Spécificités de l'interaction gestuelle dans un environnement multimodal

Thomas Baudel

Laboratoire de Recherche en Informatique
URA 410 du CNRS
Université de Paris-Sud
91 405 ORSAY Cedex

Position paper destiné à la préparation des journées IHM'91

Résumé

L'interaction gestuelle est l'une des composantes clés d'une véritable interface multi-modale. Celle-ci présente cependant des caractéristiques propres, et est encore mal connue en tant que telle. De fait, nous voudrions distinguer clairement l'analyse statique de positions dans l'espace d'une véritable analyse de la dynamique des gestes et son interprétation par une application. S'il existe à notre connaissance des interfaces multi-modales utilisant le premier type d'analyse, de véritables difficultés se posent pour combiner le second avec d'autres modes d'expression. Ce style d'interaction, bien qu'encore peu répandu, semble pourtant avoir des avantages importants, auxquels il serait dommage de devoir renoncer dans les interfaces multi-modales.

Dans une première partie, nous exposerons les avancées récentes en matière d'analyse et d'interprétation de la dynamique gestuelle, afin de faire ressortir les avantages qu'elles présentent, tant du point de vue de la concision de l'expression que de celui de l'ergonomie. Nous verrons ensuite les méthodes d'implantation existant actuellement et les améliorations que nous comptons y apporter. Enfin, nous étudierons l'intégration de la composante gestuelle dans un environnement multi-modal, avec ses apports et ses difficultés.

Introduction

Comme le signale W. Buxton dans son article "There's more to interaction than meets the eye" [Buxton86], si les progrès enregistrés ces dernières années en matière de présentation des données ont été trés importants, les interfaces utilisateur actuelles ne tirent pas tout le potentiel des capacités d'expression humaine disponibles. Si les machines de traitement de l'information permettent de visualiser et de percevoir des données très complexes, les moyens de manipulation de ces données sont restés très sommaires. Ainsi, sur un ordinateur actuel jugé convivial, les seuls périphériques d'entrée de données courants sont un clavier conçu à l'origine pour limiter la vitesse de frappe, et une souris qui ne sert qu'à désigner des objets ou les déplacer de façon ponctuelle. De plus, le mode d'interaction est souvent éloigné de toute analogie naturelle, et l'aspect métaphorique des interfaces actuelles est souvent limité à la partie visuelle, "sortie" de l'ordinateur, plus qu'à sa partie haptique, l'"entrée" des données.

Deux directions actuelles de la recherche tentent de remédier à ce problème, qui peut être résumé à tâcher d'augmenter la bande passante du canal de communication homme->machine. L'introduction de la multi-modalité permet à l'opérateur d'utiliser plusieurs de ses capacités d'expression, comme le geste, la parole, ou des périphériques annexes tels que claviers, boutons rotatifs ou joysticks. Ces modalités peuvent être employées de façon complémentaire ou alternative, suivant les modes de la communication humaine courante. Les interfaces gestuelles se spécialisent dans l'amélioration de la communication par la capture du geste dans sa généralité, en analysant sa dynamique et ses variations pour en extraire des informations pertinentes à l'application. Ces dernières peuvent figurer comme une des composantes du premier thème de recherche évoqué. Cependant, la complexité de l'analyse gestuelle, ainsi que, nous allons le voir, sa puissance en tant que modalité indépendante, en justifient une approche détaillée.

Il importe de considérer deux aspects des interfaces gestuelles. Le premier vise à tirer un profit maximum des possibilités des périphériques actuels captant des gestes simples et bidimensionels, tels que les souris ou les tablettes graphiques. Le second cherche à exploiter des périphériques plus sophistiqués tels que les Datagloves, afin de trouver des styles d'interaction à la mesure de leur puissance potentielle. De fait, les mêmes techniques d'analyse et d'interprétation peuvent être utilisées dans les deux cas. Cependant, le premier aspect a pour avantage d'être plus proche des matériels courants et de s'intégrer assez simplement pour étendre les interfaces à manipulation directe actuelles. Ainsi, des sytèmes d'exploitation tel PenPoint de GO Corporation [PenPoint91], commencent la vulgarisation et la commercialisation de ce type d'interfaces. On peut donc profiter d'une approche commune aux deux aspects pour passer sans solution de continuité des sytèmes actuels aux interfaces multi-modales complètes, proches des systèmes de réalité virtuelle, tout en restant dans un contexte concret, proche des besoins et limites des interfaces actuelles.

I Caractéristiques de la modalité gestuelle

On peut distinguer dans les interfaces actuelles deux types de périphériques d'entrée:

- *Périphériques à états discrets*: Un clavier, un lecteur de code barres, ou un système de reconnaissance de mots isolés produisent à des instants donnés une information sur un changement d'état provoqué par l'utilisateur. L'entrée d'information est discrète, ce qui permet de l'analyser au moyen d'automates simples.

- *Périphériques à "entrée continue"*: Une souris, des boutons rotatifs ou un dataglove ne traduisent pas de changement d'état simple du périphérique, mais réalisent un échantillonage du geste qu'a réalisé l'utilisateur en les actionnant. En ce sens, l'entrée est plus complexe mais contient plus d'informations que pour le premier type. L'ampleur du geste, ses paramètres d'accélération et ses variations géométriques peuvent être calculés, avec plus ou moins de détails suivant le périphérique utilisé.

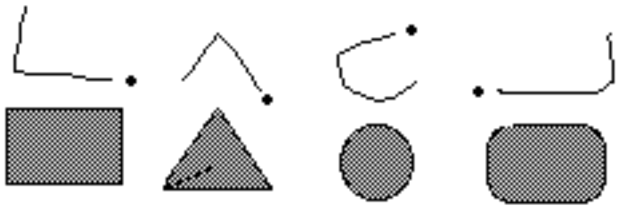
Il est à noter que la plupart du temps, les logiciels ne tirent pas partie de cette information supplémentaire. Ainsi, les systèmes graphiques diffusés actuellement n'utilisent la souris que pour fournir des positions discrètes (clic, multiple ou non) et des déplacements simples ("drag"). On peut donc considérer que les périphériques capables de traduire le geste sont trés largement sous-utilisés actuellement, et se limitent à l'interaction discrète. Nous dénommerons *interface gestuelle* tout type d'interface visant à exploiter de façon optimale les informations sur la dynamique gestuelle fournies par le second type de périphériques. Comme le décrit [Samuels90], l'objectif est double.

Tout d'abord, il s'agit d'augmenter la bande passante de ces périphériques : en interprétant plus d'information d'un simple geste, il devient possible de condenser en une seule action de l'utilisateur une commande et ses paramètres. On obtient ainsi un style d'interaction extrêmement concis et puissant. Nous verrons ci-après des exemples de ce type d'interfaces. L'autre objectif est de se rapprocher de la communication naturelle humaine. Les commandes gestuelles s'apparentent aux langages naturels de gestes, tels que le language des sourds, ou les languages de brokers sur les marchés boursiers. L'étude de ces languages nous montre l'intéret qu'apporte le mode d'expression gestuel aux interfaces : outre leur concision, souvent supérieure à la parole, ils sont utilisables en environnement bruité, ce qui permet d'envisager des interfaces gestuelles efficaces dans un environnement de bureau courant, sans présenter l'inconvénient majeur des interfaces vocales.

La principale raison qui explique ces atouts tient dans la relation naturelle que l'on peut établir entre un geste et sa signification. Ainsi, il est établi qu'il est souvent possible à un entendant de comprendre au moins partiellement le language des sourds sans initiation préalable [Moody83]. Les interfaces gestuelles, par leur iconicité fondamentale, possèdent des caractéristiques uniques en matière d'ergonomie, qui laissent présager facilité d'apprentissage et convivialité.

Plusieurs programmes utilisant les interfaces gestuelles permettent d'illustrer les atouts que nous avons exposés. Par exemple, GEdit est un éditeur de dessin structuré réalisé par G. Kurtenbach [Kurtenbach90]. C'est un prototype destiné à montrer la concision et l'ergonomie des interfaces gestuelles, leur puissance d'expression, ainsi que sa simplicité d'implantation, puisque l'application ne fait que 17 ko sur Macintosh. Le taux de reconnaissance des gestes est supérieur à 98%. Il est à noter qu'au delà d'un tel taux, les chiffres ne signifient plus grand chose : les erreurs d'interpétation sont assimilables à des erreurs de saisie, que tout utilisateur, même confirmé, fait sur un clavier de temps à autre.

Commandes de création d'objet de GEdit. Le point noir désigne la fin du geste.



Commandes d'édition de GEdit:

Déplacer - Copier-Coller - Sélectionner - Retailler - Supprimer

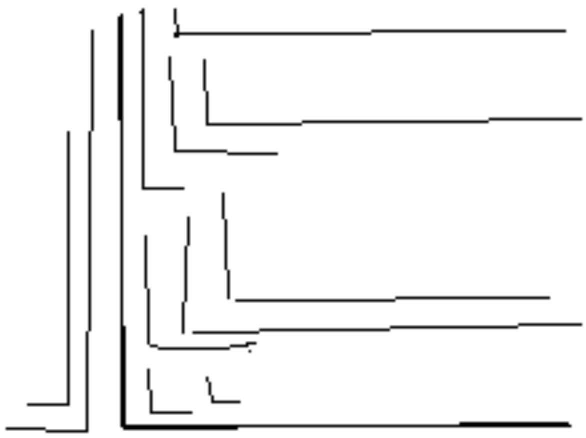
Plusieurs applications du même type ont été réalisées, comme un éditeur de partition musicales, des formateurs de textes, des extensions d'interfaces iconiques… La plus aboutie des interfaces gestuelles actuelles est sans aucun doute le système d'exploitation PenPoint. Ce système propose plusieurs applications dont un shell et une gestion de fichiers. Des bibliothèques permettent à tout nouveau programme d'implanter facilement une interface gestuelle. Destiné à priori aux ordinateurs portables, il peut cependant être utilisé sur une station de travail pour y apporter les avantages de l'interaction gestuelle.

II Méthodes d'analyse des gestes

Si de tels atouts n'ont pas permis aux interfaces gestuelles de se répandre plus rapidement, c'est essentiellement parce que l'interprétation de la dynamique gestuelle n'est pas aisée. Le problème se pose en des termes simples : quelque soit le périphérique, celui-ci transmet des suites temporelles de coordonnées de points, dans un espace à deux, trois dimensions ou plus (pour les datagloves, par exemple). Ces suites de points doivent être regroupées en suites d'éléments lexicaux, chaque léxème correspondant naturellement à une commande de base de l'application. Outre le filtrage des imprécisions apportées par l'utilisateur, il convient de prendre en compte les variations d'amplitude et de tracé pour un même type de geste qui servent à indiquer des paramétres de la commande invoquée. On obtient ainsi une suite de commandes élémentaires avec leurs paramètres, que l'on peut interpréter comme les données d'un périphérique à états discrets et intégrer à un modèle classique de gestion des entrées. Il existe à l'heure actuelle deux approches pour réaliser cette analyse.

Les méthodes de filtrage, inspirées des algorithmes de reconnaissance de l'écriture manuscrite, basées le plus souvent sur la programmation dynamique ou des réseaux neuronaux

[Rubine91] et [Helm91]. Un certain nombre d'exemples de chaque type de gestes sont fournis. Les caractéristiques de chaque type de gestes sont compilées sous la forme de vecteurs de propriétés. L'analyse se fait en comparant chaque vecteur de propriétés avec celui du geste tracé pour en extraire le plus proche. Ces méthodes sont efficaces, mais présentent un inconvénient: Si elles sont efficaces pour la reconnaissance de caractères manuscrits, elles sont peu adaptées à la spécification des paramètres associés à la commande. Lors de la fourniture des exemples à l'algorithme d'entrainement, il nous faut implicitement fournir les paramètres variants dans la forme du geste. Prenons l'exemple d'un programme de dessin comme GEdit: le dessin d'un angle droit est associé à la commande de création d'un rectangle. Le coté vertical détermine la hauteur du rectangle, sa base la largeur. Lors de la spécification du geste, il faut fournir des rectangles de différentes proportions et de différentes tailles, afin que l'algorithme sache reconnaitre la variabilité du léxème gestuel (voir figure çi-dessous). La performance de la reconnaissance est donc directement liée à la pertinence du jeu d'exemples fournis. De plus, il convient de fournir un mécanisme d'extraction des paramétres pertinents de la commande à partir des exemples.



Vue partielle des exemples de gestes pour l'apprentissage du geste "création de rectangle"
(en tout, une quarantaine d'exemples)

Les méthodes structurelles [Kojima88], [Baudel90] se proposent de partir d'une description des gestes à reconnaitre sous la forme de contraintes géométriques. Le geste "création de rectangle" sera représenté comme l'addition de deux segments joints, l'un vertical et l'autre horizontal; chaque segment est représenté par l'équation paramétrique d'une droite. Ceci permet au concepteur de l'interface de préciser les paramètres variables du geste et ceux qui doivent rester fixes. Les paramétres fixes fourniront les caractèristiques propres au type du geste, tandis que ceux qui sont variables correspondront aux paramètres passés en argument à la commande associée au geste. L'inconvénient est que la détermination des contraintes correspondant à un type de geste n'est pas aisée, et moins naturelle que la simple donnée d'exemples. L'avantage est que l'on peut être assuré d'un bon taux de reconnaissance dès lors que les contraintes fournies sont correctes. De plus, cette méthode permet d'envisager une analyse des ambiguités possibles du jeu de commandes gestuel en cours de définition. Ceci peut s'avérer utile, car notre expérience nous montre que la présence d'ambiguités entre deux gestes est souvent difficile à déterminer, et influe sur le taux de reconnaissance. Nous étudions actuellement une technique hybride permettant de tirer parti des avantages des deux approches.

III Intégration dans un environnement multimodal

L'intégration d'une composante gestuelle à une interface classique se fait de manière assez naturelle en associant des commandes de base de l'application aux gestes du vocabulaire reconnu. Dans une architecture classique basée sur une file d'évènements, un analyseur sera considéré comme un périphérique de haut niveau, captant les entrées du périphérique gestuel, les filtrant, et émettant des évènement discrets lorsqu'une commande est reconnue, dont le type désigne la commande invoquée, et les caractéristiques (telles que position, longueur, étendue…), les paramètres extraits du geste réalisé. Les gestes analysés peuvent donc être placés dans un ensemble plus global regroupant l'ensemble des dispositifs d'entrée, et permettant l'implantation de mécanismes introduisant la multi-modalité.

Cependant, un tel type d'intégration limite l'analyse d'un language gestuel à son niveau lexical. De même que les languages textuels, il est possible de doter les languages gestuels d'une syntaxe afin d'enrichir les possibilités d'expression de la modalité gestuelle. [Murakami91] et [Baudel90] proposent des méthodes de définition de grammaires gestuelles permettant la réalisation de language gestuels complets. Celles-ci se basent sur la définition de grammaires, dont les régles sont constituées de léxèmes non nécéssairement ordonnés temporellement (pour résoudre les problème de synchronisation) et de contraintes géométriques ou temporelles sur les léxèmes permettant de déterminiser le language. Nous faisons remarquer dans notre rapport qu'il est possible d'ajouter à ce type de grammaires des éléments lexicaux non gestuels, et de permettre de la sorte la définition de grammaires de définition d'interfaces multi-modales. Ces éléments lexicaux peuvent provenir par exemple d'un clavier ou d'un système de reconnaissance de mots isolés.

Ce type de grammaires est en fait assez proche des systèmes de définition d'interfaces multi-modales comme [Lewis91]. Il révèle cependant un problème à résoudre : l'une des modalités d'expression doit être prépondérante dans le style d'interaction. Dans les exemples actuels d'interfaces multi-modales, comme [Caelen91], le geste n'est utilisé que comme complément à la parole, pour désigner de façon ponctuelle des paramètres géométriques des commandes. La modalité gestuelle n'est utilisée que de façon statique. De même, dans une interface gestuelle intégrant d'autres modalités, les autres périphériques ne doivent pas entraver le flot principal des commandes, et se contenter de préciser certains paramètres ou compléments à des commandes, sans quoi on risque de perdre la cohérence du style d'interaction.

Le choix d'une modalité dominante, le geste ou la parole, limite les problèmes de la multi-modalité à la résolution des coréférences, que l'on pense solubles. En revanche, si l'on désire tirer parti pleinement de toutes les capacités de plusieurs moyens d'expression humains, on s'aperçoit que plusieurs modalités risquent de se faire concurrence dans la gestion du fil principal du dialogue. Il n'est à notre avis pas évident que la modalité vocale, qui exige d'importantes capacités de calcul, doive être choisie comme mode principal de communication. Nous espérons avoir apporté des éléments en faveur du choix du geste comme vecteur principal d'un dialogue multi-modal, tout au moins dans les applications sans prépondérance textuelle (où le clavier et éventuellement la parole ont un rôle dominant naturel).

Nous travaillons actuellement sur un prototype d'application multi-modale, à dominante gestuelle. Gest est un logiciel de dessin structuré, du type de GEdit, dans lequel le clavier (qui pourrait être remplacé par un système de reconnaissance vocale de mots) fournit des paramètres supplémentaires aux commandes : la couleur des objets, l'épaisseur du trait et d'autres variables peuvent être précisés en même temps qu'un objet est créé ou modifié par une commande gestuelle. On obtient ainsi une très grande concision dans l'expression, qui devrait permettre de vérifier que l'apport partiel d'autres modalités à une interface gestuelle permet la définition d'un style d'interaction très efficace.

Conclusion

Tout comme les interfaces multi-modales, il reste encore d'importants travaux à réaliser pour voir se généraliser les interfaces gestuelles. Nous proposons un modèle d'architecture permettant d'intégrer une composante gestuelle à une interface multi-modale. Cependant, il reste à réaliser des tests sur les performances effectives de ces types d'interfaces. En particulier, il convient d'étudier de façon approfondie les facteurs humains entrant en jeu dans ces interfaces. Un des problèmes se posant en effet, tant dans les interfaces multi-modales que dans l'analyse des gestes est le respect du "déterminisme de la présentation". Il est en effet admis qu'une interface conviviale se doit de présenter autant que possible les actions et les commandes que peut réaliser l'utilisateur dans un état donné. Si cela ne pose pas de problème particulier dans les interfaces à base de menus et de formats de saisie, qui ont une représentation visuelle intrinsèque, il est en revanche plus difficile de présenter à l'utilisateur des commandes vocales ou gestuelles accessibles dans un état donné. Notons simplement ce problème est étudié, entre autres dans [Kurtenbach91].

Nous avons présenté les atouts des interfaces gestuelles, et proposé un moyen de les intégrer dans les interfaces multi-modales, sans résoudre certains problèmes encore ouverts. Il importe de noter que les interfaces gestuelles offrent une caractéristique unique en permettant l'analyse de mouvements continus de l'utilisateur. Elles apportent une dimension totalement nouvelle au style d'interaction, ce qui en fait un des élément clés des interfaces de demain. S'il est encore difficile de savoir comment intégrer l'analyse gestuelle dans une interface multi-modale où cette modalité ne serait pas dominante, on espère en revanche trouver de grands avantages à l'adjonction de modalités complémentaires aux interfaces gestuelles.

Bibliographie

[Baudel90] Thomas Baudel, *Analyse de languages gestuels*, mémoire de DEA, L.R.I., Université Paris XI, 1990.

[Buxton86] W. Buxton, *There's more to interaction than meets the eye*, in Norman, D.A. & Draper, S.W., User centered system design: New perspectives on Human-Computer interaction, Hillsdale, N.J. Lawrence Erlbaum Associates, pp 319-337.

[Caelen91] Caelen, J. *Interaction multi-modale dans ICPdraw: expérience et perspectives.* Annales de l'école IHM multi-modale, GRECO-PRC communication Homme-Machine, 1991.

[Helm91] R. Helm, K. Marriott, M. Odersky, *Building Visual Language Parsers*. Proc. of SIGCHI'91, ACM, p.105.

[Kojima88] H. Kojima & T. Tioda, *On-line hand-drawn line figure recognition and its application*. IEEE Proc. on 9th Int. Conf. on Pattern Recognition, p. 1123. 1988.

[Kurtenbach90] Kurtenbach, G. & Buxton, W. *GEdit: a testbed for editing by continuous gesture.* SIGCHI Bulletin, 23(2), 22-26.

[Kurtenbach91] Article à paraitre dans SIGCHI Bulletin, Septembre 1991.

[Lewis91] J. Bryan Lewis, L. Kovec & D.T. Ling, *Dialogue structures for virtual words*. Proc. of SIGCHI'91, ACM, p.131.

[Moody83] Bill Moody, *La langue des signes*, Tomes 1, Ellipses, 1983.

[Murakami91] Murakami, K. & Taguchi, H. *Gesture Recognition Using Recurrent Neural Networks*. Proc. of SIGCHI'91, ACM, p.237.

[PenPoint91] *Go Corp.'s PenPoint Operating System for Mobile, Pen-Based Computers*. Techincal documentation, Go Corporation, 1991.

[Rubine91] D. Rubine, *Specifying Gestures by Example*. Proc. of SIGGRAPH'91, ACM, pp 329-337.

[Samuels90] P. Morrel Samuels, *Clarifying the distinction between lexical and gestural commands*. Int. Journal of Man-Machine studies, V. 32, p. 581, 1990.