Environnement de Réalité Augmentée : vers les jeux vidéos

Raphaël Grasset
Projet de DEA Imagerie Vision et Robotique (ENSIMAG)

Date : Mars-Juin 2000
Lieu : Équipe iMAGIS/Laboratoire GRAVIR
Responsable : Jean-Dominique Gascuel

Table des matières

Chapitre 1

Introduction

1.1 Motivations

La réalité augmentée trouve aujourd'hui des applications dans de nombreux domaines tel que la médecine, la maintenance, l'assistance au travail et dans les loisirs. Le principe consiste à ajouter des informations virtuelles à notre environnement réel augmentant la quantité d'information disponible. Les applications collaboratives se combinent parfaitement avec ce type de système, conservant un contact avec le monde réel, mais ajoutant des informations non disponibles dans le monde réel : annotations virtuelles d'objets, représentation virtuelle d'objets complexes (bâtiments en architecture), représentation sous forme d'avatar d'une personne connectée à distance.

Un cadre correspondant à ce type de système est le domaine du jeu. Actuellement la forme en pleine expansion est le jeu vidéo, avec une forte tendance aux jeux multi-utilisateurs en réseaux. Les jeux interactifs sont de plus en plus une aire de recherche pour l'image de synthèse. L'influence économique du jeu vidéo à pris une place capitale sur le marché mondial. Le chiffre d'affaire se monte à 6.4 milliards de dollars pour les États-Unis en 1999, les jeux se vendant en millions d'exemplaires. Très peu de travaux concernant l'application de la réalité augmentée dans ce domaine ont été menés malgré le fort intérêt que cela peut représenter ([1], [2]], [3]).

On présente ici quelques avantages d'une simulation de jeu en réalité augmentée. On souligne que ce type d'applications combine les avantages d'un jeu réel et d'un jeu vidéo. On se place dans le cadre d'utilisateurs positionnés autour d'une table interagissant dans un espace virtuel superposé sur la table (1.1).

L'aspect social du jeu sur plateau est conservé par un système de réalité augmentée, les joueurs se trouvent en général placés autour d'un élément commun réel (une table avec un plateau virtuel) et peuvent communiquer librement. Par le fait d'être en réalité augmentée la communication gestuelle, visuelle, et orale est totalement conservée. Actuellement dans les jeux vidéos en réseaux, cette communication est totalement absente, chaque utilisateur immiscé dans son propre environnement. L'interaction et la représentation virtuelle sont propres à chaque joueur, ce qui permet une totale indépendance d'actions. La décharge de calcul par l'ordinateur permet la simulation de jeu complexe, supprimant la difficulté de réflexion inutile présent dans certains jeux : calcul de points au scrabble, gestion des points de vie pour un jeu de rôles. L'animation et les effets spéciaux présentés dans un jeu vidéo peuvent être repris. L'utilisation d'objets manipulables vir-

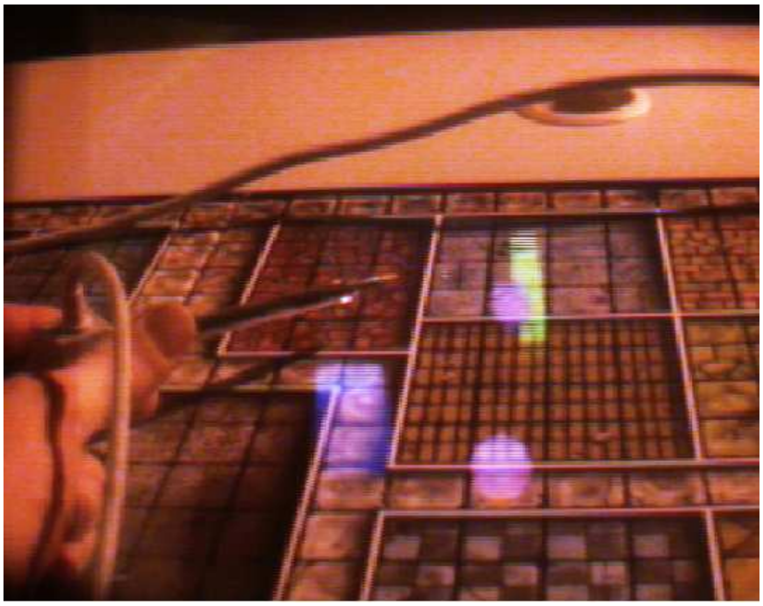

FIG. 1.1 – Vue d'un utilisateur placé autour d'une table

tuels (pion, cartes) permet une infinité de représentations, et une véritable personnalisation dans son environnement de jeu (définition de l'aspect visuel des pions selon ses goûts..). L'accès à des bases de données via Internet, permet une grand nombre de possibilité de jeu : choix parmis de nombreux scénarios, existence des personnages entre plusieurs sessions de jeu ..etc..

Aux vues des nombreux aspects de l'implication de la réalité augmentée, une étude plus précise est démarrée dans ce stage de DEA.

1.2 Contraintes du projet

On s'intéressera à développer une application de réalité augmentée combinant les avantages des deux environnements, le réel et le virtuel. L'application visée sera basée sur un jeu de rôle, où l'interaction et la communication jouent un rôle prépondérant. L'application sera temps réel et permettra de gérer deux utilisateurs et un maître de jeu. On se ciblera de plus sur une configuration manuelle du système basée sur des actions de base de la part de l'utilisateur. On s'orientera sur un environnement dédié à des applications collaboratives de toutes sortes. Un accès réseau permettra une gestion de son personnage par l'utilisateur. Enfin, on se penchera sur les problèmes de gestion du jeu : étude d'outils de création et de manipulation de scènes virtuels et de scénarios.

1.2.1 Moyens informatiques

Le développement sera réalisé sous IRIS PERFORMER en C++, qui est choisi pour sa gestion de scènes virtuelles et ses possibilités multiprocesseur. Le programme fonctionnera sur une station graphique haut de gamme (ONYX2 / ORIGIN 2000) multiprocesseur. La puissance de la machine devra être exploitée au maximum du fait de sa grande capacité et du fait d'une contrainte de temps réel. On travaillera à l'aide de capteurs magnétiques de type Flock of Birds d'Ascension Technologies et avec un système optique à plusieurs caméras de SAGEIS. Le système de visualisation sera un casque semi-transparent I-glasses de virtual I/O. Cet environnement, aujourd'hui coûteux, pourra à moyens termes être remplacé par des PC graphiques hautes performances et des capteurs économiquement compatibles avec une exploitation de salle de jeu.

1.2.2 Difficultés du projet

Les "points durs" à résoudre pour une telle étude, et que nous détaillerons dans ce rapport, sont les suivants :

- alignement monde réel/monde virtuel ;
- temps de latence faible (temps de réponse entre le mouvement physique et la réponse visuelle) ;
- gestion des périphériques et suppression des erreurs matérielles ;
- interface intuitive d'action pour l'utilisateur et manipulation d'objets virtuels ;
- gestion du multi-utilisateurs.

Chapitre 2

Réalité Augmentée

Nous présentons dans ce chapitre un survol sur les techniques de la RA. Nous présentons à partir du chapitre 4 une étude biblio-critique en fonction de nos besoins.

2.1 Définition

La réalité augmentée (RA) est une branche dérivée de la réalité virtuelle. Les technologies *"standard"* d'environnement virtuel immergent l'utilisateur dans un environnement synthétique. L'utilisateur est alors coupé entièrement de l'environnement du monde extérieur, c'est-à-dire du monde réel. Contrairement à cela, la réalité augmentée permet à l'utilisateur de voir le monde réel, et ainsi de pouvoir superposer (mélanger) des objets virtuels avec le monde réel. La combinaison se fait dans le même espace 3D de référence. On peut définir la réalité augmentée comme le niveau intermédiaire entre les environnements virtuel et réel.

D'autres termes sont souvent utilisés pour designer la réalité augmentée tel que "mixed-reality". On peut utiliser la taxonomie de Drascic (voir [4]) pour classer ces différentes types applications de réalité augmentée en 3 stratégies de base [5] :

- **augmenter l'utilisateur** : l'utilisateur porte des périphériques tels que casque, gant qui lui permettent d'obtenir des informations sur des objets réel.
- **augmenter des objets physiques** : l'objet physique est complété par des entrées sorties : supplémentaires et des périphériques informatiques.
- **augmenter l'environnement autour des objets** : Dans ce dernier cas ni l'utilisateur ni les objets ne sont affectes directement. Des périphériques indépendants sont utilisés pour fournir et collecter des informations sur l'environnement.

2.2 Historique

La réalité virtuelle (RV) a commencé au début des années 60 avec la création de la première machine virtuelle par Heilig. La réalité virtuelle consiste à plonger un utilisateur dans un monde virtuel. Les 3 points clés sont :

- **l'immersion** : l'utilisateur est coupé du monde réel, il est immergé dans le monde virtuel ;

– **l'interaction** : l'utilisateur peut interagir dans ce monde virtuel, se déplacer ou saisir des objets ;
– **l'imagination** : l'utilisateur complète, grâce à son imagination, le monde sommaire qu'on lui propose.

Durant les années 80 les premiers systèmes de RV commencent à apparaître : les recherches sont principalement effectuées sur les périphériques car la technologie de l'époque ne permet pas d'avoir des systèmes de moteur 3D et d'affichage assez puissant. Les premiers casques (" head mounted display ") et les premiers gants (" data glove ") apparaissent. Les recherches sont principalement effectuées en coopération avec l'armée américaine qui s'intéresse de près à cette technologie : elle trouve son intérêt dans les simulateurs de vol, les entraînements virtuels pour ses soldats, et pour la visualisation scientifique. Au début des années 90, les premiers véritables projets se manifestent et sont principalement dans le domaine scientifique. C'est le cas du projet GROPE dirigé par Fred Brooks, utilisant un bras manipulateur et un casque stéréoscopique pour la visualisation et la manipulation de molécules chimiques. Le projet est complété par un retour d'effort et une meilleure précision dans la manipulation [6]. Un autre projet important du début des années 90 est le projet VIRTUAL WIND TUNNEL à NASA-AMES de Steve Bryson . En utilisant un BOOM (bras articulé avec casque) et un gant virtuel comme outils d'interaction, les scientifiques sont capables de voir et d'interagir avec des images stéréoscopiques d'un champ de flux d'air autour d'un avion virtuel (2.1) .

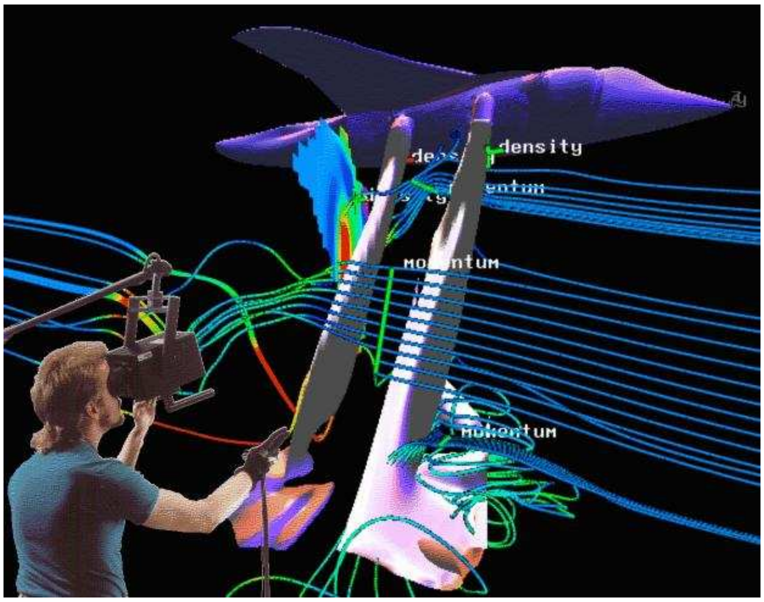

FIG. 2.1 – Simulation à l'aide du BOOM

Mais les problèmes commencent à apparaître : la communication entre les scientifiques est impossible car l'immersion totale du scientifique dans le monde virtuel le coupe totalement du monde réel, il ne peut interagir avec l'extérieur. De plus, la manipulation du BOOM nécessite un grand espace autour de l'utilisateur, qui ne peut alors pas vraiment savoir où il se déplace : on a une perte de liberté de mouvement due à une zone morte créée autour du bras et donc l'utilisation du BOOM présente des défauts. Le problème se retrouve avec les casques portables car le fait de ne pas savoir où on se trouve dans le monde réel pose problème dans certains cas. C'est ainsi que les travaux de Sutherland ont refait surface. Ce pionnier a fait de la recherche sur les casques de RV au début des années 70. En affichant simplement quelques polygones, il construit le premier casque de RV transparent (" see through system "). L'utilisation d'un casque à travers lequel on peut voir le monde réel répond complètement au problème : on permet à l'utilisateur de garder le contact avec le monde réel tout en étant dans un monde virtuel. Ceci constitue une des premières étapes de l'apparition de la réalité augmentée.

2.3 Applications

On décrit ici les diverses applications de la RA dans les domaines existants. Ceci constitue juste une présentation globale, en soulignant les travaux principaux, une étude critique est réalisé à partir du chapitre 4.

2.3.1 Médical

Les docteurs peuvent utiliser la réalité augmentée comme une aide de visualisation et d'entraînement pour la chirurgie. Il est possible de collecter des ensembles de données 3D d'un patient en temps réel en utilisant des capteurs non intrusifs tel que l'IRM,la tomographie ou encore les images ultrasonores. Ces ensembles de données peuvent être rendu et combinés en temps réel avec la vue du patient. En effet, cela donne au docteur une vue en rayon X de l'intérieur du patient. Ceci peut être très utilise lors d'intervention mineur, en réduisant le traumatisme d'une opération en utilisant des petites incisions ou pas d'incisions du tous. Les chirurgiens peuvent voir directement en 3D des points importants qu'ils n'auraient pas vue avec des systèmes MIRM ou CT scans. La réalité augmentée permet d'accéder aux deux types d'information simultanément. L'information provenant des capteurs non immersifs permet de générer une vue virtuelle constitué de l'endroit de pratique de l'opération directement sur le patient.

Plusieurs projets ont exploité cette aire de recherche. A l'UNC Chapell Hill, un groupe de recherche a réaliser un prototype de "scan" d'une ventre d'une femme enceinte avec un capteur à ultrasons, générant une représentation 3D du foetus à l'intérieur du ventre et l'affichant sur un casque de semi transparent (2.2). Le but est de voir l' évolution des mouvements du foetus à l'intérieur du ventre de la mère, en espérant que cela devienne un stéthoscope 3D ([7]). Des travaux plus récents se sont intéressés à une biopsie de tumeur de seins (2.3). La figure suivante montre une opération de biopsie du sein, où l'objet virtuel identifie la position de la tumeur et guide le chirurgien pour l'opération ([8]).

2.3.2 Maintenance et réparation

Un des premiers systèmes de RA a été réalisé par Feiner, c'est le projet KARMA (Knowledge-based Augmented Reality for Maintenance Assistance) [9]. Comme démonstration, ils choisissent un système de support du personnel pour configurer et gérer une imprimante laser. Beaucoup d'efforts sont essentiels pour générer un modèle précis. Plusieurs capteurs sont attachés à l'imprimante pour détecter leur position et leur orientation. Le monde virtuel est un complément, un

Fig. 2.2 – Système de réalité augmentée de visualisation de reconstruction 3D d'une échographie superposé à la personne ausculté

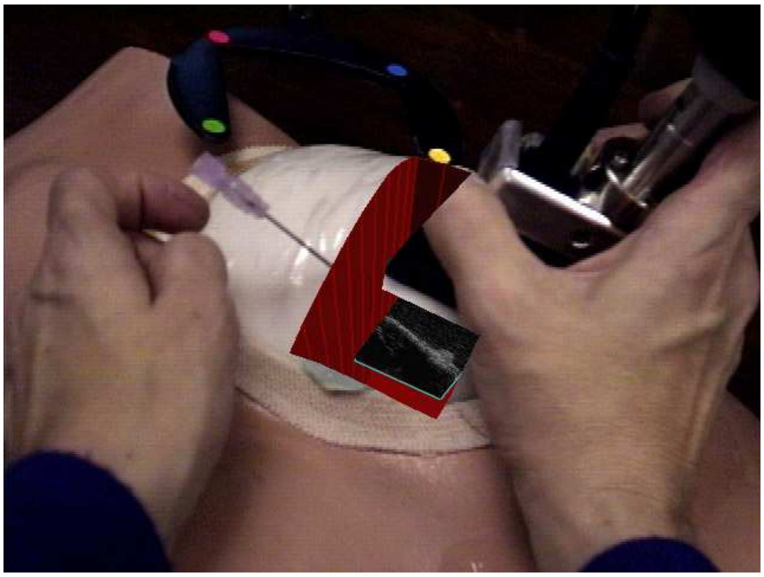

Fig. 2.3 – Biopsie du sein à l'aide d'un système de réalité augmentée

enrichissement du monde réel. Sur cet exemple (2.4) on observe ce que voit l'utilisateur à travers le casque . Dans cette application on montre à l'utilisateur comment changer le papier dans l'imprimante. Le monde virtuel possède une représentation du bac de papier et son fonctionnement.

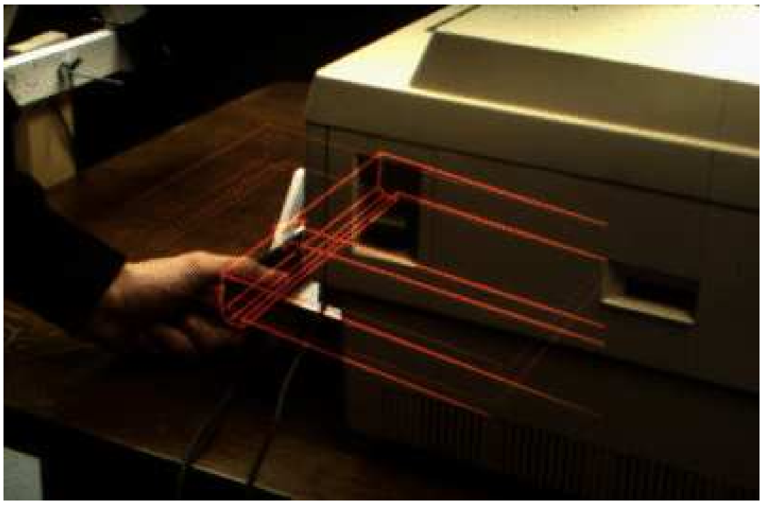

FIG. 2.4 – Vue à l'intérieur du casque

Un groupe chez Boeing s'est intéressé à un système de RA pour guider un technicien à construire un WIRING HARNESS qui forme le système électrique d'une aile d'avion. Actuellement les techniciens utilisent de grands tableaux pour construire de tels systèmes, et Boeing a besoin de beaucoup de pièces pour stocker ces systèmes. Il utilise une projection virtuelle sur un unique tableau (2.5), ce qui permet de conserver de la place et de réduire les coûts ([10]). Éventuellement la réalité augmentée peut être utilisew pour l'industrie automobile ([11]).

2.3.3 Annotations et visualisation

La réalité augmentée peut être utilisée pour annoter des objets et l'environnement avec des informations privées ou publiques.

Par exemple un "hand-held display" peut fournir des informations concernant les bouquins d'une librairie lorsque l'utilisateur se ballade autour de cette bibliothèque. Les chercheurs de Columbia ont montré ceci par la notion d'attachement d'une fenêtre d'une interface standard vers une partie spécifique du monde, ou attachée à un objet spécifique [12]. La réalité augmentée peut aussi servir d'aide générale à la visualisation. Un architecte peut regarder à travers un HMD pour être capable de voir une fenêtre et voir comment le nouveau bâtiment changera la vue. La RA peut aussi servir à un architecte pour une vision "rayon X" à l'intérieur d'un building pour observer l'installation électrique, les tuyaux d'eau, les structures de murs ([13]).

2.3.4 Téléopération

Guider un robot à distance par un système de téléopération est un problème difficile du au temps de délai de transmission et à la distance du robot. La solution consiste à manipuler une

FIG. 2.5 – Prototype de Boeing

version virtuelle du robot, et une fois le résultat convenable, déclencher le vrai mouvement du robot. Le système permet aussi de prédire et d'assister l'utilisateur qui manipule le robot pour réaliser sa tâche.

Le système ARGOS ([14]), utilise la réalise augmentée sous la forme d'un pointeur virtuel pour guider le mouvement d'un robot dans une scène réelle (leur système permet aussi de mesurer des distances entre objets). Il introduit un système stéréoscopique (à l'aide de "shutter glasses") apportant le signe de stéréoscopie facilitant la manipulation virtuelle 3D.

2.3.5 Militaire

Depuis de nombreuses années les militaires utilisent des casques HUD (Head-Up Display) et Helmet Mounted Sights (HMS) pour superposer des informations de vols(navigation - état de vol) sur la vision du pilote du monde extérieur. Mais le graphisme est aussi utile pour aligner des cibles avec l'environnement, désignant un chemin pour les missiles.

2.3.6 Loisirs

On peut citer les "Virtual Set" qui permettent d'intégrer des personnages réels avec un environnement virtuel en temps réel et en 3D. Le système repose sur l'enregistrement par une caméra en mouvement des acteurs sur fond bleu. La caméra étant poursuivie et la position des utilisateurs saisie on peut digitaliser les acteurs dans l'environnement 3D virtuel. Le projet ALIVE du laboratoire MIT Media Lab s'intéresse à la population d'environnement par des créatures intelligentes répondant aux actions utilisateurs. L'utilisateur devant une caméra peut par exemple donnér des ordres à un chien de faire le beau, s'asseoir ..etc.. (ils utilisent un système de vision et d'analyse d'image).

Chapitre 3

Périphériques utilisées en réalité augmentée

3.1 Capteurs de position et d'orientation

Ils servent à mesurer la position et l'orientation d'un objet rigide en mouvement par rapport à un repère absolu. On s'en sert principalement pour mesurer le mouvement de la tête et différents objets tels que des pointeurs ou la main de l'utilisateur. En général, on mesure une série de 6 grandeurs (3 positions et 3 angles) avec une fréquence d'échantillonage compatible avec la précision recherchée. Il existe de nombreux types de capteurs. On peut les classer par catégories : mécaniques, magnétiques, ultrasonores, optiques. Pour les comparer on utilisera les critères suivants :

- **ddl** : nombre de degrés de liberté obtenus par le système. On utilisera la notation Comb si on fait référence à plusieurs éléments pour obtenir ces DDL : combinaisons de plusieurs leds par exemple.
- **résolution** : définit la différence de pas la plus petite mesurable (fréquence d'échantillonage spatiale) ;
- **erreur** : définit la précision du système ;
- **temps de réponse** : temps entre la mesure et la réception par le système traitant ;
- **espace de travail** : espace de déplacement avec le système.
- **sensibilité** : définit la sensibilité du capteur à l'environnement extérieur pouvant induire de l'erreur dans le système (bruit électromagnétique).

3.1.1 Capteur mécanique

Ils se présentent en général sous la forme de bras articulés avec 2 ou 3 segments. Ils utilisent des dispositifs de mesures analogique ou numérique de grande précisions attaché au joint du bras pour obtenir la position et l'orientation de l'extrémité du bras. N'utilisant pas de technologies sensibles à l'environnement (électromagnétique, sonores) ils n'ont aucunes interférences avec l'extérieur. Les systèmes mécaniques ont suffisamment une bonne précision, résolution, et temps de réponse, mais leur champ de travail est limité et ils sont très peu maniable (ddl limités à chaque jointure).

Solutions commerciales

Le premier casque construit par Ivan Sutherland utilisait un bras mécanique pour mesurer la position de la tête. Un produit commercial du nom de BOOM utilise un mécanique pour mesurer la direction du regard d'une jumelle binoculaire manipulable à la main. On peut aussi citer le bras Faro Metrecom IND-1. Ces caractéristiques sont 6 DDL avec une zone sphérique de mesure de 1,8 M. Il permet d'obtenir la position (vecteur) et l'orientation (sous forme d'angles d'Euler). Il a 8 bits de résolution et une précision de 0,46 mm. Le bras le plus vendu est le Desktop Phantom de SensAble Technologies : Il possède un retour d'effort à 3 DDL, 6 DDL de position, il a une précision de 0.02 mm mais un espace de travail de 16 cm x 13 cm x 13 cm. Deux autres bras (nettement plus chère) propose des espaces de travail plus grand.

ddl	6 DDL
résolution	élevée
erreur	faible
temps de réponse	faible
espace de travail	faible
sensibilité	nulle
rem	(-) peu maniable

TAB. 3.1 – Récapitulation des caractéristiques des capteurs mécaniques

3.1.2 Capteur magnétique

C'est le système le plus utilisé actuellement. Un champ basse fréquence est modulé pour déterminer la position et l'orientation de l'objet mobile. Le champ basse fréquence est généré par un émetteur qui se compose de trois antennes fixes placées orthogonalement les unes par rapport aux autres. Un second ensemble d'antennes est placé à l'intérieur d'un récepteur. En tenant compte de la valeur du champ magnétique terrestre on peut en déduire la position et l'orientation du récepteur dans le repère de l'émetteur : 4 valeurs (champ terrestre seul, champ terrestre+champ x,y,z) sont mesurés par les 3 antennes du récepteur (x,y,z). Le signal reçu du récepteur est échantillonné grâce à un algorithme contenu dans la machine et elle détermine la position du récepteur par rapport à l'émetteur.

Le principal défaut de ce type de capteur est la grande sensibilité du capteur par rapport à tout matériau magnétique ou appareillage électromagnétique qui distord le champ émis et rend la précision faible à leur voisinage. De plus la présence de câblage sur le système augmente la complexité de manipulation. Leur intérêt est bien sur leur coût peu cher, leur poids.

Solutions commerciales

Le Flock of Birds (FOB) est un capteur magnétique à six degrés de liberté, commercialisé par Ascension Technology Corporation. Il est constitué des 3 parties standards de ce type de technologie : émetteur, récepteur, boîtier de contrôle. Ce boîtier constitue l'interface avec l'ordinateur par l'intermédiaire d'une liaison série RS232.

Le Flock of Birds a une résolution de 2.5 mm et une précision de 0.07 mm (0.5 et 0.1 pour l'orientation) avec une fréquence de 100Hz. Ils permet une sortie de l'orientation sous différents formats : angle d' Euler, matrice de rotation, quaternion. Un extension nommé ERT (Extended Range Transmitter) permet d'étendre le champ de travail (8 pieds de rayon). On peut aussi citer le Polhemus solution concurrente à ce système ayant des caractéristiques un peu inférieur.

ddl	6 DDL (ou 3 suivant modèles)
résolution	moyenne
erreur	élevée
temps de réponse	faible
espace de travail	moyen
sensibilité	magnétique
rem	(-) câblage gênant (+) faible prix

TAB. 3.2 – Récapitulation des caractéristiques des capteurs magnétiques

3.1.3 Capteur à ultrasons

Il se compose de trois parties principales : un émetteur, un récepteur, et une unité électronique. L'émetteur est constitué de 3 haut-parleurs ultrasonores montés à environ 30 cm l'un de l'autre sur un bâti triangulaire fixe. De la même manière, le récepteur comporte 3 microphones montés sur un bâti triangulaire rigide plus petit. En connaissant la vitesse du son dans l'air, et par triangulation, on arrive à récupérer la position et l'orientation du récepteur. Les haut-parleurs sont actifs cycliquement et les 3 distances aux 3 micro sont calculées. Leur principal défaut est qu'ils nécessitent une ligne directe entre l'émetteur et le récepteur, car tout obstacle fait perdre le signal. De plus les réflections et variations de la vitesse du son dans l'air due au changement de la densité de l'air rend difficile le maintien de la précision.

ddl	6 DDL (ou 3 suivant modèles)
résolution	faible
erreur	élevée
temps de réponse	faible
espace de travail	moyen
sensibilité	champ sonore
rem	(-) câblage gênant (-) ligne direct E-R (+) faible prix

TAB. 3.3 – Récapitulation des caractéristiques des capteurs à ultrasons

3.1.4 Capteur optique

Pour du travail sur des grandes distances, les capteurs optiques sont nombreux et on peut donc les classes en diverses catégories suivant le type de lumière utilisé : visible, infrarouge, source laser. Cette technique est très employé dans les films et les réalisations de jeu pour animer des mouvements de personnages 3D.

Lumière visible : On citera tout d'abord le prototype de Ferrin qui utilise la lumière visible pour obtenir l'orientation de l'utilisateur. Une cible passive est placé sur le casque et la caméra tire un point de vue unique pour chaque orientation possible. La force de ce principe est d'avoir des capteurs passifs qui minimise le poids du casque. Les réflections et d'autres sources lumineuses engendre une erreur sur ce type de système. De plus l'occultation annule leur fonctionnement.

A cause des difficultés associé à la récupération d'informations de scènes non structurés, beaucoup de système optique rapide utilise des sources de lumières lasers ou infrarouge associés à des récepteur couplés.

Il existe principalement deux techniques. Dans la première technique, on fixe sur une personne différents marqueurs réfléchissants. On utilise une caméra munie de LED infrarouges qui éclairent la personne. On utilise plusieurs caméras qui enregistrent les mouvements de l'acteur, et a partir des images on peut extraire les marqueurs. L'ordinateur, à partir des différentes vue 2D, met en corrélation les différents capteurs pour déterminer le mouvement 3D des différents capteurs. Dans la deuxième technique, on utilise des marqueurs LED infrarouges qui sont alimentés par une pile (les capteurs sont auto-éclairants). On utilise une caméra infrarouge qui enregistre le mouvement des marqueurs. Le principal défaut de ces capteurs est un temps de traitement souvent long, et l'absence de réponse si un obstacle est présent entre la caméra et le marqueurs.

Les systèmes lasers encore très peu utilisés servent principalement pour des grandes distances de l'ordre du kilomètre.

Solutions commerciales

On peut citer les travaux de [15] avec un système optique a base de panneaux de diodes et de caméras infrarouges positionnés sur la tête de l'utilisateur. Le principal défaut de ce système est le poids des caméras fixés au casque de l'utilisateur. Le principal avantage est l'utilisation de plaques indépendantes et qui permettent l'extension de l'espace de travail. Ils ont étendu leurs travaux par le remplacement de la caméra par un système appelé HiBall d'une précision de 0.02 mm en position et de 0.03° en orientation.

On peut citer le système SM3D de SAGEIS fonctionnant par infrarouge utilisant 3 caméras montés sur un même bras. Ce système a été testé au laboratoire, et évalué pendant 1 semaine (3.1). Le placement des caméras et par un algorithme ad hoc on peut retrouver la position 3D des leds. Le système a une précision d'un micron.

3.1.5 Systèmes indépendants

Caméra

On place dans la scène à analyser plusieurs marqueurs visuels. Ces marqueurs sont filmés par 2 caméras, et leurs positions sur les images générées permet d'obtenir une reconstruction 3D de

Fig. 3.1 – Système SM3D

ddl	3 DDL (ou 6 Comb)
résolution	trés élevée
erreur	trés faible
temps de réponse	élevé
espace de travail	élevé
sensibilité	infrarouge
rem	(-) ligne direct E-R (-) investissement cher

Tab. 3.4 – Récapitulation des caractéristiques des capteurs optiques

la scène. Leurs principaux problèmes sont leur temps de traitement et le problème d'occultation (lorsqu'un objet masque les marqueurs) mais aussi un champ de travail limité au champ de vue de la caméra. Dans nombreux systèmes l'utilisent tel que [2],[16].

ddl	3 DDL (ou 6 Comb)
résolution	faible
erreur	élevé
temps de réponse	élevée
espace de travail	faible
sensibilité	nulle
rem	(-) ligne direct M-R (+) faible coût

Tab. 3.5 – Récapitulation des caractéristiques des systèmes à caméra

Capteur GPS

Utilisé principalement pour les applications de réalité augmentée extérieur. Elle se base sur la position retournée par le système GPS basé sur une mesure satellite.

ddl	3 DDL (position)
résolution	faible
erreur	élevé
temps de réponse	élevée
espace de travail	élevée
sensibilité	nulle
rem	(-) trop grande imprécision

TAB. 3.6 – Récapitulation des caractéristiques des capteurs GPS

Capteur inertiel

Il sont constitué de couples d'accéléromètre. La différence d'accelération ente deux accélerantes parallèles permet de trouver la rotation. L'accéléromètre mesure une accélération linéaire par rapport à un repère de référence, la vitesse, puis la position étant déduite par intégrations successives. Le principal problème de ce type de système est une dérive dans les mesures, et l'intégration pour trouver la position est alors soumis a l'erreur. La combinaison de 3 accéléromètres permet de retrouver une position. L'offre commerciale la plus retenu est le système InterTrax d'Intersense qui nous donne une précision de 0.02° avec une latence de 38 ms.

3.2 Systèmes de visualisation

Il servent à visualiser le monde virtuel. Une visualisation en stéréo permet une manipulation 3D dans l'espace virtuel. On peut les classer sur les critères suivants :
- résolution : définit la résolution de l'image reproduite.
- stéréo : définit une reproduction stéréo ou mono.
- support : définit le support physique de l'image généré.
- champ de travail : définit le champ d'utilisation du matériel de projection.
- tracking : correspond a la possibilité du système de lui associer un capteur pour suivre le mouvement de regard de l'utilisateur. Un système direct place directement le capteur sur le système (couplé à l'utilisateur) tandis qu'une solution indirecte privilégie le capteur sur l'utilisateur.

3.2.1 Écran passif

La première technique consiste à projeter une image virtuel sur un écran passif tel qu'un mur ou une table ou retro-projeter sur un écran translucide. C'est ce principe qui est utilisé dans dans le Responsive Workbench par rétro-éclairage. Le système peut générer de la stéréoscopie recréant une vision 3D. Mais on ne pourra jamais superposer un objet réel à un objet virtuel. Le système le plus connu est la CAVE qui utilise 3 à 6 écrans carré translucide disposé pour former une pièce cubique. Ces systèmes possèdent un champ de travail assez grand pour de nombreuses manipulations (typiquement espace de 2.5 m x 2.5 m x 2.5 m). La présence de coins sur l'écran introduit des distorsions importantes : seul l'utilisateur qui porte le capteur à une vue correcte.

résolution	élevé
stéréo	oui (mais vue commune)
support	mur, table (en général)
champ de travail	variable
tracking	indirect
rem	multi-utilisateurs mais une personne traqué

TAB. 3.7 – Récapitulation des caractéristiques des écrans passifs

3.2.2 Casques de visualisation

Une seconde technique trés utilisée consiste à placer des écrans miniatures sur un casque, et à utiliser un système optique pour renvoyer l'image vers les yeux. Actuellement, presque tous en mode stéréo, ils possèdent encore un faible champ de vision, du principalement au coût de cet technologie (les prix vont de 3 000 F à 700 000 F). Il existe diverses technologies, fonction du type d'écran utilisé. Les critères retenus pour les comparer sont en général (voir [17]) :

- champ de vue horizontal et vertical (FOV) en degré;
- résolution en pixels de couleur;
- résolution angulaire en minutes d'arc par pixel de couleur;
- la dépixelisation;
- un ajustement de la distance interoculaire (IPD);
- le poids;

Casque CRT

Les modèles CRT utilisent des tubes cathodiques miniatures. Ils sont placés latéralement à la tête de l'utilisateur. L'image est d'abord réfléchi par un miroir à 45° et prise en charge par une optique spéciale qui le dirige vers l'oeil de l'utilisateur. Le câblage électrique est très étudié car le CRT est alimenté en hautes fréquences et est situé très près de la tête.

Casque LCD

On utilise la technologie à cristaux liquides. Leur grand avantage est un faible poids. C'est la technologie retenue pour les futurs développement. Le casque disponible dans l'équipe est un casque I-glasses de virtual i-o ces caractéristiques sont : une résolution de 263x230 pixels avec une distance inter pupillaire non ajustable, un champ de vision 30°x23.6° avec un résolution angulaire de 6.84.

résolution	faible
stéréo	oui (en général)
support	casque
champ de travail	élevé
tracking	direct
rem	(-) lourds et dangereux

TAB. 3.8 – Récapitulation des caractéristiques des casques CRT

résolution	faible
stéréo	oui (en général)
support	casque
champ de travail	élevé
tracking	direct
rem	(+) peu coûteux

TAB. 3.9 – Récapitulation des caractéristiques des casques LCD

Chapitre 4

État de l'art

On présente dans ce chapitre de façon méthodique les divers sujets de recherche de la réalité augmentée. On s'intéresse aux différents systèmes mis en place, mais aussi aux sujets précis de recherche, pour en tirer les difficultés rencontrées et les principes à retenir.

On décrit tout d'abord différents systèmes de réalité augmentée pour spécifier les besoins d'un tel système, avec une attention particulière pour les systèmes multi-utilisateurs. On s'intéresse ensuite à celle orienté vers le domaine du jeu. Cette étude nous conduit aux difficultés d'alignement, les problèmes matériels, et la latence. On se penchera aussi sur les problèmes d'occultation entre objet réels et virtuels. Un dernier point traité est l'interaction 3D et la gestion du multi-utilisateurs. Notre étude repose principalement sur le document de référence que constitue [18].

4.1 Système de réalité augmentée

4.1.1 Présentation

On se basera sur le travail de [7] et son prototype qui soulève les principaux problèmes de la réalité augmentée (on citera aussi les travaux de [9]). Il utilise un casque avec un capteur Polhemus lui permettant de suivre le mouvement de l'utilisateur (4.1). Il récupère des images à ultrasons à partir d'une sonde et après reconstruction 3D d'un volume de visualisation des données à partir de coupes scannées, il le projette dans le casque de l'utilisateur. Il combine l'image réel obtenu grâce à une caméra, avec l'image virtuel.

On voit qu'une des premières étape consiste à positionner les images virtuels générés sur le corps du patient. Il faut donc positionner les données virtuels par rapports à des repères réel tel que le corps du patient. Il faut aussi positionner la position de la tête de l'utilisateur et du pointeur dans cet espace, ainsi que tous les objets virtuels. Un mauvais alignement donnera un système globalement non cohérent et non réaliste (défini comme erreur statique dans [19]). Une approche consiste donc à modéliser le système ainsi : On définit un repère global (repère du monde) et toutes les objets sont définis dans ce repère (voir [20]). On étend l'exemple précédent par toutes les combinaisons possibles de ces systèmes comme visible sur 4.2.

La transformation entre le repère émetteur↔récepteur est mis a jour en temps-réel par le système. Les autres transformations, rigide, nécessitent une calibrage. On peut les décomposer de la façon suivante :

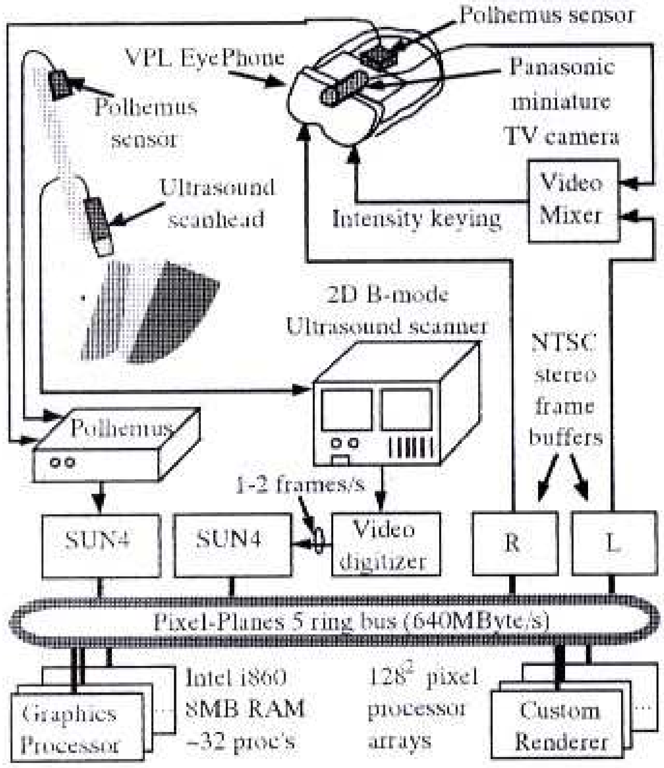

Fig. 4.1 – Schéma du système présente par Bajura

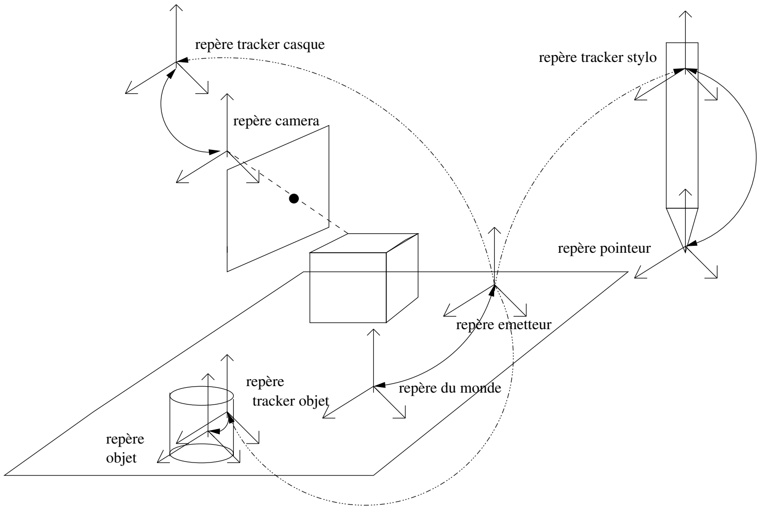

Fig. 4.2 – Systèmes de coordonnées et leur relations

1. calibrage émetteur↔monde : définit la transformation entre le repère global et le système de suivi ;
2. calibrage récepteur↔pointer : définit la transformation entre le capteur positionné sur l'outil d'interaction et le point de contact de manipulation de l'outil ;
3. calibrage récepteur↔caméra virtuel : définit la transformation entre le capteur et la caméra. On considère la caméra définissant la position des yeux de l'utilisateur ;
4. calibrage stéréo : non représenté sur le schéma il définit la transformation entre l'oeil gauche et l'oeil droit. On définit l'IPD : interpupilar distance, qui représente la distance entre les 2 yeux. Ce paramètre peut être lié au système de visualisation ;
5. calibrage paramètre interne du système de projection : ce sont principalement la distance focale (focal), le champ de vue (FOV horizontale et verticale), distance de projection hors axe pour la stéréo (off-axis), résolution de l'écran, distorsion de l'image (pour les notations voir [20] et [21]).

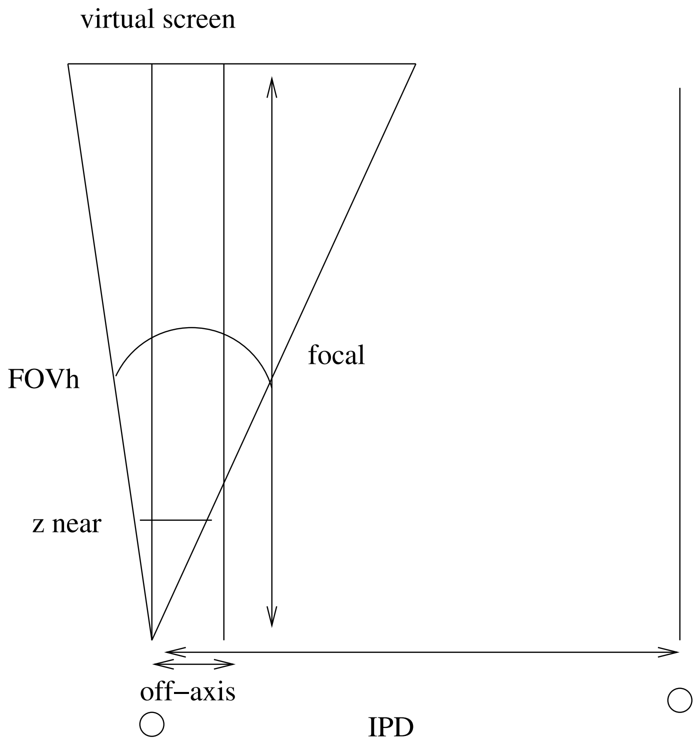

FIG. 4.3 – Système de coordonnées pour le casque. On note que z-near est le plan de projection utilisé pour faire le rendu.

6. calibrage objet↔monde : définit la transformation entre un objet (virtuel ou réel) par rapport au repère du monde.
7. calibrage objet↔capteur : définit la transformation entre un objet et un capteur positionné sur cet objet (pour objet dynamique).

Un des principal problèmes posé par son système est la latence. La latence peut-être défini comme le temps entre le mouvement physique de l'utilisateur et la réponse visuelle correspondante. Avec une forte latence, un mouvement tel qu'une rotation de la tête donne l'impression qu'un objet statique dans une scène se déplace suivant le mouvement. De même le déplacement manuel d'un objet virtuel résulte d'un objet qui suit "péniblement" le mouvement de la main.

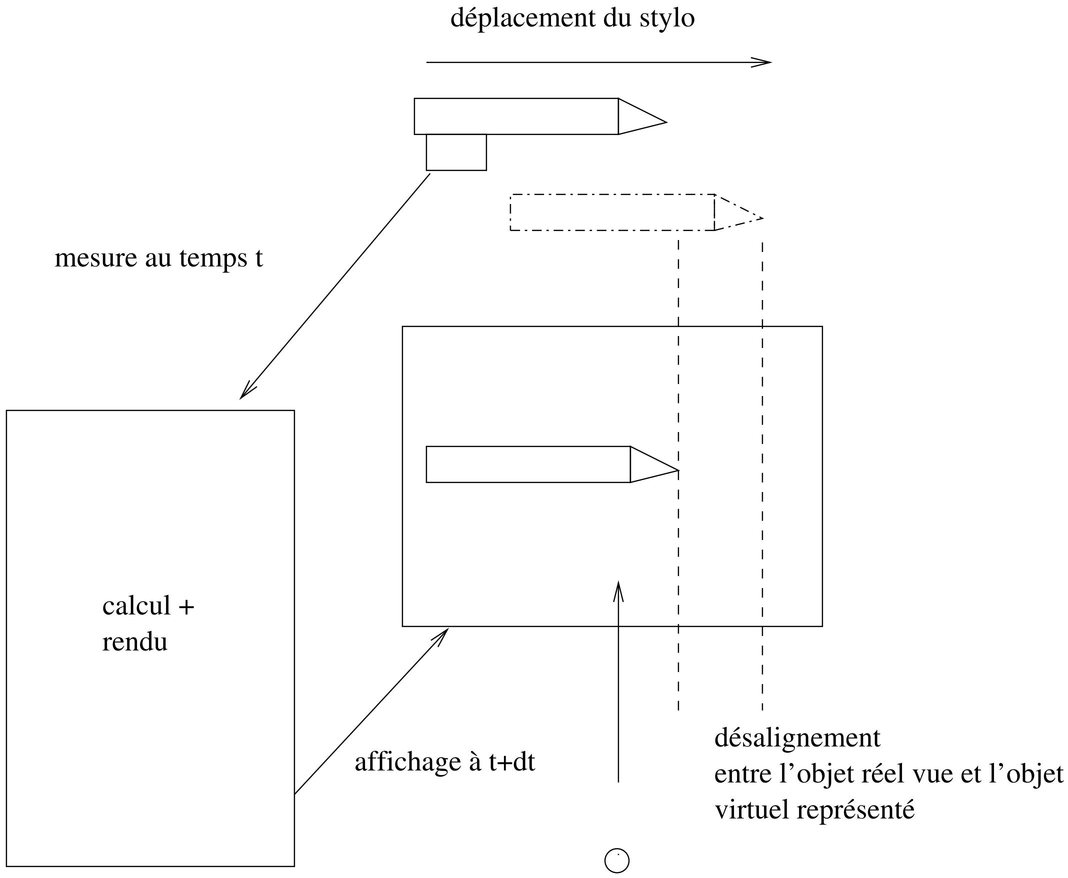

Fig. 4.4 – Problème de la latence

Lié à ce problème on trouve la faible qualité des systèmes de suivi. En plus d'avoir un long temps de réponse ces systèmes souffrent de bruit. Le résultat est des objets qui semblent "flotter" dans la scène toujours en mouvement malgré le positionnement fixe de l'utilisateur.

Bajura soulève aussi un problème lors de l'affichage de son image : tout objet virtuel généré masque tout objet réel situé dans le même alignement de la vue de l'utilisateur : les objets réels positionnés devant (dans le sens directionnel de l'utilisateur) l'objet virtuel correspondant. Ce conflit visuel conduit alors à une incohérence de la scène. Pour résoudre ce problème il est nécessaire d'avoir des informations de profondeur sur les objets du monde réel : il faut que le système gère les occultations. De plus le rendu non réaliste des objets virtuels provoque une discontinuité visuelle entre les objets : un rendu réaliste basé sur l'éclairage de la pièce permettrait une meilleur intégration (on peut citer les travaux de [22] ,[23],[24]).

4.1.2 Multi-utilisateurs

L'intégration d'un nombre plus élevé d'utilisateurs dans un système de réalité augmentée conduit a de nouveaux problèmes. Un des systèmes sur lequel on repose notre étude est le "studierstube" ([25],[26]). C'est un environnement multi-utilisateurs basé sur des capteurs magnétiques, dédies a la simulation physique . Un nombre important d'utilisateurs nécessite plus de matériel connecté ensemble (capteurs,casques), mais aussi la prise en compte de l'interaction au niveau de la gestion de la scène. Chaque utilisateur acquière un point de vue propre sur le monde et par là même défini une notion de monde virtuel privé. Une gestion des conflits est nécessaire pour résoudre l'accès concurrents aux objets.

Pour des environnements non immersifs, un besoin d'interface d'interaction est nécessaire pour faire face a la possibilités d'un nombre élevé d'actions. Une solution consiste à utiliser une boîte à outils 3D. Mais de telles solutions s'averent impraticable en RV.

Regardons plus particulierement ce qui en RA a été fait pour le jeu. Szalavari dans [1] c'est aussi intéressé à l'implication des jeux vidéos dans un système de réalité augmentée. Il utilise l'environnement "Studierstube" pour faire interagir des joueurs dans un même environnement. Pour répondre aux besoins du jeu il introduit différents concepts. Un concept de régions qui définit une partition spatiales de l'environnement et un concept de couche avec des droits sur ces différentes couches pour chaque utilisateur (il utilise un système de clé pour obtenir une vue spécifique a chaque utilisateur). L'interaction se fait via une interface de manipulation à 2 mains nommé le PIP (Personal Interaction Panel), et il utilise un système d'aimant pour combler la faible précision de leur pointeur du a l'utilisation de capteurs magnétique. Il considère que l'habileté à générer des informations différentes à chaque joueur, une communication social direct, et une interaction précise et rapide sont les facteurs clés pour un jeu en réalité augmentée.

4.2 Erreur statique

On peut définir l'erreur statique par l'erreur d'alignement entre objet réel et objet virtuel, défini lorsque l'utilisateur est en position statitique.

4.2.1 Erreur de suivi

Livingston propose l'utilisation d'une "look-up table" pour calibrer la position et l'orientation d'un capteur magnétique. Il utilise un bras Faro, pour la construction de la table, en comparant les valeurs précises obtenus. Comme Bryson il montre que les résultats obtenus réduisent de 79% l'erreur moyenne obtenu. Mais les résultats sont très mauvais pour l'orientation et la réalisation d'une table plus précise est trop coûteuse (en terme de mémoire).

La solution actuellement retenu par ces auteurs est un système hybride tel que dans [16]. C'est en général une combinaison entre un système de magnétique très peu précis et rapide, et un système de vision très précis mais lent. Cet solution est inutilisable pour un système "see-through". Les principaux défauts sont la synchronisation entre les 2 systèmes, et la difficulté d'analyse d'image sous un environnement très peu éclairé. La societé SAGEIS utilise un système basé sur une table mais fait une interpolation bilinèaire, et un banc de mesure métrologique. Avec ces collections le système optique atteint une précision de mesure l'ordre de 0.5 mm.

4.2.2 Erreur de paramétrage : calibrage

Calibrage caméra

Les paramètres internes de la caméra virtuels sont :
- focale : distance de projection ;
- FOV : champ de vue. il est définit par 2 valeurs horizontale et verticale ;
- centre de projection : en général aligné dans le centre optique, il est hors axe (off axis) pour un système de casque transparent (voir 4.3). La principal cause est l'absence d'alignement entre l'axe de l'oeil et l'axe du casque utilisateur (voir [20]). Une solution consiste a régler l'IPD du casque pour obtenir un alignement. L'autre consiste a réaliser une projection "off axis" ([21]).

On peut classifier les méthodes de calibrage en différents groupes :

1. ajustement manuels. L'utilisateurs ajuste les paramètres pour un point de vue donnée. Comme citer dans [19] ceci marche pour un point de vue fixe et le résultat peut être complètement incohérent a un autre point de vue (ceci ce justifie par l'ensemble des plans de projections possibles dans l'angle de vue de la caméra valide dans la pyramide de vue).
2. mesures avec un appareillage et des capteurs. Utilisé principalement par Janin en utilisant l'image obtenu à l'aide d'une caméra placée derrière un casque. Un outil peut servir à mesurer la distance inter pupillaire et ainsi régler ce paramètre pour un alignement des axes oeil/casques.
3. taches basé sur la vue. L'utilisateur effectue quelques manipulations pour régler les paramètres sous certaines contraintes. On peut citer les travaux d'Azuma ([19],[27]) qui effectue diverses opérations à l'aide d'une boite en bois. Il calcule tout d'abord le décalage hors axe entre la vue de l'utilisateur et le centre de l'écran par l'alignement manuel dans la pyramide de vision de l'utilisateur d'un curseur rectangulaire 2D. Il utilise aussi un principe simple pour le calcul de la transformation entre le capteur et l'oeil : Un alignement d'un point réel et virtuel lui donne des contraintes de transformations lui permettant d'en déduire la rotation entre les 2 repères. Un second alignement lui donne entièrement la position). Mais les contraintes utilisateurs sont plutôt difficiles à réaliser pour un alignement précis. [28] s'intéresse lui au paramètre de mauvais alignement mécanique et d'alignement hors axe. Ces manipulations font appel à un système mécanique complexe basé sur l'affichage de leds sur un support réel. La méthode est plutôt complexe et comme celle d'Azuma nécessite un équipement extérieur précis. On peut citer pour finir [29] qui utilise 2 capteurs pour réaliser la calibrage. La publication n'est pas très précise sur le calcul des paramètres internes. De plus leur calibrage est valable par rapport à leur 2 capteurs, mais pas par rapport au monde réel (même si une calibrage émetteur<->monde n'est pas d'une grande complexité).
4. système a base de vidéo. Cette technique très employé en robotique à été repris ici dans le cadre de la réalité augmentée. On citera [30],[11],[31].

Calibrage d'objets réels

On peut citer les travaux de Tuceryan et Whitaker dans [32]. Ils utilisent une approche par vision et une approche par saisie de points d'intérêts sur l'objet. Le problème est à mon avis, un choix précis des points de contrôle, qui est dépendant de chaque objet. Le suivi d'objet n'est possible que si aucune occlusion partiel existe entre le marqueurs et la caméra. Ce système est repris par [29]. Il ajoute un calibrage manuelle pour la transformation capteur<->objet réalisé par l'utilisateur au clavier.

4.3 Erreur dynamique

On regroupe sous le nom "d'erreur dynamique" les problèmes qui s'ajoutent quand on bouge le capteur (ou l'emetteur).

4.3.1 Latence

La latence est des principaux problèmes actuelle. Une première approche consiste à réduire le temps de rendu, mais ceci nécessite une architecture logicielle et matérielle très robuste. On peut citer les travaux de [33] qui optimise le rendu matériel sur une Pixel-plane 5. Une autre optimisation consiste à synchroniser les différents capteurs.

De nombreux travaux sont basés sur la prédiction de la futur position comme dans [19]. Peu de travaux ont été réalisé en RA, mais beaucoup en RV. La difficulté consiste à paramétrer le filtre de Kalman employer. Des études sont aujourd'hui faite basé sur les mouvements de la tête.

4.3.2 Occultation

L'occultation a été abordé dans diverses publications et pour divers contextes. Tout d'abord State qui reconstruit un modèle de la surface de la peau à partir d'échantillons et part une triangulation de Delaunay : ceci nécessite une manipulation difficile, et une reconstruction imparfaite et bruité dù au mouvement pendant la saisie des points. Une méthode plus performante par un système de mémorisation d'objet à partir d'images est réalisé dans [34]. Mais son système est très contraint et nécessite un ordinateur personnel pour l'utilisateur. Quand à Fuhrmann [35] il s'est intéresse part l'occultation de personne réel par l'utilisation de "phantom", étant une des premières approches pour l'occultation de chaîne articulée.

4.4 Réalité virtuelle multi-utilisateurs

Le jeu collaboratif présenté par "Studierstube" reprend les avantages de la réalité augmentée appliqués au jeu. Malheureusement il n'utilise que très peu le mélange entre objet virtuel et objet réel, ne tient pas compte des possibilités graphique supplémentaires apportés (animation, effets spéciaux). De plus il se limite à des interactions de déplacement entre 2 surfaces, et ne tient pas compte d'autres types d'interaction. Leur interaction avec l'environnement se limite principalement à une tablette, et il vise surtout le côté communication sociale. Ted Starner c'est aussi intéressé au jeu basé sur un Responsive Workbench (dans [3]. Il développe un système de reconaissance de geste (détection par vision) et de reconaissance de sons, qu'il associe à un déclenchement d'action dans le jeu. Son concept de jeu se limite à des déplacement simple de pions. Ar2 Hockey [2] est un jeu pour deux utilisateurs basé sur un concept de hockey sur table. Le système très lourd matériellement, se limite dans les possibilités de jeu à un déplacement d'un palet. La gestion de plusieurs utilisateurs dans un système de réalité augmentée, se rapproche des travaux effectués pour les systèmes de réalité virtuelle. On peut citer Repro-3D[36] ou RING [37] mais ces systèmes ce limite a des concepts de distribution et de vue locale et ne s'intéresse pas à l'interaction multi-utilisateurs.

4.5 Interaction 3D

De nombreux laboratoires s'intéresse à la manipulation et l'interaction avec des objets 3D. On peut citer les travaux de Zhai dans [38], [39], [40]. Il a réalisé de nombreux expériences pour évaluer des périphériques d'interaction 3D. Il utilise la taxonomie visible sur la figure 4.5.

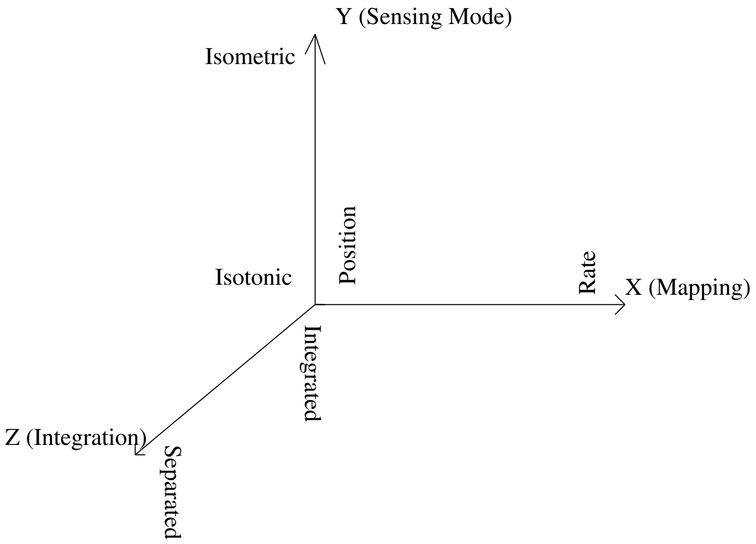

FIG. 4.5 – Taxonomie pour l'interaction 3D. L'axe X défini le type de réponse en fonction du mouvement. L'axe Y défini le retour musculaire du système. Le dernier axe (Z) défini si les composantes de la mesure nécessite un ou plusieurs capteurs.

Son étude conduit à prouver que la combinaison idéale est un système isotonique/ position ou isométrique/vitesse, même si l'évaluation reste difficile. Une idée d'un autre scientifique consiste à dire que lors de gros déplacement on utilise la vitesse et pour des petites déplacements la position. [41] [42] s'intéresse à la manipulation dans un environnement de réalité virtuel. Des métaphores et des systèmes ingénieux sont évoquées tel que "World in Miniature". Il souligne la difficulté de manipulation 3D, dù au manque de précision et à l'absence de retour d'effort.

Mais peut de travaux font références à la réalité augmentée et son interaction 3D. On peut citer [43] qui utilise un système à 2 mains : le PIP : Personnal Interaction Panel. C'est une tablette de manipulation et de support d'informations. Mais il se limite à la taille de la tablette et le fait de devoir la garder à la main constament peut devenir fort génant.

Chapitre 5

Architecture proposée et implementée

5.1 Matériel

L'espace de travail a été défini au chapitre 1. Le travail nécessite la mise en place de gestion des différents périphériques associés au différents utilisateurs :

- gestion des interactions en entrée. On se limite dans notre étude au capteurs magnétiques, la souris et le clavier. Un capteur est fixe sur une périphérique d'interaction et l'autre sur le système de sortie visuel.
- gestion des interaction en sortie. On se limite aux écrans de contrôles, au casque stéréoscopiques I-glasses.

5.1.1 Gestion du capteur Flock of Birds

Plus concrètement un driver de gestion du FOB obtenu durant mon stage d'été a été optimisé et permet la gestion de multiples FOB. Un réglage interne des filtres permet un filtrage précis. Ce réglage s'effectue par la modifications de paramètres internes aux capteurs (mais un mauvais réglage induit un temps de latence très important). Un processus séparé est chargé de récupérer et de traiter les données en provenance du Flock of Birds en parallèle avec l'application principale. Une section critique courte permet d'assurer la synchronisation, et donne la cohérence de lecture.

5.1.2 Gestion du capteur SAGEIS

Un étude est réalisé par l'utilisation du capteur SAGEIS dans notre environnement (le système étant nouveau, peu de travaux ont été effectué sur ce capteur). Le temps de prêt est très limité, on choisi donc de réaliser des mesures pouvant être resimulé plus tard dans notre application.. On s'interesse à la récuperation des valeurs mesurés sur notre station de travail (liaison série entre un PC et une station SGI). On réalise donc un pilote de gestion optimisé du système SM3D (lecture dernier valeur transmise, timestamp, émission brut des données). Des tests de base sont réalisés pour l'étude du champ de travail, de la précision, du bruitage. Comme spécifier les résultats se montrent cohérents avec les valeurs de spécification, mais le principal défaut est l'occultation trop rapide (des leds) provoqué par ce type de système. Un outil nommé "étoile", constitué de 4 leds, nous permet de mesurer une orientation par rapport au repère locale de cet outil. Un calibrage sommaire est éffectué entre cet outil et une caméra vidéo, qui nous permet d'avoir une simulation cohérente. On utilise une led supplémentaire pour effectuer des manipulations à l'aide du doigt, la led lui étant rigidement lié. On établi différents scénarios de manipulations (navigation, déplacement de pion, ajout d'objet réel), et on effectue des mesures à partir de ces scénarios (voir figure 5.1 et 5.2). On génere ensuite une simulation dans notre application à l'aide des données du capteur (à l'heure actuelle on resimule juste les données de navigation).

FIG. 5.1 – Manipulation avec SM3D. Le doigt est equipé d'une LED permettant son suivi.

Une génération d'image stéréoscopique est produite pour le casque. Ceci nécessite la génération d'une image entrelacé ou chaque champ correspond à une vue spécifique (gauche ou droite). Une sortie vidéo est alors produite.

Le système étant critique par le fait de minimisation de la latence, ces 2 phases doivent être optimal au niveau temps.

Pour l'interface tactile on privilégie un pointeur comme outil de manipulation. Dans notre cas la précision de déplacement étant recherché on privilégie un mode isotonique/position de contrôle de la manipulation. Pour faciliter l'interaction on se place dans une optique ou on privilégie le maximum de contact avec l'environnement réel, nous fournissant un retour d'effort convenable.

On se place dans un environnement réel basé sur des objets horizontaux. On considère que la surface de travail est horizontal et que le centre de l'émetteur et le centre du capteur sont en leur centre de géométrique.

5.2 Logiciel

On s'intéresse à un développement d'un environnement de réalité augmentée. On peut le définir par le fait qu'il :

- permet de gérer plusieurs applications de type différent ;

FIG. 5.2 – Manipulation avec SM3D. Le stylo est equipé d'une LED permettant son suivi.

- permet d'avoir une indépendance du matériel lors de la conception de programme;
- permet de gérer plusieurs utilisateurs;
- facilite la mise en place d'application.

On souhaite une indépendance de l'utilisateur par rapport à la manipulation du monde représenté. On met en place une architecture d'environnement (5.3) de réalité augmentée basé sur :

- une notion d'**entité physique** défini par la personne qui possède des sens d'entrée (tactile, commande vocale) et des sens de sortie (visuel, sonore) couplés avec cet environnement. Suivant le choix et les possibilités matériels l'utilisateur peut avoir un retour visuel (sur casque ou écran).
- une notion de **monde virtuel**. On défini dans cet espace les objets virtuels et les objets réel sous la même représentation.
- une notion d'**objet interface entre ces 2 entités**. C'est par exemple le cas de la caméra virtuel qui représente la vision de la personne dans le monde réel et le monde virtuel.

L'objet interface est indépendant du périphérique matériel associé à cet entité. Un scénario peut par exemple être le suivant :

- une personne avec retour visuel sous casque i-glasses suivi par un FOB et une interaction tactile avec un stylo pointeur suivi par un autre FOB.
- une deuxième personne avec interaction tactile avec le doigt suivi par un SM3D.
- une personne (passive) avec retour visuel sur écran vidéo.

On résout la difficulté d'interaction par un retour d'effort simple, basé sur le fait que l'on intègre l'interface à la table. On privilégie une manipulation a une main (contrairement au PIP). On considère que l'information peut être affiché n'importe ou sur l'environnement de la table. On développe une boite outil 3D dans ce contexte, en se basant sur des manipulations 2D mais

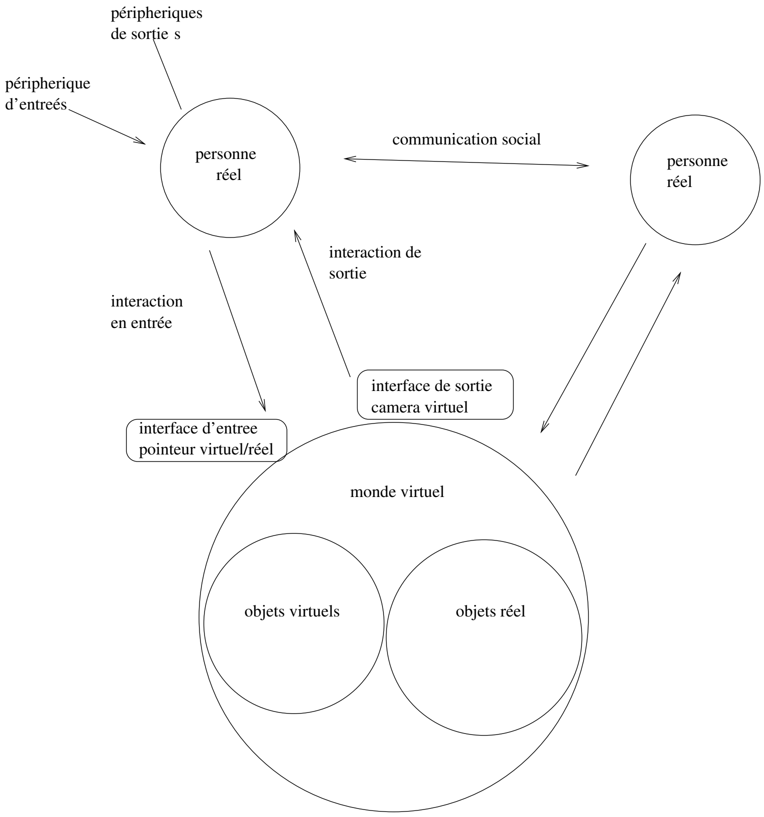

FIG. 5.3 – Représentation de l'environnement de réalité augmentée

des représentations 3D d'éléments qui ajoute de l'information. On obtient les éléments suivants :
- **bouton poussoir** posé sur support réel plan. Des annotations textuelles peuvent être posé sur ce bouton. L'enclenchement et le déclenchement du bouton est visualiser par un déplacement en hauteur du bouton.
- **bouton à 2 états** reprenant le principe précédent.
- **sliders horizontaux ou verticaux** permettant un changement de valeur dans un espace bornée.
- **boutons radios** définis comme un ensemble de boutons à 2 états mutuellements exclusif. Il sert en général à définir l'action courante de l'utilisateur. la sélection courante est visible par un changement de l'aspect visuel du boutons sélectionnées : couleur et aspect. La représentation des boutons se fera soit sous forme géométrique simple, soit sous forme géométrique annotée (ajout de texte descriptif de l'action), soit sous forme complexe englobée dans un volume cubique. Cet dernière solution permet une meilleur assimilation de l'action de ce bouton (c'est l'équivalent des icônes en 2D). On garde de plus la forme géométrique englobante pour simplifier tout les calculs d'interaction.
- **des menus** pour un nombre d'actions importantes (5.4 et 5.6, 5.7, 5.8). On privilégie un menu 3D posé sur un support plan qui s'ouvre sur la hauteur (le déplacement s'effectuant d'abord sur un plan puis sur la hauteur, permettant d'eliminer la sasie direct en 3D). Ceci favorise l'interaction en général difficile avec des menu 3D flottants. On ne considère que des menus de profondeur 1, les possibilités de placement d'une barre de menu dans divers endroits de l'espace nous préserve de menu d'une profondeur plus grande. Si on considère des élements 3D de surface de 2cmx2cm une dizaine de menu peuvent être placé pour rester dans l'espace de travail de l'utilisateur. Avec 5 actions par menu, une cinquantaine d'action est disponible. On reprend le principe précédent d'afficher des boites englobant de l'action choisi avec des éléments de menus représentés sous sous forme - d'icone - graphique 3D (on retrouve en général cet représentation dans des logiciels commerciaux complexe tel que Maya, 3DSTUDIO MAX, et les logiciels de Metacreations tel que BRYCE et CANOMA).

Ces travaux étendent ceux de [43] sur l'utilisation d'un support 2D avec un pointeur suivi et d'une interface 3D. Mais on ne se contraint plus à une manipulation à 2 mains sur un support unique, on considère plutôt un e manipulation à une main sur un support plan. On ajoute des boites d'outils et des menus d'interactions orientés réalité augmentée améliorant les services disponibles.

5.2.1 Représentation des objets virtuel

La description de la boite à outils citée précédemment a nécessité la mise en place d'une représentation dédiée aux objets dans notre environnement. Cet représentation permet la manipulation d'objet en tant qu'élément de base de notre monde. On s'attache à une représentation identique pour les objets virtuels et réels. La nuance étant qu'un objet réel a sa propre apparence et géométrie initial pouvant être augmentée, tandis qu'un objet virtuel n'a pas de possibilités d'augmentation. On s'intéresse de plus à des visualisations différentes de ces objets. On utilisera le schéma de représentation visible sur 5.9 (il s'appuie sur un schéma classique de modélisation de contexte virtuel mais appliqué à la réalité augmentée).

Un objet est donc défini par :

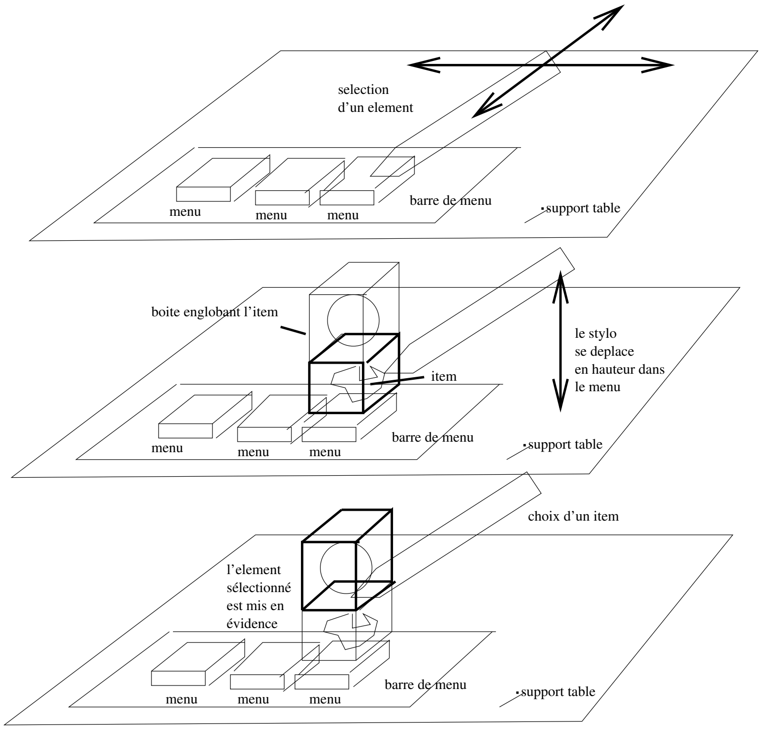

Fig. 5.4 – Représentation des menus virtuels

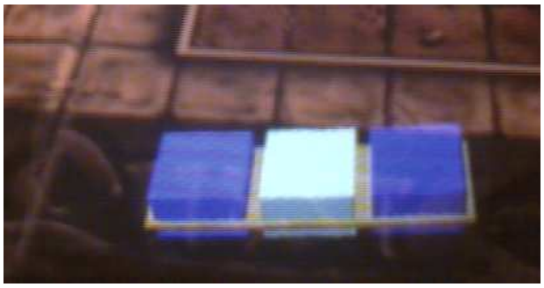

Fig. 5.5 – Boutons radios utilisé pour définir l'action courante

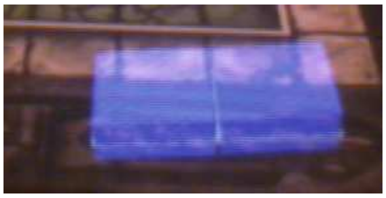

Fig. 5.6 – Représentation des 2 menus fermés

FIG. 5.7 – On a ouvert le menu de gauche par un clic sur ce menu. On peut ensuite par un déplacement en hauteur sélectionner l'item correspondant.L'item courament sélectionner est en surbrillance.

FIG. 5.8 – Déplacement dans le sous menu en horizontale sur le 2ème élement. On voit en transparence l'item contenu dans la boite englobante.

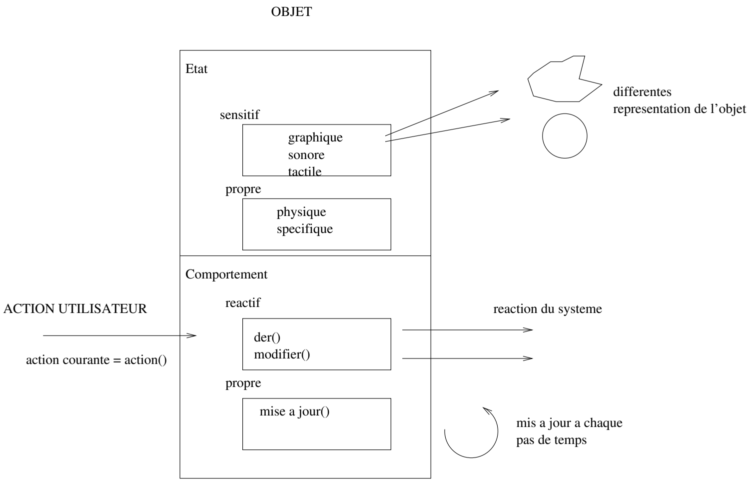

FIG. 5.9 – Représentation d'un objet

- des attributs définissant les propriétés de l'objet, regroupables en 2 classes :
  - les attributs propres définissant les caractéristiques physiques de l'objet. Pour une voiture ce peut être sa vitesse, son poids, sa résistance au choc ..etc..
  - des attributs sensoriels tel que nous les percevons. Ceci défini les attributs visuels, sonores de l'objet (au niveau informatique on le définira par sa couleur, réaction à la lumière, textures).
- le comportement spécifique de l'objet, distinguable sous 2 formes :
  - un comportement réactif à certaines actions de l'utilisateur tel que le déplacement, la saisie, la déformation de l'objet.
  - un comportement propre de l'objet régissant des lois naturels (on peut citer la cinématique, réactions aux forces physiques, oscillations .. etc. ..).

A chaque utilisateur on associe une action courante (action en cours), et une action selectionné (qui peut L'utilisateur manipule et interagit avec ces objets suivant ce qu'on défini par son **action courante** et son **action sélectionné**. Considérons que l'utilisateur sélectionne le mode 'déplacer'. Après la saisie d'un objet, le comportement réactif déplacer() est appelé sur l'objet, et l'action courante de l'utilisateur est 'déplacer'. Ce schéma simple permet de gérer n'importe quel objet.

On souligne le fait que :
- certains objets ont un comportement unique quelque soit l'action ;
- tous les objets n'implémentent pas tous les comportements réactifs, et les comportement sont extensibles suivant l'ajout d'action utilisateurs.

5.2.2 Intégration de plusieurs utilisateurs

La gestion de plusieurs utilisateurs respecte le schéma type de lectures des entrées, mise à jour de l'environnement, retour à l'utilisateur. Plus précisément pour chaque utilisateur :
- lecture des paramètres d'entrée de l'utilisateur. On se limite dans notre cas à la lecture de la position d'un pointeur réel (sens tactile).
- Suivant l'action courante de l'utilisateur, on appelle soit la méthode correspondante (ex : déplacer) sur l'objet couramment sélectionner (ex : voiture virtuel), soit on est dans un état inactif. Dans ce dernier casm on réalise alors un test d'intersection sur la scène visible de l'utilisateur.
- On met a jour la caméra virtuel de l'utilisateur.

On met de plus à jour les objets ayant un comportement propre.

Cet architecture très abstraite est tout de suite justifiable par le fait qu'elle permet une intégration avec de nombreuses applications collaboratives. Pour une réunion de travail, une représentation virtuelle d'une maquette permet à plusieurs personnes de communiquer sur les différents points de la maquette. Les possibilités de changement visuel (navigation + aspect de l'objet) apporte un plus à une réunion avec des plans et une maquette physique sommaire. Pour des opérations chirurgicales, une représentation à différents détails pour plusieurs médecins permet une séparation du travail. Pour une application scientifique le paramétrage et la manipulation interactive de la simulation est faciliter par le travail dans le même espace de travail. Pour le loisirs, les possibilités sont nombreuses tel que des jeux multi-utilisateurs, la navigation sur Internet par une interface placé sur une table ..etc..

Pour les problèmes spécifiques à la gestion multi-utilisateurs se reporter au chapitre sur la gestion Multi-Utilisateur.

Chapitre 6

Erreur statique et dynamique

On choisi une méthode par taches de la part de l'utilisateur pour résoudre les différents problèmes de calibrage statique. L'utilisation d'un casque semi-transparent et l'absence de caméra nous oblige à un calibrage utilisateur (voir les justifications dans [29]). On vise de plus une réalisation de taches très simple et très intuitive reposant sur des règles géométrique. Notre condition est que les paramètres statiques (FOV casque, IPD de la personne) ne nécessite pas une recalibrage car ils sont immuables dans le temps, tandis que les paramètres dépendants de mesures (transformation tête$\leftrightarrow$capteur) nécessite d'être recalibré. La précision recherche est de l'ordre de 0.5 cm (la méthode n'est pas très robuste, mais ne constitue pas le point clé du sujet de stage).

6.1 Erreur statique

6.1.1 Transformation capteur$\leftrightarrow$casque

Réglage des paramètres internes

La tache consiste à trouver les paramètres internes du casque qu'on utilisera pour la caméra virtuel. Les différents paramètre interne du casques sont :

- distance focale du plan virtuel ;
- FOV horizontale et verticale ;
- position du milieu du plan image ;
- résolution du casque ;
- position hors axe de l'oeil par rapport au centre optique du casque. Ce paramètre provient du fait que l'oeil n'est généralement pas aligné avec le centre optique ;
- désalignement mécanique du casque ;
- distorsion dû au système optique utilisé.

Les paramètres disponibles du casque sont distance focale (3.96 m), FOV horizontale (30°), FOV vertical (23.6°), résolution du casque (263x230 pixels). On s'intéresse principalement au décalage hors axe (off axis), les autres problèmes n'étant pas dépendant de l'utilisateur (pour des solutions nous vous conseillons [29]). La distance entre les 2 centres optiques est de 6.2 cm (mesure standard de distance inter pupillaire). Si on connaît la distance inter pupillaire d'une personne, les 2 variables sont relies par la relation :

$IPD_{personne} = IPD_{casque} + x_{offaxis}$

On obtient donc la distance hors axe entre les 2 centres en considérant la distance inter pupillaire disponible pour chaque joueur.

Pose de la caméra

On considère que les paramètres internes ont été correctement évalués. On considère que la transformation tête↔oeil est juste. Le calcul de la position de la caméra dans le repère du monde est évalué d'après la relation :

$$T_{MondeOeil} = T_{MondeEmetteur} * T_{EmetteurRecepteur} * T_{RecepteurTete} * T_{TeteOeil}$$

On cherche à évaluer $T_{recepteurtete}$.

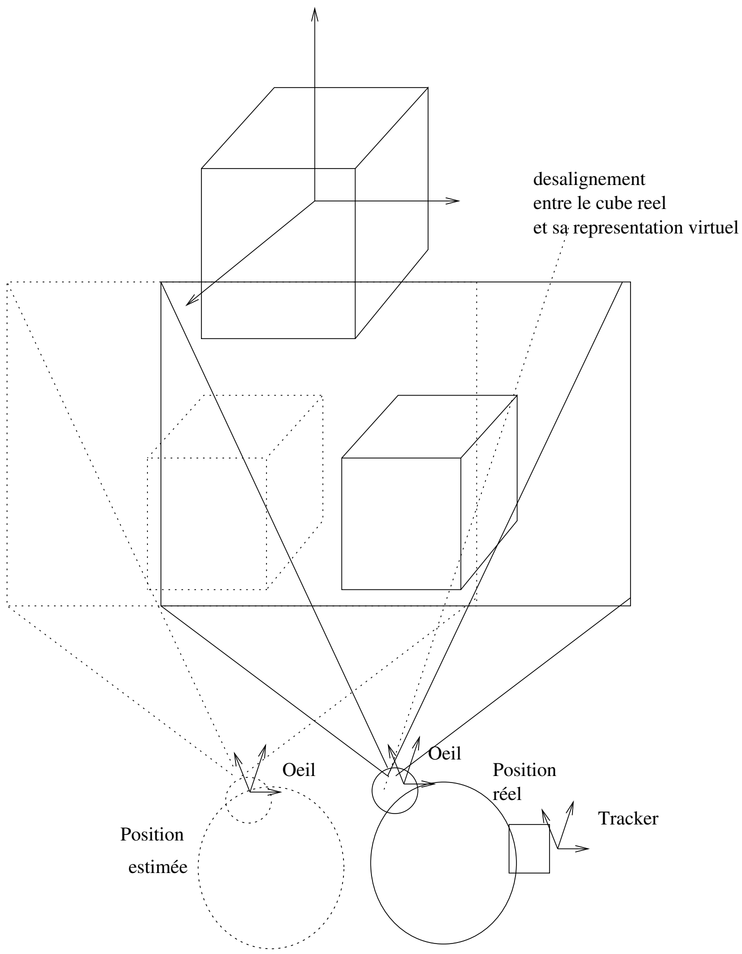

FIG. 6.1 – Alignement incorrect dû à une mauvaise estimation de la transformation tête↔oeil

On considère que la transformation $T_{MondeEmetteur}$ est l'identité : on l'évaluera dans la suite de la calibrage, on considère ici le repère du monde aligné avec le repère du capteur. La relation se simplifie donc en :

$$T_{MondeOeil} = T_{EmetteurRecepteur} * T_{RecepteurTete} * T_{TeteOeil}$$

Le principe consiste à figer le point de vue de l'utilisateur d'une scène représentant un cube c'est à dire :

$$T_{MondeOeil} = cst$$

La projection du point de vue courant est figé et l'utilisateur la corrige en déplacant sa tête jusqu'au bon alignement. La vue projeté est unique et correspond a un seul point de vue du cube réel (on peut se le représenter comme le sommet d'une pyramide ou la vue projeté est une section de cet pyramide).

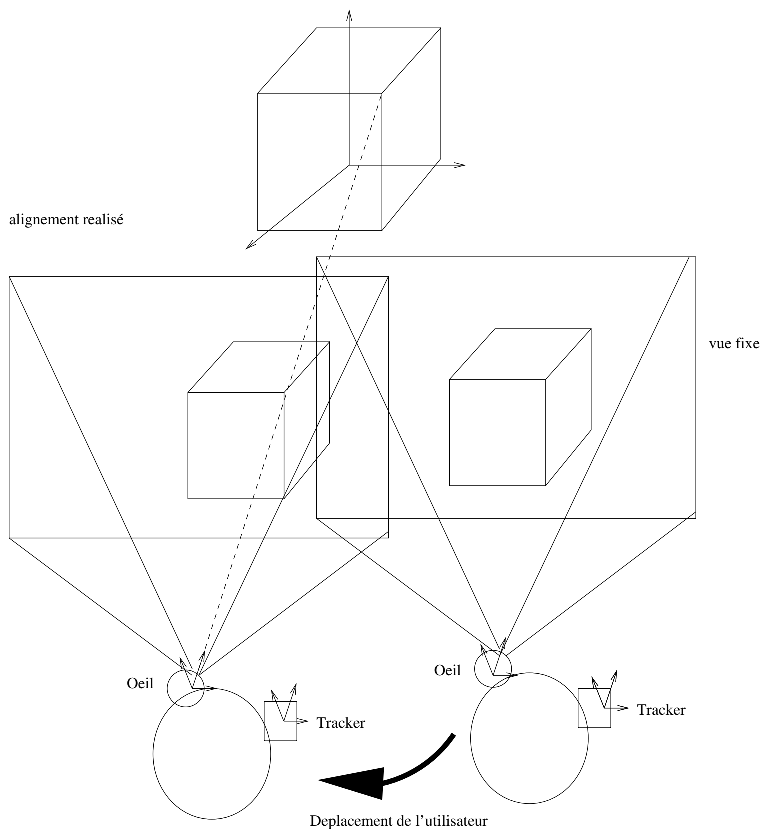

FIG. 6.2 – Principe de la calibrage

En cours d'alignement on a la relation :

$$T_{RecepteurTete} = T^{-1}_{EmetteurRecepteur} * T^{-1}_{MondeOeil} * T^{-1}_{TeteOeil}$$

Sur des vues alignés sur des axes la profondeur est difficilement réglable : on priviliegera donc des projections non orthogonales lors de la calibrage. Le système est rendu plus stable du fait qu'on utilise la stéréoscopie pour la calibrage : On fige les 2 vues de l'utilisateur donnant une unique solution pour le placement de cet objet dans l'espace.

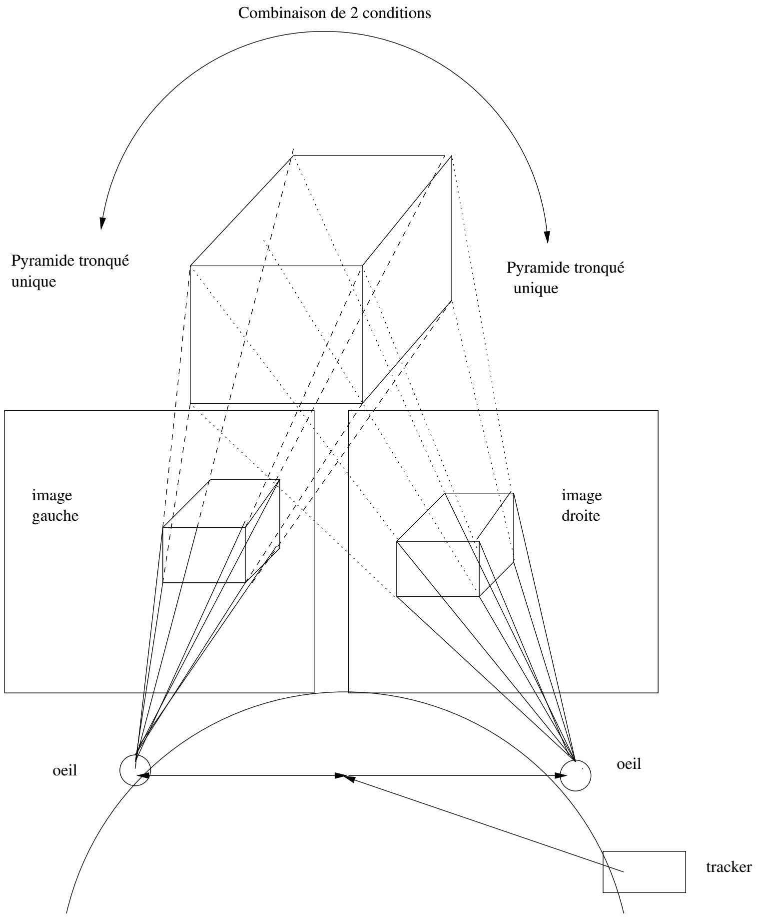

FIG. 6.3 – Système monoscopique et stéréoscopique

On note que la seule transformation variante est $T_{Emetteur Recepteur}$, et donc que l'erreur du FOB influera sur le résultat.

Lors de l'alignement correct on stoppe ce processus qui nous donne $T_{recepteurTete}$. Pour plus de fiabilité on fait une moyenne sur les dernières mesures évalues par le capteur, considérant le manque de précision de l'utilisateur. On évalue nos résultats en considérant un pion virtuel placé de 2cmx2cm place sur une case d'une grille : l'erreur de déplacement du pion virtuel lors de mouvement de la tête est évalue. On considère que le critère de validité est que le pion "reste" dans la case. Lors des mouvements de translation on observe une erreur de 2 à 3 mm. Pour des mouvements de rotation sur un axe l'erreur est de l'ordre de 5 mm. Pour des mouvements à au moins 5 DDL on à une erreur de 1 cm. L'erreur maximale ne dépasse pas 25 mm sur des positions complexes et éloigné du récepteur (le bruit du capteur influe beaucoup sur les résultats). Des mesures plus précises devront être réalisés.

6.1.2 Transformation capteur↔pointeur

On se base sur la méthode de Tuceryan. L'utilisateur pose son stylo à un point donnée et effectue différentes rotations autour de ce point laissant le point de contact invariant. En tout instant on a la relation :

$$p_w = T_m + R_m p_t$$

On peut alors modéliser le système de cet façon :

$$\begin{pmatrix} I & -R_{m1} \\ I & -R_{m2} \\ I & -R_{m3} \\ I & -R_{m4} \end{pmatrix} \begin{pmatrix} p_w \\ p_t \end{pmatrix} = \begin{pmatrix} T_{m1} \\ T_{m2} \\ T_{m3} \\ T_{m4} \end{pmatrix}$$

La solution retenue par Tuceryan consiste à réaliser de nombreuses mesures hors ligne et de sur contraindre le système, puis une minimisation au moindre carrés.

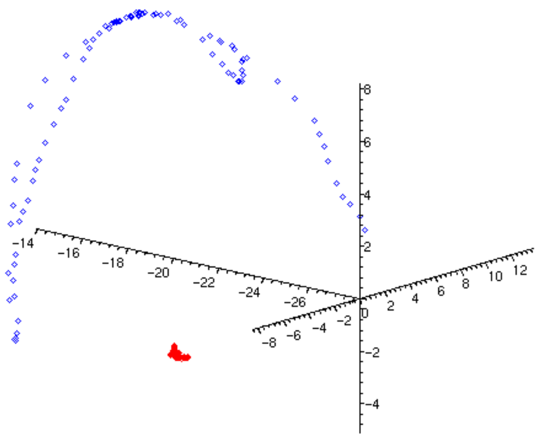

Fig. 6.4 – Système sur contraint

Notre approche est de minimiser ce nombre de mesures avec une relation entre les points simples et intuitives. On se base sur la saisie de 4 points de tel façon à ce qu'il décrivent dans l'espace une pyramide pointé vers le bas. Cet configuration permet d'obtenir les meilleurs résultats avec un faible nombre de points. On évalue cet méthode en la comparant à une méthode sur contrainte. On valide la méthode en regenerant à partir de la valeur trouvée (p_t), des calculs du point fixe dans le repere image (p_w) en déplaceant le stylo à de nombreuses orientations. L'erreur obtenu est comparé à la méthode surcontrainte. On voit que l'erreur est du même ordre (0.5 cm) donc notre méthode est valide (cet grande valeur s'explique par le bruit causé par le système).

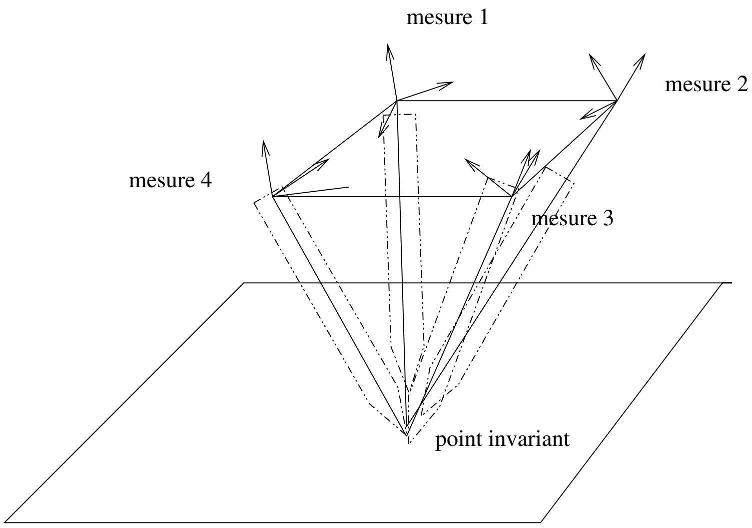

FIG. 6.5 – Schéma de la méthode

6.1.3 Transformation émetteur↔monde

On effectue la calibrage entre le repère du capteur et le repère du monde. On reprend les travaux de Tuceryan. On fait saisir trois point par l'utilisateur nous décrivant un repère (euclidien) dans le repère du capteur permettant d'en déduire le repère du monde. Les points choisis définissent le repère du monde.

Les transformations entre objet et repère du monde, et objet et capteurs sont étudiés plus en détail dans la partie ajout dynamique d'objet.

6.2 Erreur dynamique

6.2.1 Optimisation logiciel

Une première phase de diminution du temps de latence est l'optimisation logiciel. L'utilisation d'une architecture multiprocesseur est la solution retenu. On optimise de plus la sortie vidéo obtenu auparavant par un logiciel intermédiaire. Une dernière optimisation consiste en la lecture de la position de la tête de l'utilisateur dans la dernière phase de calcul avant le rendu. Ce paramètre indépendant des calculs précédents nous assure une minimisation de la latence pour le suivi de tête.

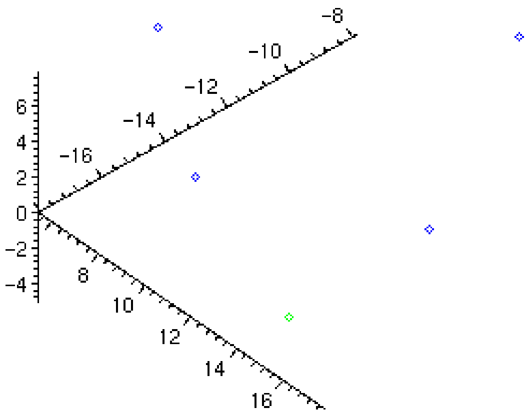

FIG. 6.6 – Mesure des 4 points (en haut de la figure) et calcul du point fixe (en bas de la figure)

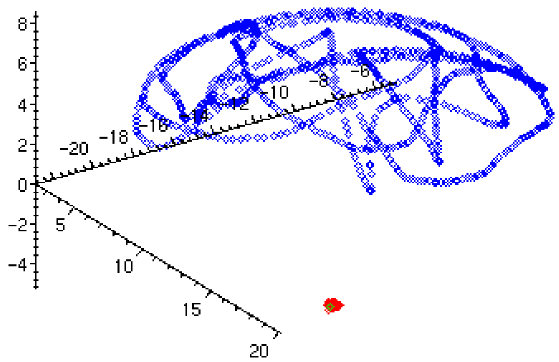

FIG. 6.7 – On génère des nouvelles mesures de la position du point fixe à partir du calibrage estimé. Ceci nous montre des résultats proches que ceux de la méthode sur contrainte, donc notre méthode est valide.

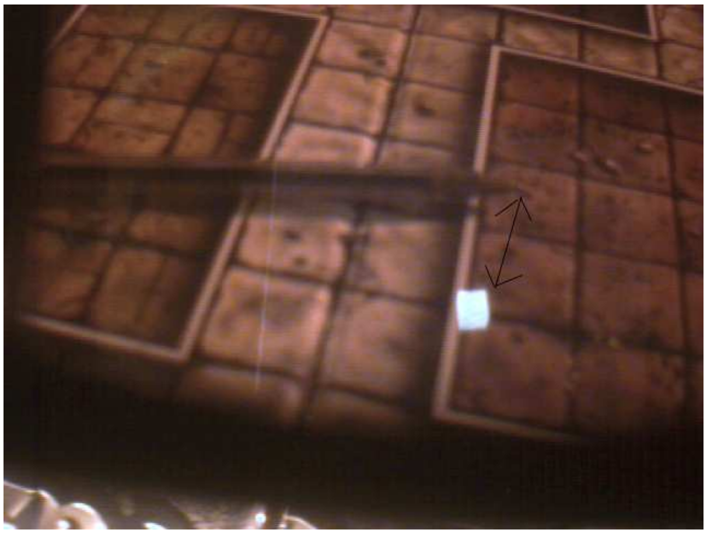

FIG. 6.8 – Désalignement entre le pointeur et le stylo. Le stylo est visible en haut de l'image et le cube virtuelle en bas de l'image devrait etre place à la pointe du stylo.

FIG. 6.9 – Alignement entre le pointeur et le stylo. On voit que le cube blanc est placé au bout du stylo ce qui correspond à un bonne alignement

6.2.2 Prédiction et filtrage

On utilise un filtre de Kalman pour réduire le bruit. Le filtre de Kalman nous sert de plus à réaliser une prédiction. Les différentes mesures sont réalisés à partir des données provenant du SM3D qui est beaucoup plus précis que Ceux du Flock of Birds. L'étude est réalisé hors-ligne après récupération et traitement des données. Un filtre de Kalman est associé à chaque composante de la transformation : on a 3 filtres pour la positions et 4 pour la rotation (sous forme de quaternion). Il est alors nécessaire de renormaliser le quaternion. Les résultats (courbre 6.10) montrent que le filtre de Kalman est une bonne solution (pour la latence et le bruit) mais le paramétrage est plutôt difficile.

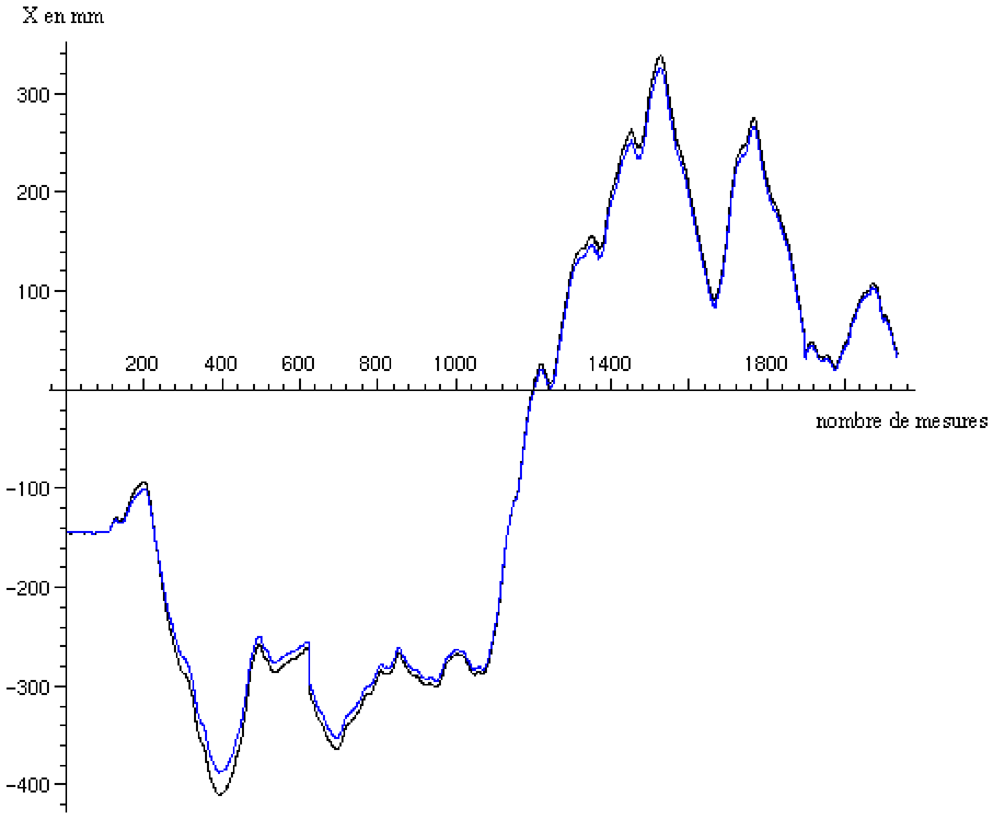

FIG. 6.10 – Prédiction et mesure de la composante x de position avec une valeur de prédiction de 400 ms et une erreur du modèle de mesure fixé à 0.001 mm. Cet courbe montre que lors d'une decroissance de x, le filtre décroit aussi en étant placé sous la courbe, et lorsque x croît le filtre croít en étant placé au dessus de la courbe. Ceci confirme l'efficacité du filtre pour réaliser une prédiction en plus du filtrage

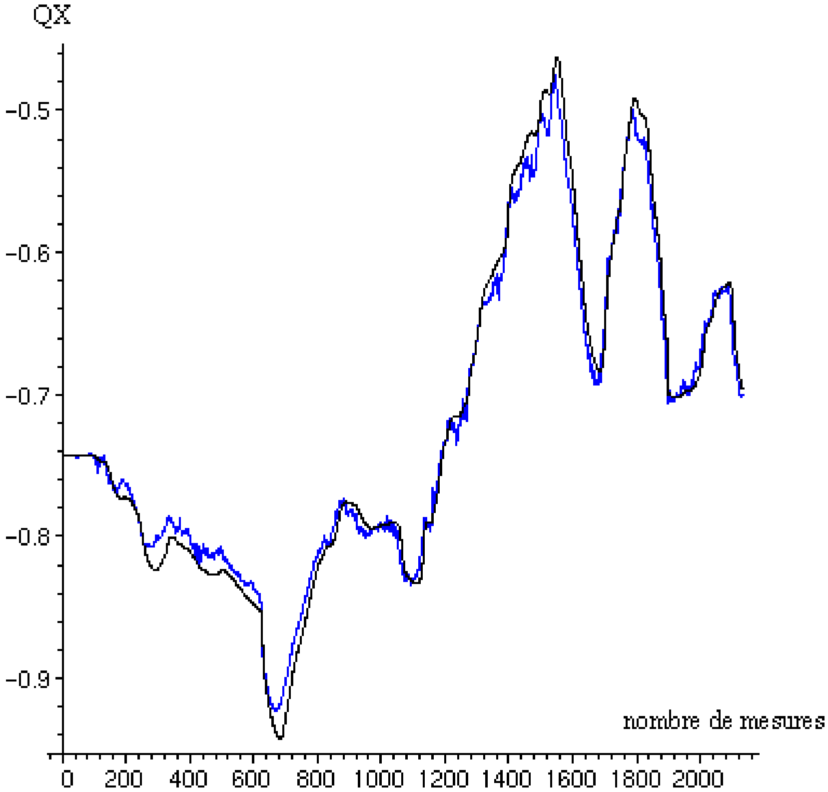

FIG. 6.11 – Prédiction et mesure de la composante x du quaternion avec une valeur de prédiction de 400 ms et une erreur du modèle de mesure fixé à 0.001. On voit ici que le filtrage sur la composante qx est trés efficace (la deuxième courbe est lisse comparé à la première) et que le filtre de Kalman permet une bonne prédiction.

Chapitre 7

Ajout dynamique d'objets

La calibrage d'un objet est le processus qui consiste à calculé la position et l'orientation d'objets réel de tel façon à ce que sa représentation virtuel puissent être placé à la position et à l'orientation exacte dans le monde virtuel. Il existe 2 cas pratique d'utilisation : On s'intéresse à l'ajout dynamique d'objets en cours d'application. Les manipulations doivent être simple et intuitive. ceci nous amène à parler du calibrage objet$\leftrightarrow$monde, et objet$\leftrightarrow$récepteur.

7.1 Ajout dynamique d'objet statique

On peut définir 2 cas principaux : ajout d'objets de géométrie connu et d'objets de géométrie inconnue (pour ce dernier cas on considère la topologie de l'objet connu). Dans les 2 cas la récupération de la géométrie est combiné avec la position de l'objet. On introduira 2 méthodes.

7.1.1 Méthode par ajout de points

Cet méthode est utilisée pour des objets de géométries inconnues, mais de topologie connue. On saisie des points spécifiques sur l'objet, le nombre de points réduit au minimum pour obtenir la position et l'orientation de l'objet ainsi que sa géométrie. On l'applique à des plans et des cubes. On affiche la construction réalisé en temps réel : après l'ajout du premier point, le deuxième est un point libre relie par une droite.

Application à des plans

Trois points (P_1, P_2, P_3) sont nécessaires pour obtenir toutes les contraintes. La méthode décrit sur 7.1 est appliqué pour saisir les points.

On définit s_x, s_y les dimensions de l'objet, et T_1, R_1 sa position et orientation. On a alors :

$$s_x = \|P_2 - P_1\|$$

$$s_y = \|P_3 - P_2\|$$

$$T_1 = \frac{P_1 + P_2 + P_3}{3}$$

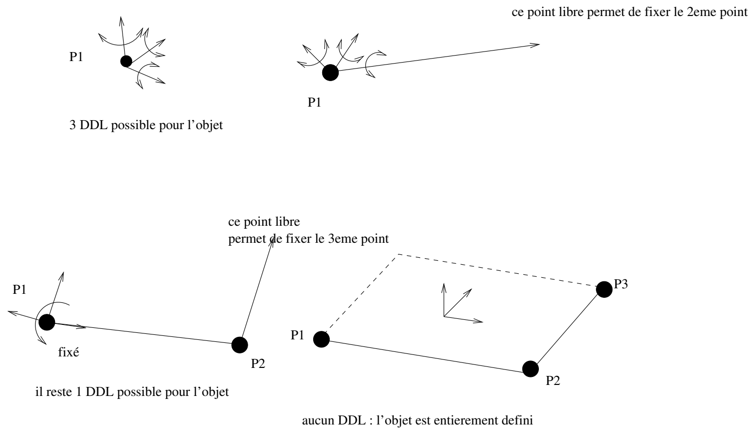

FIG. 7.1 – Déroulement d'ajout de plan par 3 points successifs

$$R_1 = \begin{pmatrix} P_2 - P_1 & P_3 - P_2 & (P_2 - P_1) \wedge (P_3 - P_2) \end{pmatrix}$$

Application à des cubes

Quatre points (P_1, P_2, P_3, P_4) sont nécessaires pour obtenir toutes les contraintes. On reprend le principe précédent avec une contrainte en plus (s_z)

On modifie les équations précédentes par :

$$s_z = \|P_4 - P_3\|$$
$$T_1 = \frac{P_1 + P_2 + P_3 + P_4}{4}$$
$$R_1 = \begin{pmatrix} P_2 - P_1 & P_3 - P_2 & P_4 - P_3 \end{pmatrix}$$

La méthode est fiable, intuitive mais est limité par la précision du capteur. Une idée complémentaire non implémenté et de vérifier les propriétés géométriques de ces objets tel que la perpendicularité dans les angles. On pourra étendre la méthode à des formes plus complexes (cylindre, octaèdre) conservant une définition mathématique précise.

7.1.2 Méthode par contraintes axiales

Cet méthode s'applique aussi bien au topologie connue, qu'aux objets ayant une représentation virtuel référencée. L'idée est d'afficher la représentation virtuel de l'objet est de la superposer a l'objet réel à partir de mise en correspondance de certains points. On ne souhaite pas se limiter à une géométrie définie pour un objet, on utilise donc une approche par alignement à partir de la boîte englobante de l'objet. On ne s'intéresse qu'a 4 degrés de liberté (position et "heading") la plupart des objets étant posé sur l'espace de travail. On retient 2 approches :

1. Une arête de la boîte englobante est confondus avec une arête du cube. La méthode consiste à placer l'arête commune avec l'objet réel puis une rotation permet l'alignement définitif et nous donne la transformation de l'objet (7.5).

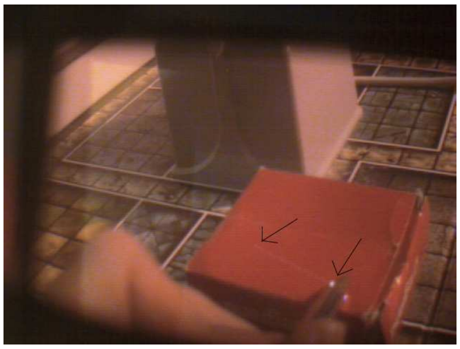

Fig. 7.2 – Ajout d'un plan virtuel par la méthode d'ajout de points (1ère phase). La flèche de gauche nous montre le premier point tracé. La flèche de droite définit le deuxième point libre. On constate le faible alignement du pointeur virtuel et du stylo provenant de la difficulté de calibrages des étapes précedentes à cause de l'utilisation d'une caméra

Fig. 7.3 – Ajout d'un plan virtuel par la méthode d'ajout de points (2ème phase). La flèche de gauche nous montre le troisième point libre. La flèche de gauche nous montre le plan tracé. Un cube virtuel a été positionné dessus visible sur la flèche de droite.

Fig. 7.4 – Plan virtuel ajouté sur le cube réel

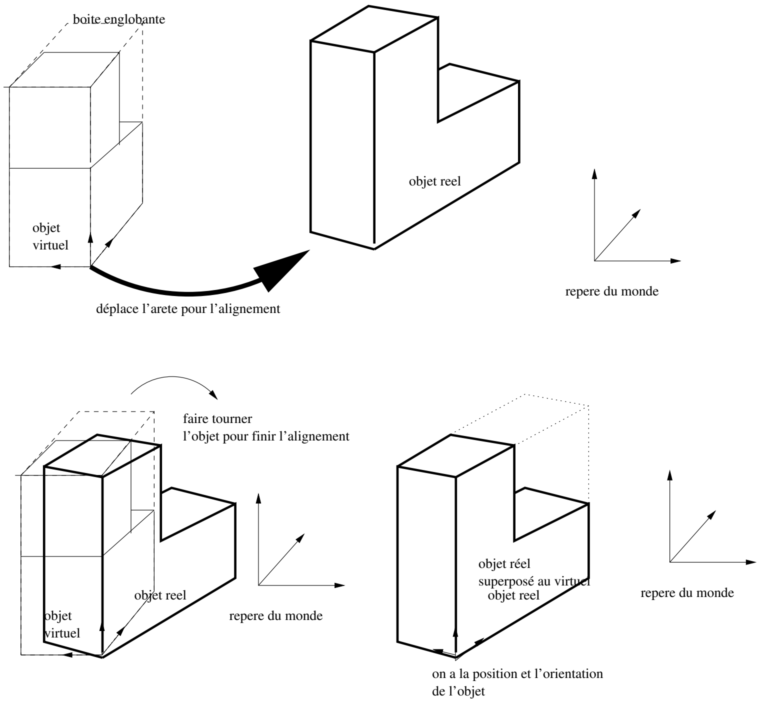

FIG. 7.5 – Méthode d'ajout par arrête commune

2. L'objet est de géométrie plus complexe. Le principe consiste à aligner les 2 centres géométriques de l'objet (la géometrie est la même donc le centre est connu). Le déplacement se fait par déplacement d'un point de la boite englobante. Puis un mouvement de translation sur un axe quelconque nous permet de regler l'orientation sur l'axe vertical. (7.6).

La répétition de la méthode sur quelques essais, nous donne une bonne approximation de la transformation de l'objet réel dans le monde virtuel. Cet méthode est beaucoup plus générique que la méthode de [32] est permet un ajout dynamique d'objet.

7.2 Ajout dynamique d'objet déplaçable

L'intégration d'objet réel dans l'espace virtuel a été réalisé précédemment. On s'intéresse maintenant au déplacement de ces objets dans l'espace. Pour cela l'objet doit être suivi pour mettre à jour sa position dans notre modèle. De plus une calibrage entre le capteur et l'objet doit être réalisé. Une technique consiste a utilise des marqueurs sur l'objet et un système de vision permettant de suivre le déplacement de l'objet. Malheureusement des occlusions partiels posent des problèmes de suivi de ces objets déplacement par capteur.

On connaît la position de l'objet dans le repère du monde. On connaît la position du capteur a tous instants. Une fois le capteur attaché a l'objet on a la relation :

$$T_{MondeObjet} = T_{MondeEmetteur} * T_{EmetteurCapteur} * T_{CapteurObjet}$$

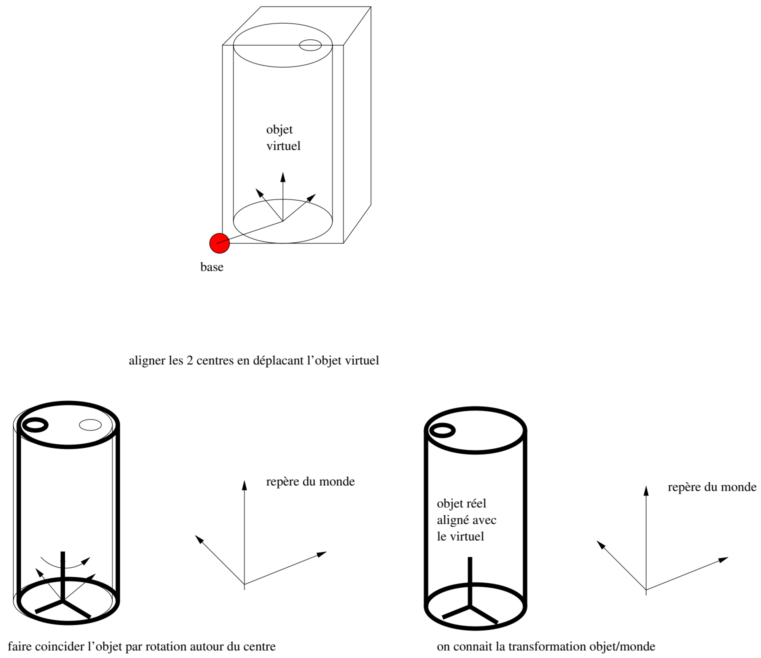

FIG. 7.6 – Méthode d'ajout par centrage.

FIG. 7.7 – Déroulement de l'ajout d'un pion virtuel avec la prèmiere méthode. Sur l'image de gauche on voit le pion réel à ajouter. Sur l'image du centre on voit le premier positionnement. Sur l'image de droite on voit le réglage final par rotation.

On initialise donc notre système par le placement du capteur sur l'objet et par calculer :

$$T_{CapteurObjet} = T_{EmetteurCapteur}{}^{-}1 * T_{MondeEmetteur}{}^{-}1 * T_{MondeObjet}$$

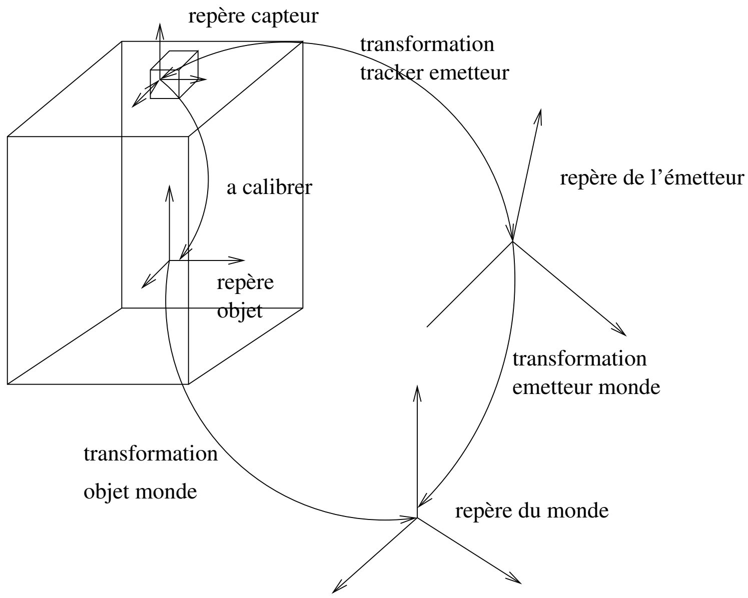

FIG. 7.8 – Relation entre les différents repère pour la calibration d'objet dynamique

Deux approches sont envisagés pour confirmer que le capteur est positionné sur l'objet :

1. Par validation de la part de l'utilisateur. L'utilisateur confirme le positionnement et valide le contact.
2. Automatique. On détecte l'intersection entre l'objet et la géométrie du capteur. Cet solution non mis en oeuvre nécessite la connaissance précise de la géométrie du capteur, et le positionnement du centre dans cet géométrie.

Le système est très dépendant de l'étape précédente et le bruit se répercute sur ce dernier processus.

Chapitre 8

Gestion multi-utilisateurs

On introduit dans cette partie un certain nombre de propositions personnelles pour la gestion du multi-utilisateurs. Cette contribution reste à l'état d'un ensemble d'idée non encore complétement définis, qui représente une parti moins abouti du reste du DEA. On introduit la notion de **maître de contrôle** de la session. (Pour une application de jeu, on le définira par maître de jeu). Il est équipé d'une interface utilisateur standard (souris, clavier, écran) et peut modifier et interagir complètement avec la scène. Ce choix se justifie par le fait qu'une session nécessite toujours un contrôle humain certains paramètres ne pouvant pas être complètement contrôles par l'application elle même.

On propose de reprendre et d'adapterles principes les principes de gestion utilisateur d'UNIX, où chaque utilisateur est défini par une entités **utilisateur** et **groupe** (les groupes regroupent des personnes en affinités tel que des architectes dans une réunion ou des magiciens dans un jeu). Ces groupes permettent d'obtenir une vision commune de la scène ou des droits communs sur des objets. **Les droits** définissent les possibilités d'actions sur chaque instance d'un objet (ces actions étant des actions sensorielles tel que voir, entendre, ou des actions de manipulations tel que prendre, déplacer). Un objet peut appartenir à la communauté, à un groupe, ou à un utilisateur, et les droits respectent le formalisme UNIX (les droits sont donc implicitement lié à chaque objet). Les différences fondamentales par rapport au système UNIX sont :

- un objet peut avoir plusieurs représentations pour différentes personnes ;
- les objets peuvent être étendus par des nouvelles actions et des nouveaux droits ;

On peut considérer l'exemple suivant.

<table>
<tr><td colspan="4">OBJET : Voiture</td></tr>
<tr><td colspan="4">représentation 1 : voiture avec tous les détails</td></tr>
<tr><td colspan="4">représentation 2 : voiture avec moins de détail</td></tr>
<tr><td colspan="4">représentation 3 : voiture avec un maillage simplifié</td></tr>
<tr><td colspan="4">propriétaire : Schumarer groupe : Ferrari</td></tr>
<tr><td>droits</td><td>utilisateur</td><td>groupe</td><td>autres</td></tr>
<tr><td>voir</td><td>OUI (rep 1)</td><td>OUI (rep 2)</td><td>OUI (rep 3)</td></tr>
<tr><td>déplacer</td><td>OUI</td><td>NON</td><td>NON</td></tr>
<tr><td>modifier</td><td>OUI</td><td>OUI</td><td>NON</td></tr>
</table>

Une interface graphique permettra au maître de jeu de contrôler et de modifier ces droits à tous instants. Contrairement a Studierstube on n'utilise pas un système de droits contraints à des régions de manipulations, ce système étant trop restrictif (2 joueurs voyant le même pion sur la table peuvent le voir avec une représentation différentes).

On s'intéresse maintenant à la gestion et à la modification du droits de visibilité, par l'étude de la gestion du graphe de scène associé à chaque utilisateur.

8.1 Gestion du graphe de scène

On associe à chaque utilisateur un **graphe de scène** qui représente la vision du monde de l'utilisateur basé sur ses droits et sa propre configuration. Le scène de graphe se découpe en 3 couches : éléments utilisateurs, éléments de groupe, éléments commun. Chacune de ces couches se différencie en éléments statiques ou dynamiques, dépendant des possibilités de changements de droits d'un objet par le maître de contrôle . La construction de l'arbre se fait par partage des objets. On définit un objet dans le scène graphe par un noeud "switch" qui représente les différentes représentations d'un objets. Lors du rendu on choisit la représentation approprié d'après les droits de l'utilisateur (effectué lors de la phase de culling sous Performer). On evite les doublons dans le scène graphe en considérant la priorité de la communauté, groupe et utilisateur pour l'ajout de l'objet.

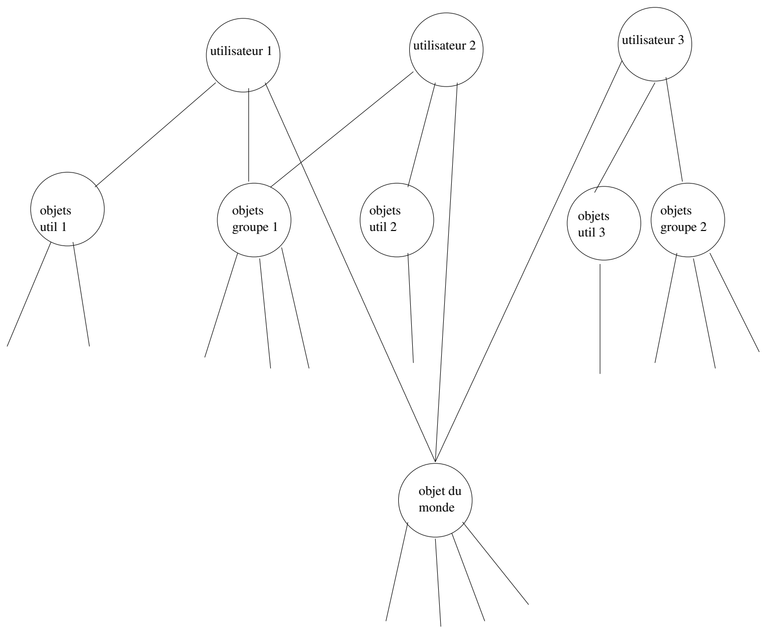

FIG. 8.1 – Graphe de scène de plusieurs joueurs

On considère les combinaisons à partir des éléments suivants :

1. géométrie de l'objet (simplifié, normal, complexe) ;
2. paramètre d'illumination de l'objet ;
3. matériau de l'objet (textures).

Chaque combinaison créer un nouveau noeud qui est une représentation unique de l'objet. On définit ainsi des **multireprésentations** qui est une complémentarité entre une géométrie et un aspect.

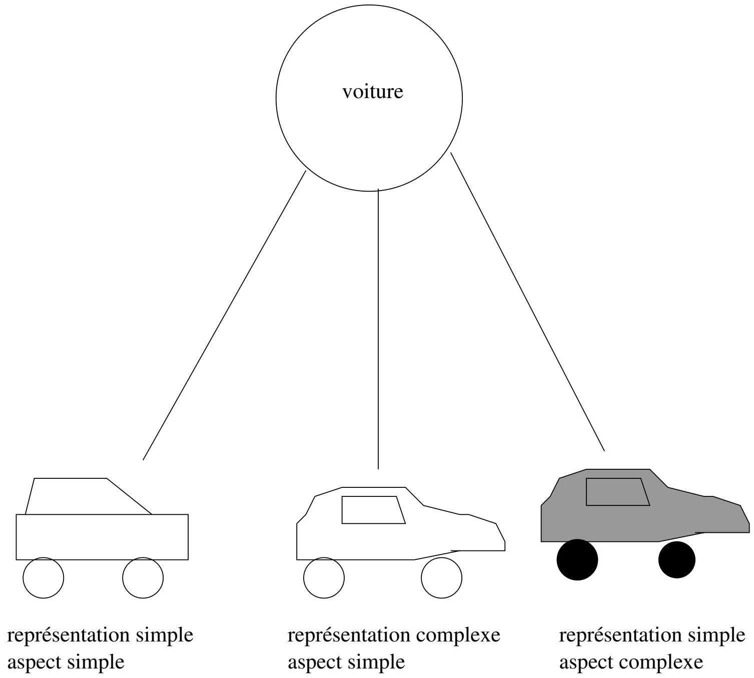

FIG. 8.2 – Multireprésentation d'un objet

Comme déjà exprimer la différence entre un objet réel et virtuel se base sur le fait qu'un objet réel à une unique géométrie et un unique aspect. Une multireprésentation est alors défini par le fait d'augmenter cet objet créant une nouvelle représentation (on a donc de nouvelles combinaisons à partir des données réels et virtuels). On introduit aussi la notion de **guide réel** qui consiste à ajouter une représentation virtuel complexe à un objet réel simple suivi n'ayant aucune corrélation visuel. Ceci permet de déplacer de façon intuitive un objet virtuel par un contact réel (par exemple déplacé une représentation d'une maison virtuel lors d'une réunion municipale sur la construction d'un nouveau complexe urbain).

Chapitre 9

Applications aux jeux vidéos

La gestion du multi-utilisateurs, l'ajout d'objets réels avec des objets virtuels, et un temps de réponse rapides constitue des points importants utilisables pour un jeu. Un scénario peut par exemple être un jeu de rôles où on a des magiciens et des guerrier. Les uns sont représentés par des pions réels, les autres par des pions virtuels. Les premiers peuvent voir à travers les murs tandis que les deuxième voit les pièges. Un déplacement sur un déclencheur provoquera par exemple l'envoie d'une boule de feu dans la direction du joueur, le joueur devra interagir vite pour éviter le contact. On s'intéresse maintenant au spécificité propre de la réalité augmentée pour le jeu.

9.1 Déplacement de pions

la difficulté de manipulation de pions 3D nous contraint à des déplacements sur des surfaces garantissant une facilité de déplacement. On a retenu principalement 2 méthodes respectant cet idée.

1. La première méthode est une métaphore du monde réel (9.2). Déplacer un pion nécessite de saisir ce pion de le soulever et de le déplacer dans l'espace et de le reposer dans l'espace de jeu. On reprend le même principe de cet façon : on détecte la sélection du pion est une information visuel correspondante est affiche, on s'assure que le contact s'effectue sur support réel. Pour permettre l'action de déplacement l'utilisateur doit soulever le pion, il effectue un mouvement de translation vertical. La fin de l'action et la pose de l'objet est induit par le contact réalisé avec le stylo sur la table. Cet méthode est utilisable sur des environnements basé sur des changements de hauteur.
2. La deuxième méthode reprend le principe de "faire glisser" un pion pour le déplacer sans perte de contact (9.3). On utilise un système de sélection de la direction puis de validation de la trajectoire. On se fixe pour le moment sur des trajectoires rectilignes. Le principe est le suivant : après sélection du pion par intersection avec le stylo, une flèche directionnelle est défini entre le centre de l'objet et le bout du stylo. Le déplacement du pointeur indique la trajectoire à prendre par le pion, le déplacement étant planaire et conservant le contact entre le stylo et l'espace de jeu. On valide le déplacement par un mouvement de translation en

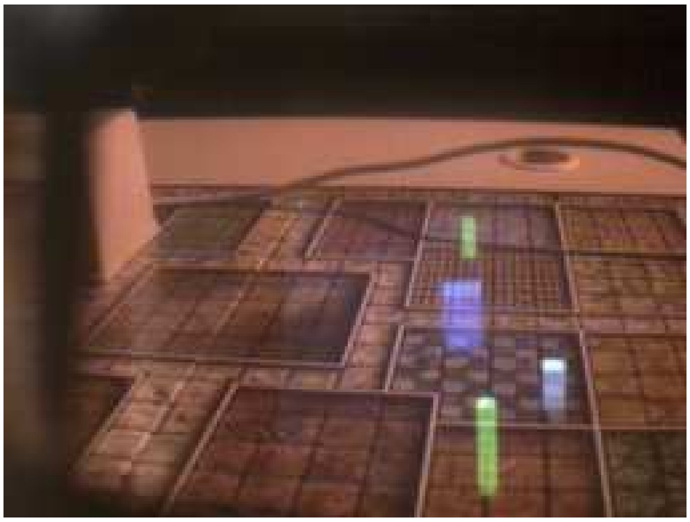

FIG. 9.1 – Plateau de jeu positionné après les calibrages

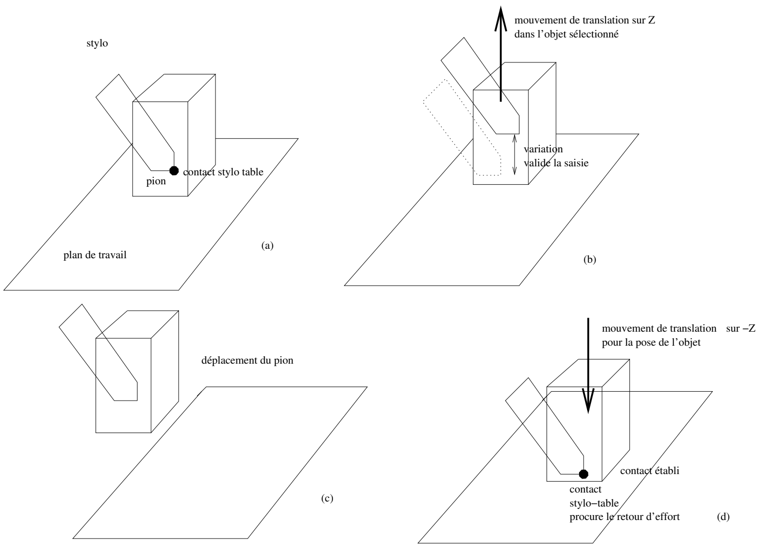

FIG. 9.2 – Saisie et déplacement de pion

hauteur. Une animation pourra alors être généré pour le déplacement tel qu'un sautillement du pion du personnage.

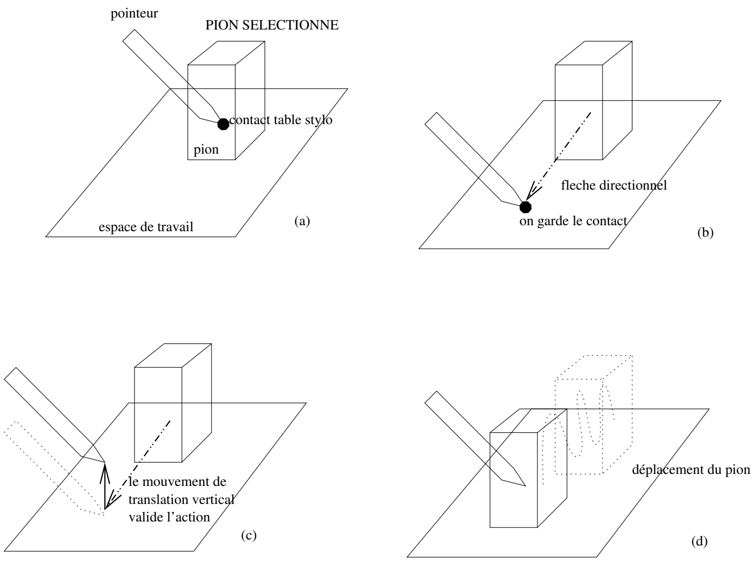

FIG. 9.3 – Saisie et déplacement de pion

Cette méthode pourra être étendu pour des déplacements plus complexe et pour la gestion du déplacement sur la trajectoire : Le joueur trace une trajectoire pour son pion à l'aide du stylo virtuel. Puis il déplace son pion sur cette trajectoire il peut avancer ou reculer le pion, le pion suivant la trajectoire prédefini.

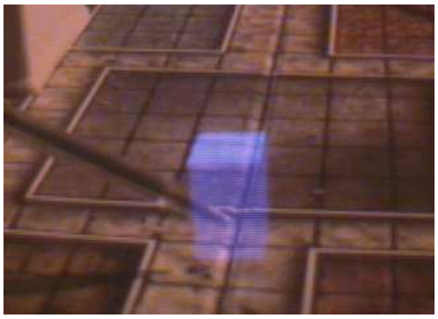

FIG. 9.4 – Pion virtuel positionné sur le plateau de jeu

9.2 Reconaissance de gestes

On s'inspire des travaux de Thad Starner qui décode les mouvements naturels comme des commandes de l'utilisateur (on citera aussi [?] avec leur "tableau magique"). Sa méthode basé

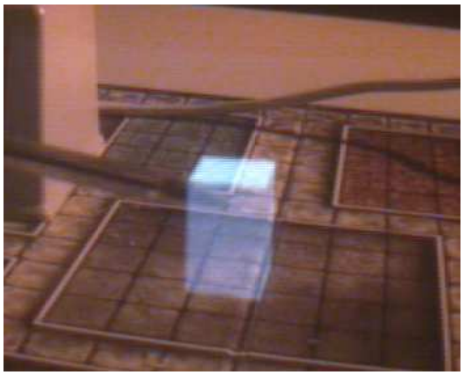

FIG. 9.5 – Sélection d'un pion virtuel avec le stylo

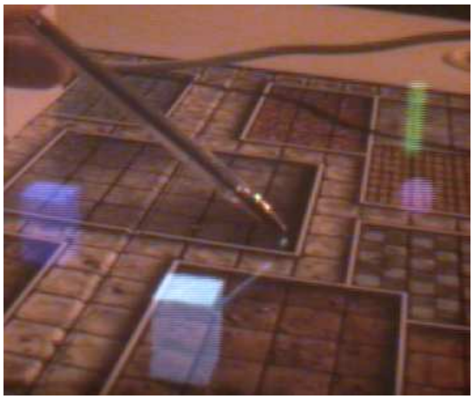

FIG. 9.6 – Déplacement d'un pion par la méthode de glisser de pion. On voit ici la trajectoire possible pour ce pion.

FIG. 9.7 – Autre point de vue de la méthode de glisser de pion.

sur une approche par vision qui nécessite un espace important pour l'utilisateur. Dans notre cas on utilisera des gestes simples d'interaction (ce principe se retrouve dans certains logiciels commerciales de reconnaissances de geste pour contrôler son bureau de travail). On considère des gestes simples comme un déplacement en ligne dans une direction du pointeur. La reconnaissance ce fait dans un temps imparti par un déclenchement précis : On considère la première et la dernière mesure car l'erreur de déplacement du joueur est négligeable du fait qu'il effectue un déplacement linéaire (une régression linéaire peut être implémentée pour une reconnaissance plus précise). On ne défini que 8 gestes de base ce qui nous permet d'obtenir facilement et rapidement la bonne actions (on compare le vecteur de déplacement au vecteur directeur de ces gestes). On limite ce nombre de gestes reservés à des actions à effectué rapidement (comme par exemple déclencher un sort spécifique sur un monstre). Pour faire la difference avec le déplacement d'un pion on gére un certains nombre d'actions correspondant à la possibilité de réaliser le geste. La méthode ici limité à un déplacement linéaire pourra être étendu par des trajectoires plus complexe, tel qu'une succession de segments lineaires (un signe du genre "z" définira une certaine action). On utilisera une notation avec plusieurs lettres pour définir les segments, ce qui nous permettra d'avoir une succession de lettres comme base de déplacement. Puis une simplification sera réaliser pour obtenir l'action finale : par exemple BBEBFFFBBB deviendra BBBBFFFBBB aprés filtrage, et finalenement BFB.

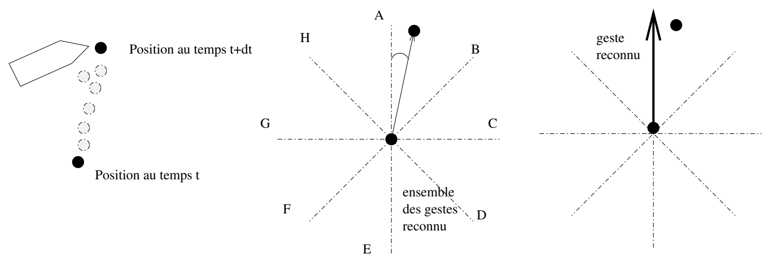

FIG. 9.8 – Reconnaissance d'un geste

9.3 Mise en place de la scène

La gestion d'un jeu par le maître de jeu nécessite la mise en place de la scène virtuel. Ils consistent en des utilitaires de création de scène et de monde et de scénario (ensemble d'actions programés). Une implémentation d'un logiciel simple de création de monde a été fait durant le stage. On réalise aussi un petit logiciel de création de personnages. Le joueur pourra contrôler l'évolution de son personnage via l'Internet (une base de donnée sécurisé distante conservera les informations sur l'utilisateur et l'évolution de son personnage). La création du graphe de scène à partir de la carte est faite implicitement suivant le type d'objet de la carte.

Fig. 9.9 – Interface de création de personnage

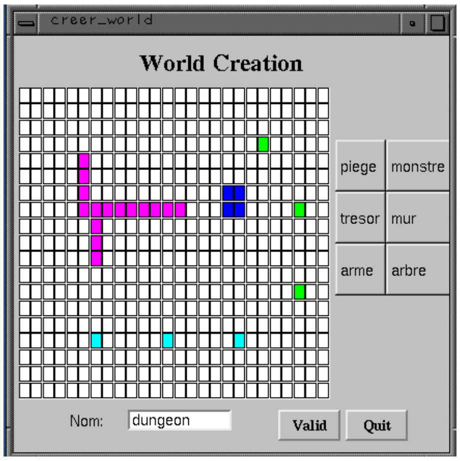

Fig. 9.10 – Interface de création de monde

9.4 Gestion du temps

Le déroulement d'un jeu peut être décomposables en 2 catégories : les jeux temps réel où le joueur interagit avec la scène de façon continu, et les jeux par tour, où le joueur dispose d'un certain moment et une certaine durée pour agir. On reprend ce principe ici, on ajoutant la possibilité d'alternance entre ces 2 types. Le passage en temps-réel produit alors une vue immersive de joueur réel dans sa représentation virtuel. Il est donc nécessaire d'effectuer une transition entre ces 2 phases : on choisit une interpolation linéaire. On effectue de plus un changement d'échelle entre les 2 points de vue, car on est maintenant dans un contexte de réalité virtuelle. Cet partie n'est actuellement pas intégrer dans le programme principal mais dans une petite démo séparée. Pour la gestion des animations, on gère une liste des différents objets ayant un comportement propre, qui nous permet de mettre a jour ces objets. Pour des animations dépendant des personnages, on génère de nouveaux objets ajoutés à cet liste.

Chapitre 10

Conclusion

Un environnement de réalité augmentée permettant de gérer l'interaction des différents utilisateurs a été réalisé. Un architecture complète a été développer basé sur la notion d'entités réelles et d'objets virtuels avec des variables d'états, des comportements réactifs et un comportement propre.

Une procédure de calibrage de capteur à tête a été établi reposant sur des manipulations utilisateurs. Cet partie non abordé dans le rapport devrait faire l'objet d'un Brevet à l'INPI. Une procédure de calibrage rapide du stylo a été formulé répondant à une méthode facile et intuitive. Un nombre de mesures minimales basé sur des points de saisie décrivant une pyramide permet un calibrage rapide. Différentes méthodes ont été développé pour introduire des objets réels de façon dynamique durant une simulation. Les méthodes reposent sur des manipulations utilisateurs, et la visualisation en temps réel de l'évolution de l'ajout des objets.

La gestion de plusieurs utilisateurs à été étudiée, ainsi que le principe de multireprésentations. Les droits sont spécifier à chaque objet et s'inspire du système UNIX. Une vue personnel à chaque utilisateur repose sur un graphe de scène personnel, ainsi sur une notion de graphe de groupe.

Différentes formes d'interaction 3D ont été introduits points ont été introduits pour répondre à l'utilisation de la réalité augmentée appliqué au jeu. On peut citer 2 méthodes de déplacement de pions répondant à des contextes différents. L'utilisation de "reconaissance de geste" et un système de changement de point de vue a été étudié.

10.1 Perspectives

Tout d'abord une étude plus précises des paramètres internes et une méthode d'évaluation de la distance inter pupillaire de l'utilisateurs devront être réalisées (actuellement ces paramètres sont fournis directement à l'aide de documentation et de mesures non robustes).

L'achat de matériel complémentaires permettra de véritablement tester l'environnement et de définir les nouveaux besoins de gestions multi-utilisateurs.

Un étude plus précise des besoins des joueurs devra être formulée, pour définir l'environnement adapté à l'utilisateur. On constate déjà que certains jeux tel que les jeux immersifs (communément nommée les "doom-like") ne correspondent pas à ce type d'environnement, mais est plus adapté à un environnement de type réalité virtuel.

Un autre point non étudié est le déplacement dans un espace virtuel infini, sur un support réel fini. Ce problème pourrait répondre au problème d'utilisation de cartes réels par les navigateurs, les militaires.

Un réalisme accru pourrait être apporté à ce système, par une étude de la luminosité ambiante pour adapté le modèle d'illumination utilisé pour le rendu virtuel (basé sur une analyse d'image). On notera que l'absence d'ombre au système donne une impression d'objets volants au dessus des supports réels. Un ajout dynamique d'objets pourrait être réalisé par un système stéréoscopiques, permettant de saisir les informations relatives à l'aspect des objets.

Enfin, l'introduction de la réalité augmentée dans les applications collaboratives est un sujet vaste, présentant de nombreux aspects évoqué au long de ce travail, qui nécessite une étude d'une ampleur d'une thèse.

Bibliographie

[1] Z. Szalavari, E. Eckstein, , and M. Gervautz. Collaborative gaming in augmented reality. In *Proc of VRST'98(Taipei, Taiwan 1998)*, pages 195–204, 1998.

[2] Toshikazu Ohshima, Kiyohide Satoh, Hiroyuke Yamamoto, and Hideyuki Tamura. AR 2 hockey. In Scott Grisson, Janet McAndless, Omar Ahmad, Christopher Stapleton, Adele Newton, Celia Pearce, Ryan Ulyate, and Rick Parent, editors, *Conference abstracts and applications : SIGGRAPH 98, July 14–21, 1998, Orlando, FL*, Computer Graphics, pages 110–110, New York, NY 10036, USA, 1998. ACM Press.

[3] Thad Starner, Bastian Leibe, Brad Singletary, and Jarell Pair. Mind-warping : Towards creating a compelling collaborative augmented reality game.

[4] D Drascic and P. Milgram. Perceptual issues in augmented reality. *SPIE :Stereoscopic Displays and Virtual Reality*, 2653(7) :123–134, 1996.

[5] Wendy E.Mackay. Augmented reality : A new paradigm for interacting with computer. *La recherche*, (284), March 1996.

[6] Russell M. Taylor, II, Warren Robinett, Vernon L. Chi, Frederick P. Brooks, Jr., William V. Wright, R. Stanley Williams, and Eric J. Snyder. The Nanomanipulator : A virtual reality interface for a scanning tunnelling microscope. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 127–134, August 1993.

[7] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging virtual objects with the real world : Seeing ultrasound imagery within the patient. In *Proc. ACM SIGGRAPH '92, Computer Graphics*, pages 203–210, July 1992. Published as Proc. ACM SIGGRAPH '92, Computer Graphics, volume 26, number 2.

[8] Andrei State, Mark A. Livingston, Gentaro Hirota, William F. Garrett, Mary C. Whitton, and Henry Fuchs. Technologies for augmented-reality systems : Realizing ultrasound-guided needle biopsies. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 439–446. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[9] S. Feiner, B. Macintyre, and D. Seligmann. Knowlege-based augmented reality. *Communications of the ACM*, 36(7) :53–61, 1993.

[10] Dave Sims. New realities in aircraft design and manufacture. *IEEE Computer Graphics and Applications*, 14 :91, 1994.

[11] Mihran Tuceryan, Douglas S. Greer, Ross T. Whitaker, David E. Breen, Chris Crampton, Eric Rose, and Klaus H. Ahlers. Calibration Requirements and Procedures for a Monitor-

Based Augmented Reality System. *IEEE Transactions on Visualization and Computer Graphics*, 1(3) :255–273, September 1995.

[12] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world : 2D windows for 3D augmented reality. In *ACM UIST'93 Symp. on User Interface Software & Technology*, pages 145–155. 93.

[13] Anthony Webster, Steven Feiner, Blair MacIntyre, William Massie, and Theodore Krueger. Augmented reality in architectural construction, inspection, and renovation.

[14] David Drascic, Julius J. Grodski, Paul Milgram, Ken Ruffo, Peter Wong, and Shumin Zhai. ARGOS : A display system for augmenting reality. In *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, Formal Video Programme : Novel Technologies, page 521, 1993.

[15] Mark Ward, Ronald Azuma, Robert Bennett, Stefan Gottschalk, and Henry Fuchs. A demonstrated optical tracker with scalable work area for head-mounted display systems. In Marc Levoy and Edwin E. Catmull, editors, *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pages 43–52, Cambridge, MA, March–April 1992. ACM Press.

[16] Andrei State, Gentaro Hirota, David T. Chen, Bill Garrett, and Mark Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 429–438. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[17] Dennis C.Neale. Head mounted displays : Product review and related design considerations.

[18] Ronald T. Azuma. A survey of augmented reality. *Presence : Teleoperators and Virtual Environnements*, 6(4) :355–385, August 1997.

[19] Ronald Azuma and Gary Bishop. Improving static and dynamic registration in an optical see-Through HMD. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 197–204. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[20] Warren Robinett and Richard Holloway. The visual display transformation for virtual reality. Technical Report TR94-031, Department of Computer Science, University of North Carolina - Chapel Hill, May 10 1994. Wed, 26 Jun 1996 18 :14 :29 GMT.

[21] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics — Principles and Practice (2nd Ed.)*. Addison-Wesley, 1990.

[22] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 45–56, New York City, NY, June 1997. Eurographics, Springer Wien. ISBN 3-211-83001-4.

[23] Celine Loscos, Marie-Claude Frasson, George Drettakis, Bruce Walter, Xavier Granier, and Pierre Poulain. Interactive virtual relighting and remodeling of real scenes. In Dani Lischinski and Greg Ward Larson, editors, *Rendering Techniques '99*, Eurographics, pages 329–340. Springer-Verlag Wien New York, 1999.

[24] Sadahiko Mihashi Yasuhiro Mukaiwaga and Takeshi Shakunaga. Photometric image-based rendering for virtual lighting image synthesis. In *Proceedings of 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99) (San Fransisco, California, October, 20–21, 1999*, pages 115–124. Rockwell Science CEnter, IEEE Computer Society, October 1999.

[25] D. Schmalstieg, A. L. Fuhrmann, M. Gervautz, and Zs. Szalavári. 'studierstube' - an environment for collaboration in augmented reality'. 1996.

[26] Anton Fuhrmann, Helwig Löffelmann, Dieter Schmalstieg, and Michael Gervautz. Collaborative visualization in augmented reality. *IEEE Computer Graphics & Applications*, 18(4) :54–59, 1998.

[27] Ronald Azuma and Gary Bishop. A Frequency-Domain analysis of head-motion prediction. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 401–408. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[28] T. Oishi and S. Tachi. Methods to calibrate projection transformation parameters for see-through head-mounted displays. *Presence*, 5(1) :122–135, 1995.

[29] W. Purgathofer A. Fuhrmann, D. Schmalstieg. Fast calibration for augmented reality. In *Proceedings of ACM Virtual Reality Software & Technology '99 (VRST'99)*, London, December 20–22 1999.

[30] Mike Bajura. Camera calibration for video see-through head-mounted display. Technical Report TR93-048, Department of Computer Science, University of North Carolina - Chapel Hill, July 7 1993. Wed, 26 Jun 1996 18 :22 :56 GMT.

[31] Kiriakos N. Kutulakos and James R. Vallino. Calibration-Free Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 4(1) :1–20, January 1998.

[32] Ross T. Whitaker, Chris Crampton, David E. Breen, Mihran Tuceryan, and Eric Rose. Object calibration for augmented reality. *Computer Graphics Forum*, 14(3) :15–28, August 1995. Proceedings of Eurographics '95. ISSN 1067-7055.

[33] Marc Olano, Jon Cohen, Mark Mine, and Gary Bishop. Combatting rendering latency. In Pat Hanrahan and Jim Winget, editors, *1995 Symposium on Interactive 3D Graphics*, pages 19–24. ACM SIGGRAPH, April 1995. ISBN 0-89791-736-7.

[34] David E. Breen, Ross T. Whitaker, Eric Rose, and Mihran Tuceryan. Interactive occlusion and automatic object placement for augmented reality. *Computer Graphics Forum*, 15(3) :C11–C22, September 1996.

[35] Anton Fuhrman, Gerd Hesina, Fronçois Faure, and Michael Gervautz. Occlusion in collaborative augmented environments. In Michael Gervaut, Dieter Schmalstieg, and Axel Hildebrand, editors, *Virtual Environments '99. Proceedings of the Eurographics Workshop in Vienna, Austria*, pages 179–189. Springer-Verlag Wien, 1999.

[36] Blair MacIntyre and Steven Feiner. A distributed 3D graphics library. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 361–370. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[37] Thomas A. Funkhouser. Ring : A client-server system for multi-user virtual environnement.

[38] S. Zhai and P. Milgram. Human performance evaluation of isometric and elastic rate controllers in a 6doF tracking task. In *Proc. SPIE Telemanipulator Technology*, volume SPIE vol. 2057, 1993.

[39] S. Zhai. Human performance evaluation of manipulation schemes in virtual environments. In *Proc IEEE Virtual Reality International Symposium (VRAIS)*, September 1993.

[40] Shumin Zhai. Investigation of feel for 6DOF inputs : Isometric and elastic rate control for manipulation in 3D environments. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting*, volume 1 of *COMPUTER SYSTEMS : 3D Input and Display*, pages 323–327, 1993.

[41] Doug A. Bowman and Larry F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In Michael Cohen and David Zeltzer, editors, *1997 Symposium on Interactive 3D Graphics*, pages 35–38. ACM SIGGRAPH, April 1997. ISBN 0-89791-884-3.

[42] Mark Raymond Mine. Exploiting proprioception in virtual-environment interaction. Technical Report TR97-014, Department of Computer Science, University of North Carolina - Chapel Hill, May 30 1997. Mon, 2 Jun 1997 19 :23 :18 GMT.

[43] Z. Szalavári and M. Gervautz. The personal interaction panel - a two-handed interface for augmented reality. *Computer Graphics Forum*, 16(3) :335–346, August 1997. Proceedings of Eurographics '97. ISSN 1067-7055.