

Entre manipulation directe et reconnaissance de l'écriture : les gestes écologiques

Christophe Mertz, Jean-Luc Vinot, Daniel Etienne

► **To cite this version:**

Christophe Mertz, Jean-Luc Vinot, Daniel Etienne. Entre manipulation directe et reconnaissance de l'écriture : les gestes écologiques. Ergo-IHM 2002, Ergonomie et informatique Avancée, Interaction Homme-Machine, Oct 2002, Biarritz, France. pp 145-152. hal-00940956

HAL Id: hal-00940956

<https://hal-enac.archives-ouvertes.fr/hal-00940956>

Submitted on 24 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Entre manipulation directe et reconnaissance de l'écriture : les gestes écologiques

Christophe Mertz^{1,3}, Jean-Luc Vinot², Daniel Etienne¹

(1) CENA
7, avenue E. Belin
31400 Toulouse, France

(2) CENA
1, rue de Champagne
91200 Athis-Mons, France

(3) CR2A-DI
32, rue des Cosmonautes
31029 Toulouse cedex, France

{mertz|vinot|etienne}@cena.dgac.fr

RESUME

Dans cet article, nous décrivons d'abord quelques techniques d'interaction à base de gestes, tracés en deux dimensions. Nous les plaçons au sein d'un continuum de plus en plus complexe, qui s'étend de la manipulation directe à la reconnaissance d'écriture. Nous rappelons ensuite le contexte applicatif de DigiStrips, un prototype de strips électroniques pour les contrôleurs aériens. Cette IHM est basée sur l'utilisation d'écran tactile et d'interactions à base de gestes. L'utilisateur peut déplacer et manipuler les strips, y saisir des valeurs, les annoter par du texte libre ou les marquer. Nous décrivons plusieurs de ces interactions qui mêlent reconnaissance de gestes, gestes de déplacement à une ou deux mains, gestes de commande, menus et autres interacteurs, saisie de texte libre et gestes déictiques. Bien choisis et harmonieusement combinés avec des animations, les gestes peuvent constituer une interface écologique. Nous les appelons des gestes écologiques. Avant de conclure nous ébauchons nos travaux futurs.

MOTS CLES : écran tactile, gestes, interactions à base de gestes, reconnaissance de gestes en 2 dimensions, interaction à deux mains, contrôle aérien, animations.

INTRODUCTION

Les interfaces homme machine (IHM) sur écrans tactiles n'utilisent généralement que le clic simple. Cela est compréhensible pour des bornes interactives destinées au grand public et avec lesquelles les utilisateurs doivent «deviner» la manière d'interagir : l'appui du doigt sur un bouton dessiné sur l'écran est naturel. Par contre des IHM dédiées à des postes de travail dont les utilisateurs sont formés ou entraînés peuvent bénéficier d'autres types d'interactions comme le tirer&placer, des gestes tracés sur l'écran (reconnus ou non par le système), le clic maintenu, le double-clic... Ainsi dans Tivoli (voir la vidéo [19]), un tableau blanc électronique, l'utilisateur peut activer une fonction (par exemple : déplacer un item d'une liste) d'un geste, puis déplacer l'item d'un autre geste. Les deux gestes de l'utilisateur ont des sémantiques différentes : le premier désigne l'item et la fonction, alors que le deuxième donne la destination. On trouve également des interactions à base de gestes dans les «assistants personnels digitaux» (PDA) pour la saisie de texte libre : Graffiti. Dans le domaine du contrôle

aérien, la maquette Grigri [6] a utilisé des gestes tracés au stylet pour la saisie d'instructions de contrôle en mêlant gestes de désignation et de choix de fonction et gestes de saisie numérique. Plus récemment, DigiStrips [17] a montré l'intérêt d'utiliser écran tactile, gestes simples, animations, et un design graphique pour obtenir une interface naturelle. Depuis peu, cet outil propose des fonctions de saisie [18], accessibles par de nouvelles interactions à base de gestes de natures très diverses. Dans cet article nous rappelons d'abord différentes techniques d'interactions gestuelles (en 2 dimensions) qui se placent dans un continuum de gestes entre la manipulation directe et la reconnaissance de l'écriture. Puis nous décrivons rapidement le contexte applicatif de DigiStrips. Ensuite nous analysons finement ces nouvelles interactions gestuelles. Nous présentons nos travaux futurs avant de conclure.

Nous allons maintenant positionner différentes interactions connues à base de gestes en 2 dimensions entre la manipulation directe et la reconnaissance de l'écriture.

UN CONTINUUM DANS LES GESTES

La majorité des interfaces sont aujourd'hui de type WIMP (Windows, Icon, Mouse, Pointer) et proposent des interactions «directes» introduites par Schneiderman [21]. Dans de telles interactions, les changements de l'état du système dépendent généralement du point de départ et du point d'arrivée d'une interaction. Par exemple, lors d'un cliquer&tirer sur un objet dans un éditeur graphique, à la fin de l'interaction, seuls comptent les points de départ et d'arrivée. Il en est de même lors du redimensionnement d'une fenêtre ou d'un défilement à l'aide d'un ascenseur. Les mouvements de l'utilisateur sont donc très peu interprétés par le système. Les menus hiérarchiques (figure 0) constituent une exception car la trajectoire du pointeur influe directement sur les ouvertures des sous-menus. A l'opposé, l'écriture manuscrite (et tout système de *reconnaissance d'écriture*) dépend essentiellement de la trajectoire. La capacité des machines à reconnaître de telles trajectoires complexes, l'écriture manuscrite, est encore limitée.

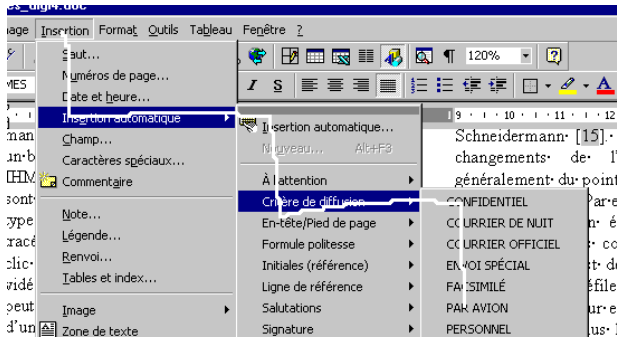


Figure 0 : trajectoire et menus hiérarchiques

Mais comme l'a préfiguré Schneiderman [22] : «[the challenge is] to explore how we might use sliding, dragging, and other gestures to move objects and invoke actions», plusieurs brèches se sont ouvertes entre le paradigme de la manipulation directe et celui de la reconnaissance de l'écriture. La manipulation directe est généralement impossible en semi-aveugle. Au contraire, ces nouveaux paradigmes d'interaction sont partiellement faisables en aveugle, tout comme l'écriture manuscrite. Ces interactions sont généralement réalisées au stylet ou sur un écran tactile. En voilà quelques exemples :

- La **saisie libre** : dans Tivoli [19], un tableau blanc électronique permet la saisie de texte ou de dessin entièrement libres, non reconnus par le système. Celui-ci permet aussi d'organiser cette information libre grâce à des gestes de commande sur le tableau. D'autres applications comme DENIM [13] et Knight [8] permettent le griffonnage (non interprété) de pages d'un site WEB et de diagrammes UML.
- Le **déplacement avec reconnaissance de gestes** : dans DigiStrips [17], des gestes simples reconnus en continu permettent de manipuler des objets graphiques à l'aide de plusieurs fonctions : le décalage, le poussage et le déplacement (voir plus bas).
- Les **marking-menus** : Kurtenbach [10] a proposé d'utiliser des gestes rectilignes simples pour sélectionner un item au sein de menus circulaires. Ces gestes peuvent être enchaînés pour former des **marking-menus hiérarchiques**. De plus, l'appui maintenu du stylet (ou de la souris !) pendant une durée suffisante provoque l'affichage du menu afin d'aider l'utilisateur qui aurait oublié la direction des items. Après quelque temps d'utilisation, il devient possible d'utiliser les marking-menus en aveugle, sans les faire apparaître.
- La **reconnaissance de gestes simples** : Graffiti, un système de reconnaissance de caractères simples, réalisés d'un trait, s'est imposé comme la solution au problème de la saisie de texte. Au contraire, les utilisateurs du Newton d'Apple, le précurseur des PDA (Personnal Digital Assistant) avaient pu constater les limites de la reconnaissance de l'écriture cursive. Le Palm Pilot, une évolution du Newton s'est ainsi imposé comme nouvelle interface pour l'utilisateur.

- Les **gestes de commande** : Chatty dans Grigri [6] a proposé la possibilité de modifier l'état d'un objet par un **geste de commande** ; l'objet et la fonction appliquée étaient indiqués par le point de départ du geste et par sa forme.
- **gestes+menus** : Grigri proposait aussi l'ouverture d'un menu par un geste et le choix d'une valeur par un clic ou un trait sur l'item sélectionné. Ces techniques ont été reprises dans DigiStrips. Elles sont décrites plus loin ainsi que des extensions possibles, comme par exemple l'outil **saisie_cap** et l'outil **geste+menu+geste**.
- **Ligne Claire** : Baudel [1] avec «Ligne Claire» a proposé un système de reconnaissance de gestes pour la saisie de courbes complexes tracées puis corrigées au stylet.
- **Teddy** : Plus récemment, Igarashi a proposé dans Teddy [9] la saisie d'objets en 3D par des dessins simples au stylet sur un écran. Grâce à la spécialisation de ce logiciel et à certaines heuristiques, Teddy ajoute la 3^{ème} dimension au dessin sans que l'utilisateur ne décrive explicitement cette dimension.

Les différentes techniques basées sur la reconnaissance de gestes, citées précédemment (en italique et gras), sont représentées sur le diagramme de la figure 1. L'abscisse représente la complexité du geste (i.e. de la forme tracée ou effectuée par l'utilisateur) alors que l'ordonnée indique la quantité d'information reconnue par le système.

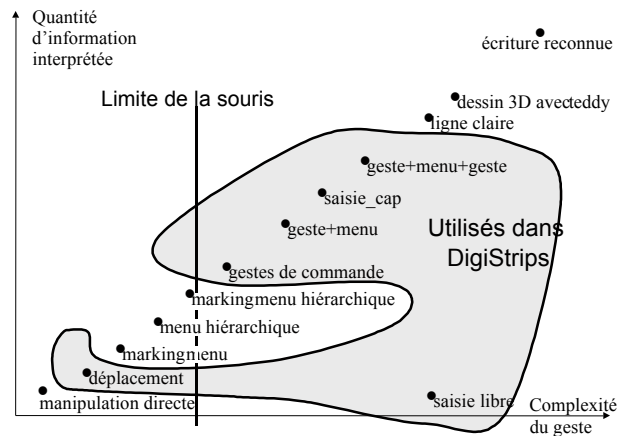


Figure 1 : un continuum des gestes

Ce graphique montre que la saisie libre est une technique particulière puisque les gestes sont très complexes mais absolument pas reconnus par le système. Nous avons aussi fait apparaître la limite au-dessus de laquelle les gestes ne sont plus facilement faisables à la souris. Ce diagramme ou continuum de gestes permet de représenter la diversité des interactions à base de geste. La zone grisée indique les types d'interactions gestuelles utilisées dans un prototype que nous allons détailler maintenant.

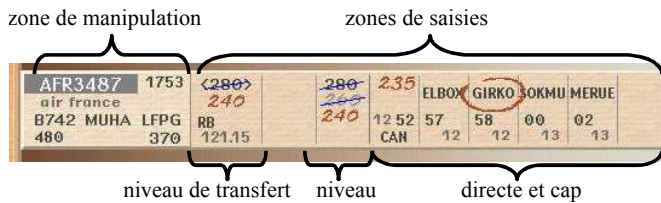


Figure 3 : la zone de manipulation et les zones de saisie

Les gestes de manipulation

Les gestes de manipulation sont similaires aux déplacements avec un doigt de petits feuillets sur une table. Il faut appuyer du doigt sur l'en-tête du strip (zone de manipulation figure 3), le faire glisser du doigt et finalement le relâcher. Les fonctions décaler, déplacer, pousser sont reconnues (voir figure 4 et aussi [15]) pendant l'opération «tirer». L'utilisateur a le choix du placement, mais le système complète le placement des strips par des animations en fin d'interaction. Les gestes de l'utilisateur sont «mimétiques» (ils imitent le déplacement de petits feuillets sur une table) et sont donc faciles à mémoriser et à reproduire. De plus ils se font à partir de l'en-tête où se trouve l'indicatif. Cela permet donc d'allier la manipulation avec le geste de désignation sur l'indicatif (l'indicatif identifie le vol), geste déictique naturel des contrôleurs aujourd'hui. Ces gestes de manipulation sont donc écologiques.

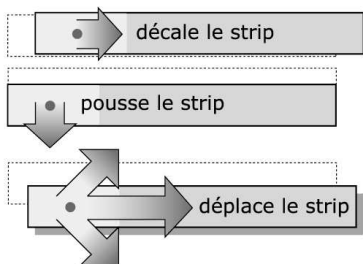


Figure 4 : les gestes reconnus pour la manipulation des strips

- Des déplacements significatifs

Les opérations de déplacement peuvent ou non être interprétées par le système. Par exemple un déplacement entre deux colonnes du tableau n'est pas interprété. Par contre, le déplacement d'un strip d'une colonne vers la «poubelle» peut être associé à la fin de vie du vol dans le secteur. Différentes zones significatives² à l'écran ont été identifiées :

- Le tableau proprement dit, composé de deux ou trois colonnes.
- L'imprimante par où apparaissent les nouveaux strips.
- La poubelle pour indiquer au système que le vol n'est plus contrôlé par ce secteur.
- La zone de transfert vers l'écran de l'autre contrôleur du même secteur

² toutes ces zones ne sont pas nécessairement disponibles simultanément pour un même utilisateur

- Une ou des zones de transfert vers les secteurs suivants.

Tout déplacement depuis ou vers l'une de ces zones se traduit par un changement d'état du vol.

- Des déplacements à deux mains

Les dalles tactiles résistives que nous utilisons ne différencient pas deux appuis simultanés. Elles intègrent les appuis multiples et rendent le barycentre. Cette propriété a été mise à profit par un contrôleur qui a utilisé DigiStrips lors d'évaluations préliminaires au Centre Expérimental d'Eurocontrol [5]. Pour déplacer un strip entre les 2 colonnes, il a désigné de l'index gauche le strip dans la colonne de gauche par un appui maintenu. En maintenant cet appui, il a indiqué de l'index droit la destination dans la colonne de droite. Le strip se déplace donc entre les deux doigts. En relâchant l'index gauche, le strip se déplace dans la colonne droite. L'utilisateur relâche l'index droit. L'interaction à deux mains est résumée dans le chronogramme de la figure 4b.

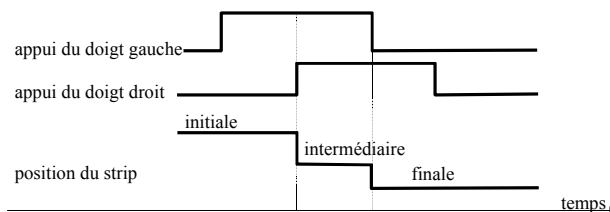


Figure 4b : chronogramme du déplacement à deux mains

Un test rapide et informel a montré que la durée de l'interaction à deux mains est similaire à la durée l'interaction à un seul doigt. Par contre les mains se déplacent très peu et cela engendre peut-être moins de fatigue. Ce type d'interaction à deux mains est permis par l'écran tactile. Il est évidemment impossible avec une souris et demande à être davantage étudié. De manière similaire, il est possible de décaler les strips à l'aide de deux doigts (index et pouce) d'une même main. Est-ce utilisable pour d'autres interactions? La question est ouverte.

Les gestes de saisie

Les différents gestes disponibles pour déclencher des saisies sont décrits figure 5. Exceptés les gestes rectilignes verticaux ou horizontaux, le sens de réalisation est indifférent (par exemple pour les >, <, ^ et V). Le geste *alpha* peut être tracé selon un miroir vertical. En effet les utilisateurs ne sont pas tous des «mathématiciens» et peuvent avoir des difficultés à mémoriser le dessin de l'*alpha*. Pour être écologiques, ces gestes doivent éviter toute ambiguïté, tout en restant les plus proches possibles de l'écriture scripte naturelle pour limiter l'apprentissage. Nous avons donc éliminé le N qui ressemble trop à un W quand le scripteur commence l'écriture en haut à gauche (par exemple : \mathcal{N}). De même le C est trop proche d'un < et a donc été éliminé de notre sélection.

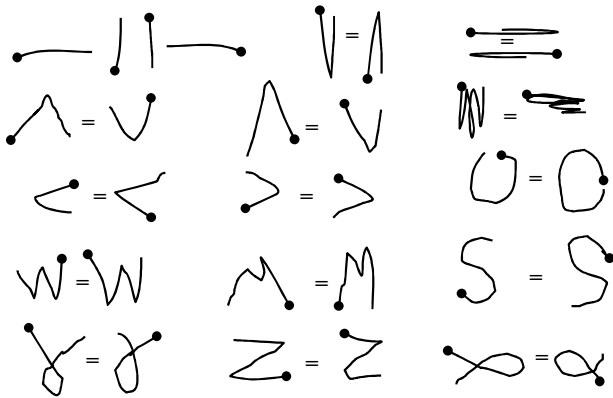


Figure 5: les gestes de commande reconnus pour la saisie d'information (tous ne sont pas utilisés actuellement)

Nous allons maintenant décrire certains des dialogues entamés par ces gestes, et montrer qu'il est intéressant de mêler gestes, clics et autres interactions.

Un dialogue geste+menu

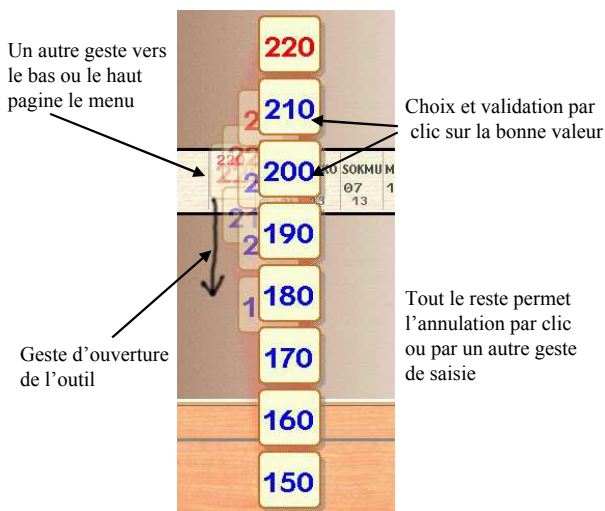


Figure 6 : l'outil de saisie de niveau

Pour déclencher la saisie d'un niveau de vol (c.-à-d. une altitude) nous avons choisi deux gestes métaphoriques tracés dans les zones de niveau (figure 3). Un trait vertical vers le bas ouvre un menu en descente alors qu'un trait vertical vers le haut ouvre le menu en montée. Ces gestes sont faciles à mémoriser et simples à réaliser. De plus ils permettent d'ouvrir le menu (figure 6) sur un ensemble de valeurs pertinentes en montée ou en descente. Le menu s'ouvre avec une animation et l'utilisateur dispose de plusieurs possibilités :

- Il peut choisir une valeur en cliquant sur celle-ci. En fait nous utilisons la méthode «take-off» [20] ; Plutôt que le point de contact initial avec l'écran, nous utilisons le point au moment du relâché du doigt. Cela permet à l'utilisateur d'ajuster la position de son doigt au cas où il aurait mal désigné la valeur dans le menu.

- Si le vol plonge ou monte vers un niveau très éloigné, le contrôleur peut répéter le geste vertical dans la case niveau. Le menu va dérouler les 7 valeurs suivantes avec une animation.
- Il peut annuler le dialogue par un clic en dehors du menu ou ouvrir un autre dialogue d'un geste (par exemple le dialogue niveau sur le bon vol, en cas d'erreur de désignation).

Un dialogue geste+menu+geste

Le dialogue précédent de saisie de niveau peut être étendu en utilisant un geste au moment de la sélection du niveau. Par exemple, saisir l'instruction complexe «descendez au niveau 200 sur la balise GIRKO», peut être fait de la manière suivante. Au lieu d'un clic sur le niveau 200, l'utilisateur trace un geste horizontal vers la droite. Cela ouvre une liste verticale des balises du plan de vol. L'utilisateur prolonge alors son geste verticalement pour sélectionner GIRKO dans la liste et relâche son appui. Cette extension n'a pas encore été réalisée, mais pourrait être très utile pour les saisies d'ordre complexe.

Le dialogue saisie_cap

Un autre geste métaphorique, le cercle (il rappelle à la fois la forme de l'outil qu'il affiche et l'avion qui tourne) entame le dialogue de saisie de cap (figure 7).

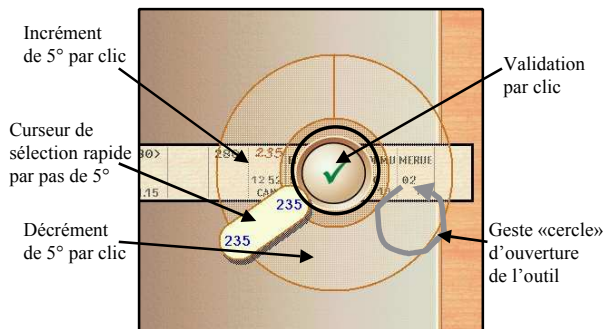


Figure 7 : l'outil de saisie de cap

Cet outil permet :

- De choisir rapidement n'importe quel cap, par pas de 5° à l'aide du curseur oblong. La précision est cependant limitée par l'écran tactile ; au moment du relâché du doigt, la valeur sélectionnée peut se modifier accidentellement.
- D'incrémenter ou décrémenter le cap, par pas de 5°, par un clic de part et d'autre du curseur oblong, sur la couronne semi-transparente. Cela permet par exemple d'ajuster la valeur sélectionnée précédemment.
- De valider la valeur sélectionnée par un clic dans le bouton central.
- D'annuler ce dialogue par un clic ou un geste tracé ailleurs sur l'écran.

Les gestes d'écriture

Sur un écran tactile résistif il est possible d'écrire du texte libre, non reconnu par le système, avec un stylet passif. Cela est largement utilisé par certains PDA et cette fonctionnalité est appréciée dans Knight [8] car elle permet de mêler des explications ou des schémas tracés librement au stylet à un diagramme UML. Dans un tel système où la majorité des saisies sont interprétées, les contrôleurs apprécient la souplesse d'une fonction bloc-notes. Celle-ci est accessible par un geste de commande (—) et le strip s'agrandit (figure 8) avec une animation.

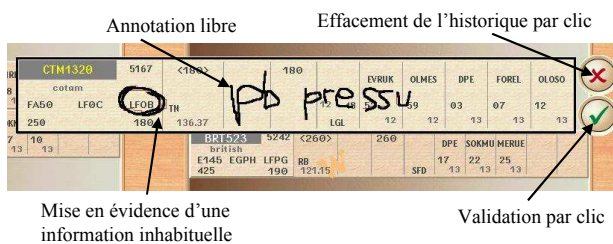


Figure 8 : strip agrandi pour la saisie d'annotations libres

L'utilisateur peut alors griffonner un texte, «souligner» ou barrer des informations système. Le strip est entièrement accessible, même la zone d'en-tête habituellement réservée au déplacement. En fait ce strip agrandi ne peut pas être déplacé et seule l'écriture y est possible. Tous les autres strips restent accessibles pour d'autres interactions, comme des déplacements ou des saisies. Le dialogue est donc faiblement modal. Ceci est important car l'utilisateur peut interrompre sa tâche d'annotation pour agir sur un autre vol. L'agrandissement du strip et sa texture plus claire explicitent cette modalité particulière. Le zoom facilite aussi l'écriture et les marques restent bien positionnées à la fermeture. Il s'agit là d'un usage particulier d'une interface zoomable [2], permis par Zinc³ [12], un widget de la toolkit Tk développé au CENA.

Le W comme warning

Bien qu'il ne s'agisse plus d'écriture libre, nous proposons une autre annotation semi-libre. Le geste W (geste symbolique qui rappelle le mot Warning) peut être tracé sur n'importe quelle zone de saisie (figure 3). Il est reconnu et un W s'affiche sur le strip à l'endroit où il a été tracé. L'information peut ainsi être marquée par l'utilisateur et le système peut la particulariser sur les autres représentations du vol sur les autres écrans à la disposition du contrôleur.

D'autres gestes de commande sont possibles, comme un trait vertical pour sélectionner une «directe» c.-à-d. la sélection d'une balise parmi celles affichées sur le strip.

³ Zinc sera disponible sous licence LGPL sur le site openatc : www.openatc.org.

Il est aussi possible d'indiquer d'un geste (V ou ^) que l'avion va «shunter» une ou plusieurs balises c.-à-d. aller directement de la balise i à la balise i+n. Dans ce cas, la fonction est indiquée par la nature de geste, alors que les points de départ et d'arrivée du geste indiquent les balises «shuntées».

L'annulation de saisies peut aussi se faire par un geste symbolique : une rature (i.e. une succession d'aller-retour horizontaux et verticaux). Dans ce cas, c'est la zone couverte par la rature qui indique l'opération à annuler.

Les gestes de type clic

Nous parlons ici de gestes (et non de manipulation directe) car ces interactions se font sur un écran tactile. L'utilisateur doit donc faire un vrai geste de sa main ou de son bras pour désigner et presser une zone sensible. Au contraire, avec la souris, ce sont de petits mouvements des doigts et de la main que l'utilisateur doit coordonner avec les déplacements du curseur à l'écran.

- le clic maintenu sur l'en-tête pour particulariser le vol

L'appui maintenu d'un doigt sur l'en-tête d'un strip (zone de manipulation, figure 3) active la particularisation du vol pendant la durée de l'appui. Le système peut ainsi particulariser ce vol sur toutes ses représentations comme par exemple sur l'image radar ou dans des listes affichées sur d'autres écrans du contrôleur.

- le clic maintenu pour accéder à l'aide-mémoire des gestes

Bien que les gestes de commande soient simples à tracer et à mémoriser, un utilisateur novice ou confirmé peut oublier comment et où réaliser un geste pour une fonction particulière. C'est pourquoi, nous proposons un mécanisme similaire à celui des marking-menus. Un appui maintenu de quelques centaines de ms sur une des zones de saisie (figure 3) provoque l'affichage d'une image d'aide en superposition au strip ainsi désigné. Cette image montre l'ensemble des gestes possibles, dessinés dans les zones respectives. Les fonctions associées aux gestes sont aussi affichées. L'utilisateur peut ainsi retrouver rapidement le geste ou la fonction désirée. Quand il relâche son doigt, l'aide disparaît. L'utilisateur peut aussi tracer un geste en continuation de son appui. Le clic maintenu est un geste «écologique» car il correspond naturellement à un geste d'hésitation et peut se prolonger par le geste de commande idoine.

- le double clic sur l'en-tête

Toutes les fonctions applicables à un vol ne peuvent pas être accessibles par des gestes de commande. En effet cela nécessiterait trop de gestes différents qui seraient trop difficiles à mémoriser. Les fonctions peu fréquentes ou trop complexes comme la modification d'un plan de vol nécessitent une IHM spécifique. Y accéder et

désigner le vol concerné se fait par un double-clic⁴ du doigt sur l'en-tête du strip.

Les gestes déictiques

Certaines interactions du contrôleur avec son environnement ont souvent été sous-évaluées. C'est le cas des gestes déictiques utilisés pour la communication non-verbale entre les deux ou trois contrôleurs collaborant sur un même secteur. Ces gestes se font aujourd'hui sur l'image radar ou sur le tableau de strips. Ils indiquent un vol ou l'une de ses caractéristiques inhabituelles. C'est la coopération silencieuse. Les interface WIMP ont rendu ces gestes moins pratiques à cause de l'éloignement des écrans. De plus une main est monopolisée par la souris et il devient plus difficile de pointer l'index. Avec les écrans tactiles ces gestes sont facilités : les écrans redeviennent accessibles des mains et surtout la main, libérée de la souris, devient l'outil d'interaction. Les gestes déictiques sont réhabilités.

TRAVAUX FUTURS

Les gestes décrits dans cet article n'ont pas encore été évalués lors de simulations. Une expérimentation [16] avait comparé écran tactile et souris pour les seuls gestes de manipulation. D'autre part, de nombreuses démonstrations et évaluations informelles [17] ont montré l'intérêt de DigiStrips. Cela est évidemment insuffisant pour valider un tel système. De plus, les postes de travail des contrôleurs aériens en-route sont en cours de changement en France. C'est le projet ODS-France et celui-ci ne prévoit pas d'introduire d'écrans tactiles adaptés à des saisies et manipulations routinières. Enfin, il est prévu d'introduire Erato [4], un outil d'aide au contrôle qui apporte des fonctions de filtrage des vols et un agenda pour planifier l'activité entre les contrôleurs du secteur. Cette dernière fonction peut difficilement cohabiter avec un tableau de strips (papier ou électronique). Par contre, d'autres pays dans le monde sont moins avancés dans la modernisation de leurs Centres de Contrôle. DigiStrips peut présenter un intérêt pour ceux-ci. C'est pourquoi, le Centre Expérimental d'Eurocontrol, en collaboration avec le CENA, évaluera prochainement le prototype, intégré au sein d'un poste de travail [5], lors de simulations réalistes. Cela nous apportera certainement de nombreuses informations.

Parallèlement à cette évaluation, nous cherchons à améliorer la coopération entre deux utilisateurs d'un système comme DigiStrips : même si un écran tactile est utilisable en alternance par deux ou trois contrôleurs, il ne peut pas être unique. Il faut donc proposer plusieurs écrans tactiles et imaginer des moyens de collaboration autour de ces multiples écrans. Nous devons aussi

étendre les fonctions de saisie, tout en maintenant le naturel des gestes écologiques actuellement utilisés. Il faudra certainement améliorer le système de reconnaissance de gestes, en fonction des évaluations menées par Eurocontrol.

Enfin, nous généraliserons les techniques utilisées dans DigiStrips au sein d'un projet appelé Digi* (ou DigiStar) qui appliquera le design graphique, les écrans tactiles, la reconnaissance de gestes et les animations à l'ensemble des IHM d'un poste de travail de contrôleurs aériens.

CONCLUSION

Dans cet article, nous avons d'abord présenté diverses interactions à base de gestes en 2 dimensions. Nous les avons classées au sein d'un espace de conception que nous avons appelé le continuum des gestes. Nous avons ensuite décrit plusieurs interactions nouvelles obtenues en combinant des interactions gestuelles, des saisies de texte libre, des menus et des interactions plus classiques comme les clics. Nous avons montré que le caractère naturel des interactions est obtenu grâce à des gestes écologiques et d'autres techniques comme l'animation. Il nous reste à évaluer ces interactions gestuelles au cours d'expérimentations avec des contrôleurs et à appliquer ces techniques dans d'autres IHM.

REMERCIEMENTS

Stéphane Chatty a contribué à de nombreuses et fructueuses discussions. Patrick Lecoanet a réalisé Zinc, avec lequel a été implémenté la dernière version de DigiStrips. Robert Parise et Yannick Jestin ont commenté et relu cet article. Merci aussi à Yannick pour sa disponibilité pour le support technique ces derniers mois.

BIBLIOGRAPHIE

1. Thomas Baudel. A Mark-Based Interaction Paradigm for Free-Hand Drawing. In *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology*, Marina del Rey, 1994. ACM
2. Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7, 3-31, 1996.
3. Marie-Christine Bressolle, Bernard Pavard, Marcel Leroux. The role of multimodal communication in cooperation in the case of ATC. In *Lecture Notes in Artificial Intelligence*, Springer, 1998, vol 13 73, pp. 326-343.

⁴ Ce double-clic pose actuellement quelques difficultés. Nous devons nous assurer qu'il est compatible avec les autres interactions.

4. Marie-Christine Bressolle, Railane Benacem, Nicole Boudes, Robert Parise. Advanced decision aids for ATC : understanding different working methods from a cognitive point of view. *3rd USA/Europe Air Traffic Management E&D Seminar*, Napoli Italy, June 2000. <http://atm-seminar-2000.eurocontrol.fr/>
5. Sophie Carlier, Future Air Traffic Control environment prototype. Internal report. SkyTools/SSD/01 version 1.0. Centre Experimental d'Eurocontrol, Brétigny, France.
6. Stéphane Chatty, P. Lecoanet. A pen-based workstation for air traffic controllers. In *Proceedings of Human Factors in Computing Systems, CHI'96*, ACM/SIGCHI. 1996, pp. 87-94.
7. S. Gruszow, La bataille du carton, du stylo bille et de la souris. Dans *La Recherche* p62-64, Avril 1999.
8. Christian Heide Damm, Klaus Marius Hansen, Michael Thomsen. Tool Support for Cooperative Object-Oriented Design: Gesture Based Modeling on an Electronic Whiteboard. In *proceedings of CHI2000 Human Factors in Computing Systems*, ACM/SIGCHI, 2000, pp. 518-525.
9. Takeo Igarashi, Satoshi Matsuoka, Hidehiko Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. In *proceedings of ACM SIGGRAPH'99*, Los Angeles, 1999, pp.409-416
10. Gordon Kurtenbach, William Buxton. Issues in Combining Marking and Direct Manipulation Techniques. In *Proceedings of ACM SIGGRAPH and SIGCHI Symposium on User Interface Software and Technology*, UIST '91, Hilton Head, SC, 1991. pp. 137-144.
11. Jean-Paul Labarthe, Expérimentation classement de strips : résultats de la simulation. Rapport Interne CENA/R94-022. Avril 1994.
12. Patrick Lecoanet. Zinc Reference Manual. bientôt disponible sur www.openatc.org
13. James Lin, Mark Newman, Jason Hong, James Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *proceedings of CHI'2000 Human Factors in Computing Systems*, ACM/SIGCHI, 2000, pp. 510-517.
14. Wendy Mackay, Anne-Laure Fayard, Laurent Frobert, Lionel Médini. Reinventing the Familiar : Exploring an Augmented Reality Design Space for Air Traffic Control. In *Proceedings of CHI'98 Human Factors in Computing Systems*, ACM/SIGCHI, 1998, pp. 558-565.
15. Christophe Mertz, Jean-Luc Vinot. Conception par maquettage rapide : application à des écrans tactiles pour le contrôle aérien. Dans les *actes d'ERGOIA98*. Biarritz 1998. ESTIA/ILS. pp. 120-129.
16. Christophe Mertz, Jean-Luc Vinot. Touch Input Screens and Animations: more Efficient and Humanized Computer Interactions for ATC(O). In *Proceedings 10th International Symposium On Aviation Psychology*. Columbus, OH. OSU. May 1999. Vol. 1, pp. 615-621
17. Christophe Mertz, Stéphane Chatty, Jean-Luc Vinot. Interface gestuelle pour écran tactile, animations et design graphique : un guide de conception. Dans les *actes d'IHM99 Cépaduès-Editions*, Montpellier France, Novembre 1999. pp. 1-8.
18. Christophe Mertz, Stéphane Chatty, Jean-Luc Vinot. Pushing the limits of ATC user interface design beyond S&M interaction : the DigiStrips experience. *3rd USA/Europe Air Traffic Management E&D Seminar*, Napoli Italy, June 2000. <http://atm-seminar-2000.eurocontrol.fr/>
19. Thomas P. Moran, William van Melle. Tivoli: Integrating Structured Domain Objects into a Freeform Whiteboard Environment. In *extended Abstracts of CHI'2000 Human Factors in Computing Systems*, ACM/SIGCHI, 2000, pp. 20-21
20. Richard L. Potter, Linda J. Weldon, Ben Shneiderman. Improving the accuracy of touchscreens: an experimental evaluation of three strategies. In *Sparks of Innovation in Human-Computer Interaction*, edited by B. Shneiderman Ablex Publishing Corporation. 1993.
21. Ben Schneiderman, Direct Manipulation : a step Beyond Programming Languages. *IEEE Computer*, 16(8). pp 57-69. 1983.
22. Ben Shneiderman. Touchscreens now offer compelling uses, In *Sparks of Innovation in Human-Computer Interaction*, edited by B. Shneiderman Ablex Publishing Corporation. 1993.