

Un système de reconnaissance de gestes générique pour la Réalité Virtuelle

A generic gesture recognition system for Virtual Reality

Bruno Bossard
Damien Touraine

Thomas Convard
Patrick Bourdot

Annelies Braffort
Michèle Jardino

LIMSI-CNRS

BP 133, 91403 Orsay CEDEX

{Bruno.Bossard,Thomas.Convard,Annelies.Braffort,Damien.Touraine,Patrick.Bourdot,Michèle.Jardino}
@limsi.fr

Résumé

Notre travail dans le domaine de la Réalité Virtuelle se focalise en partie sur les interactions multimodales et plus particulièrement sur les modalités gestuelles et vocales. Ces deux modalités permettent une interaction naturelle avec l'environnement immersif dans lequel est plongé l'utilisateur. Cet article décrit l'état actuel du système de reconnaissance de gestes qui a été développé pour permettre les interactions gestuelles en environnement immersif. Certains choix pour l'implémentation du système ont été faits en fonction de l'architecture logicielle de notre plate-forme de Réalité Virtuelle et du vocabulaire gestuel employé à ce jour. Cela a donc abouti à un système de reconnaissance de type statistique qui peut gérer les données provenant de plusieurs périphériques à la fois. D'autre part cet article expose les choix de primitives de représentation des gestes qui ont été faits ce qui permet au système de reconnaissance de distinguer plus facilement les différents types de gestes.

Mots Clef

Reconnaissance de Gestes, Réalité Virtuelle, Primitives, Gant Numérique.

Abstract

Our work in Virtual Reality is partially focused on multi-modal interactions, and more exactly on gestural and vocal modalities. These two modalities allow to the user a natural interaction with the virtual world which surround him. This article describes the current state of the gesture recognition system which has been implemented for gestural interactions in immersive environments. The implementation of the system has been done according to the properties of our Virtual Reality platform and the properties of the gestural vocabulary used at this moment. The result is

a statistical recognition system which can handle several devices at once. There is also in this article a description of different possible sets of features in order to get a better recognition system.

Keywords

Gesture Recognition, Virtual Reality, Features, Data Glove.

1 Introduction

Les environnements immersifs sont soit des dispositifs de grandes tailles composés d'écrans retroprojetés (CAVE [6], Workbench [8]) soit des dispositifs à base de HMD (visio-casque à cristaux liquides). Il est évident que les moyens habituels d'interaction (clavier, souris ...) ne sont pas adaptés à ces environnements. De nombreux types de périphériques (souris 3D, systèmes à retour d'effort, gants numériques, joysticks ...) ont été développés pour faciliter les interactions avec le monde virtuel. Certains de ces périphériques (souris 3D, joysticks ...) permettent des interactions assez intuitives mais qui n'en restent pas moins différentes des interactions que l'on a l'habitude d'utiliser dans le monde réel.

Nous avons plutôt une approche multimodale des interactions en Réalité Virtuelle ([14]) et plus particulièrement vers les interactions naturelles que sont la voix et le geste. Cette approche est issue de notre expérience dans le domaine multimodal et gestuel ([7], [5]).

Dans cet article, nous présentons l'état actuel du travail effectué sur modalité gestuelle qui a abouti à la création d'un système de reconnaissance de gestes, et nous décrivons l'intégration de ce travail dans les autres travaux menés en Réalité Virtuelle. Les environnements virtuels n'étant pas très propices au suivi de gestes par caméra (faible lumino-

sité), nous préférons travailler avec des gants numériques. Toutefois, l'architecture présentée ici reste valide quelque soit le système de capture de geste utilisé.

Tout d'abord nous allons présenter dans la section 2 le contexte du système de reconnaissance de gestes : l'architecture logicielle dans laquelle il s'intègre, puis le type de vocabulaire gestuel ciblé. Dans la section 3 nous exposons l'application de reconnaissance de geste en elle même, ce qui nous a amené au choix d'un certain type de système de reconnaissance et quelles sont les applications qui utilisent le système. La section 4 décrit l'expérimentation qui a été faite pour comparer différents ensembles de primitives de représentation du geste. Le choix de ces primitives est important car de lui dépend la capacité du système à bien distinguer les différents gestes. Le vocabulaire gestuel actuel ne comporte que des postures statiques. Mais il est prévu d'étendre ce vocabulaire à des gestes dynamiques, et la section 5 traitera des travaux qui sont en cours de réalisation sur ce sujet.

2 Contexte

2.1 La plate-forme EVI3d

La plate-forme logicielle EVI3d [14] a été développée au sein du laboratoire pour permettre la création d'applications en Réalité Virtuelle et Augmentée. Cette plate-forme est basée sur une architecture distribuée à deux canaux de communication (événementiel et flot de données). Dans le cadre de ce papier, seule la partie événementielle nous intéresse et celle-ci est basée sur l'EVserveur [4], une approche client/serveur où le serveur est lui-même distribué sur plusieurs machines.

Une application qui utilise la plate-forme EVI3d est donc composé de noeuds clients et de noeuds serveurs pouvant être répartis sur plusieurs machines. L'organisation de ces noeuds et la transmission d'événements entre eux se déroule de la manière suivante.

- A un noeud serveur peuvent se connecter plusieurs noeuds clients ou serveurs. On va donc pouvoir former un réseau de noeuds (qui aura une forme arborescente) permettant de transférer des événements d'un noeud à un autre.
- Lors de sa connection à un noeud serveur, un client est informé des différentes sources de données possibles (périphériques et éventuellement clients) et peut alors choisir parmi ces sources celles dont il a besoin.
- Un périphérique est géré par le noeud serveur de l'ordinateur auquel il est connecté. Les données acquises par ce périphérique sont envoyées sous forme d'événements à tous les noeuds clients qui ont demandé les données de ce périphérique. Un client reçoit donc uniquement les événements correspondant aux périphériques demandés.
- Un noeud client peut envoyer des événements au noeud serveur auquel il est raccordé, ce dernier renverra les événements à tous les autres noeuds qui sont demandeurs de ces événements.

Du fait de cette disposition client/serveur et de l'utilisation d'événements, le système offre une gestion simplifiée des périphériques. En effet un client n'a pas à gérer les périphériques dont il a besoin, il n'y a pas non plus de conflit d'accès à un périphérique et un client n'a pas à se soucier de l'acheminement des données jusqu'à lui (il indique juste à son serveur qu'il veut les données de ce périphérique).

D'autre part, on voit que cette architecture est hautement modulaire. En effet, un client développé pour une certaine application de Réalité Virtuelle peut très bien être réutilisé dans une autre. On peut également remplacer un périphérique par un autre du même type sans que cela pose de problème aux clients (il faut juste que les événements contiennent le même type de données).

Dans l'objectif de pouvoir développer des applications permettant des interactions multimodales, le système de reconnaissance de gestes a été développé sous forme d'un client. Ce client peut donc utiliser les données provenant de n'importe quel type de périphérique (gant numérique, capteur de position-orientation...) pour reconnaître des gestes qu'il émet sous forme d'événements.

Dans la figure 1, on peut voir un exemple de la disposition des périphériques, des serveurs et des clients sur quatre ordinateurs différents. Un ordinateur dispose d'un périphérique de capture de position-orientation (Flock of Bird), un autre d'un gant numérique (Cyberglove) et du client de reconnaissance de gestes, sur un troisième fonctionne le système de reconnaissance vocale (qui fonctionne en tant que périphérique) et enfin le quatrième ordinateur sert à faire fonctionner l'application de Réalité Virtuelle (qui est un client). Cet exemple correspond à une configuration qui a été utilisée plusieurs fois lors de manifestations de type "grand public" ou conférence auxquelles à participé le LIMSI, ce n'est donc pas seulement un exemple théorique de ce que l'on peut faire avec l'EVserveur de la plate-forme EVI3d.

2.2 Les gestes en Réalité Virtuelle

Différents types d'interactions gestuelles peuvent être utilisés suivant les tâches que l'on veut accomplir (manipulation d'objets, description d'objets, création/modification d'objets...). Dans le domaine de la manipulation d'objets, de nombreux travaux utilisent le mouvement de la main comme moyen d'interaction; ils ont principalement été appliqués dans les environnements immersifs de type "Workbench" comme dans [1]. D'autres approches à base de gestes déictiques (gestes de désignation) permettent la manipulation d'objets à distance [9]. On peut aussi bien évidemment utiliser le geste naturel de saisie pour manipuler les objets virtuels qui sont à portée de main [10].

On peut aussi utiliser le geste comme moyen de description. Dans [13], des gestes iconiques sont utilisés pour

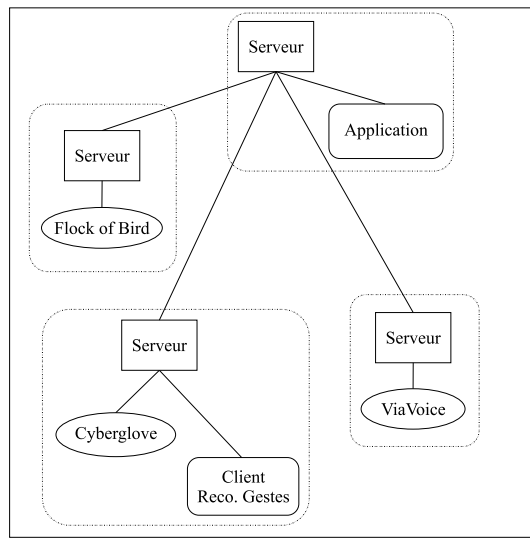


FIG. 1 – Exemple d’une configuration d’architecture distribuée avec la plate-forme EVI3d

décrire la forme d’objets virtuels. On peut également créer des formes en utilisant le geste. Par exemple dans [11], on peut voir la création de surfaces par extrusion à partir d’un profil, ou la création de primitives géométriques (sphère, cube...) à partir de la forme de la main.

On pourrait aussi utiliser le geste pour pouvoir passer des commandes à l’application. Mais cela demande un vocabulaire spécifique que l’utilisateur doit apprendre, et d’autre part cela peut se révéler fatigant et fastidieux si certaines commandes doivent être fréquemment effectuées. C’est pourquoi les interactions gestuelles sont souvent couplées à la voix qui est plus adaptée à cette tâche.

Les gestes peuvent aussi être utilisés pour la navigation [10], par exemple en pointant du doigt la direction où l’on veut se rendre. Mais là non plus les gestes ne sont pas le moyen le plus indiqué. Il vaut mieux utiliser des périphériques conçus pour cette tâche ou les mouvements de l’utilisateur (si ce dernier fait un pas en avant, on se déplace vers l’avant, si il fait un pas sur le côté on se déplace sur le côté, etc.).

2.3 Gestes utilisés

Notre démarche est de permettre l’utilisation d’interactions multimodales et naturelles dans un environnement immersif et de pouvoir utiliser ces interactions dans différents types d’applications en Réalité Virtuelle. De plus, nous souhaitons que le système de reconnaissance soit multi-utilisateur (mis à part l’inévitable phase initiale de calibration des gants numériques). Nous avons donc choisi délibérément dans un premier temps de retenir un vocabulaire gestuel le plus simple et le plus universel possible, quitte à compléter ce vocabulaire générique avec des gestes plus spécifiques aux applications dans un deuxième temps.

Ainsi, le vocabulaire visé par notre système de reconnaissance est assez limité : il y a un geste de saisie pour manipuler les objets (fig. 2), un geste de désignation pour pointer les objets (fig. 2), et éventuellement un geste pour sélectionner les objets à distance (on peut le remplacer par une commande vocale combinée au geste de désignation).

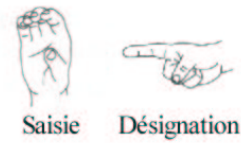


FIG. 2 – “Saisie” et “désignation”

Ces gestes sont relativement simples et s’utilisent de façon isolée (c’est à dire qu’ils ne se combinent pas entre eux), ce qui facilite la tâche de reconnaissance. D’autre part ce sont des gestes qui correspondent parfaitement aux gestes que nous utilisons dans la vie courante pour interagir avec notre environnement réel, ce qui répond bien à notre besoin d’interactions “naturelles”.

Nous allons maintenant décrire le système de reconnaissance de gestes et des choix qui ont été fait pour son implémentation.

3 Description du système

3.1 Choix d’un système statistique

Du fait que l’on ait affaire à des gestes isolés assez simples, on n’a pas besoin d’un système de reconnaissance très complexe. On n’a pas besoin de traiter les problèmes de segmentation et de coarticulation, ni d’avoir à faire une quelconque analyse syntaxique. Un simple système statistique peut résoudre le problème en modélisant les postures statiques des gestes.

Dean Rubine présente dans sa thèse [12] un système statistique permettant d’effectuer la reconnaissance de tracés en 2 dimensions. Il a l’intérêt d’être très ouvert et simple d’utilisation. De plus il fonctionne avec une phase d’apprentissage qui n’est pas fastidieuse. Nous nous sommes donc basé sur ce système qui correspond bien à nos besoins pour la reconnaissance de gestes et l’avons adapté pour le cas des gestes en 3 dimensions.

Un geste va être codé sous la forme d’un ensemble de primitives (par exemple les valeurs angulaires des articulation, la vitesse du mouvement, la courbure...), donc à un geste va correspondre un vecteur. Pour représenter chaque classe de geste, on utilise le vecteur moyen et la matrice de covariance associée. C’est la phase d’apprentissage qui va s’occuper de calculer ces deux éléments. Si on note $\bar{p}_i^c$ la i ème composante du vecteur moyen pour la classe c , p_{ei}^c la i ème composante du vecteur exemple numéro e , E^c le

nombre d'exemples appris par le système pour la classe c , et P le nombre d'éléments des vecteurs, on obtient le vecteur moyen tout simplement de la manière suivante :

$$\bar{p}_i^c = \frac{1}{E^c} \sum_{e=1}^{E^c} p_{ei}^c \quad 1 \leq i \leq P$$

On calcule ensuite chaque élément s_{ij}^c (i et j les indices de l'élément) des matrices de covariance s^c (au coefficient $\frac{1}{E^c}$ près) :

$$s_{ij}^c = \sum_{e=1}^{E^c} (p_{ei}^c - \bar{p}_i^c)(p_{ej}^c - \bar{p}_j^c) \quad 1 \leq i, j \leq P$$

Pour effectuer la classification des gestes, on utilise des fonctions linéaires dont les poids sont calculés en fonction des vecteurs moyens et d'une matrice de covariance s obtenue à partir des matrices de covariance de chaque classe. C correspond au nombre de classes, i et j aux indices d'un élément de la matrice.

$$s_{ij} = \frac{\sum_{c=1}^C s_{ij}^c}{-C + \sum_{c=1}^C E^c} \quad 1 \leq i, j \leq P$$

Pour chaque classe, on va donc avoir une fonction d'évaluation dont les poids sont calculés de la manière suivante (w_j^c est le poids pour la j ème composante du vecteur fourni à la fonction d'évaluation, w_0^c est le terme constant de la fonction d'évaluation) :

$$w_j^c = \sum_{i=1}^P (s^{-1})_{ij} \bar{p}_i^c \quad 1 \leq j \leq P$$

$$w_0^c = -\frac{1}{2} \sum_{i=1}^P w_i^c \bar{p}_i^c$$

Une fois l'apprentissage effectué (c'est à dire le calcul des vecteurs moyens, des matrices de covariance et des poids des fonctions d'évaluation) on peut classifier un geste en calculant un score pour chaque classe c à l'aide des fonctions d'évaluation. Le score v^c d'un vecteur x est obtenu avec le calcul suivant :

$$v^c = w_0^c + \sum_{i=1}^P w_i^c x_i \quad 0 \leq c < C$$

La classe qui obtient le score le plus fort est considérée comme la classe du geste que l'on veut reconnaître. Bien évidemment ce type de classification ne permet pas de déterminer si un geste appartient ou non au vocabulaire appris par le système. Dean Rubine propose de remédier à ce problème en utilisant la distance de Mahalanobis entre x et $\bar{p}_i^c$ et la probabilité que x appartienne à la classe c . Toutefois cette solution ne nous convient pas car d'une part il faut déterminer des seuils permettant de juger si

une classification est correcte ou non (et actuellement, il n'y a pas de processus permettant d'obtenir ces seuils de manière automatique); d'autre part les valeurs de comparaison obtenues dépendent de l'apprentissage effectué.

L'utilisation d'un système statistique permet également d'effectuer la reconnaissance de gestes en "temps réel". On n'est pas obligé d'attendre la fin du geste pour pouvoir le classifier, contrairement à des systèmes basés par exemple sur des modèles de Markov cachés qui analysent une séquence gestuelle dans son intégralité. Il faut noter tout de même que cela est possible uniquement parce que les gestes correspondent à des postures statiques de la main. Il suffit donc d'une seule trame de données pour pouvoir classifier un geste.

Cependant, notre système permet de travailler sur des fenêtres pouvant aller de une trame à plusieurs dizaines de trames. On pourra donc utiliser le système avec des gestes dynamiques évoluant sur plusieurs trames. On fournit au système un vecteur calculé à partir des trames de la fenêtre. Le plus souvent, on fournit la valeur moyenne, mais on peut aussi utiliser la valeur maximale, minimal, l'écart entre les extrêmes ... Selon les besoins et la fréquence d'acquisition des données, nous avons déterminé de façon empirique que des fenêtres de 1 à 4 trames convenaient. L'utilisation de fenêtres de taille plus grande induit une sensation de décalage pour l'utilisateur entre les gestes qu'il fait et leur action sur les objets de la scène virtuelle.

3.2 Intégration à la plate-forme EVI3d

Notre système de reconnaissance de gestes est donc implémenté sous la forme d'un client ce qui permet d'accéder facilement à n'importe quel type de périphérique et uniquement au(x) périphérique(s) demandé(s). On peut donc utiliser plusieurs périphériques simultanément comme sources de données (par exemple un capteur 6 degrés de liberté et un gant numérique). Et dans ce cas on peut décider de synchroniser la classification des trames sur tel ou tel périphérique (par exemple le périphérique le plus lent pour ne pas avoir de répétition de valeurs au niveau de ce périphérique). On peut aussi avoir plusieurs instances du client, chacun récupérant les données d'un périphérique donné, pour pouvoir effectuer la reconnaissance des gestes de la main droite et de la main gauche, ou les gestes de différentes personnes présentes au sein du dispositif immersif.

Les gestes reconnus sont ensuite envoyés sous forme d'événements à travers l'arborescence des noeuds de l'EV-serveur. Cela permet d'interfacer facilement le système de reconnaissance de gestes avec n'importe quelle application de réalité virtuelle. De ce fait, si deux applications immersives ont le même vocabulaire gestuel, l'utilisateur n'aura besoin de faire qu'un seul apprentissage pour les deux applications. Dans la figure 1, on peut voir le schéma du dispositif que nous utilisons pour la plupart de nos appli-

cations qui combinent le geste et la parole. On voit bien dans ce schéma que le système de reconnaissance de geste constitue un module utilisable avec n'importe quelle application et que l'on peut combiner avec d'autres modalités comme la parole. Cette conception modulaire permet également d'interchanger le système de reconnaissance actuel par un autre utilisant une autre technique de classification.

3.3 Architecture logicielle du client

L'application de reconnaissance de gestes a été divisée en deux parties : une pour l'interface avec l'utilisateur et une pour la gestion des périphériques, du système de reconnaissance et des primitives (voir figure 3). Seule la partie interface a "connaissance" de l'autre partie. Cela permet de faire évoluer les deux parties séparément et éventuellement on peut intégrer la partie système de reconnaissance à une autre application.

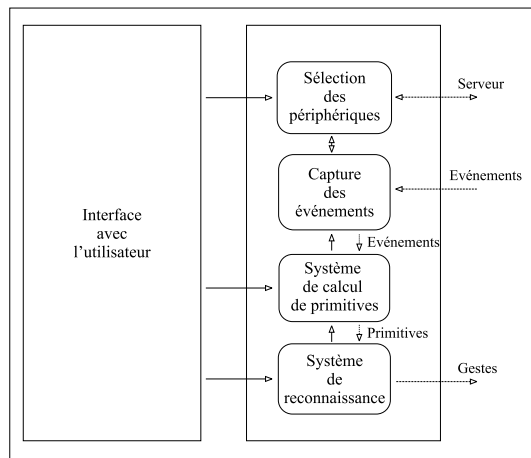


FIG. 3 – Architecture du client de reconnaissance de gestes

Dans la partie du système de reconnaissance, on trouve quatre modules :

- Un module permettant de choisir les périphériques disponibles. Ce module interagit directement avec le serveur, auquel le client est connecté, pour l'informer des besoins du client.
- Un module qui effectue l'acquisition des événements et qui gère la synchronisation par rapport à l'un des périphériques.
- Un module pour calculer les primitives. Une primitive peut aussi bien être une simple valeur brute issue d'un capteur, qu'une valeur issue d'un calcul complexe (on dispose dans ce module de tout un ensemble d'opérateurs et de fonctions que l'on peut combiner pour effectuer un calcul).
- Et enfin le module effectuant la classification des gestes. La partie interface va permettre à l'utilisateur d'interagir avec les différents modules du système. On va disposer de la liste des périphériques avec la fréquence de leurs événements ce qui permet de sélectionner les périphériques

voulus et de décider sur quel périphérique se synchroniser. On peut également choisir via l'interface les primitives que l'on veut utiliser et les fichiers correspondant à un apprentissage bien particulier. On peut ainsi passer d'un utilisateur à un autre ou changer de périphérique sans avoir besoin de relancer tout le système.

3.4 Utilisation dans des applications de Réalité Virtuelle

Dans le cadre de l'étude des environnements immersifs, on souhaiterait savoir si ces nouvelles interfaces sont capables d'améliorer les outils de CAO (Conception Assistée par Ordinateur) de sorte qu'ils puissent mieux servir de support de conception. La Réalité Virtuelle offre de nouvelles perspectives pour la modélisation géométrique, en terme de réalisme et d'interaction. Le réalisme est obtenu grâce à la visualisation stéréoscopique qui améliore la perception de la profondeur. De plus ce type de retour visuel améliore considérablement la vision et l'analyse d'objets à formes complexes en apportant des informations dénuées d'ambiguïté. Une interaction multimodale immersive permettrait de garder la composante gestuelle en CAO, qui a disparue avec l'apparition des stations de travail utilisant clavier et souris. Les gestes techniques du dessinateur ont été remplacés par des interactions de types WIMP. Or pour une tâche de conception, avoir un geste naturel vis à vis de cette tâche est important, cela favorise entre autre la créativité et l'originalité du travail de l'utilisateur. La Réalité Virtuelle permettrait de retrouver un côté manuel en CAO tout en bénéficiant des apports des ordinateurs en terme de calcul.

Une première utilisation du système de reconnaissance a été faite avec une application test (figure 4) visant à utiliser le noyau de modélisation géométrique OpenCASCADE dans un environnement immersif. Pour cette application, le geste seul a été utilisé comme moyen d'interaction pour créer, manipuler, désigner et sélectionner les objets.

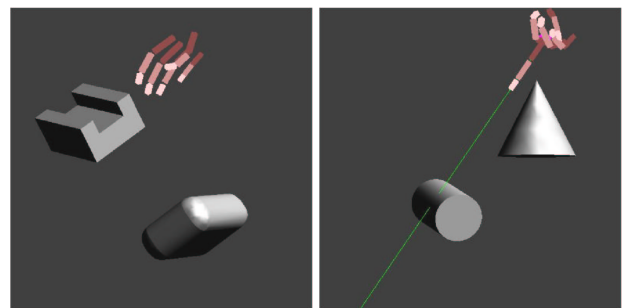


FIG. 4 – Application de test pour l'édition d'objets 3D

Suite à ce premier essai, une application de démonstration multimodale (figure 5) a été créée dans le but de démontrer les différentes interactions possibles en combinant le geste

et la parole. Le geste de saisie permet de manipuler tout objet qui est à portée de main, et le geste de désignation combiné aux commandes vocales (par exemple “prend cet objet”) permet d’interagir avec les objets situés hors de portée.

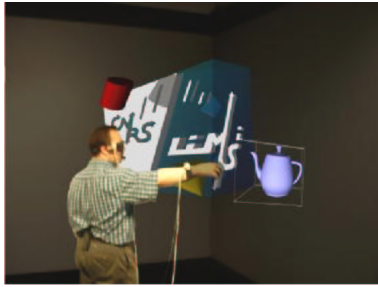


FIG. 5 – Démonstrateur “LogoMultimodal” de la plate-forme EVI3d mettant en oeuvre notre système de reconnaissance de gestes

Parallèlement à l’application de démonstration multimodale, la CAO en situation immersive a également évoluée (figure 6). Actuellement on peut interagir via la parole et un Wand (périphérique de type souris 3d) pour créer, manipuler, ou modifier des objets. Nous sommes en train de réintégrer le geste dans cette application mais cette fois ci pour la main dominée (main gauche pour un droitier, droite pour un gaucher). On peut alors manipuler les objets et le plan de travail de la main dominée tout en disposant du Wand (en tant qu’outils de travail) dans la main dominante. On favorise ainsi une interaction naturelle à deux mains pour l’édition en CAO.

A plus long terme, le but est de remplacer totalement l’utilisation de périphérique par des interactions utilisant les modalités naturelles de l’utilisateur. C’est à dire, pour l’application de CAO en Réalité Virtuelle, de substituer le périphérique de pointage dans la main dominante (Wand) par une interaction gestuelle. On aura besoin dans ce cas d’avoir un système capable de gérer les interactions bimanuelles, c’est à dire un système qui sâche détecter les éventuelles synchronisations entre les mains et différencier les gestes à deux mains des gestes à une main.

4 Choix des gestes et des primitives

4.1 Le vocabulaire

Un système de classification statistique ne permet pas de savoir si un geste, fourni en entrée au système, fait partie du vocabulaire appris ou non. Une solution consiste à utiliser des calculs de distance entre le geste reconnu et la classe dans laquelle il été classifié (par exemple la distance géométrique ou de Mahalanobis), ou à utiliser des probabilités sur le geste que l’on vient de reconnaître. Cette solution n’est pas toujours simple à mettre en oeuvre, car il

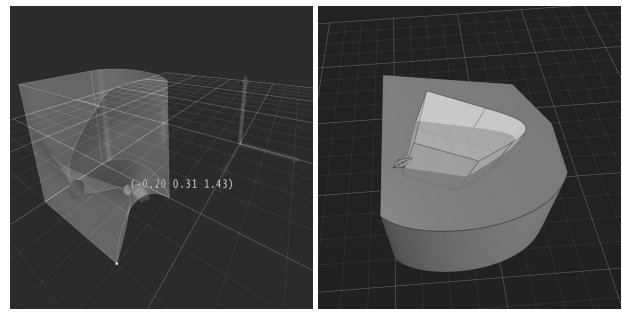


FIG. 6 – Application VRAD (Virtual Reality Aided Design) développée sur la plate-forme EVI3d

faut pouvoir estimer de manière fiable le seuil permettant de décider de la véracité du geste, et parfois les calculs effectués dépendent fortement de l’apprentissage ce qui nuit à la robustesse du système.

Une autre solution consiste à rajouter au vocabulaire de base d’autres gestes qui permettent de couvrir les plages de valeurs que nos gestes de base ne couvrent pas. On ne peut avoir alors de gestes n’appartenant pas au vocabulaire reconnu par le système. Bien évidemment cette solution ne peut être utilisée avec n’importe quel vocabulaire.

Nous avons choisi d’utiliser cette deuxième solution qui convient bien à notre vocabulaire. Aux gestes “saisie” et “désignation”, nous avons donc rajouté les gestes “plat” et “poing” (figure 7) qui permettent de couvrir les valeurs correspondant aux doigts fortement fléchis et très peu fléchis. Les gestes “saisie” et “désignation” couvrent eux les valeurs intermédiaires.



FIG. 7 – “Plat” et “poing”

4.2 Choix des primitives

Une fois que l’on connaît l’ensemble des gestes que l’on veut reconnaître, il va falloir trouver les primitives qui permettent de caractériser et différencier les différents gestes. Une primitive correspond à une caractéristique des gestes que l’on obtient à partir des données capturées.

Nous disposons de deux types de gants numériques pour effectuer la capture des données : un DataGlove 5 de chez 5DT et un CyberGlove 22 de chez Immersion. Le DataGlove 5 ne permet d’avoir comme information que la flexion globale de chaque doigt. Donc pour ce gant la problématique du choix des primitives ne se pose pas

vraiment, il nous manque même des données pour pouvoir distinguer certains mouvements des doigts. Par contre le cyberglove couvre tous les degrés de liberté de la main : les articulations des doigts, les abductions, la paume et le poignet, il faut alors déterminer quelles sont les informations nécessaires et suffisantes.

On va donc maintenant examiner divers choix de primitives et ce qui a conduit à faire ces choix.

- La solution la plus simple consiste à prendre comme primitives tous les capteurs du gant numérique. Ce qui représente 22 primitives pour le modèle Cyberglove. Un premier problème apparaît avec cet ensemble de primitives. Du fait du grand nombre de primitives, il faut au minimum une dizaine d'exemples pour faire l'apprentissage du système de reconnaissance. Alors que pour les trois autres ensembles de primitives décrits ci-dessous, 4 exemples suffisent. D'autre part il apparaît clairement que de nombreuses primitives sont inutiles.
- Le deuxième ensemble de primitives a été choisi en fonction des caractéristiques des gestes. Lors du geste de désignation, on a l'index tendu, donc on choisit d'utiliser les trois articulations de l'index comme primitives. Pour la saisie il faut au moins une information par doigt, de plus on a plutôt tendance à saisir principalement à l'aide du pouce, de l'index et du majeur ; on va donc utiliser les 3 articulations du pouce, les trois articulations du majeur et les articulations de la première phalange pour l'annulaire et l'auriculaire. Cela nous fait donc un total de 11 primitives
- Le troisième ensemble de primitives a été choisi pour vérifier que l'on ne pouvait pas réduire le nombre de primitives au niveau du pouce, de l'index et du majeur, on va donc choisir uniquement l'articulation de la première phalange pour chaque doigt. On obtient alors un total de 5 primitives qui sont relativement similaires à ce que l'on aurait eut comme données avec un gant DataGlove5.
- Pour le quatrième ensemble de primitives, on est reparti du deuxième ensemble et on a analysé plus en détail les variations des différentes articulations et cela plus particulièrement pour le geste de désignation. En effet c'est le seul geste pour lequel on ne fléchit pas tous les doigts en même temps. Dans la figure 8 on peut voir les courbes des flexions des doigts pour les gestes "poing" et "désignation". Les courbes du haut correspondent aux trois articulations du pouce, de l'index et du majeur. Les courbes du bas correspondent aux trois articulations du pouce, l'abduction pouce-index, les articulations de la première et de la deuxième phalange de l'index et la première phalange du majeur.

On peut voir, lorsque l'on n'utilise pas les abductions, que seules les valeurs des articulations de l'index permettent de distinguer efficacement les deux gestes. Or les valeurs de l'index pour une "désignation" sont proches de celles d'un geste "plat" et d'une "saisie large". Par contre l'abduction index-majeur permet de bien

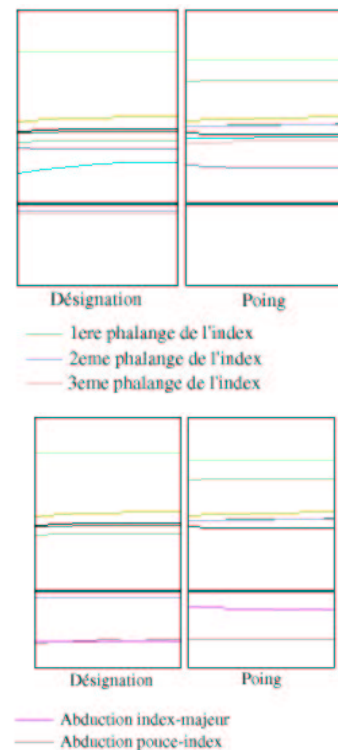


FIG. 8 – Comparaison de l'abduction aux articulations de l'index

distinguer la "désignation" du geste "poing", mais aussi de bien la distinguer des gestes "plat" et "saisie".

C'est pourquoi pour le quatrième ensemble on aura les 11 primitives suivantes : les trois articulations du pouce, les deux première phalanges de l'index, les deux premières phalanges de l'index et du majeur, l'abduction pouce-index, l'abduction index-majeur et les premières phalanges de l'annulaire et de l'auriculaire.

4.3 Expérimentation des primitives

Pour comparer ces différents ensembles de primitives nous avons fait quatre apprentissages avec les mêmes exemples (entre 10 et 12 pour chaque classe de geste) ; chacun des apprentissage correspondant à un ensemble de primitives. Nous avons ensuite fait fonctionner simultanément quatre systèmes de reconnaissance de gestes à partir de ces apprentissages pour comparer les gestes reconnus au geste produit. Pour chaque classe de geste, une douzaine de gestes ont été produits. Deux ou trois gestes parmi les douze étaient des cas "limites" permettant de tester la robustesse des ensemble de primitives. Le tableau 1 donne les résultats obtenus.

Le résultat obtenu pour le premier ensemble de primitives est très surprenant. Même si l'on s'attendait à ce que les résultats obtenus pour cet ensemble de primitives ne soient pas optimum, ils sont vraiment très bas. Une des causes à cela peut-être dû au fait que cet ensemble de primitives a besoin de plus d'exemple pour l'apprentissage que les

	Nb. gestes corrects	% gestes corrects
Ensemble 1	34	70,8
Ensemble 2	44	91,7
Ensemble 3	39	81,25
Ensemble 4	47	97,9

TAB. 1 – Taux de reconnaissance pour les différents ensembles de primitives

autres. Il est alors probable que le nombre d'exemples ait été insuffisant dans cette expérimentation.

L'ensemble 3 a un score plutôt satisfaisant étant donnée le nombre de primitives utilisé. Toutefois il est largement en deça des scores des ensembles 2 et 4. On remarque dans ces résultats que le choix des abductions pour l'ensemble 4 par rapport à l'ensemble 2 a été judicieux car il permet d'avoir une légère amélioration dans le taux de reconnaissance.

On constate, suite à cette expérimentation, que le choix des primitives n'est pas à prendre à la légère, car il peut influencer de manière importante le taux de reconnaissance d'un système statistique. D'autre part on constate qu'un grand nombre de primitives n'est pas forcément synonymes d'une bonne classification des gestes, car du fait du grand nombre de primitives on est obligé de faire un apprentissage plus long et certaines primitives ont tendance à être redondantes.

5 Gestes statiques et dynamiques

Les gestes que nous avons utilisés jusqu'à maintenant sont des postures statiques.

Le choix d'utiliser des postures statiques peut sembler simpliste et un peu éloigné des systèmes actuels qui permettent la gestion de gestes dynamiques et de la coarticulation. Toutefois ce choix a été motivé entre autre par le fait que le vocabulaire est suffisamment réduit et simple pour ne pas avoir besoin de gérer les problèmes de coarticulations. Les postures statiques permettent d'avoir une interprétation quasi instantanée d'un geste contrairement aux gestes dynamiques qui ont en général besoin d'être interprétés sur plusieurs trames d'affilées. Mais surtout les postures statiques permettent un apprentissage très facile. En effet on n'a pas besoin de faire de segmentation pour de tels gestes, le geste correspond à la trame courante. Un utilisateur quelconque peut donc effectuer l'apprentissage du système en donnant tout simplement successivement les différentes postures du vocabulaire. Il n'a pas besoin de se préoccuper du fait qu'un geste commence à telle trame et fini à telle autre. Ce choix nous a permis également de disposer très rapidement d'un premier système de reconnaissance de geste, au sein de la plate-forme de Réalité Virtuelle, avec un vocabulaire générique, c'est à dire indépendant des applications.

Bien évidemment les postures statiques ont des contreparties, et en particulier elles ont le désavantage de ne représenter qu'une partie du geste réellement effectué. Et donc parfois une toute petite différence de la flexion des doigts peut entraîner le geste dans une autre classe du système de reconnaissance. Cela est particulièrement dérangeant lorsque l'on manipule un objet que l'on "tient" en main et que tout à coup on "lâche" l'objet.

Dans le cadre de l'application de CAO immersive (section 3.4, figure 6), nous sommes en train de considérer le geste de saisie sous un autre angle. Plutôt que d'utiliser une posture statique (cf figure 2) on va utiliser le moment où l'on ferme les doigts pour débiter une saisie et le moment où l'on ouvre les doigts pour lâcher. L'avantage de ce type de gestes est que l'on n'a pas besoin de se préoccuper de la manière dont l'objet est tenu (nombre de doigts utilisés, fermeture de la main ...) entre la fermeture et l'ouverture des doigts.

Comme primitives nous avons choisi d'utiliser la variation de la flexion des doigts au cours du temps. On se ramène ainsi un peu à un geste statique, dans le sens où c'est la variation angulaire qui cette fois ci est statique. Les premiers tests effectués avec trois gestes (fermeture, ouverture et statique) sont plutôt encourageant. Actuellement nous essayons de combiner des postures statiques (désignation, plat...) aux gestes "fermeture" et "ouverture".

Nous nous intéressons par ailleurs aux gestes qui servent à décrire des formes à l'aide des mouvements et des formes des mains. Nous sommes en train de mettre au point un système de reconnaissance des différentes trajectoires de la main (droite, arc, cercle ...), ce qui nous permettra d'ajouter de nouveaux types d'interactions à nos applications.

6 Conclusion - Perspectives

On voit donc qu'avec un système de reconnaissance assez simple, des interactions gestuelles génériques permettant de manipuler des objets dans un monde virtuel sont possibles. De plus, lorsque les primitives sont bien choisies, on arrive à avoir de bon taux de reconnaissance (il faut nuancer ces résultats par le fait que le vocabulaire est très limité).

Mais le système de reconnaissance présenté ici ne se résume pas aux gestes de saisie et de désignation. Le vocabulaire de base peut-être agrandi pour permettre d'autres types d'interactions que la manipulation d'objets. Nous pensons en particulier étendre notre vocabulaire gestuel pour permettre la création des primitives standard (sphère, cube, cylindre ...) et de formes libres.

Un autre point fort de ce système de reconnaissance est la modularité due à l'architecture de la plate-forme EVI3d qui permet une intégration très facile du système dans une application de réalité virtuelle. En plus de cette facilité du côté programmeur, l'architecture logicielle du client et de la plate-forme EVI3d permettent de passer d'un vocabulaire gestuel à un autre, ou de changer de

périphérique, ce qui donne un certain confort d'utilisation. Toutefois tout n'est pas idéal du côté utilisateur, on voit apparaître un problème déjà rencontré dans le domaine de la reconnaissance de la parole : la variation inter-utilisateurs. En effet un apprentissage gestuel correspond à une personne bien précise du fait que chaque personne effectue les gestes d'une manière propre à elle. Cela est dû au fait que la différence morphologique des mains entraîne des positionnements différents des capteurs des gants, et que les amplitudes angulaires des articulations varient d'une personne à une autre.

Ceci est un problème que nous avons déjà commencé à aborder partiellement avec la calibration de l'amplitude des articulations et le remplacement de certains gestes par d'autres plus "génériques"; par exemple : remplacer la posture de saisie (qui peut être différente d'une personne à une autre) par les gestes de fermeture des doigts et d'ouverture des doigts (ce que fait tout le monde lorsque l'on veut saisir quelque chose).

Nous voulons dans le futur également explorer l'interaction bi-manuelle en Réalité virtuelle, soit dans le domaine de la CAO pour avoir une interaction directe avec les objets, soit dans le domaine des interactions à distance en utilisant des travaux que nous avons effectués dans le domaine de la Langue des Signes [2, 3].

Références

- [1] O. Bimber, *Gesture-Based Interaction in Virtual Environments*, Thèse, Université de Darmstad, 1999.
- [2] B. Bossard, *Problèmes posés par la reconnaissance de gestes en Langue des Signes*, Récital 2002, Nancy, Juin 2002.
- [3] B. Bossard, A. Braffort, M. Jardino, *Some Issues in Sign Language Processing*, Gesture Workshop 2003, Gènes, Avril 2003.
- [4] P. Bourdot, D. Touraine, *A Hierarchical Server to Manage Multimodal and Collaborative Interactions within Virtual Environments*, International Conference on Applied Informatics, Innsbruck, Février 2002.
- [5] P. Bourdot, M. Krus, R. Gherbi, *Cooperation between reactive 3D objects and a multimodal X window kernel for CAD*, LNAI 1374, Multimodal Human-Computer Communication : Systems, Techniques and Experiments, Springer, Janvier 1998.
- [6] C. Cruz-Neira, D.J. Sandin, T. DeFanti, *Surround-screen projection-based virtual reality : The design and implementation of the CAVE*, SIGGRAPH'93, Août 1993.
- [7] S. Gibet, A. Braffort, C. Collet, F. Forest, R. Gherbi, T. Lebourque, *Gesture in Human-Machine Communication : capture, analysis-synthesis, recognition, semantics*, Gesture Workshop 1996, York, Mars 1996.
- [8] W. Kruger, C. Bohn, B. Frohlich, H. Schuth, W. Strauss, G. Wesche, *The Responsive Workbench : A Virtual Work Environment*, IEEE Computer, Vol. 28, Num. 7, 1995.
- [9] M. Latoschik, I. Wachsmuth, *Exploiting Distant Pointing Gestures for Object Selection in a Virtual Environment*, Gesture Workshop 1997, Bielefeld, Septembre 1997.
- [10] M. R. Mine, *Virtual Environment Interaction Techniques*, Rapport technique, Université de Caroline du Nord, Mai 1995.
- [11] E. Pratini, *New Approaches to 3D Gestural Modeling - the 3D SketchMaker Project*, ECAADE, 2001.
- [12] D. Rubine, *The Automatic Recognition of Gestures*, Thèse, Carnegie-Mellon, 1991.
- [13] T. Sowa, I. Wachsmuth, *Interpretation of Shape-Related Iconic Gestures in Virtual Environments* Gesture Workshop 2001, Londres, Avril 2001.
- [14] D. Touraine, P. Bourdot, Y. Bellik, L. Bolot, *A framework to manage multimodal fusion of events for advanced interactions within Virtual Environments*, Eighth Eurographics Workshop on Virtual Environments, Barcelona, Mai 2002.