

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Constantine
Faculté des Sciences de l'Ingénieur

Département d'Informatique

N° d'ordre: 37/TS/2008

N° Série: 03/inf/2008

THÈSE

Pour l'obtention du diplôme de
DOCTEUR EN SCIENCE

Spécialité
Informatique

Thème

**Manipulation d'Objets Réels et Virtuels
dans un Environnement de Réalité
Augmentée Collaboratif**

Présenté par

Brahim NINI

Directeur de thèse

Prof. Mohamed Batouche

Soutenu le : 24 Juin 2008

Composition du Jury

Prof. Mohamed Benmohammed

Président

Univ. Constantine

Prof. Noureddine Djedi

Examineur

Univ. Biskra

MC. Chaoui Allaoua

Examineur

Univ. Constantine

MC. Chaouki Babahennini

Examineur

Univ. Biskra

MC. Chikhi Salim

Rapporteur

Univ. Constantine

*A l'âme de mon père qui reste toujours présent dans mon esprit,
A ma mère à qui je ne pourrais jamais rendre le bienfait quoique je fasse
A ma femme qui m'est un grand soutien,
A mes fils Fadi El-Islam, Dhia Eddine et Roua Hazar,
A mon frère et à mes Sœurs*

Remerciements

Ce travail est le résultat d'un effort non négligeable, qui a nécessité le soutien et l'encouragement d'un certain nombre de personnes envers lesquels je tiens à exprimer ma gratitude.

Je tiens tout d'abord à remercier mon encadreur, le Professeur Monsieur Mohamed Batouche pour le sujet proposé, son suivi, et son encouragement. Je tiens principalement à le remercier pour le temps qu'il m'a accordé pour la finalisation de cette thèse.

Je tiens également à remercier mon Co-Encadreur Monsieur Chikhi Salim qui m'a honoré d'avoir accepté de suivre ce travail.

Je remercie les membres du Jury pour avoir accepté d'accorder une partie de leur temps à lire cette thèse, à l'évaluer et à la critiquer.

Je remercie tous les membres de l'équipe Vision and Computer Graphics Group qui m'ont communiqué une quelconque idée d'une façon ou d'une autre à un moment donné et particulièrement mon collègue Bouzenada Mourad avec qui j'ai travaillé en étroite collaboration.

Je n'oublierais pas de remercier toute ma famille, ma femme, mon frère, mes sœurs et leurs familles chez qui j'ai trouvé soutien et encouragement.

Mes mots sont réellement des moules trop étroits pour pouvoir contenir mes sincères sentiments et remerciements envers toutes ces personnes. Merci beaucoup.

ملخص المذكرة

تعتبر هذه المذكرة وصفا شاملا لحل مبني على أساس الفيديو من أجل تمثيل التحكم في شيء افتراضي أو حقيقي على حدّ سواء داخل مشهد حقيقي محول. نناقش في هذا المحتوى المظاهر النظرية والتطبيقية للتحكم من خلال دراسة تتكون من ثلاثة أجزاء ليست متتابعة غير أنّها متممة بعضها بعضا. يهتم الجزء الأول بالخلفية الرياضية للتحكم. في هذا الإطار، تعتبر دورانية الأشياء حسب ثلاثة أبعاد قد حلت كليا فيما يبقى الحل المتعلق بالحركة يتطلب بعض التحسين. أما الجزء الثاني، فيتعلق بالعملية التي لها صلة بصور شيء حقيقي وكيفية أخذها وكذا تخزينها ثم طريقة دمجها في فلم ما من أجل تحويله. الهدف من هذه العملية هو أن تبنى علاقة بين الصور المخزنة والحركية الظاهرة. هذه العلاقة مبنية على نظرة هندسية ممثلة في شكل طريقة تحليلية. أخيرا، يعتبر الجزء الثالث حوصلة لكيفية تطبيق النتائج السابقة في بيئة مكونة من مجموعة مستعملين متعاونين. لذا، أعطيت هندسة البرنامج التي تؤسس للتعاون المقترح مع مناقشتها نسبة للغرض من هذا العمل.

Résumé de la thèse

Cette thèse décrit une solution complète, à base d'une structure vidéo, pour la simulation de la manipulation d'un objet réel ou synthétique dans une scène augmentée. Son contenu discute les aspects théoriques et pratiques à travers trois parties qui ne sont pas consécutives mais complémentaires. La première partie met le point sur les bases mathématiques en relation avec la manipulation supportée. Les trois degrés de liberté de la rotation sont entièrement satisfaits, par contre les solutions des translations nécessitent encore certaines améliorations. La seconde partie s'intéresse au processus en relation avec les images de l'objet et les opérations associées nécessaires à leur prise et à leur stockage, ainsi que leur incrustation dans la séquence à augmenter. Le but de ce processus est d'établir le lien entre les images stockées et le mouvement simulé. Cette correspondance est basée sur la géométrie et est exprimée sous une forme analytique. Finalement, la troisième partie généralise la manipulation à un groupe d'utilisateurs dans un environnement collaboratif. Une architecture logicielle qui supporte la collaboration supportée est donnée et discutée relativement à l'objectif de ce travail.

Thesis's Abstract

This thesis describes a complete movie-based solution for simulating the manipulation of both real or synthetic objects in an augmented scene. Its content discusses the theoretical and practical aspects through three parts that are not consecutive but completing each other. The first part focuses on mathematical background in relation to the supported manipulation. The three rotational degrees of freedom are entirely fulfilled whereas the translations' solutions still need some improvement. The second part is concerned with the process in relation to the images of the object and the related operations required for the grab and storage of the images and their integration in the sequence to be augmented. The aim of the process is to make a link between the stored images and the simulated movement. This correspondence is geometry-based and expressed in an analytical way. Finally, the third part generalizes the manipulation to a group of users in a collaborative environment. A software architecture that supports the proposed collaboration is given and discussed relatively to the purpose of this work.

Table des matières

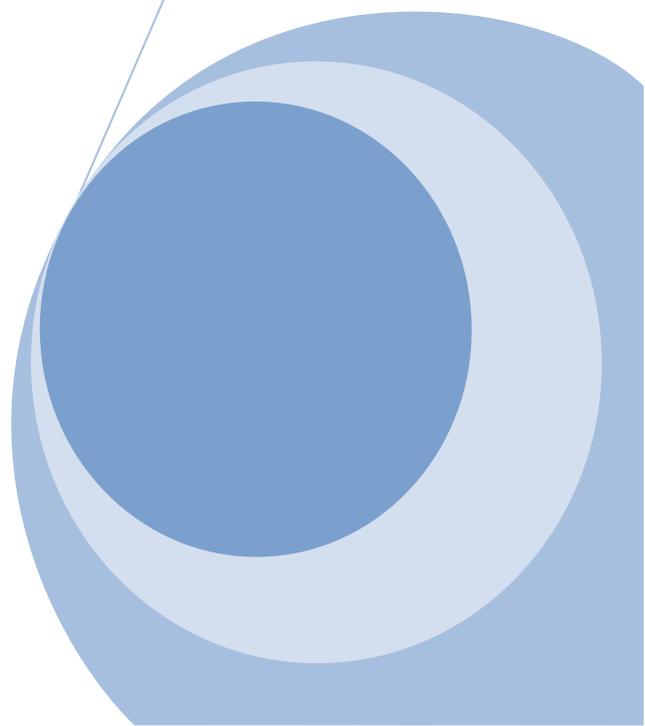
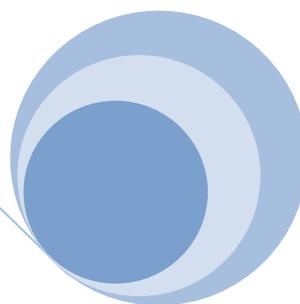
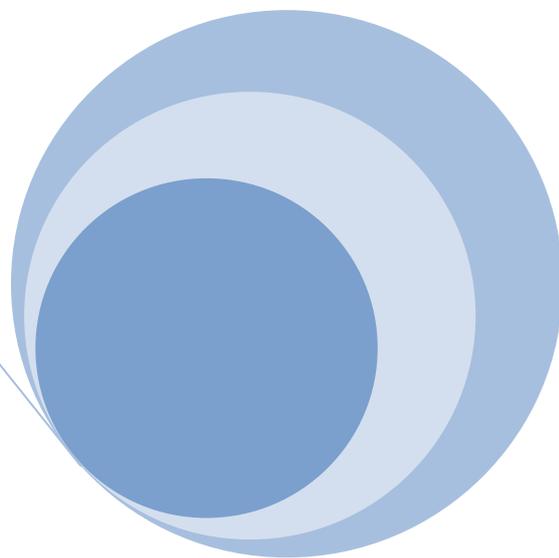


Table des matières

Introduction	1
Chapitre I. Réalité Augmentée Collaborative	
1. Introduction	8
2. Définition de la réalité augmentée collaborative	9
3. Caractéristiques de la réalité augmentée collaborative	10
3.1. Objets virtuels	10
3.2. Systèmes d'augmentation	11
3.2.1. Système optique	12
3.2.2. Système vidéo	13
3.3. Mixture visuelle de réel et virtuel	14
3.4. Interaction avec le virtuel	15
3.5. Collaboration des utilisateurs	15
3.6. Pluralité du domaine	16
3.7. Réalité augmentée: outil pour d'autres applications	16
4. Principales orientations de la recherche en réalité augmentée	17
4.1. Réalisme visuel	18
4.2. Calibration de la caméra	18
4.3. Manipulation et interaction	20
4.4. Augmentation temps réel	22
4.5. Collaboration	23
5. Domaines d'application	25
5.1. Environnements de réalité augmentée collaborative	25
5.2. Enseignement	26
5.3. Entraînement et apprentissage	27
5.4. Télé présence	28
5.5. E-Commerce	28
5.6. Conception	29
5.7. Téléconférence et vidéoconférence	30
5.8. Récits historiques	31
5.9. Informations spatiales	31

6. Conclusion.....	32
--------------------	----

Chapitre II. Calibration de la Caméra pour les Augmentations 3-D

1. Introduction	35
2. Calibration de la caméra	35
2.1. Modèles de camera	36
2.2. Modèle sténopé	37
2.3. Projection perspective.....	38
2.4. Transformation caméra-image	39
2.5. Paramètres intrinsèques.....	40
2.6. Distorsion radiale.....	42
2.7. Paramètres extrinsèques	42
2.8. Transformation référence-image.....	44
3. Types de calibration	45
3.1. Calibrage fort	45
3.2. Calibrage faible.....	45
3.3. Calibrage automatique (ou auto-calibrage)	46
4. Indices utilisés dans l'augmentation.....	47
4.1. Références explicites.....	47
4.2. Références implicites	49
5. Indice planaire et auto calibration de la caméra.....	49
5.1. Homographie	49
5.2. Estimation des éléments de la matrice de projection 3-D	51
5.3. Algorithme de calibration à base d'indice planaire	53
5.3.1. Principe général	53
5.3.2. Recherche de la région de l'indice	54
5.3.3. Suivi de l'indice dans la séquence.....	55
5.3.4. Résumé de l'algorithme	56
6. Conclusion.....	58

Chapitre III. Manipulation des Objets Virtuels

1. Introduction	60
2. Processus de transformation.....	61
3. Projection des objets virtuels	62

3.1. Projection 2-D	64
3.2. Projection 3-D	65
4. Manipulation des objets virtuels.....	65
4.1. Manipulation 2-D.....	67
4.2. Manipulation 3-D: Fondements mathématiques.....	67
4.2.1. Matrice de transformation	68
4.2.2. Rotation autour d'un axe quelconque	69
4.2.3. Décomposition d'une transformation	72
5. Outils et interfaces de manipulation	73
5.1. Caractéristiques des outils de manipulation	74
5.2. Techniques et outils utilisés	76
5.3. Emulation de la manipulation 3-D par un outil 2-D.....	78
6. Conclusion.....	81

Chapitre IV. Objets Réels Virtualisés

1. Introduction	83
2. Travaux liés	84
3. Prise d'Images	85
4. Organisation des images	87
4.1. Ordre de stockage.....	87
4.2. Résolution des images.....	90
4.3. Restitution des images	90
5. Projection de l'objet	92
5.1. Principe de la projection des images de l'objet	92
5.2. Extraction de la région de l'image de l'objet	93
5.3. Contrainte de taille	94
6. Manipulation d'un objet réel virtualisé	97
6.1. Simulation d'une manipulation artificielle	97
6.1.1. Translations	97
6.1.2. Rotations	99
6.1.3. Récapitulation du principe d'augmentation et de manipulation.....	100
6.2. Simulation d'une manipulation naturelle	100
6.2.1. Translations	101
6.2.3. Rotations	103

7. Rotations libres à base d'images	104
7.1. Interprétation de l'opération de prise d'images	105
7.2. Rotations de l'objet autour des axes de ORF	106
7.3. Rotation de l'objet autour des axes de WCS	107
7.4. Rotation de l'objet autour d'axes arbitraires	107
7.5. Réalisation de la simulation	108
8. Discussion des résultats	109
9. Conclusion	111

Chapitre V. Manipulation dans un Environnement Collaboratif

1. Introduction	115
2. Principe de la collaboration	117
2.1. Gestion des utilisateurs	118
2.2. Visualisation des objets et simulation de leurs mouvements	119
2.2.1. Utilisateurs partageant le même espace physique	120
2.2.2. Utilisateurs géographiquement répartis	120
2.3. Manipulation collaborative	121
2.3.1. Manipulation par des utilisateurs partageant le même espace physique	121
2.3.2. Manipulation par des utilisateurs distants	122
3. Prototype de la plate forme implémentée	123
3.1. Architecture logicielle	123
3.1.1. Module central	123
3.1.2. Modules utilisateurs	126
3.2. Discussion de la solution	126
4. Conclusion	127

Conclusion	130
-------------------------	-----

Bibliographie	133
----------------------------	-----

Annexe A

1. Décomposition en valeurs singulières (DVS)	144
2. Systèmes homogènes	145

Table des figures

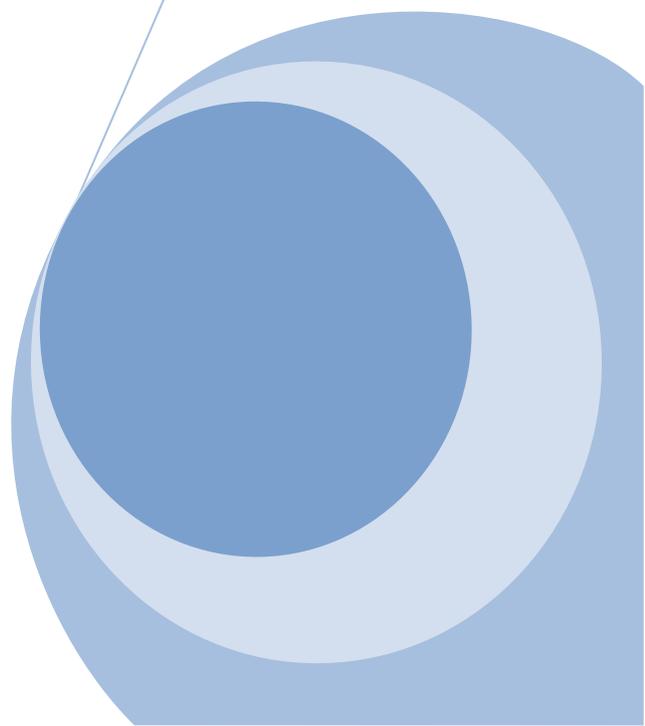
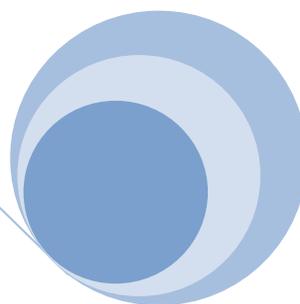
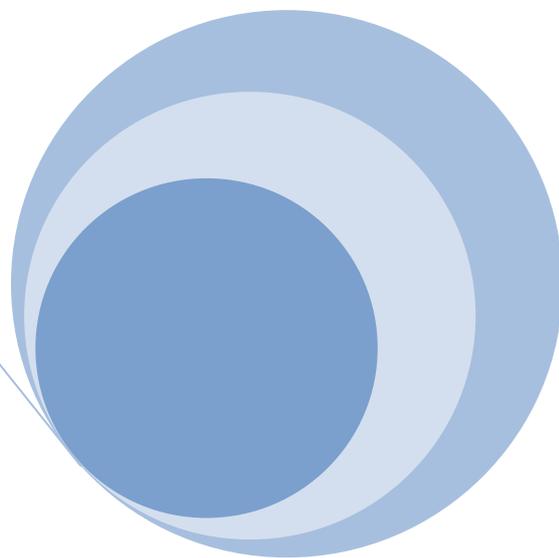


Table des figures

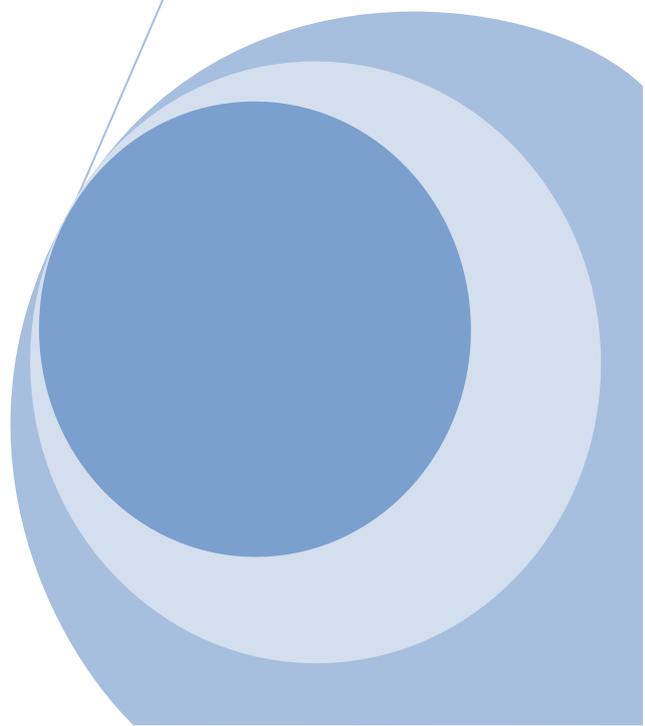
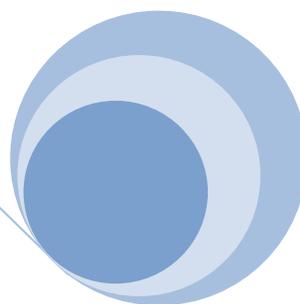
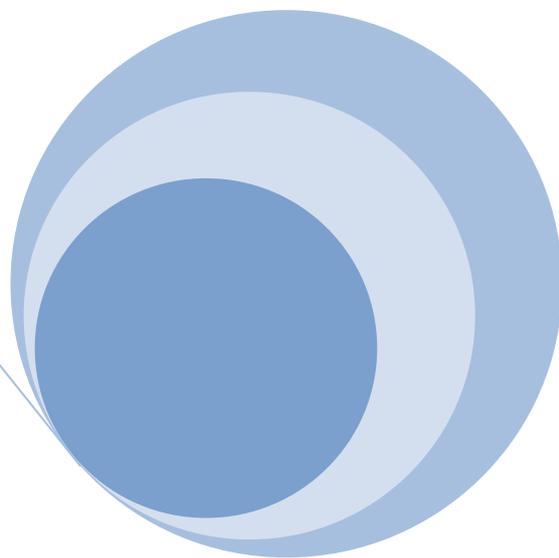
Figure I.1. Insertion d'un objet virtuel: à gauche l'image originale et à droite l'image augmentée	9
Figure I.2. Objet virtuel synthétique qui consiste en la modification de l'apparence de la scène.....	11
Figure I.3. Objet virtuel intégré en tant qu'image d'un objet réel.....	11
Figure I.4. HMD optique	12
Figure I.5. Schéma fonctionnel du HMD optique.....	13
Figure I.6. HMD vidéo.....	13
Figure I.7. Schéma fonctionnel du HMD vidéo	14
Figure I.8. Schéma fonctionnel d'une augmentation à base d'écrans ordinaires.....	14
Figure I.9. Augmentation avec une composition naturelle où il est difficile d'identifier les objets réels de ceux virtuels	15
Figure I.10. Jeux d'échec virtuel: l'interaction est faite à base d'objets réels.....	15
Figure I.11. Collaboration de deux utilisateurs dans la manipulation d'un même objet virtuel.....	16
Figure I.12. Exemple d'image où les couleurs de l'objet virtuel et ceux de la scène ne sont pas équilibrées	18
Figure I.13. Un objet virtuel suspendu dans l'air	18
Figure I.14. Indices de calibration de la caméra: à gauche un indice planaire et à droite une mire 3-D	19
Figure I.15. Indice de calibration composé de deux indices planaires ayant une forme 3-D permettant d'augmenter l'image	20
Figure I.16. Manipulation d'objets virtuels et interaction avec la scène augmentée	21
Figure I.17. Simulation du déplacement d'un robot avant son déplacement réel	21
Figure I.18. Collaboration de plusieurs utilisateurs dans la réalisation d'une tâche commune	23
Figure I.19. Plate forme spécialement conçue pour la réalisation d'applications de réalité augmentée collaborative	26
Figure I.20. Aperçu du musée virtuel du système JAPE	26
Figure I.21. Aperçu de l'application Construct3D	27
Figure I.22. Vue de l'application d'assistance dans l'apprentissage de la manipulation d'un piano.....	27
Figure I.23. Entraînement assisté dans la maintenance d'une carte électronique d'un ordinateur	28

Figure I.24. Visualisation d'une personne distante (dans ce cas un mannequin).....	28
Figure I.25. Agencement d'une pièce par des objets virtuels où en réalité elle ne contient rien (coin haut droit).....	29
Figure I.26. Application Table-Top pour la manipulation d'objets	29
Figure I.27. Représentation d'une face virtuelle d'un microbus réel durant la phase de conception.....	30
Figure I.28. Exemple d'applications de vidéo conférence	30
Figure I.29. Visualisation de personnages historiques associés à des emplacements	31
Figure I.30. Visualisation spatiale: Orientation intérieure dans une construction	32
Figure I.31. Visualisation spatiale: Recherche de livres dans une bibliothèque.....	32
Figure II.1. Modèle sténopé utilisé	37
Figure II.2. Modèle géométrique du modèle projectif linéaire	38
Figure II.3. Projection d'un point de l'espace dans le plan image	39
Figure II.4. Transformation d'un point du référentiel de la scène vers le référentiel de la caméra	43
Figure II.5. Principe de vues cylindriques ou mosaïque (gauche) et images plane ou warping (droite)	46
Figure II.6. Différents type d'indices utilisés pour réaliser des augmentations. Du haut en bas et de gauche à droite: mire 3-D, points colorés, points non colorés, formes géométriques et objets réels	48
Figure II.7. Objets planaires utilisés comme indice 2-D	48
Figure II.8. Augmentations à base d'indices planaires.....	49
Figure II.9. Augmentations se basant sur des éléments extraits de la scène et sans indices explicites	50
Figure II.10. Augmentation 2-D d'une image à base de l'homographie calculée	51
Figure II.11. Augmentation 3-D après estimation des paramètres de projection 3-D à partir de l'homographie	53
Figure II.12. Binarisation d'une image	54
Figure II.13. Détection des coins d'un indice planaire	54
Figure II.14. Augmentations 2-D et 3-D	55
Figure II.15. Coins d'un indice à rechercher et à suivre dans l'algorithme d'augmentation	56
Figure II.16. Résumé de l'algorithme d'augmentation à base d'indice planaire	56
Figure II.17. Effet de l'éclairage et du choix du seuil de binarisation sur l'image binarisée.....	57
Figure II.18. Effet d'une fausse détection des coins de l'indice sur l'augmentation obtenue: l'objet d'augmentation ne recouvre pas complètement l'indice	58

Figure III.1. Exemple d'applications pour la navigation dans une scène virtuelle (gauche) et réelle (droite).....	61
Figure III.2. Prise en compte de l'aspect d'ombre (droite) dans l'intégration d'objets virtuels	62
Figure III.3. Manipulation de l'objet par rapport au référentiel de l'indice	63
Figure III.4. Différence entre une rotation de l'objet par rapport à un référentiel suivie d'une translation (gauche) et d'une translation suivie par une rotation (droite)	66
Figure III.5. Exemple de translation et rotation 2-D d'un objet 2-D	67
Figure III.6. Exemple d'axe de rotation quelconque dans l'espace	69
Figure III.7. Exemple de translation et rotation 3-D d'objets 3-D.....	72
Figure III.8. Manipulation d'un objet virtuel 3-D via un outil tangible détecté par vision	73
Figure III.9. Différentes techniques utilisées dans la sélection et la manipulation d'objets virtuels 3-D	75
Figure III.10. Les techniques d'interaction nécessitent généralement des retours sensoriels.....	76
Figure III.11. Principe géométrique de la sphère virtuelle	78
Figure III.12. Rotation ayant comme origine pour le déplacement de la souris le centre du disque	79
Figure III.13. Rotation ayant comme origine pour le déplacement de la souris un point quelconque sur l'axe Ox du disque	79
Figure III.14. Rotation ayant comme origine pour le déplacement de la souris un point quelconque n'appartenant pas à l'axe Ox du disque	80
Figure IV.1. Sphère virtuelle utilisée pour la prise d'images des différentes vues d'un objet ...	85
Figure IV.2. Exemple d'images d'objets réels associées à différentes vues	86
Figure IV.3. Objet photographié à des distances égales en des positions référencées par deux angles: pour chaque position de $\gamma \in [-\pi, \pi]$, $\beta \in [0, 2\pi]$	87
Figure IV.4. Images d'objets réels projetées dans des orientations particulières	92
Figure IV.5. Réarrangement de la région détectée de l'indice parallèlement au plan focal (gauche) suivie par son changement en taille pour l'adapter à la taille effective de l'objet (droite)	93
Figure IV.6. Exemple d'extraction de la région des objets et déclaration de leur arrière-plan comme région transparente (blanche)	94
Figure IV.7. Deux images de tailles égales dont les tailles des objets réels sont différentes (Figure IV.8)	95
Figure IV.8. Proportions réelles de l'indice et des objets ainsi que les principes de mesures utilisés.....	95
Figure IV.9. Tailles à considérer dans l'image de référence de l'objet pour le calcul des proportions.....	96

Figure IV.10. Les bases géométriques associées au déplacement de l'objet le long de l'axe Oz. La figure du bas est la vue de face suivant la flèche de celle du haut	98
Figure IV.11. Diagramme de l'algorithme général de l'augmentation d'une scène avec les images d'un objet réel et permettant à l'utilisateur de le manipuler. Le symbole \otimes représente le 'et' logique et le \oplus représente le 'ou' logique.	101
Figure IV.12. Bases géométriques associés au déplacement de l'objet le long des axes Ox ou Oy	102
Figure IV.13. Bases géométriques associées au déplacement de l'objet le long de l'axe Oz...	103
Figure IV.14. Les deux systèmes de coordonnées utilisés pour le contrôle de la rotation. L'image de gauche reflète la rotation de CVP qui correspond à la vue de l'image de droite où ORF est orienté avec l'objet	105
Figure IV.15. (De haut en bas et de gauche à droite) Rotation complète d'un objet autour de l'axe Oy de WCS avec un pas $st = \frac{\pi}{6}$ après une première rotation d'un même angle $\frac{\pi}{6}$ autour de l'axe Ox de ORF (première image)	110
Figure IV.16. Axe de rotation utilisé pour la rotation de l'objet de la Figure IV.17	111
Figure IV.17. (De haut en bas et de gauche à droite) Rotation complète d'un objet autour d'un axe simulé présenté à la Figure IV.16 et utilisant un pas $st = \frac{\pi}{6}$	112
Figure V.1. Visualisation d'objets différents pour deux utilisateurs différents.....	118
Figure V.2. Fonctions et relations des modules utilisateurs du prototype avec le module central.....	122
Figure V.3. Architecture globale du prototype du système collaboratif implémenté.....	124
Figure V.4. Relations entre le module central et un module utilisateur	124

Introduction



Introduction

Lorsqu'un enseignant donne un cours en relation avec un phénomène dynamique à ses étudiants, il a toujours tendance à appuyer ses explications par des schémas descriptifs des concepts sous-jacents. Le même besoin est généralement exprimé dans une exposition devant un public pour pouvoir faire passer des idées importantes de la manière la plus aisée. Dans de telles situations et dans beaucoup d'autres vécues dans notre vie quotidienne, des images explicatives font gagner du temps précieux dans l'explication et dans la compréhension. Si ces images peuvent être animées d'une certaine façon, elles offrent d'avantage d'appuis pour les idées communiquées. Mais si elles peuvent être manipulées au sens où leur animation est entièrement contrôlée au moment même de leur utilisation pour qu'elles soient adaptées, selon le besoin, au cours de l'explication, la qualité de l'information communiquée est sans nul doute la meilleure.

Le besoin de compléter notre vision par des informations supplémentaires à ce qui est réellement perçu par notre œil s'avère assez important dans de nombreuses situations. Notre pratique même dans la manipulation, la réparation et l'organisation d'objets connus à priori est précédée, de manière inconsciente, d'une imagination. Devant des situations inconnues, cette pratique permettrait de faciliter grandement une tâche difficile si elle venait à être soutenue par des images qui remplaceraient cette imagination. On serait alors capable de pratiquer des tests virtuels sans coût y afférant. Ils sont simplement l'expression visuelle du virtuel imaginaire. Ils permettent d'amplifier la sémantique de la communication entre humains et donnent une nouvelle dimension à la façon d'accomplir certaines tâches. Dans ce contexte, la finalité de la réalité augmentée est de profiter de l'avancée technologique dans le domaine du multimédia pour faire de l'ordinateur un moyen de communication, non comme une boîte noire, mais en tant que technique mixée aux autres outils naturels.

C'est dans ce contexte d'idées que nous pouvons définir la réalité augmentée en tant que nouveau domaine de recherche. Son apport dans ce sens est d'améliorer la perception humaine par ce qui n'existe pas réellement dans le champ de vue actuel d'un utilisateur donné. On dit que l'environnement de l'utilisateur est augmenté. Des images représentant des objets inexistantes à priori dans cet environnement sont ajoutées à l'image perçue naturellement ou via une caméra. Ainsi, la réalité perçue visuellement est augmentée par des informations visuelles inexistantes en réalité. L'objectif de cet enrichissement par des objets virtuels est de faciliter la compréhension et la maîtrise de cet environnement. Pour que cet objectif puisse être atteint, l'un des problèmes majeurs à résoudre est que cette augmentation paraisse visuellement la plus naturelle possible. Cette contrainte est d'autant respectée que la mixture entre réel et virtuel est faite en respect des relations naturelles connues sur les objets réels. Hors, les objets virtuels sont par définition des images et satisfaire à de telles contraintes requiert une maîtrise totale de leur insertion afin de contrôler leur corrélation avec les objets réels de

l'environnement. Cet aspect naturel n'est pas uniquement lié à l'aspect visuel mais aussi au temps nécessaire au processus d'augmentation pour le traitement associé et à la possibilité d'interaction de l'utilisateur avec la scène augmentée. En termes de contraintes relationnelles entre objets réels et virtuels, l'apparence naturelle devrait être contrôlée dynamiquement pour pouvoir simuler une manipulation. L'utilisateur serait capable d'agir sur les objets virtuels et en qualité d'action c'est ce qui serait qualifiée de manipulation. C'est cette forme de possibilité d'agir sur les images tout en respectant leurs contraintes relationnelles visuelles et temporelles qui permet d'outrepasser la compréhension de l'environnement pour aller au delà de sa maîtrise. L'intérêt de tous ces concepts prend de l'ampleur au moment où l'augmentation d'une scène devient partagée par un ensemble d'utilisateurs. En termes de partage de bénéfice et d'intérêt, cela est évident. Plusieurs utilisateurs profitent simultanément de l'augmentation réalisée. Par contre en termes de communication, c'est une nouvelle dimension qui se définit à l'horizon des résultats de la recherche. C'est l'orientation prise par la branche de la réalité augmentée collaborative.

Le travail présenté dans cette thèse est en majorité intéressé par le problème lié à la nature des objets virtuels dans le domaine de la réalité augmentée. Il discute en détail la possibilité d'utiliser des images d'objets réels afin de contourner leur rendu par ordinateur. Il fait ressortir les avantages d'une telle solution et discute les inconvénients qui en surgissent. Ce travail présente également l'apport fait au domaine relativement à l'ensemble de la communauté de la réalité augmentée collaborative. Il exploite l'importance de la collaboration qui réside dans l'intérêt d'étendre la manipulation ou les actions sur les objets virtuels à un ensemble d'utilisateurs. Bien que la majorité des résultats du domaine soient encore au niveau expérimental, ils sont cependant très prometteurs. Tout comme d'autres domaines, les solutions sont d'abord recherchées à un niveau centralisé ensuite elles sont généralisées pour un partage particulier. C'est la raison qui a fait que ce travail est constitué d'une première idée associée à la nature des objets virtuels et à leur manipulation et d'une deuxième idée qui propose une solution pour les manipuler dans un environnement collaboratif.

Cette orientation est due à l'impact de la nature des objets virtuels sur leur manipulation. Selon cette nature, les concepts sous-jacents du rendu des images sont différents. Ils diffèrent suivant que les objets virtuels sont construits par ordinateur ou représentent des images acquises d'objets réels. C'est particulièrement ce dernier cas qui constitue le cœur de cette étude en raison de l'existence de situations particulières où les objets réels, dont le rendu est réalisé à base d'images, sont mieux adaptés que les objets synthétiques.

A ce propos, malgré les avancées récentes dans la technologie du rendu des objets 3-D, la modélisation d'objets réels ayant une morphologie complexe reste encore une tâche difficile et les images obtenues ne sont généralement pas réalistes. Evidemment, des images rendues d'objets réels ne peuvent en aucun cas égaliser le réalisme de photographies de haute qualité. A base de moyens très sophistiqués de prise d'images

modernes, les photographies reflètent facilement des géométries de textures très complexes et tous les détails de l'éclairage de surfaces sans nécessiter le moindre effort. Rajouté à cela, plus l'environnement ou l'objet virtuels sont complexes et moins naturelle sera la simulation de la navigation ou du mouvement des objets synthétiques. Ceci est dû au fait que toutes les vues des différentes positions et orientations avec les détails associés doivent être calculées à partir de modèles géométriques.

L'impact du rendu 3-D en temps réel est d'affecter le temps global des processus le nécessitant comme une étape de leur tâche sans faire appel au parallélisme. Cet impact se ressent même avec l'utilisation de cartes accélératrices 3-D dans le rendu des images dans certains travaux. Ainsi, des difficultés dans la modélisation et le rendu 3-D restent toujours rencontrées, du moins jusqu'à ce jour, malgré les avancées dans la technologie de l'accélération. Une première difficulté persiste à cause de l'aspect manuel inévitable du processus de conception d'entités 3-D. La deuxième difficulté réside dans les limites de complexité et de rendu généralement imposées par le matériel. La troisième difficulté est due au coût toujours élevé du matériel professionnel, ce qui restreint les solutions qui l'utilisent aux laboratoires. C'est ainsi que l'option de création d'objets 3-D à base de photographies constitue une solution possible et prometteuse pour la minimisation du temps de rendu dans un processus d'augmentation.

Durant les quelques années passées, les techniques de rendu à base d'images (image-based rendering, IBR) ont été investiguées par plusieurs chercheurs comme une alternative au rendu d'images conventionnel à base de modèles. L'objectif des systèmes IBR est d'utiliser les photos comme primitive de base pour la création d'images virtuelles de synthèse. Son avantage particulier est celui de la haute qualité du photoréalisme obtenu et le temps du rendu qui est indépendant de la complexité. Par exemple, le changement du niveau de détail dans de tels systèmes peut être facilement réalisé à travers la résolution multi-niveaux. Le principe repose sur l'idée de prendre au moins une photo de chaque surface importante et visible pour l'utilisateur. La manipulation ou le déplacement se basent alors sur l'affichage successif d'images ayant des relations de continuité.

Dans le contexte des systèmes IBR, ce travail - objet de cette thèse - ajoute une nouvelle brique aux travaux précédents dans le contexte de la manipulation. Au moment où la manipulation est entièrement contrôlée pour les objets synthétiques, quelques difficultés existent encore dans la connexion entre un nombre limité d'images pour créer les orientations infinies pouvant être requises. Ce travail présente une solution générale à ce problème. Cette solution couvre toutes les orientations possibles et naturelles obtenues à partir d'un nombre limité d'images prises pour un objet. Malgré que le principe d'interpolation ne soit pas utilisé, la solution permet d'offrir un sentiment de rotation naturelle de l'objet au moment de sa manipulation.

Pour donner la possibilité de manipuler un objet virtuel, les images doivent être projetées suivant les requêtes utilisateurs. Dans un premier temps, l'objet est photographié en suivant une certaine trajectoire bien définie. La simulation d'un

mouvement consiste simplement à re-parcourir cette même trajectoire dans le même sens ou dans un sens différent et projeter les images correspondantes. Cependant, la simulation d'un objet manipulé suit rarement cette même trajectoire et requiert donc une solution générale. Malheureusement, beaucoup de travaux explicitent les détails des concepts mathématiques relatifs à la navigation dans une scène dans le contexte de l'IBR, alors qu'il n'existe pas beaucoup d'accentuation sur ceux relatifs à la manipulation d'objets malgré leur importance. C'est pourquoi l'accentuation sur ce point particulièrement constitue une partie importante des résultats de notre travail. Ces résultats sont essentiellement l'établissement de relations mathématiques entre les images représentant les différentes vues permettant leur projection correcte ainsi que d'une méthode de leur stockage ayant une structure linéaire.

Sur le plan de l'algorithmique, le processus d'augmentation est assez complexe. Par conséquent, le temps processeur inhérent requis n'est pas négligeable et influe négativement sur l'augmentation d'une scène en temps réel. Plusieurs étapes de ce processus sont assez complexes dans le calcul associé et rechercher des solutions optimales ou optimiser celles existantes par certaines heuristiques serait très bénéfique. Entre autres, la partie spécifique au rendu devrait être minimisée par la projection d'images simples. C'est la raison qui fait que cette solution peut procurer une nette amélioration aux systèmes de réalité augmentée.

Basée sur les principes précédemment cités, la thèse montre comment il est possible de simuler la manipulation d'un objet réel à base de ses images et de partager cette manipulation dans un environnement collaboratif. Dans un premier temps, les concepts de base ainsi que certains travaux associés sont présentés afin de situer le travail dans le contexte général de la réalité augmentée. Par la suite, un détail de la problématique et des solutions associées à ce travail sont présentés. La présentation fait un parcours depuis la prise d'images des objets réels, leur stockage dans une structure linéaire particulière, leur accès pour les projeter, les bases mathématiques en relation avec le sens de la projection et enfin la simulation de leur manipulation. Nous montrons comment cette manipulation est décomposée en une rotation et une translation et comment est traitée chacune d'elles. Finalement, une projection de tous ces concepts dans le domaine du collaboratif est faite. Cette structure des idées a résulté sur une logique de partage de la thèse en cinq chapitres.

Dans le premier chapitre, un tour d'horizon est fait sur les principaux travaux ayant obtenus un impact sur le cours de la réalité augmentée collaborative. Ceci est important dans un but de définition du domaine. A travers cette présentation, les caractéristiques du domaine ou plus particulièrement les éléments de base de la réalité augmentée sont discutés. Ils sont énumérés dans un but de faire ressortir les axes de recherche les plus importants. Tout au long du chapitre, un positionnement de notre travail relativement au domaine est fait chaque fois que ceci est nécessaire.

Le principe d'une augmentation 3-D est présenté au second chapitre. Les fondements mathématiques associés ainsi que les concepts de vision relatifs particulièrement aux

besoins du travail présenté dans cette thèse sont détaillés. Le chapitre cerne la théorie de base du principe d'augmentation d'une scène depuis le film généré par la caméra jusqu'à l'insertion des objets virtuels. Particulièrement, la calibration de la caméra à base du principe de correspondance est détaillée. Cependant, il ne traite pas toutes les théories associées à la calibration. Il ne touche qu'à celles ayant une relation avec ce travail.

Le troisième chapitre est le cœur de la théorie de tout ce travail. Il fixe les assises de la manipulation des objets virtuels dans une séquence augmentée. Il se base sur les concepts présentés au second chapitre pour montrer comment il est possible d'agir sur les objets virtuels. Il discute les fondements mathématiques de la manipulation et particulièrement les outils utilisés.

Le quatrième chapitre synthétise tous les principes précédents pour les appliquer aux objets réels virtualisés à base de leurs images. Il donne les détails des résultats de recherche obtenus et les contraintes relatives. Les concepts essentiels de géométrie et les expressions analytiques associées sont détaillés. En particulier, les bases mathématiques qui lient les images aux orientations possibles résultant de manipulations sont démontrées. En ce sens, il démontre comment est ce que cette manipulation est implémentée. Ce chapitre peut être considéré comme une synthèse de la partie la plus dominante des résultats de nos recherches.

Le cinquième et dernier chapitre discute l'extension de la manipulation des objets réels virtualisés à un ensemble d'utilisateurs. Il présente une architecture simple permettant une collaboration distante des utilisateurs via un système vidéo. Il revoit les systèmes déjà existants dans le domaine et situ notre architecture relativement à ceux-ci.

La thèse est close par une conclusion où nous résumons les conclusions des différents chapitres et indiquons quelques perspectives pour la recherche sur le sujet traité. Des détails et des preuves complétant le contenu de certains chapitres sont donnés en annexe.

The page features a decorative graphic consisting of three blue circles of varying sizes, each with a lighter blue ring around its center. These circles are arranged vertically, with the largest at the top, a medium one in the middle, and a large one at the bottom right. Two thin blue lines intersect at the top left, forming a V-shape that frames the circles.

Chapitre I

Réalité Augmentée Collaborative

Le chapitre définit le domaine de la réalité augmentée collaborative et les applications associées ainsi que le principe de la collaboration

Réalité Augmentée Collaborative

1. Introduction

L'un des objectifs récents de la conception dans le domaine de l'interaction homme-machine est d'étendre les capacités des ordinateurs pour combiner le réel et le virtuel afin d'aider les utilisateurs dans leur interaction avec l'environnement physique. Cette tendance a donné naissance aux systèmes de la réalité augmentée. L'objectif de ces systèmes est de briser la frontière existante entre le monde réel, incluant l'utilisateur et son environnement, et le monde informatique. Un système de réalité augmentée offre ainsi la possibilité de réaliser une tâche virtuelle ou réelle appartenant respectivement au monde informatique ou réel.

L'idée de la réalité augmentée est née avec les travaux de [129] qui ont permis de réaliser le premier système [128] basé sur un casque suivi par un capteur de mouvement. Avec ce dispositif, l'utilisateur peut visualiser et naviguer autour d'éléments virtuels positionnés dans un espace réel. Durant les années 80, le concept a été utilisé surtout dans un cadre militaire pour l'affichage d'informations virtuelles sur les visières des casques des pilotes d'avions. Le coup de pouce le plus important a été celui de [70] au début des années 90 pour donner au médecin la possibilité de visualiser directement des données à ultrasons sous forme d'images sur le corps du patient.

La réalité augmentée est généralement considérée comme un prolongement de la réalité virtuelle. En effet, après l'essor qu'a connu la réalité virtuelle dans de nombreux domaines, la réalité augmentée propose l'extension de ces résultats au monde réel. En ce sens, son objectif est de fournir à l'utilisateur un environnement qui est une mixture de la réalité avec des entités virtuelles. Cette mixture, dite augmentation, consiste à compléter une scène d'un environnement réel par l'adjonction d'objets synthétiques ou réels, communément appelés objets virtuels. Ces derniers sont manipulés, dans certains contextes, à travers des outils interfaces. La collaboration est le concept qui sous entend une manipulation des objets virtuels dans un environnement multiutilisateurs.

Ainsi, la réalité augmentée effectue une incrustation du virtuel dans le réel, ou à l'inverse une intégration du réel dans le virtuel. En ce sens, la réalité virtuelle n'est pas une sous partie de la réalité augmentée. Dans sa présentation de cette relation, Milgram [93] la présente comme étant un continuum entre deux environnements, réel et virtuel. En réalité virtuelle, l'utilisateur est complètement plongé dans un monde virtuel et est coupé du monde réel. Par contre, en réalité augmentée, l'utilisateur est maintenu au contact de son environnement réel.

Plus précisément, les systèmes de réalité augmentée se distinguent par le fait que l'objet de la tâche réside soit dans le monde réel, soit qu'il porte sur des objets virtuels faisant ainsi partie du monde informatique. Ces deux systèmes sont respectivement dits

"Réalité Augmentée", comme les systèmes de chirurgie assistée par ordinateur [85], et "Virtualité Augmentée", comme l'utilisation de cubes réels pour modifier des objets informatiques tels que des fichiers [19]. Dans [92], un continuum allant du tout réel au tout virtuel en passant par la réalité augmentée, la virtualité augmentée puis la réalité virtuelle est défini pour classer les dispositifs d'affichage de réalité augmentée.

Afin de bien définir le domaine et comprendre l'apport de la réalité augmentée, ce chapitre présente un ensemble d'applications développées essentiellement dans le domaine du collaboratif. A travers cette présentation, nous apportons une assise justificative des caractéristiques et les axes de recherche essentiels du domaine qui sont énumérés avant et que nous avons fait ressortir suite à notre synthèse. Citons que tout au long de la présentation des différents points du chapitre, certains aspects relatifs à notre travail sont situés relativement aux autres travaux. En guise de conclusion, nous expliquons la problématique et les objectifs globaux de notre travail.

2. Définition de la réalité augmentée collaborative

Définir la réalité augmentée n'est pas une tâche évidente en raison de la pluralité des aspects associés. La difficulté résulte essentiellement du fait qu'elle est généralement définie à travers ses objectifs, qui ne sont pas tous atteints. Les applications fonctionnelles ou réalisées peuvent servir comme base pour la définir de manière étendue mais un consensus reste difficile à établir. Azuma [96] a défini la réalité augmentée comme étant un système qui combine des images réelles et virtuelles, respectant un alignement 3-D, de manière interactive et en temps réel.

D'une manière générale, l'augmentation est l'adjonction d'entités visuelles au film obtenu de la séquence. L'augmentation d'une scène consiste en l'intégration ou l'incrustation en temps réel d'objets virtuels dans le film relatif sans dégradation de son homogénéité ou de son réalisme (Figure I.1). A priori, les objets virtuels ne sont pas directement perceptibles par l'œil humain et ils n'ont aucune existence propre dans l'environnement réel actuellement filmé. Cependant, ils peuvent être manipulés virtuellement par un utilisateur au même titre que les objets réels.



Figure I.1. Insertion d'un objet virtuel: à gauche l'image originale et à droite l'image augmentée

La réalité augmentée collaborative est l'extension de la réalité augmentée au fait que la manipulation des objets virtuels peut se faire simultanément par plusieurs utilisateurs. Les objets virtuels sont alors partagés et servent comme un nouveau moyen de communication. Ainsi, l'augmentation de la scène pour un utilisateur n'est plus restreinte à ses propres actions mais elle est affectée par les actions des autres utilisateurs. Trois facteurs ont alors un impact sur la scène augmentée : les caractéristiques de la scène elle-même, les actions de l'utilisateur cible et celles des autres avec lesquels ce dernier collabore.

Dans le contexte de la manipulation et de la collaboration, nous distinguons essentiellement deux types d'augmentation. La première qui est l'exécution augmentée et qui consiste à permettre à l'utilisateur d'effectuer des tâches dans le monde réel d'une façon nouvelle. Un exemple consiste à porter un active-badge [103] qui permet d'ouvrir une porte sans la toucher. L'autre type d'augmentation est la perception augmentée, qui est la plus communément mise en œuvre, où des informations pertinentes pour la tâche sont ajoutées à l'environnement de l'utilisateur. Le musée augmenté [60] en est un exemple.

Il existe des variantes associées à la réalité augmentée qui spécialisent le domaine selon les applications réalisées. Citons à titre d'exemple le cas où la tâche à réaliser représente un objectif commun à l'ensemble des utilisateurs. La collaboration devient alors une coopération où chacun réalise une partie. Une autre variante est lorsque le système est concerné par la position des utilisateurs. On parle souvent de réalité augmentée mobile. Ainsi, un tel système est celui où l'augmentation est liée aux informations de la position de l'utilisateur et de son environnement [94]. Le terme de réalité augmentée est également utilisé pour désigner la combinaison d'images réelles et virtuelles en 3-D, mais sans considération de la contrainte temps réel [5]. Dans ce cas, on parle de postproduction, c'est-à-dire que l'insertion des objets virtuels se fait généralement dans une étape postérieure à l'acquisition de la séquence vidéo. Cela conduit généralement à un résultat plus précis et plus réaliste. C'est la technique utilisée essentiellement en cinéma dans la réalisation des scènes de fiction.

3. Caractéristiques de la réalité augmentée collaborative

Les caractéristiques citées dans cette section ne sont pas exclusivement celles de la réalité augmentée collaborative. D'une manière générale, ce sont celles de la réalité augmentée dans son aspect général telle que définie précédemment et donc de la réalité augmentée collaborative puisque cette dernière est plus générale. Nous les avons inclus sous la coupole du collaboratif en raison de la spécificité du sujet.

3.1. Objets virtuels

Les objets virtuels sont la caractéristique la plus importante du domaine. Ils peuvent être de natures différentes. Ils sont soit synthétiques, construits par ordinateur, soit réels, préenregistrés dans d'autres séquences ou images pour être directement projetés. Nous

détaillons par la suite ces deux types mais nous pouvons d'ores-et-déjà les différencier. Les objets synthétiques sont des entités purement informatiques, même s'ils sont une reproduction de modèles d'objets existants. Les objets réels en tant que virtuels sont des images reflétant des objets réels photographiés. Ainsi, dans le premier cas, tous les détails graphiques sont réalisés par ordinateur, alors que dans le second, ce sont des images résultat de photographies ou de vidéos fournies par du matériel spécialisé.

Lorsqu'ils sont synthétiques, les objets virtuels peuvent varier de la simple modification de la scène réelle, où il n'existe aucun objet ajouté au sens d'entité mais juste une modification de l'apparence d'objets réels existants (Figure I.2), jusqu'à l'adjonction d'images d'objets préalablement inexistantes dans la scène (Figure I.1). Dans l'un ou l'autre des cas, un seul flux d'images - la scène filmée - est analysé et traité afin de procéder à l'augmentation.



Figure I.2. Objet virtuel synthétique qui consiste en la modification de l'apparence de la scène

Lorsque les objets virtuels sont introduits directement en tant qu'images d'objets réels, l'augmentation consiste en une mixture de deux entités réelles: la séquence et les objets (Figure I.3). Le système informatique doit alors traiter deux flux d'images.



Figure I.3. Objet virtuel intégré en tant qu'image d'un objet réel

3.2. Systèmes d'augmentation

Actuellement, deux systèmes sont utilisés pour la réalisation des augmentations: optique et vidéo. Le système optique est à base de verres; le système vidéo est à base d'écrans.

La perception des objets virtuels se fait généralement par l'intermédiaire d'un HMD (Head Mounted Display, littéralement « écran monté sur la tête ») ou simplement via un écran ordinaire. Notons qu'un système vidéo peut ne pas être à base d'un HMD. Mais dans l'un ou l'autre des systèmes, l'utilisateur perçoit les objets virtuels au même titre que l'environnement réel filmé dans lequel il évolue. Pour ce faire, ces objets sont incrustés dans la scène filmée puis l'ensemble est projeté sur écran, à l'exception du système optique qui ne projette que les objets virtuels.

3.2.1. Système optique

Les HMD optiques sont essentiellement à base d'un verre qui est partiellement transparent, c'est-à-dire qu'il est traversé par la lumière du monde réel permettant au porteur de percevoir son environnement et partiellement réfléchissant, permettant ainsi aux images virtuelles d'être projetées au dessus (Figure I.4). Ce système permet alors de cumuler ces images à la vue du monde réel. Le système informatique, ou plus précisément le générateur de scène, devrait aligner correctement les objets virtuels à ceux du monde réel afin que l'utilisateur les perçoive aux endroits escomptés. Aussi le monde réel est perçu directement et non traité comme un flux, à l'opposé des objets virtuels projetés qui constituent un flux vidéo. La Figure I.5 présente le schéma conceptuel d'un tel système.



Figure I.4. HMD optique

Ce système présente principalement deux inconvénients majeurs difficiles à surmonter. Le premier inconvénient est relatif aux objets virtuels projetés qui sont transparents à cause de la nature même du système. Cet aspect va à l'encontre du réalisme à atteindre dans une augmentation puisque les objets virtuels sont facilement discernables des objets réels. L'adaptation de cette transparence reste cependant restreinte à un certain nombre limité d'applications. Le deuxième inconvénient réside dans le décalage des objets virtuels par rapport à leurs endroits prévus. En effet, ils paraissent autant décalés que la vitesse du système de traitement est relativement réduite par rapport à la vitesse de déplacement du HMD et donc de la tête de l'utilisateur.

Par contre, un avantage important d'un tel système est le traitement qui est limité aux objets virtuels. Cet aspect offre à ce système une simplicité dans l'implémentation. Le flux réel n'est utilisé que pour les calculs de recalage mais pas pour la génération d'une nouvelle séquence. C'est alors que la vue du monde réel n'est pas dégradée par la numérisation qui engendre, en plus de son traitement, des temps de réponse faibles pour

les systèmes vidéo. D’ailleurs, en cas de coupure de courant, le HMD joue le rôle d’une simple paire de lunettes. A l’inverse, même la vue du monde réel sera supprimée pour le cas de la solution vidéo.

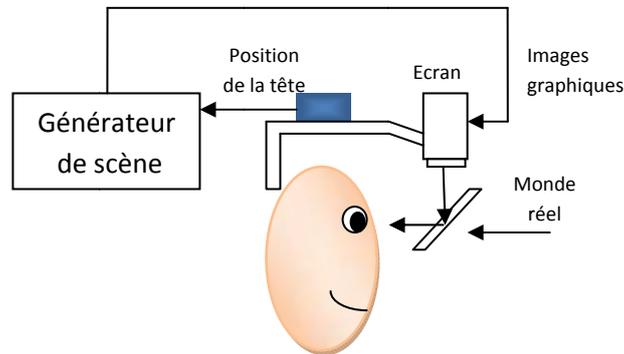


Figure I.5. Schéma fonctionnel du HMD optique

3.2.2. Système vidéo

Le système vidéo est constitué d’un casque opaque qui occlut totalement les yeux et ne permet pas de percevoir directement le monde réel (Figure I.6). A l’intérieur de ce casque existe un petit écran qui joue le rôle de fenêtre sur l’environnement. La scène réelle est filmée par une ou deux caméras fixées sur le HMD et le film généré est projeté, après avoir été combiné aux images virtuelles, sur l’écran du HMD. Ce système traite donc les deux sources d’images, en l’occurrence réelles et virtuelles, avant de les projeter. Le schéma fonctionnel relatif à ce système est représenté à la Figure I.7.



Figure I.6. HMD vidéo

Ce système offre une maîtrise d’augmentation beaucoup plus importante pour la composition que le système optique. Il est particulièrement flexible dans la composition pour l’occlusion entre objets réels et virtuels, puisque les deux flux sont totalement contrôlés [97]. En effet, le système optique ne permet pas une occlusion complète à cause de la lumière provenant du monde réel. Elle est difficile, voire impossible, parfois à éliminer. Elle cause même des problèmes de contraste qui correspondent au risque de suppression des images virtuelles, particulièrement si le monde réel est trop éclairé. Par ailleurs, le désavantage du retard de numérisation est exploité au profit de celui de la génération et de l’intégration des images virtuelles. Ce retard ne peut être compensé par un système optique puisque le monde réel est perçu en direct.

Le système vidéo peut être étendu aux écrans d'affichage ordinaires pour certaines applications utilisant ce système. La différence essentielle réside dans la mobilité de l'utilisateur qui devient restreinte; ce qui offre plus de réalisme pour le HMD vidéo. Le système à écrans ordinaires, par contre, est plus souple puisqu'il ne nécessite aucune familiarisation de la part de l'utilisateur. C'est d'ailleurs le système pour lequel nous avons opté dans notre travail présenté dans cette thèse. Conceptuellement, c'est le schéma de la Figure I.8.

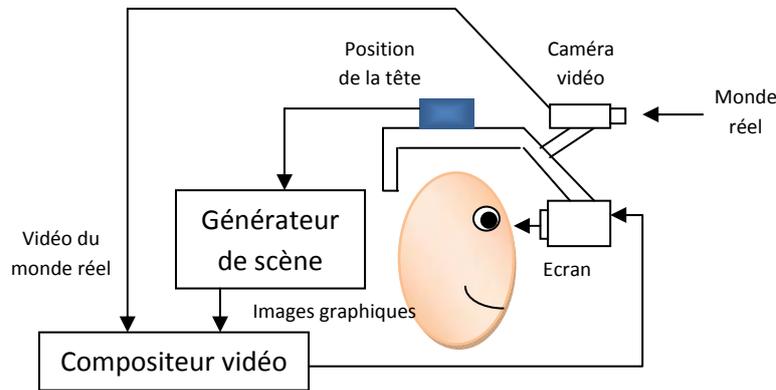


Figure I.7. Schéma fonctionnel du HMD vidéo

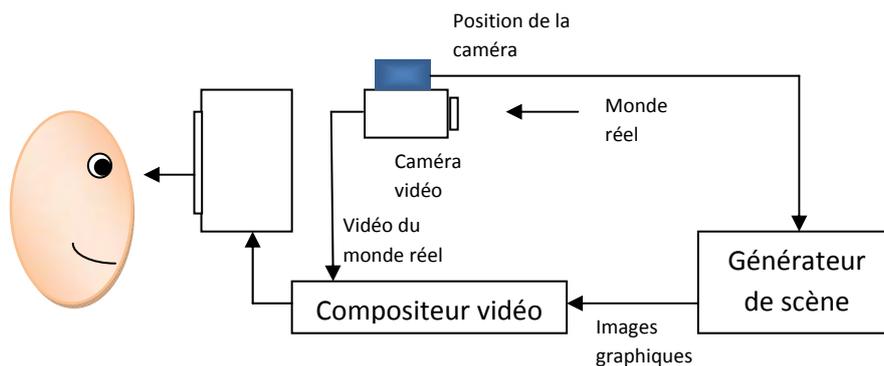


Figure I.8. Schéma fonctionnel d'une augmentation à base d'écrans ordinaires

3.3. Mixture visuelle de réel et virtuel

La mixture visuelle entre réel et virtuel est une caractéristique spécifique de la réalité augmentée. Cette opération n'est devenue possible que suite à la puissance croissante des ordinateurs. Elle nécessite une analyse et un traitement profonds de la scène réelle filmée et des objets virtuels intégrés. De plus, la tendance de la réalité augmentée est d'offrir une composition naturelle. Cela insinue que le sentiment d'effets artificiels doit être minimisé. Afin d'atteindre cet objectif, l'incrustation doit refléter un photoréalisme dans la scène générée respectant les lois de la physique. Ainsi, l'incrustation devrait respecter les principes d'occlusion, de zoom, d'éclairage, d'ombre, etc. Toutefois, cet objectif impose, généralement, un ensemble de contraintes à respecter afin de maintenir le réalisme de la scène comme nous le verrons tout au long de cette thèse plus ou moins en détail selon le besoin. Le réalisme visuel est d'ailleurs l'un des grands axes de la recherche dans le domaine (Figure I.9).



Figure I.9. Augmentation avec une composition naturelle où il est difficile d'identifier les objets réels de ceux virtuels

3.4. Interaction avec le virtuel

Les axes de recherche scrutent une composition interactive en temps réel. Ainsi, l'interaction est une des caractéristiques les plus importantes du domaine de la réalité augmentée. D'ailleurs, tout le domaine du collaboratif en est basé. L'interaction est le fait que l'utilisateur puisse interagir avec les objets virtuels à travers des outils de maniement comme s'ils existaient réellement (Figure I.10). Notons que c'est le système d'augmentation qui devrait permettre cette manipulation. En offrant cette possibilité, l'utilisateur serait en mesure de manipuler les objets virtuels au même titre que les objets réels. Il serait alors capable de les créer, de les faire déplacer, de les tourner, et même de les détruire, tout en prenant en considération le recalage correct des objets réels et virtuels. Une partie importante de notre travail rentre principalement dans ce cadre et traite, d'un point de vue particulier, les différents aspects visuels d'un objet virtuel qui résultent de sa manipulation.



Figure I.10. Jeux d'échec virtuel: l'interaction est faite à base d'objets réels

3.5. Collaboration des utilisateurs

Le besoin de collaborer en réalité augmentée est une extension naturelle inhérente au principe de la manipulation, exactement comme la majorité des tâches humaines nécessitant une participation collaborative. Ainsi, la collaboration dans un

environnement de réalité augmentée consiste à permettre à plusieurs utilisateurs d'interagir à travers et - simultanément - avec des objets virtuels partagés. La collaboration stipule que les différents utilisateurs peuvent communiquer naturellement tout en renforçant cette communication par le virtuel supporté. Leur collaboration se fait selon leurs habitudes naturelles sauf qu'elle est enrichie par les objets virtuels (Figure I.11). Contrairement à la réalité virtuelle, l'interaction et la communication naturelles des utilisateurs sont maintenues afin de ne pas les isoler de leur milieu naturel.

Malgré que les interfaces des applications de la réalité augmentée soient prometteuses, le développement d'un nouveau type spécifique pour le collaboratif s'avère important et même d'un grand intérêt. Ceci peut servir dans l'amélioration des collaborations face à face ou distantes qui sont difficiles à réaliser par les moyens technologiques traditionnels. D'ailleurs, l'un des objectifs de tels environnements est de promouvoir l'interaction sociale entre les utilisateurs d'un même espace physique par le collaboratif virtuel.

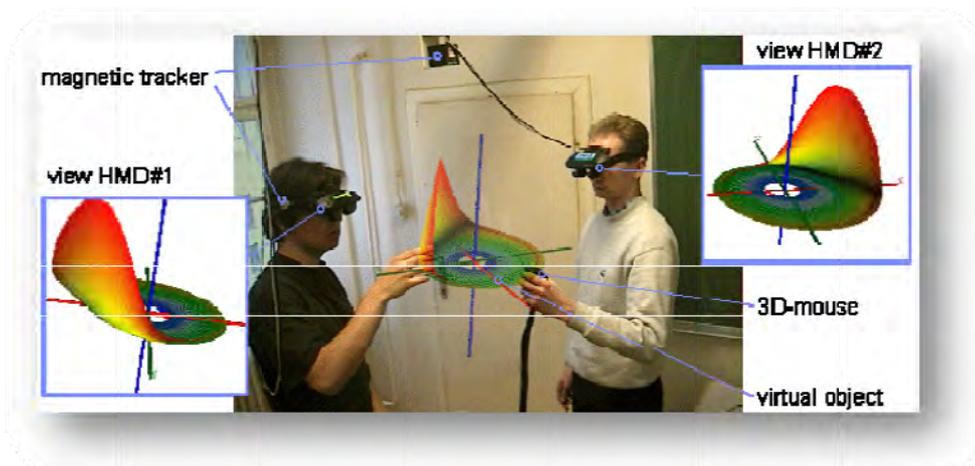


Figure I.11. Collaboration de deux utilisateurs dans la manipulation d'un même objet virtuel

3.6. Pluralité du domaine

La réalité augmentée est une approche multidisciplinaire qui exploite les résultats de plusieurs domaines. Particulièrement, ceux de la vision artificielle, l'infographie, les mathématiques (particulièrement la géométrie dans l'espace) et l'optique constituent ses piliers de base. Les résultats de la vision artificielle sont utilisés principalement pour le calcul du point de vue des caméras. Les augmentations ayant une touche naturelle doivent obligatoirement faire appel à l'infographie puisque la composition de scènes naturelles avec des objets virtuels revient à manipuler les images à leur bas niveau, les pixels. La géométrie dans l'espace est le support de base de tous les aspects 3-D. Enfin, l'optique est inhérente aux caméras qui sont l'outil de base de la réalité augmentée.

3.7. Réalité augmentée: outil pour d'autres applications

La réalité augmentée fournit une assise importante pour certaines applications afin de simplifier et/ou améliorer l'accomplissement de tâches réelles requérant un aspect visuel

supplémentaire. Dans une multitude de cas, elle ne constitue pas un objectif en soit mais un outil utilisé par d'autres applications. Toutefois, il existe des applications où l'objet de la tâche est virtuel et est inhérent aux objets virtuels [38]. Ainsi, l'augmentation de scènes est généralement utilisée pour améliorer la perception du monde réel afin de rendre plus performantes des tâches visuelles. Nous présentons ultérieurement dans ce chapitre les différentes applications faisant appel à la réalité augmentée.

4. Principales orientations de la recherche en réalité augmentée

Afin de pouvoir présenter clairement la problématique de notre travail et son objectif dans la suite de cette thèse, cette section présente brièvement les grands axes de recherche dans le domaine de la réalité augmentée tout en précisant les orientations liées à notre travail. Certains axes sont détaillés dans la suite de cette thèse suivant l'intérêt qu'ils présentent par rapport aux objectifs du travail.

Les orientations de la recherche vont dans le sens de surpasser certaines difficultés rencontrées durant le processus d'augmentation. Il existe celles de nature algorithmique et celles relatives à la sémantique de la scène. La première catégorie se résume dans l'alignement correct des objets virtuels incrustés qui doivent s'agencer avec la scène en taille et en position. La deuxième catégorie est relative à l'aspect de réalisme de la scène. Elle est relative aux problèmes d'éclairage, d'occlusion, d'ombre, etc. où chaque cas est un axe de recherche en soit. Afin de les cerner de manière synthétique, nous avons procédé à un groupage de ces orientations dans des classes.

4.1. Réalisme visuel

L'aspect visuel est une composante centrale en réalité augmentée. C'est l'aspect qui a trait à l'apparence des objets virtuels. En effet, les objets virtuels sont purement visuels et ne sont pas directement perceptibles par l'œil humain; ils n'ont d'existence qu'à l'intérieur du système informatique et ne sont perçus que via un écran. Leur apparence devrait refléter des objets réels non discernables des objets effectivement réels. Ce sont des informations visuelles supplémentaires introduites dans des séquences réelles afin d'atteindre des objectifs fixés à priori.

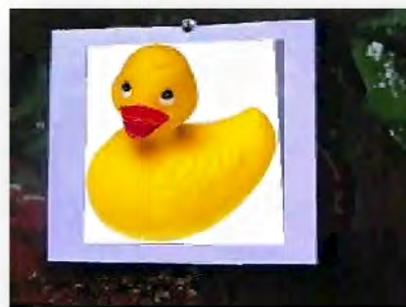


Figure I.12. Exemple d'image où les couleurs de l'objet virtuel et ceux de la scène ne sont pas équilibrés

Le réalisme visuel impose la prise en compte de deux aspects: la qualité des images et la disposition des objets virtuels. Le premier est une tendance d'augmenter les scènes tout en gardant leur touche naturelle telle qu'elle a été générée par les caméras. C'est un défi majeur des recherches en réalité augmentée. Le fait de ne pas dégrader le réalisme de la scène par l'introduction de nouvelles entités nécessite de prendre en considération un certain nombre de facteurs. L'éclairage, l'ombre, l'occlusion entre les objets réels et virtuels, et bien d'autres paramètres visuels doivent être pris en compte afin de ne pas introduire une apparence artificielle (Figure I.12). Cette dernière peut être ressentie suite à une dégradation apparente des images.

Le deuxième aspect qui est relatif à la disposition des objets virtuels est également important. En effet, l'apparence artificielle de la scène peut également être ressentie en cas où les objets virtuels ne respectent pas les lois de la nature. La plus évidente de ces lois est relative à la pose de ces derniers sur une surface qui les supporte afin de ne pas flotter visuellement dans l'air (Figure I.13). C'est un problème majeur relatif à l'alignement correct des objets virtuels – dans le temps et l'espace - relativement aux emplacements cibles de la scène. C'est ce qui est communément appelé "Registration Problem". Les décalages éventuels conduisent à des augmentations non réalistes pouvant avoir un impact sur la finalité de la tâche à réaliser.

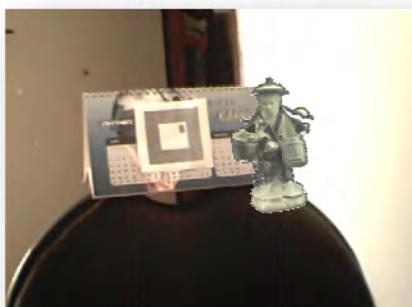


Figure I.13. Un objet virtuel suspendu dans l'air

Le problème d'alignement est cité par Azuma [96] comme étant un facteur limitatif des applications de la réalité augmentée qui nécessitent des alignements précis. Il indique que de nombreuses applications ne peuvent être acceptées sans un alignement correct.

4.2. Calibration de la caméra

La caméra est l'un des outils clés du domaine de la réalité augmentée. Pour pouvoir être augmentée, toute scène nécessite d'être filmée. Son augmentation peut être en temps réel ou en différée. L'augmentation en temps réel se fait directement sur le film au fur et à mesure de son acquisition. Cette manière de faire nécessite une puissance non négligeable du système informatique chargé de l'augmentation. A titre d'exemple, pour une caméra générant un flux vidéo de 25 images par seconde, le processus d'augmentation doit traiter chaque image en moins de 1/25 seconde afin d'avoir une

augmentation sans retard. Dans le cas contraire, des images seront perdues et non augmentées.

Afin de pouvoir réaliser des augmentations, il est fondamental de déterminer les paramètres de la caméra dits intrinsèques et extrinsèques, également appelés respectivement statiques et dynamiques. C'est ce qui constitue sa calibration. Les paramètres intrinsèques - ou internes - expriment les caractéristiques techniques internes de la caméra. C'est la partie optique de l'augmentation. Les paramètres extrinsèques ou externes définissent la correspondance entre le monde réel et les images générées. Mathématiquement, ils représentent les positions relatives ou correspondance de la scène filmée et de la caméra. Mieux est cette correspondance, meilleure sera l'augmentation obtenue et tendra vers une apparence réaliste de la scène augmentée. C'est la partie qui fait appel à la géométrie dans l'espace.

Plus formellement, calibrer une caméra consiste à déterminer de manière analytique la fonction qui associe à un point de l'espace tridimensionnel sa projection dans l'image générée par celle-ci. Il s'agit d'un problème de la vision artificielle utilisant la géométrie dans l'espace pour analyser la scène observée à travers une ou plusieurs caméras. La recherche de la solution revient à chercher la position dans l'espace des points dont on connaît l'image par la projection sur l'écran de la caméra. En général, cette fonction de projection appartient à une famille de fonctions liées au modèle de caméra choisi. Le modèle de caméra le plus couramment utilisé est le modèle perspectif linéaire, appelé aussi modèle projectif ou sténopé que nous détaillons au troisième chapitre.

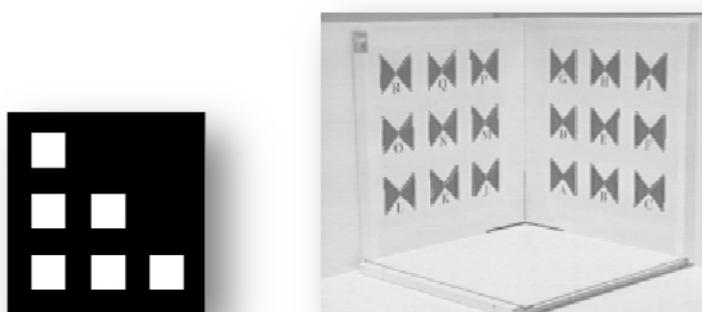


Figure I.14. Indices de calibration de la caméra: à gauche un indice planaire et à droite une mire 3-D

Pour augmenter une scène, il est indispensable qu'elle contienne des indices (Figure I.14). Ils peuvent exister par défaut comme étant des caractéristiques de la scène, ou bien explicitement insérés. Dans ce dernier cas, la terminologie habituellement utilisée est celle de mire. Ces indices servent comme repères de calcul et parfois d'insertion des objets virtuels. Ils constituent le point de départ de la calibration. Le principe réside dans l'observation d'objets 2-D ou 3-D de coordonnées connues et ayant des formes particulières qui servent à la calibration de la caméra. Ce sont généralement des points répartis sur un, deux ou trois plans particuliers (Figure I.15). Une fois ces points identifiés, le calcul peut être effectué de façon relativement simple.

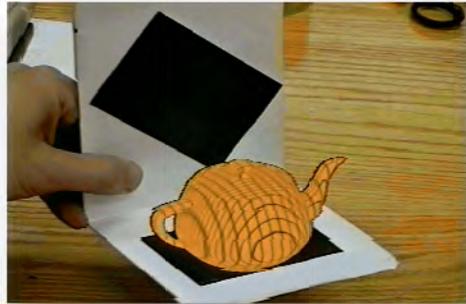


Figure I.15. Indice de calibration composé de deux indices planaires ayant une forme 3-D permettant d'augmenter l'image

Certains travaux utilisent d'autres méthodes pour la calibration de la caméra dans les environnements non préparés. Citons à titre d'exemple celui de [6] qui présente une auto-calibration de la caméra à base de compas électronique. Un autre cas est celui d'un sujet de recherche au sein de notre équipe traitant du temps réel et des environnements non préparés.

Notons que le réalisme d'une augmentation est influencé par la précision des paramètres statiques et dynamiques estimés par la calibration de la caméra. Cette estimation est généralement entachée d'erreurs inhérentes aux outils utilisés pendant la prise d'images et nécessite parfois des corrections. Les déformations optiques, qui induisent des images distordues, sont un exemple de paramètres statiques à corriger. L'analyse directe des images générées peut fournir des résultats erronés. Par analogie, les paramètres relatifs à la mobilité des objets du monde réel et de la caméra peuvent être erronés pour des applications requérant une grande précision. De telles imprécisions peuvent engendrer des retards de génération visualisant les objets virtuels à des emplacements décalés dans le temps ainsi que des défauts d'alignement. C'est la raison pour laquelle beaucoup de travaux sont spécifiquement orientés vers la calibration de la caméra et ce indépendamment de tout domaine. Pour notre cas, nous puisons essentiellement des résultats des travaux relatifs à l'auto-calibration que nous décrivons au deuxième chapitre.

4.3. Manipulation et interaction

Généralement, l'augmentation n'est pas restreinte à une simple vue combinée de réel et de virtuel. Les axes de recherche examinent également une augmentation interactive. En ce sens, la manipulation est le fait de donner à un utilisateur la possibilité d'interagir avec les objets virtuels via des outils où le couple manipulation-outils doit être adapté au contexte. En ce sens, des outils de maniement sont nécessaires variant en fonction des applications et des besoins. La manipulation de ces objets est alors conditionnée par des gestes et des actions que le système doit interpréter. On cite, à titre d'exemple, les stylos attachés à des capteurs magnétiques (Figure I.10).



Figure I.16. Manipulation d'objets virtuels et interaction avec la scène augmentée

L'interaction est exprimée par le fait que l'évolution des utilisateurs ainsi que leurs actions sont conséquentes des résultats visuels. En d'autres termes, leurs actions futures sont des réactions aux évolutions des objets virtuels. Un exemple est présenté dans [27] où les utilisateurs sont capables de manipuler des bâtiments virtuels et de créer de nouveaux modèles 3-D (Figure I.16).

L'objectif de cette interaction peut être réel ou virtuel. Le cas de la manipulation ayant un objectif réel est assez restreint dans les applications actuelles de la réalité augmentée. Celles réalisées jusqu'à nos jours ne sont pas encore assez matures au point d'avoir des effets sur des objets réels. C'est plutôt l'interaction ayant un objectif virtuel qui est la plus fréquente. Elle consiste généralement à simuler des actions réelles par celles virtuelles, ou plus exactement, à manipuler un objet virtuel à la place d'un objet réel et étudier les effets et les conséquences que peuvent avoir des déplacements réels de l'objet. Ce genre de technique est assez important notamment pour un robot télécommandé à distance et agissant dans un milieu dangereux ou d'accessibilité limitée à un être humain (Figure I.17). Il peut alors être important de simuler visuellement les actions d'un bras afin d'étudier et enregistrer la trajectoire adéquate à suivre réellement.

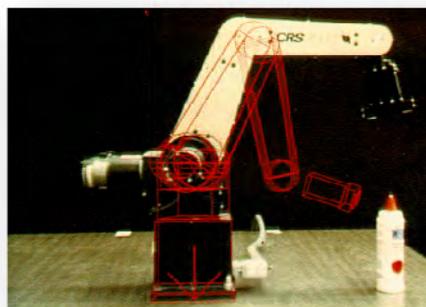


Figure I.17. Simulation du déplacement d'un robot avant son déplacement réel

D'une manière générale, les recherches dans le domaine de l'interaction et de la manipulation touchent à deux points essentiels. Le premier touche à l'aspect visuel. Il concerne les différentes apparences qu'il faudrait donner aux objets virtuels suite aux actions des utilisateurs de sorte qu'elles reflètent exactement l'objectif escompté. Le

second est relatif à l'étude des outils à utiliser, leur nature et la façon d'interpréter leur mouvement. En fonction de la nature de chaque type d'outil, il faudrait généralement trouver une interprétation de sa manipulation relativement au mouvement qu'il faudrait faire appliquer à l'objet virtuel. Sur ce plan également, nous utilisons dans notre travail les résultats développés pour les outils 2-D dont nous justifions l'utilisation au quatrième chapitre.

4.4. Augmentation temps réel

Par définition, une augmentation doit se faire en temps réel. Au sens général du principe, cela revient à augmenter une séquence sans perte et en respect de toutes les contraintes relatives au réalisme visuel. Cependant, cela reste un objectif difficile à atteindre sauf pour des applications spécifiques. C'est un problème en majorité dû au temps de latence qui est le retard qui définit le temps pris par le système entre la mesure d'un capteur et le retour visuel pour l'utilisateur. C'est la raison de l'orientation de certains travaux spécifiquement pour la recherche de solutions minimisant le temps d'augmentation pour parvenir au temps réel. Le notre constitue dans ce sens un apport au domaine d'un point de vue de la réduction du temps global de l'augmentation par rapport à la partie de rendu des images des objets virtuels. Un détail concernant ce point est donné au quatrième chapitre.

Typiquement, la contrainte du temps influe sur l'apparence de l'évolution des objets virtuels. Elle signifie que leur mouvement doit être adapté à celui des objets réels afin de paraître naturel. Pour cela, toutes les images de la séquence doivent obligatoirement être traitées sans perte. C'est pourquoi, la simulation d'une évolution naturelle des objets virtuels dans la scène réelle est influencée par deux aspects essentiels: l'analyse graphique et le temps d'évolution des objets virtuels. L'analyse graphique est relative au temps de traitement du système d'augmentation. Elle inclue le temps d'analyse et de traitement des images de la séquence ainsi que le calcul pour l'alignement des objets virtuels. Par contre, le temps d'évolution est influencé par la vitesse de déplacement et d'acquisition des caméras. Il peut également être affecté par la manipulation si elle est supportée. Plus encore, l'analyse graphique en soit influe sur le temps d'évolution puisque le temps du rendu est directement liée à la complexité du dessin. Le non respect du temps réel implique une traînée apparente des objets virtuels par rapport à l'évolution naturelle de la scène. Un retard d'alignement devient visible dans la séquence et produit un effet de décalage temporel. Des objets qui sont lents ou rapides relativement à leur nature et/ou à la scène dénaturent l'augmentation, ce qui peut faire altérer l'objectif de la tâche.

Une évolution naturelle des objets virtuels, requise pour un réalisme visuel, n'est pas restreinte au simple déplacement. Le changement de l'apparence de leurs faces visibles durant le mouvement doit également être pris en considération. La contrainte temporelle est alors affectée si les objets sont amenés à se déplacer sous l'effet d'actions utilisateurs, et donc à changer d'apparence, ce qui est par nature l'objectif de certaines applications de la réalité augmentée. Les calculs inhérents au respect de ce facteur sont

généralement couteux en temps et ont un impact non négligeable sur le temps global de traitement.

Dans le même contexte, la manipulation des objets virtuels influe également sur le temps d'évolution. Une telle manipulation impose une contrainte. Il faut faire évoluer les objets virtuels, suite aux requêtes utilisateur, au même rythme de la scène réelle afin d'avoir une apparence adaptée aux actions qui leurs sont appliquées. En fonction de la nature des applications, cette contrainte peut être une impérative en tant que condition vitale pour le système. Pour certaines catégories d'applications par contre, elle est juste liée au réalisme visuel.

4.5. Collaboration

La réalité augmentée collaborative a pour objet d'établir un nouveau moyen de communication pour plusieurs utilisateurs par leur interaction simultanée avec les objets virtuels partagés. Dans ce contexte, ce nouveau mode de communication s'avère assez important pour certains types de tâches collaboratives.

La problématique inhérente est la synchronisation des actions des utilisateurs afin de simuler une manipulation naturelle. En plus du problème de partage de ressource – les objets virtuels en l'occurrence, se pose également le problème de partage de vues. Chacun des utilisateurs devrait disposer d'une vue spécifique de l'objet partagé. Cette vue est alors sujette à modification suite à toute action d'un utilisateur quelconque (Figure I.18).



Figure I.18. Collaboration de plusieurs utilisateurs dans la réalisation d'une tâche commune

La coprésence des utilisateurs, c'est-à-dire leur partage du même espace de travail pour la sémantique d'une telle collaboration, n'est pas une condition nécessaire. Ils peuvent être géographiquement distants. Cet aspect est relatif aux différents points de vue que chacun des utilisateurs peut avoir. D'une manière générale, à chaque caméra (ou ensemble de caméras pour le cas de la stéréovision) correspond un point de vue différent. Ainsi, les vues des utilisateurs relativement aux objets virtuels sont adaptées à la position de la caméra liée à chacun et relativement à la scène augmentée. Cependant, dans le cas où il n'existe qu'une seule caméra, particulièrement lorsque les utilisateurs

sont distants, un seul point de vue est disponible. La scène se trouve ainsi partagée et les objets virtuels manipulés selon ce seul point de vue.

D'une manière générale, il existe trois catégories de classes de systèmes de réalité augmentée collaborative. Chacune d'elles est basée sur la distance entre les utilisateurs et l'objet de la tâche. Le principe de base est qu'au moins un utilisateur doit être prêt de l'objet augmenté. Autrement, le système sera classé télé-opération n'ayant pas de relation avec la réalité augmentée collaborative [94].

- La première catégorie, dite collaboration distante, comprend les systèmes où au moins un utilisateur est physiquement devant l'objet de la tâche et quelques utilisateurs sont distants. L'environnement réel est augmenté par les informations fournies par les utilisateurs distants. C'est le cas du système présenté dans [47].
- La seconde catégorie, dite collaboration distante dans plusieurs réalités augmentées, concerne les systèmes comprenant plusieurs objets virtuels relatifs à plusieurs tâches. Ces objets sont physiquement répartis où chaque utilisateur agit sur son propre objet de la tâche. Un tel système implique que l'environnement réel de chaque utilisateur est augmenté par des informations qui lui parviennent des autres. Une telle catégorie de système est présentée par exemple dans le travail de [117] pour une augmentation collaborative sur tableau blanc qui fut l'idée de départ de plusieurs travaux récents dans le même contexte.
- La dernière catégorie, dite collaboration locale, représente les systèmes où les utilisateurs sont tous positionnés près de l'objet de la tâche. L'espace physique partagé est augmenté par les informations et les actions des utilisateurs. Le point clé de cette classe de systèmes est que la façon dont l'environnement est augmenté est différente pour chaque utilisateur. Un exemple de cette catégorie est le système présenté dans [71] du livre magique ou magicbook.

Les résultats actuels de la recherche en collaboratif sont encore à un stade primaire. Beaucoup d'efforts restent à fournir. Cependant, les résultats sont très prometteurs. La difficulté réside dans le fait qu'en plus des domaines auxquels la réalité augmentée fait appel, cette branche nécessite en plus les protocoles de communication interprocessus, les protocoles de synchronisation pour l'accès aux ressources partagées ainsi que les concepts associés aux systèmes répartis. C'est le minimum de concepts auxquels la réalité augmentée collaborative fait appel. De plus, malgré que des techniques d'interactions spécifiques pour la réalité augmentée aient été développées [17], les environnements collaboratifs exigent une considération particulière. Des utilisateurs distants nécessitent certainement des outils plus spécifiques dans leur interaction avec les objets virtuels distants. Pour ce qui est de notre travail dans ce contexte, nous nous sommes intéressés exclusivement à la communication et à la synchronisation. C'est ce qui fait l'objet du cinquième chapitre.

5. Domaines d'application

Nous présentons dans ce qui suit une liste de domaines d'application de la réalité augmentée, et particulièrement collaborative. Cette liste reste non exhaustive à cause de la diversité des objectifs des applications. Certaines constituent juste des prototypes tels que la médecine malgré que ce soit un domaine très investi [78]. D'autres sont des travaux achevés pouvant être éventuellement améliorés. Pour la majorité des applications, les moyens précédemment développés de la réalité virtuelle sont utilisés.

La réalité augmentée collaborative peut être utilisée dans de nombreux domaines basés sur les aspects collaboratifs. Les réunions de travail, la conception architecturale, l'enseignement visuel ou les jeux collaboratifs sont de bons exemples [8]. Citons également le cas du projet des 'Table-top systems' ou celui d'une collaboration d'un expert distant avec un apprenti pour la réparation d'un système compliqué [32, 33].

L'implémentation d'applications de la réalité augmentée est une tâche assez difficile. La mise en place d'une architecture nécessite une gestion de l'hétérogénéité des périphériques, des procédures de calibrage et du contrôle des différents algorithmes de traitement et du contrôle de l'affichage, etc. Différentes approches ont été utilisées dans le développement qu'il est possible de classifier comme suit [98]:

- monolithique;
- bibliothèques dédiées à la vidéo, à la vision (ARToolkit) et aux capteurs distribués;
- architectures objets;

Les bibliothèques supportent alors le mono-utilisateur et le multiutilisateur co-localisé ou distribué. Elles permettent le développement d'applications de la réalité augmentée. Notons que quelques travaux récents s'intéressent à la définition de méthodologie de réalisation d'architecture de réalité augmentée, telle que [119] qui définit un certain nombre de recommandations.

5.1. Environnements de réalité augmentée collaborative

Comme exemples de ce genre de travaux, nous présentons deux systèmes. Chacun d'eux supporte la modélisation d'environnements virtuels collaboratifs. Nous ne les présentons pas à titre comparatif mais à titre démonstratif. Ce qui est important dans ces systèmes, c'est qu'ils ne constituent pas des applications en eux-même mais des supports pour développer des prototypes d'applications.

Le premier système est caractérisé par son adaptation à plusieurs catégories d'objets virtuels [98] (Figure I.19). C'est un environnement multiutilisateur, multi-périphériques dédié aux applications collaboratives. Il repose sur une architecture modulaire et facilement configurable pour une mise en place rapide d'une session de travail. La plateforme a été utilisée pour démontrer des techniques d'interactions pour la manipulation d'objets réels et virtuels et la visualisation à plusieurs échelles. Des méthodes pour ajouter de façon dynamique des objets réels et virtuels ont été introduites, qui s'intègrent

alors dans le même espace. Basé sur des concepts de vue privée et de droits d'accès aux éléments de la scène, l'environnement gère l'interaction multiutilisateur.



Figure I.19. Plate forme spécialement conçue pour la réalisation d'applications de réalité augmentée collaborative

Le second système, JAPE [87], permet le développement d'applications et le test d'algorithmes spécifiques pour la réalité augmentée collaborative. Ses composants comprennent une boîte à outils pour le rendu 3-D, l'acquisition vidéo, la transmission vocale, ainsi que le contrôle des outils de tracking et d'interaction. Une importante démonstration d'une application implémentée à base de ce système est celle d'un musée virtuel (Figure I.20).



Figure I.20. Aperçu du musée virtuel du système JAPE

5.2. Enseignement

Des travaux ont été orientés vers la réalisation d'applications utilisant la réalité augmentée collaborative pour la création d'environnements d'apprentissage virtuel. C'est le cas de Construct3D basé sur un système mobile [48] (Figure I.21). Son objectif est de faciliter l'apprentissage et, particulièrement, l'amélioration des capacités spatiales de l'étudiant(e).

La version actuelle de Construct3D permet la construction de fonctions de base de primitives telles que des points, des lignes, des plans, des cubes, des sphères, des cylindres et des cônes. La construction inclue les intersections, les opérations

booléennes, les lignes et plans normaux, les opérations de symétrie et les prises de mesures.



Figure I.21. Aperçu de l'application Construct3D

5.3. Entraînement et apprentissage

Un nombre non négligeable de travaux ont été développés dans ce sens. Ils permettent à l'utilisateur d'être assisté dans l'apprentissage ou dans l'entraînement de la manipulation d'un objet réel [122]. [52] par exemple, présente une panoplie d'applications de l'idée d'agents animés. Une des applications consiste à assister un utilisateur d'un piano dans son apprentissage des touches et dans son entraînement à jouer de la musique (Figure I.22).



Figure I.22. Vue de l'application d'assistance dans l'apprentissage de la manipulation d'un piano

[126] présente une application assez importante à base d'ordinateurs portables supportant la distribution des utilisateurs. Ces derniers, situés sur des sites éloignés, collaborent pour des tâches d'entraînement par le partage de la vue d'un utilisateur local. Le prototype leur permet de manipuler les objets virtuels disposés dans un espace de travail partagé. L'importance de telles applications vient du fait que le coût d'entraînement d'un client ou d'un employé pour l'utilisation d'équipement est généralement cher. Le même principe est présenté par [24] (Figure I.23), mais pour le cas de l'entraînement et de l'assistance dans la maintenance d'équipement informatique (PC).



Figure I.23. Entraînement assisté dans la maintenance d'une carte électronique d'un ordinateur

5.4. Télé présence

Ce travail est principalement représenté par celui de [124] (Figure I.24). C'est un prototype démontrant la possibilité de visualiser un coté caché d'un office de travail afin d'avoir une collaboration distante. Le papier présente une démonstration d'une porte virtuelle ouverte sur un lieu de travail réel avec une définition graphique de haute qualité. Il est cependant limité à une reconstruction statique de la scène. Dans l'implémentation actuelle, la porte virtuelle est ouverte sur un mannequin, alors que l'extension future sera sur un collaborateur réel tel que décrit par l'auteur.



Figure I.24. Visualisation d'une personne distante (dans ce cas un mannequin)

5.5. E-Commerce

Ce genre d'application est en voix d'expansion. C'est d'ailleurs l'une des orientations de notre travail. Les avancées récentes dans les technologies de communication offre un E-commerce limité en information sur les dimensions physiques et l'effet tangible des produits. L'environnement réel de l'utilisateur est complètement ignoré. En superposant des produits virtuels 3-D au dessus d'un environnement réel, la réalité augmentée devrait résoudre ce problème. Comme exemple d'applications, [58] présente un système de réalité augmentée basé sur le tracking d'un laser projeté pour une application de E-commerce. Dans le même sens, [63] présente un travail similaire mais qui est beaucoup plus orienté vers la conception.



Figure I.25. Agencement d'une pièce par des objets virtuels où en réalité elle ne contient rien (coin haut droit)

5.6. Conception

Les idées associées à ce type d'application sont très variées. Dans [63], une application de design intérieur est présentée (Figure I.25). Elle a été créée pour démontrer le principe des interfaces distribuées dans le contexte de la réalité augmentée. Le scénario de l'application fait appel à un client qui désire acheter des fournitures pour décorer une chambre. Il contacte alors un vendeur distant. Lorsque ce dernier rejoint la session, il peut visualiser la chambre de son client. Les objets sont choisis à partir d'une base de données contenant les informations nécessaires telles que la couleur, le fabricant et le prix. Les deux utilisateurs peuvent ajouter, supprimer et arranger les objets selon gré, cependant, un objet n'est manipulé que par un seul utilisateur à la fois.

Dans [45], plusieurs utilisateurs sont capables de déplacer des objets virtuels dans une interface de réalité augmentée spécialisée appelée 'table-top' (Figure I.26). Les utilisateurs peuvent interagir simultanément avec la scène virtuelle.

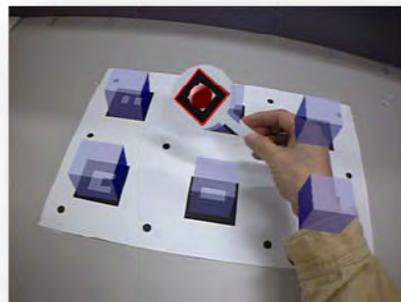


Figure I.26. Application Table-Top pour la manipulation d'objets

[53] décrit quelques exemples de l'utilisation de la réalité augmentée dans le processus de développement d'automobiles durant la phase de conception de nouvelles carrosseries. Les applications développées complètent les prototypes d'automobiles réels par des composants virtuels pour montrer une variété de conception ou aider dans la maintenance conceptuelle. La Figure I.27 donne une représentation d'un composant frontal virtuel sur une voiture réelle.



Figure I.27. Représentation d'une face virtuelle d'un microbus réel durant la phase de conception

5.7. Téléconférence et vidéoconférence

Dans [47] et [46], un système supportant la visualisation de plusieurs collaborateurs distants est présenté (Figure I.28). Leurs images sont visualisées sur des écrans virtuels. Les utilisateurs peuvent voir et interagir avec les objets virtuels en utilisant un tableau blanc partagé.

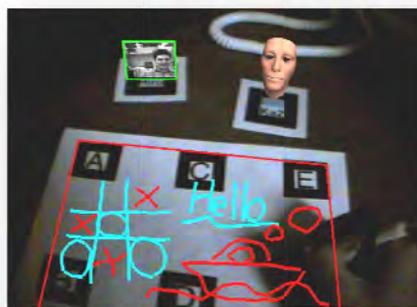


Figure I.28. Exemple d'applications de vidéo conférence

Le prototype supporte la collaboration entre un utilisateur portant un HMD et d'autres utilisateurs avec des interfaces de bureaux classiques. Seul l'utilisateur avec le HMD peut visualiser les images des autres utilisateurs. Les utilisateurs distants peuvent seulement voir la vidéo générée par la caméra du système d'augmentation. Ils peuvent alors librement dessiner sur le tableau blanc partagé. Un travail similaire est présenté

dans [73]. Dans [46], l'idée de l'élimination du background de la scène des écrans virtuels est présentée. Les auteurs stipulent que cette fonction renforce l'effet de coprésence des utilisateurs distants.

5.8. Récits historiques

Le retour dans le temps de manière visuelle est une application assez distrayante et très attractive d'un point de vue culturelle et didactique. C'est l'idée de certains travaux de récits historiques (storytelling) dans un environnement de réalité augmentée collaborative, tel que celui présenté dans [83]. Le travail est décrit comme étant une offre de plusieurs possibilités d'une telle application dans un environnement de réalité augmentée collaborative pour supporter l'expérience collaborative des utilisateurs. Il définit l'interaction dans une telle application et la positionne relativement aux jeux. Le travail est une combinaison de théâtre avec audience participante et une approche morphologique du récit historique. La Figure I.29 présente un utilisateur de l'histoire 'Geist' qui regarde l'esprit de 'Katharina' dans l'environnement historique.



Figure I.29. Visualisation de personnages historiques associés à des emplacements

Dans [34], une idée qui se rapproche de l'histoire mais sous une forme narrative est présentée. L'utilisateur, portant un HMD, prend le rôle d'Alice et participe à une conversation avec trois caractères virtuels qui lui paraissent assis à la même table. Un autre travail proche des deux précédents est également présenté dans [118].

5.9. Informations spatiales

Le travail présenté dans [39] et [79] représente un exemple d'application de localisation dans une construction dans le cadre de la réalité augmentée mobile (Figure I.30). Elle permet de visualiser des informations spatiales telles que la position et la direction à prendre dépendant de la position de l'utilisateur dans son environnement. Ce qui est intéressant dans ce travail, c'est le partage de l'environnement en espaces identifiables par des indices spécifiques. L'application présentée dans [79] est une aide pour la navigation intérieure dans une construction. La deuxième, [39] en l'occurrence, fournit une aide à la recherche de la position des livres dans une bibliothèque (Figure I.31).



Figure I.30. Visualisation spatiale: Orientation intérieure dans une construction



Figure I.31. Visualisation spatiale: Recherche de livres dans une bibliothèque

6. Conclusion

Après avoir défini la réalité augmentée et particulièrement l'aspect du collaboratif associé, le contenu du chapitre est structuré en trois parties. Une première partie fait une présentation des caractéristiques du domaine. Elle met la main sur les éléments de base et les propriétés essentielles de la réalité augmentée. Elle recense les composants matériels qui entrent en jeu dans une augmentation ainsi que les aspects qui procurent un sentiment de réalisme à l'utilisateur. La deuxième partie discute les principales orientations de la recherche dans le domaine. Elle est importante dans le sens de faire émerger les axes les plus importants et les plus investies par les travaux passés, actuels et futurs. A travers cette partie, nous avons fait apparaître les points centraux des travaux de recherche et les limites actuelles. Enfin, la dernière partie présente les domaines d'application de la réalité augmentée collaborative dont les travaux associés ont obtenu des résultats tangibles.

Dans le contexte de cette présentation, il devient clair que le domaine est assez vaste et qu'il nécessite de faire appel à certains résultats spécifiques trouvés dans d'autres domaines de recherche. Cela implique éventuellement l'utilisation d'outils spécialisés et de matériel puissant. En raison de cette pluralité, l'objectif de notre travail est fixé dans le sens de l'exploitation d'outils requérant le minimum de puissance, non spécialisés et utilisés à grande échelle. Ceci devrait servir à procurer une collaboration entre un

ensemble d'utilisateurs qui soit la moins coûteuse possible. C'est ainsi que ce travail est définie dans le sens de rechercher des solutions à la problématique de la minimisation du temps d'augmentation global. Il se base sur un système vidéo avec écrans ordinaires, il opte pour la solution image quant à la nature des objets virtuels, il utilise les outils de manipulation 2-D les plus largement et les plus communément utilisés et il propose une solution pour la communication distante d'utilisateurs requérant le minimum de moyens possibles mis à part les outils standards.

Notre travail touche à la nature des objets virtuels dont la solution proposée est présentée au quatrième chapitre. Il utilise la solution vidéo comme système d'augmentation où le réalisme visuel est étudié d'un point de vue particulier. Ce réalisme visuel auquel nous nous intéressons concerne le déplacement et la rotation des objets virtuels mais pas le graphisme proprement dit. Ce dernier, s'il est pris en considération, viendrait supporter notre travail. Ces concepts font l'objet d'une discussion faite au troisième et quatrième chapitre. En tant que caractéristiques de la réalité augmentée touchées par ce travail, l'interaction et la collaboration en sont également des objectifs importants. Elles sont traitées respectivement aux troisième et cinquième chapitres. Finalement, le chapitre suivant présente les bases théoriques supportant la majorité du travail.

The page features a decorative graphic consisting of three blue circles of varying sizes, each with a lighter blue ring around its center. These circles are arranged along a diagonal line that descends from the top left towards the bottom right. The largest circle is at the top right, a smaller one is in the middle, and a third, partially cut off by the bottom edge, is at the bottom right. The background is white with thin blue lines extending from the top left corner.

Chapitre II

Calibration de la Caméra pour les Augmentations 3-D

Le chapitre synthétise les fondements théoriques du principe d'augmentation relatif à notre travail.

Calibration de la Caméra pour les Augmentations 3-D

1. Introduction

Une augmentation vidéo consiste à analyser la scène numérique générée par la caméra, de trouver sa correspondance avec le monde réel et d'y insérer des objets virtuels. Plusieurs techniques sont utilisées suivant que l'environnement est préparé ou non [96]. Dans le cadre de notre travail, nous nous limitons à la technique vidéo dans un environnement préparé pour la raison de sa simplicité, son exactitude et son adaptation aux processus temps réels.

La technique d'augmentation utilisée dans ce travail consiste à rechercher des indices (appelés aussi marqueurs, référence ou patterns), dont la forme et la géométrie sont connues, dans la séquence générée. L'accent est mis particulièrement sur les indices 2-D. Une fois ces indices trouvés, ils permettent de déduire la géométrie de la scène et de procéder à l'augmentation. Cette opération d'augmentation nécessite au préalable une calibration de la caméra. Cette calibration peut être faite à priori pour les paramètres statiques avant le processus d'augmentation. Notons que le désavantage majeur de cette technique d'augmentation est de ne pas pouvoir effectuer des augmentations si les marqueurs sortent du champ de vision de la caméra.

Bien que notre travail ne soit pas directement concerné par le problème de la calibration de la caméra, il puise profondément dans les résultats obtenus. En effet, l'augmentation d'une scène nécessite généralement une procédure de calibration à l'exception d'un cas restreint que nous explicitons ci-après. C'est la raison pour laquelle nous détaillons dans ce qui suit certains concepts liés à notre travail et associés au problème de la calibration.

2. Calibration de la caméra

La calibration consiste à déterminer les paramètres intrinsèques et extrinsèques de la caméra. Ces paramètres permettent la connaissance de ses caractéristiques internes ainsi que de sa position et son orientation relativement à la scène réelle. En ce sens, ils permettent d'établir la relation entre les objets générés dans les images et leurs positions relatives dans la scène filmée. C'est ce qui donne la possibilité d'analyser la séquence générée pour déterminer l'emplacement d'incrustation des objets virtuels. Signalons que le zoom de la caméra est un facteur délicat à traiter durant la procédure de calibration de la caméra [41].

Formellement, calibrer une caméra consiste à déterminer de manière analytique la fonction qui associe à un point 3-D sa projection dans l'image fournie par la caméra. Généralement, cette fonction est liée à un modèle de caméra. Plusieurs modèles ont été

développés et présentés dans la littérature. [89] présente une étude détaillée dans sa thèse.

Afin d'estimer les paramètres de la caméra, la prise en compte d'une référence dans son champ de vision est nécessaire. Cette référence comprend un ensemble de points dont les coordonnées sont soit connues, soit déterminées par correspondance. Plusieurs techniques sont proposées. A titre indicatif et non restrictif, nous renvoyons par exemple à [18] et [90]. Les techniques existantes sont classifiées dans deux catégories [133]:

- **Calibration Photogrammétrique:** la calibration de la caméra est faite par l'observation d'un objet de référence dont la géométrie 3-D est connue avec précision. Il consiste souvent en deux ou trois plans orthogonaux. Cette catégorie comprend les techniques implémentées dans les environnements préparés. Ce genre d'approches requiert généralement un appareillage cher.
- **Self-calibration:** la calibration est faite à base du mouvement de la caméra dans une scène statique. Elle se base sur le principe de correspondance. Les images prises par la même caméra ou des caméras différentes avec des paramètres intrinsèques fixes sont suffisantes pour déterminer toutes leurs caractéristiques. Cette approche est très flexible mais elle n'est pas encore assez robuste [42]. Elle est coûteuse en temps de calcul et s'apprête difficilement aux applications temps réels [6, 81]. Elle s'apprête aux environnements non préparés telle qu'une forêt sans références connues à priori.

2.1. Modèles de camera

Un modèle de caméra est l'ensemble de lois géométriques définissant la façon dont se projette, lors du processus d'acquisition d'une image, un point de l'espace 3-D sur le plan 2-D de l'image. En ce sens, il permet de calculer, à partir des coordonnées cartésiennes d'un point de l'espace, les coordonnées en pixels sur l'image. Selon la précision désirée, plusieurs modèles ont été proposés, prenant en compte, plus ou moins fidèlement, la façon dont se déplace le faisceau de lumière pour constituer l'image.

En fonction de la précision escomptée, un modèle de caméra est basé sur une projection perspective ou orthographique. La transformation orthographique est une large approximation qui suppose que les objets dans l'espace 3-D sont projetés orthogonalement sur le plan image. Elle est applicable dans les applications de vision nécessitant peu de précision. Un modèle orthographique fournit des solutions simples et peu coûteuses en calcul. A l'inverse, la projection perspective offre un cadre mathématique assez robuste pour les systèmes de caméra de haute qualité.

Plusieurs modèles qui se basent sur la projection perspective sont définis dans la littérature. Le plus utilisé est le modèle sténopé. D'autres modèles qui constituent une simplification de ce dernier ou une spécificité pour certains domaines ou types de caméras sont utilisés. Cependant, les résultats obtenus sont généralement moins précis. Citons à titre d'exemple les modèles affine et œil de poisson (ou fish-eye) [55]. A base

du modèle affine, deux autres modèles sont également définis: modèle de projection à perspective faible et modèle de projection para-perspective. Pour notre cas, nous nous intéressons essentiellement au modèle sténopé que nous développons ci-après puisqu'il est à la base de tout notre travail.

2.2. Modèle sténopé

Le modèle de caméra le plus largement utilisé est le modèle perspectif linéaire (Figure II.1), aussi appelé modèle projectif ou modèle sténopé (pin-hole camera model). Il est utilisé pour les caméras perspectives. La plupart des méthodes de calibrage reposent sur ce modèle. En effet, non seulement il permet de modéliser fidèlement la plupart des capteurs projectifs, mais en plus il permet de simplifier les mathématiques mises en jeu pour l'estimation des paramètres du modèle. Son désavantage, par contre, est la perte de la perspective réelle ainsi que la distorsion des objets virtuels lorsqu'ils sont vus de près.

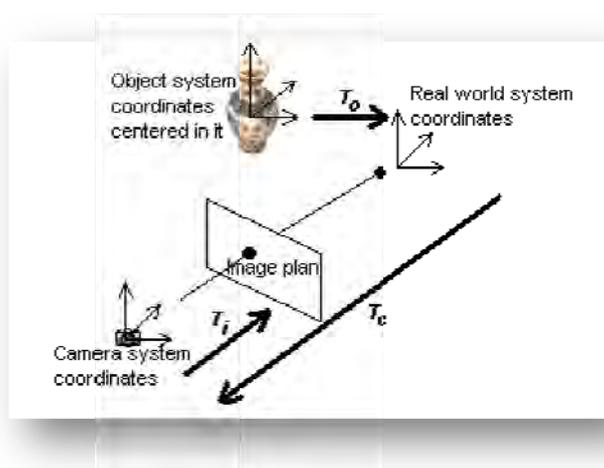


Figure II.1. Modèle sténopé utilisé

Le modèle sténopé est basé sur l'estimation des douze paramètres suivants:

f – longueur de la focale

k – coefficient de distorsion radiale

u_0, v_0 – coordonnées du centre optique

k_u, k_v – facteur d'échelle relatif à la taille des pixels

R_x, R_y, R_z – composantes de la rotation dans la transformation entre les systèmes de coordonnées du monde réel et de la caméra

T_x, T_y, T_z – composantes de la translation dans la transformation entre les systèmes de coordonnées du monde réel et de la caméra

Notons que deux autres paramètres sont cités dans [54] mais avec des contraintes supplémentaires détaillées.

La fonction de projection peut être décomposée en trois transformations distinctes. La première est une transformation entre l'espace 3-D et le système de coordonnées de la caméra (T_c). Elle fournit les paramètres extrinsèques de calibrage. La seconde est une transformation entre les coordonnées des points 3-D dans le repère associé à la caméra et les points 2-D dans le plan image (T_i). Les paramètres de cette transformation sont appelés intrinsèques. La troisième est une transformation entre le référentiel de l'objet et celui de l'espace 3-D (T_o). Elle permet de faire passer les coordonnées de l'objet à insérer, exprimés dans son propre système, vers celles de l'espace 3-D. La fonction de projection d'un point M de l'espace se décompose alors en trois parties : un changement de repère de l'espace, une projection perspective et un changement de repère dans le plan image.

Dans le modèle sténopé, la caméra est représentée par un plan rétinien π^i (Figure II.2), appelé plan image et référencé par un repère affine $(o, \vec{u}, \vec{v}, \vec{w})$, et un centre optique C (ou centre de projection) qui n'appartient pas à π^i , origine du repère $R_C(C, C_x, C_y, C_z)$. L'image d'un point $M(x, y, z)$ de l'espace est l'intersection de la droite (CM) avec le plan rétinien π^i . Le plan $\pi^f(C, C_x, C_y)$ parallèle à π^i passant par C est appelé plan focal. La projection orthogonale p de C sur π^i est appelé point principal. La droite $(CC_z) = (Cp)$ est appelée axe optique et la distance $f = [Cp]$ est dite distance focale. L'axe (CC_x) est parallèle au vecteur \vec{u} du plan rétinien.

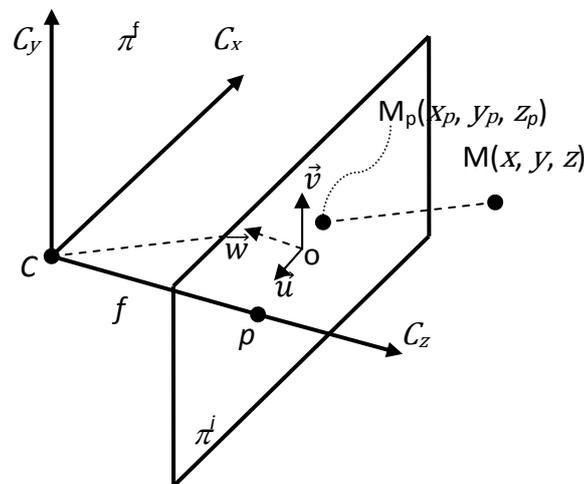


Figure II.2. Modèle géométrique du modèle projectif linéaire

2.3. Projection perspective

Considérons un point $M(x, y, z)$ quelconque dans l'espace référencé dans le repère R_C . Le point $M_p(x_p, y_p, f)$ associé dans le plan image, qui constitue sa projection, est l'intersection de la ligne passant par M et le centre de projection C avec le plan image. Ses coordonnées dans R_C sont facilement exprimées par le principe des triangles semblables (Figure II.3).

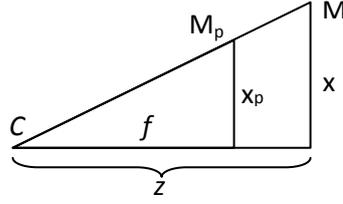


Figure II.3. Projection d'un point de l'espace dans le plan image

En supposant un plan image non distordu, il est facile de déduire que:

$$x_p = f \frac{x}{z}, y_p = f \frac{y}{z} \quad (\text{II.1})$$

et que le point projeté a donc pour coordonnées (II.2):

$$(x_p, y_p, f) = \left(f \frac{x}{z}, f \frac{y}{z}, f \right) \quad (\text{II.2})$$

Sachant que tout point $m = [x \ y \ z]^T$ est représenté en coordonnées homogènes par $m = \left[\frac{x}{\tilde{w}} \ \frac{y}{\tilde{w}} \ \frac{z}{\tilde{w}} \ 1 \right]^T$ où \tilde{w} est un facteur quelconque, l'expression précédente peut être écrite sous une forme matricielle $M_p = P \cdot M$, ou encore de manière explicite (II.3):

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = P \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f/z & 0 & 0 & 0 \\ 0 & f/z & 0 & 0 \\ 0 & 0 & f/z & 0 \\ 0 & 0 & 1/z & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{II.3})$$

Les vecteurs $M_p = [x_p \ y_p \ z_p]^T$ et $M = [x \ y \ z]^T$ représentent respectivement les coordonnées cartésiennes des points M_p et M . Notons que dans ce cas, $z_p = f$.

2.4. Transformation caméra-image

Pour écrire la matrice de transformation du repère caméra vers le repère image, notons les points suivants:

- Les points du plan image sont mesurés en pixels dans le repère affine $(o, \vec{u}, \vec{v}, \vec{w})$. La composante suivant l'axe \vec{w} est toujours nulle.
- Les coordonnées de C dans le repère image sont représentées par le vecteur $[u_0 \ v_0 \ w_0]^T$ avec $w_0 = -f$
- Les facteurs d'échelles horizontal et vertical sont respectivement k_u et k_v , mesurés en pixels/mm. Ils sont importants du fait que les pixels d'une caméra sont rarement carrés.

Ainsi, la transformation suivante (II.4) permet de calculer les coordonnées du point M_p (projection de M dans le plan image) dans le repère image à partir de celles dans le repère caméra:

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \begin{bmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \\ w_0 \end{bmatrix} \quad (\text{II.4})$$

La composante suivant l'axe \vec{w} étant toujours nulle, on peut alors ignorer la troisième ligne dans (II.4) et écrire la transformation sous la forme d'une matrice 3 x 4. Pour cela, posons K la matrice suivante (II.5):

$$K = \begin{bmatrix} -k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.5})$$

C'est la matrice de la transformation affine en coordonnées homogènes. L'expression (II.4) devient alors (II.6):

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix}_{pixel} = K \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}_{mm} \quad (\text{II.6})$$

Donc, les coordonnées du point M_p dans le repère $(o, \vec{u}, \vec{v}, \vec{w})$ sont représentées par le vecteur $\begin{bmatrix} \frac{u_p}{w_p} & \frac{v_p}{w_p} & 1 \end{bmatrix}^T$, avec $w_p = 1$.

2.5. Paramètres intrinsèques

En multipliant les matrices P et K (projection perspective suivie d'une transformation affine: $K.P$), nous pouvons écrire les équations du modèle géométrique de la caméra. Ces équations (II.7) expriment la relation entre les coordonnées caméra du point M et les coordonnées image du point M_p :

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = K.P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{II.7})$$

Soit, de manière explicite:

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \begin{bmatrix} -k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f/z & 0 & 0 & 0 \\ 0 & f/z & 0 & 0 \\ 0 & 0 & f/z & 0 \\ 0 & 0 & 1/z & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{II.8})$$

ce qui donne les relations analytiques (II.9):

$$\begin{cases} u_p = -k_u f \frac{x}{z} + u_0 \\ v_p = k_v f \frac{y}{z} + v_0 \end{cases} \quad (\text{II.9})$$

Avec $w_p = 1$, puisque le produit $K.P$ est (II.10):

$$K.P = \begin{bmatrix} -k_u \frac{f}{z} & 0 & \frac{u_0}{z} & 0 \\ 0 & k_v \frac{f}{z} & \frac{v_0}{z} & 0 \\ 0 & 0 & \frac{1}{z} & 0 \end{bmatrix} \quad (\text{II.10})$$

En multipliant tous les coefficients de la matrice par z (ce qui ne change pas le résultat puisque les coordonnées homogènes sont définies à un facteur multiplicatif près), on obtient la matrice des paramètres intrinsèques (II.11):

$$M_{int} = \begin{bmatrix} -k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.11})$$

Cette transformation est composée d'une projection perspective sur le plan image suivie d'une transformation affine. D'où l'on obtient:

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \frac{1}{z} M_{int} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha M_{int} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (\text{II.12})$$

Ce modèle comporte donc quatre paramètres : $\alpha_u = -k_u f$, $\alpha_v = k_v f$, u_0 et v_0 qui sont estimés par calibrage. Notons que la distance focale ne peut être calculée explicitement. De même, le paramètre α fait que les coordonnées du point M deviennent sans unité, i.e. $(\frac{x}{z}, \frac{y}{z}, 1)$. En posant $x_c = \frac{x}{z}$ et $y_c = \frac{y}{z}$, on obtient (II.13):

$$\begin{cases} u_p = \alpha_u x_c + u_0 \\ v_p = \alpha_v y_c + v_0 \end{cases} \quad (\text{II.13})$$

Soit en notation matriciel, en faisant réintroduire les coordonnées homogènes:

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = M_{int} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{II.14})$$

En pratique, les vecteurs du repère image en coordonnées pixels ne sont pas tout à fait perpendiculaires. Plus particulièrement, c'est l'axe de vecteur directeur \vec{u} qui n'est pas parfaitement parallèle à l'axe C_x . Pour modéliser cet effet, l'angle θ entre les vecteurs \vec{u} et \vec{v} est pris en considération. Un nouveau repère de coordonnées image est alors défini ayant comme relation (II.15):

$$\begin{cases} u'_p = u_p \sin(\theta) + v_p \cos(\theta) \\ v'_p = v_p \end{cases} \quad (\text{II.15})$$

La matrice de paramètres intrinsèques devient donc (II.16):

$$M_{int} = \begin{bmatrix} -k_u f & k_v f \cos(\theta) & u'_0 \\ 0 & \frac{k_v f}{\sin(\theta)} & v'_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.16})$$

Notons qu'en pratique, rares sont les applications qui prennent en considération ce facteur de non perpendicularité des axes. Cela n'est nécessaire que si une grande précision est requise au niveau des résultats. Pour notre cas, nous utilisons l'expression (II.14) en raison de la précision réduite escomptée relativement à notre travail.

2.6. Distorsion radiale

Lorsque le champ de vue de la caméra utilisée est large, la distorsion radiale – une des distorsions non-linéaires les plus sévères – doit être prise en considération. Ceci est nécessaire particulièrement lorsqu'une grande exactitude dans toutes les parties de l'image est requise. En effet, ces distorsions sont particulièrement ressenties sur les périphéries de l'image. La tâche d'estimation de la distorsion radiale consiste à trouver un modèle qui permet de l'éliminer [68]. Elles peuvent être corrigées en utilisant un simple déplacement radial de la forme (II.17):

$$\begin{cases} x = u_p(1 + k_1 r^2 + k_2 r^4) \\ y = v_p(1 + k_1 r^2 + k_2 r^4) \end{cases} \quad (\text{II.17})$$

où (u_p, v_p) représente le point distordu dans le repère de la caméra et $r^2 = u_p^2 + v_p^2$. Les facteurs k_1 et k_2 sont les paramètres de distorsion où $k_2 \ll k_1$. D'ailleurs, k_2 est généralement fixé à 0. C'est la raison pour laquelle on ne considère qu'un seul paramètre de distorsion k faisant partie des paramètres intrinsèques à déterminer.

2.7. Paramètres extrinsèques

Le modèle sténopé présenté dans la Figure II.1 montre que le problème à résoudre pour la calibration de la caméra pour augmenter une scène correspond aux trois transformations T_o , T_c et T_i . Ces transformations permettent de faire le passage des coordonnées de l'objet à insérer, exprimés dans son propre système, vers celles du système de coordonnées du plan image. La transformation T_o permet de faire un changement de repère entre le système de coordonnées d'un objet (qui en est centré) et celui de la scène réelle afin de déterminer sa position et son orientation dans ce dernier. La transformation T_c représente les paramètres extrinsèques qui sont une rotation R et une translation T de la caméra (explicitées ci-après). Ils permettent d'exprimer la position de chaque point $p(x, y, z)$ de la scène dans le repère de la caméra. Cette transformation est dite point de vue de la caméra (ou camera viewpoint). La transformation T_i transforme tout point $p(x, y, z)$ exprimé dans le système de la caméra en sa projection $p'(u_p, v_p, w_p)$ sur le plan image. C'est la transformation explicitée précédemment (II.14) et qui permet d'estimer les paramètres intrinsèques.

Les paramètres extrinsèques sont concernés par les propriétés externes de la caméra et sont parfois dits externes. Ils déterminent la transformation T_c entre le système de

coordonnées inconnu de la caméra et le système de coordonnées connu de la scène réelle. Comme représentés sur la Figure II.4, ces paramètres sont:

- Une matrice rotation R (3x3) qui relie les axes des deux systèmes de coordonnées.
- Un vecteur de translation T (3-D) qui décrit les positions relatives des deux systèmes de coordonnées.

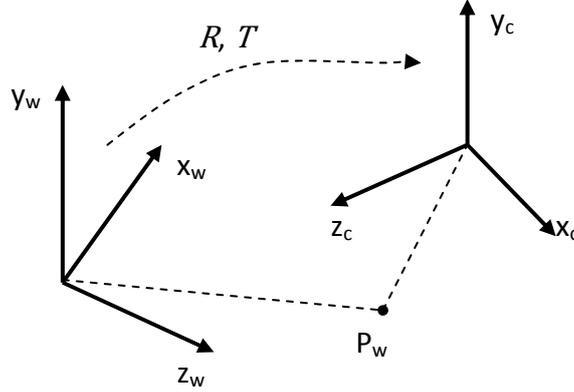


Figure II.4. Transformation d'un point du référentiel de la scène vers le référentiel de la caméra

En notation matricielle, les positions relatives des deux systèmes sont exprimées par (II.18):

$$P_c = RP_w + T \quad (\text{II.18})$$

où P_c et P_w sont respectivement les coordonnées du même point dans le repère de la caméra et celui de la scène réelle. De manière plus explicite:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (\text{II.19})$$

En coordonnées homogènes, la transformation (II.19) peut être réécrite sous la forme (II.20):

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (\text{II.20})$$

La matrice (II.21):

$$M_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (\text{II.21})$$

est dite matrice de transformation homogène et représente les paramètres extrinsèques.

2.8. Transformation référence-image

Par substitution de l'équation (II.14) avec cette dernière (II.20) ([113] présente les détails), il est possible de passer des coordonnées d'un point exprimées dans le système du référentiel vers celles exprimées dans le système de coordonnées du plan image. On obtient l'expression de la projection globale (II.22):

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \begin{bmatrix} -k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (\text{II.22})$$

Soit:

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \begin{bmatrix} -k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_{int} M_{ext} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (\text{II.23})$$

Pour enfin obtenir la matrice de projection perspective (II.24):

$$P = M_{int} M_{ext} = \begin{bmatrix} -k_u f r_1 + u_0 r_3 & -k_u f t_1 + u_0 t_3 \\ k_v f r_2 + v_0 r_3 & k_v f t_2 + v_0 t_3 \\ r_3 & t_3 \end{bmatrix} \quad (\text{II.24})$$

où $r_i = [r_{i1} \ r_{i2} \ r_{i3}]$, $i = 1..3$

Si nous imposons les valeurs $u_0 = 0$ et $v_0 = 0$, ce qui est acceptable dans la majorité des cas puisqu'ils sont contrôlés au niveau de la fabrication, la matrice se simplifie en (II.25):

$$P = \begin{bmatrix} -k_u f r_{11} & -k_u f r_{12} & -k_u f r_{13} & -k_u f t_1 \\ k_v f r_{21} & k_v f r_{22} & k_v f r_{23} & k_v f t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (\text{II.25})$$

En prenant en considération l'angle fait entre les axes \vec{u} et \vec{v} , le modèle global de la caméra devient (II.26):

$$P = \begin{bmatrix} \alpha_u r_1 + \alpha_u \cos(\theta) r_2 + u'_0 r_3 & \alpha_u t_1 + \alpha_u \cos(\theta) t_2 + u'_0 t_3 \\ \frac{\alpha_v}{\sin(\theta)} r_2 + v'_0 r_3 & \frac{\alpha_v}{\sin(\theta)} t_2 + v'_0 t_3 \\ r_3 & t_3 \end{bmatrix} \quad (\text{II.26})$$

Finalement, si nous posons:

$$P = \begin{bmatrix} m_1 & m_{14} \\ m_2 & m_{24} \\ m_3 & m_{34} \end{bmatrix} \quad (\text{II.27})$$

avec $m_i = [m_{i1} \ m_{i2} \ m_{i3}]$, $i = 1..3$, il est possible de calculer les paramètres intrinsèques et extrinsèques en fonction des coefficients de P en tenant compte des propriétés d'ortho normalité de la rotation. On obtient:

$$\left\{ \begin{array}{l} r_3 = m_3 \\ u_0 = m_1 m_3 \\ v_0 = m_2 m_3 \\ k_u f = -\|m_1 \wedge m_3\| \\ k_v f = \|m_2 \wedge m_3\| \\ r_1 = \frac{-1}{k_u f} (m_1 - u_0 m_3) \\ r_2 = \frac{-1}{k_v f} (m_2 - v_0 m_3) \\ t_1 = \frac{-1}{k_u f} (m_{14} - u_0 m_{34}) \\ t_2 = \frac{-1}{k_v f} (m_{24} - v_0 m_{34}) \\ t_3 = m_3 \end{array} \right. \quad (\text{II.28})$$

L'ensemble de ces équations permet donc de trouver les paramètres intrinsèques et extrinsèques de la caméra après avoir estimé les coefficients de la matrice de projection P . Ceci est possible lorsque l'on dispose de points de l'espace euclidien et de leurs projections dans l'image. Pour cela, il suffit de six correspondances $P_c \leftrightarrow P_w$ pour estimer P linéairement [56]. Comme la matrice de projection est définie à un facteur multiplicatif près, il faut rajouter une contrainte supplémentaire, soit:

- $m_{34} = 1$ qui conduit à la résolution d'un système linéaire sur-contraint, ou
- $\|m_3\|^2 = 1$ qui conduit à la méthode proposée par [86].

Il est également possible d'utiliser des méthodes non linéaires en minimisant la distance entre la projection théorique d'un point de l'espace 3-D et la mesure de sa position dans l'image. Nous renvoyons à [7] et [101] pour une étude détaillée.

3. Types de calibration

3.1. Calibrage fort

Le principe de calibration de la caméra dit fort consiste à utiliser une grille ou tout autre objet de calibrage, pour lequel les positions de certains points particuliers marqués sur l'objet, appelés points de contrôle, sont connues. Ces points peuvent être des coins, des points, des intersections de droites ou toutes autres primitives pouvant être facilement extraites à partir de l'image numérique. Typiquement, cet ensemble de points de contrôle étant de coordonnées 3-D connues, le problème consiste à déterminer les éléments de la matrice de projection P pour que leur projection corresponde au mieux aux mêmes points extraits de l'image.

3.2. Calibrage faible

Le calibrage dit faible consiste à calibrer deux caméras (ou plus) sans aucune connaissance à priori des paramètres des caméras ou de la scène observée. En effet, on peut montrer qu'à partir seulement de correspondances de points entre les images des

deux cameras, on arrive à calibrer complètement la géométrie épipolaire du système stéréoscopique, c'est-à-dire un ensemble de deux cameras ou plus rigidement liées observant la même scène [100]. Notons que le besoin de plusieurs cameras n'est pas une impérative puisque les images d'une seule caméra relatives à plusieurs points de vues différents de la même scène peuvent être utilisées. En pratique, c'est le type d'application qui impose le nombre de cameras requises pour la calibration.

Lorsque les paramètres intrinsèques des cameras sont connus, la géométrie épipolaire est décrite par la matrice essentielle, qui agit sur les points des images en coordonnées normalisées, et le calcul de cette matrice correspond à la détermination du mouvement et de la structure à partir de correspondances. Par contre, si on ne dispose pas de ces paramètres, la géométrie est décrite par la matrice fondamentale, qui agit sur les images en coordonnées pixels, et qui contient des informations permettant de simplifier la mise en correspondance de points entre les images et de trouver la structure 3-D projective de la scène. Ce type de calibrage est important pour la reconstruction d'images. Citons, à titre d'exemple, les objets cachés, les images mosaïques et le warping (Figure II.5).



Figure II.5. Principe de vues cylindriques ou mosaïque (gauche) et images plane ou warping (droite)

3.3. Calibrage automatique (ou auto-calibrage)

La dernière technique de calibrage est celle qui ne nécessite aucun point de coordonnées 3-D connues. Elle est nommée méthode d'auto-calibrage. Elle permet de retrouver la structure euclidienne de l'espace à un facteur d'échelle globale près. Le mouvement de la caméra filmant la scène statique est supposé connu. Des contraintes sur les paramètres intrinsèques supposés statiques et la rigidité des objets filmés sont imposées afin de calibrer la caméra en utilisant uniquement les informations de l'image.

Malgré son imprécision dans les résultats obtenus relativement aux autres méthodes de calibration de la caméra, cette dernière technique a été choisie dans nos travaux en raison de sa simplicité de mise en œuvre et de sa flexibilité. De plus, un certain nombre de travaux récents ont montré son efficacité, particulièrement avec les HMD [72]. C'est pour cette raison que nous développons particulièrement ci-après le principe d'auto calibration de la caméra à base d'indices planaires [16].

4. Indices utilisés dans l'augmentation

Telle que nous l'avons établi précédemment, une augmentation nécessite généralement des références ou des indices connus dans la scène qu'il est nécessaire d'identifier pour pouvoir calibrer la caméra. Plusieurs types d'indices ont été utilisés variant de l'explicite à l'implicite. Le but en est de trouver une correspondance entre les points de l'espace et ceux générés dans l'image à augmenter. La correspondance consiste en fait à trouver les différentes transformations (T_o , T_c , T_i) du modèle sténopé reliant ces points. Ce qui est important, c'est que leurs positions dans le monde réel doivent en général être connues. Pour cette raison, les indices utilisés doivent être de formes et de tailles bien connues puisqu'ils jouent le rôle de références pour le processus d'augmentation.

4.1. Références explicites

Les références explicites sont celles qui sont insérées par l'utilisateur suivant une disposition particulière. Elles ont des formes connues et sont disposées de manière à stabiliser et accélérer le système d'augmentation. Les augmentations à base de références explicites ont prouvé leur efficacité malgré leur inconvénient majeur de devoir déformer l'environnement naturel par des marqueurs. Ils appartiennent à deux catégories suivant qu'ils forment une structure 2-D ou 3-D. La structure 3-D permet de déterminer directement la matrice de projection P . La structure 2-D, quant à elle, nécessite un calcul supplémentaire que nous détaillons ci-après. Cette dernière méthode est moins précise que la première mais elle est plus souple particulièrement en raison de l'avancée assez considérable constatée dans beaucoup de travaux récents.

Les indices 3-D varient de la mire aux points colorés disposés suivant une forme particulière sur des plans de référence connus, jusqu'aux formes géométriques particulières d'objets ou de dessins. Chaque type d'indice est spécifique à un cas particulier d'application. Nous ne détaillons pas les concepts théoriques relatifs à ce type d'indices. Cependant, les fondements associés restent proches de ceux 2-D. Les images suivantes (Figure II.6) donnent un aperçu de tels indices. Nous renvoyons par exemple à [4], [64], [77], [105], [122] et [131] pour des cas d'étude.

Les indices 2-D sont des objets planaires, généralement de forme rectangulaire, comportant des dessins de modèles particuliers (pattern) et ayant des tailles connues (Figure II.7). D'ailleurs, beaucoup d'applications se basent sur ces formes pour réaliser des augmentations spécifiques. L'utilisation de tels indices prend de plus en plus de l'ampleur dans le domaine de la réalité augmentée en raison de l'importance des applications réalisées [62, 104, 133].

L'importance de tels indices planaires réside dans la technique de calibrage associée qui ne nécessite aucun point de coordonnées 3-D connues. Elle permet de retrouver la structure euclidienne de l'espace à un facteur d'échelle globale près, puisqu'il n'y a à priori aucun étalon de distance pour avoir une notion d'échelle (Figure II.8). Notons que beaucoup de travaux sont réalisés à base d'indices planaires.



Figure II.6. Différents type d'indices utilisés pour réaliser des augmentations. Du haut en bas et de gauche à droite: mire 3-D, points colorés, points non colorés, formes géométriques et objets réels

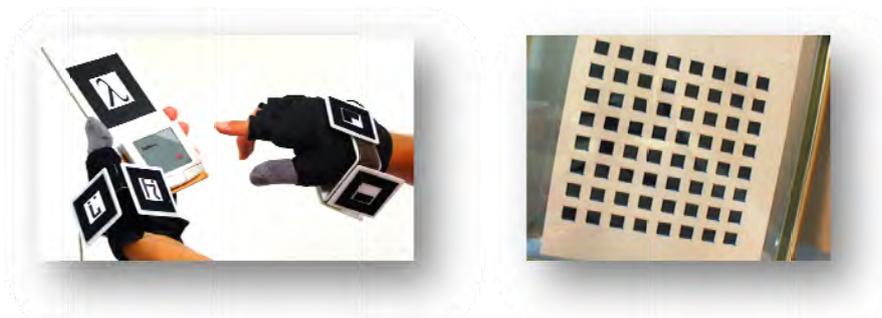


Figure II.7. Objets planaires utilisés comme indice 2-D

C'est le type d'indices pour lequel nous avons opté dans nos travaux d'augmentation. Il est particulièrement utile pour le type d'objet virtuel proposé que nous détaillons dans le quatrième chapitre. La méthode de calibration à base de tels indices et les concepts théoriques associés sont détaillés ci-après. Une étude comparative de trois méthodes de calibration utilisant des indices planaires peut être trouvée dans [51].



Figure II.8. Augmentations à base d'indices planaires

4.2. Références implicites

Les environnements non préparés sont ceux qui utilisent des indices implicites. En fait, ils se basent sur l'extraction d'indices visuels faisant naturellement partie de la scène [40, 95, 125, 130]. Ces indices peuvent être un plan, des formes d'objets rigides connus, des ombres et toutes autres caractéristiques pouvant être exploitées (lignes, textures, primitives géométriques, etc.) et dépendent de l'application cible (Figure II.9). Certains travaux font appel à l'interaction humaine durant la phase initiale du processus d'augmentation à cause de la difficulté inhérente à l'automatisation de la tâche. Les augmentations obtenues suivant ce principe sont plus naturelles que celles utilisant des indices explicites puisqu'elles n'introduisent aucune entité artificielle dans la scène. Cependant, les résultats actuels de cette catégorie de travaux ne s'adaptent généralement pas au temps réel.

5. Indice planaire et auto calibration de la caméra

Cette section présente un développement des concepts associés à la calibration de la caméra à base d'indices planaires. Elle consiste en une démonstration du principe de calcul de la matrice de projection à défaut d'informations spatiales. Le modèle de base utilisé est toujours le modèle sténopé. Ce type de calibration est à la base de tous nos travaux d'augmentation.

5.1. Homographie

L'utilisation d'un indice planaire 2-D simplifie les équations précédemment étudiées pour la calibration de la caméra du fait que la coordonnée $z = 0$ pour tous les points disposés sur le plan de l'indice. En effet, la correspondance de points 3-D dans l'image n'est plus possible. Les points à utiliser sont tous coplanaires. Plusieurs travaux ont

montré que ce type d'indice permet de calibrer une caméra et de procéder à l'augmentation [7, 18]. Dans ce cas, la projection globale est exprimée sous la forme (II.29):

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = \lambda H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad (\text{II.29})$$

où λ est un facteur d'échelle et la matrice H est dite matrice Homographie.



Figure II.9. Augmentations se basant sur des éléments extraits de la scène et sans indices explicites

Pour trouver les éléments de la matrice H , seulement quatre points non colinéaires $(x_{w_i}, y_{w_i})_{i=1,4}$ et leurs correspondants dans l'image $(u'_{p_i}, v'_{p_i})_{i=1,4}$ (II.30) sont nécessaires. La solution du système d'équation ne peut être obtenue par la méthode linéaire du fait que le nombre d'inconnus est supérieur au nombre d'équations. En effet, la matrice H comporte neuf inconnus à déterminer où le nombre d'équations disponible est seulement de huit. Ainsi, il est facile de réaliser que (II.30):

$$\begin{cases} \frac{u_p}{w_p} = \frac{h_{11}x_w + h_{12}y_w + h_{13}}{h_{31}x_w + h_{32}y_w + h_{33}} = u'_p \\ \frac{v_p}{w_p} = \frac{h_{21}x_w + h_{22}y_w + h_{23}}{h_{31}x_w + h_{32}y_w + h_{33}} = v'_p \end{cases} \quad (\text{II.30})$$

En notation matricielle, cela donne (II.31):

$$\begin{bmatrix} x_w & y_w & 1 & 0 & 0 & 0 & -x_w u'_p & -y_w u'_p & -u'_p \\ 0 & 0 & 0 & x_w & y_w & 1 & -x_w v'_p & -y_w v'_p & -v'_p \end{bmatrix} h = Ah = 0 \quad (\text{II.31})$$

avec $h = [h_{11} \ h_{12} \ h_{13} \ h_{21} \ h_{22} \ h_{23} \ h_{31} \ h_{32} \ h_{33}]^T$ les éléments de la matrice homographie à déterminer. En conséquence, quatre points de correspondance ne permettent pas de déterminer les neuf éléments de H , c'est-à-dire qu'il n'existe pas de solution unique au système. C'est pourquoi la solution est recherchée via la méthode de décomposition par valeurs singulières (SVD) [35] (voir annexe A pour un détail relatif). Ceci est possible par l'imposition de la contrainte $\|h\| = 1$. h est alors le vecteur propre correspondant à la valeur propre minimale de la matrice $A^T A$.

La matrice H obtenue permet de faire des projections directes d'objets planaires 2-D dans la séquence générée. Ceci est réalisé par la projection des points de l'objet virtuel dans le plan image. Ce qui est important dans cette projection (Figure II.10), c'est qu'elle ne nécessite aucune calibration de la caméra.



Figure II.10. Augmentation 2-D d'une image à base de l'homographie calculée

5.2. Estimation des éléments de la matrice de projection 3-D

La détermination des paramètres de calibration est indispensable pour pouvoir augmenter la scène avec des objets virtuels 3-D. [113] montre qu'il est possible d'estimer les paramètres de la caméra, en l'occurrence la matrice de projection P , en utilisant l'homographie calculée. Cette solution proposée élimine le besoin de l'étape de calibrer la caméra de manière séparée. Elle est auto-calibrée. Pour cela, il suffit d'exploiter la relation qui existe entre les éléments des matrices H et P et par conséquent déduire ceux de P . A base des relations (II.29) et (II.25), on peut établir (II.32):

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = P \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -k_u f r_{11} & -k_u f r_{12} & -k_u f r_{13} & -k_u f t_1 \\ k_v f r_{21} & k_v f r_{22} & k_v f r_{23} & k_v f t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix} \quad (\text{II.32})$$

La matrice H sera donc de la forme (voir [113] pour le détail) (II.33):

$$\lambda H = \begin{bmatrix} -k_u f r_{11} & -k_u f r_{12} & -k_u f t_1 \\ k_v f r_{21} & k_v f r_{22} & k_v f t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \quad (\text{II.33})$$

En exploitant les propriétés des matrices rotations et particulièrement (II.34):

$$\begin{cases} r_{11}^2 + r_{21}^2 + r_{31}^2 = 1 \\ r_{12}^2 + r_{22}^2 + r_{32}^2 = 1 \\ r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} = 0 \end{cases} \quad (\text{II.34})$$

Il est possible dans un premier temps de calculer $k_u f$, $k_v f$ et λ . En remplaçant les éléments de H identifiés par l'expression (II.33) dans la dernière équation du système (II.34), on obtient (II.35):

$$\frac{h_{11}h_{12}}{k_u^2 f^2} + \frac{h_{21}h_{22}}{k_v^2 f^2} + h_{31}h_{32} = 0 \quad (\text{II.35})$$

De la même façon, le remplacement dans les deux premières équations de (II.34) permet d'obtenir (II.36) et (II.37):

$$\lambda^2 \left(\frac{h_{11}^2}{k_u^2 f^2} + \frac{h_{21}^2}{k_v^2 f^2} + h_{31}^2 \right) = 1 \quad (\text{II.36})$$

et

$$\lambda^2 \left(\frac{h_{12}^2}{k_u^2 f^2} + \frac{h_{22}^2}{k_v^2 f^2} + h_{32}^2 \right) = 1 \quad (\text{II.37})$$

Par soustraction de (II.37) de (II.36), on obtient (II.38):

$$\frac{h_{11}^2 - h_{12}^2}{k_u^2 f^2} + \frac{h_{21}^2 - h_{22}^2}{k_v^2 f^2} + h_{31}^2 - h_{32}^2 = 0 \quad (\text{II.38})$$

Par résolution du système d'équation formé des équations (II.35) et (II.38), on obtient (II.39):

$$\begin{cases} k_u f = \sqrt{\frac{h_{11}h_{12}(h_{21}^2 - h_{22}^2) - h_{21}h_{22}(h_{11}^2 - h_{12}^2)}{h_{21}h_{22}(h_{31}^2 - h_{32}^2) - h_{31}h_{32}(h_{21}^2 - h_{22}^2)}} \\ k_v f = \sqrt{\frac{h_{21}h_{22}(h_{11}^2 - h_{12}^2) - h_{11}h_{12}(h_{21}^2 - h_{22}^2)}{h_{11}h_{12}(h_{31}^2 - h_{32}^2) - h_{31}h_{32}(h_{11}^2 - h_{12}^2)}} \end{cases} \quad (\text{II.39})$$

Une fois ces valeurs calculées, on peut déduire la valeur de λ (II.40):

$$\lambda = \frac{1}{\sqrt{\frac{h_{11}^2}{k_u^2 f^2} + \frac{h_{21}^2}{k_v^2 f^2} + h_{31}^2}} \quad (\text{II.40})$$

puis le reste des éléments de R et T par simples remplacements (II.41):

$$\begin{cases} r_{11} = -\lambda \frac{h_{11}}{k_{uf}}, r_{12} = -\lambda \frac{h_{12}}{k_{uf}}, r_{13} = r_{21}r_{32} - r_{31}r_{22} \text{ et } t_1 = -\lambda \frac{h_{13}}{k_{uf}} \\ r_{21} = \lambda \frac{h_{21}}{k_{vf}}, r_{22} = \lambda \frac{h_{22}}{k_{vf}}, r_{23} = r_{31}r_{12} - r_{11}r_{32} \text{ et } t_2 = \lambda \frac{h_{23}}{k_{vf}} \\ r_{31} = \lambda h_{31}, r_{32} = \lambda h_{32}, r_{33} = r_{11}r_{22} - r_{21}r_{12} \text{ et } t_3 = \lambda h_{13} \end{cases} \quad (\text{II.41})$$

C'est ce qui permet de déterminer la matrice de projection P à partir de la matrice homographie H et d'augmenter la scène avec des objets 3-D, supposés positionnés relativement au système de coordonnées de l'indice planaire (Figure II.11). En d'autres termes, l'homographie peut être utilisée pour auto-calibrer la caméra. C'est une des raisons qui a fait que nous avons opté pour cette solution dans notre travail.

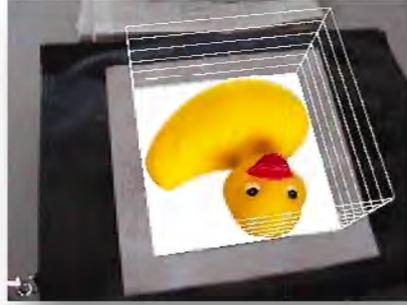


Figure II.11. Augmentation 3-D après estimation des paramètres de projection 3-D à partir de l'homographie

5.3. Algorithme de calibration à base d'indice planaire

5.3.1. Principe général

La résolution des équations de la section précédente impose une démarche séquentielle dans l'augmentation [26]. La première étape nécessite la recherche de l'indice dans l'image et donc au minimum quatre points connus de celui-ci, en l'occurrence ses coins. Cela passe par une binarisation (formée de pixels noirs et blancs) de l'image à base d'un seuil choisi dynamiquement en fonction de l'éclairage de la scène (Figure II.12). Le choix du seuil pour chaque image de la séquence est la moyenne (II.42):

$$Th = \frac{\sum_{j=1}^n v_j}{n} \quad (\text{II.42})$$

où n est le nombre de pixels dans l'image et v_j la valeur du pixel j après être transformé en niveau de gris. Ainsi, chaque pixel j change sa valeur en (II.43):

$$\begin{cases} 0 \text{ si } v_j < Th \\ 1 \text{ si } v_j \geq Th \end{cases} \quad (\text{II.43})$$

Cette binarisation est suivie d'une recherche des régions connexes noires possédant quatre coins (Figure II.13). Ceci est une conséquence de la forme des indices utilisés dans ce domaine [24]. Pour chacune des régions trouvées, quatre homographies

$H_k, k = 1..4$, correspondant aux quatre orientations possibles de l'indice, sont calculées. Chaque H_k permet de transformer (projeter) la région dans le repère de l'indice et donc d'obtenir l'image relative. Par une différence binaire simple avec l'image d'origine de l'indice, la région qui donne un résultat minimal, relativement à un certain seuil fixé à priori, sera considérée comme étant l'indice. Le H_k correspondant permet de déduire les matrices M_{int} et M_{ext} . Le seuil choisi à ce niveau est purement aléatoire et il n'existe pas de technique pour le choisir de manière optimale. Il est même possible qu'une région anormale donne une différence meilleure que celle de l'indice. Notons qu'il n'est pas nécessaire de réévaluer la matrice M_{int} pour chaque image de la séquence en cas où les paramètres de zoom restent inchangés puisque ses valeurs restent constantes.

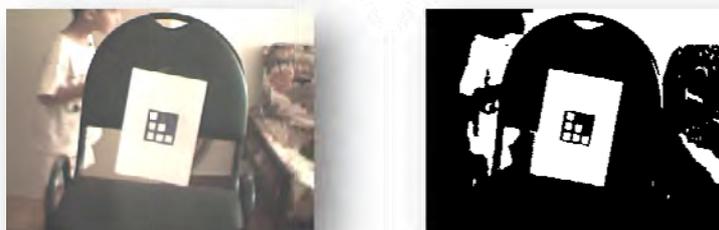


Figure II.12. Binarisation d'une image



Figure II.13. Détection des coins d'un indice planaire

5.3.2. Recherche de la région de l'indice

Pour rechercher les régions connexes, un principe d'indexation est utilisé. Chaque pixel noir trouvé est étiqueté comme faisant partie d'une région i et, de manière récurrente, tout autre pixel connexe fera partie de cette région. Pour ce faire, l'image est balayée de haut en bas et de gauche à droite. Tout pixel, non encore étiqueté, adjacent à un autre déjà étiqueté i , sera également étiqueté i .

Un seuil est généralement utilisé pour rejeter les régions infructueuses. On cite le cas de celles ayant un nombre de pixels inférieur à un certain seuil, un rapport $\frac{\text{largeur}}{\text{longueur}}$ trop petit ou encore ayant un nombre de coins inacceptable relativement à ceux de l'indice. Ceci a pour objectif essentiel l'élimination du bruit. Par exemple, le seuil utilisé à ce niveau peut être un rapport de la taille de la région relativement à celui de l'image si

cette dernière est de petite taille. S'il est inférieur à 5%, la région sera considérée comme inacceptable.

Pour chaque région détectée i , une recherche de ses coins est faite. D'abord, tous les coins de l'image courante sont recherchés par l'algorithme de Harris [22] appliqué à l'image entière. Ensuite, une fenêtre virtuelle est déplacée autour des bords de la région. Les coins détectés qui donnent les meilleurs scores et coïncident avec les coins détectés par l'algorithme de Harris seront retenus comme étant les coins effectifs.

Si l'indice est identifié durant la phase de recherche, l'augmentation de la séquence par un objet virtuel 2-D ou 3-D peut être réalisée (Figure II.14). Nous détaillons cette procédure au troisième chapitre. Notons toutefois que cette phase de recherche est très gourmande en temps de calcul.



Figure II.14. Augmentations 2-D et 3-D

5.3.3. Suivi de l'indice dans la séquence

Dans un but de réduction du temps de traitement pour la suite de la séquence, un simple tracking des coins de l'indice est fait. Ceci est dû au fait que l'objet ou la caméra peuvent se déplacer. Par conséquent, pour pouvoir faire ce suivi, nous avons besoin d'implémenter un algorithme. Un nombre très important d'algorithmes de suivi, qui ont prouvé leur efficacité, sont proposés dans la littérature. Parmi ces travaux, il existe ceux qui sont spécifiques à d'autres domaines pour le suivi d'objets rigides ou déformables en mouvement, tels que [36], [84] et [116]. Il existe également ceux qui sont spécifiques à la réalité augmentée mais qui sont généralement orientés vers des cas particuliers d'applications. Citons à titre d'exemple [3], [57], [120] ou celui de [2] dont les résultats obtenus permettent des augmentations temps réel. Tous ces algorithmes utilisent des fondements mathématiques assez robustes tels que les chaînes de Markov [25] ou le filtre de Kalman [23, 43] et, par conséquent, ils sont utilisés dans des domaines requérant une grande précision. Comme cela n'est pas l'objectif de notre travail, nous avons opté pour celui proposé dans [112] en raison de sa robustesse et de sa simplicité. En effet, le tracking escompté dans notre travail n'est pas celui d'un objet complexe. Ce sont juste des coins ayant des formes assez simples, éventuellement déformés, à suivre (Figure II.15).

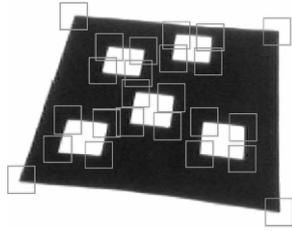


Figure II.15. Coins d'un indice à rechercher et à suivre dans l'algorithme d'augmentation

Le principe de l'algorithme de suivi implémenté consiste à prédire le mouvement des coins à partir de deux images consécutives. Les coins en déplacement sont liés par la relation:

$$x'_c = Hx_c + v_c\Delta t \quad (\text{II.44})$$

x'_c étant le coins prédit, initialement situé à la position Hx_c dans l'image précédente. La distance prédite parcourue $v_c\Delta t$ durant le temps Δt , séparant la dernière image de celle en cours de traitement, est ajoutée à Hx_c . v_c représente la vitesse de déplacement du coin dans les deux dernières images. Une fenêtre de recherche est alors établie autour des coins prédits pour y rechercher ceux effectifs. Sa taille est choisie de manière dynamique variant en fonction de la surface occupée par l'indice dans l'image.

L'identification des positions réelles des coins permet de mettre à jour l'homographie et par conséquent d'augmenter la scène. Si par contre cette recherche échoue à un moment donné, c'est-à-dire que l'algorithme ne parvient plus à identifier les coins réels, il retourne à la phase de recherche de l'indice.

5.3.4. Résumé de l'algorithme

Un résumé de l'algorithme est présenté à la Figure II.16.

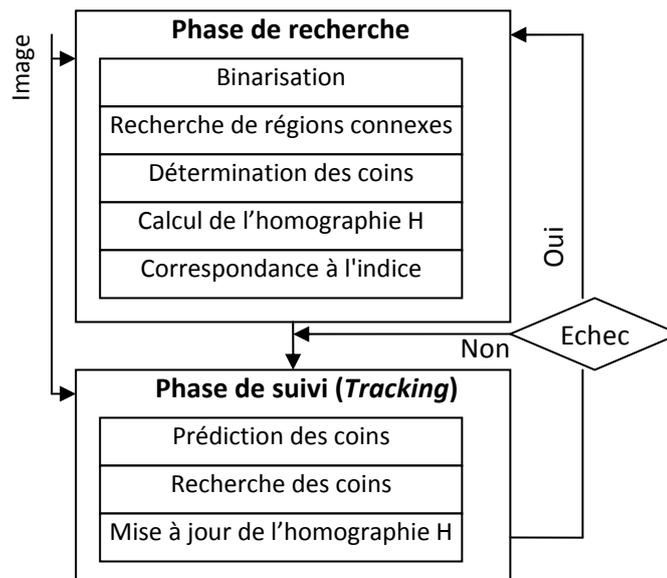


Figure II.16. Résumé de l'algorithme d'augmentation à base d'indice planaire

L'algorithme d'augmentation est composé de deux grandes phases: recherche et suivi. La phase de recherche est très gourmande en temps de calcul. La phase de suivi est très souple relativement à la première mais elle n'est pas assez robuste en face de certaines situations limites. En général, l'algorithme présente des limites que nous énumérons:

1. l'inclinaison maximale supportée de l'indice, relativement au point de vue, est limitée.
2. le simple principe de différence utilisé pour identifier l'indice à partir des régions connexes n'est pas très satisfaisant. Cependant, utiliser des méthodes plus précises consommera un temps de calcul considérable au dépend du temps réel escompté pour l'augmentation.
3. si l'éclairage est mal réparti à la surface de l'indice, la recherche peut échouer à cause de la binarisation qui ne donne pas le résultat escompté (Figure II.17). L'échec peut être total en cas de déformation complète de la région. Il peut être partiel dans le cas contraire mais il conduit parfois à des augmentations superflues (Figure II.18).

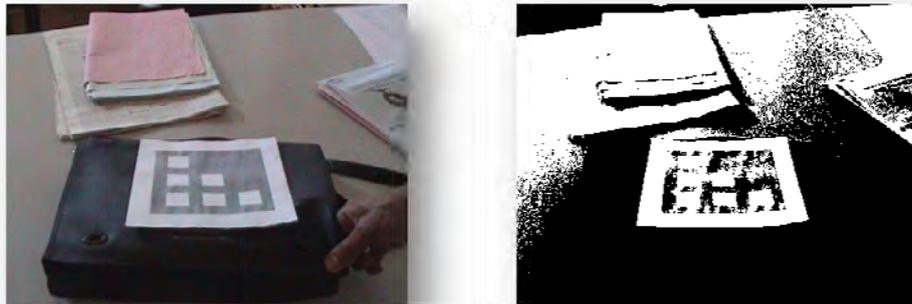


Figure II.17. Effet de l'éclairage et du choix du seuil de binarisation sur l'image binarisée

4. l'existence de plusieurs coins (bruit) autour de la région de l'indice peut mener à des augmentations insensées ou partiellement décalées (Figure II.18). Cette situation est également remarquable dans le cas d'existence de plusieurs coins dans la fenêtre de suivi d'un coin pendant la phase de tracking. Sa conséquence en est un scintillement de l'objet augmenté dans la scène finale.
5. les algorithmes de lissage et de filtrage de bruit seraient d'une grande utilité, mais ils ne s'adaptent pas généralement aux traitements temps réel.

Notons que de simples heuristiques de programmation fournissent généralement de bons résultats et contribuent dans la minimisation du temps de traitement.

Il existe des facteurs qui influent sur le temps global du processus d'augmentation et qui ont été explicitement négligés à cause de leur effet relativement réduit. Ainsi, les traitements d'augmentation peuvent prendre en compte les temps de latence variables

des outils qui composent le système. Leurs dissemblances (fiabilité, fréquence, nature, etc.) nécessitent des corrections spatiales et temporelles. En plus, l'imprécision inévitable des parties mécaniques est un autre point à considérer. Pour une étude détaillée, nous renvoyons par exemple à [75].



Figure II.18. Effet d'une fausse détection des coins de l'indice sur l'augmentation obtenue: l'objet d'augmentation ne recouvre pas complètement l'indice

6. Conclusion

En guise de conclusion, nous pouvons résumer le chapitre dans trois points essentiels. Le premier point est relatif à la description du modèle sténopé utilisé pour la caméra. Il décrit le principe à base duquel la relation entre l'image générée et l'objet virtuel à insérer est faite. Le deuxième point est associé aux différents indices utilisés dans le processus d'augmentation. Il discute les différentes formes ainsi que les avantages et inconvénients de chacune. Enfin, le dernier point lie les deux premiers points pour présenter l'algorithme d'augmentation utilisé dans notre travail. Il démontre comment, à partir des images de la séquence, il est possible d'insérer des objets virtuels 3-D en suivant un ensemble d'étapes séquentielles pour obtenir une scène augmentée.

Les bases mathématiques relatives à tout le processus concerné par ce travail sont présentées. Cependant, des renvois à des références particulières sont faits pour certains détails puisque cette thèse ne peut pas faire l'objet d'une référence complète. Indiquons finalement que malgré les inconvénients et les limites des algorithmes utilisés, ceux-ci sont largement suffisants pour nos objectifs. En effet, ceux-ci constituent juste des outils pour l'implémentation et le test de notre travail proprement dit.

Le prochain chapitre décrit comment est-ce que les objets virtuels sont d'abord projetés dans les images des séquences à augmenter. Ensuite il discute la façon dont ces objets insérés peuvent être manipulés. Un exposé de la manière d'interaction des utilisateurs avec ces objets est fait en utilisant les concepts présentés dans ce chapitre comme base.

A decorative graphic on the right side of the page. It features three blue circles of varying sizes, each composed of concentric circles with a gradient from dark blue to light blue. Two thin blue lines intersect at the top left, forming a large 'V' shape that frames the circles. The circles are positioned at the top, middle, and bottom right of the page.

Chapitre III

Manipulation des Objets Virtuels

La manipulation dans ce chapitre est présentée du point de vue des concepts théoriques ainsi que de celui des outils d'interaction.

Manipulation des Objets Virtuels

1. Introduction

Les 'Computer Supported Collaborative Work' (CSCW) étudient comment utiliser l'ordinateur comme moyen de communication humaine où la réalité virtuelle représente son aspect 3-D. La tendance actuelle des CSCW est d'adapter l'ordinateur aux outils naturels des utilisateurs [127]. La réalité augmentée, particulièrement collaborative, s'apparente comme l'une des façons de l'adapter selon cette tendance par le fait de garder l'utilisateur en contact avec le monde réel l'entourant. C'est pourquoi la manipulation des objets virtuels procure à la réalité augmentée une nouvelle dimension dépassant le cadre d'une simple application informatique, particulièrement lorsque les objets virtuels sont manipulés à travers des gestes naturels. Ainsi, différents utilisateurs peuvent communiquer naturellement tout en renforçant cette communication par le virtuel supporté. Leur collaboration est faite selon leurs habitudes naturelles et est enrichie par les objets virtuels.

La manipulation des objets virtuels consiste à simuler leur déplacement et leur rotation dans la scène augmentée suite à des requêtes utilisateurs. Les mouvements résultants devraient paraître les plus naturels possibles et avoir les mêmes apparences qu'un objet réel déplacé. Dans certaines applications de la réalité augmentée, cette manipulation peut ne pas avoir l'aspect d'un mouvement mais exprimée en tant qu'effet d'une action sur la forme ou la couleur de l'objet par exemple. Mais d'une manière générale et afin de simuler cette manipulation, la première étape du processus consiste à projeter l'objet virtuel dans une position initiale relativement à celle de l'indice détecté. Cette position est choisie à priori par l'utilisateur ou dépend de l'objectif de l'augmentation. Mathématiquement, ce positionnement est celui du référentiel de l'objet virtuel relativement à celui de l'indice. Par conséquent, tout repositionnement ultérieur consiste en celui du référentiel de l'objet virtuel par rapport à sa position initiale.

Beaucoup de méthodes d'interaction avec la scène augmentée ont été développées pour les systèmes de réalité augmentée. Elles peuvent être divisées en deux catégories [98]:

- d'une part, celles qui reprennent les tâches de base de l'interaction 3-D (sélection, manipulation, navigation, contrôle d'application [29])
- d'autre part, les environnements de haut niveau gérant une interface hybride multi-utilisateurs, multi-périphériques et multi-documents.

Pour notre cas, nous nous intéressons dans ce chapitre à la première catégorie dont l'interaction avec les objets virtuels est typée sous quatre formes [115]: navigations 3-D et 2-D, manipulation 3-D et 2-D.

Beaucoup de travaux sont dédiés à la navigation dans une scène. Une telle opération consiste en une approche de navigation dans un environnement [91]. Si la scène est virtuelle, la navigation consiste à simuler son parcours [104] (Figure III.1 gauche). Par contre, si elle est réelle, sa navigation est réelle mais elle est agrémentée par des informations sous forme d'annotations par exemple [21] (Figure III.1 droite). Les travaux sont particulièrement liés à la gestion de la caméra et au déplacement du point de vue. Dans ce contexte, la navigation est parfois considérée comme une manipulation indirecte.



Figure III.1. Exemple d'applications pour la navigation dans une scène virtuelle (gauche) et réelle (droite)

Notre travail, par contre, est particulièrement concerné par la manipulation 2-D et 3-D des objets virtuels. Cette interaction fait appel à deux volets dans l'étude. Le premier est lié à l'aspect théorique et les fondements mathématiques qu'il est nécessaire d'assoir. Le deuxième est relatif aux outils et interfaces qui peuvent être utilisés pour pratiquer cette manipulation par des utilisateurs et la façon dont sont interprétés leurs mouvements.

2. Processus de transformation

Le processus de transformation pour la projection d'un objet virtuel est analogue à la prise de photo par une caméra. Les étapes à suivre pour la génération de son image dans l'image à augmenter sont similaires à celles d'une scène complète. Elles sont formées des transformations de vue, de modélisation, de projection et de taille. La transformation de vue (viewing) est liée au positionnement de la caméra, éventuellement virtuelle, dans la scène. Celle de modélisation (modeling) correspond à la forme proprement dite de l'objet à dessiner. La transformation de projection est associée à la correspondance liant la caméra, la scène et l'image générée. Elle correspond à la partie de la scène ou de l'objet qui reçoit le focus ainsi que le zoom à appliquer dans le but de produire le détail escompté. Finalement, la taille de l'image ou de l'objet obtenue est contrôlée par la dernière transformation (viewport).

La transformation de modélisation est traitée dans un nombre de références très grand. Des bibliothèques graphiques très développées sont disponibles avec leur documentation, parfois totalement gratuites. La richesse des documents associés fait que

nous ne nous sommes pas intéressés et que nous avons considéré des objets assez simples du fait que notre objectif n'est pas directement concerné par la qualité des objets dessinés. Toutes les projections faites sont à base des pixels.

La transformation de projection a fait l'objet du chapitre précédent. Les principes sont légèrement différents d'un domaine à une autre. Le rendu d'une scène complète est différent de celui de la projection d'un objet virtuel. La scène nécessite un positionnement virtuel de la caméra à l'intérieur de la scène elle-même, alors que l'objet impose un positionnement d'une caméra virtuelle similaire à celui de la caméra réelle. C'est la raison qui a fait que la matrice de projection est estimée en réalité augmentée.

La transformation de vue est celle qui est principalement étudiée dans ce chapitre. Elle concerne les différentes positions à appliquer à l'objet virtuel suite aux manipulations. D'un point de vue géométrique, cette transformation est liée à des positionnements différents de la caméra virtuelle relativement à l'objet afin de pouvoir disposer, voir calculer, les différentes vues correspondant aux requêtes de manipulation. Un raisonnement inverse peut être mené dans le sens où c'est l'objet qui change de position. Cependant, les résultats présentés ci-après reposent tous sur la première hypothèse.

La manipulation des objets virtuels revient à leur appliquer des transformations avant leur projection. Dans ce qui suit, nous explicitons premièrement le principe de projection, ensuite nous détaillons les concepts théoriques relatifs à la manipulation.

3. Projection des objets virtuels

La projection d'un objet virtuel est simplement le collage de son image au dessus de celle de la séquence à augmenter. La différence entre un objet 2-D et un objet 3-D réside dans le fait que le premier est généralement un mappage direct des pixels de l'image alors que le second nécessite au préalable une construction 3-D. L'opération de construction la plus importante est celle du calcul du point de vue associé à l'orientation actuelle de l'objet et la détermination des parties cachées.



Figure III.2. Prise en compte de l'aspect d'ombre (droite) dans l'intégration d'objets virtuels

La qualité de l'image projetée dépend d'un certain nombre de facteurs. L'éclairage, l'ombrage et le rendu sont les propriétés les plus importantes du réalisme. L'éclairage nécessite une analyse complète de l'image. L'ombrage nécessite en plus une étude particulière des objets réels qui composent l'image ainsi que leurs ombres [108, 109] pour la détermination de la direction de la lumière (Figure III.2). Dans le cadre de notre travail, nous ne nous intéressons pas à ce type d'étude puisqu'elle constitue une orientation de recherche en soit. Les projections réalisées sont généralement faites avec des images à l'état brut sans aucune adaptation. Notre objectif étant celui de la manipulation et non celui des propriétés citées.

Le rendu de l'image est fait relativement au référentiel de l'indice (Figure III.3). En d'autres termes, lors de la première projection, l'objet virtuel est supposé généralement posé sur l'indice dans le monde réel. Sa base est collée à l'indice dont l'origine du référentiel est en son centre. La position de tout point à projeter est alors exprimée relativement à la position d'origine (0, 0, 0). Les ordres de grandeur doivent être connus et particulièrement la taille de l'indice qui est généralement exprimée en millimètre. Nous ne discutons pas ici les modes de constructions des objets graphiques, tels que ceux à base de points ou vectoriels, mais nous présentons le principe suivie pour la création des objets utilisés dans nos augmentations.

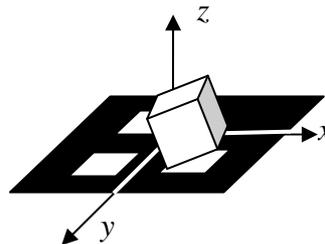


Figure III.3. Manipulation de l'objet par rapport au référentiel de l'indice

A ce niveau de la réflexion se pose le problème de proportion des tailles réelles. En effet, un objet virtuel est en fait un objet imaginaire ou réel et dans les deux cas, il est généralement construit par des outils graphiques. Dans ce cas, les points (x_w, y_w) , exprimés normalement en millimètres, doivent être convertis en pixels puisqu'en réalité on ne dispose que des images de l'indice et de l'objet à projeter. Il existe deux situations possibles:

- Soit que la calibration de la caméra est faite et par conséquent, la taille des pixels est connue. Le respect des tailles des objets virtuels passe simplement par une adaptation aux pixels de l'image générée par la caméra. En d'autres termes, les coordonnées (x_w, y_w, z_w) qui sont exprimées en millimètre seront converties pour être générés en pixels.
- Soit que la caméra n'est pas totalement calibrée - cas de l'homographie pour les projections 2-D - et dans cette situation les proportions des tailles réelles par rapport aux objets à projeter sont considérées. En effet, les proportions des

images ne reflétant en aucun cas les proportions réelles, la proportion des tailles réelles est d'abord exprimée (III.1):

$$P_t = \frac{T_i}{T_o} \quad (\text{III.1})$$

où T_i est la taille de l'indice et T_o celle de l'objet. Cette proportion est utilisée avant la projection de l'image de l'objet dans celle de l'image à augmenter.

Pour garantir une qualité acceptable, la projection fait appel généralement à des moteurs graphiques assez puissants et même à des accélérateurs graphiques matériels. OpenGL est un des moteurs graphiques les plus utilisés puisqu'il offre des versions libres et est assez puissant. Des primitives de base de traitement d'image sont implicitement utilisées dans la projection et sont implémentées par certaines architectures du matériel pour compenser la qualité de l'image obtenue. Les points sont manipulés en tant que vertex. C'est en fait les pixels considérés dans l'espace 3-D ayant des coordonnées homogènes et des propriétés particulières. En l'occurrence, les surfaces des objets sont approximées en faisant appel par exemple aux constructions à base de triangles.

3.1. Projection 2-D

Les travaux correspondant à la projection d'une simple image exploitent une projection 2-D vers 2-D. La majorité d'entre eux ne sont pas spécifiques au domaine de la réalité augmentée. On trouve par exemple ceux liés aux mappages de textures 2-D sur des surfaces [1, 66] mais ayant une orientation de recherche particulière. Ils traitent particulièrement les problèmes liés aux surfaces non planes. Pour notre cas, nous nous intéressons essentiellement aux cas spécifiques à notre domaine de travail. En l'occurrence, le mappage d'images 2-D sur des surfaces planes.

Le mappage sur des surfaces planes ne nécessite aucune calibration de la caméra. L'homographie de correspondance entre quatre points de l'espace et quatre points de l'image est amplement suffisante. En effet, connaissant les coordonnées des quatre coins de l'indice et leurs correspondants dans l'image, l'homographie H peut être calculée (voir deuxième chapitre). L'objet virtuel - en tant qu'image - étant supposé agglutiné à cet indice et ayant l'origine du référentiel de l'indice positionné en son centre, chacun de ses points est alors projeté dans l'image selon l'expression (III.2):

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = \lambda H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad (\text{III.2})$$

Le pixel correspondant est dessiné à la position (III.3):

$$\left(\frac{u_p}{w_p}, \frac{v_p}{w_p} \right) \quad (\text{III.3})$$

Pour garantir une proportion visuelle acceptable, la proportion P_t (III.1) est utilisée pour redimensionner l'image de l'objet avant sa projection. Ainsi, l'image de l'objet est

directement projetée dans l'image à augmenter en ayant les pixels comme unité de mesure. De cette façon, les proportions réelles des objets virtuels relativement à l'ensemble de tous les objets réels seront visuellement maintenues (Figure II.10). Notons que d'un point de vue pratique, un agrandissement de l'image de l'objet peut entraîner sa dégradation. C'est pourquoi, il est toujours souhaitable de disposer d'images de haute résolution pour les objets virtuels à projeter.

3.2. Projection 3-D

La projection d'un objet virtuel 3-D est similaire à un objet 2-D en ce sens que la construction de l'objet est faite d'abord dans un plan image virtuel en arrière plan avant d'être projetée. La différence réside dans la façon dont est générée cette image. La projection 2-D est faite directement, pixel par pixel, puisque l'image à projeter est déjà construite. La projection 3-D, par contre, crée des trous vides dans l'image générée qu'il est indispensable de combler. Cela est dû à la vue perspective qui projette des points différents au même endroit et laisse des endroits vides sans correspondance. Un traitement spécifique est alors nécessaire pour combler ces trous créés. De plus, la troisième coordonnée z de tout point doit être considérée pour la prise en compte des parties cachées. Ainsi, un seul de deux points ayant la même position dans l'image générée doit être affiché et il correspond à celui qui possède la valeur de z la plus proche de la caméra virtuelle. Pour cela, tout point (x_w, y_w, z_w) de l'objet est projeté en (III.4):

$$\begin{bmatrix} u_p \\ v_p \\ w_p \end{bmatrix} = M_{int} M_{ext} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad \text{(III.4)}$$

Il est évident que la première projection ne nécessite aucune détermination des parties cachées puisque c'est le z_w initial de tout point qui définit directement si ce dernier doit être projeté ou non, i.e. s'il est caché ou non. Les tests sont faits lors de la projection d'autres vues de l'objet par suite aux manipulations de l'utilisateur qui induisent des rotations de l'objet.

4. Manipulation des objets virtuels

La manipulation des objets virtuels consiste à les afficher à des endroits différents de la scène augmentée et dans des orientations différentes en réponse aux requêtes utilisateur. L'intérêt d'une telle manipulation est de donner la possibilité de placer les objets virtuels à un endroit dépendant du besoin de l'utilisateur et non à une position prédéfinie à l'avance. Les problèmes à résoudre sont liés aux projections des différentes faces qui doivent respecter le point de vue de la caméra.

Pratiquement, l'indice inséré dans la scène est utilisé comme référence pour la première incrustation. Sa position détectée dans la scène joue le rôle de référence de base pour les prochains déplacements de l'objet. En ce sens, pour simuler un objet qui est déplacé vers l'arrière ou vers l'avant par exemple, sa taille doit visuellement changer afin de simuler

l'effet de zoom résultant de la perspective. La simulation d'une translation vers la gauche ou la droite doit également faire changer la face apparente, en d'autres termes, c'est la face associée au nouveau point de vue qui doit être affichée. Ainsi, simuler une manipulation d'un objet virtuel revient à jouer sur son apparence visuelle.

Mathématiquement, simuler la manipulation d'un objet virtuel revient à lui faire subir des transformations géométriques. Ces transformations sont respectivement des translations et des rotations. Elles sont appliquées, en fonction des requêtes utilisateurs, relativement au système de coordonnées de l'indice ou à celui de l'objet. Les transformations relatives au système de coordonnées de l'indice sont celles qui permettent de disposer l'objet à n'importe quel endroit de la scène relativement à celui de l'indice. Ce sont principalement des translations, puisque les rotations n'ont pas d'intérêt pratique (Figure III.4). Celles relatives au référentiel de l'objet lui-même sont généralement limitées à des rotations et agissent sur les points de vue de l'objet.

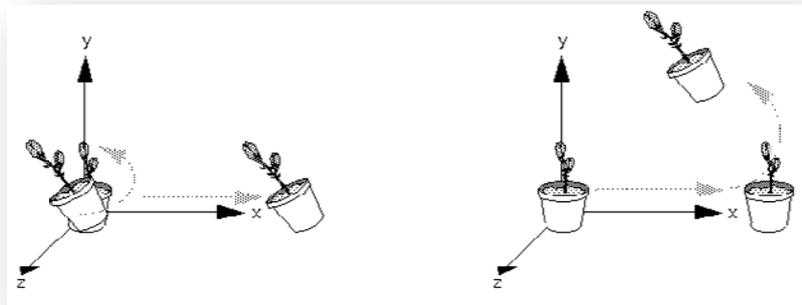


Figure III.4. Différence entre une rotation de l'objet par rapport à un référentiel suivie d'une translation (gauche) et d'une translation suivie par une rotation (droite)

Ces principes sont basés sur les concepts liés aux travaux relatifs à la vision par ordinateur. Pratiquement, appliquer une transformation à un objet revient à appliquer celle-ci sur chaque point représentatif de son image. En particulier, toute transformation est la composition d'une rotation R et/ou d'une translation T . Toute rotation transforme tout point $p(x, y, z)$ de l'objet en un autre $p_r(x_r, y_r, z_r)$ (III.5):

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{(III.5)}$$

ou en coordonnées homogènes (III.6):

$$\begin{bmatrix} x_r \\ y_r \\ z_r \\ w_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{(III.6)}$$

et toute translation le transforme au point $p_t(x_t, y_t, z_t)$ (III.7):

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \quad (\text{III.7})$$

4.1. Manipulation 2-D

Un objet 2-D est par définition un objet plat agglutiné à l'indice. A cet effet, sa translation suivant l'axe Oz n'est pas possible sur le plan pratique et n'a pas de sens du fait que pour le calcul de la matrice H , il est supposé que $z = 0$. Les rotations suivant les axes Ox et Oy sont également possibles mais insensées. En effet, elles consistent simplement à changer la taille de l'objet suivant ces axes au lieu de lui faire subir de vraies rotations.

Les rotations effectuées sont appliquées relativement au centre de l'objet. Ainsi, son centre est d'abord superposé au centre de l'indice, ensuite les transformations géométriques requises sont appliquées. Les transformations suivantes peuvent être appliquées pour chaque pixel (x_i, y_i) d'un objet 2-D (III.8) et (III.9):

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_i - x_0 \\ y_i - y_0 \end{bmatrix} \quad (\text{III.8})$$

$$\begin{bmatrix} x''_i \\ y''_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (\text{III.9})$$

où (x_0, y_0) représente le décalage du repère de l'objet par rapport à l'origine du système de l'indice, α représente l'angle de la rotation appliquée relativement à l'axe Oz , $[t_x \ t_y]^T$ la translation à effectuer et (x'_i, y'_i) ou (x''_i, y''_i) les nouvelles positions calculées du point suite respectivement à une rotation ou à une translation (Figure III.5).

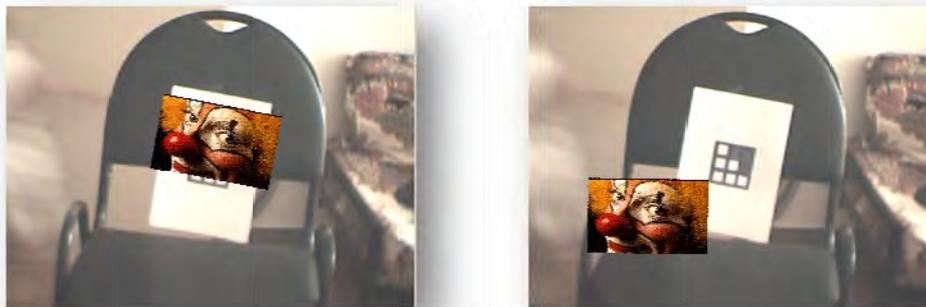


Figure III.5. Exemple de translation et rotation 2-D d'un objet 2-D

4.2. Manipulation 3-D: Fondements mathématiques

La manipulation 3-D permet un déplacement naturel de l'objet. Ce dernier peut être translaté ou tourné par l'utilisateur suivant les trois axes Ox , Oy ou Oz des systèmes de coordonnées de l'objet lui même ou de l'indice. Pour obtenir les augmentations correspondantes aux transformations choisies, la nouvelle position, de chaque point constituant l'objet dans l'image, est d'abord déterminée en utilisant les paramètres

calculés lors du processus de calibration en plus de ceux relatifs aux requêtes utilisateur, ensuite le pixel relatif est inséré.

Toute manipulation 3-D, comme pour celle 2-D, peut généralement être décomposée en une translation et une rotation. La translation est évidente. Elle consiste à traduire chaque point de l'objet suivant l'un des trois axes du référentiel de l'objet ou de l'indice. Normalement, les zooms résultant sont directement obtenus suite à l'opération de projection. La rotation de l'objet est généralement faite relativement à son propre référentiel. Avant la projection de l'objet, les parties cachées résultant de la rotation sont simplement omises.

4.2.1. Matrice de transformation

Dans l'une ou l'autre des transformations 2-D ou 3-D, une matrice de rotation R et une autre de translation T sont calculées. Les matrices calculées à une étape sont soit appliquées à l'objet déjà transformé par les actions précédentes, soit cumulées aux précédentes pour obtenir une matrice résultante à appliquer pour l'objet de référence. Afin de simplifier ces opérations, une matrice de transformation est généralement utilisée. Elle consiste en le produit $T_r = R.T$ (III.10) avec $T = [t_1 \ t_2 \ t_3 \ 1]^T$ en format homogène:

$$T_r = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{11}t_1 + r_{12}t_2 + r_{13}t_3 \\ r_{21} & r_{22} & r_{23} & r_{21}t_1 + r_{22}t_2 + r_{23}t_3 \\ r_{31} & r_{32} & r_{33} & r_{31}t_1 + r_{32}t_2 + r_{33}t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.10})$$

Quant à la matrice rotation, il existe plusieurs méthodes pour déterminer ses éléments et plusieurs façons d'interpréter leurs valeurs au sens de la rotation. La plus simple considère que toute rotation à appliquer à l'objet est une composition de trois rotations consécutives suivant les trois axes du référentiel. Ces trois matrices R_x , R_y et R_z , qui expriment respectivement les rotations autour des axes Ox , Oy et Oz d'un référentiel droit avec des angles α , θ et φ , sont exprimées comme suit (III.11):

$$\begin{aligned} R_x &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ R_y &= \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ R_z &= \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{III.11})$$

La matrice résultante est $R = R_z R_y R_x$. Si une matrice est réduite en la matrice unité, elle exprime une rotation nulle autour de l'axe correspondant.

Notons que la matrice R implique une rotation autour de l'axe Ox en premier, ensuite autour de l'axe Oy et enfin autour de l'axe Oz . Toute autre rotation nécessitant un ordre différent implique une multiplication des matrices dans l'ordre adéquat, en particulier la première la plus à gauche et la dernière la plus à droite. Cet aspect est dû au fait que la multiplication matricielle n'est pas commutative.

4.2.2. Rotation autour d'un axe quelconque

La rotation de l'objet autour des axes principaux du référentiel, afin de l'orienter dans une position donnée, n'est pas pratique et est difficile en général à trouver pour simuler des rotations naturelles. Certaines situations imposent des rotations autour d'un axe quelconque. Pour réaliser une telle pratique, des méthodes autres que celle du produit matricielle sont utilisées. Citons, à titre d'exemples, la construction directe de la matrice rotation, les angles d'Euler et les Quaternions. Nous nous intéressons à la première méthode dans cet exposé puisqu'elle fut celle utilisée.

Il existe deux méthodes différentes dans la construction directe de la matrice rotation qui fait tourner un objet autour d'un axe quelconque. La première méthode consiste à arranger les axes de l'objet avec l'axe de rotation en question, ensuite appliquer la rotation proprement dite et enfin replacer les axes à leur position d'origine. La deuxième méthode part de la position actuelle de l'objet et construit les éléments de la matrice rotation qui lui permet de le faire tourner à la position cible. A la différence de la première, cette dernière construit les éléments de la matrice à base de calcul analytique et non de produit matriciel (voir [30] pour le détail de cette deuxième méthode, par exemple). Les deux méthodes étant équivalentes, par la suite nous détaillons uniquement la première méthode puisque c'est celle qui a été utilisée dans les résultats obtenus dans nos travaux.

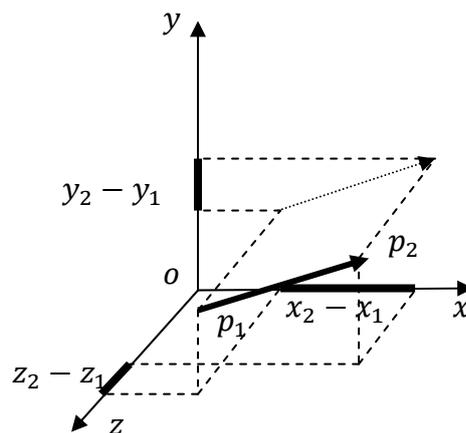


Figure III.6. Exemple d'axe de rotation quelconque dans l'espace

Supposons que l'axe de rotation est défini par deux points de l'espace $p_1(x_1, y_1, z_1)$ et $p_2(x_2, y_2, z_2)$ (Figure III.6). Il est possible de calculer le vecteur directeur \vec{u} associé à cet axe ayant pour coordonnées (a, b, c) par la normalisation du vecteur $\overrightarrow{p_1 p_2}$. Le vecteur \vec{u} aura pour coordonnées (III.12):

$$\left(\frac{x_2 - x_1}{|\vec{p}_1 \vec{p}_2|}, \frac{y_2 - y_1}{|\vec{p}_1 \vec{p}_2|}, \frac{z_2 - z_1}{|\vec{p}_1 \vec{p}_2|} \right) \quad (\text{III.12})$$

avec:

$$|\vec{p}_1 \vec{p}_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (\text{III.13})$$

Notons \vec{u}_v la composante de \vec{u} suivant un axe $Ov = (Ox, Oy \text{ ou } Oz)$ et \vec{u}_{vw} sa projection sur un plan $(vw) = (xy, xz \text{ ou } yz)$. En particulier, sa projection sur le plan (yz) a pour module $|\vec{u}_{yz}| = d = \sqrt{b^2 + c^2}$. Si elle est nulle, toute la rotation est réduite à celle autour de l'axe Ox . Dans le cas contraire, avant de procéder à la rotation de l'objet d'un angle β autour de l'axe \vec{u} , il faut faire une rotation du système de coordonnées de l'objet autour de l'axe Oy , de sorte que l'axe de rotation repose sur le plan (yz) , suivi d'une autre rotation autour de l'axe Ox afin qu'il soit colinéaire avec l'axe Oz .

Cette dernière rotation est en fait autour de l'axe Ox d'un angle égal à l'angle fait entre la projection de l'axe de rotation sur le plan (yz) , soit \vec{u}_{yz} , et l'axe Oz ou simplement \vec{u}_z . Sachant que le produit scalaire de \vec{u}_{yz} et \vec{u}_z est (III.14):

$$\vec{u}_{yz} \cdot \vec{u}_z = |\vec{u}_{yz}| |\vec{u}_z| \cos(\widehat{\vec{u}_{yz}, \vec{u}_z}) = 0.0 + b.0 + c.c \quad (\text{III.14})$$

d'où l'on peut déduire (III.15):

$$\cos(\widehat{\vec{u}_{yz}, \vec{u}_z}) = \frac{c^2}{c \cdot \sqrt{b^2 + c^2}} = \frac{c}{d} \quad (\text{III.15})$$

De même, on sait que la norme du produit vectoriel de ces mêmes vecteurs est (III.16):

$$|\vec{u}_{yz} \wedge \vec{u}_z| = |\vec{u}_{yz}| |\vec{u}_z| \sin(\widehat{\vec{u}_{yz}, \vec{u}_z}) \quad (\text{III.16})$$

or:

$$\vec{u}_{yz} \wedge \vec{u}_z = \begin{vmatrix} b & c \\ 0 & c \end{vmatrix} \vec{x} - \begin{vmatrix} 0 & c \\ 0 & c \end{vmatrix} \vec{y} + \begin{vmatrix} 0 & b \\ 0 & 0 \end{vmatrix} \vec{z} = bc\vec{x} \quad (\text{III.17})$$

où \vec{x} , \vec{y} et \vec{z} représentent les vecteurs directeurs respectifs des axes Ox , Oy et Oz . D'où l'expression du sinus de l'angle (III.18):

$$\sin(\widehat{\vec{u}_{yz}, \vec{u}_z}) = \frac{bc}{c \cdot \sqrt{b^2 + c^2}} = \frac{b}{d} \quad (\text{III.18})$$

Ainsi la matrice rotation à appliquer relativement à l'axe Ox est définie par (III.19):

$$R_{x\vec{u}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.19})$$

Suivant le même raisonnement que le précédent, mais cette fois pour une rotation autour de l'axe Oy pour que l'axe de rotation repose sur le plan (yz) , il est possible de calculer le produit scalaire (III.20):

$$\vec{u} \cdot \vec{u}_{yz} = |\vec{u}| |\vec{u}_{yz}| \cos(\widehat{\vec{u}, \vec{u}_{yz}}) = a \cdot 0 + b \cdot b + c \cdot c \quad \text{(III.20)}$$

et le cosinus sera donc (III.21):

$$\cos(\widehat{\vec{u}_{xz}, \vec{u}_z}) = \frac{d^2}{d} = d \quad \text{(III.21)}$$

De même que:

$$\vec{u} \wedge \vec{u}_{yz} = \begin{vmatrix} b & c \\ b & c \end{vmatrix} \vec{x} - \begin{vmatrix} a & c \\ 0 & c \end{vmatrix} \vec{y} + \begin{vmatrix} a & b \\ 0 & b \end{vmatrix} \vec{z} = -ac\vec{y} + ab\vec{z} \quad \text{(III.22)}$$

et donc le sinus de l'angle est (III.23):

$$\sin(\widehat{\vec{u}_{xz}, \vec{u}_z}) = \frac{ad}{d} = a \quad \text{(III.23)}$$

La matrice qui permet la rotation de l'axe \vec{u} autour de l'axe Oy est définie par (III.24):

$$R_{y\vec{u}} = \begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(III.24)}$$

Ainsi, en appliquant ces deux rotations, $R_{x\vec{u}}$ et $R_{y\vec{u}}$, à l'axe de rotation, ce dernier devient colinéaire avec l'axe Oz . Par conséquent, une rotation autour de l'axe Oz équivaut à celle autour de l'axe \vec{u} puisque la rotation précédente a été appliquée en réalité à tout le référentiel de l'objet. Elle sera donc définie ainsi (III.25):

$$R_{z\vec{u}} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & 0 \\ \sin(\beta) & \cos(\beta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(III.25)}$$

Comme le référentiel complet se serait tourné en raison de cette rotation composée, il est indispensable de le faire retourner à sa position initiale. Il est donc nécessaire d'appliquer des rotations inverses à $R_{x\vec{u}}$ et $R_{y\vec{u}}$. En considérant les propriétés des matrices rotations qui établissent que l'inverse de toute matrice rotation est simplement sa transposée, nous pouvons établir (III.26) et (III.27) tels que:

$$R_{x\vec{u}}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(III.26)}$$

et

$$R_{y_{\bar{u}}}^{-1} = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.27})$$

Finalement, la rotation complète de l'objet d'un angle β autour d'un axe quelconque est obtenue par l'application de la matrice rotation R_β à chaque point $p(x, y, z)$ de ce dernier. En notation matricielle, R_β est définie par (III.28):

$$R_\beta = R_{x_{\bar{u}}}^{-1} \cdot R_{y_{\bar{u}}}^{-1} \cdot R_{z_{\bar{u}}} \cdot R_{y_{\bar{u}}} \cdot R_{x_{\bar{u}}} \quad (\text{III.28})$$

Après application de cette rotation à l'objet, tout point $p(x, y, z)$ aura effectué une rotation telle que (III.29):

$$\begin{bmatrix} x_r \\ y_r \\ z_r \\ w_r \end{bmatrix} = R_\beta \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{III.29})$$



Figure III.7. Exemple de translation et rotation 3-D d'objets 3-D

4.2.3. Décomposition d'une transformation

La manipulation d'un objet est généralement une transformation complexe requérant une décomposition sous forme d'une translation et d'une rotation (Figure III.7). Cette approche est due au fait que c'est la matrice de transformation T_r qui est utilisée dans les calculs. Autrement dit, chaque action de l'utilisateur sur l'objet est généralement une combinaison d'une rotation et d'une translation qui ne sont pas appliquées de manière isolée, mais sous forme d'une matrice de transformation unique.

Il est parfois nécessaire d'extraire les matrices rotation et translation. L'extraction de la matrice rotation à partir de la matrice de transformation est assez évidente. C'est simplement la matrice de transformation sans sa dernière colonne. Cette dernière est remplacée par le vecteur $[0 \ 0 \ 0 \ 1]^T$.

L'extraction de la matrice de translation est un peu particulière. Sachant que toute matrice de rotation possède la propriété suivante (III.30):

$$R^T . R = I \quad \text{(III.30)}$$

avec I la matrice identité, la matrice de translation est obtenue simplement par (III.31):

$$R^T . R . T = R^T T_r = T \quad \text{(III.31)}$$

En conséquence, tout point de l'objet est d'abord transformé par T_r en sa nouvelle position, ensuite projeté dans l'image à augmenter en utilisant la matrice de projection.

Notons que des transformations successives font que les éléments calculés de la matrice ne respectent plus les propriétés des matrices rotations à cause des erreurs de précision cumulées. Pour les corriger, une ré-estimation des ces valeurs est nécessaire. Une méthode de correction est présentée dans [133].

5. Outils et interfaces de manipulation

La manipulation des objets virtuels nécessite des outils d'interaction (Figure III.8). Les moyens les plus simples et les plus intuitifs sont le clavier, la souris et la manette de jeux. Des techniques spécifiques ont été introduites telle que le PIP (Personal Interaction Panel) [132], les tangibles user interface [37, 61] - qui consistent en l'utilisation d'éléments réels pour l'interaction avec des éléments virtuels - ou le système Real Reality [123]. Les plus simples des outils étant les indices eux-mêmes qui peuvent être manipulés par les utilisateurs. L'objet projeté suit alors automatiquement la position et l'orientation données au marqueur.

Des techniques de réalité virtuelle ont également été reprises telles que le WIM (World In Miniature) [102], ou les widgets 3D [110]. Actuellement, les méthodes les plus utilisées sont celles manipulant des outils dont les positions 3-D sont déterminées par des traqueurs magnétiques [99]. C'est un principe qui reflète un certain réalisme où l'utilisateur acquiert un sentiment d'interaction réel.

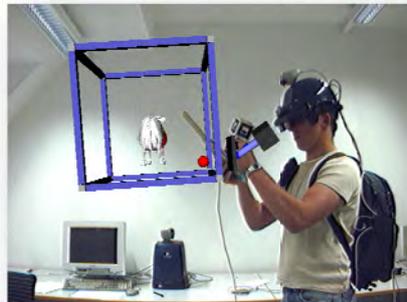


Figure III.8. Manipulation d'un objet virtuel 3-D via un outil tangible détecté par vision

Ainsi, la manipulation des objets virtuels repose sur un certain nombre d'outils assez variés qui tirent leur origine parfois de la réalité virtuelle. Dans le contexte de la collaboration, certains outils sont applicables exclusivement aux situations imposant une

coprésence des utilisateurs. D'autres, par contre, sont utilisés dans des situations générales où les utilisateurs peuvent être distants. A cause de cette dernière situation, ces outils présentent généralement des limites de manipulation.

Les interfaces utilisateurs varient de simples outils, tels que les fenêtres et les icônes, jusqu'à ceux non traditionnels. Les traqueurs magnétiques, les outils de pointage 3-D et les gants permettant des entrées gestuelles sont des exemples. Ainsi, il n'est pas opportun de les énumérer au cas par cas en raison de leur variété qui est spécifique pour chaque application, mais il est important de les classer en différentes approches. Particulièrement, on distingue trois approches différentes ([98] cite quatre):

- techniques classiques de bureau ou de réalité virtuelle tels que les pointeurs et les manettes 3-D;
- interaction digitale et gestuelle tels que les gants et les tables digitales;
- interfaces utilisateurs tangibles. Elles sont basées sur les capteurs magnétiques, sonores, la vision, etc.

5.1. Caractéristiques des outils de manipulation

Une distinction doit être faite entre les outils d'entrée et les techniques d'interaction. Généralement, un outil d'entrée supporte différentes techniques d'interaction dépendant du nombre de degrés de liberté supportés et des types d'évènements générés [28]. Les évènements peuvent être discrets, flux continus ou hybrides. Un outil à évènements discrets dispose généralement de boutons qui ne permettent de générer qu'un seul évènement à la fois. A l'inverse, les outils générant des flux d'évènements continus permettent en particulier des positionnements et des orientations. C'est le cas des gants où les courbatures des angles des doigts peuvent être contrôlées. La souris circulaire (mouse ring) est considérée comme un outil générant des évènements hybrides. Elle peut épouser les doigts de l'utilisateur en combinant un suivi ultrasonique avec deux boutons et une table à base de stylo.

Les techniques de manipulation devraient fournir au minimum les moyens pour accomplir l'une des trois tâches de sélection, de positionnement et de rotation de l'objet. Ce sont les trois opérations de base de manipulation directe d'objets réels par une main dans un environnement physique réel. Ainsi, les outils offrant de telles opérations devraient les offrir d'une façon adaptée à la nature physique de chaque outil. Il existe une multitude de d'outils [45] (Figure III.9, en haut à gauche), mais l'approche la plus classique des techniques de manipulation repose sur la simulation de la manipulation d'une main virtuelle ou une image similaire qui, elle, agit sur les objets virtuels. La main peut être manipulée par un outil quelconque tel qu'une souris ou un outil spécialisé tel que des capteurs reflétant les mouvements d'une main réelle. Un des principes de manipulation consiste à sélectionner l'objet avec la main virtuelle, ensuite positionner et orienter cette main [74] (Figure III.9, en haut à droite). Une autre technique utilise un principe de rayon virtuel généré par la main virtuelle (Figure III.9, en bas). Lorsque ce rayon traverse un objet, ce dernier peut être pris et manipulé [59].

Puisque généralement toutes les techniques présentent des points forts et des points faibles, certains travaux tentent de mixer plusieurs d'entre elles. En effet, il n'existe pas de technique pouvant être considérée comme étant la meilleure. La performance de chacune dépend de la tâche et de l'environnement auxquels elle est dédiée. Parfois, des techniques inventées présentent des performances meilleures que celles dont le principe est basé sur l'interaction naturelle [28]. C'est pourquoi les recherches prennent de l'ampleur dans le domaine de l'évaluation et de la quantification des performances des techniques d'interaction. [59] cite une catégorisation des techniques d'interaction dans un but d'évaluation de la grande variété des tâches. Trois catégories sont identifiées: le contrôle du mouvement de point de vue qui se réfère au positionnement et à l'orientation du point de vue de l'utilisateur dans l'environnement, la sélection et la manipulation. Parfois, ces deux dernières tâches sont considérées comme étant une seule catégorie de tâches.

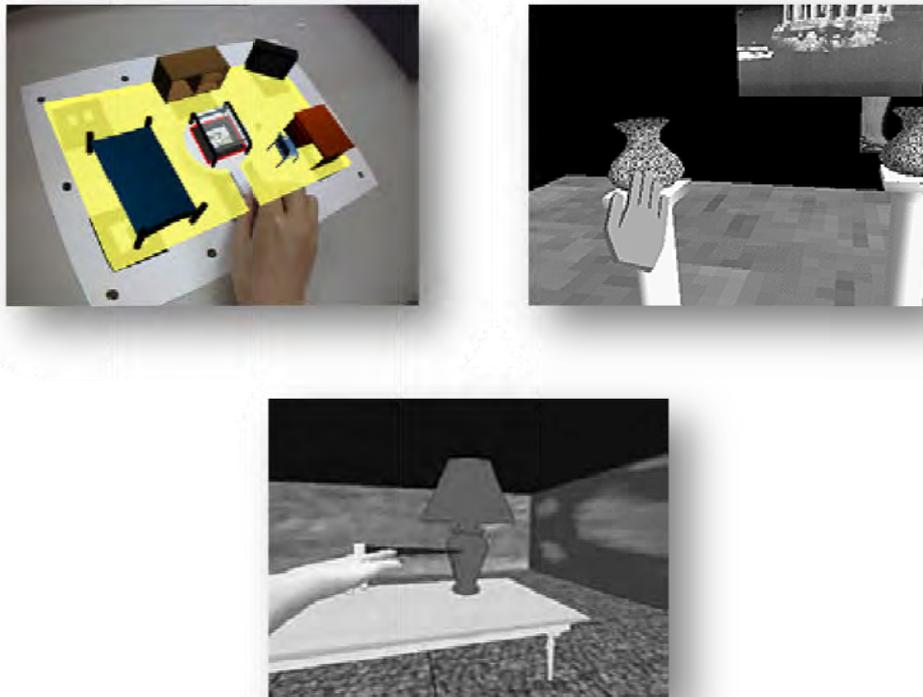


Figure III.9. Différentes techniques utilisées dans la sélection et la manipulation d'objets virtuels 3-D

Bien que le temps soit généralement le facteur quantitatif le plus utilisé dans la mesure de la performance en informatique, il n'est pas le seul utilisé dans ce contexte. La rapidité d'exécution d'une tâche par tel ou tel outil ainsi que son exactitude sont des critères importants à considérer. De même, dans le domaine des outils d'interaction, la mesure de la performance prend également en considération la facilité d'utilisation, la facilité d'apprentissage ainsi que le confort de l'utilisateur. Bien plus que beaucoup d'autres domaines, la réalité virtuelle et la réalité augmentée considèrent que l'utilisateur est impliqué dans son interaction pour la réalisation de sa tâche non seulement par ses

sens mais aussi avec son corps. Une étude complète de mesure de performance peut être trouvée dans [59].

La majorité des applications de réalité augmentée utilisent des augmentations avec des objets virtuels 3-D dans des environnements qui nécessitent des interactions 3-D. En pratique, les interactions 2-D offrent parfois différents avantages pour certaines tâches relativement à celles 3-D. Ceci est vrai particulièrement lorsqu'un retour sensoriel est important. C'est le cas de surfaces physiques offrant par exemple la possibilité de créer des annotations [107]. La combinaison des techniques d'interaction 2-D et 3-D est donc importante. Elle permet de tirer profit des avantages des deux. Les applications requérant une interaction 3-D offrent alors une manipulation facile et intuitive (Figure III.10 et I.10).



Figure III.10. Les techniques d'interaction nécessitent généralement des retours sensoriels

5.2. Techniques et outils utilisés

L'objectif de notre travail est la manipulation des objets non pas d'un point de vue des actions mais d'un point de vue de l'affichage. En d'autres termes, notre objectif est de déterminer les vues associées à chaque action utilisateur et non pas la façon dont sont exprimées les actions. De plus, pour des tâches nécessitant une manipulation complexe, un outil supportant tous les degrés de liberté et aussi spécialisé serait plus approprié. L'inconvénient de tels outils est alors leur accessibilité limitée à cause de leur coût. Aussi, il est très remarquable que les outils 2-D traditionnels restent de loin les plus dominants. Ils sont plus accessibles et largement disponibles. Pour ces raisons, nous avons opté pour les outils universels de manipulation 2-D les plus simples bien que dépourvus de sentiments de manipulation naturelle, en l'occurrence le clavier et la souris. Le but essentiel étant de fournir à l'utilisateur une méthode robuste pour des translations et des rotations complètes d'un objet 3-D. Ainsi, il est important de concevoir une manipulation 3-D à base de tels outils 2-D.

Le clavier est un outil offrant une manipulation discrète. Chaque requête utilisateur correspond à une et une seule opération géométrique. En ce sens, à chaque appui sur une touche, une seule translation ou une seule rotation sont appliquées à la fois. C'est-à-dire que l'utilisateur doit explicitement décomposer sa requête en un ensemble d'actions

élémentaires. Par exemple, pour déplacer un objet suivant une direction donnée, l'utilisateur doit spécifier les déplacements suivant les trois axes principaux qui résultent sur le déplacement voulu.

L'utilisation de la souris offre une possibilité de manipulation beaucoup plus riche que le clavier. C'est une interface qui a eu un grand succès pour la manipulation relativement aux deux axes Ox et Oy , séparés ou couplés. Cependant, l'extension de son utilisation pour s'adapter à la dimension 3-D reste délicate. Plusieurs méthodes existent en pratique à base desquelles la souris peut agir sur l'objet. Le travail présenté dans [76] décrit et évalue quatre contrôleurs virtuels associés à un objet 3-D. De ce fait, l'utilisateur est sensé adapter son esprit et ses actions conformément à la structure du contrôleur. [76] conclut que la sphère virtuelle est le meilleur contrôleur des quatre en étant plus naturel que les autres. En suivant cette conclusion, nous avons opté, dans la l'implémentation de notre travail, pour l'utilisation de la sphère virtuelle à base d'une souris.

Les déplacements de la souris permettent d'obtenir un flux d'évènements continu. Comme la souris dispose généralement d'un bouton ou plus, ajouté à sa combinaison avec le clavier, elle permet de contrôler l'objet d'avantage. Sa restriction est qu'il n'est pas possible d'obtenir des rotations combinées avec des translations. Par contre, la translation seule de l'objet ou sa rotation seule sont possibles dans n'importe quelle direction.

La translation de l'objet dans la scène est évidente, particulièrement suivant les axes Ox et Oy . La translation suivant l'axe Oz peut simplement être obtenue par une combinaison avec une touche particulière. Ce qui est difficile à obtenir est la translation suivant un axe quelconque qui n'appartient pas au plan (xy) . Dans ce cas, il faudrait le définir explicitement où chaque application fournit des outils spécifiques à cette opération. Dans le cas contraire, une décomposition explicite de la translation de la part de l'utilisateur est nécessaire.

Les rotations élémentaires suivant les axes Ox et Oy sont possibles directement par le déplacement de la souris combiné avec un bouton ou une touche. L'angle de rotation à appliquer à l'objet est simplement proportionnel à la distance parcourue par le curseur de la souris. La proportion est évaluée à la base des distances estimées de la scène après calibration de la caméra. Ainsi, la distance relative aux mouvements droite-et-gauche et haut-et-bas est utilisée pour faire tourner l'objet respectivement et de manière proportionnelle autour de Oy et Ox dans l'écran.

La rotation autour d'un axe quelconque dans le plan (xy) est également possible par le déplacement de la souris dans un sens diagonal. Ce qui est délicat, c'est la rotation de l'objet autour de l'axe Oz (le sens des aiguilles d'une montre et le sens inverse). Certains systèmes offrent la possibilité de manipuler trois vues différentes, chacune associée à un des plans (xy) , (xz) et (yz) et ainsi couvrir toutes les rotations possibles dans l'espace 3-D. Malgré la robustesse de cette solution, elle présente un aspect non naturel comme

si l'objet est manipulé en 2-D. La solution pour laquelle nous avons opté, présentée à la section suivante, permet des rotations 3-D réelles sans pour autant changer de plan.

Conséquemment, la difficulté d'une solution complète au problème de rotations 3-D réside dans les rotations libres autour d'axes quelconques, y compris l'axe Oz . Bien que pratiquement peu de personnes soient capables de faire des raisonnements sur des modèles de rotations dans d'autres plans autres que (xy) , moins encore dans l'espace (xyz) , la solution est importante. Le problème associé à ceci est double. Celui de l'utilisateur qui se trouve obligé d'interpréter, voir imaginer, les translations de la souris comme étant des rotations de l'objet puisque les actions physiques agissent sur la sphère virtuelle. Dans ce cas, la solution est simplement l'apprentissage et l'habitude de l'utilisateur. L'expérience et la familiarité sont des facteurs importants. Le second problème est la détermination de l'axe de rotation voulu ainsi que la valeur de l'angle relatif. Ce dernier étant facilement évalué par la proportion de déplacement effectuée, l'axe de rotation étant le seul paramètre qui reste à déterminer.

5.3. Emulation de la manipulation 3-D par un outil 2-D

Un aspect important des applications de graphique 3-D est la possibilité de faire tourner les objets de sorte qu'ils puissent être vus de plusieurs cotés. Un des principes repose sur la simulation de la manipulation d'une sphère virtuelle en tant que contrôleur, comme précisé précédemment, ayant son centre coïncidant avec celui de l'objet et englobant ce dernier. Elle est supposée être comme étant une sphère en verre et transparente qui peut être tournée arbitrairement autour de son centre. Ce système virtuel constitue ainsi un contrôleur qui permet de faire des rotations 3-D continues autour d'axes quelconques entraînant avec lui l'objet qui est à l'intérieur. Cette section décrit la technique permettant d'obtenir des rotations continues 3-D à base d'un outil 2-D.

L'émulation de cette manipulation est faite à travers la souris, par exemple, dont le curseur est supposé toucher la sphère en un point de sa surface (Figure III.11). Si l'on suppose que la sphère est touchée au point P pour la faire tourner dans la direction du vecteur tangent \vec{d} , il est possible de déterminer l'axe de rotation ayant pour direction positive le vecteur \vec{u} , résultat du produit vectoriel des vecteurs \vec{OP} et \vec{d} où o est le centre de la sphère:

$$\vec{u} = \vec{OP} \wedge \vec{d} \tag{III.32}$$

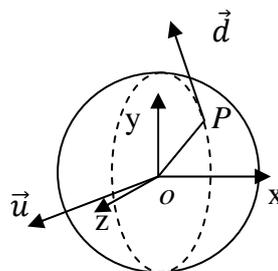


Figure III.11. Principe géométrique de la sphère virtuelle

Pour déterminer le vecteur \vec{u} , la position de départ de la souris et la direction de déplacement sont les paramètres à considérer. Soient p' et \vec{d}' respectivement ces deux paramètres. Particulièrement, afin de trouver une solution générale aux rotations à appliquer à la sphère, nous considérons trois situations particulières: celles où le point p' coïncide avec le centre o' du disque associé à la sphère aplatie sur le plan (xy) , celles où p' est sur l'axe Ox du disque et enfin lorsqu'il est à une position quelconque à l'intérieur ou à l'extérieur du disque.

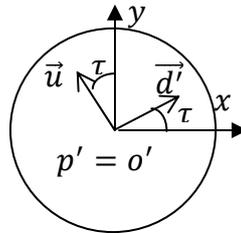


Figure III.12. Rotation ayant comme origine pour le déplacement de la souris le centre du disque

La première situation c'est lorsque le déplacement de la souris se fait le long d'un axe ayant le centre o' comme origine (Figure III.12). Si \vec{d}' fait un angle τ avec l'axe Ox alors l'axe \vec{u} est par convention situé sur le plan (xy) et a pour coordonnées:

$$[-\sin(\tau) \quad \cos(\tau) \quad 0]^T \quad (\text{III.33})$$

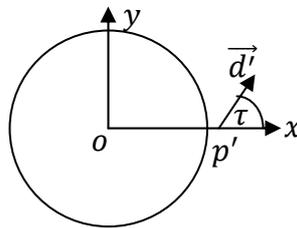


Figure III.13. Rotation ayant comme origine pour le déplacement de la souris un point quelconque sur l'axe Ox du disque

La deuxième situation à considérer c'est lorsque le point p' est situé sur l'axe Ox positif mais ne coïncide pas avec le centre O (Figure III.13). Dans ce cas, l'axe de rotation est le vecteur \vec{u} précédemment calculé mais tourné d'un angle ω autour de l'axe Oy . Cet angle ω est défini ayant une valeur entre 0° et 90° autour de l'axe Oy . Plus la position de p' est proche de O , plus la valeur de ω se rapproche de 0° . Si elle est sur le disque ou à l'extérieur, la valeur de ω est égale à 90° . Les valeurs intermédiaires peuvent être définies par n'importe quelle fonction croissante monotone. D'une manière générale, le vecteur \vec{u} se trouve alors dans le quart de l'espace limité par les x négatifs et les z positifs. En particulier, il a pour coordonnées:

$$\begin{bmatrix} -\sin(\tau) \\ \cos(\tau) \\ 0 \end{bmatrix}^T \begin{bmatrix} \cos(\omega) & 0 & -\sin(\omega) \\ 0 & 1 & 0 \\ \sin(\omega) & 0 & \cos(\omega) \end{bmatrix} \quad (\text{III.34})$$

où l'angle ω est définie par n'importe quelle fonction croissante monotone ayant les contraintes suivantes:

$$\omega = f(x) = \begin{cases} 0^\circ & \text{si } x \leq 0 \\ 90^\circ & \text{si } x \geq r \end{cases} \quad (\text{III.35})$$

avec r le rayon du disque. On constate suivant cette contrainte que l'on retrouve le cas précédent lorsque la position de départ est sur le centre du disque. De même, toutes les positions à l'extérieur du disque définissent un axe de rotation situé sur le plan (yz) .

Dans le troisième cas, qui est le cas général, c'est la situation où le point p' est situé à n'importe quelle position autre que sur le prolongement de l'axe Ox (Figure III.14). Cette situation est simplement la précédente mais avec une rotation de $\overline{op'}$ d'un angle δ autour de l'axe Oz . En conséquence, $\overline{d'}$ fera un angle $\delta + \tau$ avec l'axe Ox . Ainsi les coordonnées du vecteur \vec{u} seront:

$$\begin{bmatrix} -\sin(\tau) \\ \cos(\tau) \\ 0 \end{bmatrix}^T \begin{bmatrix} \cos(\omega) & 0 & -\sin(\omega) \\ 0 & 1 & 0 \\ \sin(\omega) & 0 & \cos(\omega) \end{bmatrix} \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{III.36})$$

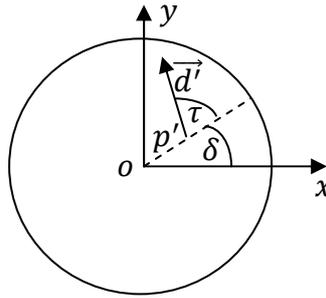


Figure III.14. Rotation ayant comme origine pour le déplacement de la souris un point quelconque n'appartenant pas à l'axe Ox du disque

Une fois les coordonnées de l'axe de rotation \vec{u} calculées, il est possible de déterminer la matrice rotation à appliquer à l'objet. Elle est calculée à base de l'angle β avec lequel il faut faire tourner l'objet autour de l'axe \vec{u} . Pour cela, la technique présentée à la section 4.2.2 est utilisée. De manière synthétique, la matrice rotation aura pour forme générale:

$$R_{\vec{u}}(\beta) = \begin{bmatrix} tu_x^2 + c & tu_xu_y + su_z & tu_xu_z - su_y \\ tu_xu_y - su_z & tu_y^2 + c & tu_yu_z + su_x \\ tu_xu_z + su_y & tu_yu_z - su_x & tu_z^2 + c \end{bmatrix} \quad (\text{III.37})$$

$(u_x \ u_y \ u_z)$ sont les composantes de \vec{u} précédemment calculées, $s = \sin(\beta)$, $c = \cos(\beta)$ et $t = 1 - \cos(\beta)$.

La détermination de l'angle β est faite à base de la distance parcourue par le curseur. Un facteur d'échelle peut être établi suivant des paramètres variant d'une application à une

autre. Les tailles de l'image et de la sphère virtuelle dans celle-ci peuvent être prises en considération dans le calcul de ce facteur.

6. Conclusion

Dans un but de suivre une démarche constructive dans les idées pour présenter les concepts relatifs à la manipulation, ce chapitre commence par présenter les principes de projection des objets virtuels dans une image à augmenter. Pour cela, le processus de transformation est d'abord présenté. Il présente les différentes transformations existantes et celle associée à la manipulation. Ensuite, le chapitre met les assises de base de la projection 2-D et 3-D et situe le principe associé à notre travail, en l'occurrence la projection d'une image sur une surface plane. C'est une sorte de mappage de texture. A base de cette présentation, il démontre comment est-ce que la manipulation des objets virtuels projetés est implémentée. Il commence par modéliser la manipulation relative aux objets 2-D, ensuite il généralise la notion aux objets 3-D. La rotation autour d'un axe quelconque est présentée comme un cas particulier associé aux objets 3-D, cependant elle constitue une généralisation de la rotation. La présentation faite est ensuite suivie par une présentation de la manipulation traitée par la matrice de transformation et de la façon de sa décomposition si besoin est. Finalement, le chapitre fait un tour des techniques et outils utilisés dans la pratique d'une manière générale et particulièrement ceux de ce travail et discute les propriétés relatives. La relation mathématique qui existe entre ces outils et la manipulation décrite est établie pour enfin définir le concept de la sphère virtuelle utilisée pour pouvoir appliquer des rotations autour d'axes quelconques à un objet en utilisant un outil 2-D à flux continue.

Quant à la situation de notre travail relativement aux concepts définis, le chapitre spécifie l'orientation faite vers les méthodes d'interaction qui supportent les tâches de base de l'interaction 3-D, particulièrement celles supportant la manipulation 2-D et 3-D. Relativement aux outils utilisés, nous avons opté pour les outils 2-D procurant une sensation de retour à base de la sphère virtuelle. La construction des objets se base sur le mappage 2-D sur des surfaces planes.

Après avoir défini tous les concepts associés à la construction des objets virtuels ainsi qu'à leur manipulation dans ce chapitre et le précédent, il est actuellement possible de pouvoir présenter le nouveau paradigme d'objets réels virtualisés. Tous les concepts associés à leur modélisation, implémentation et manipulation sont présentés dans le chapitre suivant. Le fond de notre apport de recherche y est justement inclus, avec une partie relative au collaboratif qui est différée jusqu'au cinquième chapitre.

The page features a decorative graphic consisting of three blue circles of varying sizes, each composed of concentric circles in different shades of blue. These circles are arranged vertically, with the largest at the top, a medium one in the middle, and a large one at the bottom right. Two thin blue lines intersect at the top left and extend diagonally across the page, framing the circles.

Chapitre IV

Objets Réels Virtualisés

Les bases théoriques et pratiques pour l'augmentation avec des objets virtuels reflétant des objets réels en utilisant les images de leurs différentes vues sont présentées et discutées dans ce chapitre.

Objets Réels Virtualisés

1. Introduction

Notre travail touche à l'augmentation d'une scène en utilisant un indice planaire en utilisant le principe d'un système d'augmentation vidéo. Les objets virtuels peuvent être construits de deux manières différentes: soit par l'utilisation d'une bibliothèque graphique pour le contrôle des vues de l'objet, soit à base d'images de l'objet qui sont projetées suivant des points de vue particuliers. Alors que l'orientation actuelle des travaux consiste à utiliser des bibliothèques graphiques pour le rendu des images des objets virtuels 3-D à base de modèles [50], le notre est orienté essentiellement vers l'utilisation d'images d'objets réels [14].

L'utilisation d'une bibliothèque impose un traitement d'arrière plan relatif au rendu de l'image de l'objet virtuel avant sa projection. Généralement, le temps associé n'est pas négligeable. C'est pour cette raison que certains travaux exploitent les possibilités du matériel pour accélérer cette phase [109]. A l'opposé, la projection d'une image est limitée à la détermination de la position de chaque pixel constituant.

Nous avons mentionné au chapitre précédent que les travaux correspondant à la projection d'une simple image sont minimes parce qu'ils exploitent une projection 2-D vers 2-D dont les applications sont relativement restreintes. On retrouve par exemple ceux liés au mappage de textures sur des surfaces planes [1, 66]. Notre idée est relative à l'intégration d'images d'objets rigides réels dans des séquences de scènes. Elle permet de simuler l'apparence d'un objet 3-D projeté à partir d'une image 2-D. Pour ce faire, ce n'est pas l'image entière qui est projetée mais juste la région de l'objet filmé. De plus, nous utilisons pour la gestion de plusieurs points de vue un ensemble d'images, chacune relative à une vue particulière. Ainsi, les différents points de vue calculés de la caméra servent comme référence pour contrôler les points de vue de l'objet via le choix de l'image à projeter. L'augmentation donne alors un objet virtuel dont les points de vue changent en conséquence de ceux de la scène ou des requêtes de manipulation de l'utilisateur.

La procédure d'augmentation utilisée suit un processus standard dans la recherche de l'indice ainsi que dans la calibration de la caméra. C'est à l'étape de la projection que nous utilisons les images de l'objet correspondant au point de vue estimé. La procédure de projection de ces images conduit à l'intégration d'objets réels virtualisés. Cette démarche est importante sur deux plans :

1. Projection des images comme étant des objets 2-D, ce qui réduit le temps de traitement relatif aux objets 3-D et qui est proportionnel à leur complexité.
2. Gain de coût et d'effort dans la reproduction de modèles 3-D d'objets réels pour des applications nécessitant une reproduction fidèle des ces derniers.

Dans un but d'accélérer la manipulation des images et éviter les entrées-sorties relatives à l'ouverture et à la fermeture de plusieurs fichiers associés à différentes vues, nous avons opté pour leur stockage dans un seul fichier. Le choix a été fait sur une structure vidéo qui ne nécessite qu'une seule opération d'ouverture et une seule opération de fermeture. L'ensemble des images est manipulé comme un seul flux d'images vidéo organisées suivant une structure permettant leur accès en fonction de la vue escomptée.

2. Travaux liés

Beaucoup de travaux ont été réalisés en suivant ce chemin. Le premier exemple, lui-même basé sur des travaux précédents, était Quicktime VR [106] qui suggère que le processus de modélisation/rendu traditionnels peut être évité. Il peut être remplacé par une série d'images capturées d'un environnement qui permettent à un utilisateur de regarder à une scène l'entourant à partir de points fixes de l'espace. Le travail applique une approche utilisant des images panoramiques de formes cylindriques à 360 degrés dans la reconstruction de l'environnement. Pour le cas spécifique de la rotation d'un objet, le film est constitué d'un tableau d'images à deux dimensions qui correspondent aux directions de vues de toutes les orientations possibles de l'objet. Les travaux de [80] et [111] établissent des résultats qui ne se basent pas sur des représentations géométriques. Cependant, ils requièrent un grand nombre d'images. Ils définissent une fonction 4-D qui peut générer de nouvelles images de l'objet indépendamment de la complexité de la géométrie et de l'illumination. [65] décrit une méthode pour l'affichage d'objets réels scannés. Elle est située entre les méthodes basées sur de pures modèles et celles basées sur les images. Une autre méthode intéressante présentée par [69] permet de visualiser les objets de points de vue changeant de manière continue et interactive. Sa limite, cependant, est la restriction du point de vue. Notons que la majorité des travaux font appel à la méthode d'interpolation de vues qui stocke les images clés seulement et déduit les images inexistantes.

Dans un même contexte, signalons que la majorité des travaux dans ce champ de recherche sont orientés vers la représentation de scènes. Par exemple, [67] fait une présentation d'une technique de rendu d'images spatiales basée sur le concept du z-buffer. Elle permet une navigation dans des environnements complexes mais statiques. Dans [121] la conception de WarpEngine est présentée. C'est une architecture matérielle 3-D pour le rendu en temps réel à base d'images de scènes naturelles à partir de points de vue arbitraires par la mixture d'images dépendant de la profondeur. Dans ce travail, le rendu des images des objets est fait avec la scène et non pas de manière isolée. Récemment, des méthodes hybrides ont été établies pour l'objectif d'obtenir des vues réelles avec un coût minimum. Le travail de [114] discute une façon de pouvoir mixer entre IBR et GBR (Geometry Based Rendering) dans un système de simulation de conduite dans un but d'expérimentation du trafic routier. D'un autre côté, [44] présente une vue d'ensemble de l'implémentation pratique d'un système IBR pour la représentation de scènes réelles et complexes en photoréalisme.

3. Prise d'Images

Plusieurs images correspondant à différentes vues d'un objet sont prises dans le but de permettre la projection de l'objet dans la scène suivant différentes orientations. Ces images sont prises par la caméra à partir de positions situées sur une sphère virtuelle contenant l'objet en son centre (Figure IV.1). Ainsi, la prise d'images est toujours faite à une distance fixe d et en direction du centre de la sphère où est positionné le centre de gravité de l'objet. Ainsi, chaque image est caractérisée par une propriété indiquant le point de vue de la caméra au moment de sa prise.

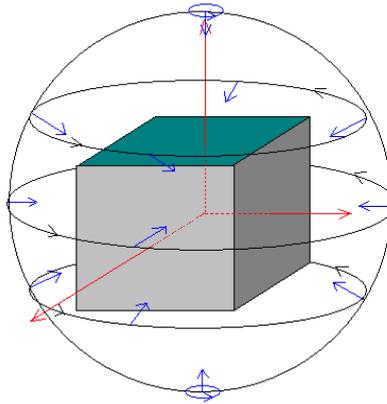


Figure IV.1. Sphère virtuelle utilisée pour la prise d'images des différentes vues d'un objet

Le principe adopté consiste à balayer l'objet par la caméra en des positions discrètes séparées par un angle fixe $st = \frac{\pi}{m}$ dans les sens de la longitude et de la latitude de la sphère (Figure IV.3). m est un entier positif choisi à priori. Les différentes positions sont alors fixées suivant le sens de la latitude (ou la verticale) d'abord ensuite un parcours dans le sens longitudinal (ou horizontal) est fait pour la prise de chacune des images. Toutes les vues de l'objet sont ainsi représentées par des images associées aux différentes positions de la caméra lors de la prise. Ces positions sont mesurées par les coordonnées polaires de cette dernière. Le référentiel utilisé pour ce positionnement est un système de coordonnées virtuel centré au centre de gravité de l'objet et donc au centre de la sphère.

Durant l'opération de balayage de l'objet, le référentiel de la caméra est toujours orienté vers l'origine du référentiel de la sphère. En ce sens, le sens positif de l'axe Oz , en l'occurrence la direction de l'axe focal, est toujours orienté vers le centre de gravité de l'objet. Les images obtenues correspondent alors toujours aux faces de l'objet vues dans la direction qui leur est perpendiculaire (Figure IV.2).

Une des images parmi celles prises est considérée comme étant celle de référence et est de coordonnées $(d, 0, 0)$. C'est l'image associée à la vue dans le sens positif de la direction de l'axe Oz du référentiel de l'objet. Toutes les autres vues, et donc les images correspondantes, sont exprimées relativement à cette dernière. Le choix de l'image de référence est purement aléatoire et il n'existe aucun critère à imposer pour sélectionner

une face de l'objet qui soit une référence. Ce qui est important, c'est qu'elle soit fixée à priori pour définir les parcours de la caméra afin de pouvoir indexer toutes les autres images de l'objet.



Figure IV.2. Exemple d'images d'objets réels associées à différentes vues

Notons que le réalisme obtenu durant la simulation du mouvement de l'objet est directement lié à la qualité des images et à l'angle de prise d'image st . Plus ce dernier est petit, meilleure est l'apparence de la simulation produite. Cependant, la performance du système d'augmentation est inversement proportionnelle. C'est la raison pour laquelle cet angle devrait être choisi de manière optimale de sorte qu'il forme un équilibre entre le réalisme et le temps réel escompté.

Notons également que le choix du pas st affecte le nombre d'images prises et par conséquent, la taille du fichier vidéo les contenant aussi bien que le temps d'accès. Par exemple, pour un angle $st = \frac{\pi}{12}$, le nombre total des images à prendre pour toutes les

vues est de $\left(\frac{\pi}{st} + 1\right) \cdot 2 \frac{\pi}{st} = 13 \times 2 \times 12 = 312$. Si chaque image est un bitmap de type RVB ayant une taille de 120×120 pixels, sa taille serait de 43200 octets et celle de toutes les images de 13478400 octets. Cependant, il est possible de faire appel à quelques heuristiques dans l'objectif de réduire cette taille totale. Les objets ayant des vues sans importance ou symétriques sont des exemples de cas permettant la réduction de la taille sans perte d'information. Il reste que la solution la plus optimale est celle de compresser les images.

4. Organisation des images

Dans le but de faciliter l'accès aux images et de réduire le temps d'entrées-sorties qui peut accompagner l'accès à une multitude de fichiers images, nous avons opté pour une structure formée d'un seul fichier. Toutes les images sont stockées dans un seul fichier AVI et accédées comme un flux vidéo. Cette structure impose donc la connaissance de la position de chaque image associée à un point de vue donné de l'objet afin de pouvoir l'utiliser. Ceci incite donc à considérer une organisation qui permet de les retrouver et les accéder facilement. Pour cela, l'ordre dans lequel est filmé l'objet est celui qui est considéré dans le stockage.

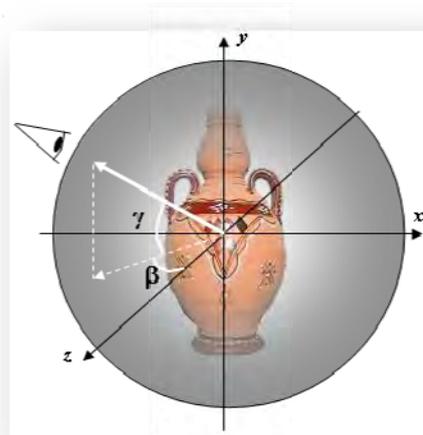


Figure IV.3. Objet photographié à des distances égales en des positions référencées par deux angles: pour chaque position de $\gamma \in [-\pi, \pi]$, $\beta \in [0, 2\pi[$

4.1. Ordre de stockage

Comme toutes les images sont prises relativement à la position qui définit celle de référence, il est possible d'exprimer analytiquement leurs positions relativement à celle-ci. Puisque toute position fait nécessairement un angle γ par rapport au plan (xz) et un angle β par rapport au plan (yz) éventuellement nuls, et que la vue de référence est située sur l'intersection de ces deux plans dans le sens positif, les différentes positions sont obligatoirement situées à $\beta = n \cdot st$ et $\gamma = k \cdot st$ selon l'hypothèse faite à la section 3 (Figure IV.3). st représente le pas de l'angle choisi pour les différentes prises, et n et k des entiers représentant le nombre de pas st faits par la caméra relativement à la

position de référence respectivement dans les sens de la longitude (ou horizontal) et dans le sens de la latitude (ou vertical). Les angles sont mesurés dans le sens positif d'un système de coordonnées droit.

Dans ce cas, une première solution non générale peut être proposée. Elle est exprimée empiriquement de manière assez simple (IV.1). Elle permet de situer une image à une position p dans le flux:

$$p = n + 2k \frac{\pi}{st} \quad (\text{IV.1})$$

L'expression (IV.1) a le sens suivant. La caméra est considérée faire un demi tour complet, soit π , autour de l'objet dans le sens de la latitude et à des positions multiples de $\pi/2st$. Pour chaque position k où la caméra est inclinée d'un angle $\gamma = k.st$ par rapport à l'horizontale (plan (xz)), les images de l'objet sont prises à partir du plan horizontal - parallèle à (xz) et situé à une position k - et ce en faisant un tour complet autour de l'axe Oy de l'objet. Chacune de ces prises est située à une position n sur le plan horizontal qui est également un multiple de l'angle st . Ainsi, les positions longitudinales doivent être au nombre de $\frac{\pi}{st} + 1$ et celles horizontales au nombre de $\frac{2\pi}{st}$. Pour passer d'un plan k à un plan $k + 1$, il faut donc considérer $(k + 1) \frac{2\pi}{st}$ images précédentes (k commence à partir de 0). Les images prises au plan $k + 1$ seront, quant à elles, comptées par leur position n .

Cependant, cette expression (IV.1) est valable pour des valeurs de n et k positives. En particulier, celles de $n \in \left[0, \frac{2\pi}{st}\right[$ puisque la dernière position horizontale $\frac{2\pi}{st}$ est elle-même la position 0. Ceci est vrai puisque l'angle $\beta \in [0, 2\pi[$. En effet, la valeur maximale prise par n est $2m = \frac{2\pi}{st}$ qui est également le nombre de pas d'angles existant dans 2π . Dans le cas contraire, c'est à dire pour des valeurs de k négatives, il est possible d'avoir des valeurs de p négatives, ce qui n'a pas de sens. Cela pose un problème puisque l'objectif escompté est d'avoir l'image de référence à la position $n = 0$ et $k = 0$ et ceci n'est possible que si $k \geq 0$. Hors, si cela n'est pas le cas, il serait difficile de la repérer dans le flux.

En effet, les différentes valeurs de l'entier k sont comptées à partir de 0 pour une première position et évoluent jusqu'à la valeur maximale de $\frac{\pi}{st}$. Les valeurs de k sont donc dans l'intervalle $\left[0, \frac{\pi}{st}\right]$. La valeur 0 devrait représenter les vues de l'objet situées sur sa diagonale horizontale dont l'une d'elles ($n = 0$) est la vue de référence. Hors, cela impose que la caméra fasse des prises à partir de la demi-sphère haute ou basse pour faire sa rotation dans le sens de la latitude de π , ce qui est insensé puisque les mêmes images seront prises deux fois. Cette situation n'a qu'une seule solution, c'est celle d'imposer des valeurs négatives pour k en raison des angles négatifs situés sur la demi-sphère basse, ce qui conduit au problème de positions négatives insensées.

Ainsi, afin de donner une solution aux valeurs négatives de p et également généraliser la formule pour l'accès à toutes les images prises et avoir l'image de référence située dans la position $n = 0$ et $k = 0$, nous avons opté pour une organisation de sorte que les images relatives aux vues de bas soient stockées les premières dans le flux. Ainsi, les angles des différentes positions relatives à k doivent être dans l'intervalle $[-\pi, \pi]$, soit $k \in \left[-\frac{\pi}{2st}, \frac{\pi}{2st}\right]$. Comme les positions doivent être positives, il faut compléter l'expression précédente (IV.1) pour les valeurs négatives de k afin que les valeurs de p ne soient pas négatives. Conséquemment, les valeurs de $k \in \left[0, \frac{\pi}{2st}\right]$ représentent les images prises des vues de la demi-sphère du haut et celles de $k \in \left[-\frac{\pi}{2st}, 0\right]$ sont celles prises pour les vues de la demi-sphère du bas.

Pour cela, il est clair que les prises d'images faites dans la demi sphère du bas sont égales au nombre de $\frac{\pi}{2st} \times 2 \frac{\pi}{st}$. Le premier $\frac{\pi}{st}$ correspond au nombre de plans, autres que (xz) , où des images sont prises par la caméra dans la demi sphère du bas. $2 \frac{\pi}{st}$ est simplement le nombre d'images prises par plan. En ajoutant la valeur de $\frac{\pi^2}{st^2}$ à l'expression de p dans (IV.1), les valeurs de p seront toujours positives ou nulles. La forme générale de l'expression devient alors (IV.2):

$$p = n + \left(2k + \frac{\pi}{st}\right) \frac{\pi}{st} \quad (\text{IV.2})$$

Selon cette expression (IV.2), l'image de la vue de référence est stockée à la position $p = \frac{\pi^2}{st^2}$ où $n = 0$ et $k = 0$. Par exemple, pour une valeur de $st = \pi/4$, la position de la première image dans le flux à $\beta = -\pi$ correspond à $k = -2$ comme étant la vue du bas de l'objet. Aussi, pour $n = 0.7$, les positions relatives des images correspondantes sont respectivement $p = 0.7$. L'orientation suivante pour $\beta = -\pi/4$ correspond à $k = -1$. C'est alors que pour $n = 0.7$, les positions relatives des images sont maintenant $p = 8.15$. L'image de référence devrait être à la position $p = 16$. Elle correspond aux valeurs $n = 0$ et $k = 0$ et ainsi de suite.

Enfin, pour que l'image de référence soit située sur le plan (xz) et pouvoir faire la prise de l'image de référence à $\gamma = 0$ et $\beta = 0$, une condition supplémentaire doit être respectée. Cette condition impose que st soit de la forme $\frac{\pi}{m} = \frac{\pi}{2a}$ où a est un entier non nul et positif. Si cette condition n'est pas respectée, aucune image ne sera prise par la caméra dans le plan (xz) . C'est le cas, par exemple, de la valeur $st = \frac{\pi}{3}$. Les seuls plans possibles pour les prises sont situés à des angles $\pm \frac{\pi}{6}$ par rapport au plan (xz) .

Cette organisation est importante pour la simulation des rotations de l'objet comme nous le détaillons plus loin. Selon ce principe, les actions ou requêtes de l'utilisateur peuvent être traitées comme des opérations algébriques sur les valeurs des paramètres n et k . Toute requête de rotation de l'objet autour des axes Ox ou Oy est transformée

respectivement en incrément de k ou de n par $+1$ ou -1 . Ainsi, l'orientation en cours correspond à l'image située à la position évaluée de p .

4.2. Résolution des images

Le problème de la résolution des images des objets virtuels est assez important. Si elles ont toutes été prises à partir d'une certaine distance d constante, un besoin particulier de détail requis dans le système d'augmentation peut conduire à une dégradation de la qualité. Ceci se produit lorsqu'une opération de zoom avant d'un taux élevée est requise. L'image de l'objet virtuelle serait alors dégradée et déformée quelque soit l'algorithme de zoom utilisé et ne reflète plus la qualité de l'objet une fois agrandi. Dans cette situation, il est préférable de disposer de plusieurs images de la même vues prises à des distances différentes. On obtient alors un ensemble de couches d'images correspondant à plusieurs sphères virtuelles. Ainsi, en fonction des besoins de zoom, le système d'augmentation fait un choix entre les images d'une couche particulière.

L'expression (IV.2) pour le stockage des images d'un objet est obtenue de manière empirique, comme nous l'avons déjà signalé, mais elle correspond à une seule résolution seulement. Malgré qu'il existe plusieurs solutions pour le stockage des images appartenant à des systèmes de résolution multiples [80, 106, 111], ce travail est une extension de l'organisation à structure linéaire permettant de fournir une résolution multiple. Pour parvenir à cet objectif, nous profitons de l'expression (IV.2) pour implémenter un système multi-résolution et au même moment rassembler dans le même fichier, l'ensemble de toutes les images relatives à différentes résolutions.

Cette extension de l'implémentation utilise le même principe précédent. Considérons un niveau de résolution r dont les valeurs sont supérieures à 0. Il exprime le nombre de sphères virtuelles utilisées dans la prise d'images autres que la première. Lorsqu'un seul niveau est utilisé, sa valeur est nulle. De plus, comme c'est déjà indiqué, le nombre d'images prises dans les sens de la latitude et de la longitude pour une sphère sont respectivement $1 + \frac{\pi}{st}$ et $\frac{2\pi}{st}$. Ainsi le nombre total d'images de chaque niveau est (IV.3):

$$s = \left(1 + \frac{\pi}{st}\right) \frac{2\pi}{st} \quad (\text{IV.3})$$

Le principe suivi est que chaque image soit stockée simplement à la position $r.s + p$. Conséquemment, l'expression (IV.2) devient:

$$p = n + \left(2k + 2r \left(1 + \frac{\pi}{st}\right) + \frac{\pi}{st}\right) \frac{\pi}{st} \quad (\text{IV.4})$$

4.3. Restitution des images

La restitution des images de l'objet réel à partir du flux vidéo pour les projeter dans les images à augmenter est simplement l'opération inverse de leur stockage. Il suffit de sélectionner le point de vue d'une caméra virtuelle photographiant l'objet dans la scène en cours et déterminer l'image à projeter. Ceci est valable pour la projection initiale du

processus d'augmentation avant toute action externe sur l'objet. Par hypothèse, ce point de vue est identique à celui de la caméra réelle filmant la scène réelle. Pour cela, il suffit de déduire les angles β et γ - définis dans la section précédente - de la position de la caméra pour estimer la position p (IV.2) de l'image correspondante dans le flux. La position de la caméra, et donc les angles β et γ , peuvent être déduits par la matrice de rotation estimée à partir de l'homographie dans l'expression (II.41).

Le principe d'estimation des angles β et γ se base sur une propriété de la matrice rotation. Ses lignes sont les coordonnées des vecteurs unitaires du système de coordonnées obtenu par rotation dans le système de coordonnées original et ses colonnes représentent les coordonnées des vecteurs unitaires du système de coordonnées original dans le système de coordonnées obtenu par rotation. Comme la matrice rotation estimée exprime la position de l'indice relativement à celui de la caméra et comme l'axe Oz représente la direction de l'axe focal, il suffit d'estimer les angles que fait l'axe représenté par la troisième colonne de la matrice par rapport aux plans (xz) et (yz) . Cette estimation permet de déduire les angles β et γ avec des corrections d'orientation. Ainsi, il est rare que les angles estimés par la procédure précédente soient des multiples entiers du pas st utilisé lors des prises des images de l'objet. Dans ce cas, il est nécessaire d'estimer la position p de l'image à projeter parmi celles prises et en même temps qui est la plus proche de l'orientation calculée.

Sachant que l'objectif est de chercher les valeurs des deux paramètres n et k qui permettent d'estimer la position p , les angles obtenus par l'estimation précédente vérifient certainement les inégalités suivantes:

$$\begin{cases} k' \leq \frac{\gamma}{st} \leq k' + 1 \\ n' \leq \frac{\beta}{st} \leq n' + 1 \end{cases} \quad (\text{IV.5})$$

où k' et n' sont respectivement les résultats des divisions entières γ/st et β/st . Le choix de l'image disponible la plus proche est alors celle située à la position p calculée à la base des valeurs minimales de $((\gamma/st) - k', (k' + 1) - (\gamma/st))$ pour γ et $((\beta/st) - n', (n' + 1) - (\beta/st))$ pour β . Les paramètres n et k sont alors estimés comme suit:

$$\begin{cases} k = \begin{cases} k' \text{ si } (\gamma/st) - k' < (k' + 1) - (\gamma/st) \\ k' + 1 \text{ sinon} \end{cases} \\ n = \begin{cases} n' \text{ si } (\beta/st) - n' < (n' + 1) - (\beta/st) \\ n' + 1 \text{ sinon} \end{cases} \end{cases} \quad (\text{IV.6})$$

Dans cette dernière expression (IV.6), γ/st et β/st sont des divisions réelles. La valeur de p trouvée sera utilisée comme index de l'image à utiliser dans le flux correspondant au niveau de résolution voulu.

5. Projection de l'objet

Pour que le système d'augmentation fonctionne correctement du point de vue de l'aspect visuel, des contraintes doivent être respectées et elles concernent particulièrement la manière dont est faite la projection et le contenu projeté. En effet, comme la projection ou l'intégration d'un objet réel dans une image à augmenter passe nécessairement par ses images qui sont particulièrement des objets planaires, l'augmentation en soit n'est pas faite en 3-D mais en 2-D. Ceci implique l'inévitable conclusion que la projection fasse appel uniquement à l'homographie sans déduction de la matrice de projection. Cependant, la projection d'une image par le principe de l'homographie résulte sur une perspective inhérente à la matrice homographie calculée et une taille adaptée à celle des points de correspondance déterminés qu'il est nécessaire de corriger. De plus, c'est uniquement la région de l'objet dans l'image qu'il faut projeter car la région l'entourant n'a aucun intérêt dans l'augmentation (Figure IV.4).

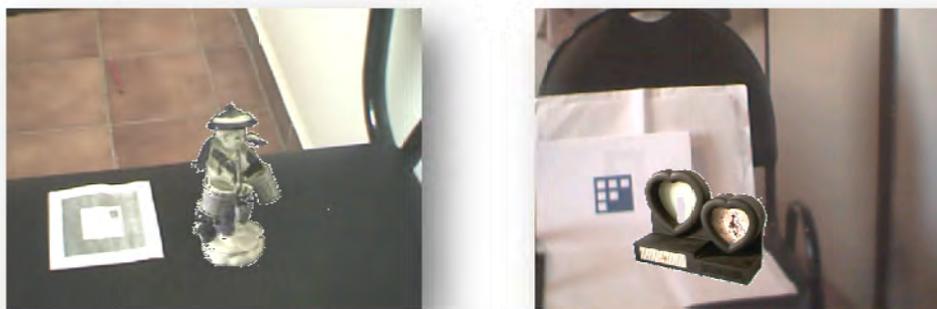


Figure IV.4. Images d'objets réels projetées dans des orientations particulières

5.1. Principe de la projection des images de l'objet

La projection directe d'une image de l'objet à partir de l'homographie calculée donnerait un objet projeté de manière déformée dans le sens du plan détecté et ayant une taille qui n'est pas obligatoirement réelle. Au sens visuel, l'objet paraîtrait incliné dans le sens du plan de l'indice. Plus ce dernier est incliné par rapport à la caméra, plus la déformation perspective est ressentie. Cette inclinaison est acceptable si la projection concernait une image ordinaire dans le but de respecter la perspective visuelle résultante. Or cette perspective est déjà compensée par l'image de l'objet lors de sa prise puisque la caméra a été inclinée dans ce sens. Dans ce cas, le système d'augmentation devrait plutôt choisir l'image de la vue de l'objet qui correspond à cette inclinaison et non pas incliner une image d'une vue sans relation.

La solution à la projection des images de l'objet pour qu'elles ne soient pas déformées, est de les projeter relativement à la région détectée de l'indice après être réarrangée (Figure IV.5). Le réarrangement à appliquer pour la région relativement à son centre concerne l'orientation et la taille. Il consiste à estimer un carré virtuel non tourné de la taille de l'indice qui est parallèle au plan focal et ayant comme centre celui de la région détectée. Autrement exprimé, ce réarrangement constitue deux rotations virtuelles

appliquées à la région détectée. Une première rotation est appliquée à la région toute entière pour la redresser parallèlement au plan (xy) . Une autre rotation est appliquée relativement au centre de gravité calculé de sorte à avoir un carré droit ayant ses cotés parallèles aux axes Ox et Oy . C'est cette nouvelle région virtuelle qui servira pour le calcul d'une nouvelle homographie.

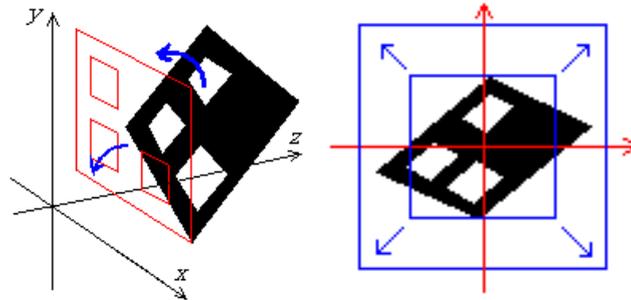


Figure IV.5. Réarrangement de la région détectée de l'indice parallèlement au plan focal (gauche) suivie par son changement en taille pour l'adapter à la taille effective de l'objet (droite)

A ce niveau de traitement, il faudrait trouver la taille à utiliser des cotés de la région virtuelle. Les côtés de la région détectée ne sont généralement pas égaux à cause de l'effet de la perspective et nécessitent donc la détermination de la taille à considérer après réarrangement. Pour cela, une restriction a été imposée sur l'indice afin qu'il soit de forme carré. De cette façon, les longueurs des quatre lignes reliant les coins de la région détectée sont d'abord estimées à base d'une seule. La plus longue est considérée comme étant le côté le plus proche de la caméra et sa longueur sera utilisée pour les trois autres côtés de la région virtuelle. Malgré que l'erreur augmente proportionnellement à l'inclinaison de l'ensemble des côtés de l'indice dans la scène, les résultats pratiques sont généralement acceptables sur le plan visuel.

5.2. Extraction de la région de l'image de l'objet

La contrainte relative à la sélection du contenu à projeter est également importante en raison de la non-importance du background de l'objet en question. C'est pourquoi, lors de la photographie initiale de l'objet, ce dernier doit obligatoirement posséder un arrière-plan (background) sans importance pour l'opération d'augmentation. La projection ne devrait donc pas concerner tous les pixels de son image mais plutôt uniquement ceux de la région opportune. Particulièrement, son arrière-plan doit être extrait par segmentation pour être ignoré dans la projection. C'est l'une des raisons qui fait que lors des prises des images des vues de l'objet, son arrière plan est toujours choisi uniforme de couleur claire ou sombre. Cette uniformité permet d'extraire facilement la région interne qui est supposée être celle de l'objet. Le choix d'un background sombre ou clair est fait à base de la couleur dominante de l'objet qui lui doit être opposée. Cette technique est généralement utilisée au cinéma pour simuler un acteur dans un environnement virtuel. Voir par exemple un travail similaire dans [88].

Ainsi, l'extraction de la région convenable de l'objet à partir de l'image entière est faite après la prise de chaque image et avant son stockage. Ce prétraitement consiste à

déclarer la région entourant la partie de l'objet comme transparente. Afin de simuler la transparence d'une région, une couleur blanche pure ou noire pure lui est imposée après être détectée (Figure IV.6). Ainsi, lors de la projection de l'image, les pixels appartenant à la partie transparente sont simplement exclus de l'opération.

Un inconvénient relatif à cette méthode est évident. Lorsque certaines extrémités de la région de l'objet présentent des couleurs qui sont proches de l'arrière plan, l'opération de segmentation peut ne pas donner de bons résultats. Ceci peut être dû à l'effet de fausses couleurs générées par la caméra. Dans ce cas, la partie extraite de l'objet peut être déformée et pourrait résulter sur une apparence anormale de ce dernier. C'est ce qui est remarquable lorsque certaines parties de l'image de l'objet sont perdues dans le processus de projection puisqu'elles sont ignorées et donc non projetées.

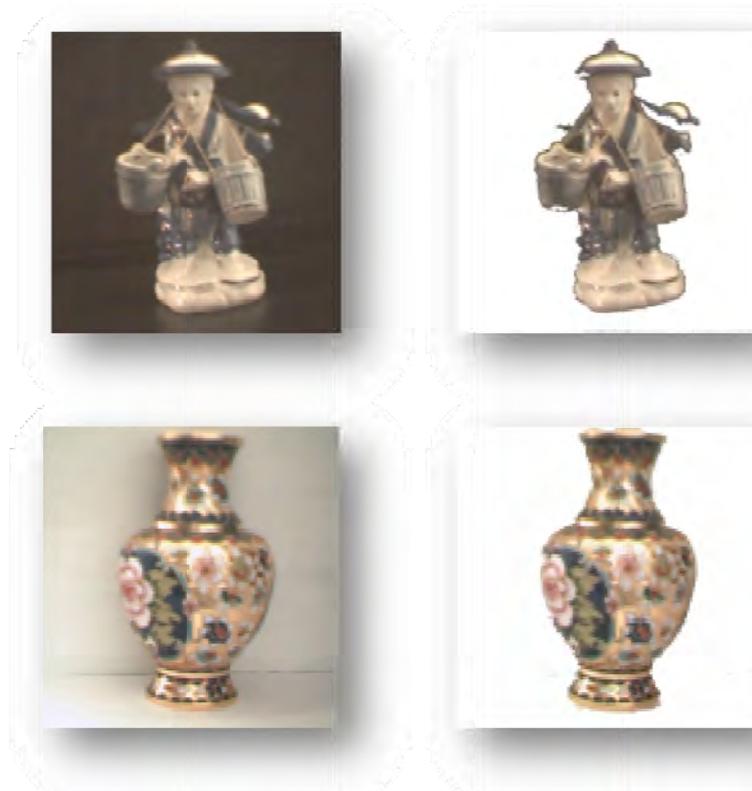


Figure IV.6. Exemple d'extraction de la région des objets et déclaration de leur arrière-plan comme région transparente (blanche)

Cette opération de prétraitement peut être réalisée en utilisant n'importe quel utilitaire de traitement d'image. L'inconvénient conséquent est le temps nécessaire à attribuer lorsque le nombre d'image augmente. C'est pourquoi l'automatisation de cette phase est l'une des améliorations que nous comptons apporter au système.

5.3. Contrainte de taille

L'une des contraintes à respecter pour parvenir à un réalisme acceptable est l'adaptation de la taille de l'objet projeté à celles des objets de la scène. Deux solutions peuvent être utilisées. Une première consiste à connaître ses dimensions relativement aux unités de mesure du repère de l'indice. Or, cette solution nous fait retourner au problème de la

projection 3-D et c'est ce que nous voudrions éviter. Une deuxième solution consiste à manipuler exclusivement les images disponibles mais en considérant les tailles relatives réelles de l'objet et de l'indice.

Il est évident que si l'objet projeté est adapté en taille à la région détectée de l'indice, il le sera également par rapport à tous les objets réels de la scène. Mais ceci ne peut être obtenu par la considération directe des tailles relatives des images. Ces dernières ne reflètent en rien les proportions des tailles réelles (Figure IV.7). C'est alors que, connaissant les tailles relatives de l'objet réel, dont les images sont à projeter et de l'indice recherché dans la séquence, il est possible de déduire les proportions de taille qu'il faudrait appliquer à leurs images. Ces proportions permettent d'avoir les tailles des objets virtuels adaptées en proportion à celles des objets réels de la scène augmentée.



Figure IV.7. Deux images de tailles égales dont les tailles des objets réels sont différentes (Figure IV.8)

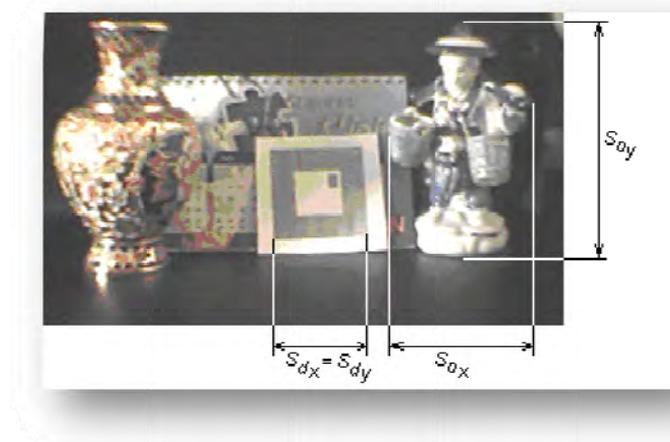


Figure IV.8. Proportions réelles de l'indice et des objets ainsi que les principes de mesures utilisés

Appelons S_o et S_d les vecteurs des tailles respectives de l'objet réel et de l'indice (Figure IV.8). Notons que la taille à considérer de l'objet virtuel est celle qui est associée à la vue de référence. S_o aussi bien que S_d doivent être des paramètres d'entrée du processus d'augmentation. A base de ces paramètres, il est possible d'évaluer une proportion des tailles réelles $P_r = S_o/S_d$. De plus, supposons que la région extraite associée à l'objet

dans son image de référence ait pour vecteur taille S_{i_o} et que l'image elle-même ait pour vecteur taille S_i (Figure IV.9). La proportion de la taille de la région de l'objet dans son image est $P_i = S_{i_o}/S_i$. En prenant en considération que la taille et donc que chacun des vecteurs précédents est constitué de deux composantes, largeur (W) et hauteur (H), nous obtenons en notation matricielle (IV.7) et (IV.8):

$$P_r = \begin{bmatrix} W_{P_r} \\ H_{P_r} \end{bmatrix} = \begin{bmatrix} W_o/W_d \\ H_o/H_d \end{bmatrix} \quad (\text{IV.7})$$

$$P_i = \begin{bmatrix} W_{P_i} \\ H_{P_i} \end{bmatrix} = \begin{bmatrix} W_{i_o}/W_i \\ H_{i_o}/H_i \end{bmatrix} \quad (\text{IV.8})$$



Figure IV.9. Tailles à considérer dans l'image de référence de l'objet pour le calcul des proportions

Sachant que la proportion de taille de l'objet et de l'indice est une constante, sous l'hypothèse qu'ils soient des objets rigides, elle devrait normalement être maintenue entre la région détectée de l'indice dans la scène et l'objet après sa projection. Il suffit alors de projeter l'image de l'objet en respectant cette proportion pour satisfaire un réalisme visuel des tailles. Pour cela, il faudrait déterminer la proportion existante entre la taille de l'objet dans son image et celle de la région détectée de l'indice pour adapter la taille de l'image toute entière de l'objet à cette dernière.

Nous pouvons écrire:

$$S_{i_{os}} = P_r \cdot S_{i_d} \quad (\text{IV.9})$$

où $S_{i_{os}}$ est la taille de l'objet à utiliser dans l'image en cours et qui doit être mise à l'échelle de la taille S_{i_d} de la région détectée de l'indice. De même, puisque la taille de l'objet à projeter est mise à l'échelle, c'est indirectement l'image entière de sa vue qui, elle aussi, doit être mise à l'échelle pour avoir une taille S_{i_s} (IV.10):

$$S_{i_s} = \frac{S_{i_{os}}}{P_i} = \frac{P_r \cdot S_{i_d}}{P_i} \quad (\text{IV.10})$$

Ainsi, la proportion à utiliser pour changer la taille de l'image de l'objet avant sa projection afin d'avoir une proportion de taille acceptable est (IV.11):

$$P_s = \begin{bmatrix} W_s \\ H_s \end{bmatrix} = \frac{S_{i_s}}{S_{i_d}} = \frac{P_r}{P_i} = \begin{bmatrix} W_{P_r}/W_{P_i} \\ H_{P_r}/H_{P_i} \end{bmatrix} \quad (\text{IV.11})$$

qui est donc une constante.

La valeur P_s est alors utilisée pour l'évaluation de la taille de la région virtuelle pour le calcul de l'homographie associée à l'image. A base de P_s , une estimation des côtés de la région détectée de l'indice, qui est d'abord retournée, est faite par la translation de chaque coin $p_{d_i}(x_{d_i}, y_{d_i})$, $i = 0..4$, exprimé relativement au centre de la région, à la position $p_{o_i}(x_{o_i}, y_{o_i})$. En notation matricielle cela donne:

$$p_{o_i} = \begin{bmatrix} x_{o_i} \\ y_{o_i} \end{bmatrix} = \begin{bmatrix} W_s & 0 \\ 0 & H_s \end{bmatrix} \begin{bmatrix} x_{d_i} \\ y_{d_i} \end{bmatrix} \quad (\text{IV.12})$$

Ce résultat est appliqué au cadre virtuel parallèle au plan (xy) qui est la région détectée et réarrangée de l'indice de sorte que l'image projetée de l'objet virtuel serait adaptée en taille et non déformée.

6. Manipulation d'un objet réel virtualisé

Le choix de l'image à projeter dépend de deux facteurs. Le point de vue de la caméra et les requêtes de l'utilisateur traduites en des actions sur l'objet pour le positionner dans différentes positions et orientations dans la scène augmentée. Comme toute manipulation est une composition de translations et de rotations, il est nécessaire de traduire chacune d'elles en termes de projection d'images de l'objet réel. La translation étant une opération relativement simple par rapport à la rotation, nos premiers résultats se sont intéressés particulièrement à la rotation et ont restreint l'utilisateur à ne manipuler l'objet que dans le sens des prises d'images. C'est ce qui est qualifié de manipulation artificielle. Ensuite, une solution générale complète fut construite pour des rotations quelconques.

6.1. Simulation d'une manipulation artificielle

Cette sous-section traite les manipulations que nous qualifions d'artificielles. En ce sens, le fait de restreindre la manipulation à des directions et des orientations limitées constitue une touche artificielle. Des translations qui ne respectent pas le changement de zoom ou de vue suite à une avance, à un recule, à un déplacement vers la gauche, vers la droite, vers le haut ou vers le bas sont considérées comme artificielles. De même, des rotations qui ne peuvent être appliquées dans tous les sens constituent une restriction au sentiment de réalisme. Bien que cela soit une limite, nous présentons une description des solutions développées dans certains de nos travaux [8, 9, 10, 14] parce qu'ils constituent un fondement pour les suivants.

6.1.1. Translations

Les translations purement artificielles suivant les axes Ox et Oy constituent de simples manipulations d'images. L'image courante est simplement traduite dans le système de

coordonnées de l'indice avant sa projection dans l'image à augmenter. Les translations suivant l'axe Oz affectent essentiellement la taille de l'image à projeter afin de simuler l'effet de zoom dû à la perspective.

Les translations à considérer suivant l'axe Oz où l'objet est supposé translaté le long de l'axe focal sont limitées aux requêtes utilisateur. Ceux résultant du déplacement de la caméra sont pris en compte indirectement par le biais de la taille détectée de l'indice. La taille estimée de la région virtuelle fait que l'objet s'adapte en taille de manière indirecte. Ainsi, suite à une action de l'utilisateur et dans l'objectif de simuler un déplacement m de l'objet vers l'avant ou vers l'arrière le long de l'axe optique, la taille apparente de la région virtuelle, en conséquence de ce déplacement, devrait être réduite ou agrandie. Cet effet visuel de perspective qui induit un zoom est à estimer en termes de taille et cela avant le calcul de l'homographie.

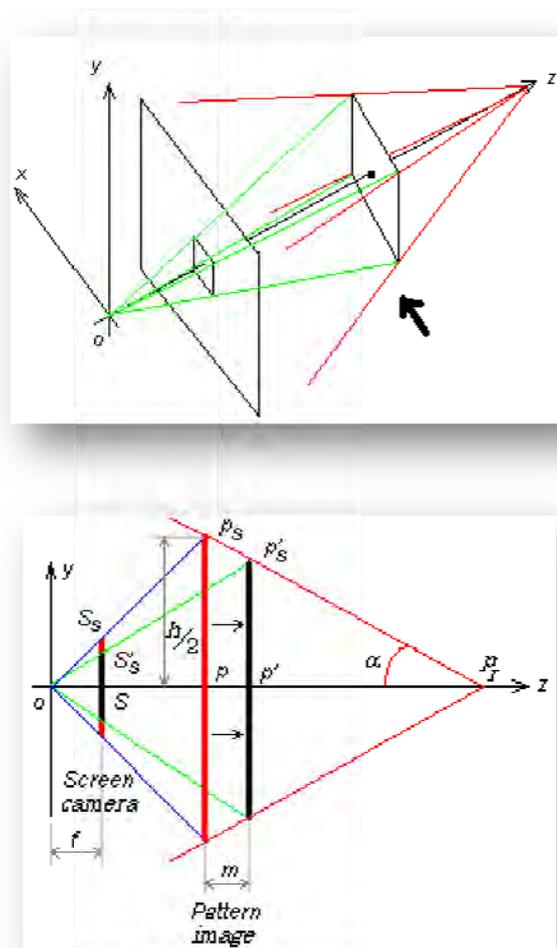


Figure IV.10. Les bases géométriques associées au déplacement de l'objet le long de l'axe Oz . La figure du bas est la vue de face suivant la flèche de celle du haut

Supposons que l'indice est déplacé à une position définie par le vecteur \vec{op} suivant l'axe Oz du repère de la caméra (Figure IV.10). Sa translation vers une nouvelle position définie par le vecteur \vec{op}' le long de l'axe Oz résulte sur un changement de taille de son

image générée sur l'écran de la caméra. La différence de taille $S_s S'_s$ qui en découle est directement liée à la proportion de déplacement $|\overrightarrow{pp'}|$ effectuée. Formellement, le calcul de la nouvelle taille requiert l'estimation du segment SS'_s où S est le centre de la région détectée qui constitue aussi son intersection avec l'axe Oz . Autrement exprimé, il faudrait estimer la différence de taille $S_s S'_s$ entre la taille SS_s détectée de l'indice dans l'image et celle à estimer SS'_s . Ceci est vrai sous l'hypothèse que l'indice est de forme carrée et est centré sur l'axe Oz . Par l'application de simples principes des triangles semblables, il est facile d'établir que (IV.13):

$$SS'_s = \frac{f \cdot p'p'_s}{|\overrightarrow{op}| + m} \quad (\text{IV.13})$$

m est la valeur algébrique associée au déplacement, f est la distance focale estimée de la caméra, $p'p'_s$ est la nouvelle demi-taille estimée de l'indice après déplacement due à l'effet de perspective et $|\overrightarrow{op}|$ la position actuelle de l'indice dans le repère de la caméra.

Suivant le même principe, il est possible d'écrire que (IV.14):

$$|\overrightarrow{op}| = \frac{f \cdot pp_s}{SS_s} = \frac{f \cdot h}{2SS_s} \quad (\text{IV.14})$$

avec h représentant la longueur d'un coté de l'indice détecté. De même, il est simple d'établir l'expression (IV.15):

$$p'p'_s = p'p_r \cdot tg\alpha = (pp_r - m)tg\alpha = \left(\frac{h}{2tg\alpha} - m \right) tg\alpha \quad (\text{IV.15})$$

Par remplacement de l'expression de $p'p'_s$ (IV.15) et de $|\overrightarrow{op}|$ (IV.14) dans l'expression (IV.13), on obtient l'expression de SS'_s donnée dans (IV.16). La valeur de l'angle 2α relative à l'angle de projection est une donnée fixée par l'utilisateur.

$$SS'_s = \frac{f(h - 2mtg\alpha)}{\left(\frac{f \cdot h}{SS_s} + 2m \right)} \quad (\text{IV.16})$$

6.1.2. Rotations

La rotation autour de l'axe Oz dans le prolongement de l'axe focal consiste simplement à faire tourner l'image dans le système de coordonnées de l'indice avant sa projection dans la séquence. Cependant, les requêtes de l'utilisateur relatives aux rotations suivant les axes Ox et Oy sont utilisées pour rechercher l'image reflétant le point de vue demandé. Les valeurs courantes des paramètres n et k sont simplement incrémentées ou décrétementées en fonction du sens requis par l'utilisateur. Elles peuvent être traduites en des valeurs d'angles qui sont constamment misent à jour et servent comme indexes pour identifier l'image correspondante.

A chaque modification de n ou k , l'image de l'objet réel qui se trouve à la position p associée est projetée dans l'image à augmenter. Une rotation autour de l'axe Ox est interprétée par ± 1 de la valeur du paramètre n . Celle qui se fait autour de l'axe Oy change la valeur du paramètre k par ± 1 . Malheureusement, cette façon de manipuler l'objet restreint les vues possibles dans la scène augmentée à celles prises par la caméra. Néanmoins, il est possible d'avoir un parcours différent que celui fait initialement par la caméra. Toutefois, les seules transitions possibles pour accomplir des rotations sont celles qui font tourner l'objet suivant la longitude ou la latitude.

6.1.3. Récapitulation du principe d'augmentation et de manipulation

Afin de résumer le processus d'augmentation utilisant les images d'objets réels ainsi que la manipulation artificielle associée, précisons initialement les paramètres qui doivent être indiqués au début d'une session d'augmentation. Le premier est celui relatif au pas st utilisé durant la phase initiale de prise d'images. L'autre paramètre associé au même contexte est le nom du fichier du flux vidéo qui contient les vues de l'objet. Concernant le contrôle du réalisme visuel à travers les proportions de taille, les valeurs des vecteurs S_o et S_p représentant respectivement les mesures réelles de l'objet et de l'indice sont également nécessaires. Finalement, la valeur de l'angle 2α qui contrôle les translations en z relativement au champ de vue doit également être spécifié. La valeur que nous avons utilisé dans nos prototypes développés est de $\frac{\pi}{4}$. La figure suivante (Figure IV.11) est une représentation schématique du processus d'augmentation pour chaque image de la séquence.

L'augmentation traite deux types de flux en entrées: la séquence vidéo et les requêtes de l'utilisateur (Figure IV.11). L'aspect le plus important est relatif au calcul de la matrice homographie. Une première matrice est estimée à partir des coins détectés et est utilisée juste pour l'estimation des paramètres de la caméra et est ensuite rejetée. Celle qui est utilisée dans la projection est celle estimée à partir des coins de la région réarrangée après avoir été translatée et/ou tournée.

6.2. Simulation d'une manipulation naturelle

Une manipulation ayant un aspect naturel est celle qui permet de simuler une apparence naturelle de l'objet suite à des requêtes de translations ou de rotations. Dans ce contexte, la manipulation d'objets synthétiques est entièrement contrôlée. Tous les calculs de positionnement et de projection sont faits avec une maîtrise totale. Un très grand nombre de références existent dans le domaine et l'outil le plus simple, très complet et est libre d'utilisation (GNU), c'est OpenGL. Par contre, dans le domaine des objets réels manipulés, peu de travaux existent. Cette sous-section présente nos travaux relatifs à la simulation d'une translation et d'une rotation naturelles. Ceux associés à la translation ne sont pas encore finalisés et feront l'objet d'une continuité de ce travail. Ceux relatifs à la rotation, par contre, constituent des résultats complets relativement à l'objectif mais peuvent éventuellement être améliorés.

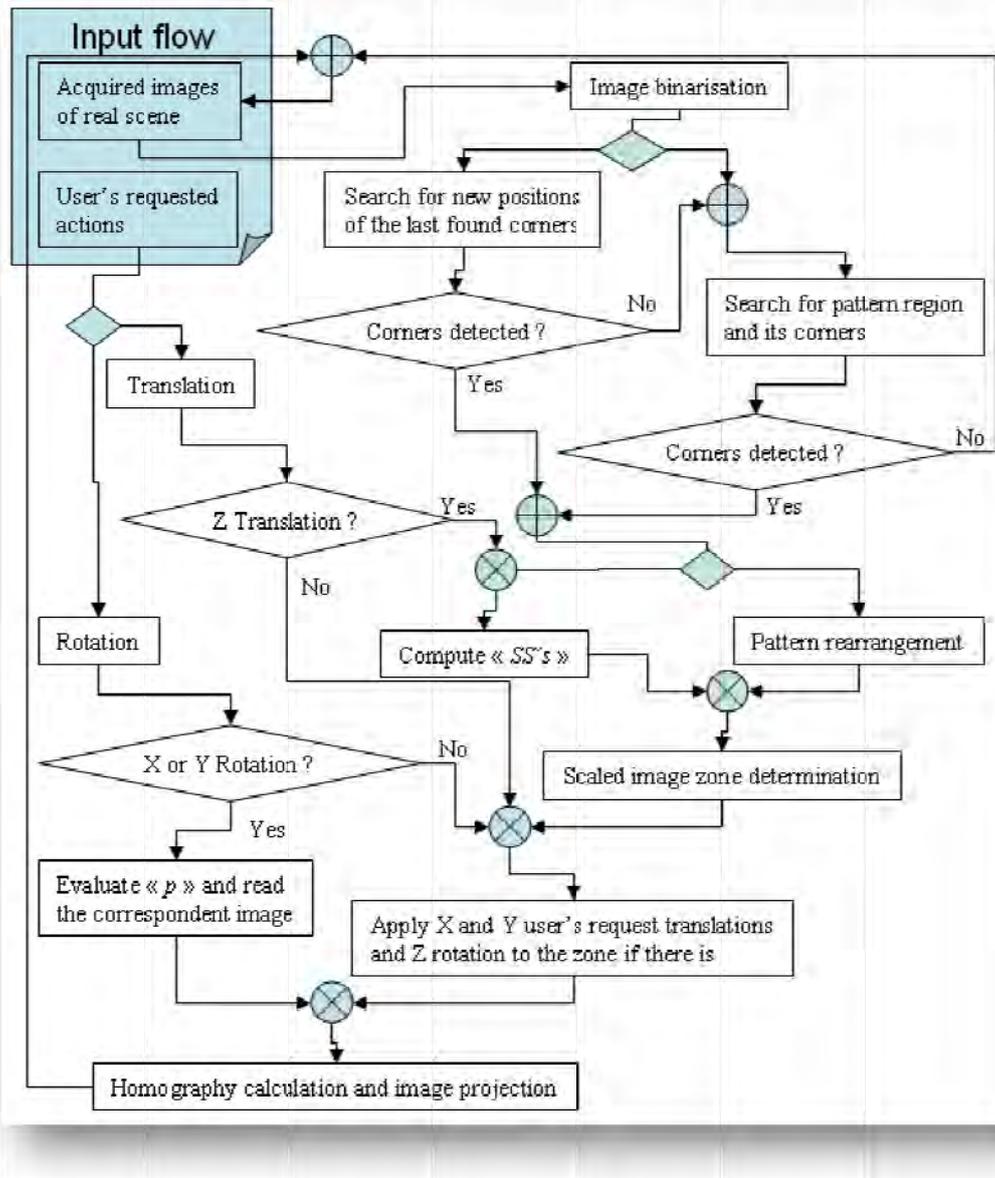


Figure IV.11. Diagramme de l'algorithme général de l'augmentation d'une scène avec les images d'un objet réel et permettant à l'utilisateur de le manipuler. Le symbole \otimes représente le 'et' logique et le \oplus représente le 'ou' logique.

6.2.1. Translations

Suite à une translation, un minimum de réalisme visuel exige un double effet sur l'image. Le premier c'est lorsque l'objet est déplacé. Sa taille doit être changée en fonction de la distance relative qui le sépare de la caméra. Cette taille doit visuellement changer pour simuler l'effet de zoom dû à la perspective. Ceci peut être réalisé par la connaissance de la position virtuelle de l'objet relativement au référentiel de la caméra. Le deuxième effet c'est qu'après chaque action résultant sur une translation, la vue utilisée, et par conséquent l'image associée, peut également changer. Cela signifie qu'une rotation implicite est appliquée à l'objet virtuel qu'il est nécessaire de retrouver l'image correspondante.

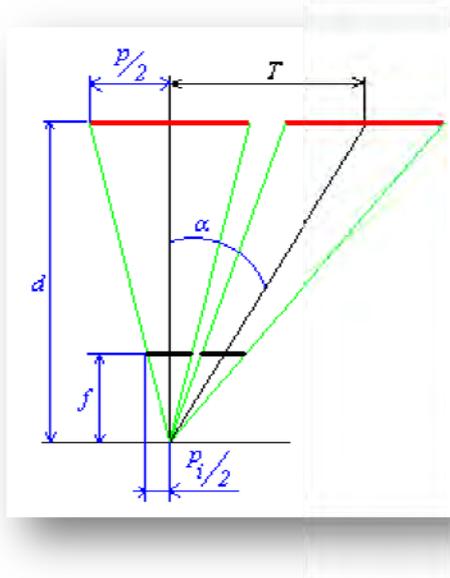


Figure IV.12. Bases géométriques associés au déplacement de l'objet le long des axes Ox ou Oy

Les effets des translations le long des axes Ox et Oy sont similaires, à l'exception que chaque sens de déplacement agit respectivement sur les paramètres n et k (Figure IV.12). La valeur de la distance d qui sépare l'objet de la caméra permet d'estimer l'angle de vue et la nouvelle distance suite à une translation. Elle peut être évaluée par:

$$d = p \frac{f}{p_i} \quad (\text{IV.17})$$

p est la taille connue de l'objet, f est la distance focale et p_i est la taille de la région virtuelle détectée et réarrangée.

Notons que pour utiliser les paramètres de cette expression, il est indispensable de déterminer les paramètres intrinsèques de la caméra, en l'occurrence les paramètres 3-D de la matrice de projection. La valeur de p_i est estimée dans la même unité que p en utilisant la taille des pixels déduite par les paramètres de calibration. La taille des pixels est exprimée dans les directions horizontale et verticale.

A chaque position où l'objet est déplacé par une translation T , l'angle α résultant doit être estimé. Comme $\text{tg} \alpha = T/d$, il est facile de déduire une valeur de α où T est une valeur connue relativement à la requête utilisateur de translation. Ainsi, on peut déduire (IV.18):

$$\text{tg} \alpha = p_i \frac{T}{p \cdot f} \quad (\text{IV.18})$$

Une fois estimée, la valeur de α donne celle de $n' = \alpha/st$ ou celle de $k' = \alpha/st$ suivant que la translation est respectivement faite le long de Ox ou de Oy . Cette valeur est algébrique et peut être négative et donc pouvant engendrer des valeurs de n' et de k' négatives. C'est pourquoi, ces valeurs de n' et de k' estimées doivent être ajoutées à

celles de n et k actuelles. Bien évidemment, les valeurs estimées de n' et k' doivent suivre les mêmes principes cités dans l'expression (IV.6) pour les intervalles de valeurs.

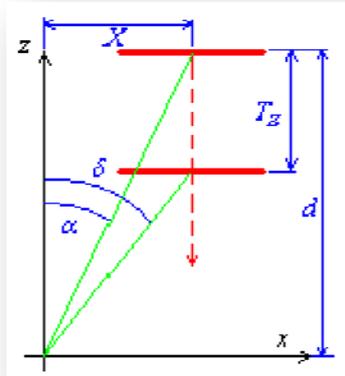


Figure IV.13. Bases géométriques associées au déplacement de l'objet le long de l'axe Oz

Les translations en z prises en considération dans ce travail sont celles qui affectent seulement le paramètre n en plus du zoom résultant exprimé par (IV.16). Ceci est vrai sous l'hypothèse que la translation accomplie est dans le plan (xz) (Figure IV.13). Ainsi, la nouvelle valeur de n doit être évaluée dans le but de déduire l'image à projeter. Supposons que l'indice est situé à une position X relativement de l'axe Ox faisant un angle α avec le plan (yz) . Cette position est exprimée par $X = d \cdot \operatorname{tg} \alpha = (d - T_z) \operatorname{tg} \delta$. Ceci permet d'estimer la valeur de l'angle δ par (IV.19):

$$\operatorname{tg} \delta = \frac{d \cdot \operatorname{tg} \alpha}{(d - T_z)} \Rightarrow \delta = \operatorname{arctg} \left(\frac{d \cdot \operatorname{tg} \alpha}{(d - T_z)} \right) \quad (\text{IV.19})$$

De même que pour la translation le long des axes Ox et Oy , l'angle δ devrait permettre de déduire directement la nouvelle valeur de $n = \delta / \operatorname{st}$.

6.2.3. Rotations

La simulation d'une rotation complète d'un objet réel virtualisé dans tous les sens est accomplie à travers la navigation dans l'ensemble des images associées. Cette navigation est complétée par le rendu de l'image qui reflète l'orientation courante requise par l'utilisateur. En effet, les requêtes utilisateur doivent être transformées en des positionnements de la caméra virtuelle supposée filmer l'objet. Pour ce faire, le système devrait lier entre les images prises et les points de vue de la caméra. Comme les images de l'objet ont été générées par une caméra réelle à partir d'orientations particulières, celle qui est choisie est simplement projetée après avoir été tournée pour concorder avec la direction du point de vue virtuel courant.

A la différence des rotations possibles présentées dans la section 5.1.2 qui sont restreintes aux images exactement telles qu'elles ont été prises par la caméra et suivant le chemin parcourue par cette dernière, une simulation de rotations réelles doit donner la

possibilité de faire tourner l'objet dans n'importe quel sens possible en réalité. Pour cela, comme un parcourt à base des paramètres n et k ne peut pas fournir une telle possibilité, une étude plus approfondie est indispensable. De nouveaux concepts utilisant de nouvelles structures complétant les premiers sont nécessaires. La section suivante donne le détail de la solution trouvée.

7. Rotations libres à base d'images

Dans le but de lier les images prises aux besoins de rotation de l'utilisateur, deux systèmes de coordonnées particuliers qui sont ortho-normaux, 3-D et droits sont utilisés (Figure IV.14) [11]. Ils sont supposés être positionnés au centre de gravité de l'objet relativement à un référentiel de base fixe. Le principe repose sur le fait que lorsque l'objet tourne autour de son centre de gravité, ces deux systèmes tournent également mais dans des directions opposées. Un des deux est considéré lié à l'objet. Il se déplace et tourne dans le même sens que l'objet lui-même. C'est le référentiel de l'objet (RO ou ORF pour Object Reference Frame). Il reflète n'importe quelle orientation courante de l'objet par rapport à une position initiale. Le sens positif de l'axe Oz est toujours orienté vers l'image de référence. L'autre système de coordonnées simule une caméra virtuelle. Le sens positif de son axe Oz pointe toujours vers le point de vue de la caméra virtuelle (PVC ou CVP pour Camera View Point) quand elle tourne autour de l'objet. Ainsi, le sens de l'axe focal de la caméra et celui de CVP sont dans la même direction mais de sens opposés. Par conséquent, la direction de l'axe Oz de CVP par rapport à la position initiale fournit l'image qui reflète la vue courante.

Pour simuler les rotations de l'objet, CVP est considéré tourner à l'intérieur de l'objet qui est assumé sans mouvement. En effet, faire tourner l'objet autour de son centre est équivalent à faire tourner la caméra autour de lui. Plus précisément, les rotations de CVP sont équivalentes à faire tourner l'environnement autour de l'objet tout en ayant un regard orienté vers le centre de cet objet. C'est pourquoi, CVP tourne dans le sens opposé de celui de ORF. Ainsi, les coordonnées sphériques de l'axe Oz de CVP relativement à la position initiale fournissent l'information requise pour l'extraction de l'image prise par la caméra réelle à une telle position.

D'un point de vue géométrique, il existe trois systèmes de coordonnées liés entre eux par une relation de rotations relatives nécessitant d'être exprimée analytiquement. De façon explicite, les deux systèmes cités précédemment tournent et sont référencés dans l'espace régi par son système de coordonnées (SCE ou WCS pour World Coordinate System) qui est supposé fixe et où le centre de gravité de l'objet est supposé être positionné à son origine. C'est pourquoi, l'expression analytique de la relation entre les systèmes de coordonnées devrait nous permettre de localiser l'image qui correspond à une requête de l'utilisateur et de la projeter dans l'orientation adéquate. Cela signifie qu'il faudrait exprimer des rotations combinées de l'objet virtuel autour d'axes quelconques parce que le but du système est de permettre de visualiser l'objet dans n'importe quelle direction.

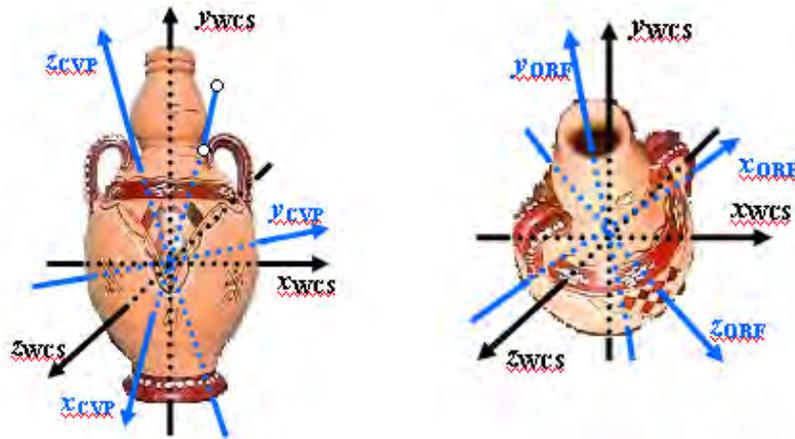


Figure IV.14. Les deux systèmes de coordonnées utilisés pour le contrôle de la rotation. L'image de gauche reflète la rotation de CVP qui correspond à la vue de l'image de droite où ORF est orienté avec l'objet

Plus précisément, d'un point de vue analytique, deux matrices rotation représentent CVP et ORF et sont appelées respectivement R_v et R_o . Elles sont mises à jour dans le même sens que CVP et ORF sont supposés tourner. Ainsi, dans un but d'utiliser ces principes et d'établir une relation analytique générale, une analyse est faite pour comprendre comment est ce que ces systèmes travaillent ensemble via une interprétation des premières prises d'images suivant la nouvelle sémantique de CVP et ORF.

7.1. Interprétation de l'opération de prise d'images

Il est possible de modéliser deux réflexions différentes mais équivalentes dépendant du fait que soit c'est la caméra, soit c'est l'objet qui tourne lors de l'opération initiale de prise d'images. Si c'est l'objet qui est fixe, il est évident que la caméra tourne autour de l'axe Oy de WCS pour les prises dans le sens de la longitude et autour de l'axe Ox de CVP, qui est également en rotation pour suivre la direction de la caméra, pour les prises dans le sens de la latitude. Si c'est la caméra qui est fixe, l'objet devrait tourner autour des axes Oy et Ox des systèmes de coordonnées ORF et WCS respectivement pour pouvoir prendre ses images. Ces rotations sont faites pour assurer respectivement des prises dans le sens de la longitude et de la latitude. Dans ces deux cas, ORF tourne en même temps avec l'objet.

En conséquence, projeter des images successives dans l'ordre de leurs prises dans la direction de la longitude est équivalent à faire tourner l'objet autour de son axe Oy , i.e. ORF tourne autour de son propre axe Oy et CVP autour de l'axe Oy de WCS. En suivant le même principe de réflexion, pour projeter les images dans la direction de la latitude, l'objet, et donc ORF, devrait tourner autour de l'axe Ox du système de coordonnées WCS et CVP autour de son propre axe Ox . Pour les deux cas cités, la constatation que ORF tourne dans une direction opposée que CVP est évidente et donc que l'objet tourne dans une direction opposée que celle de la caméra. Finalement, la rotation de l'objet autour de l'axe Oz est simplement la rotation de l'image projetée.

En se basant sur l'analyse précédente, nous pouvons établir la conclusion suivante. Lorsque CVP tourne autour des axes de WCS, ORF tourne dans une direction opposée autour de ses propres axes similaires à ceux de WCS autour desquels tourne CVP. L'inverse est également vrai. En ce sens, les rotations de ORF autour des axes WCS entraîne la rotation de CVP dans une direction opposée autour de ses axes.

Pour faire une analogie avec ce qui a été présenté en section 5.1.2, il est facile de constater que les rotations précédemment présentées sont identiques à celles utilisées pour projeter les images de l'objet dans le même sens qu'elles ont été prises. Cependant, ces rotations ne couvrent pas les trois degrés de liberté d'une rotation libre parce qu'elles sont limitées. De manière spécifique, ce sont des rotations autour des axes Oy et Ox de ORF et CVP respectivement et elles sont nettement insuffisantes pour satisfaire toutes les vues possibles. La situation la plus apparente est celle de ne pas pouvoir faire tourner l'objet dans une orientation particulière. C'est la raison pour laquelle le raisonnement établi devrait être généralisée pour les autres axes.

7.2. Rotations de l'objet autour des axes de ORF

Comme ORF tourne avec l'objet, les rotations autour de ses axes référencées dans WCS devraient être estimées puis interprétées. Elles peuvent être considérées comme étant appliquées autour d'axes arbitraires où chacun est considéré colinéaire avec ceux de ORF. Puisque ORF est représenté par la matrice R_o , la rotation de l'objet autour de l'un des axes de ORF peut être évaluée en appliquant une rotation aux axes de ORF autour de l'axe correspondant de R_o . En d'autres termes, tourner l'objet autour d'un axe de ORF constitue une rotation autour d'un axe particulier dont les coordonnées sont les éléments de la colonne de R_o associée. Particulièrement, sachant que les colonnes des matrices rotation représentent les coordonnées des vecteurs unitaires de l'espace obtenu relativement aux axes de l'espace d'origine, R_o est simplement tournée autour du vecteur $R_{o_{\overline{axe}}}$ qui représente la première, la seconde ou la troisième colonne de R_o , dépendant de la valeur de \overline{axe} , qui peut être \vec{x} , \vec{y} ou \vec{z} . Cela signifie que les colonnes de la matrice R_o représentent les coordonnées des vecteurs unitaires de ORF dans WCS et qu'ils constituent des axes quelconques relativement à WCS après avoir appliqué des rotations à l'objet. De manière explicite:

$$R_o = R_{o_{\overline{axe}}}^\alpha \cdot R_o \quad (\text{IV.20})$$

La notation $R_{o_{\overline{axe}}}^\alpha$ représente la matrice rotation autour de l'axe $R_{o_{\overline{axe}}}$ par un angle α (voir chapitre III, section 4.2.2), c'est-à-dire:

$$R_{o_{\overline{axe}}}^\alpha = R_{x_{\overline{axe}}}^{-1} \cdot R_{y_{\overline{axe}}}^{-1} \cdot R_{z_{\overline{axe}}} \cdot R_{y_{\overline{axe}}} \cdot R_{x_{\overline{axe}}} \quad (\text{IV.21})$$

De plus, quand le référentiel ORF tourne, CVP devrait faire de même. Malgré que la réflexion concernant les rotations autour des axes de ORF est facile en raison du comportement naturel du référentiel vis-à-vis de l'aspect visuel, certaines directions de vues de CVP en relation avec les rotations de ORF et qui sont difficiles à déduire

devraient être trouvées. La plus évidente, présentée à la section précédente, est la rotation de CVP autour de l'axe Oy de WCS quand l'objet tourne autour de son propre axe Oy . Par conséquent, les rotations autour des axes Ox et Oz doivent être comprises.

Pour trouver la relation qui lie les rotations de ORF autour de ses axes avec celles de CVP, il est possible de vérifier empiriquement que CVP tourne autour des mêmes axes de WCS afin de pointer vers la direction du point de vue de la caméra virtuelle. Normalement, de la même façon que ORF, CVP tourne autour du même axe mais après avoir été transformé par R_v qui est inverse de R_o dans ce cas. Cette réflexion est aussi basée sur la conclusion établie dans la précédente sous-section en raison de la rotation qui est faite autour de l'un des axes de ORF. Particulièrement, quand l'objet est requis à tourner α degrés autour de l'un des axes de ORF, CVP devrait tourner par $-\alpha$ autour du même axe mais celui de WCS. En format matriciel (IV.22), tourner les axes de CVP autour de ceux de WCS est équivalent à multiplier la matrice R_v par une matrice rotation autour d'un des axes de WCS:

$$R_v = R_{\overrightarrow{axe}}^{-\alpha} \cdot R_v \quad (\text{IV.22})$$

$R_{\overrightarrow{axe}}^{-\alpha}$ est la matrice rotation autour de l'axe Ox , Oy ou Oz de WCS par un angle $-\alpha$.

7.3. Rotation de l'objet autour des axes de WCS

Quand l'objet fait une rotation autour des axes de WCS, ORF effectue la même rotation. Ainsi, chaque rotation autour de n'importe quel axe de WCS par un angle α est directement exprimé par:

$$R_o = R_{\overrightarrow{axe}}^{\alpha} \cdot R_o \quad (\text{IV.23})$$

A base de la sous-section précédente, il est facile de déduire que CVP tourne autour de ses propres axes dans une direction inverse de celle faite par ORF autour de ceux de WCS (IV.24). Ainsi, en faisant la même réflexion que précédemment, la rotation de CVP est équivalente à faire tourner WCS par R_v et ensuite tourner CVP autour de l'axe d'intérêt. En effet, ce type de rotation est exactement l'inverse de celui qui est fait autour des axes de ORF. C'est pourquoi, CVP est tourné autour de $R_{v_{\overrightarrow{axe}}}$ par $-\alpha$ qui est l'une des colonnes de R_v dépendant de la valeur de \overrightarrow{axe} qui représente l'un des trois axes Ox , Oy ou Oz :

$$R_v = R_{v_{\overrightarrow{axe}}}^{-\alpha} \cdot R_v \quad (\text{IV.24})$$

7.4. Rotation de l'objet autour d'axes arbitraires

Nous devons généraliser la rotation de sorte qu'elle soit appliquée autour d'axes arbitraires. Les rotations précédentes deviennent alors des cas particuliers. Notons que le principe de la rotation de CVP par $-\alpha$ quand l'objet tourne par α reste toujours vrai.

A travers la présentation des rotations de l'objet autour d'axes appartenant à des systèmes de coordonnées différents, il est maintenant possible d'établir, à base de résultats expérimentaux, que ORF tourne directement autour d'axes arbitraires. En effet, n'importe quel axe dont l'objet est requis de tourner autour est référencé dans WCS. Par conséquent, ceci est similaire au cas de le faire tourner autour des axes de ORF, en considérant ses axes comme des cas particuliers. La différence est que n'importe quel autre axe arbitraire n'est colinéaire avec aucun axe de ORF. En forme matricielle, nous obtenons (IV.25):

$$R_o = R_{W_{AA}}^\alpha \cdot R_o \quad (\text{IV.25})$$

$R_{W_{AA}}^\alpha$ est une matrice rotation exprimant la rotation de l'objet autour de l'axe arbitraire \vec{AA} d'un angle α .

Dans le même sens de réflexion, la déduction que CVP tourne autour de l'axe transformé par R_v peut être faite sans grand effort. Ceci est évident puisque nous savons qu'au même moment quand ORF tourne autour de l'un des axes de WCS ou de ORF, CVP tourne respectivement autour de ses propres axes ou ceux de WCS. Ceci peut être interprété comme si l'axe est l'un de WCS ou de ORF transformé par R_v . Par conséquent, si l'axe est à une position arbitraire référencée dans WCS, il doit être transformé par R_v avant d'appliquer la rotation. Mathématiquement, l'opération consiste à transformer l'axe par la rotation R_v ensuite évaluer la rotation requise. Explicitement, la matrice R_v devient:

$$R_v = R_{W_{TAA}}^{-\alpha} \cdot R_v \quad (\text{IV.26})$$

Avec:

$$TAA^T = R_v \cdot AA^T \quad (\text{IV.27})$$

7.5. Réalisation de la simulation

La simulation du mouvement de l'objet est accomplie à travers la liaison entre les images et les orientations courantes de ORF et CVP. Les actions de l'utilisateur affectent les orientations des axes qui sont utilisés pour l'extraction de l'image à projeter.

Cette liaison entre l'orientation de ces systèmes et les images stockées est faite en deux étapes. La première étape est de localiser l'image réelle en relation avec le point de vue actuel. Ceci peut être fourni par l'orientation de l'axe Oz de CVP, i.e. la troisième colonne de R_v . La deuxième étape consiste à estimer l'angle de l'orientation qu'il faudrait appliquer à l'image pour qu'elle reflète effectivement la rotation escomptée. Souvent, ceci n'est pas un résultat direct. Il faut quantifier certaines rotations particulières qui doivent être déterminées. Ceci est simplement l'angle fait par l'axe Oy de ORF par rapport au plan Oyz de WCS.

L'accès à une image particulière, comme nous l'avons déjà indiqué, passe par l'estimation des paramètres n et k . Sachant que l'axe Oz de CVP est orienté vers la direction de la caméra virtuelle, son orientation simultanée, relativement aux plans Oyz et Oxz de WCS, est utilisée pour déterminer respectivement les valeurs de n et k . Plus précisément, les coordonnées sphériques de la troisième colonne de la matrice R_v sont évaluées de sorte à pouvoir estimer les angles γ et β . Explicitement, en supposant que Oz de CVP, ou la troisième colonne de R_v , fasse un angle de ϕ avec le plan Oyz et un angle θ avec le Oxz de WCS, les valeurs de n et k peuvent être déterminées par l'expression (IV.6) ou celles données par (IV.28) et (IV.29). En effet, des rotations quelconques ne permettent pas d'obtenir des angles qui sont multiples entiers de st :

$$n = \text{mod} \left(\text{round} \left(\frac{\phi}{st}, \frac{2\pi}{st} \right), \frac{2\pi}{st} \right) \quad (\text{IV.28})$$

et

$$k = \text{round} \left(\frac{\theta}{st} \right) \quad (\text{IV.29})$$

où ϕ et θ doivent être conformes avec certaines contraintes. Particulièrement, les valeurs de θ sont changées en $\frac{\pi}{2} - \theta$ et celles de ϕ en $2\pi + \phi$ si $\phi < 0$.

La rotation de l'image dans la direction z peut se décider sur la base de l'étude de l'inclinaison de l'axe Oy de ORF relativement au plan Oyz de WCS. En effet, l'axe Oz de WCS est orienté vers la direction de la vue de l'utilisateur. Ainsi, l'objet devrait apparaître tourné si son axe vertical est incliné par rapport au plan vertical Oyz de WCS. Par conséquent, la seconde colonne de la matrice R_v est utilisée pour estimer cette inclinaison. L'inclinaison de l'axe Oy par rapport au plan Oyz exprime simplement celle de l'objet tout entier. Les Figure IV.15 et Figure IV.17 montrent un objet tournant autour de deux axes différents. L'utilisation des images originales sans prétraitement dans ces figures a pour but de montrer l'inclinaison induite par une rotation particulière déduite de celle de ORF.

8. Discussion des résultats

Les résultats de ce chapitre fournissent deux principaux résultats. La solution mathématique proposée peut être considérée comme une nouvelle donnée pour les résultats précédents de la recherche traitant des systèmes de rendu basés sur les images spatiales. La structure linéaire est un autre résultat qui supporte plus facilement les besoins du domaine que les structures précédentes.

Particulièrement, la plupart des travaux définissent des chemins restreints que l'utilisateur est contraint de suivre durant la simulation de la manipulation et qui sont généralement ceux suivi par la caméra durant les prises initiales. De plus, la seule façon dont les images sont projetées est le sens dans lequel elles ont été prises. Leurs rotations

sont limitées aux requêtes utilisateur explicites, cependant ce n'est généralement pas dans le but de simuler un mouvement naturel. Ce travail donne non seulement la possibilité de satisfaire ces requêtes, mais il établit en plus les bases pour accomplir des rotations naturelles indépendantes des prises initiales ainsi que du nombre d'images.

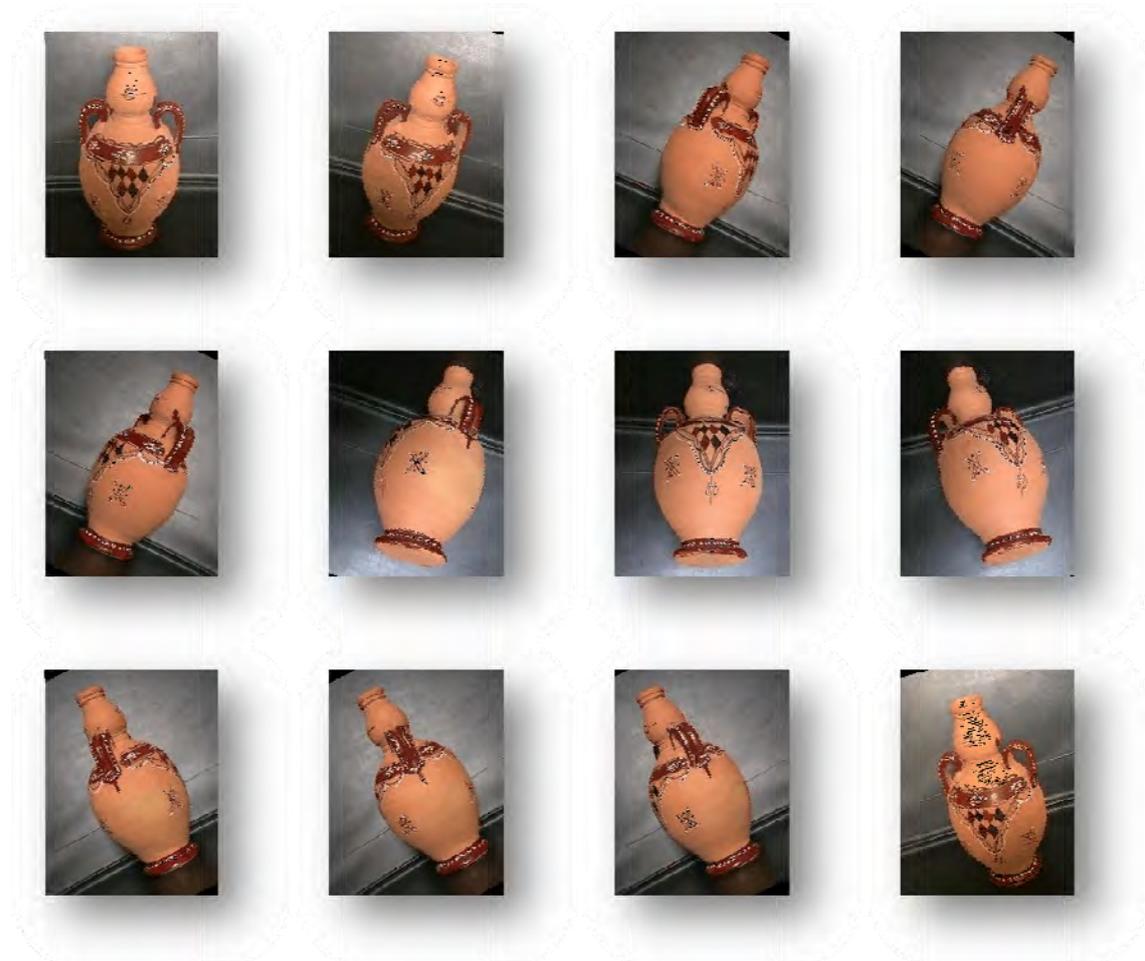


Figure IV.15. (De haut en bas et de gauche à droite) Rotation complète d'un objet autour de l'axe Oy de WCS avec un pas $st = \frac{\pi}{6}$ après une première rotation d'un même angle $\frac{\pi}{6}$ autour de l'axe Ox de ORF (première image)

De plus, la structure linéaire proposée fournit un avantage relativement aux autres structures. Elle permet de donner accès aux images dans une forme matricielle puisqu'elles sont organisées dans une structure multicouche. A travers les paramètres n , k et r de l'expression (IV.4), chaque couche r peut être manipulée comme un ensemble d'images en relation. Ceci est important pour une continuation de ce travail pour l'utilisation du principe d'interpolation pour la génération d'un mouvement naturel sans saut d'images. Les images qui entourent une image particulière sont simplement accédées à travers n et k . Pour cette raison, cette structure peut être bénéfique pour d'autres travaux. Notons également que malgré l'utilisation d'un fichier AVI, la structure est indépendante de n'importe quelle structure de fichier particulière. Ainsi, les algorithmes de compression les plus usuels peuvent être appliqués à notre structure où uniquement des directions particulières, relativement aux sens de prises d'images,

peuvent être exploitées dans certains travaux. Ceci est le cas, par exemple, pour [111] et [106].

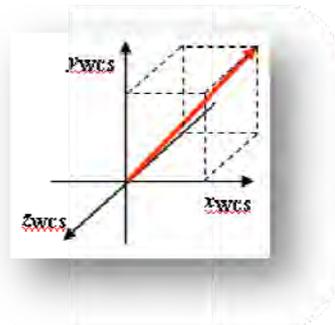


Figure IV.16. Axe de rotation utilisé pour la rotation de l'objet de la Figure IV.17

Malgré les avantages précédents, il existe certaines caractéristiques qui affectent négativement la qualité des résultats obtenus. Parmi ceux-ci, une vibration de l'objet peut être constatée durant sa rotation. Généralement, ce problème est dû à la qualité des appareils mécaniques utilisés durant l'opération de prise initiale. L'impact de ceci est le déplacement du centre de gravité de l'objet filmé de sa position d'une image à une autre. Un autre paramètre qui affecte négativement la qualité de la rotation est le bruit dans les données estimées. C'est pourquoi, la méthode présentée par [133] est utilisée pour l'estimation de la meilleure matrice rotation pour compenser le bruit et les erreurs cumulées de la multiplication matricielle.

9. Conclusion

Ce chapitre est une synthèse de la majorité de nos travaux relatifs à l'augmentation avec des objets réels virtualisés et à leur manipulation. Il commence par faire une présentation du principe de prise d'images et des contraintes associées, ensuite il discute la structure utilisée pour le stockage des images prises. Particulièrement, la réflexion utilisée dans la déduction de l'expression de la relation d'ordre des images et de la possibilité de son extension à un système multi-résolution sont détaillées. Une autre partie assez importante traite de la manipulation des objets après leur intégration. Elle est présentée sous deux aspects. Le premier se veut être un tour de la simulation démunie de sentiment de réalisme visuel. Le deuxième par contre s'intéresse à la possibilité d'avoir une manipulation visuellement réelle. Dans cette partie, une étude théorique d'un point de vue géométrique est détaillée pour montrer comment est ce que les objets réels virtualisés peuvent être manipulés librement. Une démarche constructive basée sur la géométrie, la logique et l'empirisme est suivie afin de montrer les résultats obtenus. Finalement, une discussion des résultats est faite dans un sens comparatif aux travaux du même domaine. Particulièrement, la rotation des objets virtuels et la structure linéaire utilisée pour le stockage des images sont considérées dans cette comparaison.

Ce travail présente certaines limites qu'il est préférable de devancer. L'une des perspectives essentielles pour le travail futur, en plus de celles déjà mentionnées pour chaque partie du chapitre, est celle de donner une solution complète naturelle englobant

les translations. En effet, lorsque l'objet virtuel est translaté le long de l'axe Oz suivant un plan quelconque autre que (xz) , les deux paramètres n et k peuvent changer simultanément. L'impact de ce changement sur l'objet est d'avoir un effet de rotations implicites suivant deux axes simultanément. Dans le même sens, faut-il aussi faire une étude plus approfondie quant à l'outil 2-D utilisé pour permettre une manipulation supportant des rotations et des translations simultanées.

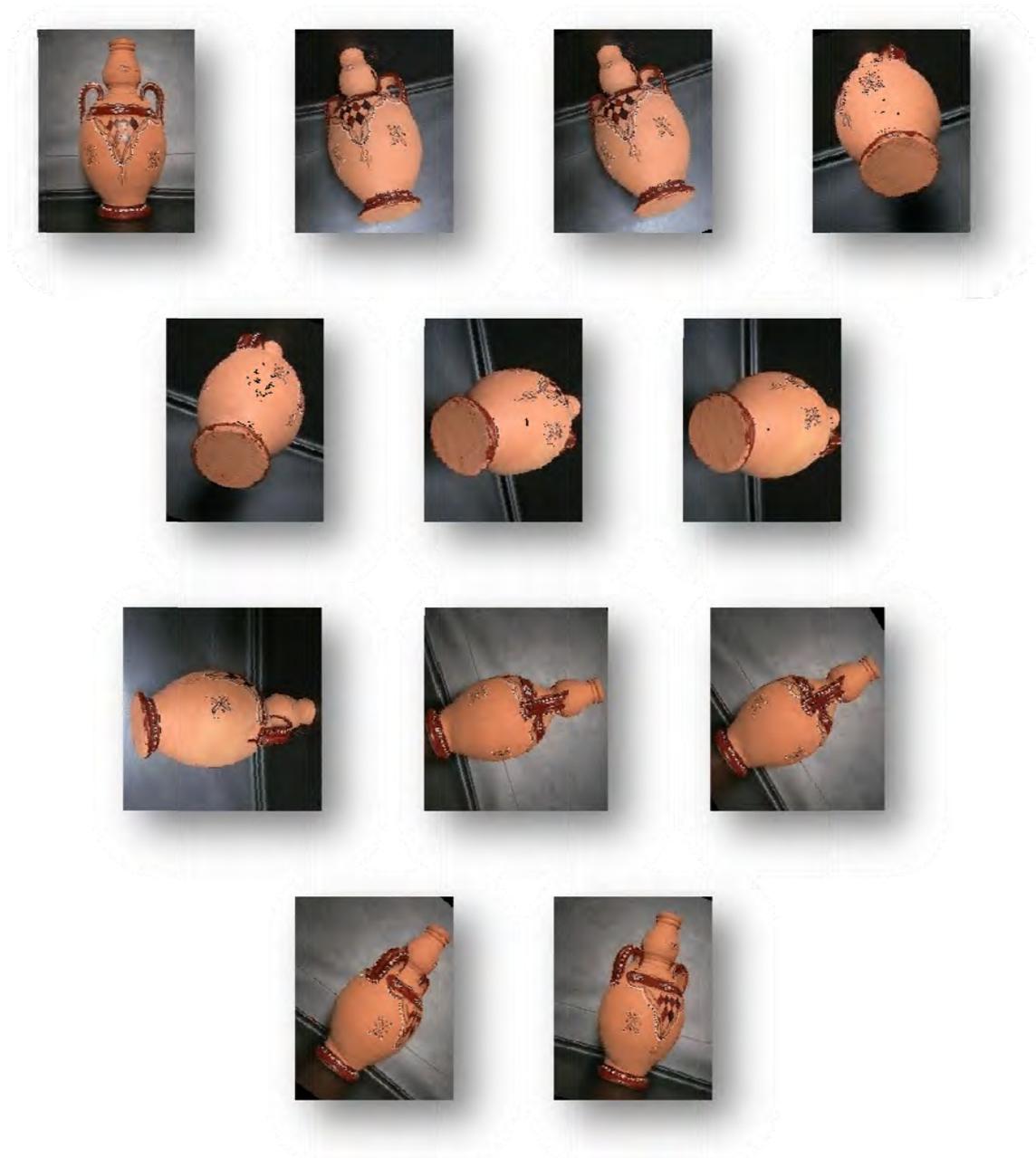


Figure IV.17. (De haut en bas et de gauche à droite) Rotation complète d'un objet autour d'un axe simulé présenté à la Figure IV.16 et utilisant un pas $st = \frac{\pi}{6}$

Une autre extension assez importante peut fournir une maîtrise plus générale de l'objet manipulé. C'est l'étude de la possibilité de manipuler l'objet dans le sens de la rotation relativement à un point autre que son centre de gravité. Ce concept étant totalement

contrôlé pour les objets synthétiques, il n'en est pas le cas pour les objets réels virtualisés.

Dans le but de clore cette étude, le prochain chapitre généralise les résultats obtenus au collaboratif. C'est une partie essentielle dans nos travaux puisque le thème de notre recherche tourne autour. Il présente le prototype d'une architecture logiciel permettant d'implémenter les résultats de ce chapitre afin d'être exploités par un ensemble d'utilisateurs distants.

The page features a decorative graphic consisting of three blue circles of varying sizes, each with a lighter blue ring around its center. These circles are arranged vertically and are connected by thin blue lines that extend from the top-left and bottom-right corners of the page towards the center, creating a sense of depth and movement.

Chapitre V

Manipulation dans un Environnement Collaboratif

Ce dernier chapitre généralise les résultats présentés au chapitre précédent à un environnement collaboratif sous forme d'une architecture répartie.

Manipulation dans un Environnement Collaboratif

1. Introduction

L'augmentation d'une scène par des objets virtuels en temps réel sans dégrader son réalisme visuel est généralement renforcée par la possibilité de manipuler ces objets afin de compléter le sentiment de réalisme de l'utilisateur par des gestes physiques. Cet aspect est d'avantage naturel lorsqu'il est enrichi par la possibilité de les manipuler de manière partagée dans la scène augmentée. C'est ce partage qui fait de l'augmentation de la scène collaborative. Le besoin de collaborer est inhérent à la nature de la majorité des tâches accomplies en réalité augmentée tout comme la plupart des opérations qui nécessitent une collaboration humaine. L'intérêt majeur serait de réaliser des tâches collectives utilisant ces objets. Ainsi, la collaboration dans de tels environnements concerne particulièrement les objets virtuels et consiste à permettre à plusieurs utilisateurs d'interagir à travers et avec ces objets qui deviennent partagés. Selon la nature de la tâche, cette collaboration peut être coopérative où les différentes tâches accomplies sont complémentaires pour l'accomplissement d'une tâche principale.

Comme déjà défini pour un environnement de réalité augmentée, l'utilisateur reste toujours en contact avec son environnement. Dans le cas du collaboratif, l'interaction et la communication naturelle des utilisateurs sont souhaitables afin de ne pas les isoler de leur milieu naturel. C'est pourquoi le collaboratif peut jouer un rôle important dans l'enrichissement de la sémantique de la communication et de la compréhension de l'environnement. La réalité augmentée collaborative constitue alors une nouvelle forme d'interface et d'interaction. En effet, les utilisateurs ne sont plus restreints aux simples descriptions de leurs idées, mais sont en mesure de les visualiser. C'est ce qui constitue l'objectif de la tendance actuelle des recherches pour l'implémentation du principe de la manipulation dans un cadre collaboratif.

La coprésence ou co-localisation des utilisateurs, c'est-à-dire leur partage du même espace physique de travail, pour la sémantique d'une telle collaboration n'est pas une condition nécessaire. Au premier chapitre, section 4.5, nous avons définis trois catégories de classes de systèmes de réalité augmentée collaborative basées sur la distance séparant les différents utilisateurs entre eux ainsi que chacun d'eux par rapport à l'objet virtuel ou la scène augmentée. La différence essentielle réside dans la façon dont sont vus et manipulés les objets virtuels. Chaque utilisateur dispose d'une vue propre des objets virtuels dépendant de son point de vue, de la fonction de sa tâche et de sa position relative à la scène augmentée. C'est alors que si les utilisateurs se partagent le même espace de travail, le principe stipule l'existence d'une caméra ou plus qui correspondent au point de vue de chaque utilisateur. Par contre, lorsque les utilisateurs sont distants, un seul point de vue est généralement partagé par l'ensemble des

utilisateurs et qui est généralement généré par une seule caméra. La scène se trouve alors partagée et principalement les objets virtuels selon ce seul point de vue. C'est l'exemple d'une collaboration d'un expert distant avec un apprenti pour la réparation d'un système compliqué [126, 127].

Afin de concrétiser les idées, on peut imaginer deux scènes différentes en réalité augmentée où des utilisateurs collaborent à travers des objets virtuels. La première est celle de plusieurs architectes réunis autour d'une table discutant les différentes solutions de construction sur un terrain. Au lieu de faire des esquisses ou des maquettes de construction, ils leur seraient possibles de modéliser et de manipuler les structures 3-D puis de les adapter visuellement au terrain réel. Leurs actions exploiteront une séquence vidéo du terrain prise en direct. Dans une telle situation, il est évident que l'aspect visuel de l'augmentation est un facteur favorisant la solution optimale et est plus expressif. La deuxième scène est celle d'un vendeur de meubles présentant sur Internet ses produits pouvant être consultés par des clients distants. Ces derniers, possédant des caméras simples ou disposées sur des HMDs, peuvent consulter virtuellement et en direct les différentes dispositions possibles des meubles chez eux. Les actions peuvent même se traduire en une collaboration entre le vendeur et le client pour trouver une disposition optimale. Ainsi, il serait possible de choisir le meuble le plus adéquat sans l'avoir au préalable ramener chez soi.

L'un des objectifs essentiels de ce travail est d'apporter une nouvelle idée au concept d'e-commerce basé sur la réalité augmentée qui est actuellement entrain de prendre de l'ampleur [58]. Au moment où les applications actuelles sont développées à base d'objets synthétiques, notre tendance est de pouvoir proposer des images d'objets réels tels que présentés au chapitre précédent. Il serait possible à un client de visualiser des objets chez soi de manière virtuelle sans avoir à les déplacer. Dans ce cas, la scène est filmée chez le client au moment où son augmentation est faite par des objets stockés dans une base de données distante. Si ces objets virtuels sont représentés par des images d'objets réels, on serait capable de visualiser les objets comme s'ils étaient réels. La manipulation détaillée dans le quatrième chapitre serait d'un grand apport puisque le client serait capable de disposer l'objet tel qu'il le voit bien adapté. Il serait donc capable de décider de son adaptabilité avant même de l'acheter.

Le collaboratif peut être défini dans la taxonomie de [20] basé sur un classement à deux dimensions, un axe de temps (synchrone/asynchrone) et un axe d'espace (local/distant). Dans ce cadre, les travaux de réalité augmentée se sont principalement tournés vers les systèmes distants ou face-à-face (co-localisés) synchrones.

D'une manière générale, des techniques d'interactions spécifiques pour la réalité augmentée ont été introduites [98] et particulièrement pour les environnements collaboratifs exigeant une coprésence. Au troisième chapitre, section 5, une discussion étalée est consacrée aux techniques de manipulation des objets virtuels. Nous ne reprenons la discussion dans ce chapitre que d'un point de vue citation relativement au domaine du collaboratif.

Ce chapitre met le point sur une méthode simple d'augmentation permettant une collaboration de plusieurs utilisateurs à travers des objets virtuels. Il présente une architecture logicielle permettant l'implémentation de cette collaboration. Particulièrement, elle exploite une augmentation vidéo pour l'ensemble de tous les utilisateurs.

2. Principe de la collaboration

En complément à la réalité augmentée, dans [31] sont définies cinq composantes clés d'un environnement collaboratif:

- la virtualité: les objets sans existence réelle peuvent être vus et examinés dans le monde réel;
- l'augmentation: les objets réels peuvent être augmentés avec des annotations virtuelles;
- la coopération: plusieurs utilisateurs peuvent se voir l'un l'autre et coopérer de façon très naturelle;
- l'indépendance: chaque utilisateur peut contrôler son propre point de vue;
- l'individualité: l'information peut apparaître sous différentes formes choisies par l'utilisateur.

Dans le domaine du collaboratif, les recherches sont orientées en partie vers la possibilité d'interaction simultanée des utilisateurs avec les objets virtuels [49]. L'augmentation d'une scène est alors considérée comme une nouvelle façon de permettre une communication visuelle. Lorsque les objets virtuels et/ou la scène sont partagés par plusieurs utilisateurs, l'augmentation est dite collaborative. Dans ce contexte, une collaboration de plusieurs utilisateurs est accomplie pour la réalisation de tâches différentes ou communes. En ce sens, l'action par un utilisateur sur la scène augmentée est répercutée sur l'ensemble des utilisateurs suivant l'objectif et le point de vue de chacun [82].

Le principe utilisé pour la collaboration repose sur la possibilité d'effectuer les transformations géométriques - étudiés aux troisième et quatrième chapitres - sur les objets virtuels relativement à l'environnement de manière simultanée par plusieurs utilisateurs. Ces transformations vont avoir un impact sur les façons de leurs incrustations dans la scène. Chaque action sur un objet virtuel par un utilisateur entraîne la modification de la vue de l'objet aussi bien pour celui qui a effectué l'action que pour les autres utilisateurs. C'est pourquoi, la création d'une situation collaborative nécessite la définition de trois types d'actions principales:

- manipuler et ajouter des objets dans l'environnement: c'est une action obligatoire qui n'existe pas obligatoirement dans un environnement de réalité augmentée ordinaire.
- visualiser le résultat: avoir différents points de vue dans la scène, chacun spécifique à un utilisateur particulier.
- réaliser des simulations: agir sur les objets virtuels et simuler en temps réel leurs déplacements.

A base de ces types d'actions, il devient clair que le domaine du collaboratif nécessite principalement une adaptation des concepts déjà définis de la réalité augmentée. Le premier point est associé à la manipulation qui n'est plus considérée du même point de vue telle qu'elle a été présentée précédemment. Les actions telles qu'elles ont été définies restent identiques mais elles nécessitent une synchronisation. Le deuxième et le troisième point sont associés à la gestion simultanée de plusieurs vues dont les modifications ne résultent pas forcément des actions de l'utilisateur concerné. On s'intéresse donc à la définition de techniques d'interaction répondant à ces besoins.

2.1. Gestion des utilisateurs

Lors de l'augmentation d'une scène dans un contexte collaboratif, les utilisateurs n'ont pas tous obligatoirement les mêmes objectifs ni les mêmes tâches. En conséquence de cette diversité, les utilisateurs ne nécessitent pas le même type d'information suivant leur rôle dans la session. Un instructeur souhaite manipuler et visualiser des informations différentes que celles d'un apprenti. Ceci conduit parfois à considérer des vues personnelles pour chaque utilisateur en plus des différentes apparences qui sont spécifiques au point de vue de chacun (Figure V.1). Cette personnalisation des vues peut être soit sous la forme d'une visualisation de la même donnée sous différents aspects (en fil de fer, en faces cachées, texturé), soit sous forme de données visuelles différentes.



Figure V.1. Visualisation d'objets différents pour deux utilisateurs différents

Pour mettre en place ce système, [98] par exemple, définit un degré d'intégration de l'utilisateur dans la session:

- observateur passif: l'utilisateur a un retour visuel sur l'action, mais ne peut interagir avec l'environnement.
- observateur actif: l'utilisateur a un retour visuel et peut agir sur l'environnement à faible échelle (navigation, immersion).
- participant actif: l'utilisateur peut interagir complètement sur le système (navigation et interaction).
- coordinateur: l'utilisateur a des droits privilégiés sur le système (modification de la configuration).

Pour chaque objet virtuel, un droit d'accès est spécifié pour chaque utilisateur. Ce droit est appliqué à la visibilité et à la manipulation. Pour cela, à chaque objet est défini un droit universel, de groupe et personnel. Appliqué à un jeu par exemple, des droits pour chaque joueur, suivant son rôle dans le jeu, sont définis.

Pour ce qui est de notre plateforme prototype présentée ci-après dans ce chapitre, tous les utilisateurs possèdent les mêmes droits. Notre objectif est plutôt de synchroniser la manipulation de l'objet virtuel entre plusieurs utilisateurs sans pour autant avoir une vue différente pour chacun.

2.2. Visualisation des objets et simulation de leurs mouvements

Les applications de réalité augmentée collaborative sont essentiellement de deux types. Celles qui se basent, dans la manipulation des objets virtuels, sur un système client-serveur et celles qui sont à base de modules indépendantes. Dans le premier cas, il existe toujours un module qui est sensé contrôler complètement les objets virtuels. Les autres utilisateurs constituent de simples spectateurs pour la scène augmentée sur le plan de la vision. Si plusieurs caméras sont utilisées, les différents points de vue sont les seules informations gérées au profit des autres utilisateurs. Dans le cas où une seule caméra est utilisée, une vue unique est gérée, ce qui détermine cette situation de l'appellation de collaboratif à l'exception des différentes actions simultanées possibles. En particulier, la pluralité des vues n'est possible que lorsque plusieurs caméras sont utilisées. Dans le second cas, la vue des objets virtuels de chaque utilisateur est construite de manière indépendante des autres utilisateurs. Il est même possible que chaque utilisateur dispose de ses propres objets virtuels spécifiques à sa tâche.

Dans l'un ou l'autre des cas, une communication entre les différents modules d'augmentation ou de visualisation est nécessaire. Ainsi, la visualisation des objets virtuels qui peut nécessiter ou non la gestion de plusieurs points de vues simultanées dépend principalement de la distance des utilisateurs et des caméras filmant la scène à augmenter et non de l'architecture logicielle utilisée. En d'autres termes, la structure des modules gérant les différents utilisateurs n'a pas d'effet sur la façon de visualiser les différents objets. Cette visualisation dépend des moyens matériels et de la répartition des utilisateurs.

2.2.1. Utilisateurs partageant le même espace physique

La majorité des systèmes collaboratifs existants reposent sur l'utilisation de HMD optiques en raison de leur simplicité relativement à l'objectif de la collaboration [50]. Dans ce cas, la coexistence des utilisateurs dans un même espace physique est une condition nécessaire. A chaque utilisateur est associé un module chargé de l'opération d'augmentation. Chaque module réalise l'opération dans sa totalité et pour cela il utilise les indices partagés de la scène. Une association directe est faite entre chacun des modules et une caméra - ou paire de caméras - du système. Ainsi, l'apparence des objets virtuels est adaptée au point de vue de chaque utilisateur. Au sens de l'augmentation visuelle, les modules n'ont pas besoin d'établir une communication à cause de l'indépendance de chacun. La seule communication nécessaire est en raison de la manipulation supportée. Elle est relative à l'échange des informations des actions des utilisateurs utilisées dans la mise à jour des vues spécifiques de chacun d'eux.

Un avantage principal pour de tels systèmes est la possibilité d'interaction naturelle des utilisateurs durant la session d'augmentation. Le désavantage, par contre, pour la plupart de ces systèmes est l'utilisation de traqueurs magnétiques, infrarouges ou autres. Ceci augmente le nombre de flux à manipuler et peut être source d'erreurs sinon de difficulté de traitement.

Une architecture client-serveur peut également être utilisée lorsque les utilisateurs se partagent le même espace de travail. Elle pourrait avoir lieu avec un module qui réalise le rendu visuel des objets virtuels associés aux vues relatives de chaque utilisateur afin que ce traitement soit centralisé. Ceci est valable pour un système d'augmentation vidéo lorsqu'un seul module assure l'opération d'augmentation et les autres modules ne font que reproduire visuellement l'augmentation faite. C'est une situation très similaire au cas d'utilisateurs répartis présentée dans la sous-section suivante.

2.2.2. Utilisateurs géographiquement répartis

Lorsque les utilisateurs sont distants, une seule vue peut être partagée et c'est obligatoirement un système client-serveur qui est utilisé. Ceci est dû au fait que la scène filmée ne peut exister que dans un seul emplacement. Elle ne peut donc être filmée que par une seule caméra. Un système vidéo classique à base d'écrans ordinaires, ou de moyens similaires, est généralement utilisé pour la visualisation de la scène augmentée pour les utilisateurs distants. L'utilisateur local par contre peut éventuellement utiliser un HMD. Mais, qu'il l'utilise ou non, c'est toujours le point de vue de sa caméra locale qui est utilisé dans l'augmentation. Ce type d'augmentation ne nécessite donc aucune synchronisation pour l'affichage. Le module d'augmentation doit simplement dispatcher la seule vue produite à l'ensemble des utilisateurs.

Il est possible qu'une situation similaire soit utilisée lorsque les utilisateurs ne sont pas distants. C'est le cas où la scène est filmée par une seule caméra. Cependant, cette situation impose obligatoirement une augmentation vidéo sans HMD. Le principe réside

dans le fait qu'un seul module accomplit l'opération d'augmentation dont le résultat est dispatché aux autres utilisateurs.

2.3. Manipulation collaborative

La manipulation, quant à elle, est possible qu'elle soit opérée par l'utilisateur local ou par des utilisateurs distants. Les outils d'interaction peuvent être différents en fonction de la distance de l'utilisateur à la scène augmentée. Un utilisateur local peut par exemple manipuler les indices d'augmentation eux même, où les utilisateurs distants ne peuvent le faire. Mais d'un point de vue de l'augmentation, les actions d'un utilisateur quelconque affectent sa vue et les vues des autres au même moment où les leurs affectent la sienne. Les modifications des vues doivent donc être gérées pour l'ensemble.

La gestion des utilisateurs pour la manipulation des objets virtuels utilise plusieurs solutions en fonction du type de l'environnement d'augmentation. Pour une étude approfondie dans le domaine, nous renvoyons à [82] pour une présentation exhaustive d'un cadre de travail dans un environnement virtuel distribué. Nous renvoyons également à [50] pour une présentation de l'aspect théorique d'une plate forme d'augmentation collaborative. Notons cependant que la manipulation, à l'inverse de la visualisation, nécessite obligatoirement des modules différents associés à chaque utilisateur afin d'intercepter les actions relatives. La communication inter-modules est également importante afin de pouvoir s'échanger les informations des requêtes relatives. L'architecture logicielle est généralement complexe et nécessite des études approfondies faisant appel des systèmes parallèles et/ou répartis. Comme notre objectif n'est pas celui d'aller en profondeur dans les concepts associés, nous nous sommes intéressés juste à l'implémentation d'une nouvelle idée permettant une collaboration des utilisateurs indépendamment des architectures logicielles sous-jacente. Les sous sections suivantes présentent les concepts associés à l'implémentation non pas d'un point de vue architectural pure mais d'un point de vue fonctionnel de l'augmentation.

2.3.1. Manipulation par des utilisateurs partageant le même espace physique

Les systèmes traitant de l'augmentation collaborative pour des utilisateurs partageant le même espace physique sont légèrement contraignants. Afin qu'ils puissent être fonctionnels, des modules indépendants traitant la manipulation de chaque utilisateur doivent être définis. De plus, les informations des actions de chaque utilisateur doivent être communiquées à tous les autres modules pour pouvoir mettre à jour les vues relatives. Ce type de conception impose obligatoirement une calibration spécifique de la caméra de chaque utilisateur. Notons également que des traqueurs, autres que visuels, sont généralement nécessaires dans ce genre d'applications. Ceci augmente le nombre de flux à manipuler et peut être source d'erreurs sinon difficulté de traitement. En effet, les champs magnétiques et infrarouges, par exemple, souffre du problème d'un large taux d'erreur dans le positionnement. Ce qui impose en général une manipulation restreinte à un environnement préparé interne d'un laboratoire pour pouvoir contrôler ces champs afin qu'ils soient sans parasites. Ces solutions sont également difficiles à généraliser à des environnements externes.

En plus du moyen de communication qui doit être défini s'il est par message, par sockets ou autre, la méthode de synchronisation doit également être spécifiée. Ceci est vrai particulièrement au niveau d'une requête utilisateur qui doit être traitée comme une entité indivisible. Des entrelacements éventuels entre deux requêtes différentes peuvent créer des incohérences dans leurs applications. Aussi, l'échange d'information entre les différents modules doit couvrir l'ensemble; en d'autres termes, chaque module doit pouvoir communiquer avec tous les autres modules.

2.3.2. Manipulation par des utilisateurs distants

Les systèmes où les utilisateurs sont distants sont moins contraignants mais aussi moins naturels. Ils reposent principalement sur des systèmes vidéo. Les utilisateurs se partagent habituellement une même vue dans laquelle sont manipulés des objets virtuels conformément aux actions de chacun. A travers un réseau, les différentes actions sont contrôlées et synchronisées afin de ne pas créer des situations conflictuelles. Plus précisément, elles sont contrôlées pour être conformes aux spécifications de la tâche et sont utilisées pour mettre à jour la scène générée. Dans ce contexte, il est évident que la calibration de la caméra est effectuée pour la seule scène augmentée et non pas pour chaque utilisateur à part.

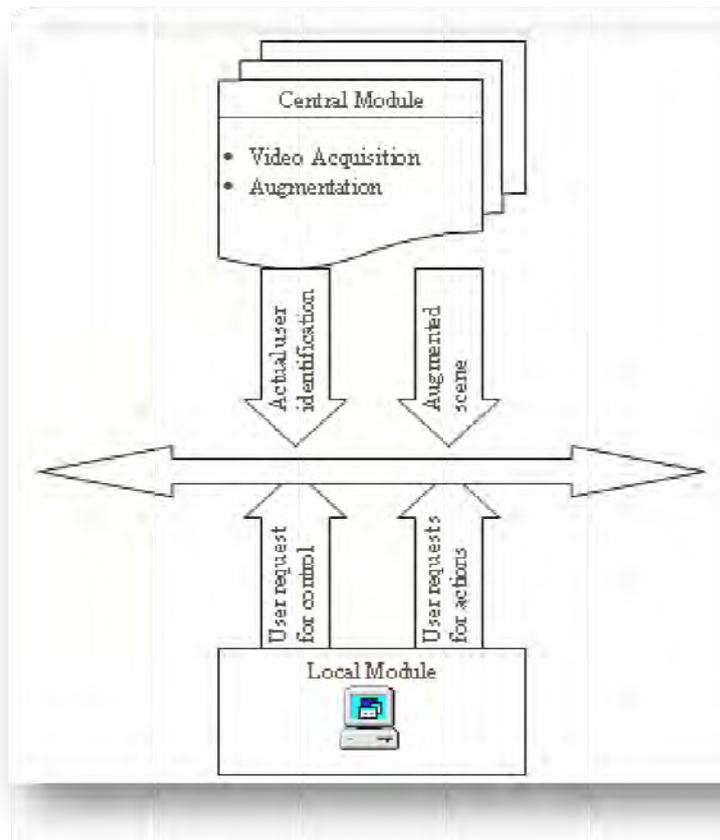


Figure V.2. Fonctions et relations des modules utilisateurs du prototype avec le module central

3. Prototype de la plate forme implémentée

Pour ce qui est de notre travail [13, 15], il consiste à faire collaborer plusieurs utilisateurs dans un réseau. Le concept implémenté utilise une solution vidéo à base d'écrans ordinaires. La scène à augmenter est filmée par une seule caméra auto-calibrée selon le principe présentée dans le second chapitre. L'augmentation de la scène est faite en utilisant un indice planaire inséré dans cette dernière. Les objets virtuels introduits dans la scène sont construits de manière synthétique ou représentent des objets réels virtualisés. La scène augmentée est partagée simultanément entre l'ensemble des utilisateurs à travers un réseau. Chacun d'eux dispose d'un écran pour la visualisation de l'augmentation qui lui parvient en même temps que les autres. A tout moment, chacun des utilisateurs peut interagir avec les objets virtuels en utilisant le clavier et la souris de son poste. C'est ce qui constitue la collaboration escomptée et nécessite une coordination entre les participants à l'aide de mécanismes de communication.

Le principe de la manipulation consiste en un ensemble d'actions sur le repère de l'objet virtuel dans le sens de la présentation faite au troisième chapitre. Ce dernier étant supposé initialement superposé à celui de l'indice de la scène où toute action d'un utilisateur quelconque est considérée comme une requête pour une translation ou une rotation relativement à l'un des axes selon le même principe utilisé par OpenGL et explicité au troisième ainsi qu'au quatrième chapitre. L'augmentation se fait alors à base des transformations géométriques appliquées en réponse aux actions des utilisateurs. Chacun d'eux peut agir sur le repère suivant un principe d'exclusion mutuelle pour assurer un traitement cohérent. Ce principe de synchronisation garanti que l'action entamée par un utilisateur masque toutes les autres actions sur l'objet virtuel jusqu'à sa libération. Par libération, nous insinuons la cession de réception d'évènements de la part de l'utilisateur détenteur ou possesseur actuel de l'objet. En ce sens, chaque utilisateur verrouille l'objet virtuel à son profit afin de pouvoir le manipuler (Figure V.2).

3.1. Architecture logicielle

L'architecture utilisée est basée sur un ensemble de modules égal au nombre d'utilisateurs faisant partie de la session d'augmentation. Un module supplémentaire, dit central, est utilisé pour assurer la synchronisation des autres modules et assurer le traitement commun. L'ensemble des modules assure une collaboration de l'ensemble des utilisateurs dans le sens du partage de la manipulation de l'objet de la scène augmentée. Chacun des utilisateurs peut manipuler les objets virtuels comme s'il est le seul à le faire. Le principe d'augmentation précédemment présenté au troisième et quatrième chapitre est enrichi par une collaboration de plusieurs utilisateurs. C'est pourquoi l'architecture développée est une simple extension de celle précédemment implémentée (Figure V.3).

3.1.1. Module central

Le module central est le module le plus important d'un point de vue fonctionnel (Figure V.4). Il assure trois fonctions de base:

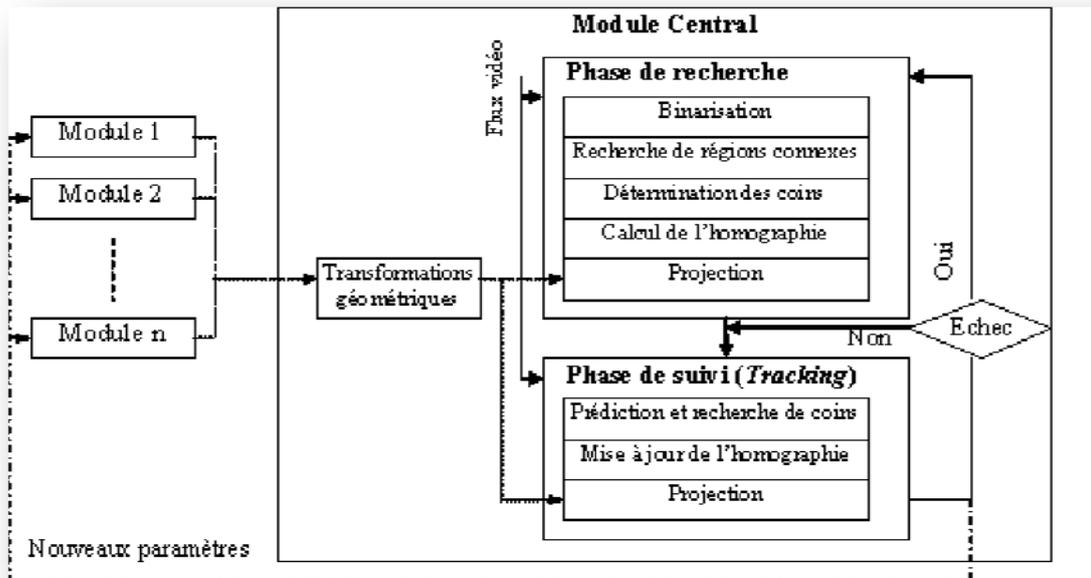


Figure V.3. Architecture globale du prototype du système collaboratif implémenté

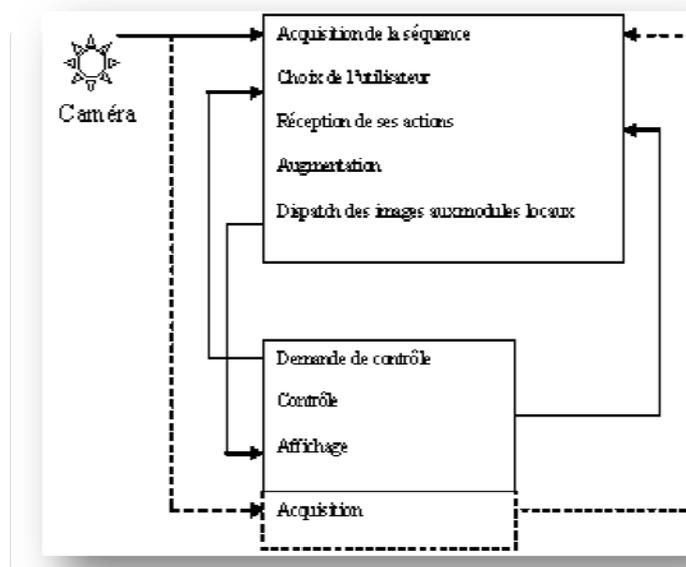


Figure V.4. Relations entre le module central et un module utilisateur

- Il réalise les augmentations de la scène filmée. Toute la tâche d'augmentation avec toutes les étapes inhérentes, y compris l'acquisition du flux vidéo, est accomplie par ce module dans un sens centralisé;
- Il gère la communication avec les autres modules en tant que serveur et assure la synchronisation des opérations de manipulation des utilisateurs transmises par les autres modules;
- La séquence augmentée générée, avec prise en compte des requêtes de manipulation pour la mise à jour de la disposition des objets virtuels, est

dispatchée à l'ensemble des modules pour être visualisée au niveau des écrans des utilisateurs correspondants.

Pour assurer les fonctions précédentes, le module effectue une sélection de l'utilisateur qui peut avoir le contrôle de l'objet. Ce dernier est simplement donné au premier demandeur après la fin des opérations du précédent, suivant un principe de service en FIFO. Pour ce faire, chaque utilisateur est identifié par un numéro correspondant au numéro du socket sur lequel il s'est connecté à l'application serveur. En conséquence, les objets virtuels partagés et manipulés par les différents utilisateurs sont contrôlés de sorte qu'ils le soient en exclusion mutuelle. Le module s'assure que les événements traités parvenant de l'un des utilisateurs inhibent ceux parvenant au même moment des autres. Ainsi, la manipulation n'est possible par un utilisateur donné que si le module central l'autorise à le faire. Le module central utilise un système de verrou de la manipulation par les différents modules utilisateurs qui sont en concurrence. Seules les requêtes du module détenteur du verrou sont traitées à un moment donné.

Pour augmenter la scène, le module central peut recevoir directement le flux vidéo de la caméra, comme il peut déléguer cette tâche à un autre module utilisateur. Le dernier cas est important si la scène à augmenter n'est pas située sur le site physique du module central. Dans ce cas, c'est l'un des modules qui accomplirait la tâche de réception du flux pour la communiquer au module centrale qui procède à son augmentation. Subséquemment, la répercussion des changements dans la scène augmentée dues aux manipulations d'un utilisateur est faite pour dispatcher la séquence produite à l'ensemble des modules, y compris le module qui récupère le flux de la scène.

Le principe suivi pour la gestion des utilisateurs dans leur manipulation des objets virtuels utilise un principe d'exclusion mutuelle. Les événements, relatifs aux actions d'un utilisateur sont traités comme un seul bloc. Pendant ce traitement, tout autre événement reçu d'un autre utilisateur est ignoré. La première raison est de ne pas créer des comportements aléatoires où un utilisateur ne connaît pas si la dernière opération appliquée à l'objet est relative à ses actions ou non. La seconde raison est de ne pas créer de conflits lorsque deux utilisateurs demandent des opérations opposées résultant sur une action nulle sur l'objet. Le résultat visuel dans ce cas serait un objet qui ne répond pas aux actions qui lui sont requises. Sur le plan communicationnel, ce principe donne la possibilité de connaître à tout instant l'utilisateur en droit d'action.

Pour contrôler la fin des opérations d'un utilisateur afin qu'il ne puisse monopoliser indéfiniment les objets virtuels, le module central active un timer. Si une demande de libération a été explicitement demandée, le timer est réinitialisé. Si à la fin du décompte aucune demande d'action d'un autre utilisateur n'a été enregistrée, le timer est réinitialisé. Si par contre au moins un autre utilisateur a demandé le contrôle, une requête de libération implicite est automatiquement exécutée.

3.1.2. Modules utilisateurs

Chaque utilisateur est représenté par un module local sur son site pour assurer la communication avec le module central. Une des fonctions de ce module est de demander, sous forme d'une requête, au module central le contrôle des objets virtuels. Une fois obtenu, son rôle devient celui de transmettre les actions demandées par l'utilisateur pour réaliser les transformations correspondantes. A la fin des actions, il transmet une requête de libération.

Ajoutées à cette tâche, les modules utilisateurs locaux ont pour fonctions de récupérer les résultats de l'augmentation réalisée par le module central. Ils mettent à jours l'affichage de l'augmentation résultante suite à une action par un utilisateur quelconque. Ils affichent la séquence obtenue et collectent en parallèle les différents évènements locaux pour les communiquer au module central (Figure V.4).

Une autre fonction peut être dédiée aux modules locaux mais à un seul d'entre eux à la fois. C'est celle de la réception du flux de la caméra filmant la séquence, ensuite sa transmission continue image par image au module central pour le traitement. Cette transmission, si elle a lieu, est faite en temps réel, c'est-à-dire que chaque image réceptionnée est transmise au fur et à mesure de la réception des images de la séquence augmentée. Le module source de la vidéo aura donc deux flux à considérer. Le premier, qui est considéré comme source, à transmettre au serveur et l'autre, traité par le serveur et augmenté, à afficher à l'utilisateur. Cependant, ce système peut nécessiter un débit du réseau supportant le flux conséquent. Si la caméra génère 10 images par seconde et que chaque image présente une taille de 210×180 pixels, la quantité de données à transmettre par seconde via le réseau est de $210 \times 180 \times 3 \times 8 \times 10 = 1,08 Mo = 8,65 Mbits$. Remarquons que cette taille des données est transmise dans un seul sens et est relative à une taille d'images assez petite et à une caméra ayant un débit assez réduit (simple webcam par exemple). Elle sera doublée dans deux sens différents pour l'un des modules et multipliée par le nombre d'utilisateurs. Ceci constitue nécessairement une charge pour le réseau. En plus des données des images, les données d'échange d'informations telles que les requêtes de manipulation et le trafic déjà existant sur le réseau affectent négativement le temps de visualisation de l'augmentation. Dans un tel contexte, l'augmentation peut être faite avec perte et le temps réel est difficile à atteindre.

L'extension des modules utilisateurs pour l'acquisition de la scène est utile afin de donner la possibilité à chaque utilisateur d'augmenter sa propre scène. Afin qu'un module utilisateur puisse être déclaré comme source du flux vidéo, il doit le demander explicitement au moment de sa connexion au module central. C'est une négociation entamée avec le module central qui peut résulter sur un refus.

3.2. Discussion de la solution

L'avantage de la solution centralisée, quant à l'augmentation de la scène est qu'elle inhibe la différence de puissance entre les différents postes. La charge de traitement de

la scène, qui nécessite une certaine puissance, est dédiée au seul poste serveur contenant le module central. Chacun des autres postes aura comme fonction le défilement d'un flux vidéo simple sans aucun traitement lourd d'arrière plan. Les traitements relatifs aux entrées des requêtes de manipulation ne constituent en aucun cas une lourde charge. Comme nous l'avons signalé plusieurs fois, l'objectif de notre travail est d'utiliser les outils d'interaction les plus rudimentaires et par conséquent, leur traitement est négligeable d'un point de vue puissance du processeur. La seule charge incombe au réseau qui devrait supporter le débit du flux vidéo et qui dépend du nombre de postes connectés. Avec l'avancée technologique récente, particulièrement pour les réseaux locaux, cela ne devrait pas poser de problème.

Comme déjà indiqué, l'objectif de notre recherche est de parvenir à faire collaborer des utilisateurs à travers internet. Il s'insère dans le cadre du développement d'un moyen offrant une méthode de vente/achat visuel qui permet de visualiser l'objet à vendre/acheter chez le client sans avoir à le déplacer réellement. Cette idée pourrait renforcer les concepts déjà développés dans une application e-commerce. De cette façon, un vendeur et un client seraient à même de discuter et de choisir des dispositions et des adaptations différentes, vérifiées par le principe de proportion présenté au troisième chapitre. Le problème, par contre qui incombe à cet objectif est le débit internet qui est encore relativement trop faible dans notre pays.

4. Conclusion

L'objectif de notre recherche dans le sens de la collaboration est de parvenir à faire collaborer des utilisateurs à travers internet. Des contrôles, impliquant des effets naturels des objets virtuels en tant qu'extension de ce travail sont à effectuer : occlusion des (par les) objets réels, respects des lois naturelles de la physique tel que le repos sur une surface plane et l'adaptation dans l'éclairage et dans l'ombre à la scène. Notons que le contrôle de l'occlusion est une tâche difficile. Elle nécessite l'acquisition et la connaissance par le système des formes géométriques des objets réels, de leurs positions ainsi que de leurs apparences au moment de l'augmentation [99].

Dans ce chapitre, notre solution présentée dans [13] est détaillée. Afin qu'elle soit construite de manière logique et située relativement aux travaux existants, une première partie du chapitre est consacrée au principe de la collaboration. Elle fait le lien entre les chapitres précédents et le domaine du collaboratif. Le domaine est d'abord présenté d'un point de vue théorique. Cette partie discute donc les différents aspects liés tels que la gestion des utilisateurs et l'impact de leurs situations géographiques sur les concepts développés dans la littérature.

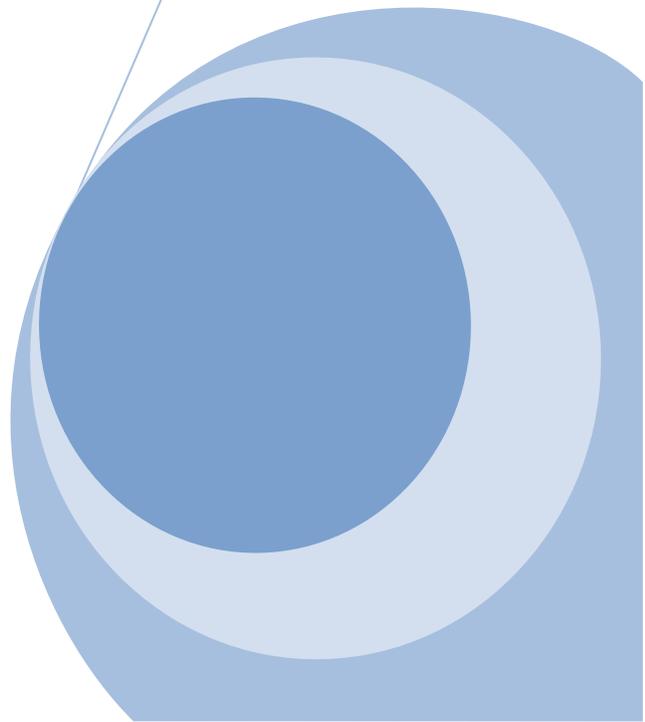
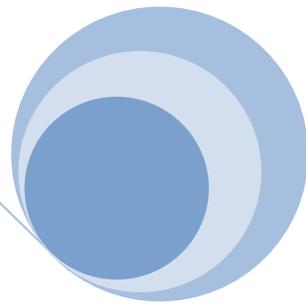
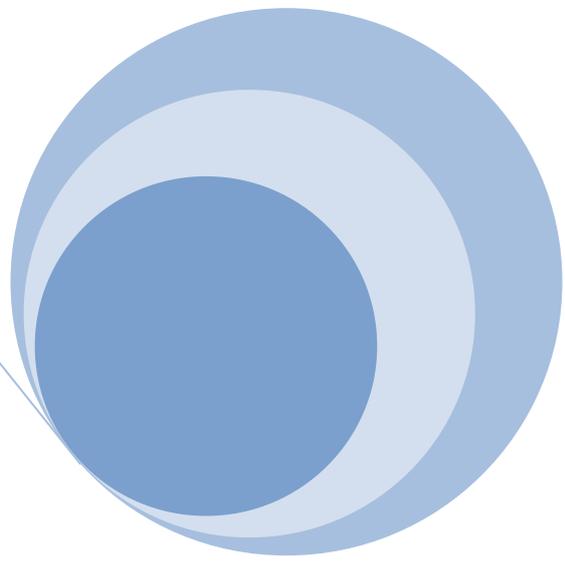
La deuxième partie est consacrée purement au travail développé. Le prototype de la plateforme est d'abord présenté. Dans cette partie est expliqué comment est ce que cette plateforme est particulièrement applicable à des utilisateurs distants ou co-localisés mais utilisant une seule vue de la scène augmentée. Par conséquent, la collaboration concerne uniquement des actions sur des objets virtuels. Dans le sens de l'implémentation,

l'architecture logicielle de la plateforme est également présentée. Elle consiste en deux parties logiques. Une première est formée d'un module central gérant l'augmentation proprement dite ainsi que la synchronisation des requêtes utilisateurs pour la manipulation des objets virtuels. Une seconde est constituée d'un ensemble de modules associés chacun à un utilisateur particulier mais ayant tous la même structure. En particulier, chacun est chargé de récupérer les requêtes utilisateurs, obtenir l'autorisation de contrôle de l'objet virtuel pour la manipulation et enfin transmettre les requêtes au module central. Exceptionnellement, la fonction de récupération du flux vidéo peut être dédiée à un des modules utilisateurs. Finalement une discussion de cette solution est faite.

Une première extension de ce travail peut être faite en implémentant les concepts développés dans ce chapitre sur le réseau internet. Ceci permet de connaître ses limites et recenser les difficultés qui peuvent surgir. La difficulté de cette extension réside dans le fait de devoir faire une étude à part entière des protocoles de communication supportés. Une deuxième extension non négligeable est celle de faire une étude plus formelle de la communication des modules en faisant appel par exemple aux agents à base de XML. Ces extensions sont d'autant intéressantes pour la mixture de domaines apparemment hétérogènes et sans relation.

Ce chapitre a fait l'objet d'une synthèse globale de notre travail associé au domaine de la réalité augmentée collaborative. Malgré que son volume soit réduit relativement aux autres chapitres, il est l'objectif principal de tout le travail. Tous les concepts théoriques et pratiques présentés avant n'en sont en fait qu'une base pour les résultats présentés à ce niveau.

Conclusion



Conclusion

La réalité augmentée est un domaine assez vaste et regroupe les résultats d'autres recherches qui sont considérées comme domaines en soit. L'objectif majeur tente à satisfaire des augmentations temps réel tout en préservant le réalisme des scènes. Dans ce contexte, les recherches sont axées sur la minimisation des temps de recherche et de tracking des indices d'augmentation, de génération et d'incrustation des objets virtuels, de leurs manipulation et de leur partage dans une situation collaborative.

Dans cette thèse, nous présentons une idée de base pour l'implémentation d'une application de réalité augmentée permettant l'incrustation des images d'objets réels. C'est une méthode d'augmentation permettant un compromis entre le temps de traitement et le réalisme de la scène. Le principe suivi est une augmentation vidéo. A partir d'un indice 2-D, nous montrons comment il est possible d'aligner un objet virtuel 3-D tout en réalisant une auto calibration de la caméra. Le concept essentiellement discuté tourne autour de l'utilisation d'images de plusieurs vues d'un objet dans la génération de l'objet virtuel 3-D au lieu de sa construction synthétique. Nous appelons l'objet virtuel qui est incrusté à la scène augmentée objet réel virtualisé. C'est un objet 3-D obtenu à partir d'image 2-D. Son avantage réside dans le détail existant dans les images prises qui n'est pas simple à modéliser. Cette idée permet particulièrement d'implémenter une multitude d'applications entre autres mentionnées dans cette thèse, celle de réaliser des ventes avec tests visuels à distance.

La manipulation des objets insérés renforce le sentiment de réalisme de l'augmentation. Elle consiste à déplacer les objets virtuels et à les disposer à des endroits de la scène qui dépendent de l'objectif de l'utilisateur. Elle permet de simplifier l'accomplissement de certaines tâches et/ou de les rendre plus performantes. Mieux encore, il est possible de parvenir à des résultats plus intéressants lorsque cette manipulation est faite par plusieurs utilisateurs dans un cadre collaboratif. En effet, elle donne une nouvelle dimension à la notion de communication de groupe.

Ainsi, la manipulation des objets virtuels étant un objectif de recherche, nous montrons dans cette thèse comment est simulée cette manipulation dans une scène augmentée. Une démonstration est ensuite faite sur la manière dont cette manipulation est appliquée aux objets réels virtualisés. Les concepts mathématiques et techniques informatiques sous-jacent sont discutés. Cette discussion est menée dans un contexte qui permet de satisfaire un réalisme visuel acceptable. Les expressions analytiques résultats de notre recherche sont établies à base de réflexions géométriques et supportées par des expérimentations pratiques démontrant leur efficacité.

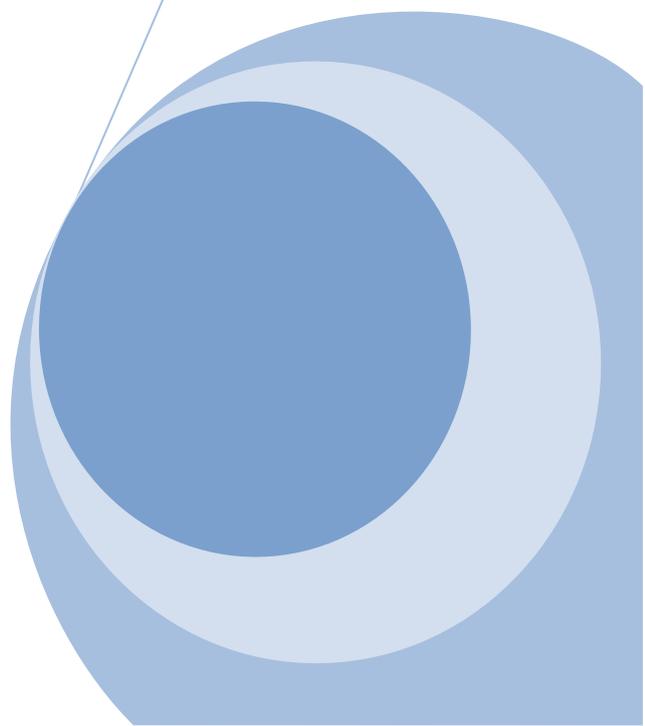
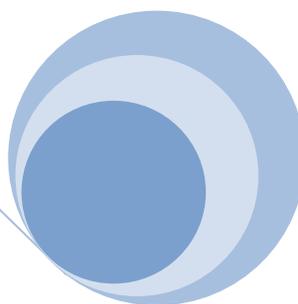
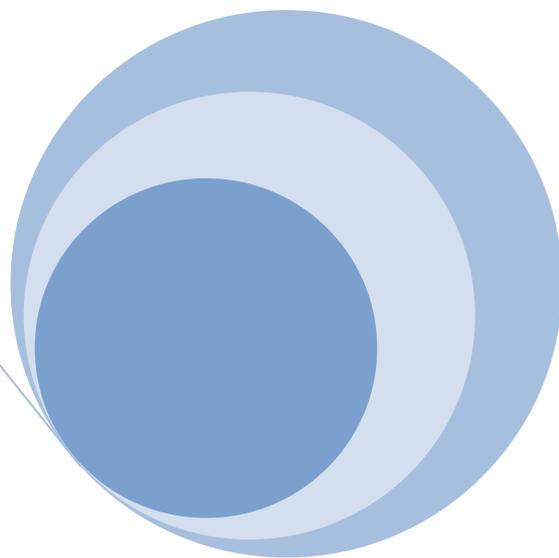
En ce qui concerne les images prises des vues des objets réels, nous proposons une structure linéaire afin de les intégrer dans un seul fichier de sorte qu'il serait possible de les traiter comme un flux vidéo. Une présentation de leur organisation pour les opérations d'accès est faite où son extension à supporter une augmentation multi-

résolution est démontrée. Cette organisation, tout en étant linéaire, est assez robuste sur le plan de l'accès. Elle permet d'avoir une vue abstraite des images comme si elles sont disposées sur une sphère virtuelle. Pour ces raisons, cette organisation présente des avantages par rapport à d'autres méthodes implémentant les mêmes principes.

Dans le même sens, la manipulation de l'objet inséré dans la scène est étendue pour faire collaborer plusieurs utilisateurs. Cette collaboration est synchronisée suivant un principe d'exclusion mutuelle. L'architecture générale est construite à base d'un module central qui contrôle l'augmentation proprement dite et d'un ensemble de modules locaux et spécifiques pour chaque utilisateur. Les différents événements reçus par les modules locaux sont communiqués au module central. De leurs côtés, ils mettent à jour les paramètres renvoyés, en l'occurrence l'affichage, par le module central relatifs à l'augmentation effectuée.

Des insuffisances sont facilement remarquables dans le travail réalisé mais cela est dû essentiellement à l'ouverture du domaine. C'est pour cette raison que notre travail futur touche essentiellement à l'amélioration des résultats obtenus. Le plus important est relatif à l'amélioration des vues utilisées suite aux translations suivant les axes Ox et Oy , qui naturellement devraient utiliser des vues différentes. Le deuxième est celui relative à la déduction de vues intermédiaires par interpolation à partir d'images adjacentes. Cela devrait permettre de faire face au problème de taille du fichier vidéo lorsque les images sont de haute résolution. Une autre amélioration qui concerne une autre orientation de recherche est celle des résultats obtenus particulièrement relativement à l'ombrage. L'effet de réalisme visuel sur le plan de l'éclairage, du rendu, ainsi que l'occlusion correcte des objets réels et virtuels fera également l'objet de notre recherche futur. D'un point de vue de la collaboration, nous voulons améliorer le principe de la communication des différents modules. Il serait plus opportun de faire appel à des méthodes plus formelles dans l'implémentation des protocoles de communication. Les agents peuvent être d'un apport non négligeable sur ce plan puisqu'ils peuvent faire appel aux automates à base desquels des preuves peuvent être faites. Le langage XML peut également être important si une tentative est faite pour étudier la possibilité de son utilisation et l'apport qu'il peut faire à la communication définie. Ceci est dû à sa large utilisation, sa simplicité, ses preuves d'efficacité et sa puissance expressive.

Bibliographie



Bibliographie

- [1] A. Danesi D. Fontanelli P. Murrieri V.G. Scordio A. Bicchi. *Cooperative Visual SLAMS by Homography*. Technical report, Interdepartamental Research Center "Enrico Piaggio" University of Pisa.
- [2] A. I. Comport, E. Marchand, F. Chaumette. *A real-time tracker for markerless augmented reality*. *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03, Tokyo, Japan, October 2003*.
- [3] A. Shahrokni, L. Vacchetti, V. Lepetit, P. Fua. *Automated Initialization of Polyhedral Object Tracking for Augmented Reality Applications*. *Computer Graphics Laboratory, Swiss Federal Institute of Technology Lausanne, CH 1015 Lausanne, Switzerland*.
- [4] A. Shahrokni, L. Vacchetti, V. Lepetit, P. Fua. *Polyhedral Object Detection and Pose Estimation for Augmented Reality Applications*. *In Proceedings of Computer Animation, 2002*.
- [5] A. Zisserman, A. Fitzgibbon, G. Cross. *VHS to VRML: 3D graphical models from video sequences*. *In: Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia. 1999*.
- [6] B. Hoff, R. Azuma. *Autocalibration of an Electronic Compass in an Outdoor Augmented Reality System*. *Proceedings of IEEE and ACM International Symposium on Augmented Reality 2000 (Munich, Germany, 5-6 October 2000), 159-164*.
- [7] B. J. P. Horn. *Tsai's camera calibration method revisited*. pp. 1–13, Feb 2000. <http://www.ai.mit.edu/people/bkph/papers/tsaiexplain.pdf>
- [8] B. Nini, M. Batouche. *Approche Simple pour la Manipulation d'Objets Virtuels en Réalité Augmentée Collaborative*. *CGE'04 (2005). EMP Bordj El-Behri, Alger, Algérie*.
- [9] B. Nini, M. Batouche. *Geometric Concepts Encapsulation Using a Storage Structure for VRO Manipulation in AR*. *SNIB'06. Biskra. Algérie*.
- [10] B. Nini, M. Batouche. *Real Time Virtualized Real Object Manipulation in an Augmented Reality Environment*. *1st International Symposium on Brain, Vision and Artificial Intelligence, 19-21 October. M. De Gregorio et al. (Eds.): BVAI 2005, LNCS 3704, pp. 477 – 486, 2005. Naples. Italy*.
- [11] B. Nini, M. Batouche. *Simulation of the Handling of Real Objects with a Complete Control of Rotation*. *Information Technology Journal 6(5): 672-680, 2007. ISSN 1812-5638. Asian Network for Scientific Information*.
- [12] B. Nini, M. Batouche. *Utilisation d'une Séquence pour l'Augmentation en Réalité Augmentée*. *JIG'05, Biskra. Algérie*.
- [13] B. Nini, M. Batouche. *Virtual Object Manipulation in Collaborative Augmented Reality Environment*. *IEEE-ICIT 2004, December 8-10 (2004). Tunis. ISBN: 0-7803-8663-9*.

-
- [14] B. Nini, M. Batouche. [Virtualized Real Object Integration and Manipulation in an Augmented Scene](#). *CAIP 2005. The 11th International Conference on Computer Analysis of Images and Patterns, 5-9 September, 2005*. A. Gagalowicz and W. Philips (Eds.): *CAIP 2005, LNCS 3691*, pp. 248 – 255, 2005. Paris. France.
- [15] B. Nini, M. Berkane, M. Bouzenada. [Manipulation d'objets virtuels dans un cadre collaboratif](#). *SNIB 2004. Biskra. Algérie*.
- [16] B. Nini, M. Bouzenada, M. Batouche. [Augmentation 3D à base de pattern planaire](#). *CGE'03 (2004), EMP Bordj El Bahri, Alger, Algérie*.
- [17] B. Nini, M. Bouzenada, M. Batouche. [Réalité augmentée temps réel](#). *JSTE 2003, Guelma, Algérie*.
- [18] B. Triggs. [Autocalibration from Planar Scenes](#). In *Proc. the 5th European Conference on Computer Vision*, pp. 89–105, Freiburg, Germany, June 1998.
- [19] B. Ullmer, H. Ishii, D. Glas. [MediaBlocks: Physical Containers, Transport, and Controls for Online Media](#). *Actes de la conférence SIGGRAPH'98, 1998*, p.24-32.
- [20] C. A. Ellis, S. J. Gibbs, G. Rein. [Groupware: some issues and experiences](#). *Communications of the ACM*, 34(1) :39–58, 1991.
- [21] C. Furmanski, R. Azuma, M. Daily. [Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information](#). In *Proc. ISMAR '02 (Int. Symposium on Augmented Reality)*, pages 215–224, Darmstadt, Germany.
- [22] C. Harris, M. Stephens. [A Combined Corner and Edge Detector](#). In *Alvey Vision Conf, 1988*. pp. 147-151.
- [23] C. Jaynes, Jing Hou. [Temporal Registration using a Kalman Filter for Augmented Reality Applications](#). *University of Kentucky, Department of Computer Science, Metaverse Lab, in Vision Interface, Montreal, Canada May 2000*.
- [24] C. Ke, B. Kang, D. Chen, X. Li. [An Augmented Reality-Based Application for Equipment Maintenance](#). *J. Tao, T. Tan, and R.W. Picard (Eds.): ACII 2005, LNCS 3784*, pp. 836 – 841, 2005.
- [25] C. Kervrann. [A Hierarchical Markov Modeling Approach for the Segmentation and Tracking of Deformable Shapes](#). *Graphical Models and Image Processing Vol. 60, No. 3, May*, pp. 173–195, 1998 Article no. IP980469.
- [26] C. McDonald. [Hand Interaction in Augmented Reality](#). Master of Ottawa-Carleton Institute for Computer Science. May 2002, Canada.
- [27] C. Yuan. [Visual Tracking for Seamless 3D Interactions in Augmented Reality](#). *G. Bebis et al. (Eds.): ISVC 2005, LNCS 3804*, pp. 321–328, 2005.
- [28] D. A. Bowman, E. Kruijff, J. L. J. LaViola, I. Poupyrev, [An introduction to 3-D user interface design](#). *Presence: Teleoperators and Virtual Environments*, 10(1), 96–108 (2001).
- [29] D. A. Bowman. [Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application](#). *PhD thesis, Georgia Institute of Technology, 1999*.
-

-
- [30] D. Gruber. The Mathematics of the 3D Rotation Matrix. <http://www.makegames.com/3Drotation/>
- [31] D. Schmalstieg, A. Fuhrmann, Z. Szalavari, M. Gervautz. An environment for collaboration in augmented reality. In *Collaborative Virtual Environments (CVE), Extended abstract, Nottingham, United Kingdom, September 1996*.
- [32] D. Schmalstieg, G. Reitmayr, G. Hesina. Distributed Applications for Collaborative Three-Dimensional Workspaces. *Presence, 12-1, p. 53 – 68, Vienna University of Technology, Austria. Februar 2001*.
- [33] E. Dubois, L. Nigay, J. Troccaz. Combinons le monde virtuel et le monde réel. *Acte des Rencontres Jeunes Chercheurs en IHM, Île de Berder, France, Mai 2000, p. 31-35*.
- [34] E. Moreno, B. MacIntyre, J. D. Bolter. Alice's Adventure's in New Media: An Exploration of Interactive Narratives in Augmented Reality. *CAST'01, Bonn, Germany, September 21-22, 2001*.
- [35] E. Trucco, A. Verri. *Introductory Techniques for 3D Computer Vision. Prentice-Hall, 1998*.
- [36] F. Remondino. Tracking of human movements in image space. *Internal technical report at IGP - ETH Zurich, April 2001*.
- [37] G. Jacucci, A. Oulasvirta, A. Salovaara, T. Psik, I. Wagner. Augmented Reality Painting and Collage: Evaluating Tangible Interaction in a Field Study. *INTERACT 2005: 43-56*.
- [38] G. Reitmayr, D. Schmalstieg. A Platform for location based Augmented Reality Applications. *Published in ÖGAI journal, Vol 21, Nr 1, 2002. TR-188-2-2002-04*.
- [39] G. Reitmayr, D. Schmalstieg. Location based Applications for Mobile Augmented Reality. *Australian Computer Society, Inc. Fourth Australasian User Interface Conference (AUIC2003), Adelaide, Australia. Vol. 18*.
- [40] G. Simon, A. W. Fitzgibbon, A. Zisserman. Markerless Tracking using Planar Structures in the Scene. *International Symposium on Augmented Reality, 2000, IEEE CS Press, Los Alamitos, Calif., pp. 137-146*.
- [41] G. Simon, M. O. Berger. Registration with a Zoom Lens Camera for Augmented Reality Applications. *In Proceedings of 6th European Conference on Computer Vision, Trinity College Dublin, Ireland, June 2000*.
- [42] G. Simon. *Vers un Système de Réalité Augmentée Autonome. Thèse de doctorat de Henri Poincaré. 1999. Loria. France*.
- [43] G. Welch, G. Bishop. An Introduction to the Kalman Filter. *TR 95-041. Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599-3175, March 2002*.
- [44] H. Bakstein, T. Pajdla. Omnidirectional Image-based Rendering. *In Computer Vision Winter Workshop, February 6-8 2006*.
- [45] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, K. Tachibana. Virtual Object Manipulation on a Table-Top AR Environment. *In Proceedings of the International*
-

- Symposium on Augmented Reality ISAR 2000, pages 111–119. IEEE Computer Society Press, Oct 5–6 2000.*
- [46] H. Kato, M. Billinghurst, K. Morinaga, K. Tachibana. [The effect of spatial cues in augmented reality video conferencing](#). In *Proceedings of the 9th International Conference on Human-Computer Interaction*, pages 478–481, 2001.
- [47] H. Kato, M. Billinghurst. [Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System](#). In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, 1999.
- [48] H. Kaufmann. [Collaborative Augmented Reality in Education](#). Monte Carlo, Monaco. *Position paper for keynote speech at Imagina 2003 conference, Feb 3rd*.
- [49] H. T. Regenbrecht, M. T. Wagner. [Interaction in a Collaborative Augmented Reality Environment](#). *CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA. ACM 1-58113-454-1/02/0004*.
- [50] H. Tramberend. [Avocado: A Distributed Virtual Environment Framework](#). *Bielefeld University, Doctoral thesis, Mars 2003*.
- [51] H. Zollner, R. Sablatnig. [Comparison of Methods for Geometric Camera Calibration Using Planar Calibration Targets](#). *28th OAGM/AAPR Workshop. June 17-18, 2004*.
- [52] I. Barakonyi, D. Schmalstieg. [Augmented Reality Agents in the Development Pipeline of Computer Entertainment](#). *F. Kishino et al. (Eds.): ICEC 2005, LNCS 3711, pp. 345 – 356, 2005*.
- [53] J. Fründ, J. Gausemeier, C. Matysczok, R. Radkowski. [Using Augmented Reality Technology to Support the Automobile Development](#). *W. Shen et al. (Eds.): CSCWD 2004, LNCS 3168, pp. 289 – 298, 2005*.
- [54] J. Heikkilä. [Geometric camera calibration using circular control points](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence 22.10, pp. 1066-1077, 2000*.
- [55] J. Kannala, S. Brandt. [A Generic Camera Calibration Method for Fish-Eye Lenses](#). *In Proceedings of the 2004 Virtual Reality (2004), IEEE*.
- [56] J. Min, J. H. Kim, J. H. Im, K. H. Yoon, J. Choi. [Real-time Augmented Reality System in 3D object](#). *2nd IASTED International Conference Visualization, Imaging and Image Processing (VIIP 2002) September 9-12, 2002 Benalmadena, Malaga, Spain*.
- [57] J. Park, B. Jiang, U. Neumann. [Vision-based Pose Computation: Robust and Accurate Augmented Reality Tracking](#). *In proceedings of International Workshop on Augmented Reality (IWAR)'99 (October 1999)*.
- [58] J. Park, W. Lee. [Augmented E-commerce: Making Augmented Reality Usable in Everyday E-commerce with Laser Projection Tracking](#). *V. Roca and F. Rousseau (Eds.): MIPS 2004, LNCS 3311, pp. 242–251, 2004*.

- [59] J. Pierce, A. Forsberg, M. Conway, S. Hong, R. Zeleznik, M. Mine. [Image Plane Interaction Techniques in 3D Immersive Environments](#). *Proceedings of the ACM Symposium on Interactive 3D Graphics (1997)*.
- [60] J. Rekimoto, N. Katashi. [The World through the Computer: Computer Augmented Interaction with Real World Environments](#). In *Robinson, G. (ed.), Actes du Symposium User Interface Software and Technology, UIST'95, ACM Press, 1995, pp.29-36*.
- [61] J. Underkoffler, H. Ishii. [Illuminating light: An optical design tool with a luminous tangible interface](#). In *CHI, pages 542–549, 1998*.
- [62] J.-S. Park, M. Y. Sung, S.-R. Noh. [Virtual Object Placement in Video for Augmented Reality](#). *Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part I, LNCS 3767, pp. 13–24, 2005*.
- [63] K. H. Ahlers, A. Kramer, D. E. Breen, P. Y. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R. T. Whitaker, D. Greer. [Distributed Augmented Reality for Collaborative Design Applications](#). *Technical report published in Eurographics '95 Proceedings, Maastricht, NL, 1995*
- [64] K. N. Kutulakos, J. R. Vallino. [Calibration-Free Augmented Reality](#). *IEEE Transactions on Visualization and Computer Graphics, 4(1), pp. 1-20, January 1998*.
- [65] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, W. Stuetzle. [View-based rendering: Visualizing real objects from scanned range and color data](#). In *proceedings of the 8th Eurographics Workshop on Rendering, 1997, 23-34*.
- [66] K. Wnuk, F. Dang, Z. Dodds. [Dense 3D Mapping with Monocular Vision](#). *The second International Conference on Autonomous Robots and Agents. December 13-15, 2004, Palmerston North, New Zealand*.
- [67] L. Darsa, B. Costa, A. Varshney. [Navigating static environments using image-space simplification and morphing](#). In *Michael Cohen and David Zeltzer editors, Symposium on Interactive 3D Graphics, April 1997. ACM SIGGRAPH, 25-34*.
- [68] L. Ma, Y. Chen, K. L. Moore. [Flexible Camera Calibration Using a New Analytical Radial Undistortion Formula with Application to Mobile Robot Localization](#). *Proceedings of the IEEE. International Symposium on Intelligent Control Houston, Texas. October 5-8, 2003*.
- [69] L. Wang, H.-Y. Shum, S. B. Kang. [Object Representation and Rendering using Inverse Concentric Mosaics](#). In *proceedings of The 5th Asian Conference on Computer Vision, ACCV, January 2002, Melbourne, Australia, 23-25*.
- [70] M. Bajura, H. Fuchs, R. Ohbuchi. [Merging virtual objects with the real world: seeing ultrasound imagery within the patient](#). In *SIGGRAPH, pages 203–210. ACM, 1992*.
- [71] M. Billinghurst, H. Kato, I. Poupyrev. [The MagicBook: A Transitional AR Interface](#). *Computers and Graphics, November 2001, pp. 745-753*.

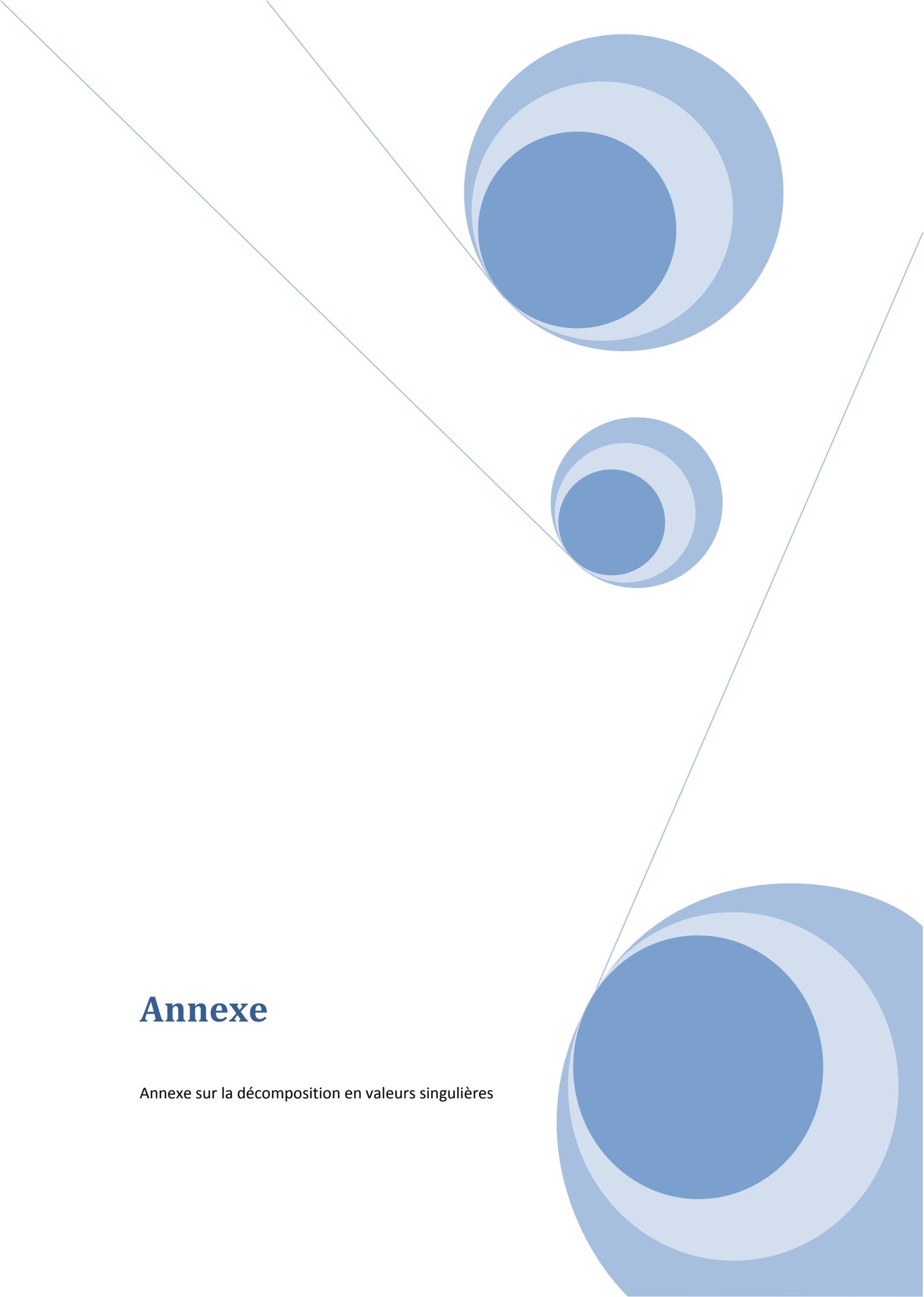
- [72] M. Billinghurst, H. Kato, I. Poupyrev. [The MagicBook—Moving Seamlessly between Reality and Virtuality](#). *IEEE Computer Graphics and Applications*. Volume 21, Issue 3 (May 2001). Pages: 6 – 8. Year of Publication: 2001. ISSN:0272-1716.
- [73] M. Billinghurst, H. Kato. [Out and About: Real World Teleconferencing](#). *British Telecom Technical Journal (BTTJ), Millennium Edition, Jan 2000*.
- [74] M. Billinghurst, I. Poupyrev, H. Kato, R. May. [Mixing Realities in Shared Space: An Augmented Reality Interface for Collaborative Computing](#). *Proc. IEEE Int'l Conf. Multimedia and Expo (ICME2000), 2000*.
- [75] M. C. Jacobs. [Managing Latency in Complex Augmented Reality Systems](#). *Proceedings of the 1997 symposium on Interactive 3D graphics, Providence, Rhode Island, United States*.
- [76] M. Chen, S. J. Mountfurd, A. Sellen. [A Study in Interactive 3-D Rotation Using 2-D Control Devices](#). *In proceedings of Computer Graphics SIGGRAPH 1988, 121-130*.
- [77] M. Fiala. [Automatic Projector Calibration Using Self-Identifying Patterns](#). *Procams 2005, IEEE International Workshop on Projector-Camera Systems. Saturday June 25, 2005, San Diego, California, USA*.
- [78] M. Jethwa, A. Zisserman, A. Fitzgibbon. [Real-time Panoramic Mosaics and Augmented Reality](#). *On-Line Proceedings of the Ninth British Machine Vision Conference. 1998. UK*.
- [79] M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann, D. Schmalstieg. [Structured Visual Markers for Indoor Pathfinding](#). *Proc. of The First IEEE International Augmented Reality Toolkit Workshop (ART02), Sep. 2002*.
- [80] M. Levoy and P. Hanrahan. [Light field rendering](#). *In Computer Graphics Proceedings SIGGRAPH'96, August 1996. New Orleans, 31-42*.
- [81] M. O. Berger, B. Wrobel-Dautcourt, S. Petitjean, G. Simon. [Mixing Synthetic and Video Images of an Outdoor Urban Environment](#). *Machine Vision and Applications, 11(3), Springer-Verlag, 1999*.
- [82] M. S. Pinho, D. A. Bowman, C. M. D. S. Freitas. [Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques](#). *VRST'02, November 11-13 (2002), Hong Kong. ACM 1-58113-530-0/02/0011*.
- [83] N. Braun. [Storytelling in Collaborative Augmented Reality Environments](#). *WSGS'(2003), February 3-7. Pizen. Czech Republic*.
- [84] N. M. Vallidis. [WHISPER: A Spread Spectrum Approach to Occlusion in Acoustic Tracking](#). *These of Doctor of Philosophy in the Department of Computer Science at the University of North Carolina. 2002*.
- [85] O. Chavanon, C. Barbe, J. Troccaz, L. Carrat, C. Ribuot, D. Blin. [Computer Assisted Pericardial Punctures: animal feasibility study](#). *In G. Goos, J. Hartmanis and J. van Leeuwen (eds.), Actes de la conférence MRCAS'97, Springer Verlag, 1997, p. 285-291*.

- [86] O. D. Faugeras. Stratification of 3-D vision: projective, affine and metric representations. *Journal of the Optical Society of America A* 12(3): 465-485, March 1995.
- [87] O. G. Staadt, M. Näf, E. Lamboray, S. Würmlin. JAPE: A Prototyping System for Collaborative Virtual Environments. *EUROGRAPHICS 2001 / A. Chalmers and T.-M. Rhyne. Volume 20 (2001), Number 3, pp. C-8–C-16.*
- [88] P. Alvarado, A. Berner, S. Akyol. Combination of high-level cues in unsupervised single image segmentation using Bayesian Belief Networks. *Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2002, June 24-27, 2002), Las Vegas, 2002.*
- [89] P. F. Sturm. Vision 3D non calibrée: Contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibration. *Thèse de doctorat de l'INPG, 1997.*
- [90] P. F. Sturm, L. Quan. Camera Calibration and Relative Pose Estimation from Gravity. *ICPR00, Barcelona, September 2000.*
- [91] P. Liu, X. Sun, N. D. Georganas, E. Dubois. Augmented Reality: A Novel Approach for Navigating In Panorama-Based Virtual Environments (PBVE). *The 2nd IEEE International Workshop Proceedings on Haptic, Audio and Visual Environments and Their Applications, HAVE 2003. 20-21 Sept. 2003. pp. 13-18.*
- [92] P. Milgram, F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems, E77-D(12), 1994, pp.1321-1329.*
- [93] P. Milgram, H. Takemura, A. Utsumi, F. Kishino. Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. *SPIE Vol. 2351 Telemanipulator and Telepresence Technologies, 1994.*
- [94] P. Renevier, L. Nigay. Mobile Collaborative Augmented Reality: the Augmented Stroll. *In Proceedings of EHCI'01, Toronto, May 2001, LNCS 2254, Springer-Verlag, pp.315-334.*
- [95] R. Azuma, J. W. Lee, B. Jiang, J. Park, S. You, U. Neumann. Tracking in Unprepared Environments for Augmented Reality Systems. *Computers & Graphics, Vol 23, No 6 December 1999, pp. 787-793.*
- [96] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments. vol. 6, no. 4, Aug. 1997, pp. 355-385.*
- [97] R. Azuma. Recent advances in Augmented Reality. *IEEE Computer Graphics and Applications 21 (Nov-dec 2001) 34–47 0272-1716/01/*
- [98] R. Grasset, J. D. Gascuel. Environnement de Réalité Augmentée Collaboratif: Manipulation d'Objets Réels et Virtuels. *AFIG'01 (Actes des 14èmes journées de l'AFIG) pages 101-112, Novembre 2001.*
- [99] R. Grasset, X. Decoret, J. D. Gascuel. Augmented Reality Collaborative Environment: Calibration and Interactive Scene Editing. *VRIC, Virtual Reality International Conference, Laval Virtual 2001, May 16-18.*

- [100] R. Hartley, A. Zisserman. *Epipolar Geometry and the Fundamental Matrix. Multiple View Geometry in Computer Vision pp. 219-242 (Cambridge University Press 2000).*
- [101] R. Horaud. *Vision 3-d projective, affine et euclidienne. Technical report, INRIA Rhône-Alpes and GRAVIR-CNRS, Janvier 2000.*
- [102] R. Stoakley, M. J. Conway, R. Pausch. *Virtual reality on a WIM: Interactive worlds in miniature. In Proceedings CHI'95, 1995.*
- [103] R. Want, A. Hopper, V. Falcao, J. Gibbons. *The Active Badge Location System. ACM Transactions on Information Systems, 10(1), 1992, 91-102.*
- [104] R. Wojciechowski, K. Walczak, M. White, W. Cellary. *Building Virtual and Augmented Reality Museum Exhibitions. In: The ninth international conference on 3D web technology, 2004.*
- [105] S. Bourgeois, H. Martinsson, Quoc-Cuong Pham, S. Naudet. *A Practical Guide to Marker Based and Hybrid Visual Registration for AR Industrial Applications. CAIP 2005. The 11th International Conference on Computer Analysis of Images and Patterns, 5-9 September. A. Gagalowicz and W. Philips (Eds.): LNCS 3691, pp. 248 – 255. Paris. France.*
- [106] S. E. Chen. *Quicktime VR - an image-based approach to virtual environment navigation. In proceedings of Computer Graphics, Annual Conference Series, 1995, 29-38.*
- [107] S. Feiner. *A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In Proc ISWC, October 13–14, 1997, pages 74–81.*
- [108] S. Gibson, A. Murta. *Interactive Rendering with Real-World Illumination. In Rendering Techniques 2000: 11th Eurographics Workshop on Rendering, pp. 365–376, June 2000.*
- [109] S. Gibson, J. Cook, T. Howard, R. Hubbard. *Rapid Shadow Generation in Real-World Lighting Environments. ESR 2003. ACM International Conference Proceeding Series; Vol. 44 Proceedings of the 14th Eurographics workshop on Rendering table of contents Leuven, Belgium. pp 219–229. ISBN ~ ISSN: 1727-3463, 3-905673-03-7.*
- [110] S. Greenberg, C. Fitchett. *Phidgets: easy development of physical interfaces through physical widgets. In UIST, pages 209–218, 2001.*
- [111] S. J. Gortler, R. Grzeszczuk, R. Szeliski, M. F. Cohen. *The lumigraph. In Computer Graphics Proceedings SIGGRAPH'96, August 1996, New Orleans, 43-54.*
- [112] S. Malik, G. Roth, C. McDonald. *Robust 2D Tracking for Real-time Augmented Reality. In Proceedings of Vision Interface (VI) 2002, Calgary, Alberta, Canada.*
- [113] S. Malik. *Robust Registration of Virtual Objects for Real-Time Augmented Reality. Master of Ottawa-Carleton Institute for Computer Science. May 2002, Canada.*

- [114] S. Ono, K. Ogawara, M. Kagesawa, K. Ikeuchi et al. A Photo-Realistic Driving Simulation System for Mixed-Reality Traffic Experiment Space. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV), June 2005*.
- [115] S. Subramanian, Z. Aliakseyeu, J.-B. Martens, M. Rauterberg. Interaction techniques for navigation through and manipulation of 2nd and 3rd data. In *Eurographics Workshop on Virtual Environments, May 2002*.
- [116] S. Wachter and H. H. Nagel. Tracking Persons in Monocular Image Sequences. *Computer Vision and Image Understanding Vol. 74, No. 3, June, pp. 174–192, 1999*.
- [117] Streitz, Geibler, Haake, Hol. DOLPHIN: Integrated Meeting Support across Local and Remote Desktop Environments and LiveBoards. *Conference Proceeding of Computer Supported Cooperative Work, CSCW'94, ACM (1994) 345-358*.
- [118] T. Höllerer. Situated Documentaries: Embedding Multimedia Presentations in the Real World. *Proceedings of ISWC '99, IEEE October 18–19, 1999, pp. 79–86*.
- [119] T. Reicher, A. MacWilliams, B. Brügge, G. Klinker. Results of a study on software architectures for augmented reality systems. In *Proceedings of the International Symposium on Mixed and Augmented Reality, Tokio, Japan, October 2003*.
- [120] U. Neumann, S. You, Y. Cho, J. Lee, J. Park. Augmented Reality Tracking in Natural Environments. *Mixed Reality - Merging Real and Virtual Worlds, Chapter 6, Ohmsha Ltd. & Springer-Verlag, 1999*.
- [121] V. Popescu, J. Eyles, A. Lastra, J. Steinhurst, N. England, L. Nyland. The WarpEngine: An Architecture for the Post-Polygonal Age. In *proceedings of the 27th Annual Conference on Computer Graphics, ACM SIGGRAPH, July 23-28, 2000, 443-454*.
- [122] W. A. Hoff, K. Nguyen. Computer vision-based registration techniques for augmented reality. *Proceedings of Intelligent Robots and Computer Vision XV, SPIE Vol. 2904, Nov 18-22, 1996, Boston, MA, pp. 538-548*.
- [123] W. Bruns, V. Brauer. Bridging the gap between real and virtual modeling - a new approach to human-computer interaction. In *IFIP5.10-Workshop Virtual Prototyping, Arlington, USA, May 1996*.
- [124] W.-C. Chen, H. Towles, L. Nyland, G. Welch, H. Fuchs. Toward a Compelling Sensation of Telepresence: Demonstrating a portal to a distant (static) office. *Proceedings of the 11th IEEE Visualization 2000 Conference (VIS) table of contents Year of Publication: 2000. ISBN:0-7803-6478-3*.
- [125] X. Cao, M. Shah. Camera Calibration and Light Source Estimation from Images with Shadows. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Volume: 2, pp. 918- 923 vol. 2*.
- [126] X. W. Zhong, P. Boulanger, N. D. Georganas. Collaborative Augmented Reality: A Prototype for Industrial Training. *Proc. of 21st Biennial Symposium on Communications, Canada, June 2002*.

- [127] X. W. Zhong, P. Liu, N. D. Georganas, P. Boulanger. Designing a Vision-based Collaborative Augmented Reality Application for Industrial Training. *Information Technology 45 (January 2003)*.
- [128] Y. E. Sutherland. A head-mounted three-dimensional display. In *AFIPS Conference, volume 33, pages 757–764, 1968*.
- [129] Y. E. Sutherland. The ultimate display. In *IFIPS Congress, volume 2, pages 506–508, New York, USA, May 1965*.
- [130] Y. Uematsu, H. Saito. Vision-Based Registration for Augmented Reality with Integration of Arbitrary Multiple Planes. *F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 155–162, 2005*.
- [131] Z. Huang, B. Boufama. A Semi Automatic Camera Calibration Method for Augmented Reality. *Systems, Man and Cybernetics Society (2002) IEEE International Conference. 6 pp. Volume: 4, ISSN: 1062-922X*.
- [132] Z. Szalavari, M. Gervautz. The personal interaction panel—A two-handed interface for augmented reality. *Computer Graphics Forum, 16(3): C335–C346, 1997*.
- [133] Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 22, No. 11, pp. 1330–1334 November 2000. Tr98-71, MSR Redmond*.



Annexe

Annexe sur la décomposition en valeurs singulières

Annexe A

1. Décomposition en valeurs singulières (DVS)

La décomposition en valeurs singulières (Singular Value Decomposition) ou SVD est un outil permettant, entre autres, d'obtenir la solution d'un système linéaire exact, la solution au sens des moindres carrés d'un système linéaire sur-contraint et donne une famille de solutions pour un système linéaire sous-contraint. La décomposition en valeurs singulières donne aussi un diagnostic de la situation.

Dans cette thèse, elle est principalement utilisée pour la résolution de systèmes linéaires homogènes, c'est-à-dire de la forme:

$$Ax = 0$$

La décomposition en valeurs singulières est basée sur le théorème d'algèbre linéaire suivant:

Toute matrice A de taille $(n \times m)$ peut être décomposée comme suit:

$$A = UDV^T$$

Où $U(n \times m)$ et $V(n \times n)$ sont des matrices orthogonales et $D(n \times n)$ une matrice diagonale. Les colonnes de U et V sont orthonormées: $U^T U = I_{m \times m}$ et $V^T V = I_{n \times n}$

Calculer la SVD consiste à trouver les valeurs propres et les vecteurs propres de AA^T et $A^T A$. Les vecteurs propres de AA^T constituent les colonnes de U et les vecteurs propres de $A^T A$ constituent les colonnes de V . Les valeurs singulières (valeurs réelles non négatives) de A sont les racines carrées des valeurs propres de AA^T ou $A^T A$.

Les valeurs singulières sont les éléments de D , elles y sont disposées dans un ordre descendant.

Les valeurs singulières sont toujours des nombres réels. Si A est une matrice réelle alors U et V le sont aussi.

Par exemple, la décomposition de la matrice suivante est comme suit:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$AA^T = A^tA = \begin{bmatrix} 6 & 10 & 6 \\ 10 & 17 & 10 \\ 6 & 10 & 6 \end{bmatrix}$$

Les valeurs propres de AA^T et A^tA sont:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} 28,86 \\ 0,14 \\ 0 \end{bmatrix}$$

Les vecteurs propres de AA^T ou A^tA sont:

$$u_1 = v_1 = \begin{bmatrix} 0,454 \\ 0,765 \\ 0,456 \end{bmatrix}$$

$$u_2 = v_2 = \begin{bmatrix} 0,542 \\ -0,643 \\ 0,542 \end{bmatrix}$$

$$u_3 = v_3 = \begin{bmatrix} -0,707 \\ 0 \\ -0,707 \end{bmatrix}$$

NOTE : Le rang d'une matrice est égal au nombre de valeurs singulières différentes de zéro.

2. Systèmes homogènes

Pour la résolution du système linéaire homogène:

$$Ax = 0$$

où il n'y aura pas une solution unique, il est nécessaire d'imposer une contrainte supplémentaire sur x :

$$\|x\| = 1$$

Cas spécial: $\text{Rang}(A) = n - 1$ ($m \geq n - 1, \mu_n = 0$)

Alors x est le vecteur propre correspondant à la valeur propre minimale de A^tA .

Cas général: $\text{Rang}(A) = n - k$ ($m \geq n - k, \mu_{n-k+1} = \mu_n = 0$)

La solution est $x = a_1 v_{n-k+1} + a_2 v_{n-k} + \dots + a_k v_k$ (a_i est une constante telle que $a_1^2 + a_2^2 + \dots + a_k^2 = 1$).