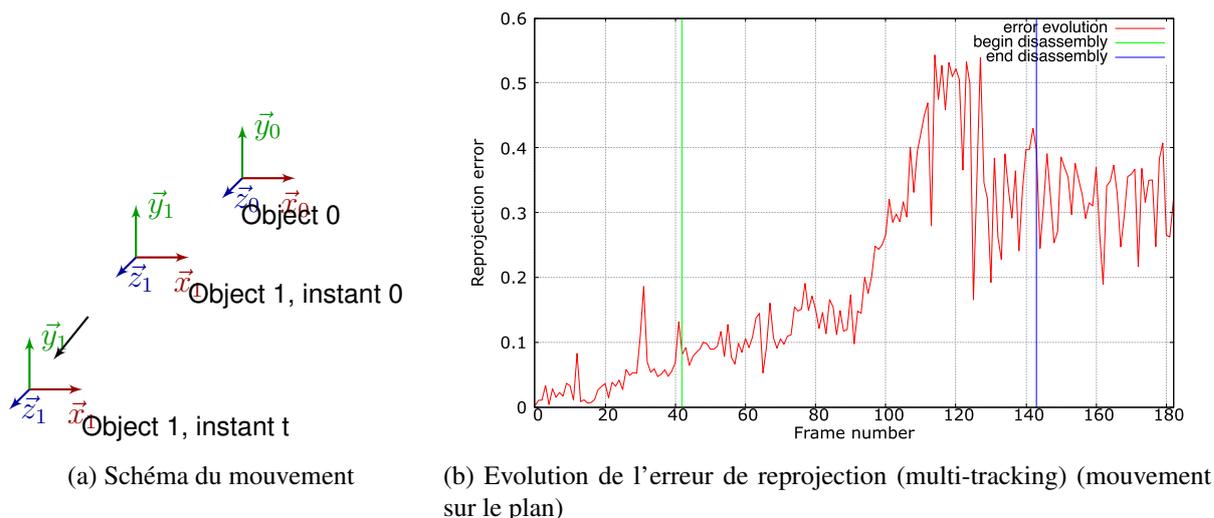


FIGURE A.6 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une translation selon l'axe Z de l'objet 0.

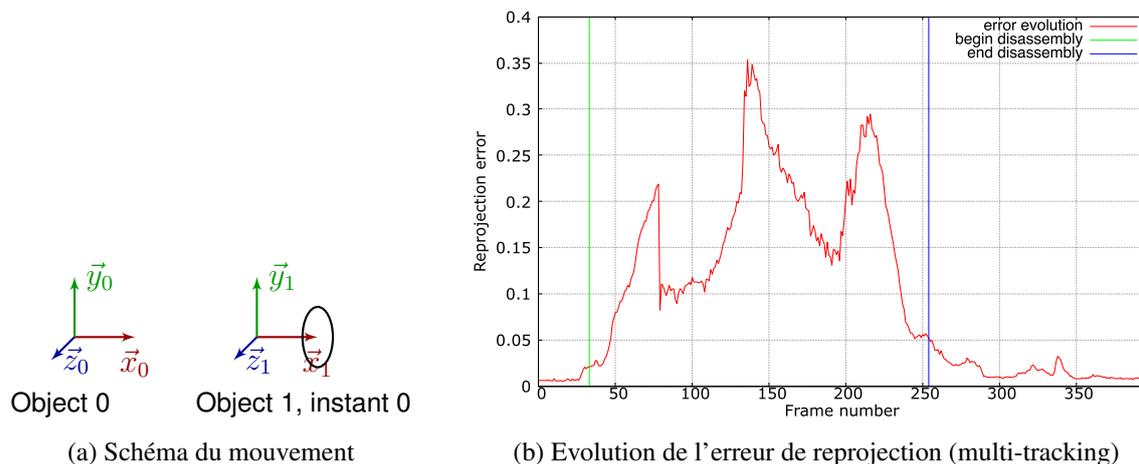


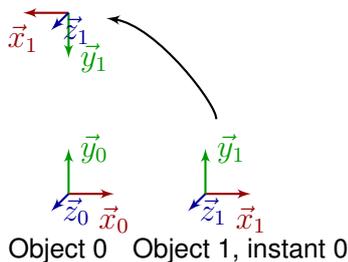
FIGURE A.7 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant un pivot selon l'axe X de l'objet 0.

A.3 Scénarios non concluants

Certains mouvements ne permettent pas de confirmer notre hypothèse (fig. A.12a, A.13a). Cette infirmation est due à une mauvaise identification de la configuration des objets 3D à l'instant initial. En effet, l'objet qui n'est identifié que par un certain nombre de points appartenant à une seule et unique face donc à un seul plan est susceptible d'induire une estimation erronée de la pose de la caméra.

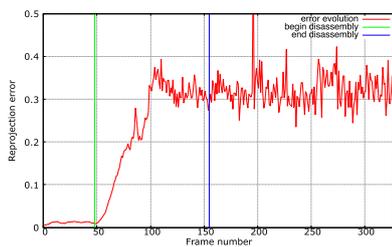
Cependant des expérimentations effectuées en choisissant comme objet de référence l'objet dont tous les points détectés appartiennent au moins à deux faces non planaires de l'objet permettent de confirmer notre hypothèse de départ selon laquelle l'erreur de reprojection permettait de caractériser les tâches d'assemblage et de désassemblage.

Object 1, instant t

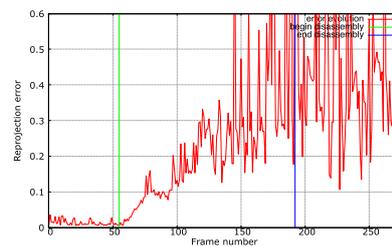


Object 0 Object 1, instant 0

(a) Schéma du mouvement

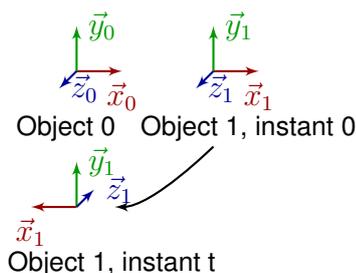


(b) Evolution de l'erreur de reprojection (multi-tracking)(mouvement dans l'espace)



(c) Evolution de l'erreur de reprojection (multi-tracking)(mouvement sur le plan)

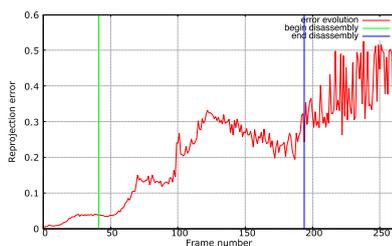
FIGURE A.8 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une rotation autour de l'axe Z de l'objet 0.



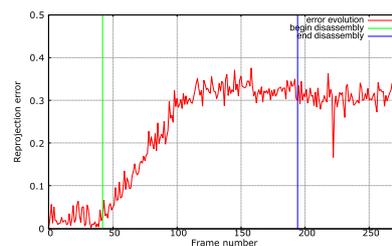
Object 0 Object 1, instant 0

Object 1, instant t

(a) Schéma du mouvement

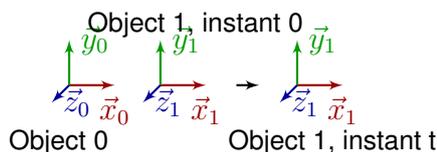


(b) Evolution de l'erreur de reprojection (multi-tracking) (mouvement dans l'espace)

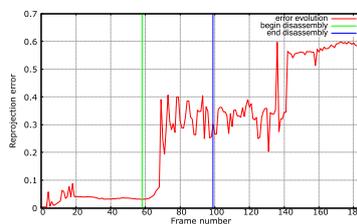


(c) Evolution de l'erreur de reprojection (multi-tracking) (mouvement sur le plan)

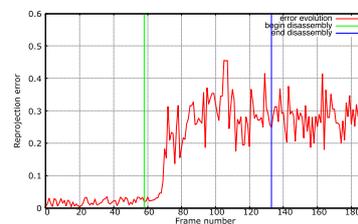
FIGURE A.9 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une rotation autour de l'axe Y de l'objet 0.



(a) Schéma du mouvement



(b) Evolution de l'erreur de reprojection (multi-tracking) (mouvement dans l'espace)



(c) Evolution de l'erreur de reprojection (multi-tracking) (mouvement sur le plan)

FIGURE A.10 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant une translation selon l'axe X de l'objet 0.

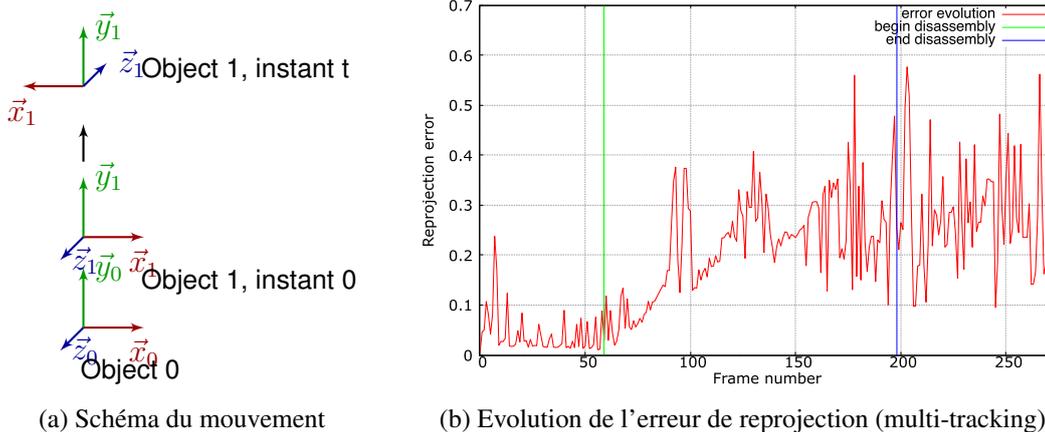


FIGURE A.11 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant un vissage selon l'axe Y de l'objet 0.

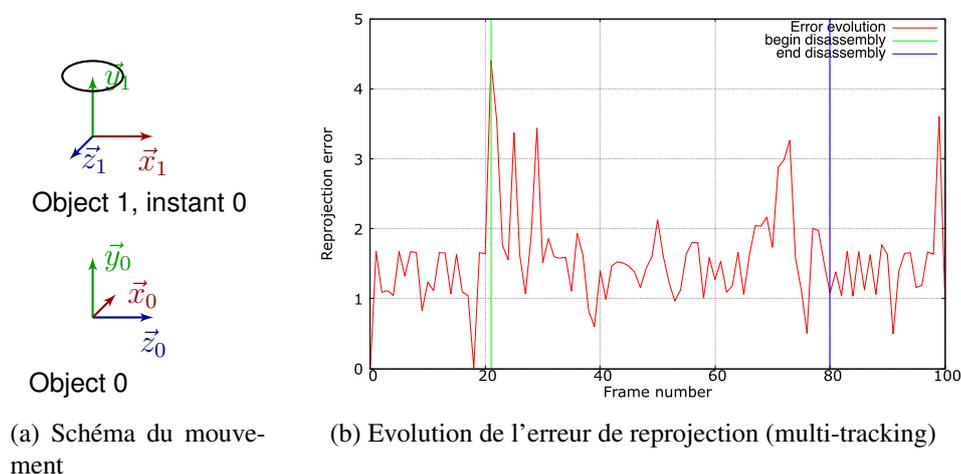


FIGURE A.12 – Vidéo synthétique représentant le désassemblage de deux objets suivant une rotation autour de l'axe Y de l'objet 0.

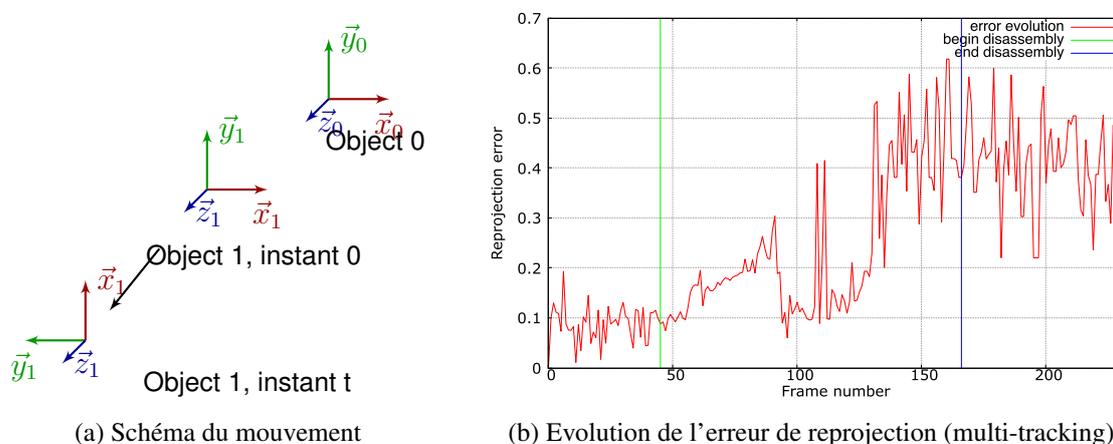


FIGURE A.13 – Vidéo dans des conditions réelles représentant le désassemblage de deux objets suivant un vissage selon l'axe Z de l'objet 0.

Annexe B

Résultats : Caractérisation du mouvement à l'aide de la matrice de transformation

De plus amples résultats obtenus pour la caractérisation du mouvement à l'aide de la matrice de transformation entre deux objets.

B.1 Mouvement par rapport à un objet de référence statique

La plupart des graphiques illustrent parfaitement les mouvements ayant lieu dans la scène (fig. B.5a, B.4a). D'autres sont beaucoup plus bruitées notamment lors de la phase d'assemblage (fig. B.1a, B.2a). Ceci est dû à la mauvaise estimation des matrices de projection lors de cette phase du fait de la coplanarité des points d'un des objets.

NB : A revoir potentiellement les seuils pour les cas des pivots dont l'angle de rotation diminue et dont le vecteur de rotation change de sens. (fig. B.3a, ??)

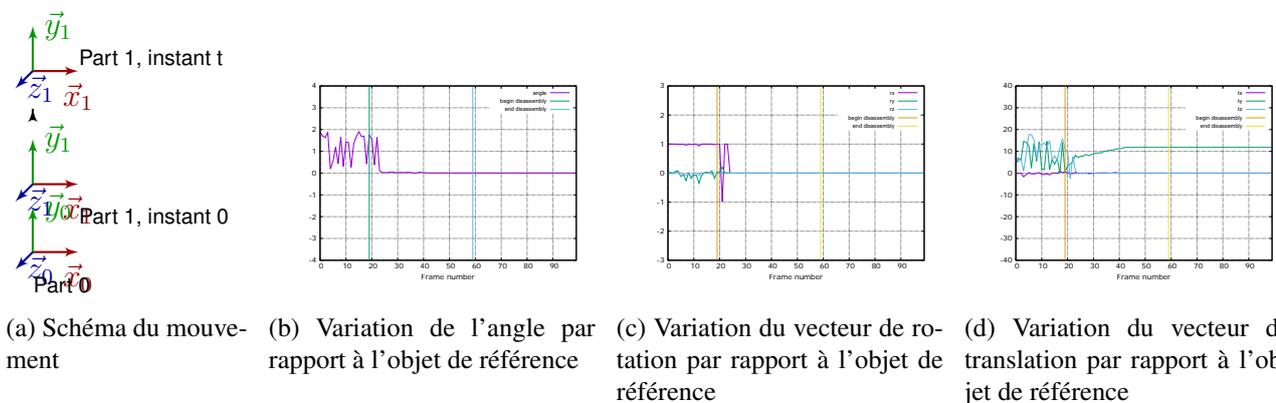


FIGURE B.1 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Y de l'objet 0.

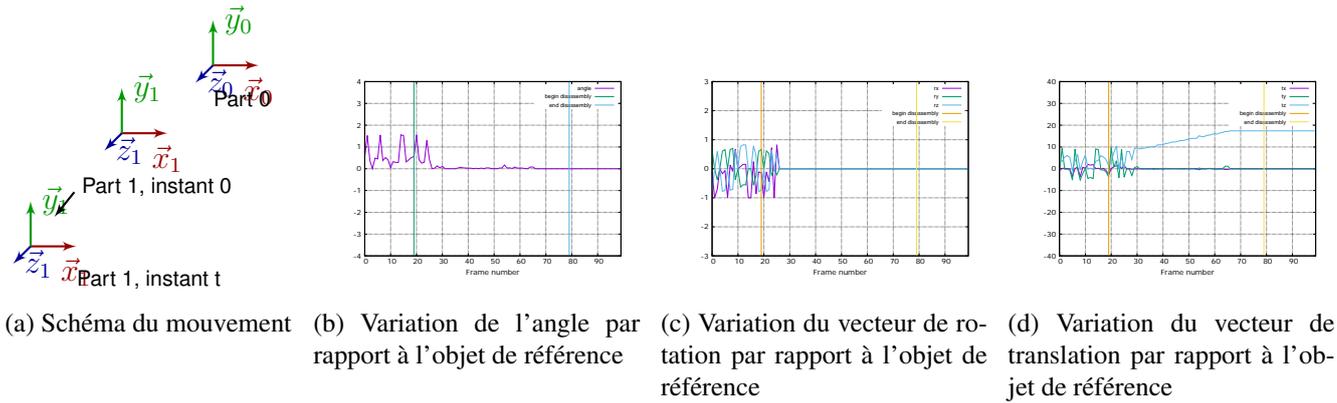


FIGURE B.2 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Z de l'objet 0.

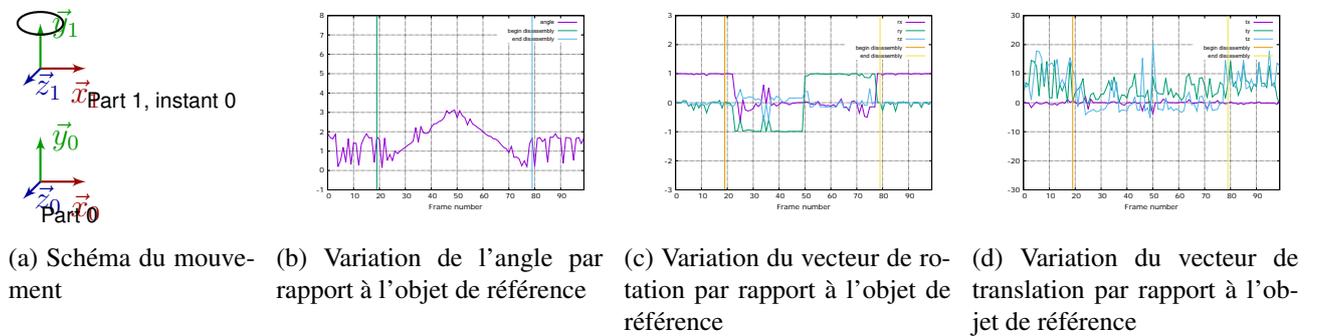


FIGURE B.3 – Vidéo synthétique représentant le désassemblage de deux pièces suivant un pivot autour de l'axe Y de l'objet 0.

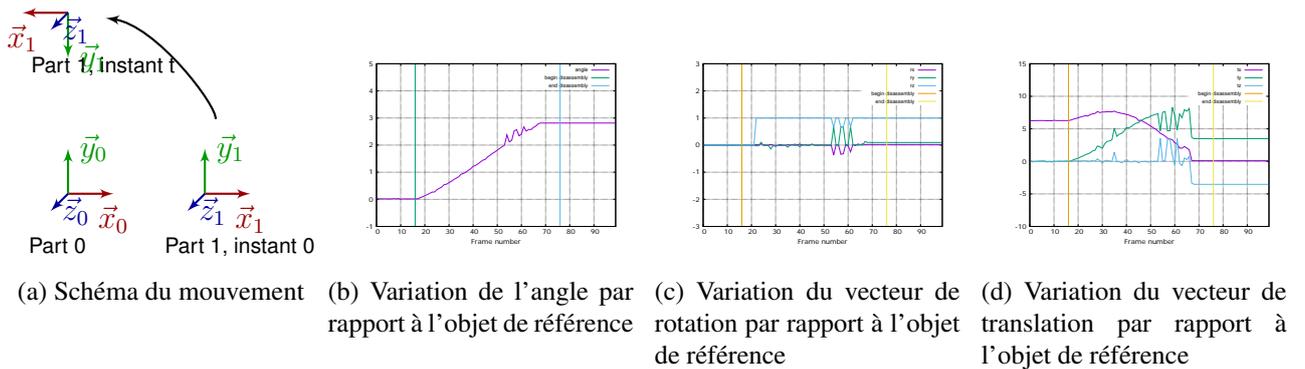


FIGURE B.4 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Z de l'objet 0.

B.2 Mouvement par rapport à la position initiale de l'objet en mouvement

Les mouvements correctement interprétés (fig. B.9a, B.10a)

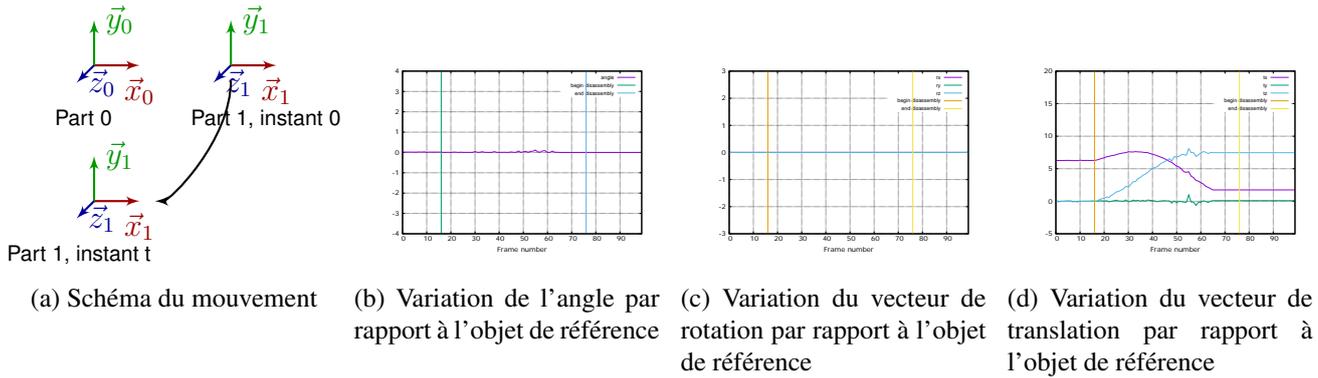


FIGURE B.5 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0.

NB : Mouvement à revoir (fig. B.6a, B.7a, B.8a, ??)

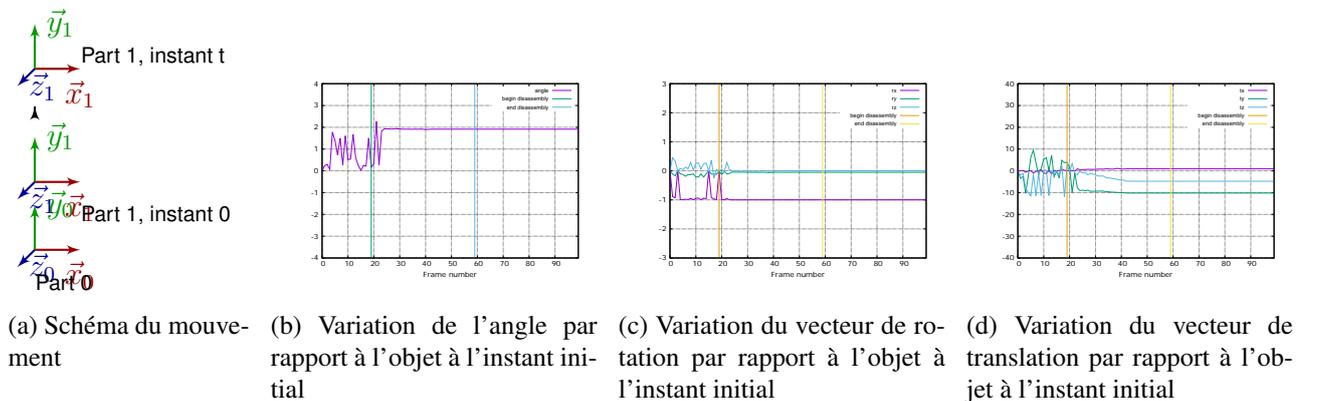


FIGURE B.6 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Y de l'objet 0.

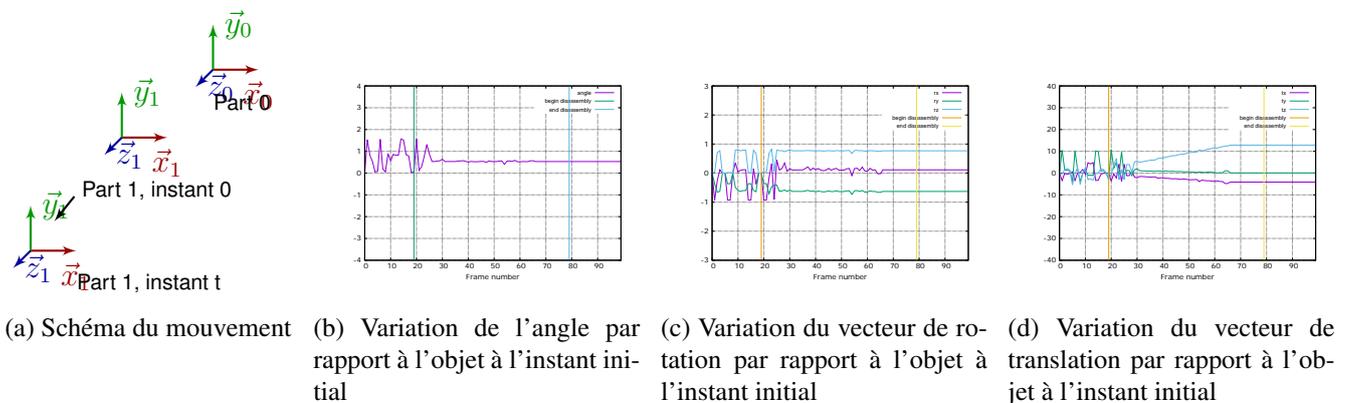


FIGURE B.7 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une translation selon l'axe Z de l'objet 0.

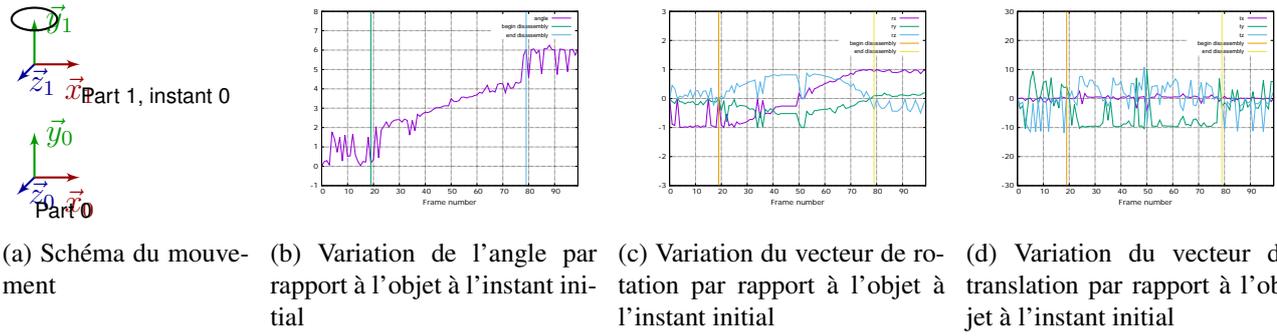


FIGURE B.8 – Vidéo synthétique représentant le désassemblage de deux pièces suivant un pivot autour de l'axe Y de l'objet 0.

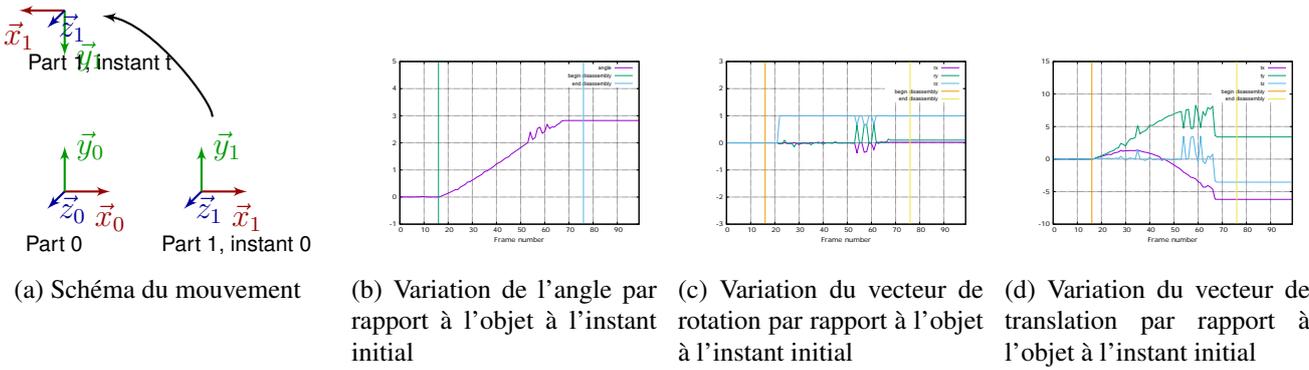


FIGURE B.9 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Z de l'objet 0.

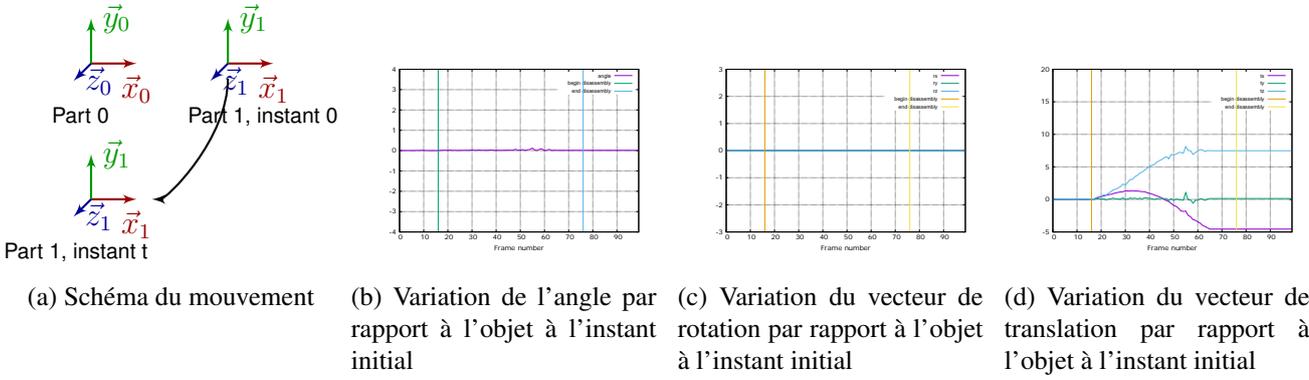


FIGURE B.10 – Vidéo synthétique représentant le désassemblage de deux pièces suivant une rotation autour de l'axe Y de l'objet 0.

Annexe C

Travaux en entreprise

Durant le cours de l'année, j'ai été impliquée dans différents travaux notamment dans celui concernant l'implémentation d'un SDK "What's That Face" qui permet une reconnaissance des visages, du genre, des émotions et de divers accessoires (lunettes, moustaches, piercings,...) et celui concernant un PoC (proof of competence) pour GDF Suez qui devait permettre la visualisation d'un champ d'éoliennes.

C.1 SDK "What's That Face"

Le SDK "What's That Face" en cours de développement par l'entreprise Wassa est un outil qui devrait mettre à la disposition d'autres développeurs sur mobile des modules permettant la reconnaissance des visages, du genre, des émotions et de divers accessoires. Ces caractéristiques peuvent être intéressantes dans des domaines telles que la publicité en permettant d'adapter le contenu des spots publicitaires suivant les intérêts potentiellement identifiables du spectateur.

Cet outil a été notamment intégré dans l'application "What's That Game" (produit par Wassa) qui est disponible sur Google Play Store. Ce dernier regroupe de mini-jeux et permet de personnaliser son avatar à l'aide de la reconnaissance des caractéristiques du visage de l'utilisateur.

Dans ce projet, mon objectif a été d'implanter un algorithme pour la reconnaissance des genres.

C.1.1 Etat de l'art

La classification des genres a été largement explorée dans la littérature scientifique. Même si pour l'être humain, identifier si un visage est de type féminin ou masculin est assez simple, cette tâche est un challenge pour un ordinateur. De nombreuses approches ont été proposées pour résoudre cette problématique en terme de précision, de rapidité et de robustesse.

Les principales étapes pour la détermination du genre correspondent à, tout d'abord, une phase de pré-traitement suivie d'une extraction de descripteurs et enfin d'une classification [Khan et al., 2011].

Certaines catégories des méthodes utilisent directement les valeurs de niveau de gris ou de couleur comme caractéristiques [Moghaddam and Yang, 2000, Abdi et al., 1995, Gutta et al., 2000] ou encore des critères géométriques notamment des distances entre des points d'intérêt détectés sur le visage [Mozaffari et al., 2010, Fellous, 1997, Brunelli and Poggio, 1993]. D'autres extraient des descripteurs tels que les HoE (Histogram of edges) ou HoG (Histogram of Gradient) [Ardakany and Joula, 2012], les LBP (Local Binary Pattern) [Lian and Lu, 2006, Yang and Ai, 2007, Alexandre, 2010], les SIFT (Scale Invariant Feature Transform) [Demirkus et al., 2010, Wang et al., 2010].

De nombreuses méthodes de classification sont également disponibles. L'algorithme du SVM (Support Vector Machine) est très populaire et donne des résultats de l'ordre de 92-96% de bonne classification sur un sous-ensemble de la base d'images FERET et CAS-PEAL [Mozaffari et al., 2010, Lian and Lu, 2006, Alexandre, 2010]. Adaboost et le RBF (Radial Basis Functions) combiné à un arbre de décision sont également utilisés avec des résultats de l'ordre de 79-96% [Gutta et al., 2000, Ng et al., 2012].

C.1.2 Algorithmes choisis et implémentés

En se basant sur les résultats obtenus dans la littérature et en tenant compte du fait que les algorithmes doivent être exécutés sur smartphones, nous avons choisi de tester les descripteurs suivant HoG et LBP du fait de leur rapidité de calcul combinés au classifieur SVM, ainsi que le fisherface qui applique la discriminante linéaire de Fisher sur des visages afin de déterminer le genre d'un visage.

La phase de prétraitement consiste à d'abord détecter le visage en utilisant l'algorithme de Viola et Jones [Viola and Jones, 2004] qui est implémenté sur OpenCV. L'alignement des visages est basée sur une détermination de la position des yeux à l'aide des gradients [Timm and Barth, 2011]. Cette étape est importante pour "redresser" les visages, notamment celles potentiellement inclinées.

Notre base d'images est une combinaison des bases de FERET, JAFFE, MUCT. Au début des expérimentations, l'apprentissage et les tests ont été effectués sur un ensemble de 323 femmes et 323 hommes pour l'apprentissage et de 79 femmes et de 190 hommes pour les tests extraites de la base d'images FERET. Nous avons obtenu :

- 71% de bonne reconnaissance au total (72,15% pour les femmes et 70,5% pour les hommes) avec le descripteurs HoG et un classifieur SVM linéaire.
- 82% de bonne reconnaissance au total (92,4% pour les femmes et 78,94% pour les hommes) avec le fisherface. Un pourcentage de 87% était obtenu en utilisant le canal

Value du HSV.

Au vu des résultats obtenus, le Fisherface a été retenu pour la classification. Comme lors des tests en ligne, des variations dans la prédiction du genre d'une même personne selon la manière dont elle regarde la caméra avant du smartphone. En effet, les visages légèrement inclinés vers le haut ou vers le bas ou qui sont dirigés vers la gauche ou vers la droite n'étaient pas incluses dans la base de données. Afin d'améliorer les résultats, nous avons enrichi et agrandi notre base d'images d'apprentissage et de tests avec un ensemble de 5005 femmes et de 5005 hommes pour l'apprentissage et de 3527 femmes et de 836 hommes pour les tests. Nous avons obtenu 80,79% de bonne reconnaissance au total avec 84,91% pour les femmes et 63,4% pour les hommes.

C.1.3 Conclusion et perspectives

Au cours de nos expérimentations, nous avons constaté, en général, un décalage entre les résultats annoncés par les auteurs de diverses méthodes et ceux obtenus. Par ailleurs, l'enrichissement de la base d'apprentissage a permis de tenir compte des différents positionnement du visage de l'utilisateur ainsi que de leur origine ethnique. Cependant, la méthode employée n'est pas suffisamment robuste à la luminosité qui peut rendre la prédiction aléatoire. Des améliorations peuvent être réalisées en intervenant aussi bien dans la phase d'extraction de détection des points d'intérêt et d'extraction de descripteurs mais également dans le choix du classifieur.

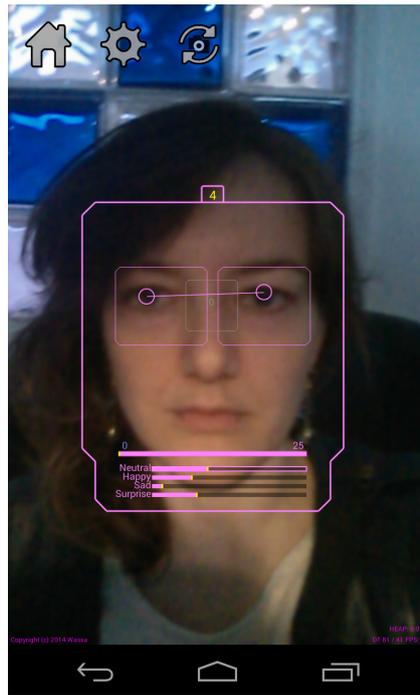
C.2 PoC GDF Suez

Un PoC pour GDF Suez a été réalisé par Wassa dans le cadre d'une campagne de "dédiabolisation" des implantations des champs d'éoliennes prévues en bordure de mer au Tréport et à Noirmoutier. Cette application devait permettre aux habitants de visualiser ces champs d'éoliennes et favoriser leur acceptabilité auprès du public.

L'analyse de l'impact paysager lié à l'installation de matériels de production d'énergie renouvelable est importante pour son acceptabilité. Il est nécessaire de prendre en compte l'aspect visuel et sonore de la perception humaine lors du processus d'implantation d'un parc d'éoliennes aussi bien terrestres qu'offshore.

Habituellement pour réaliser cette analyse et déterminer le meilleur emplacement pour l'installation des éoliennes, d'énormes tours de mesure sont érigées à un endroit précis pour recueillir les données relatives à l'environnement immédiat de la tour. Cette procédure est très coûteuse aussi bien financièrement qu'en terme de temps.

De plus, les représentations proposées par les opérateurs éoliens sont basées sur des photo-montages avec un nombre de point de vue réduit, des cartes ou des croquis qui ont un point de vue statique et ne donnent ainsi qu'un aperçu de l'installation à un instant et une échelle donnés



(a) Bonne reconnaissance femme.



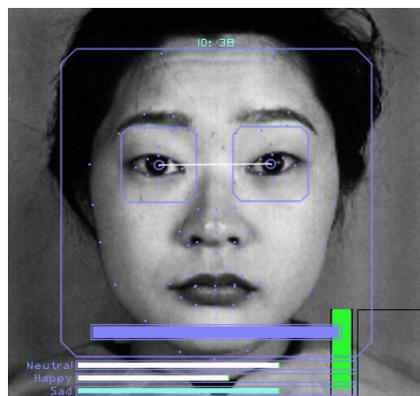
(b) Bonne reconnaissance homme.



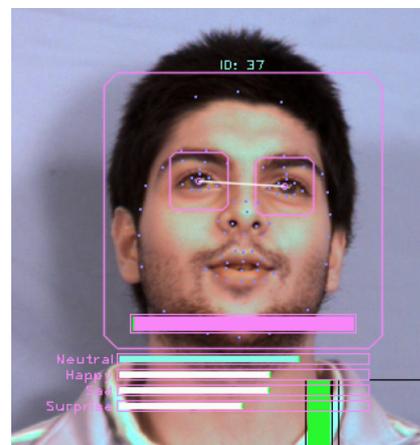
(c) Bonne reconnaissance femme.



(d) Bonne reconnaissance homme.



(e) Mauvaise reconnaissance femme.



(f) Mauvaise reconnaissance homme.

FIGURE C.1 – Résultats de la classification des genres à l'aide du Fisherface.

et qui sont, eux aussi, figés.

Une solution de réalité augmentée serait une alternative abordable notamment sur des appareils mobiles qui permettraient une observation moins contraignante du potentiel du parc d'éoliennes. En effet, les appareils mobiles tels que les smartphones et les tablettes ont largement évolué ces dernières années. Ils incluent du matériel spécifique comme des écrans à haute définition, des processeurs efficaces et performants, des capacités d'établir un réseau, des moteurs de jeu, des caméras intégrées et d'autres capteurs. Il s'agit de matériel de type « tout en un » pouvant constituer un support idéal pour des applications de réalité augmentée. Dans ce domaine, il est nécessaire d'avoir à sa disposition des capteurs tels que le magnétomètre qui agit comme une boussole, l'accéléromètre qui permet de déterminer l'orientation, le gyroscope pour le calcul de la rotation angulaire et la stabilité du mouvement. Or ces informations peuvent être retrouvées sur un smartphone ou une tablette permettant ainsi non seulement de localiser exactement et correctement l'utilisateur par rapport à son environnement mais également de répondre à la question suivante : « Qu'observera-t-on après l'implantation des champs d'éoliennes ? ». Cependant une autre question précède cette dernière : « Comment ? ».

C.2.1 Etat de l'art

Afin d'établir des solutions et des alternatives à cette dernière question, un récapitulatif des systèmes de réalité augmentée permettant la visualisation de parcs d'éoliennes sera présenté en première partie. En deuxième partie, les méthodes de visualisations dans d'autres domaines seront évoquées.

a) Suivi à l'aide de marqueurs

[Ota et al., 2010] proposent de positionner, dans l'environnement, les modèles 3D des éoliennes l'aide de quatre marqueurs, au lieu d'un seul. Les quatre marqueurs sont placés de telle sorte qu'ils forment un large carré (cf. fig.C.2). Celui-ci sera ensuite traité comme un unique marqueur. Ils utilisent l'ARToolkit et OpenCV pour l'identification des marqueurs. Ils récupèrent ainsi l'origine de chaque marqueur qui constitue un des sommets du marqueur final. Ils obtiennent un positionnement correct et efficace de l'objet virtuel. Cette solution proposée permet de remédier à l'instabilité du recalage lors de l'utilisation d'un unique marqueur dès le départ.

Cependant, même si l'utilisation de cibles codées pour le suivi est pratique, facile à mettre en place et rapide, elle reste très contraignante dans un environnement extérieur. En effet, il faudrait au préalable « préparer » l'environnement de test afin d'obtenir une connaissance a priori de l'endroit.



FIGURE C.2 – Insertion d’une éolienne dans un environnement extérieur. [Ota et al., 2010]

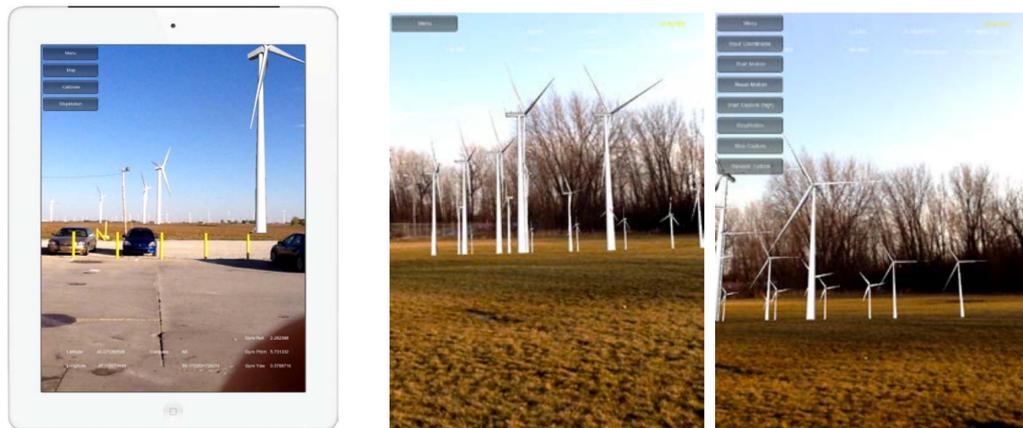
b) Suivi sans marqueurs à l’aide de capteurs

Le suivi sans cibles codées est une alternative la mieux adaptée pour des applications de réalité augmentée dans un environnement extérieur. Une des méthodes assez récurrentes est celle se réalisant avec des capteurs de position et d’orientation.

Ainsi, [Dekker et al., 2013] proposent MARWind (Mobile Augmented Reality Wind Farm Visualization). Il s’agit d’une application de réalité augmentée sur mobile (notamment sur Apple iOS) pour l’implantation d’un parc d’éolienne proposée dans le cadre d’un projet intitulé « Mixed Reality Simulation for Wind Energy Education » (cf. fig.C.3a). Les utilisateurs visés étaient aussi bien des étudiants que le public ou des professionnels de l’énergie éolienne. Ils utilisent les données du GPS (Global Positioning System) couplées à celle du magnétomètre, de l’accéléromètre et du gyroscope de l’appareil pour déterminer la position des modèles 3D des éoliennes dans l’environnement réel ainsi que l’orientation de l’utilisateur. Leur application permet à l’utilisateur de créer des éoliennes virtuelles et de les superposer sur les données réels et ainsi de voir la répartition de celles-ci. Les modèles 3D ont été créés à l’aide de 3DSMax et chargé dans l’application. L’utilisateur peut décider de la position GPS de chaque turbine et la visualiser directement sur Google Map View ou sur une fenêtre de rendu. Il peut également inspecter les informations relatives aux turbines tels que les coordonnées GPS, la distance entre l’utilisateur et la turbine, le modèle de l’éolienne,...

Cependant des problèmes d’occultations peuvent exister. En effet, l’utilisateur est censé se trouver à 5 mètres de la turbine afin d’éviter ces désagréments. Par ailleurs, la surface sur laquelle doivent être implantée les éoliennes doit correspondre à une surface plane texturée.

Une solution mobile de réalité augmentée utilisant les mêmes données en entrée est commercialisée par Linknode Ltd. Il s’agit de Ventus AR. Il récupère des données géospaciales à l’aide de Giality [Hoult, 2012] pour la visualisation en temps réel de parcs d’éoliennes pour



(a) MARWind : Application de RA pour l'implantation d'éoliennes utilisée dans un champ d'éoliennes pré-existante montrant ainsi les turbines réelles et virtuelles.

(b) Rendu des turbines selon différentes orientations dans un environnement réel à l'aide de MARWind.

FIGURE C.3 – MARWind. [Dekker et al., 2013]

leur implantation et pour analyser la faisabilité du projet ainsi que leur impact.

c) Visualisation de divers sites industriels

D'autres domaines industriels s'aident de la réalité augmentée lors du processus d'installation d'une infrastructure particulière tels que pour les réseaux de câbles de courants ou de canalisation ou encore lors du processus de réaménagement du parc urbain.

i) Smart Vidente

Smart Vidente [Schall et al., 2013] est un système mobile de réalité augmentée tirant parti des données géo spatiales qui peuvent être nécessaire aux architectes et aux ingénieurs civils. Il est utilisé dans le cadre d'installations de structures suburbaines (cf. fig.C.4c). Ces infrastructures enfouies telles que les câbles d'électricité, les conduits d'eau ou d'aération sont superposées à la séquence d'images récupérées en temps réel de l'environnement.

Il donne la possibilité à l'utilisateur de modifier les données géo spatiales et ainsi de les mettre à jour. Il lui permet ainsi d'inspecter le site en temps réel, de planifier le projet, de recueillir des données et de réaliser l'arpentage du site.

Plusieurs capteurs sont intégrés dans le système Smart Vidente pour un suivi et un recalage correcte et efficace des modèles 3D à insérer dans la scène réelle (cf. fig.C.4b). Il s'agit du GPS, d'un capteur inertiel et d'une caméra. Les données du GPS combinées au filtre de Kalman permettront de réaliser le suivi de la position. Tandis que les données visuelles de la caméra ainsi que les informations produites par le capteur inertiel combinées également au filtre de Kalman

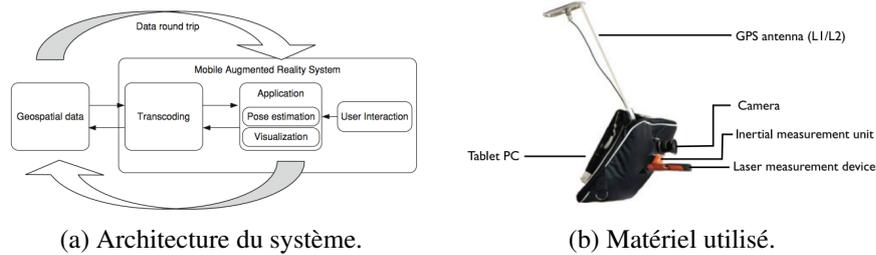


FIGURE C.4 – Smart Vidente. [Schall et al., 2013]



FIGURE C.5 – Utilisation de SiteLens pour l'analyse spatiale des données. Ici, il s'agit de l'analyse de la production de CO sur une rue donnée. [White and Feiner, 2009]

permettront le suivi de l'orientation de l'appareil mobile.

ii) SiteLens

SiteLens [White and Feiner, 2009] est un système de réalité augmentée permettant aux urbanistes et aux architectes de mieux analyser spatialement les données recueillies (cf. fig.C.5). Il s'agit d'une « visualisation située ». Les données géo spatiales enregistrées au préalable dans une base de données sont utilisées pour situer les informations que l'utilisateur désire visualiser dans une scène réelle. Le suivi est réalisé à l'aide cibles codées disséminées dans l'environnement.

C.2.2 Algorithme choisi et implémenté

Comme l'implantation des éoliennes a été prévu en mer, la stratégie adoptée a consisté à, tout à d'abord, détecter la ligne d'horizon, puis à effectuer un recalage des données en fonction de la position de cette ligne et des données géospatiales fournies par GDF Suez. Mon travail a essentiellement consisté à implémenter un algorithme de détection de ligne d'horizon.

La détection de la ligne d'horizon est une étape souvent incluse pour la stabilisation automatiques des véhicules aériens sans pilotes [Chen et al., 2011, Georgilas et al., 2014] ou des micro-véhicules aériens [Ettinger et al., 2002b]. Plusieurs algorithmes ont été proposés pour effectuer cette détection notamment dans un environnement marin. Ils permettent de déterminer la zone de séparation en le ciel et la mer. On peut relever notamment, dans ce cadre, les algorithmes basés sur :

- une covariance régionale [Ettinger et al., 2002a],
- une détection de contours et la transformée de Hough [Gershikov et al., 2013],
- une détection de contours et une minimisation aux moindres carrés,
- un filtrage médian et une régression linéaire.

Une comparaison de ces différentes méthodes a été réalisée par [Gershikov et al., 2013]. Ils montrent que l'algorithme basé sur une détection des contours et la transformée de Hough présentent de meilleurs résultats en terme de précision et de rapidité par rapport aux autres approches. Ce critères étant des éléments important pour des applications mobiles, nous avons ainsi implémenté cette méthode.

L'approche choisie consistait à réaliser, d'abord, une érosion morphologique sur l'image à l'aide d'un élément structurant circulaire afin de la lisser et supprimer les contours faibles (correspondant à des éléments n'appartenant à la ligne d'horizon à détecter) (cf. fig.C.6b). L'algorithme de Canny [Canny, 1986] est ensuite appliqué afin d'extraire les contours dans l'image (cf. fig.C.6c). La transformée de Hough [Duda and Hart, 1972] permet de déterminer les lignes droites parmi les contours. La ligne d'horizon choisie est celle qui sera la plus longue et avec un écart d'angle le plus faible (cf. fig.C.6d).

En tenant compte de la position de la ligne d'horizon, le parc d'éoliennes superposé à l'image acquise (cf. fig.C.7).

C.2.3 Conclusion et perspectives

Les modèles 3D virtuelles géo localisées sont superposées sur les données visuelles de la scène réelles à l'aide de méthodes utilisant des cibles codées ou sans cibles à l'aide de capteurs tels que le GPS, le magnétomètre, l'accéléromètre et le gyroscope.

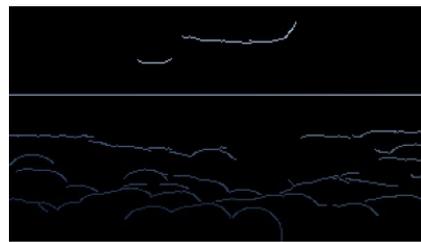
Très peu d'applications de réalité augmentée sont effectivement destinées à la visualisation des parcs d'éoliennes. Certaines sont basées sur des marqueurs à placées dans l'environnement



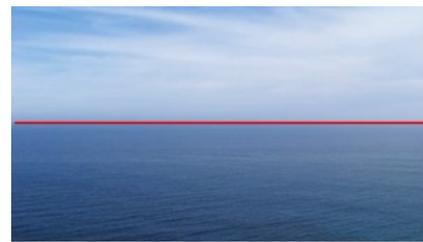
(a) Image originale.



(b) Erosion morphologique.



(c) Détection de contours à l'aide de l'algorithme de Canny.



(d) Application de la transformée de Hough et résultat.

FIGURE C.6 – Etapes pour la détection de la ligne d'horizon.



FIGURE C.7 – Rendu final de la superposition.

réels qui est une méthode contraignante dans un environnement inconnu. D'autres imposent des caractéristiques spécifiques concernant la surface du sol sur lequel seront installées les éoliennes.

Une application de réalité augmentée permettant aussi bien l'évaluation non seulement visuelle mais également sonore ajoutera plus de réalisme lors des simulations. Si elle affichait également spatialement les données déjà recueillies par les professionnels éoliens, une localisation plus rentable des éoliennes sur un site précis pourrait être établie. Une visualisation des infrastructures environnantes tels que le réseau de distribution d'électricité associé au parc éolien pourrait être un plus.

Notre approche s'est basée sur la détection de la ligne d'horizon afin de recalibrer les données acquises par la caméra et les autres capteurs du smartphone et les données géospatiales fournies. Des résultats assez satisfaisants ont été obtenus. Cependant, la rapidité des calculs aurait pu être améliorée en restreignant les zones de l'image à traiter en tenant compte de la ligne d'horizon qui pouvait être théoriquement déterminée à l'aide de capteurs tels que l'accéléromètre et la centrale inertielle.

Titre : Vers un système interactif de réalité augmentée mobile pour la supervision de scénarios de maintenance industrielle

Mots clefs : réalité augmentée, supervision de scénario, suivi, contrôle, maintenance industrielle

Résumé : En fournissant à l'utilisateur des informations pertinentes au bon moment au bon endroit, la réalité augmentée pourrait améliorer la productivité et l'efficacité des agents de maintenance industrielle. Cependant, les systèmes de suivi des scénarios implémentés nécessitent souvent leur intervention manuelle induisant ainsi insidieusement un manque de concentration, une fatigue visuelle, psychologique et physique et laissant également l'utilisateur seul juge de la bonne exécution des tâches de maintenance. C'est dans ce contexte que la société Wassa, spécialisée dans le web et les solutions mobiles et l'équipe IRA2 du laboratoire IBISC, spécialisée dans la réalité augmentée, la réalité virtuelle et la robotique, ont initié ce sujet de thèse. Il s'agit de concevoir un système de réalité augmentée permettant :

- de fournir à l'utilisateur les instructions de maintenance de manière automatique et en temps réel.
- de contrôler la bonne réalisation de la tâche et d'identifier les erreurs commises par l'opérateur à travers des dispositifs mobiles. Une notification doit être affichée afin de signaler à l'utilisateur qu'une tâche a été mal exécutée ou qu'une pièce mécanique n'a pas été correctement placée. Le système doit pouvoir enregistrer chacune de ces erreurs.
- de fonctionner sur n'importe quel appareil mobile en tenant compte des limitations liées à l'utilisation ces appareils, notamment concernant leur capacité de calcul et de stockage.

Title : Towards an interactive mobile augmented reality system for industrial maintenance scenario supervision

Keywords : augmented reality, scenario supervision, control, mobile, industrial maintenance, tracking

Abstract : By providing to the user relevant data at the right time in the right place, augmented reality might increase productivity and performance of industrial maintenance workers. However, implemented scenario tracking systems often need manual intervention. This may induce insidiously less of concentration, eyestrain, psychological and physical fatigue. Moreover, the operator is the sole judge of the maintenance task execution. It is in this context that Wassa company, specialized in web and mobile solutions, and IRA2 team of IBISC Laboratory, specialized in augmented reality, virtual reality and robotics, have initiated this PhD thesis. The goal is to design an

augmented reality system which will :

- provide automatically and in real time maintenance instructions to the user .
- control, through the mobile devices, the maintenance task achievement and identify the mistakes done by the operator. A notification will be displayed to report to the user that the task is incorrectly performed or that a mechanical piece is inaccurately placed. The system should store these errors.
- run on any mobile devices by taking into account their computation and storage limitations.

