Environnement de Réalité Augmentée Collaboratif : Manipulation d'Objets Réels et Virtuels

Raphaël Grasset - Jean-Dominique Gascuel
iMAGIS[1]- Laboratoire GRAVIR/IMAG UMR 5527
INRIA Rhone-Alpes
655, avenue de l'Europe 38330 SAINT-ISMIER
`raphael.grasset@imag.fr`

Résumé : *La réalité augmentée est un outil indispensable pour certaines situations collaboratives. Nous proposons un environnement multi-utilisateurs, multi-périphériques dédié aux applications collaboratives. Il repose sur une architecture modulaire et facilement configurable pour une mise en place rapide d'une session de travail. Cette plate-forme est utilisée pour démontrer des techniques d'interactions très intuitives pour la manipulation d'objets réels et virtuels et la visualisation à plusieurs échelles. On introduit également des méthodes pour ajouter de façon dynamique des objets réels et virtuels, qui s'intègrent alors dans le même espace. Basé sur des concepts de vue privée et de droits d'accès aux éléments de la scène, l'environnement gère l'interaction multi-utilisateurs.*

Mots-clés : Réalité Augmentée, Collaboratif, Interaction, Tangible User Interface.

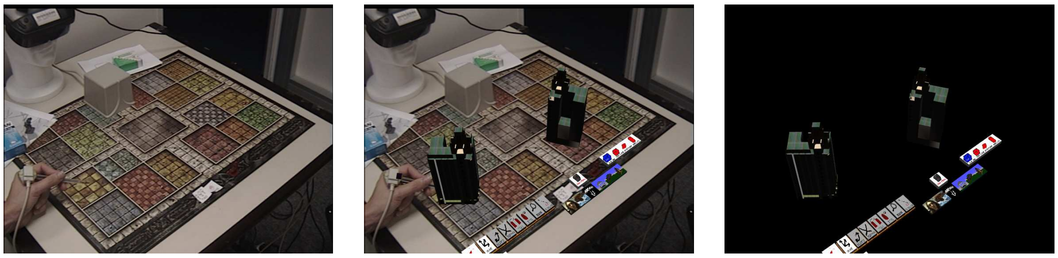

FIG. 1 – Réunion entre le réel et le virtuel : la réalité augmentée.

1 Introduction

La réalité virtuelle (RV) s'intègre de plus en plus dans les processus de développement industriel. Mais depuis quelques années ces systèmes ont montré quelques limites : la coupure totale entre l'utilisateur et le monde extérieur entraîne une difficulté de communication entre les participants ; et la complexité de l'interaction 3D pose des problèmes dans l'utilisation de ces systèmes pour des applications collaboratives. La Réalité Augmentée (RA) peut être une solution adéquate pour répondre à ces différents problèmes. Elle permet de retrouver les métaphores naturelles de communications (verbales et gestuelles) des situations collaboratives usuelles et offre de nouvelles possibilités : rajouter des animations et effets visuels, définir de nouvelles métaphores d'interactions, et effectuer automatiquement par l'ordinateur certains calculs coûteux. La réunion du même espace d'action et de perception permet à l'utilisateur d'interagir de façon complètement naturelle avec les objets réels (carte, livre, objets 3D), mais aussi virtuels (textes, images, modèles 3D), et d'avoir une parfaite intégration entre le réel et le virtuel (voir figure 2).

La réalité augmentée peut être utilisée dans de nombreux domaines basés sur un aspect coopératif : réunions de travail, visualisation urbaine entre cartographes et clients, visualisation architecturale et jeu collaboratif.

Dans cet article, nous présentons notre environnement de réalité augmentée qui permet à différents utilisateurs d'interagir avec des objets réels et virtuels. En plus des techniques d'interaction connues, notre plate-forme en présente de nouvelles. Notre système s'oriente vers une mise en oeuvre rapide d'une session et repose sur une interaction très intuitive.

[1]iMAGIS est un projet commun entre le CNRS,l'INPG,l'INRIA et l'UJF

FIG. 2 – L'espace de travail permet d'interagir avec les éléments réels (note, carte, stylo, boite) mais aussi avec les éléments virtuels (boutons, immeuble, photo, annotations virtuelles).

2 État de l'art

On se restreint ici à une présentation extrêmement rapide de la bibliographie sur le sujet. Des références plus spécifiques sont données dans le reste dans l'article. Les premiers travaux de réalité augmentée ont débuté de façon parallèle entre les domaines de l'interaction homme-machine ([Wel91, Kru91, Rob, Ber99]) et de la réalité virtuelle ([BFO92, AB94, SLG$^+$96, Aoar97]). Les principales difficultés techniques rencontrées sont l'alignement correct des objets réels et virtuels (dû à des erreurs de calibrage et des temps de latence), l'intégration visuel ;e des objets (occultation entre les objets des 2 mondes, aspect réaliste des objets virtuels).

Différents systèmes collaboratifs basés sur la RA ont déjà été réalisés, on peut citer [SFGS96, BoF$^+$99, ITM$^+$01, RBLM97, UI97, RS99, RWC$^+$, ABM$^+$97, WET00]. Ces systèmes proposent de manipuler différents éléments (réels ou virtuels), ou de visualiser des éléments virtuels dans un espace réel. La configuration matérielle repose en général sur des techniques de vision, ou sur des capteurs à 6 degrés de liberté tels que les capteurs magnétiques.

Des techniques d'interactions spécifiques pour la RA ont été introduites telle que le PIP (Personal Interaction Panel) [SG97], les *tangibles user interface* [IU97], [FIB95] ou le système RealReality [BSB97]. Mais un grand nombre de techniques de réalité virtuelle ont été reprises telles que le WIM[SCP95], ou les widgets 3D[CSH$^+$92].

L'ensemble de ces solutions ne permettent pas d'interagir et de visualiser de façon similaire - à l'aide de divers périphériques - avec des objets réels et virtuels 3D, et ce pour plusieurs utilisateurs en situation de réunion réelle. La plate-forme d'expérimentation présentée ci-dessous, nous permet d'introduire de nouvelles techniques d'interaction et de gérer plusieurs utilisateurs.

3 Plate-forme d'expérimentation

L'environnement doit permettre qu'une session soit facilement et rapidement mise en place par un utilisateur de base, ou optimisable par un utilisateur expert. L'architecture doit être assez souple et ouverte pour pouvoir supporter et moduler différentes configurations de périphériques, d'utilisateurs et d'objets manipulés.

3.1 Architecture logicielle

On définit une architecture modulaire basée sur la configuration d'une session (de travail) intuitive et rapide. Notre système repose sur 3 notions spécifiques :
– **les périphériques** : éléments d'interface qui permettent à l'utilisateur d'agir avec le monde virtuel ;
– **les utilisateurs** : personnes pouvant agir avec le monde réel et virtuel ;
– **les objets** : élément manipulés par l'utilisateur à l'aide des périphériques (les objets peuvent être de différents types : modèle 3D, image, texte, vidéo).

Pour manipuler des objets virtuels, on utilise des méthodes d'interaction haut niveau associées à des périphériques (mode "fly", "drive", trackball", "head-tracked" pour la navigation, "stylo", "wand" pour la manipulation d'objets ..etc ..). L'indépendance entre les 2 niveaux nous permet d'associer différents périphériques à différentes méthodes. Pour manipuler les éléments réels, l'utilisateur conserve ses métaphores naturelles d'interaction (si l'on souhaite une interaction entre les objets des deux mondes, on doit représenter l'objet réel par un modèle numérique, comme précisé dans le chapitre 4.2).

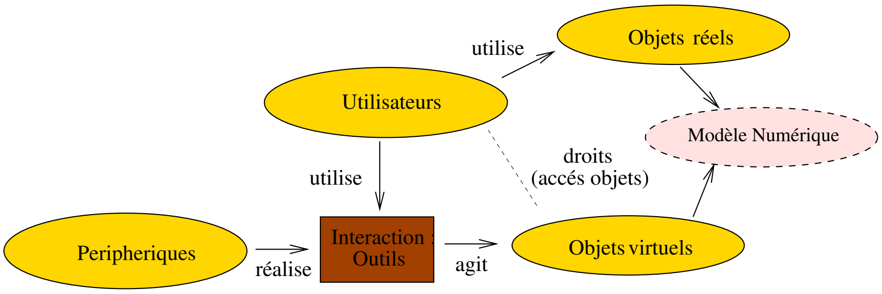

FIG. 3 – Modèle de gestion de l'interaction entre l'utilisateur et les objets réels et virtuels.

Pour pouvoir supporter un grand nombre de configurations (périphériques, utilisateurs, objets différents), notre système est découpé en modules indépendants (figure 4). Chaque module se configure séparément à l'aide d'un fichier de configuration (qui masque alors les paramètres d'initialisation des modules). L'utilisateur peut ainsi définir sa session à partir d'un éditeur visuel qui génère alors les fichiers de configuration. On se base sur XML pour définir nos fichiers de configuration (syntaxe DOM à partir de la version XML de la librairie Qt).

OpenGL Performer est utilisé pour la gestion des objets graphiques. Basé sur une architecture en graphe de scène, il supporte un grand nombre de formats d'objets (VRML, 3DS, Inventor) et de gestion automatique de plusieurs cartes graphiques. Ceci nous permet d'avoir un système multi-utilisateurs entièrement temps réel avec une qualité d'image exceptionnelle.

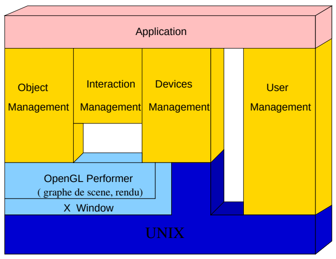

FIG. 4 – Architecture logicielle de notre système découpé en modules. La mise en oeuvre repose sur le langage C++, qui nous permet d'utiliser les avantages des langages objet..

3.2 Gestion souple du matériel

Le système doit supporter des configurations très variées : il doit assurer une connexion rapide des périphériques personnels (par exemple des casques à bas prix et des capteurs peu sophistiqués) ou considérer des environnement fixes (solution pour réunion de travail à base de stations et de matériel très performant). On doit donc pouvoir supporter plusieurs types de périphériques (capteurs, système de visualisation, souris, clavier) et les incorporer facilement au système. Chaque périphérique doit être calibré et ses données filtrées (bruit) pour avoir un parfait alignement entre les objets réels et virtuels.

On choisit un système reposant sur la notion de "plug'n play". Pour le démarrage rapide d'une simulation, des techniques de calibrage **simples et intuitives** reposant sur des manipulations par l'utilisateur ont été définies : à l'aide de quelques opérations il peut configurer les différents périphériques (voir [GDG01]). Pour réduire la latence et le bruit on utilise un filtrage de Kalman (service intégrable dans le fichier XML de façon similaire à [RS01]). Chaque utilisateur possède un fichier qui le définit (caractéristiques morphologiques). Cette définition est prise en compte lors de la configuration des périphériques par le système (ce fichier peut être accessible en ligne, ou fourni par l'utilisateur). Cette approche introduit une notion de contexte utilisateur dans le système qui peut être constitué d'un grand nombre d'informations : modèle 3D du joueur, infos personnelles, choix de configurations ..etc..

La gestion des périphériques est complètement indépendante de l'application. L'utilisation de fichier de configuration permet de masquer la mise en place parfois complexe des périphériques, laissant à l'utilisateur un objet facilement utilisable. A l'instar de nombreuses librairies (Diverse[dgi], VrJuggler[Vrj], Vess[ves]) des type de bases ont été définis : Key (valeur textuelle), Valuator (valeur numérique), Button (valeur booléenne), Locator (position 6D). Chaque périphérique (implémenté par héritage, voir figure 5) peut renvoyer un ou plusieurs de ces types.

Notre système permet de mélanger et d'intégrer des périphériques de visualisation et de sortie (HMD, moniteur, Reality Center, système de projection), périphériques d'entrée (souris, clavier, tablette Intuos, souris 3D SpaceMouse, capteur Flock of Birds) mais aussi vidéo et son.

Par exemple, pour utiliser un capteur magnétique, un casque semi-transparent I-glasses, et une souris on aura un fichier device.xml :

```
<Devices>
   <Device>
    <Type>FobInput</Type>
    <Name>Flock</Name>
    <NumFob>2</NumFob>
    <Port>3</Port>
  </Device>
  <Device>
   <Type>MouseInput</Type>
   <Name>mouse</Name>
  </Device>
  <Device>
   <Type>IGlassesView</Type>
   <XPos>128</XPos>
   <YPos>96</YPos>
   <Pipe>1</Pipe>
  </Device>
</Devices>
```

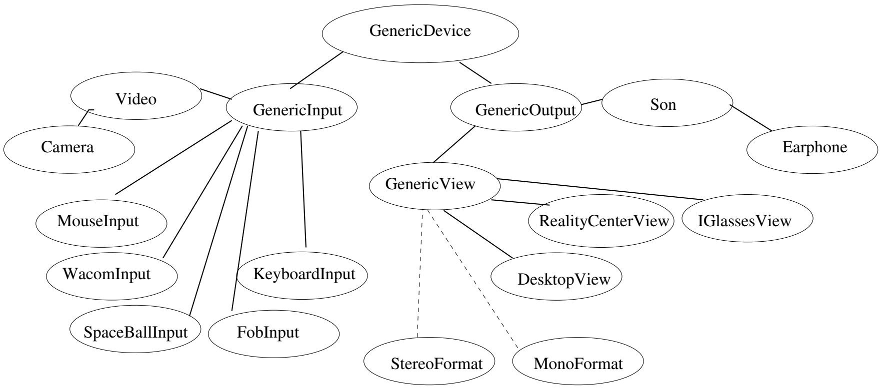

FIG. 5 – Hierarchie des périphériques utilisés.

3.3 Configuration de test

On s'intéresse principalement à des situations collaboratives co-localisées. D'après Schlichter[SKB98] la collaboration nécessite une coordination importante entre les participants à l'aide de mécanismes de communication. On peut différencier la communication directe entre participants (par geste, expression, voie) et indirecte par la manipulation d'artefacts partagés. La réalité augmentée répond parfaitement à la communication directe qui reste ici la même que pour un scénario réel. Dans notre cas la communication indirecte se base sur la manipulation d'objets réels et virtuels (dans la même approche que [FLB+94]).

On choisit donc de positionner les utilisateurs autour d'une table réelle, en position assise, équipés de différents périphériques, comme pour une réunion réelle (voir figure 6). On présente ici la configuration

FIG. 6 – Deux utilisateurs manipulant des objets virtuels intégrés dans l'environnement réel. Ils peuvent librement interagir avec les objets tout en gardant la possibilité de communiquer par voie orale, et gestuelle.

utilisée dans notre laboratoire, elle permet à l'utilisateur de naviguer dans la scène (déplacement de la tête autour de la table) et de manipuler les éléments de la scène.

Comme on souhaite conserver la configuration spatiale de réunion réelle l'espace d'action est découpé en zones (figure 7) :
- on définit la **zone partagée** comme étant la zone centrale où l'activité de groupe se crée (en général le centre de l'espace de travail) ;
- on définit la **zone privée** comme étant la zone personnelle de l'utilisateur ou il place généralement ses propres éléments (note, stylo, tasse ..etc..).

.

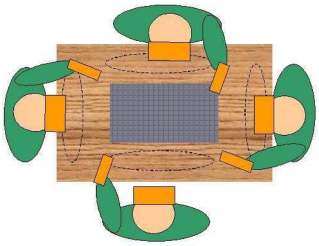

FIG. 7 – réprésentation de la zone partagée (au centre de la table) et privée (devant chaque utilisateur).

On introduit un périphérique nommé "stylo mixte" qui est un stylo réel attaché à un capteur magnétique Flock of Birds (visible figure 9). A l'aide de ce système, l'utilisateur peut interagir dans le monde réel (texte, schéma) mais aussi dans le monde virtuel (déplacement d'objets, tracé virtuel). Le temps d'apprentissage de ce système est faible de par l'usage naturel de cet objet.

Pour la visualisation on a opté pour des casques de visualisation semi-transparent de type I-glasses associés a un capteur magnétique Flock of Birds. Comme le mouvement de sa tête est suivi, l'utilisateur peut librement la déplacer pour visualiser la scène sous différents points de vues. Malgré la faible maniabilité et la faible résolution de ces systèmes d'affichages, ils sont les seuls actuellement qui permettent d'assurer une vue privée du monde et d'obtenir une vision stéréoscopique. La vue privée est indispensable pour pouvoir associer des informations spécifiques à chaque utilisateur. Les progrès technologiques permettront de remplacer ce type de périphériques par de simple lunettes avec un affichage stéréoscopique ou une projection directe sur la rétine (voir les prototypes dans [mica] ou [micb]).

Les différents périphériques sont reliés à une Onyx3 équipé de deux cartes Infinite Reality (+ une carte vidéo DIVO), 8 processeurs, 8 Go de RAM. Cette solution centralisée nous permet de nous astreindre des problèmes de distribution, mais une approche distribuée basée sur des PC équipés de cartes vidéo haute performance serait aussi satisfaisante.

FIG. 8 – Configuration de test. On peut noter le stylo équipé du capteur (en premier plan), le casque semi-transparent également équipé d'un capteur (sur la droite) et le casque virtuel opaque (sur la gauche). L'interaction se fait sur le plateau de jeu.

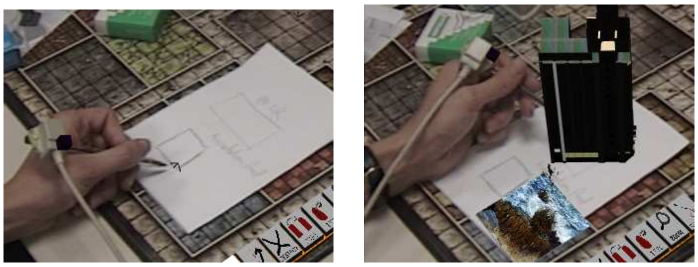

FIG. 9 – Utilisation du périphérique mixte. Sur la vignette 1, il sert à tracer des schémas et prendre des notes. Mais il peut aussi servir pour des manipulations virtuelles, telles que pour le déplacement de l'immeuble (vignette 2).

4 Nouvelles techniques d'interaction

La plate-forme nous a permis d'expérimenter des scénariis collaboratifs et de proposer un nouvel ensemble de métaphores adaptées à notre système.

Pour une situation collaborative on définit trois types d'actions principales :
- manipuler et ajouter des objets dans l'environnement : déplacer et mettre en place des éléments.
- visualiser le résultat : naviguer, avoir différents points de vue sur la scène présentée sur la table.
- réaliser des simulations : agir sur la création, simuler en temps réel le déplacement d'objets, de flux.

On s'intéresse donc à la définition de techniques d'interaction répondant à ces besoins. Les objets réels et virtuels doivent pouvoir être utilisables dans le même espace d'action et visualisés dans le même espace de perception (contrairement à une configuration classique clavier-souris-écran). On propose donc des techniques pour incorporer efficacement ces deux types d'objets.

4.1 Manipulation d'objets

On souhaite une manipulation intuitive des objets. Comme souligné dans de nombreux travaux ([SG97], [SCP95]) le retour haptique et visuel est indispensable pour avoir une véritable sensation de manipulation des objets. On se place dans un contexte de **"retour haptique faible"** qui cherche à obtenir un contact avec l'objet manipulé **suffisant** pour une sensation de retour haptique. Pour cela, et pour un meilleur réalisme, les objets virtuels respectent la loi de gravité (pas d'objet virtuel flottant dans l'espace).

Pour manipuler les objets, on considère deux types d'approche : soit l'action qui repose sur une métaphore naturelle (correspondant à une situation réelle), soit l'action repose sur une métaphore virtuelle (telle que secouer la main pour faire disparaître un objet). Le résultat peut être purement virtuel (détruire un objet) ou réel (déplacer un objet). Pour donner à l'utilisateur le choix de plusieurs actions, on place différents outils sous forme de boutons (figure 10) dans la zone personnelle de l'utilisateur (la sélection de l'outil par contact sur la table assure un retour haptique).

FIG. 10 – Utilisation d'une boîte d'outils positionnée sur la table. L'utilisateur peut changer d'outil par simple contact avec l'objet à l'aide du stylo. L'image obtenue, de faible qualité, correspond à une vue à travers le casque, filmé après avoir placé une caméra derrière le casque semi-transparent.

L'utilisateur a alors accès à un ensemble de fonctions pour interagir aisement avec les objets placés dans la zone partagée. Comme on se base sur des applications collaboratives, le système fournit un ensemble de services de base :
– déplacer les objets en 2.5D (métaphore naturelle). La 2.5D permet une parfaite intégration entre objet réel et virtuel (un objet réel repose sur un support plan).
– tourner les objets (métaphore naturelle).
– obtenir des informations sur les objets (métaphore virtuelle).
– détruire, copier des objets (métaphore virtuelle).
– annoter les objets (métaphore réel-virtuel).
– intégrer des objets de différents types : texte, image, vidéo.
Pour la réalisation , chaque objet peut ou non définir ce service. Selon l'outil et l'état courant de l'utilisateur (en action, au repos) l'objet réagit. Par exemple la table ne peut être détruite, on ne peut annoter les menus, un objet réel ne se réduit pas.

Déplacement d'objets

Pour déplacer les objets, on définit deux techniques inspirées de leur correspondantes réelles et basés sur le "retour haptique faible" :
– **le "up and down"** qui correspond à une situation où on soulève un objet, le déplace et le repose (comme pour un jeu d'échec où on passe par dessus des pions). Notre système détecte un mouvement vertical de l'objet, puis le déplacement. Ensuite, à l'aide d'un algorithme de collision, il détecte le contact de notre objet avec d'autres objets (table ou éléments réels) et le contact du stylo avec la table qui permet de valider le mouvement et désactive la sélection de l'objet (voir figure 11).
– le **"push and stop"** qui correspond à une situation où on fait glisser l'objet sur une table (comme pour un jeu de dames où on pousse les pions à l'aide de l'index). Comme on considère que l'objet reste en contact avec la surface, après sélection avec le stylo, l'utilisateur fait glisser le stylo sur la table et la fin de la sélection est activée après déplacement par un mouvement dans l'axe vertical non utilisé (voir figure 12).

FIG. 11 – Déplacement d'un pion avec la méthode "up and down". Après sélection du pion par contact du stylo sur la table et dans le pion (vignette 2), l'utilisateur peut déplacer le pion (vignette 3), puis le contact de l'objet avec la table (figure 4) affiche un retour visuel et le contact du stylo avec la table désactive la sélection le pion.

Les principales difficultés rencontrées pour la manipulation d'objets sont l'exiguité de l' espace de travail qui limite l'utilisation d'un grand nombre d'outils et la difficulté à détecter précisement le du mouvement de l'utilisateur, dûe à la faible précision de notre capteur.

FIG. 12 – Déplacement d'un pion avec la méthode "push and stop". l'utilisateur peut définir une trajectoire pour le pion (déplacement différé sur vignette 2), et un mouvement vertical termine la sélection du pion qui se place alors à la position fixée (on peut l'animer le long de la trajectoire).

4.2 Ajout d'objets

Pour permettre de manipuler des objets réels et virtuels, il est nécessaire de les importer dans notre environnement. Pour pouvoir avoir une intégration des deux mondes dans un même espace, la contrainte est la nécessité d'un modèle numérique des objets réels et virtuels. On propose d'ajouter de façon dynamique ces modèles (autant réel que virtuel) avec des méthodes spécifiques suivant le type d'objet.

Ajout d'élements virtuels

On définit une zone de boutons 3D - correspondant chacun à un modèle d'objets - dans la zone personnelle de l'utilisateur pour aisement ajouter des objets virtuels (le click sur un bouton 3D nous donne un objet virtuel à la pointe du stylo qui peut alors être placé dans la zone partagée). Pour accèder à un grand nombre d'objets, on introduit un menu à deux niveaux. Le contact sur la table lors de la sélection des icônes 3D nous procure encore une sensation de retour haptique pour la création de ces objets. L'utilisation de ces icônes 3D dans la zone personnelle (voir figure 2) permet d'introduire des objets virtuels en cours d'une simulation, et réduit la fatigue physique par rapport à un système à palette tel que le PIP (n'ayant pas de sens dans une situation assise,la palette se substituant à un plan incorporé dans l'espace de travail).

Ajout d'élements réels statiques : décor

Pour des réalisations architecturales, on souhaiterait pouvoir mélanger une maquette réelle (construction déjà réalisée) avec une maquette virtuelle (futurs travaux envisagés sur le bâtiment). Pour cela il est nécessaire d'obtenir un modèle numérique de l'objet (géométrie), et sa position dans l'environnement. Pour la géométrie du modèle numérique on peut envisager 3 approches : avoir un modèle prédéfini CAO de l'objet, réaliser un scan 3D de l'objet, saisir de façon interactive l'objet.

Pour les modèles prédéfinis, notre technique consiste à faire correspondre la représentation virtuelle de l'objet réel (modèle) avec l'objet réel lui même. A l'aide d'un périphérique d'interaction, on déplace le modèle virtuel que l'on superpose à l'objet réel (en une ou deux étapes). L'utilisateur valide de façon visuelle la bonne superposition des deux éléments. On en déduit alors la position de l'objet réel.

Si l'on ne dispose pas de modèles, on peut réaliser une saisie interactive de l'objet. On peut observer qu'un grand nombre d'objets de notre environnement on des formes géométriques basiques : parallélépipèdes (livre, boite, CD), cylindriques (tasse, bouteille), plans (feuille). En se basant sur les contraintes géométriques de ces objets, on définit une méthode de construction d'objets réels de forme générique : dans la même approche que précédemment, on construit un modèle virtuel que l'on superpose à l'objet réel de façon interactive, ce qui nous permet de déduire la géométrie et la position de l'objet réel (voir figure 13).

A partir de ces modèles numériques on peut réaliser une intégration visuelle performante de nos objets réels avec les éléments virtuels : il répond au collision permettant le placement d'objets virtuels sur ce modèle et il gère les occultations avec les objets virtuels en réalisant un rendu uniquement en z du modèle numérique de l'objet réel.

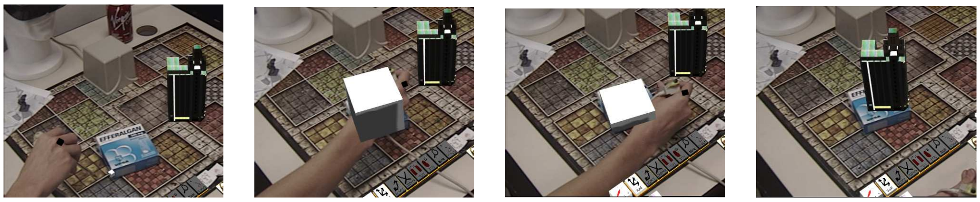

FIG. 13 – Ajout d'un cube réel à l'environnement. L'utilisateur clique sur différents points clés du cube (4 points pour obtenir position et orientation), et une prévisualistion dynamique du modèle final est présenté (vignette 1, 2, 3). Après la saisie de tous les points, on déduit le modèle, et on peut alors l'utiliser pour poser des éléments virtuels dessus (vignette 4).

Ajout d'élements réels dynamiques : tangible user-interface

Les objets réels intégrés peuvent servir comme élément de décor mais aussi servir d'interface de manipulation avec un objet virtuel. On peut par exemple associer un objet réel à un élément virtuel, le déplacement de l'objet réel correspondra par la suite au déplacement de l'objet virtuel. On obtient ainsi un meilleur retour haptique. Ce type d'interface, nommé *tangible user interface*, consiste à utiliser un élément réel pour réaliser une action virtuelle. L'action peut être le déplacement d'un objet, d'une caméra, la copie d'un objet ..etc..

Après sélection d'un élément réel (position et géométrie connue), il suffit de sélectionner un élément virtuel, pour obtenir une transformation rigide entre les deux éléments. Comme le montre la figure 14, on peut alors déplacer l'**objet réel interface** equipé d'un capteur "scratch". L'utilisation d'un capteur amovible permet de l'utiliser pour déplacer n'importe quel objet réel sans avoir une solution statique.

Bien entendu, au delà de l'utilisation de *tangible user interface* il sert aussi à déplacer un objet réel comme une maquette de bâtiment dans une simulation architecturale.

FIG. 14 – Utilisation d'un cube réel pour déplacer un élément virtuel. Le cube de la vignette 1 à été reconstruit suivant la méthode interactive précédente. On peut alors l'associer à un immeuble virtuel (vignette 2), et déplacer cet immeuble virtuel à l'aide du cube (vignette 3).

4.3 Passage de et vers le virtuel

On s'intéresse à la visualisation de la scène 3D sous différents points de vue. Milgram a défini un continuum entre le Réel et le Virtuel (figure 15) qui intègre la réalité augmentée (voir [Mil94]). Comme le montrent Kiyokawa[KTY99] et Bilinghurst[MBP01], l'intérêt du passage du réel au virtuel est guidé par la tâche : la RA pour une discussion face à face sur un élément virtuel, la RV pour une visualisation à plusieurs points de vue et à plusieurs échelles. La métaphore World In Miniature ([SCP95]) montre l'intérêt d'une telle perception pour favoriser la navigation dans une scène.

Contrairement à Billinghurst nous n'utilisons aucun un objet réel comme interface de transition entre réel, AR, et virtuel, et on définit deux cas d'immersions :

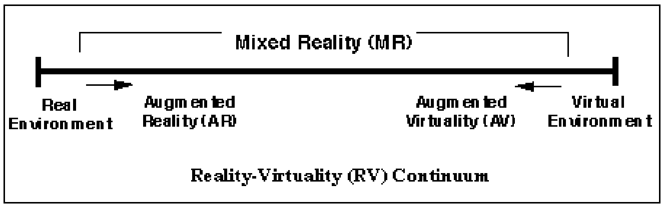

FIG. 15 – Taxinomie de Milgram pour représenter le continuum entre réel et virtuel.

– **transition automatique déclenchée par le contexte**. Indépendant de l'utilisateur elle répond a un besoin direct d'immersion déclenché par le système : visualisation d'un élément précis (guidé par le coordinateur de la session) ou phase d'action immersive (combat dans un jeu par exemple).
– **transition définie de façon interactive** : l'utilisateur spécifie une position dans la scène virtuelle à l'aide d'un périphérique d'interaction (stylo) et il se retrouve immergé à cette position. Il peut alors pleinement naviguer dans cette scène.

Au niveau de la mise en oeuvre, la transition visuelle entre RA et RV est réalisée par interpolation spline (figure 16) entre le point de vue actuel et le point de vue défini (on évite un saut spatial trop direct).

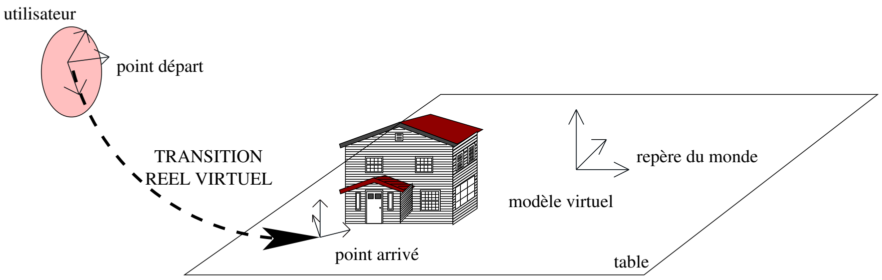

FIG. 16 – Passage du réel au virtuel. L'utilisateur plonge dans la scène.

Dans la même optique que les travaux de [oFT+99] on s'intéresse à la collaboration à différents niveaux : un utilisateur à un niveau de RA, peut guider un utilisateur immergé, au niveau RV. Il peut lui indiquer des éléments, le déplacer de façon interactive, lui définir des trajectoires. De cette façon, deux utilisateurs peuvent facilement préparer des "walktrough" ou "fly" en coopérant, ce qui évite à un mono-utilisateur un changement difficile entre les vues (application à la création de mouvements de caméra pour les jeux 3D, pour des films en image de synthèse ..etc..).

On ajoute aussi la possibilité de positionner différentes caméras permettant à l'utilisateur d'avoir différents points de vues (il peut alors "switcher" entre les caméras). Et pour finir, on intègre un service de zoom sur les objets avec la contrainte de conservation de contact de l'objet sur la zone partagée.

5 Gestion multi-utilisateurs

Dans une session de travail, les utilisateurs ne nécessitent pas le même type d'informations suivant leur rôle dans la session. Un architecte ne souhaite pas manipuler et visualiser les mêmes données qu'un directeur ou un commercial. Ceci nous amène donc à définir des **vues personnelles** pour chaque utilisateur, ou chacun visualise soit la même donnée sous différents aspects (en fil de fer, en faces cachées, texturé) soit des données différentes (liaisons électriques pour un électricien, tuyauterie pour un plombier dans une visualisation architecturale).

Pour mettre en place ce système, on définit un degré d'intégration de l'utilisateur dans la session :
– **observateur passif** : l'utilisateur a un retour (visuel) sur l'action, mais ne peut interagir avec l'environnement.
– **observateur actif** : l'utilisateur a un retour visuel et peut agir sur l'environnement à faible échelle (navigation, immersion).

– **participant actif** : l'utilisateur peut interagir complètement sur le système (navigation et interaction).
– **coordinateur** : l'utilisateur a des droits privilégiés sur le système (modification de la configuration).
Chaque niveau donne alors des possibilités (sous forme de droits) sur l'interaction et la manipulation des objets. Le profile professionnel d'une personne est utilisé pour la hiérarchisé dans des groupes (architecte, ingénieur, secrétaire, commercial) et définir le type d'informations qu'elle visualise.

Similairement à [Smi96] cette méthode utilise des matrices creuses pour chaque objet spécifiant le droit d'accés de chaque utilisateur (qu'on applique ici à la visibilité, la manipulation). Les droits obeissent à une hiérarchie de type Unix : pour chaque objet on définit un droit universel, de groupe et personnel (voir figure 17). Pour le droit de visibilité, on met en pratique cette méthode sous OpenGL Performer par l'utilisation de masques sur les noeuds et les vue de l'utilisateur (utilisé par [SEG]).

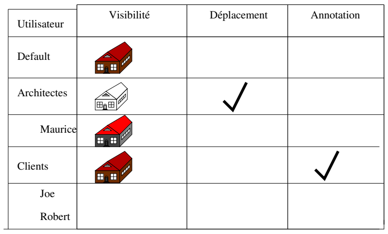

FIG. 17 – Représentation de la matrice d'accès à un objet. A chaque utilisateur (classé) des droits spécifique sont précisés. Si la case est vide soit on hérite du droit de plus haut niveau, soit l'action est interdite. La redéfinition d'un droit est prioritaire pour un utilisateur par rapport à un groupe. Par exemple, Maurice hérite du droit de déplacement de la maison, mais redéfinit la visibilité de la maison par un modèle en faces cachées.

Chaque objet comprend différentes géométries (à différents niveaux de détails) et différents aspects (mode fil de fer, faces cachées, texturé) qu'on associe à chaque utilisateur. Appliqué à un jeu, on aura des droits pour chaque joueur suivant leur rôle dans le jeu, comme expliqué figure 18.

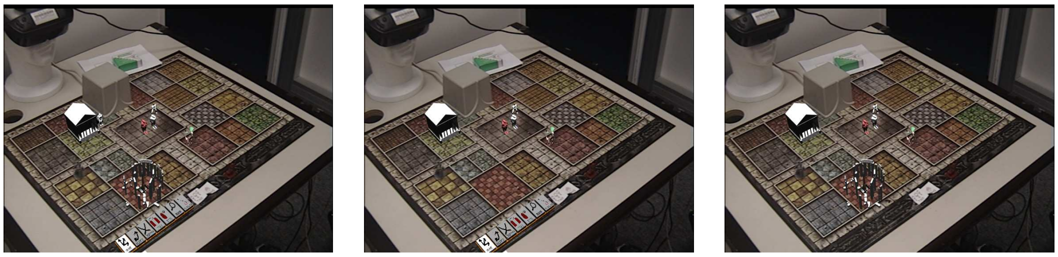

FIG. 18 – La vue de la même scène par trois utilisateurs différents. Le coordinateur voit tous les éléments de la scène (vignette 1). Le premier utilisateur voit et accède à ses menus et voit les autres joueurs (vignette 2). Le dernier joueur verra les "bâtiments cachés", mais ne visualisera pas les autres éléments tel que le menu du premier joueur (vignette 3).

6 Conclusion et Futurs Travaux

On a présenté un environnement de réalité augmentée basé sur une architecture très souple, pour mettre en place des sessions de travail avec différents utilisateurs, différents périphériques. On a proposé plusieurs techniques pour la manipulation d'objets réels et virtuels dans un même espace. L'interaction est réalisée principalement par notre périphérique le "stylo mixte" et par des *tangible user interface*. On peut facilement ajouter des objets réels et virtuels en cours de session par de simples manipulations. De nombreux services dédiés aux applications collaboratives ont été définis, tel que la possibilité de s'immerger dans la scène placée dans l'espace de travail. La gestion du multi-utilisateurs passe par l'utilisation de vue privée permettant un affichage spécifique pour chaque utilisateur.

Notre système permettant de gérer des animations, il serait intéressant de pouvoir visualiser des simulations dans l'espace de travail. Un autre sujet d'étude serait la gestion de la phase précédente d'une session et son analyse à posteriori en proposant des services pour re-simuler la session. Pour une meilleure intégration des objets virtuels et réels, un modèle de lumière pourra être défini (modulation de l'éclairage à partir d'un précalcul de radiosité). L'utilisation de méthodes de déplacement d'objets réels sans capteurs devra être intégrée à notre système. De nouvelles techniques d'interactions de RA plus naturelles doivent être étudiées (en s'inspirant des travaux de [HE99]).

Références

[AB94] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through hmd, 1994.

[ABM+97] Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. The two-user responsive workbench : Support for collaboration through independent views of a shared space. *Computer Graphics*, 31(Annual Conference Series) :327–332, 1997.

[Aoar97] R. Azuma, A. of, and a reality. Presence : Teleoperators and virtual environment, 1997.

[Ber99] François Berard. *Vision par ordinateur pour l'interaction homme-machine fortement couplée,*. PhD thesis, Université Joseph Fourier, Grenoble, France, 1999.

[BFO92] Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi. Merging virtual objects with the real world : Seeing ultrasound imagery within the patient. *Computer Graphics*, 26(2) :203–210, 1992.

[BoF+99] A. Butz, T. ollerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping users and computers in a collaborative 3d augmented reality, 1999.

[BSB97] F. Bruns, K. Schfer, and V. Brauer. A new approach to human-computer interaction – synchronous modelling in real and virtual spaces, 1997.

[CSH+92] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics, Vol. 26*, pages 183–188, 1992.

[dgi] Dgipf. http ://www.diverse.vt.edu/dgipf.

[FIB95] George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks : Laying the foundations for graspable user interfaces. In *CHI*, pages 442–449, 1995.

[FLB+94] M. Fjeld, K. Lauche, M. Bichsel, F. Voorhorst, H. Krueger, and M. Rauterberg. Physical and virtual tools : Activity theory applied to the design of groupware, 1994.

[GDG01] Raphael Grasset, Xavier Décoret, and Jean-Dominique Gascuel. Augmented reality collaborative environment :calibration and interactive scene editing. In *VRIC 2001*, May 2001. Laval Virtual.

[HE99] Willi Bruns Hauke Ernst, Kai Schäfer. Creating virtual worlds with a graspable user interface. In *15th Twente Workshop on Language Technology (TWTL 15) : Interactions in Virtual Worlds*, 1999.

[ITM+01] Poupyrev I., D.S. Tan, Billinghurst M., Kato H., Regenbrecht H., and Tetsutani N. Tiles : A mixed reality authoring interface. In *Interact*, 2001.

[IU97] Hiroshi Ishii and Brygg Ullmer. Tangible bits : Towards seamless interfaces between people, bits and atoms. In *CHI*, pages 234–241, 1997.

[Kru91] Myron Krueger. *Artificial Reality II*. Addison-Wesley, Reading, Massachusetts, U.S.A., 1991.

[KTY99] Kiyoshi Kiyokawa, Haruo Takemura, and Naokazu Yokoya. Seamlessdesign : A face-to-face collaborative virtual/augmented environment for rapid prototyping of geometrically constrained 3-d objects. In *ICMCS, Vol. 2*, pages 447–453, 1999.

[MBP01] Hirokazu Kato M. Billinghurst and Ivan Pouyrev. The magicbook :a transitional ar interface. *Computer and Graphics*, 25(5) :745–753, october 2001.

[mica] Microoptical http ://www.microopticalcorp.com/.

[micb] Microvision. http ://www.mvis.com/.

[Mil94] P. Milgram. Augmented reality : a class of displays on the reality-virtuality continuum, 1994.

[oFT+99] T. ollerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring mars : developing indoor and outdoor user interfaces to a mobile augmented reality system, 1999.

[RBLM97] M. Rauterberg, M. Bichsel, U. Leonhardt, and M. Meier. Build-it : a computer vision-based interaction technique of a planning tool for construction and design, 1997.

[Rob] S. Robinson. Brightboard : A video-augmented environment.

[RS99] Jun Rekimoto and Masanori Saitoh. Augmented surfaces : A spatially continuous work space for hybrid computing environments. In *CHI*, pages 378–385, 1999.

[RS01] G. Reitmayr and D. Schmalstieg. Opentracker - an open software architecture for reconfigurable tracking based on xml, 2001.

[RWC+] R. Raskar, G. Welch, M. Cutts, A. Lake, and L. Stesin. Fuchs : The office of the future : A unified approach to image-based modeling and spatially immersive displays.

[SCP95] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual reality on a WIM : Interactive worlds in miniature. In *Proceedings CHI'95*, 1995.

[SEG] Zsolt Szalavári, Erik Eckstein, and Michael Gervautz. Collaborative gaming in augmented reality.

[SFGS96] D. Schmalstieg, A. L. Fuhrmann, M. Gervautz, and Zs. Szalavári. 'studierstube' - an environment for collaboration in augmented reality'. In *Collaborative Virtual Environments '96, Nottingham, UK, Sep. 19-20*, 1996.

[SG97] Z. Szalavari and Michael Gervautz. The personal interaction panel - a two-handed interface for augmented reality. *Computer Graphics Forum*, 16(3) :335–346, 1997.

[SKB98] J. Schlichter, M. Koch, and M. Burger. Workspace awareness for distributed teams, 1998.

[SLG+96] Andrei State, Mark A. Livingston, William F. Garrett, Gentaro Hirota, Mary C. Whitton, Etta D. Pisano, and Henry Fuchs. Technologies for augmented reality systems : realizing ultrasound-guided needle biopsies. *Computer Graphics*, 30(Annual Conference Series) :439–446, 1996.

[Smi96] Gareth Smith. Cooperative virtual environments : Lessons from 2d multi user interfaces. In *Computer Supported Cooperative Work*, pages 390–398, 1996.

[UI97] Brygg Ullmer and Hiroshi Ishii. The metadesk : Models and prototypes for tangible user interfaces. In *ACM Symposium on User Interface Software and Technology*, pages 223–232, 1997.

[ves] Vess. http ://vess.ist.ucf.edu/.

[Vrj] Vrjuggler. http ://www.vrjuggler.org/.

[Wel91] P. Wellner. The digitaldesk calculator : Tangible manipulation on a desk top display. In *Proc. ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 107–115., 1991.

[WET00] Broll W., Meier E., and Schardt T. The virtual round table - a collaborative augmented multi-user environment. In *ACM CVE 2000 : The 3rd International Conference on Collaborative Virtual Environments*, pages 39–46, 2000.