

Technique multi-points indirecte relative pour l'interaction avec des écrans de grandes dimensions

Jérémie Gilliot

Inria Lille
jeremie.gilliot@inria.fr

Géry Casiez

Inria Lille, LIFL, Université Lille 1
gery.casiez@lifl.fr

Nicolas Roussel

Inria Lille
nicolas.roussel@inria.fr

ABSTRACT

Multitouch interaction shows its limits with large display surfaces. Indirect interaction allows to use control surfaces that are much smaller than display surfaces. Absolute indirect interaction raises accuracy problems and relative indirect interaction only allows to interact with a single cursor. We present a relative indirect multitouch interaction technique allowing to create, control, delete several cursors without sacrificing precision for interacting with small objects.

Author Keywords

Multi-touch devices ; indirect interaction ; multi-cursors.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation] : User interfaces

General Terms

Design ; Human Factors

INTRODUCTION

La majorité des interfaces multi-points sont basées sur une interaction co-localisée, c'est-à-dire que les utilisateurs interagissent directement avec les objets d'intérêt, sans l'intermédiaire d'un curseur. Ce paradigme d'interaction permet de facilement désigner et manipuler plusieurs objets en même temps, en particulier lorsque la surface tactile est suffisamment grande pour interagir à deux mains ou à plusieurs. L'interaction co-localisée pose cependant des problèmes d'occultation et peut dans certaines situations augmenter la fatigue physique, lorsque l'amplitude des gestes réalisés est importante par exemple. Dans le cas de surfaces tactiles de très grandes dimensions, à ces problèmes s'ajoutent la difficulté d'atteindre des objets en périphérie

de l'écran et la nécessité de se déplacer pour interagir avec des objets hors de portée.

Les pavés tactiles (*touchpads*) multi-points sont généralement utilisés en mode indirect. L'utilisateur contrôle dans ce cas un unique curseur de manière relative, ce curseur lui permettant de désigner des objets qu'il peut ensuite manipuler à l'aide de gestes réalisés avec plusieurs doigts. Une mise en correspondance absolue entre la surface d'interaction et la surface d'affichage est un moyen simple de ré-introduire plusieurs curseurs. Cependant, des facteurs d'échelles importants entre les surfaces de contrôle et d'affichage introduisent rapidement des problèmes de précision et peuvent rendre impossible la manipulation de petits objets avec plusieurs doigts.

Les travaux antérieurs ont exploré l'utilisation de l'interaction multi-points pour contrôler un seul curseur de manière relative [6, 4, 8], ou alors contrôler plusieurs curseurs mais en utilisant un périphérique par curseur [5]. Dans cet article, nous proposons une technique permettant de contrôler plusieurs curseurs de manière indirecte et relative en utilisant un seul périphérique d'interaction multi-points. Après une revue des travaux antérieurs, nous détaillerons notre technique avant de conclure.

TRAVAUX ANTERIEURS

Comparée à l'interaction mono-point, l'interaction multi-points permet d'augmenter la bande passante entre l'utilisateur et la machine puisqu'il devient possible de sélectionner ou manipuler plusieurs objets simultanément. Dans le cas d'une interaction co-localisée, Kin et al. [3] ont montré que l'interaction multi-points permet de diviser par deux le temps de pointage par rapport à une souris, et que l'usage de plusieurs doigts de la même main permet de manipuler aisément plusieurs degrés de liberté. L'interaction multi-points indirecte permet cependant moins facilement de manipuler plusieurs curseurs simultanément pour tirer partie de la bande passante offerte par ce paradigme d'interaction.

Comme évoqué dans l'introduction, le moyen le plus simple pour introduire plusieurs curseurs en interaction multi-

points indirecte est d'utiliser une mise en correspondance absolue entre la surface tactile et l'écran. Cette approche pose cependant des problèmes de pré-visualisation des contacts dans l'espace visuel et de précision de manipulation. Schmidt et al. ont comparé les interactions directe et indirecte absolues dans une tâche de positionnement [6]. Pour l'interaction indirecte, ils ont mis en évidence l'importance de pré-visualiser la position du contact dans l'espace visuel, en affichant des curseurs lors du survol de la surface, par exemple. Par ailleurs, lorsque le rapport des dimensions des surfaces d'affichage et de contrôle devient important, la manipulation de petits objets avec un contrôle absolu indirect devient difficile. Pour remédier à ces problèmes, Malik et al. ont proposé le *Visual Touchpad* [4] qui permet d'interagir de manière absolue dans une sous région de l'espace visuel. La main non-dominante permet de contrôler de manière absolue la position et la taille de la zone de travail de la main dominante. De façon similaire *iPodLoupe* [8] est une technique d'interaction reposant également sur la définition d'un espace de travail et le positionnement absolu des doigts à l'intérieur de cet espace. Ces techniques permettent d'augmenter la précision de manipulation, mais au prix d'une réduction de la zone de travail dans l'espace visuel.

L'interaction multi-points relative indirecte permet de lever ces limitations au prix de la manipulation d'un seul curseur. L'affichage permanent d'un curseur permet de prédire où l'interaction aura lieu et le contrôle relatif permet d'obtenir le degré de précision souhaité, sans limiter la zone d'action du curseur à l'écran. L'utilisation de fonctions de transfert non-linéaires permet en outre de rester précis tout en limitant le débrayage [2]. Les travaux antérieurs ont notamment proposé d'augmenter la souris d'une surface tactile pour manipuler des éléments situés sous le curseur [9, 7, 1]. Les pavés tactiles multi-points, comme le *Magic Trackpad*¹ d'Apple, sont un autre exemple d'interfaces de ce type. L'utilisation d'un doigt permet de contrôler de manière relative un curseur tandis que les gestes réalisés avec plusieurs doigts sont interprétés pour déclencher diverses commandes, e.g. zoom et rotation d'objets, ou commandes liées au système de fenêtrage. Pour interagir avec plus d'un curseur, Moscovich et Hughes ont introduit le *HandCursor* [5], qui permet de contrôler de manière relative et indirecte deux curseurs. Chaque curseur est cependant contrôlé par un périphérique spécifique. Tous les points de contacts sont affichés dans l'espace de travail. Le premier doigt est placé au centre du curseur et les doigts suivants sont placés de manière absolue par rapport au premier, pour faciliter leur identification. Séparer une surface tactile en plusieurs zones ou périphériques pour contrôler différents curseurs limite cependant la surface utile pour chaque curseur et rend le changement dy-

namique du nombre de curseurs délicat.

PROPOSITION

Notre proposition est une technique d'interaction permettant de contrôler plusieurs curseurs de manière relative et indirecte au moyen d'un unique périphérique multi-points. Cette technique est inspirée du *HandCursor* de Moscovich et Hughes [5], adapté pour permettre la manipulation de petits objets et enrichie d'un mécanisme de création, manipulation et destruction de curseurs. L'exemple suivant illustre un scénario concret d'utilisation de notre technique :

Alice est architecte d'intérieur et fait la revue d'un projet en présence de clients devant un écran de grande taille. Tous sont assis dans des fauteuils face à l'écran, Alice disposant d'une tablette tactile pour interagir avec les objets affichés. Par défaut, un unique curseur est visible à l'écran. Alice le contrôle avec sa tablette de la même manière que celui d'un ordinateur portable avec son pavé tactile. Dès qu'elle pose plusieurs doigts, des curseurs secondaires apparaissent à proximité du premier, lui permettant de faire facilement pivoter un petit objet à la demande des clients, ou de déplacer des objets proches les uns des autres à un autre endroit. La revue avançant, ses clients discutent de modifications à apporter dans deux chambres visibles aux extrémités opposées de l'écran. Pour ne pas avoir à déplacer de manière répétée son curseur d'une pièce à l'autre, Alice en fait apparaître un second en faisant un double-tap à la position voulue sur la tablette. Les clients lui demandant de juxtaposer deux lits dans l'une des chambres, elle fait apparaître un troisième curseur qui lui permet de manipuler simultanément les deux lits.

Les différentes opérations sont détaillées dans les paragraphes qui suivent. La Figure 1 présente l'automate représentant l'ensemble des actions qui ont lieu à chaque fois qu'un doigt est posé. Chaque doigt possède une machine à état, la liste des curseurs et des positions mémorisés sont communes à toutes les machines à états. Les transitions sont écrites sous la forme "Événement [& condition] [/ actions]".

Curseurs multi-points

L'utilisateur agit sur les objets au travers de curseurs multi-points. Un curseur représente un ensemble de doigts agissant dans une même zone. Nous avons repris les principes du *HandCursor* : un curseur multi-points est contrôlé de manière relative par un périphérique tactile et la génération d'un événement discret permet de distinguer le survol de la sélection. Dans le cas du *HandCursor*, cet événement était produit par un bouton placé sous la surface tactile. Nous proposons d'utiliser un *tap* (détaillé plus loin).

1. <http://www.apple.com/magictackpad/>

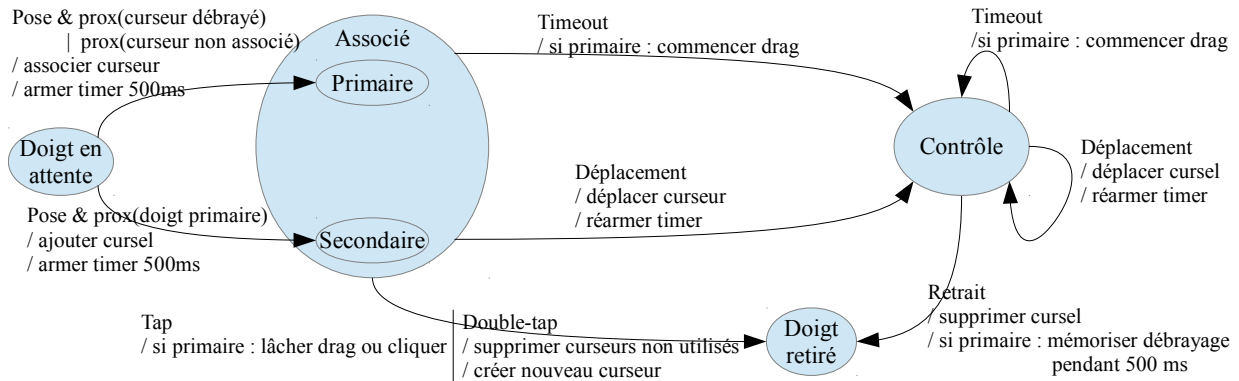


Figure 1 : Automate de fonctionnement. Chaque doigt est associé à un curseur dès l'instant où il est posé sur le périphérique, il devient doigt primaire s'il est plus proche d'un curseur non utilisé ; secondaire s'il est plus proche d'un autre doigt primaire.

Lorsque qu'un seul doigt contrôle le curseur, le point d'interaction est le centre du curseur. Celui-ci se comporte alors de manière similaire à un curseur de souris qui permet de sélectionner précisément un objet à la fois. Le premier doigt posé sur le périphérique sert de référence (Figure 1, *primaire*) pour le placement des doigts suivants.

Chaque doigt posé ensuite sur la surface tactile ajoute un curseur secondaire à l'écran, placé à proximité du premier selon la position du nouveau doigt par rapport au doigt primaire. De cette façon, l'utilisateur peut facilement identifier quel doigt bouger en fonction du curseur secondaire à déplacer. Ces curseurs secondaires apparaissent autour du curseur comme des éléments satellites. Nous les appelons par la suite *cursel* pour "*cursor element*" (Figure 2). Un curseur est représenté de manière permanente, même lorsque le doigt primaire n'est pas en contact avec la surface alors que les cursels ne sont affichés que lorsque les doigts secondaires sont en contact.

L'ensemble des doigts en contact contrôle le déplacement relatif du curseur. Le déplacement du centre de gravité du groupe de points est utilisé pour déplacer le curseur. La fonction de transfert choisie est celle utilisée par OS X pour les pavés tactiles [2], combinée à un facteur d'échelle de 7. Le déplacement indépendant de chaque doigt implique un mouvement global de l'ensemble du curseur et des cursels.

Multi-sélection et regroupement de doigts

Un curseur multi-points permet soit de manipuler un objet ou d'en sélectionner plusieurs selon que les cursels sont positionnés ou non sur d'autres objets. Pour pouvoir manipuler un objet, le curseur doit nécessairement être positionné dessus. Les cursels peuvent par contre être positionnés en dehors de l'objet (les cursels situés sur d'autres objets ne sont pas pris en compte pour la manipulation de celui sous le curseur). Contrairement à *HandCursor* qui nécessite de

placer tous les curseurs sur l'objet à manipuler, nous avons fait ce choix afin de faciliter la manipulation de petits objets. Les cursels positionnés sur d'autres objets permettent de les sélectionner, voire de les manipuler si plusieurs cursels sont placés sur un même objet.

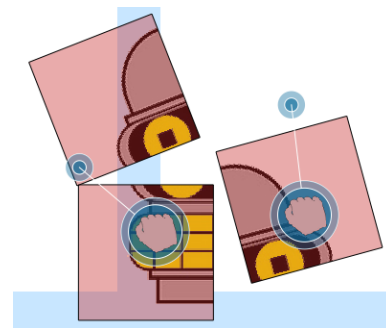


Figure 2 : Aperçu des curseurs et cursels. Manipulation de deux objets avec le curseur gauche et manipulation à deux doigts d'un seul objet à droite.

Gestion de la sélection d'objets

Nous utilisons un événement *tap* pour que les curseurs génèrent un événement discret permettant la sélection d'objets, comme sur un pavé tactile classique. Cet événement est détecté si le doigt est posé puis retiré avant un délai de 500 ms sans avoir été déplacé (Figure 1, transition "Tap"), temps qui correspond à celui utilisé par défaut pour un double-clic par la plupart des systèmes. La sélection peut être maintenue en conservant le doigt primaire immobile pendant 500ms (Figure 1, transitions "Timeout"). La sélection est désactivée par un nouveau *tap*.

Création et destruction de curseurs

La création d'un nouveau curseur se fait par un double tap sur la surface (Figure 1, transition "Double-tap"). Un nouveau curseur apparaît alors à l'écran à la position absolue correspondante. Lors de la création d'un nouveau curseur,

les curseurs non utilisés sont détruits. Un curseur est considéré comme non utilisé si aucun doigt ne lui est associé.

Association doigts / curseurs

L'association entre les doigts et les curseurs se fait en considérant les positions absolues des doigts et des curseurs. Un doigt est associé au curseur le plus proche dans les repères absolus. Lorsqu'un doigt est posé sur la surface, il est classé en doigt primaire ou secondaire. Si le doigt est plus proche d'un curseur non associé, il est considéré comme primaire (Figure 1, transition "Pose & prox(curseur non associé)"). S'il est plus proche d'un doigt primaire, il est considéré comme secondaire et ajouté en tant que curseur au curseur correspondant au doigt primaire (Figure 1, transition "Pose & prox(doigt primaire)"). Lors d'un débrayage, les curseurs débrayés deviennent attractifs pendant 500 ms : l'association est faite avec le plus proche de ces curseurs en priorité (Figure 1, transition "Pose & prox(curseur débrayé)"). Cette méthode permet de débrayer un curseur, passer d'un curseur à un autre et faire apparaître des curseurs à volonté.

DISCUSSION

La technique proposée permet de contrôler un curseur avec un doigt sans aucun apprentissage, puis favorise l'utilisation de plusieurs doigts pour la manipulation d'objets et enfin requiert un apprentissage modéré pour créer et détruire plusieurs curseurs. L'efficacité de la technique d'interaction proposée repose sur la capacité des utilisateurs à contrôler les curseurs désirés. Nous avons réalisé des tests informels avec une application de positionnement d'objets (Figure 2). Les résultats indiquent que les utilisateurs arrivent facilement à créer et détruire des curseurs. Surtout ils montrent que la méthode d'association doigts / curseurs est efficace. Sans explication de la méthode utilisée, les utilisateurs arrivent facilement à contrôler le curseur souhaité. Il peut arriver que l'utilisateur se trompe au moment de sélectionner un curseur ou que l'association échoue en ajoutant un curseur au mauvais curseur, mais l'erreur n'est pas pénalisante car elle peut être facilement corrigée.

Nos tests ont aussi montré que le "tap et demi" implémenté dans un premier temps pour le maintien de la sélection d'un curseur est jugé trop contraignant et que l'utilisation d'un clic long est plus aisée au prix d'une légère temporisation. Contrairement aux techniques de l'état de l'art, la répartition dynamique de la surface tactile permet de profiter de toute la surface lorsqu'on ne manipule qu'un ou plusieurs curseurs. Contrairement au *HandCursor* de Moscovich et Hughes, le mécanisme d'association doigt/curseur simplifie l'interaction bimanuelle indirecte : quand deux curseurs se croisent (e.g. le curseur contrôlé par la main gauche va à droite et inversement), chaque main pourra facilement reprendre le contrôle du curseur le plus proche dès qu'ils seront dé-associés et ré-associés.

CONCLUSION

Nous avons proposé une technique d'interaction permettant le contrôle de plusieurs curseurs de manière relative et indirecte, en utilisant un seul périphérique multi-points. Cette technique semble particulièrement adaptée pour interagir avec de grandes surfaces d'affichage en utilisant une surface de contrôle réduite. Elle permet d'interagir avec précision sans réduire la zone d'interaction à l'écran tout en profitant de l'intégralité de la surface tactile. Des tests préliminaires indiquent que l'association entre les doigts et les curseurs est facile à gérer. Des expériences complémentaires permettraient de valider ces premiers résultats.

REMERCIEMENTS

Ce travail a été financé par le projet ANR-09-CORD-013 "InSTInCT" et le Conseil Régional Nord-Pas de Calais.

BIBLIOGRAPHIE

1. Benko, H., Izadi, S., Wilson, A. D., Cao, X., Rosenfeld, D., and Hinckley, K. Design and evaluation of interaction models for multi-touch mice. In *Proc. of GI'10*, 253–260. CIPS, 2010.
2. Casiez, G., and Roussel, N. No more bricolage ! Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proc. of UIST'11*, 603–614. ACM, 2011.
3. Kin, K., Agrawala, M., and DeRose, T. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proc. of GI'09*, 119–124. CIPS, 2009.
4. Malik, S., Ranjan, A., and Balakrishnan, R. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proc. of UIST'05*, 43–52. ACM, 2005.
5. Moscovich, T., and Hughes, J. F. Multi-finger cursor techniques. In *Proc. of GI'06*, 1–7. CIPS, 2006.
6. Schmidt, D., Block, F., and Gellersen, H. A comparison of direct and indirect multi-touch input for large surfaces. In *Proc. of INTERACT'09*, 582–594. Springer, 2009.
7. Villar, N., Izadi, S., Rosenfeld, D., Benko, H., Helmes, J., Westhues, J., Hodges, S., Ofek, E., Butler, A., Cao, X., and Chen, B. Mouse 2.0 : multi-touch meets the mouse. In *Proc. of UIST'09*, 33–42. ACM, 2009.
8. Voidsa, S., Tobiasz, M., Stromer, J., Isenberg, P., and Carpendale, S. Getting practical with interactive tabletop displays : designing for dense data, "fat fingers," diverse interactions, and face-to-face collaboration. In *Proc. of ITS'09*, 109–116. ACM, 2009.
9. Yang, X.-D., Mak, E., McCallum, D., Irani, P., Cao, X., and Izadi, S. Lensmouse : augmenting the mouse with an interactive touch display. In *Proc. of CHI'10*, 2431–2440. ACM, 2010.