RECONNAISSANCE DE GESTES DE LA MAIN EN ENVIRONNEMENT RÉEL

Thomas Baudel
L.R.I. - CNRS URA 410
Bâtiment 490,
Université de Paris-Sud
91405 Orsay Cedex - FRANCE
33+ (1) 69 41 69 10
thomas@lri.fr

Annelies Braffort
L.I.M.S.I. - CNRS
Bâtiment 508,
Université de Paris-Sud
91405 Orsay Cedex - FRANCE
33+ (1) 69 85 81 10
braffort@limsi.fr

RÉSUMÉ

Nous proposons une application de navigation dans un système Hypertexte au moyen de gestes de la main. Notre objectif est de montrer que de telles interfaces gestuelles peuvent s'insérer dans des applications courantes, sans faire appel aux techniques des réalités virtuelles. Nous nous attachons essentiellement à fournir une technique qui nous permette, au moyen de règles simples, de détecter efficacement *l'intention* de l'utilisateur de s'adresser au système, afin d'éviter le "*syndrôme d'immersion*", c'est à dire l'impossibilité pour l'utilisateur d'agir et de communiquer efficacement dans le monde réel lorsque l'ordinateur capte ses mouvements. Cette application a été testée avec succés dans des conditions réelles d'utilisation. Ceci nous permet d'envisager d'autres applications du modèle sous-jacent, entre autres pour la définition d'interfaces multi-modales guidées par le geste.

ABSTRACT

We propose a browser for an Hypertext system using hand gesture recognition. Its purpose is to show that such interfaces can be used in common applications, without the drawbacks of virtual realities techniques. Our main objective is to avoid the "*immersion syndrome*" that affects usual hand gesture recognition techniques: when the system captures the user's gestures, he or she may not be able to interact anymore in the real world. We designed a simple interaction technique that is able to detect the *intention* of the user to issue a command. This application was succesfully tested in a real setting. This allows us to suggest other applications of the underlying interaction model, such as multi-modal interfaces using hand gestures.

MOTS CLÉS

Techniques d'interaction, Interfaces gestuelles, Reconnaissance des gestes de la main, Réalité Informatisée, Interfaces Multi-Modales, Interfaces de Télécommande.

KEYWORDS

Interaction Techniques, Gestural Interfaces, Hand Gesture Recognition, Computerized Reality, Multi-Modal Interfaces, Remote Controlled Interfaces.

1. INTRODUCTION

1.1. Objectifs des Interfaces Gestuelles

Les interfaces gestuelles se proposent d'améliorer l'interaction homme-machine en utilisant la reconnaissance des gestes de l'utilisateur pour entrer des commandes. Par geste, on entend tout mouvement *continu* induit par un opérateur sur un périphérique et analysé par la machine. On distingue deux types d'interfaces gestuelles: soit le système analyse la trace laissée par le geste (par exemple dans le cas de la reconnaissance de l'écriture, ou de signes tracés au moyen d'une souris ou d'une tablette), auquel cas on parlera d'interfaces gestuelles *instrumentales*; soit il interprète le geste lui même, par exemple au moyen d'un gant numérique ou d'une caméra vidéo, auquel cas nous parlerons d'interfaces gestuelles *pures*. L'utilisateur d'une interface gestuelle peut s'attendre à bénéficier de trois améliorations majeures:

• *Interaction plus naturelle* : La communication gestuelle avec une machine fait appel aux moyens de communication humains naturels. Certains signes, plus ou moins universels, sont utilisés instinctivement dans certaines situation : confirmation, désapprobation, expression d'une envergure, ou de "passage à la suite". De plus l'interaction gestuelle se rapproche de la manipulation directe des objets dans le monde réel : il est possible avec une interface gestuelle de prendre explicitement des objets, au lieu d'avoir à se satisfaire d'une métaphore de préhension symbolisée par l'enfoncement du bouton d'une souris.

• *Interaction plus concise et plus efficace* : l'aspect dynamique du geste est porteur d'informations supplémentaires ; l'utilisateur peut exprimer en une seule action une commande, et, par des variations dans l'aspect dynamique, préciser des paramètres tels que la portée de la commande ou les objets la concernant. La prise en compte de la dynamique gestuelle permet donc d'élargir la bande passante du canal de communication Homme->Machine.

• *Interaction directe* : Avec une interface gestuelle pure, la machine n'analyse pas les modifications induites sur un périphérique, mais capte le mouvement même de la main : c'est celle-ci qui devient le périphérique, éliminant la nécéssité de prise en compte cognitive d'un dispositif externe à l'usager. De plus, il devient possible en théorie d'émuler d'autres périphériques tels que souris ou clavier.

On distingue trois types d'interfaces gestuelles pures:

• *Interfaces des réalités virtuelles*, basées sur la manipulation directe d'objets représentés en trois dimensions. L'utilisateur est plongé dans un monde virtuel, fonctionnant comme le monde réel, mais dans lequel l'ordinateur interprète toutes ses actions comme des commandes. Krueger [6] décrit en détail ce type d'interfaces.

• *Interfaces Multi-Modales*: Etymologiquement, ces interfaces visent à interpréter en parallèle plusieurs des canaux de communication humains (voix, geste, direction du regard), pour améliorer l'interaction. La plupart des expériences menées dans ce domaine ont pour objectif de rapprocher l'utilisation d'une machine d'une communication naturelle entre humains [2, 4, 12]. Nous envisageons pour notre part d'autres types d'interaction multi-modale, n'utilisant pas nécéssairement une métaphore de communication humaine [1]. Dans ce cas, ce n'est pas la voix qui constitue la modalité dominante, entretenant le fil du "dialogue", mais le geste.

• *Langages gestuels opératifs*: Ces interfaces utilisent la reconnaissance de langages gestuels spécialisés. K. Murakami et H. Taguchi [8] reconnaissent des mots de la langue des signes japonaise (langue utilisée par les sourds-muets). D. Sturman [11] reconnait un vocabulaire utilisé pour orienter les grues sur des chantiers. Morita et al. [7] utilisent un gant numérique et une baguette pour permettre à un chef d'orchestre de diriger un orchestre synthétique.

L'application que nous proposons entre dans cette dernière catégorie. Nous nous inspirons de l'approche de *réalité informatisée,* définie par Mark Weiser [13]. Cette démarche s'oppose aux réalités virtuelles: nous ne cherchons pas à recréer un univers synthétique pour y plonger l'utilisateur, mais à améliorer l'existant par des techniques simples mais soigneusement évaluées, discrètes et *compatibles* avec notre expérience du monde réel. En étendant le modèle qui sous-tend notre application, nous entendons également appliquer nos résultats à la réalisation d'interfaces multi-modales.

1.2. Limites Actuelles des Interfaces Gestuelles

La recherche en interfaces gestuelles est un domaine actif. Nous disposons de matériel comme le gant numérique VPL [14] et de méthodes de reconnaissance de gestes efficaces. Cependant, les expériences menées actuellement ne dépassent pas le cadre d'applications "jouet", testées en laboratoire, sans considérations de l'insertion de cette application dans un environnement de travail courant.

Une des raisons pour lesquelles, à notre avis, ces systèmes sont difficilement utilisables est le "syndrôme d'immersion" : la machine capte tous les gestes de l'utilisateur ; ce dernier peut vouloir ou avoir besoin d'agir dans le monde réel alors que la machine interprète ses gestes, contre sa volonté. Un des problèmes important à résoudre avec les interfaces gestuelles pures (comme avec la reconnaissance de la parole), consiste donc à détecter efficacement l'*intention* de l'utilisateur, avant d'interpréter la nature de ses actions. D'autres problèmes doivent également être résolus pour que ces interfaces puissent se répandre : les périphériques capteurs existant sont encore coûteux et inconfortables (l'utilisateur est relié par un cordon à la machine).

Ayant caractérisé certains aspects de l'interaction gestuelle [1], nous avons réalisé une application prenant en compte les avantages et les limites du geste. Son objectif n'est pas tant d'offrir des fonctionnalités inusitées et des algorithmes performants que de s'intégrer de façon *transparente* dans un environnement de travail habituel. Nous ne prétendons pas pour autant fournir un produit effectivement commercialisable (le matériel utilisé ne pouvant être répandu), mais tout au moins nous assurer de son "utilisabilité" dans un contexte réel et concret. Notre parti-pris a été de choisir des solutions simples mais basées sur une étude de la nature de l'interaction gestuelle et des conséquences induites sur un système de reconnaissance. Notamment, nous n'avons pas utilisé dans notre interface de modélisation de l'usager et de l'application : ces modèles sont expérimentaux et nécéssitent des méthodes logicielles lourdes. De plus, ils engendrent une dépendance du modèle d'interaction vis à vis de l'application qui ne nous a pas été nécéssaire.

2. EXEMPLE D'INTERFACE GESTUELLE PURE

L'application que nous avons réalisé utilise pour l'instant le gant numérique VPL Dataglove et fonctionne sur Apple Macintosh. Elle permet la navigation dans un système Hypertexte, en l'occurence HyperCard. Au moyen de 16 commandes gestuelles, l'utilisateur peut accéder rapidement à toute page d'un Hypertexte, passer en mode défilement automatique, marquer des pages pour y revenir plus tard instantanément (empilement), désigner des objets de l'écran pour les mettre en inversion vidéo, etc.

Nous utilisons ce système pour assister la présentation interactive (Figure 1). Un rétro-projecteur reproduit l'écran de l'ordinateur sur un panneau de projection. Le présentateur (notre utilisateur) porte le gant numérique et se sert de commandes gestuelles pour diriger sa conférence. Bien que le système capte tous les gestes de l'utilisateur, celui-ci reste libre d'interagir dans le monde réel, pour communiquer ou utiliser d'autres appareils. En effet, les commandes sont interprétées par le système seulement si :

• l'utilisateur désigne explicitement avec la main une "zone active", en l'occurence la zone de projection murale.

• il adopte certaines postures de la main reconnues comme des configurations de début de commande gestuelle. L'utilisateur peut donc désigner des portions de l'écran sans pour autant déclencher de commande.

Les types de gestes reconnus sont choisis de façon à correspondre à des gestes usuels, facilitant ainsi l'apprentissage et l'utilisation du système. Le jeu de commandes évite soigneusement les postures correspondant à un état de repos du bras et de la main : ainsi, l'utilisateur ne déclenchera pas de commande involontairement. Toute commande gestuelle reconnue ne peut résulter que d'un "effort" intentionel de l'utilisateur.

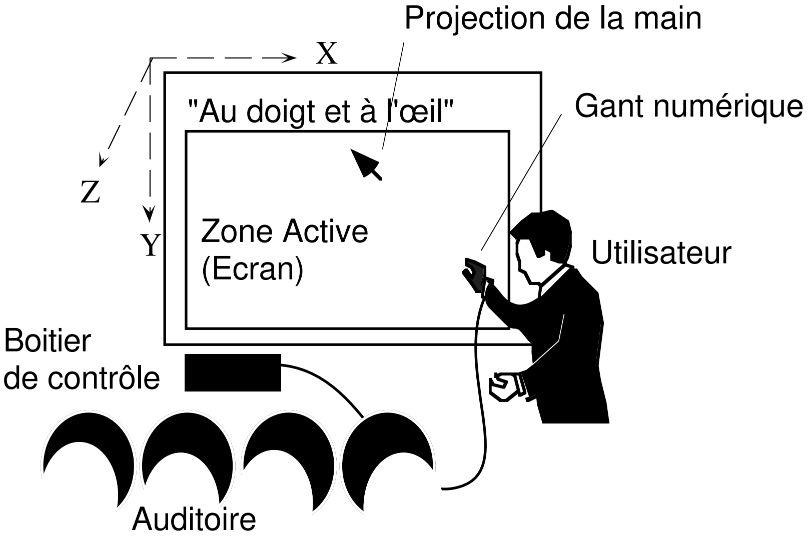

Figure 1 : Système de présentation assistée par ordinateur utilisant la reconnaissance de gestes de la main.

Cette application apporte quelques avantages par rapport aux outils classiques d'aide à la présentation : on constate à l'utilisation que la plupart des commandes gestuelles sont éxécutées presque inconsciemment ; le présentateur accompagne sa présentation de gestes naturellement. Lorsqu'il a une certaine pratique de l'application, il se tourne légérement et instinctivement vers l'écran, et les gestes de passage à l'écran suivant, retour au sommaire, etc, sont effectués sans que le déclenchement des commandes n'engendre d'effort cognitif perceptible. Tandis qu'avec les dispositifs classiques la présentation est en quelque sorte "hachée" par le passage d'écran en écran, l'utilisation des commandes gestuelles permet de joindre le geste à la parole, d'accompagner la présentation orale pour mettre en valeur son articulation.

Le système permet au conférencier d'adapter librement sa présentation à l'auditoire, revenir à des passages précis pour répondre aux questions, sauter des passages non adaptés au public… Certaines fonctions permettent d'ajouter un caractère multi-média au système : le défilement automatique permet de déclencher des animations, des commandes peuvent être ajoutées pour contrôler d'autres dispositifs de présentation : enregistrements audio ou vidéo…

Dans la suite de notre exposé, nous présentons les méthodes qui ont permis la réalisation de cette interface gestuelle, les choix faits et les limites que nous avons rencontrées.

3. CONCEPTION DU JEU DE GESTES

L'ergonomie et la facilité d'utilisation dépendent étroitement du choix des commandes gestuelles. Notre application s'est avérée utilisable car nous avons clairement défini la forme des commandes adressables au système et évité les gestes susceptibles de se produire dans une interaction avec l'environnement réel.

3.1. Limitations du Paradigme

Nous avons tout d'abord observé un certain nombres de limitations intrinsèques au paradigme de désignation utilisé et vraisemblablement aux interfaces gestuelles pures :

- *Fatigue* : Les commandes gestuelles font intervenir tous les muscles du bras, et celui-ci ne dispose pas de point d'appui : elles doivent impérativement être brèves pour ne pas engendrer à terme de fatigue.
- *Imprécision* : Le système ne capte pas les gestes avec une fidélité parfaite. De plus, l'utilisateur ne peut pas désigner à distance avec suffisamment de précision pour utiliser, par exemple, des programmes de C.A.O. ou de dessin. Avec notre système, la meilleure résolution utilisable est d'environ 10 pixels, ce qui permet de manipuler des objets symboliques tels que fenêtres et icones, mais pas d'effectuer des tâches courantes telles que sélectionner du texte ou dessiner des objets.
- *Capacités d'abstraction limitées* : Certains types de commandes ne peuvent pas être traduites au moyen de commandes gestuelles intuitives : changer une police de caractère, enregistrer un fichier…

Afin d'éviter les difficultés engendrées par ces limitations, nous avons adopté un parti pris défini par Ben Shneidermann dans sa définition de la manipulation directe [10] : offrir un jeu de commandes basé sur des actions rapides, incrémentales et réversibles. Actions rapides, afin d'éviter la fatigue ; Actions incrémentales, c'est à dire permettant une construction progressive du but à atteindre, afin de guider l'utilisateur et lui permettre de mieux conceptualiser le fonctionnement de l'interface ; Actions réversibles, afin de pallier à l'imprécision des gestes, aux erreurs de reconnaissance ou à l'éventuelle éxécution involontaire de commandes. Pour contourner la limitation de capacités d'abstraction, les interfaces à manipulation directe proposent l'emploi de menus répértoriant les commandes sans équivalent métaphorique clair dans le modèle d'interaction proposé. Dans notre cas, l'utilisation de la reconnaissance vocale en complément au geste nous parait tout indiquée pour répondre mieux à ce problème : elle permet de préserver l'aspect direct de l'interaction. En attendant une extension multi-modale de notre modèle d'interaction, celui-ci s'avère adapté à la navigation dans un système complexe, la manipulation d'objets ou la télécommande d'appareils informatisés.

3.2. Notion de "Posture Tendue"

Un des points les plus importants de la conception de jeux de commandes consiste à fournir des commandes à la fois intuitives, et qui aient peu de chances d'être réalisées par inadvertance ou lors d'une interaction dans le monde réel. Pour cela, nous utilisons la notion de "posture tendue". Les postures de début de gestes sont choisies de telle sorte qu'elles requièrent une certaine tension des muscles de l'utilisateur, par exemple en serrant le poing ou en tournant la paume vers le haut. Cette tension peut être brève et faible, et donc ne génère pas de fatigue : notre algorithme ne requiert pas de seuil temporel minimal de maintien d'une posture pour reconnaitre celle-ci. Cette tension a l'avantage de rendre explicite l'intention de réaliser un geste qui doit être suivi d'effet : habituellement, les muscles de la main et du bras sont relachés. Cette tension est également présente, comme le décrit W. Buxton [3], lorsqu'un utilisateur maintient le bouton d'une souris enfoncé, lors du choix dans un menu ou d'un déplacement d'objet. Dans ce cas, la succession des moments de tension et de relâchement contribue à renforcer la structure perçue du dialogue homme-machine, et résulte en une interface plus intuitive et facile à contrôler. Inversement, les postures de fin de geste correspondent le plus souvent à un relâchement musculaire. Nous avons observé que la plupart du temps, l'utilisateur effectue ses commandes en "lançant" sa main en direction de la zone active, adoptant la configuration de début durant le mouvement, puis rabaisse son bras, ce qui correspond à la phase de relâchement.

Cette correspondance entre la tension musculaire utilisée instinctivement, et l'expression des commandes nous offre un moyen simple de détecter efficacement l'*intention* de l'utilisateur. Elle correspond du point de vue vocal aux variations d'intonations qui distinguent l'ordre explicite du discours habituel. Ces intonations vocales sont cependant difficiles à détecter dans les systèmes actuels de reconnaissance de la parole. Elles ne correspondent pas non plus tout à fait à nos habitudes d'utilisation du canal vocal. C'est pourquoi le canal gestuel nous apparait plus approprié dans de nombreux cas pour constituer le fil directeur de l'interaction, la parole permettant de préciser efficacement la *nature* de l'action envisagée, lorsqu'elle s'avère un média efficace.

4. NOTATION DES GESTES

Un inconvénient des interfaces gestuelles ne nous est apparu qu'à l'usage de notre application : la difficulté de leur documentation. Les notations actuelles de langages de signes sont complexes et incomplètes, car les langues qu'elles représentent n'ont pas de formalisation sous-jacente. Dans notre cas, la forme des commandes est clairement délimitée par notre paradigme d'interaction et le domaine de l'application. Nous avons donc pu réaliser une notation iconique simple, qui se veut accessible rapidement à des utilisateurs non spécialistes. Celle-ci permet la représentation de l'aspect dynamique de notre interface gestuelle.

Une commande gestuelle est représentée par un bloc de trois icones, représentant respectivement la posture de début de geste, la dynamique du geste et la configuration de fin de geste qui déclenche l'éxécution de la commande. Cette dernière est optionnelle, car la commande est éxécutée dès lors que l'utilisateur, ayant commencé une commande, cesse de désigner la zone active (en abaissant son bras, par exemple).

Les icones de début (resp. de fin) de geste représentent la posture qui doit être adoptée par la main pour que le système débute (resp. termine) la reconnaissance d'un geste. Cette posture est schématisée. En effet, afin de permettre à l'utilisateur d'obtenir aisément une posture recherchée, les dimensions gestuelles utilisées pour l'analyse des postures sont discrétisées. Au lieu de considérer l'angle des articulations et l'orientation du poignet comme des valeurs variant continûment, seules 4 valeurs de flexion pour chaque doigt (2 pour le pouce) et 7 valeurs d'orientation du poignet sont prises en compte. Malgré cette restriction, il est possible de représenter 3584 postures, dont au moins 300 peuvent être obtenues avec quelque effort, et de 30 à 80 sont effectivement utilisables. Ces valeurs sont notées par les symboles suivants (Figures 2 et 3):

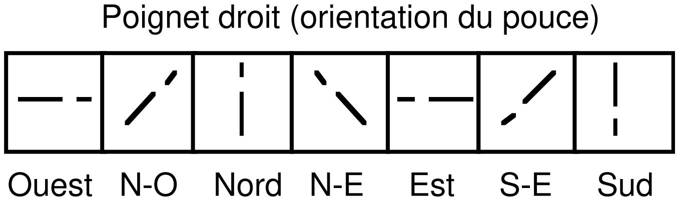

Figure 2 : notation de l'orientation du poignet.

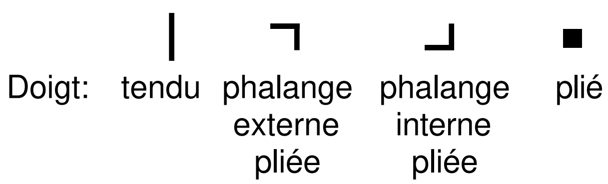

Figure 3 : notation des courbures de doigts.

L'icone centrale représente la dynamique du geste. Une flèche indique le mouvement général de la projection de la main sur la zone active. Cette flèche représente une trajectoire moyenne. Des variations d'amplitude ou de trajet sont admises par l'algorithme de reconnaissance. Le plus souvent ces variations sont utilisés pour préciser des paramètres de la commande : par exemple l'étendue de la zone à sélectionner, dans le cas de la commande "sélection", ou les objets servant de paramètres, dans le cas de la commande "aller chapitre…". Si la commande réclame explicitement ces paramètres, un ou plusieurs cercles dans l'icone indiquent quels paramètres sont pris en compte. La dynamique des doigts est représentée lorsqu'elle n'est pas implicitement définie par la différence entre la configuration de début et celle de fin: une forme en "V" indique une flexion/extension d'un des doigts pendant l'expression de la commande. Enfin, une ligne parallèle au trajet de la main indique une variation de profondeur (pour indiquer par exemple l'action de pousser).

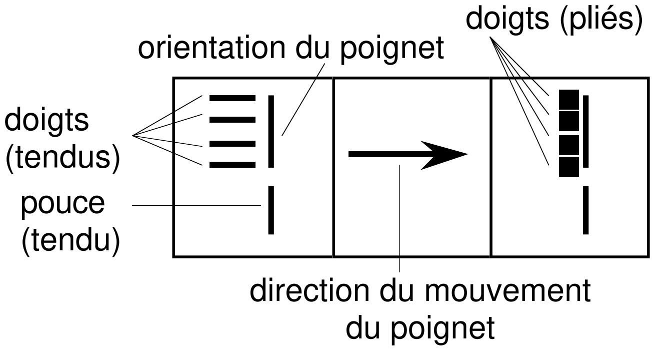

Figure 4 : notation de la commande "Chapitre Suivant".
Cet exemple utilise la main *droite*. La commande est obtenue, lorsque l'utilisateur désigne la zone active, en étendant la main, tournant le pouce vers le bas (paume à droite) et en effectuant un mouvement de balayage de la gauche vers la droite. La commande est reconnue lorsque l'utilisateur referme la main, ou lorsqu'il ne désigne plus la zone active.

Cette notation nous permet de présenter le jeu de commandes complet de notre application dans son état actuel :

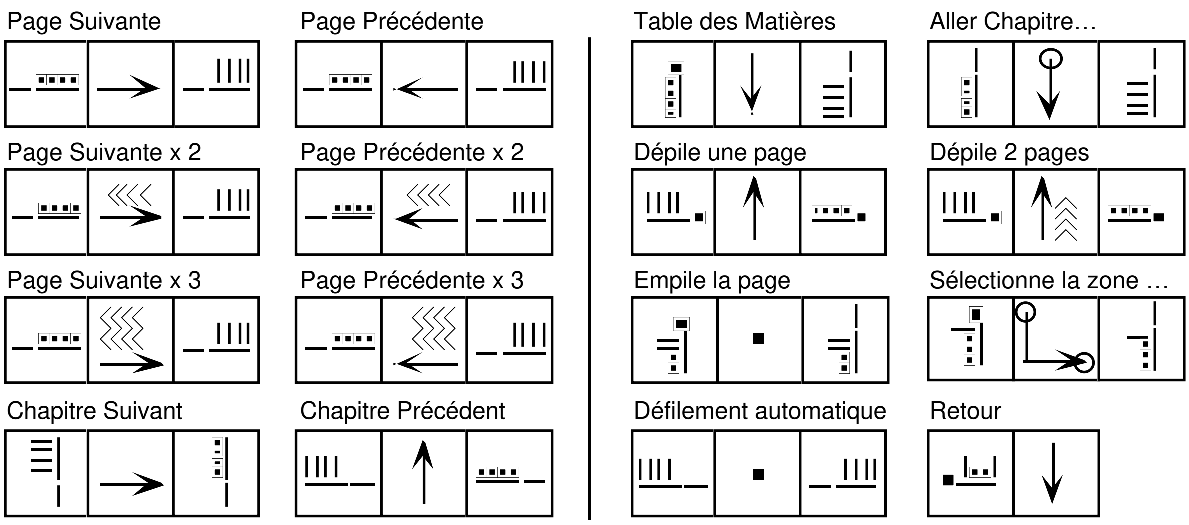

Figure 5 : le jeu de commandes complet de notre application.

5. IMPLANTATION

Notre algorithme de reconnaissance est implanté comme un gestionnaire de périphérique, qui capte les entrées du gant numérique, les analyse, et émet un événement de haut niveau lorsqu'une commande est reconnue. Nous échantillonnons le gant à 60 Hz, et le système fonctionne en temps réel sur Macintosh IIx, sans ralentissement perceptible de l'interaction : le temps de transfert des événements et leur interprétation dans l'application est en fait plus long que le temps d'analyse proprement dit. Le code représente 22 kOctets, et la description du jeu de gestes environ 40 kOctets.

5.1. Détection de l'intention

L'algorithme principal analyse en permanence les échantillons reçus pour déterminer s'ils correspondent à une posture de début ou de fin de geste. Il calcule tout d'abord si la projection de la main sur le plan de l'écran est dans la zone active. Si c'est le cas, il utilise un arbre de codage des postures pour déterminer si l'échantillon courant est une posture de début (figure 6).

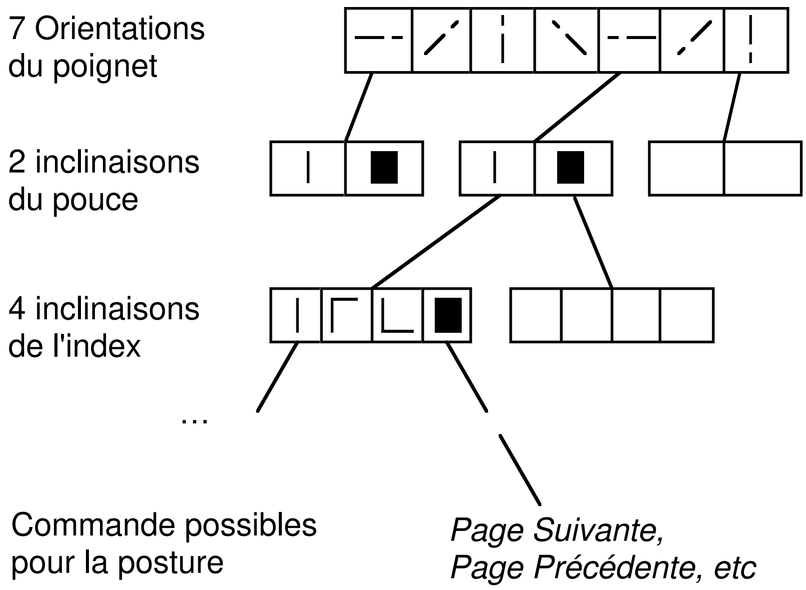

Figure 6 : Arbre de recherche d'une configuration de début de geste.

Il ne faut donc pas plus de 6 comparaisons de pointeurs par échantillon pour déterminer si une posture correspond ou non au début d'un ou plusieurs gestes.

5.2. Analyse de la dynamique du geste

Nous avons étendu l'algorithme défini par D. Rubine [9] pour interpréter la dynamique du geste. Cette méthode a été conçue pour l'analyse de gestes en deux dimensions tracés avec une souris ou sur une tablette. Elle permet d'associer un geste donné à la plus proche classe de gestes d'un ensemble préalablement constitué à partir d'exemples d'apprentissage. Elle consiste à extraire d'un tracé un certain nombre de caractéristiques telles que : l'angle total traversé, la longueur du chemin parcouru, le rectangle englobant le geste (figure 7)… Ces caractéristiques sont comparées (avec une pondération adéquate) avec les représentants des classes déterminées lors de l'apprentissage. La classe associée est celle dont le réprésentant est le plus proche du geste donné. Le calcul des caractéristiques d'un tracé est incrémental, et le temps de classement est proportionnel au nombre de classes. Cet algorithme est donc particulièrement efficace et fonctionne parfaitement sur des machines aussi rudimentaires qu'un Macintosh Plus (Processeur Motorola 68000 à 7,5 MHz).

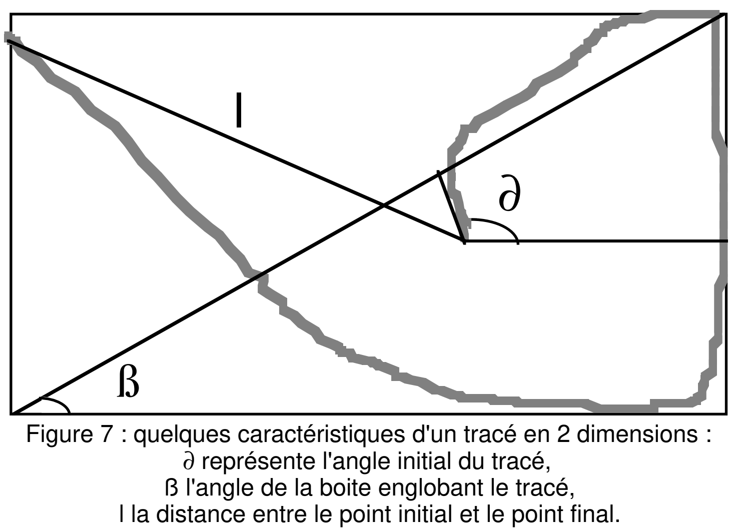

Figure 7 : quelques caractéristiques d'un tracé en 2 dimensions :
∂ représente l'angle initial du tracé,
ß l'angle de la boite englobant le tracé,
l la distance entre le point initial et le point final.

Nous avons adapté cet algorithme à la classification de gestes de la main en ajoutant des caractéristiques : les variations des valeurs de flexion pour chaque doigt et de distance entre la main et l'écran ainsi que les rotations du poignet sont prises en compte pour la classification.

5.3. Apprentissage et Paramétrisation

Afin de pouvoir tester plusieurs jeu de gestes, nous utilisons une application dédiée pour spécifier les jeux de commande gestuels. Les configurations de début et de fin de geste sont saisies au moyen de dialogues interactifs (figure 8).

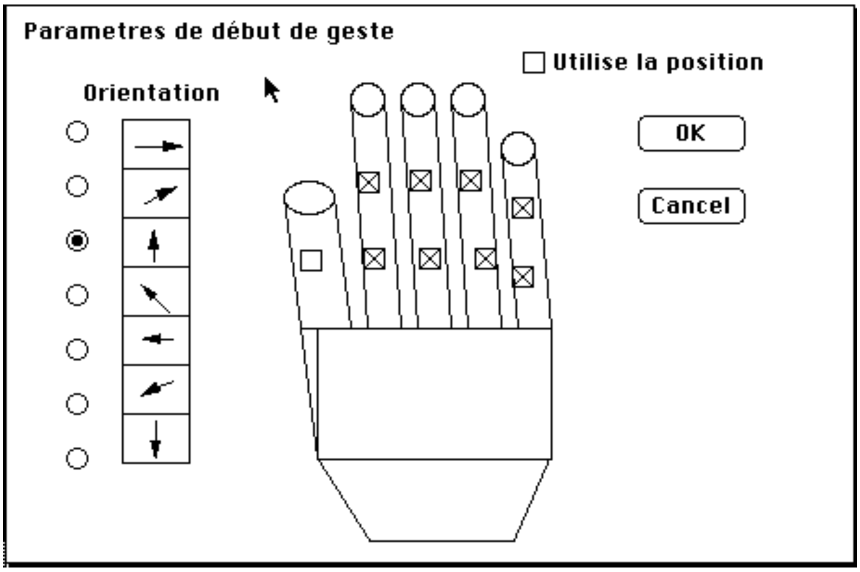

Figure 8 : Spécification interactive des paramètres de début de geste
dans l'application d'apprentissage de "grammaires" gestuelles.

Les caractéristiques dynamiques du geste sont extraites d'exemples de gestes de chaque classe fournis par le concepteur de l'application. Une dizaine d'exemples par commande gestuelle suffit en général à permettre une reconnaissance indépendante de l'utilisateur.

6. EVALUATION DE L'APPLICATION

6.1. Rapidité d'apprentissage

Tout d'abord, nous avons testé la facilité et la vitesse d'apprentissage de l'application pour des néophytes : l'application et la notation des gestes ont été présenté à 10 utilisateurs. Chacun essaye une fois chaque geste, puis le système, adapté pour l'occasion, propose une ou plusieurs séries de 50 gestes. Le taux de reconnaissance est déterminé par le rapport du nombre de gestes correctement reconnus sur le nombre de gestes demandés. Le gant était manifestement trop grand pour deux des testeurs, leur taux de reconnaissance était voisin de 50 %. Les huit autres personnes ont obtenu un taux de reconnaissance moyen de 78 % (variant de 72 à 84 %). Par comparaison, un utilisateur entrainé obtient régulièrement entre 90 et 98 %.

Nous avons remarqué deux types d'erreurs :

• Les gestes qui diffèrent essentiellement par leur aspect dynamique (tels que page suivante x 2 et page suivante x 3) sont souvent confondus, surtout lorsque la variation dynamique concerne le mouvement des doigts. Ceci est en partie dû aux problèmes de calibration du gant et à sa résolution limitée. Mais nous envisageons également des améliorations de notre adaptation de l'algorithme de Rubine pour mieux prendre en compte la dynamique des doigts.

• La plupart des erreurs des débutants sont dûes à une hésitation ou une pause lors de l'expression d'une commande : dans ce cas, l'analyse de la partie dynamique est totalement faussée, et le système peut reconnaitre successivement plusieurs commandes erronées tant que l'utilisateur ne s'est pas rendu compte de ce qu'interprétait le système. Ces erreurs disparaissent avec quelque pratique, lorsque l'utilisateur intègre le principe de l'application et exécute les commandes plus naturellement.

6.2. Utilisation en environnement réel

Afin de valider notre hypothèse principale, notre application a été utilisée lors séminaires internes. L'objectif n'était pas de déterminer un taux de reconnaissance, mais de vérifier l'acceptation de cette application et ses apports. Très peu d'erreurs ont été constatées, en partie parce que les commandes les plus usuelles sont aussi les plus faciles à réaliser et celles qui ont la plus faible ambiguité. Quelques commandes gestuelles ont été réalisées involontairement. Chaque commande gestuelle ayant un inverse, la plupart des erreurs de ce type étaient corrigées immédiatement, en une seule action. Dans de rares cas, l'utilisateur ne s'est pas immédiatement rendu compte qu'une commande avait été déclenchée, et l'erreur a été un peu plus longue à corriger.

Le taux d'erreurs est suffisament faible pour de pas engendrer de désagrément conséquent. L'acceptation du système et son efficacité sont par conséquent trés bonnes. L'interface est facile à utiliser et les effets de présentation obtenus valent le (faible) temps d'apprentissage.

6.3. Limites du paradigme

Malgré les apports de ce type d'interaction gestuelle, nous ne prétendons pas que cette application puisse se répandre à l'heure actuelle. Certains aspects sont rédhibitoires : le coût du gant numérique ne se justifie pas, hors du cadre de la recherche, pour l'utilisation qui en est faite. Son relatif inconfort, et surtout le lien qui relie l'utilisateur à la machine enlèvent également des attraits à l'application. D'autres limites sont corrigibles ou sont moins cruciales : les algorithmes de reconnaissance et la précision de pointage peuvent vraisemblablement être améliorés au prix de peu d'effort. Nous pensons malgré tout avoir obtenu un exemple utilisable de la reconnaissance de gestes de la main, qui peut se généraliser à d'autres applications. Nous nous proposons également de résoudre les problèmes prévenant la généralisation de ce type d'interfaces gestuelles.

7. PERSPECTIVES D’EVOLUTION & CONCLUSION

L'application présentée peut déboucher sur d'autres études, telles que l'amélioration de l'ergonomie des vastes salles de contrôles utilisées pour la télésurveillance, l'intervention sur les marchés boursiers, ou le contrôle d'usines. Dans ces salles, de nombreux agents inspectent et agissent collectivement sur des panneaux de contrôle partagés. Notre paradigme d'interaction parait tout indiqué pour permettre la désignation et la manipulation à distance de ces panneaux, d'autant plus que notre système fonctionne en milieu bruité.

Afin de résoudre l'inconvénient majeur de notre interface, l'inconfort et le coût du gant numérique, nous envisageons d'utiliser la reconnaissance de gestes par caméra vidéo. Des systèmes nécéssitant la pose de marqueurs sur la main sont déjà commercialisés. Fukumoto [5] propose une méthode de reconnaissance des gestes de la main utilisant deux caméras vidéo, sans aucun marqueur. Enfin, comme nous l'avons déjà mentionné, l'utilisation en complément d'un module de reconnaissance vocale devrait nous permettre d'étendre les possibilités de notre modèle.

A partir de considérations sur la nature des interfaces gestuelles pures, nous avons défini, implanté et évalué une application simple mais utilisable dans un contexte réel. Nous avons pu dégager de cette expérience certains principes nous permettant de généraliser la démarche à d'autres applications :

- L'utilisation d'une zone active et de "postures tendues" pour détecter de façon naturelle l'intention de l'utilisateur et éviter le "syndrome d'immersion".
- Une notation des gestes applicable à notre modèle.
- Quelques délimitations du domaine d'application des interfaces gestuelles pures.

Nous comptons à présent dégager de ces leçons un modèle d'interaction définissant plus précisément les choix faits et les principes sous-jacents à notre expérimentation. Nous envisageons ensuite deux améliorations du modèle : l'adaptation aux contraintes de la reconnaissance de gestes par caméra vidéo, et l'extension pour la prise en compte de la parole, afin de définir des interfaces multi-modales guidées par le geste.

REMERCIEMENTS

Ces travaux ont été partiellement financés par le PRC-Communication Homme-Machine. L'application présentée a été réalisée lors d'un séjour du premier auteur au LIMSI. Nous tenons à remercier Joseph Mariani, Françoise Néel, Gérard Sabah et Daniel Teil pour leur accueil et la mise à disposition du matériel nécessaire aux expérimentations. Enfin, Michel Beaudouin-Lafon nous a constamment aidé à préciser notre pensée et à en améliorer l'expression.

BIBLIOGRAPHIE

1. Baudel, T. *Spécificités de l'Interaction Gestuelle dans un Environnement Multi-Modal*. Actes de la conférence IHM'91, AFCET, à paraitre.
2. Bolt, R. *The Human Interface*, Van Nostrand Reinhold, New York, 1984.
3. Buxton, W. *There's More to Interaction than Meets the Eye: Some Issues in Manual Input*. in Norman, D.A. and Draper, S.W. (Eds.), *User Centered System Design*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1986, pp. 319-317.
4. Caelen, J. *Interaction multi-modale dans ICPdraw: expérience et perspectives*. Annales de l'école IHM multi-modale, GRECO-PRC communication Homme-Machine, 1991.
5. Fukumoto, M., Mase, K. and Suenaga, Y. *“Finger-pointer”: A Glove Free Interface,* CHI’92 Conference Proceedings, Poster and Short Talks booklet, page 62, 1992.
6. Krueger, M., *Artificial Reality (2nd ed.)*, Addison-Wesley, Reading, MA, 1990.
7. Morita, H., Hashimoto, S. and Ohteru, S. *A Computer Music System that Follows a Human Conductor*. IEEE Computer, July 1991, pp.44-53.
8. Murakami, K. and Taguchi, H. *Gesture Recognition Using Recurrent Neural Networks*, CHI’91 Conference Proceedings, ACM Press, 1991, pp. 237-242.
9. Rubine, D. *The Automatic Recognition of Gestures*, Ph.D. thesis, Carnegie-Mellon University, 1991.
10. Shneidermann, B. *Direct Manipulation: A Step Beyond Programming Languages*, IEEE Computer, August 1983, pp. 57-69.
11. Sturman, D. *Whole-Hand Input*, Ph.D. thesis, Media Arts & Sciences, Massachusetts Institute of Technology, 1992.
12. Thorisson, K., Koons, D. and Bolt R. *Multi-Modal Natural Dialogue*, CHI'92 Conference Proceedings, ACM Press, 1992, pp. 653-654.
13. Weiser, M. *The Computer for the 21st Century*, Scientific American, September 1991.
14. Zimmerman, T. and Lanier, J. *A Hand Gesture Interface Device*. CHI'87 Conference Proceedings, ACM Press, 1987, pp. 235-240.