



HAL
open science

SmartCom: Exploiter un Smartphone pour Interagir avec les Barres d'Outils

Elio Keddiseh, Marcos Serrano, Emmanuel Dubois

► **To cite this version:**

Elio Keddiseh, Marcos Serrano, Emmanuel Dubois. SmartCom: Exploiter un Smartphone pour Interagir avec les Barres d'Outils. 32ème Conférence francophone sur l'Interaction Humain-Machine (IHM 20.21), Association Francophone d'Interaction Homme-Machine (AFIHM), Apr 2021, Metz (Virtual Event), France. pp.3:1-10, 10.1145/3450522.3451244 . hal-03567057

HAL Id: hal-03567057

<https://hal.science/hal-03567057>

Submitted on 11 Feb 2022

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.

SmartCom: Exploiter un Smartphone pour Interagir avec les Barres d'Outils

SmartCom: Using a Smartphone for Toolbar Interaction

Elio Keddissseh
Université Toulouse III – Paul
Sabatier – IRIT, Toulouse, France;
Oktal Sydac, Toulouse, France

Marcos Serrano
Université Toulouse III – Paul
Sabatier – IRIT, Toulouse, France

Emmanuel Dubois
Université Toulouse III – Paul
Sabatier – IRIT, Toulouse, France

ABSTRACT

In this article, we discuss the challenge of reducing mouse pointer transitions between the work object (e.g. a text document) and toolbars on desktop computers. To this end, we present SmartCom, an innovative approach to the selection of commands on toolbars, based on the use of both keyboard and touchscreen. The toolbar is displayed on the touch screen, which is placed beneath the keyboard. Users can select commands by performing gestures that combine pressing a key with the pinky finger and touching the screen with the thumb of the same hand. After analyzing the design properties of SmartCom, a preliminary study established that users could reach the entire surface of the touch screen with their thumb. Then, a user study revealed that direct pointing surpasses indirect pointing to select items on a simple toolbar displayed on the touch screen.

CCS CONCEPTS

• **Human-computer interaction (HCI)** → Interaction techniques; Gestural input.

KEYWORDS

Toolbar, Command selection, Object-to-command transition, Mobile device

RÉSUMÉ

Cet article s'intéresse à la réduction des transitions du pointeur de la souris entre l'objet de travail et les barres d'outils sur les ordinateurs de bureau. Nous présentons SmartCom, une approche innovante pour la sélection des commandes sur les barres d'outils, basée sur l'utilisation du clavier et d'un écran tactile, placé sous le clavier. La barre d'outils est affichée sur l'écran tactile. Les utilisateurs peuvent sélectionner des commandes en effectuant des gestes qui combinent l'appui sur une touche avec le doigt auriculaire et le toucher de l'écran avec le pouce de la même main. Après avoir analysé les propriétés de conception du SmartCom, une expérience préliminaire a établi que les utilisateurs peuvent effectuer de tels gestes et atteindre toute la surface de l'écran tactile avec leur

pouce. Ensuite, une étude utilisateur a révélé que le pointage direct surpasse le pointage indirect pour atteindre des éléments sur une simple barre d'outils affichée sur l'écran tactile.

MOTS-CLÉS

Barre d'outils, Sélection de commandes, Transition d'objet à commande, Appareil mobile

1 INTRODUCTION

Bon nombre des applications de bureau les plus utilisées aujourd'hui (par exemple, les suites Office ou Adobe) reposent sur l'utilisation de barres d'outils (e.g. le ruban Microsoft, les palettes outils d'Adobe), généralement situées sur un bord de la fenêtre de l'application. Toutefois, ces widgets occupent l'espace de l'écran et induisent une "transition d'objet à commande" [14] qui prend un temps précieux, et perturbe le flux d'interaction [2, 16] : les utilisateurs doivent déplacer le pointeur de la souris de l'objet de travail (par exemple, un texte dans MS-Word) vers la barre d'outils, puis revenir à l'objet de travail pour poursuivre leur tâche.

Les raccourcis clavier sont une alternative intéressante aux barres d'outils, mais ils nécessitent la mémorisation des raccourcis [26], ce qui conduit les utilisateurs à ne les employer que pour les commandes les plus fréquemment utilisées [21]. Les menus contextuels, tels que les «marking menus» [23], ne sont pas toujours visibles et ne peuvent contenir qu'une partie des éléments de la barre d'outils. De nouveaux gestes sur ou autour du clavier (par exemple FingerChord [39], Hot-Strokes [13]) peuvent faciliter l'accès aux commandes mais exigent des utilisateurs qu'ils apprennent et mémorisent les gestes. Enfin, l'augmentation des dispositifs de saisie actuels ou la conception de nouveaux dispositifs peut améliorer la sélection des commandes (par exemple, LensMouse [37] RPM [31] et TDK [8]). Cependant, cela nécessite que les utilisateurs mettent à niveau leur environnement de bureau habituel souris-clavier.

Nous proposons donc SmartCom, une solution originale pour la sélection d'éléments sur des barres d'outils, basée sur la combinaison du clavier avec un écran tactile (un smartphone dans notre implémentation). Avec SmartCom, les utilisateurs sélectionnent des commandes sur une barre d'outils affichée sur l'écran tactile,

IHM '20.21, April 13–16, 2021, Virtual Event, France

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Actes de la 32e conférence Francophone sur l'interaction Humain-Machine (IHM'20.21), April 13-16, 2021, Virtual Event, France.*

<https://doi.org/10.1145/3450522.3451244>

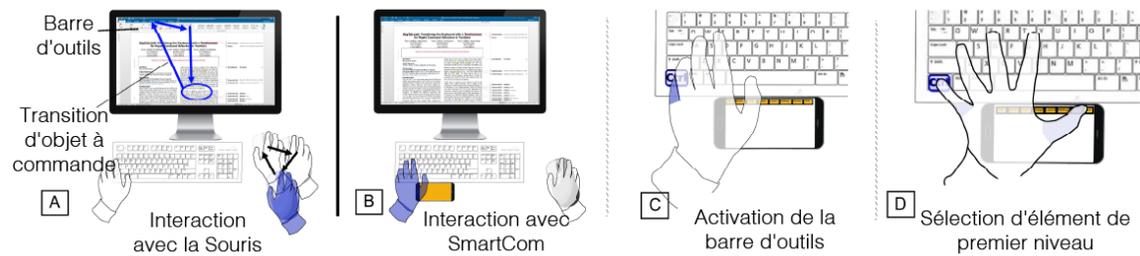


Figure 1: SmartCom réduit les transitions d'objet à commande (A) en affichant une barre d'outils sur un écran tactile sous le clavier (B). Avec SmartCom, le processus d'exécution des commandes commence par l'activation de la barre d'outils (C) par une pression sur la touche Ctrl ; puis une sélection d'élément de premier niveau (D) sur la barre d'outils de l'écran tactile.

par des gestes combinant l'appui sur les touches du clavier et le toucher sur l'écran. SmartCom offre plusieurs avantages : 1) réduire la transition entre l'objet et la commande, puisque le pointeur de la souris reste sur l'objet de travail ; 2) réduire l'espace occupé sur l'écran en déportant la barre d'outils sur l'écran tactile secondaire et 3) utiliser un dispositif qui est à la disposition de l'utilisateur, c'est-à-dire un smartphone ordinaire.

Dans une étude préliminaire, nous validons d'abord la capacité d'un utilisateur à atteindre l'écran tactile lors de l'utilisation combinée d'une touche clavier et un toucher sur l'écran tactile. Ensuite, nous comparons différentes techniques d'interaction utilisant SmartCom avec une technique de référence basée sur la souris pour la sélection des commandes. Dans une étude utilisateur, nous examinons si l'interaction avec l'écran tactile doit être basée sur le pointage direct ou indirect.

Nos contributions comprennent : 1) la conception de techniques d'interaction combinant un smartphone et un clavier pour la sélection des commandes ; 2) une étude préliminaire démontrant que 80 % de l'écran d'un smartphone de 5,8 pouces peut être atteint confortablement tout en utilisant le clavier ; 3) une étude utilisateur établissant la possibilité de sélectionner des commandes avec SmartCom plus efficacement qu'avec une souris (40 % de temps supplémentaire requis).

2 TRAVAUX EXISTANTS

Après avoir décrit les transitions objet-commande, nous discutons des travaux antérieurs sur la sélection des commandes à l'aide de dispositifs de saisie traditionnels, d'environnements de bureau améliorés et de combinaisons de clavier et d'écran tactile.

2.1 Transition d'objet à commande

Lors de la sélection de commandes sur les barres d'outils, l'utilisateur doit faire un mouvement de va-et-vient entre l'objet de travail et la commande souhaitée. Dillon *et al.* [14], ont décomposé la procédure de sélection des commandes en deux parties : 1) une transition de l'objet de travail à la commande, et 2) une transition de la commande à l'objet de travail. Bailly *et al.* [2] a identifié deux nouvelles étapes dans cette procédure : 1) une transition d'une commande à une autre lorsque les utilisateurs doivent sélectionner plusieurs commandes consécutives, et 2) une transition de la commande au widget d'intérêt lorsque les utilisateurs

doivent configurer des paramètres. Dans l'ensemble, malgré le peu de recherches effectuées dans le passé, il est admis que la transition d'objet à commande entrave l'interaction. En effet, elle interrompt le flux d'interaction et prend du temps [16].

2.2 Sélection des commandes avec les dispositifs de saisie traditionnels

De nombreux travaux ont été menés pour concevoir des techniques d'interaction visant à améliorer la sélection des commandes à l'aide des dispositifs de saisie traditionnels. Avec la souris, la solution la plus explorée est celle des menus contextuels [22] tels que les marking menus [38] ou les pie menus [10], qui ont l'inconvénient d'occulter la zone de travail. De plus, les menus contextuels sont limités en termes de nombre de commandes affichés, et ne sont à priori pas visibles sans une action explicite de l'utilisateur qui doit donc avoir une connaissance préalable de l'existence du menu contextuel et de son mécanisme d'activation. Les raccourcis de pointage, tels que les UIMarks [12], permettent de téléporter le pointeur de la souris vers des positions prédéfinies, mais nécessitent une personnalisation préalable pour définir les positions et les gestes pour chaque application. Avec un trackpad de multiples gestes peuvent être mis en correspondance avec des commandes, comme le montre MarkPad [18]. Mais en mode novice MarkPad induit aussi une occultation complète de l'interface utilisateur principale. Dans l'ensemble, aucune de ces solutions ne peut remplacer complètement la barre d'outils et ses avantages intrinsèques (grand nombre de commandes, disposition spatiale constante, pas besoin de mémoriser les gestes).

Les raccourcis clavier sont utiles pour les commandes les plus fréquentes, mais difficiles à mémoriser [26]. KeyMap propose d'afficher un clavier virtuel enrichi avec des étiquettes de commande sur ses touches [28] et limite ainsi le besoin de mémorisation. Mais l'approche n'a été testée que sur un petit ensemble de 13 commandes. ExposeHK [29] superpose des raccourcis clavier aux icônes de commande existantes lorsqu'une touche de modification est enfoncée. Cependant, dans sa version menu, ExposeHKM affiche tous les éléments du menu à l'écran lorsque le modificateur est appuyé, prenant ainsi une grande partie de l'espace à l'écran. Giannidakis *et al.* [19] a également mis en place une transformation visuelle des icônes de la barre d'outils pour afficher les raccourcis clavier correspondants. Appelée IconHK, cette transformation améliore la

sélection des commandes en améliorant la transition du novice à l'expert. Cependant, son intégration dans des applications réelles est complexe car elle nécessite de changer la barre d'outils originale et de concevoir la transformation pour chaque icône. Hotkey Palette [1] affiche un clavier virtuel sur l'écran contenant des raccourcis définis par les utilisateurs. De plus ces mêmes raccourcis fonctionnent différemment selon le contexte d'utilisation. Berthelley *et al.* [5] a proposé 4 techniques d'interaction basées sur la combinaison du clavier et du pavé tactile. Après un appui long sur la touche Fn du clavier, une grille (SpotPad) contenant des commandes s'affiche sur l'écran et l'utilisateur touche la zone correspondante sur le pavé tactile avec l'autre main pour faire une sélection. Néanmoins, ces techniques ne résolvent pas le problème de l'occultation de l'écran principal.

Enfin, CommandMaps [35] est une technique qui affiche une vue aplatie et stable de la barre d'outils lorsque l'utilisateur appuie sur une touche d'activation. Cette approche tire parti de la mémoire spatiale pour faciliter la sélection des commandes mais n'élimine pas la transition d'objet à commande.

Une autre solution consiste à combiner les dispositifs de saisie habituels, ce qui permet d'enrichir le vocabulaire de saisie. Cui et ses collaborateurs [13] ont mis au point une technique dans laquelle les utilisateurs appuient sur une touche du clavier et font ensuite un geste sur le pavé tactile pour sélectionner une commande. Cette approche élimine la nécessité de rechercher une commande dans les menus linéaires ou les barres d'outils et réduit le temps d'activation des commandes de 43% par rapport aux menus linéaires. Toutefois, cette technique est exécutée de manière bimanuelle, ce qui introduit un coûteux effet de transition de la main (homing) entre le clavier et le pavé tactile [11].

2.3 Environnements de bureau améliorés

Les extensions de bureau consistent à modifier la souris, le clavier ou l'environnement global du bureau.

Souris augmentées : De nombreuses souris augmentées ont été proposées pour enrichir le vocabulaire d'entrée. Nous décrivons deux approches étroitement liées à nos travaux qui 1) utilisent un affichage secondaire, comme dans LensMouse, ou 2) ont été conçues pour sélectionner des commandes dans la barre d'outils, comme dans Rolling-Menu. La LensMouse [37] est une souris traditionnelle avec un écran tactile sur le dessus. Ce dispositif peut être utilisé comme un écran supplémentaire pour déporter toutes les fenêtres auxiliaires de l'écran principal (palette d'outils, boîtes de dialogue). En utilisant le toucher direct, il réduit les mouvements du curseur pour interagir avec ces fenêtres. Par ailleurs, les auteurs ont montré que la séparation visuelle entre l'écran de la souris et l'écran principal n'affecte pas les performances de sélection. Rolling-Menu [16] a étudié l'avantage d'une souris hémisphérique pour sélectionner des éléments sur une barre de menu : les gestes de roulement sont exclusivement utilisés pour parcourir et sélectionner des éléments de menu sans affecter la position du curseur, ce qui permet de gagner jusqu'à 24% de temps par rapport à l'utilisation de la souris.

Claviers augmentés : Métamorphe [3] améliore la sélection sans les yeux en permettant d'appliquer de nouvelles actions aux touches du clavier (tels que tourner ou tirer). Le clavier à écran tactile [8] (TDK) ajoute un affichage dynamique et des touches tactiles au

clavier traditionnel. Le TDK permet de nouvelles techniques de sélection des commandes, comme un menu circulaire qui apparaît sur les touches adjacentes à la touche enfoncée. Zheng *et al.* ([39] a aussi mis en place des raccourcis sensibles aux doigts ([40], où le fait d'appuyer sur des touches ayant des postures différentes de la main entraîne des sélections de commandes différentes dans des menus à deux niveaux. Leurs techniques ont permis une sélection de commandes plus rapide et moins sujette aux erreurs, ce qui a encouragé l'utilisation de raccourcis.

Augmentation du bureau : Le Magic Desk [7] enrichit l'environnement du bureau (autour du clavier et de la souris) grâce à la projection d'un écran tactile. Ils ont montré que la sélection des commandes dans la zone située sous le clavier est la zone la plus efficace autour du clavier.

Le principal désavantage de toutes ces approches d'augmentation est qu'elles obligent les utilisateurs à remplacer des éléments de leur environnement de bureau habituel. Pourtant, ces travaux ont inspiré notre approche en établissant 1) les avantages de la réduction de la transition objet-commande, 2) l'effet non perturbant de l'utilisation d'un écran secondaire ainsi que sa meilleure position autour du clavier, et 3) l'intérêt des gestes autour du clavier pour la sélection des commandes.

2.4 Combinaison d'un écran tactile et d'un clavier

Récemment, le marché de l'informatique a vu l'apparition d'ordinateurs portables intégrant un écran tactile secondaire pour accéder aux commandes, soit au-dessus (comme la TouchBar d'Apple [41]), soit en-dessous du clavier. Le fait de placer le second écran tactile au-dessus du clavier, comme sur la TouchBar d'Apple, va toutefois à l'encontre des conclusions de [7] concernant les performances. En outre, pour atteindre l'écran tactile, il faut lever la main du clavier, ce qui peut éventuellement entraîner des pressions inattendues sur les touches, comme le soulignent des études antérieures sur des paramètres similaires [20].

Par ailleurs, en raison de sa taille limitée, les écrans tactiles situés au-dessus du clavier ne peuvent pas contenir autant d'icônes que le SmartCom. Quant aux écrans tactiles situés en-dessous du clavier, ils peuvent naturellement occuper la position du trackpad : Asus a remplacé le trackpad traditionnel par un écran tactile appelé ScreenPad [42]. De même, Razer a proposé une solution appelée Project Linda [43] qui permet de fixer un smartphone sur la position du trackpad. Récemment, Lee et ses collègues [27] ont étudié l'utilisation d'un écran tactile comme fenêtre de visualisation pour modifier, avec un stylo, une zone d'intérêt de l'écran principal de l'ordinateur portable. Cependant, à notre connaissance, il n'existe aucune étude sur la meilleure façon de concevoir un menu pour ces appareils et sur les avantages de la sélection des commandes en termes de temps de sélection ou de taux d'erreur. En outre, notre approche étend l'utilisation de l'écran tactile des ordinateurs portables à tout environnement de bureau.

3 PROPRIÉTÉS DE CONCEPTION DU SMARTCOM

Dans cette section, nous présentons les grands principes de SmartCom sur la manière de combiner un clavier et un écran tactile pour

sélectionner des commandes dans les barres d'outils. Nous donnons d'abord un aperçu de la configuration spatiale de SmartCom, puis nous discutons de l'utilisation de la main, et enfin nous analysons le processus d'exécution des commandes avec SmartCom.

3.1 Aménagement de l'espace

La combinaison d'un écran tactile et d'un clavier soulève d'abord la question de l'organisation spatiale des deux dispositifs. Bi *et al.* [7] ont déjà établi que les utilisateurs effectuent de meilleures interactions mono-manuelles en-dessous du clavier, plutôt qu'au-dessus ou à côté de celui-ci. En outre, nous avons effectué des tests informels qui ont écarté la position au-dessus, bien qu'Apple ait choisi la barre tactile, car elle oblige les utilisateurs à éloigner la main du clavier. Placer l'écran sur la souris, comme dans LensMouse [37], empêcherait son utilisation pendant la saisie du texte. Nous avons donc placé l'écran tactile sous le clavier (cf. Figure 1).

Le positionnement de la barre d'outils est aussi une propriété de conception. Lors de l'utilisation de SmartCom, l'écran de bureau et l'écran tactile peuvent être utilisés pour afficher la barre d'outils. Pour rendre les raccourcis plus mémorisables, la représentation visuelle du menu doit correspondre le plus possible aux actions physiques de l'utilisateur [28] : cette correspondance renforce la mémoire spatiale. C'est pourquoi nous avons choisi d'afficher la barre d'outils sur l'écran tactile, de sorte que les actions physiques de rendu et d'entrée soient situées au même endroit, maximisant de fait la correspondance.

3.2 Usage des mains

Nous pouvons différencier deux situations principales dans lesquelles un utilisateur utiliserait SmartCom pour sélectionner des commandes : pendant l'interaction avec la souris ou pendant la saisie de texte. *Pendant l'interaction avec la souris*, la main dominante (DH) est sur la souris tandis que la main non dominante (NDH) est généralement sur le clavier pour exécuter des raccourcis. Dans ce cas, une utilisation de SmartCom avec la main NDH est plus appropriée, afin d'éviter tout effet de recherche sur la DH (c'est-à-dire les mouvements de la souris vers le clavier). *Lors de la saisie de texte*, les deux mains sont déjà au-dessus du clavier. Dans ce cas, l'utilisateur pourrait interagir avec les deux mains, ce qui pourrait potentiellement augmenter le vocabulaire de saisie de SmartCom.

3.3 Exécution des commandes avec SmartCom

Pour concevoir les techniques de sélection des commandes avec SmartCom, nous nous sommes concentrés sur les trois étapes du processus d'exécution des commandes : 1) activation de la barre d'outils, 2) navigation dans la barre d'outils, et 3) validation de l'élément sélectionné.

Méthode d'activation : Pour activer la barre d'outils avec SmartCom, nous avons décidé de nous appuyer sur un quasimode [33], c'est-à-dire un mode spécifique du système qui se termine automatiquement une fois que la sélection d'un élément est effectuée. Pour activer ce quasimode, nous avons choisi d'utiliser la touche Ctrl : pour afficher la barre d'outils sur l'écran tactile, l'utilisateur appuie sur la touche Ctrl et la maintient enfoncée à l'aide de son auriculaire, le doigt le plus utilisé pour cette touche [17]. L'utilisation de la touche Ctrl est compatible avec les raccourcis claviers habituels

: les utilisateurs peuvent toujours effectuer des combinaisons de touches lorsque le SmartCom est activé.

Mécanisme de navigation : Une fois que le SmartCom est activé, pour atteindre la commande ciblée, l'utilisateur doit naviguer parmi les éléments de l'écran tactile avec la même main que celle tenant la touche Ctrl. La pression sur la touche Ctrl avec l'auriculaire limite le mouvement de la main et donc permet l'utilisation des autres doigts sur l'écran tactile : le pouce est donc a priori le meilleur candidat pour assurer une couverture confortable et large de l'écran tactile (cf. Figure 1).

Validation d'une sélection d'un élément de menu : Dans l'étape finale, l'utilisateur doit valider l'élément sélectionné dans la barre d'outils. Nous avons simplement considéré que la validation se produit lorsque l'utilisateur relâche la touche Ctrl, et entraîne également l'exécution de la commande correspondante. Ce choix de conception permet à l'utilisateur de modifier ou d'annuler sa sélection avant de valider. De plus, le fait de relâcher la touche Ctrl met explicitement fin au quasimode et ramène le focus sur l'objet de travail.

Avant d'explorer ces propriétés de conception pour créer des techniques d'interaction SmartCom permettant de sélectionner des commandes, nous menons une étude d'accessibilité pour valider que les utilisateurs peuvent atteindre confortablement l'écran tactile avec le pouce tout en appuyant sur une touche.

4 ÉTUDE PRÉLIMINAIRE : ATTEIGNABILITÉ DES ÉCRANS TACTILES

L'étude préliminaire vise à mesurer la taille de la zone de l'écran tactile que l'utilisateur peut atteindre avec le pouce, tout en appuyant sur une touche de modification du clavier avec l'auriculaire de la même main.

4.1 Protocole expérimental

Tâche et instruction : Nous avons choisi de distinguer deux zones que le pouce peut atteindre, telles que définies dans [30] : la zone de confort et la zone utile. La zone de confort est celle que l'utilisateur peut atteindre en déplaçant uniquement les articulations du pouce, tandis que la zone utile est celle que l'utilisateur peut atteindre en déplaçant également la paume de la main. La tâche consistait d'abord à appuyer et maintenir une touche de modification avec l'auriculaire, puis à appuyer et relâcher une touche intermédiaire (nous avons testé avec et sans touche intermédiaire), et enfin à toucher et balayer l'écran tactile avec le pouce, à l'intérieur de la zone de confort ou utile selon les instructions. L'utilisation de la touche intermédiaire visait à évaluer l'effet potentiel de l'exécution de combinaisons de touches plus complexes sur l'accessibilité de l'écran. En effet si les commandes exécutables via SmartCom sont disposées sous la forme d'un menu hiérarchique à deux niveaux, les deux niveaux peuvent être affichés sur l'écran tactile ou seulement le second, le premier restant alors contrôlé par des raccourcis clavier nécessitant l'usage d'une touche intermédiaire.

Participants : Nous avons recruté 12 étudiants du laboratoire de recherche local (3 femmes, 2 gauchers), âgés en moyenne de 26 ans (ET = 3.43), avec une taille moyenne de main de 10.1 cm (ET = 0.67, ce qui correspond à la mesure 2 en [44]) et une taille moyenne de pouce de 6 cm (ET = 0.54).

Apparatus : Nous avons utilisé une souris Microsoft traditionnelle, un clavier QWERTY (disposition britannique) et un écran de 24 pouces (1920x1080, 96dpi). Pour mesurer la zone accessible, nous avons utilisé une tablette Galaxy Tab S4 avec un écran de 10,1", connectée au PC par des sockets TCP. L'objectif de l'utilisation d'un écran tactile aussi grand était de pouvoir obtenir des résultats de couverture de l'écran tactile par le pouce pour différentes tailles de smartphones (jusqu'à 10 pouces). La tablette était centrée entre les deux touches Ctrl du clavier. Pour éviter tout contact indésirable avec la paume de la main, les participants portaient un gant sans doigt. Nous avons fourni un retour visuel sur la tablette en traçant les points de contact en bleu. Pour garantir un setup commun à tous les participants et éviter les déplacements de l'écran tactile, nous avons positionné fermement l'écran contre le clavier avec du Patafix.

Design : Cette étude préliminaire a suivi une conception 2x6x2 intra-sujets avec comme facteurs Main (gauche et droite), Combinaison de touches (6 combinaisons possibles), et Type de zone (Confort et Utile). En effet, nous avons voulu comparer l'atteignabilité de l'écran tactile avec le pouce lors de l'utilisation de la main droite et de la main gauche. Nous avons également considéré l'usage de la touche Ctrl seule, ainsi que sa combinaison avec 5 touches intermédiaires uniformément réparties sur la droite ou la gauche du clavier selon la main utilisée (cf. Figure 2). Nous avons limité l'illustration à la main gauche uniquement, faute de place et de façon à privilégier les conditions utilisées dans la suite de l'article. Toutes ces combinaisons ont été répétées 3 fois. En conséquence, chaque participant a effectué 72 essais (2 mains x 6 combinaisons de touches x 2 types de zones x 3 répétitions). Nous avons contrebalancé le facteur Main et randomisé les combinaisons de touches. Pour chaque combinaison de touches, les participants ont toujours commencé par la zone de confort, puis sont passés à la zone utile. Les participants ont suivi une phase d'entraînement avec chaque technique avant de commencer avec chaque main.

Données collectées : Pour chaque essai, nous avons enregistré les coordonnées de contact et nous avons déduit la taille des zones atteignables grâce à l'algorithme «Convex Hull »[32]. À la fin de l'expérimentation, nous avons demandé aux participants de remplir un questionnaire à base d'échelle de Likert mesurant la difficulté à effectuer des gestes avec chaque main et pour chaque zone.

4.2 Résultats

Nous présentons des résultats en termes de taille des zones Confort et Utile et, pour chacune d'elles, nous détaillons le résultat pour la main gauche et la main droite. Nous présentons ensuite les résultats subjectifs.

Zone Confort et zone Utile : Les zones Confort et Utiles atteintes lors d'une utilisation avec la main gauche sont illustrées dans la Figure 2. Nous avons calculé leurs tailles en cm² pour chaque main : la zone de confort de la main gauche couvre 61 cm² (~72% de l'affichage 5.8"), tandis que la zone de confort de la main droite couvre 74 cm² (~87% de l'affichage 5.8"). La surface utile couvre 180 cm² pour la main gauche et 177 cm² pour la main droite. D'après les résultats, la taille des zones reste très similaire pour chaque combinaison de touches : le fait d'appuyer (et de relâcher) une touche intermédiaire avant de balayer l'écran tactile ne semble

pas affecter l'agilité des participants : nous avons en effet observé durant l'expérimentation que les participants relâchaient la touche avant d'aller toucher SmartCom (i.e. Appuyez sur CTRL -> Appuyez sur Q -> Relâchez Q -> Atteignez la tablette avec le pouce). La zone atteignable n'est donc pas affectée. Pour rendre nos résultats plus pratiques pour les concepteurs et les lecteurs, nous avons superposé trois contours de smartphones (Huawei P10 Lite, Samsung Galaxy S8, et Huawei Mate 20) en fonction de leur taille d'écran (5.2", 5.8" et 6.5").

Zones privilégiées : Pour affiner les résultats dans la zone de confort, nous avons examiné la répartition des points de contact. Nous avons construit une carte de densité pour identifier les zones les plus touchées (voir la Figure 2 pour la carte de densité de gauche). Nous avons observé que les zones les plus touchées par la main gauche sont situées sur la gauche et le quart supérieur des smartphones représentés sur la Figure 2. Nous pensons que cela indique la zone la plus facile à atteindre, et donc pourrait être un bon endroit pour placer les commandes les plus fréquemment utilisées.

Retour d'information subjectif : Nous constatons tout d'abord qu'aucune des conditions n'a été considérée comme très difficile à remplir. Seuls deux participants ont indiqué que l'interaction sur la zone Utile était plutôt difficile. Ce résultat est très encourageant car il établit que SmartCom peut être utilisé facilement ou très facilement pour atteindre la zone de confort.

Résumé : La plus grande partie de la surface d'un écran tactile de 5.8 pouces placé en-dessous d'un clavier peut être confortablement atteinte avec le pouce tout en maintenant la touche Ctrl, quelque soit la main. Pour assurer la validité externe et écologique de nos études, nous avons choisi d'explorer plus SmartCom en utilisant un smartphone de 5.8" avec la main non dominante uniquement, la main dominante reposant sur la souris.

5 TECHNIQUES D'INTERACTION SMARTCOM

À partir des propriétés de conception initiales et des résultats de l'étude préliminaire, nous avons conçu différentes techniques d'interaction pour la manipulation de la barre d'outils d'une seule main avec SmartCom. Pour toutes ces techniques, SmartCom est activé lorsque l'utilisateur tient la touche Ctrl. Lorsque l'utilisateur relâche la touche Ctrl, l'élément en cours de sélection est validé, mettant fin au processus d'exécution de la commande comme expliqué ci-dessus.

Les barres d'outils que nous considérons ne contiennent qu'un seul niveau d'éléments. Dans ce cas, le principal défi de conception est de savoir comment faciliter l'accès à tous les éléments de la barre d'outils : l'étude préliminaire a montré que l'accès à la partie droite de l'écran tactile nécessitait un geste dépassant le confort. C'est pourquoi nous avons conçu deux techniques basées sur le pointage direct ou indirect : le pointage indirect pourrait permettre d'atteindre les cibles sur la partie droite de l'écran tactile avec un geste plus petit et plus confortable, comme l'illustre la Figure 2. Nous détaillons les deux approches ci-dessous.

Manipulation directe : SmartCom Touch (SCT). SmartCom Touch (SCT), est basée sur la manipulation directe pour sélectionner des éléments de la barre d'outils. Une fois que SmartCom est activé

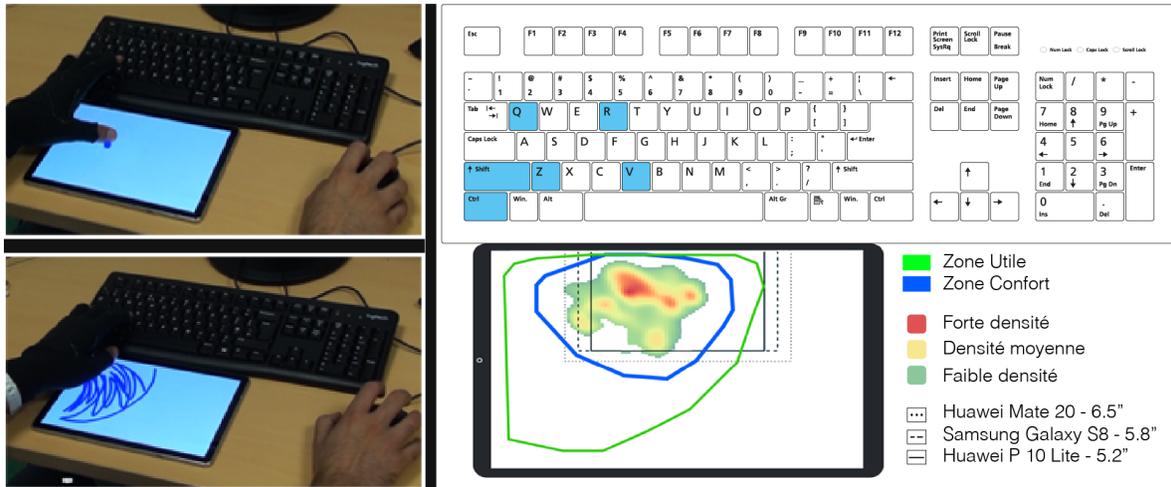


Figure 2: Gauche : Utilisateur touchant l'écran en zone de Confort (haut) et zone Utile (bas). Droite : Répartition des touches utilisées sur le clavier pour la main gauche (haut). Illustration des zones utile et de confort pour la main gauche sur la tablette : Les rectangles représentent les tailles des smartphones ; La carte de densité dénote les zones les plus touchées pour la main gauche.

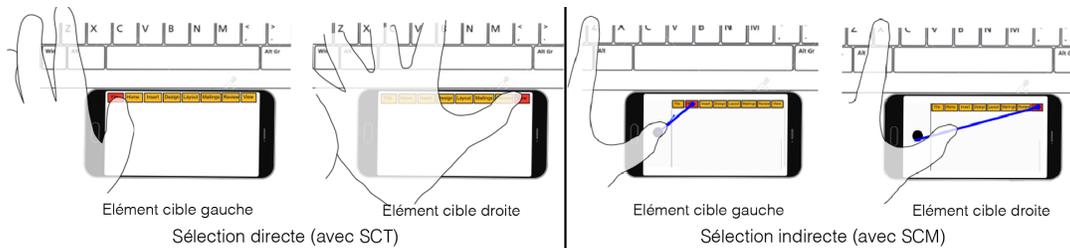


Figure 3: Sélection directe et indirecte des éléments de la barre d'outils.

et la barre d'outils affichée, les utilisateurs touchent directement l'élément de menu cible sur l'écran tactile (cf. figure 3).

Concernant la disposition visuelle de la barre d'outils, et conformément aux résultats de l'étude préliminaire, la barre d'outils doit être positionnée le plus près possible du clavier pour favoriser des gestes confortables. Ainsi, la barre d'outils est positionnée en haut de l'écran tactile, laissant le reste de l'écran pour accueillir éventuellement une prévisualisation de l'effet de la commande ou les éléments d'un potentiel second niveau de menu, hors de l'étude présentée dans cet article.

Comme cette technique repose sur le toucher direct, il est important de définir une taille de cible appropriée. Sur un smartphone de 5.8" en mode paysage, le système Android permet d'afficher jusqu'à 5 colonnes d'icônes avec leurs labels, tout en garantissant un usage confortable et précis. Sans les labels, cela signifie que la barre d'outils peut facilement contenir 10 grandes icônes (72x72 px) ou 12 icônes moyennes (48x48 px) sur une ligne

Manipulation indirecte : SmartCom Magnet (SCM) : SmartCom Magnet (SCM), est basée sur le pointage indirect pour la sélection des éléments de la barre d'outils. Sa conception s'inspire de MagStick [34] et Extended Thumb [25] : une fois que SmartCom est activé et la barre d'outils affichée, l'utilisateur pose son pouce

gauche sur un point noir placé sur le côté gauche de l'écran tactile et déplace son pouce vers la cible ; un pointeur virtuel se déplace dans la même direction mais plus loin avec un rapport prédéfini empiriquement de 1:3. En outre, lorsque le pointeur virtuel est proche d'un objet, le curseur magnétisé se colle au centre de l'objet (cf. figure 3).

Comme pour SCT, la barre d'outils est positionnée en haut de l'écran tactile, laissant le reste de l'écran pour des informations complémentaires selon l'usage envisagé. Toutefois, par rapport à la version tactile directe, les icônes sont plus petites car il y a un espace vide sur la gauche dédiée à la zone de départ du geste. Nous avons utilisé le côté gauche car il est le plus proche du pouce lorsque l'on utilise la main gauche.

6 ÉTUDE : SÉLECTION D'ÉLÉMENTS DE MENUS DANS UNE BARRE D'OUTILS

L'objectif de cette étude est d'évaluer les performances des techniques SmartCom directe (SCT) et indirecte (SCM) pour sélectionner un élément de menu dans une barre d'outils, constituée d'une seule ligne d'éléments et affichée sur l'écran tactile. Nous les comparons avec l'utilisation de la souris pour sélectionner un élément sur une simple barre d'outils sur l'écran du PC.

6.1 Protocole expérimental

Tâche et instructions : La tâche consistait à sélectionner un élément sur une barre d'outils située soit sur l'écran du PC (pour la condition de la souris), soit sur l'écran tactile (pour les conditions SmartCom). Dans les environnements de bureau, la sélection d'un élément se fait normalement pendant que l'utilisateur se concentre sur la zone de travail d'un document affiché sur l'écran du PC. Nous avons donc veillé à ce que les essais commencent toujours lorsque l'utilisateur regarde l'écran du PC. Pour toutes les conditions, les participants ont été invités à sélectionner l'élément cible aussi rapidement et précisément que possible.

Techniques d'interaction : Pour les essais avec la souris, nous avons affiché une barre d'outils (de la même taille que la barre d'outils MS Word) en haut de l'écran du PC, avec l'élément cible mis en évidence en bleu pour éviter tout biais dû à la nécessité de mémoriser sa position, de lire un label ou de reconnaître une icône. Pour démarrer un essai, les participants devaient utiliser la souris avec leur main dominante pour cliquer sur un bouton de démarrage situé au centre de l'écran du PC. Ensuite, les participants devaient sélectionner la cible, et enfin cliquer sur un bouton de fin situé au centre de l'écran, indiquant la fin de l'essai. Le bouton de début/fin représente la zone de travail, c'est-à-dire l'objet dans la transition objet-commande, une approche déjà adoptée lors d'expériences précédentes [16]. Le bouton était un cercle de 2.7 cm (102 px).

Pour les essais avec SmartCom, après avoir cliqué sur le bouton de démarrage avec la souris, une instruction s'affichait sur l'écran principal, demandant aux participants d'appuyer et de maintenir le bouton Ctrl gauche avec leur auriculaire pour activer la barre d'outils. Cliquer sur le bouton de démarrage n'était pas possible dans le cas où l'utilisateur avait déjà appuyé sur le bouton Ctrl gauche. La barre d'outils était alors affichée sur l'écran tactile, la cible étant surlignée en bleu pour les mêmes raisons qu'exposées ci-dessus. Les participants sélectionnaient la cible sur l'écran tactile en utilisant une des deux techniques SmartCom. Le curseur de la souris ne se déplace pas lors de l'utilisation de SmartCom, il n'est donc pas nécessaire de remettre le curseur de la souris dans sa position initiale à l'issue de l'essai.

Conception de la barre d'outils SmartCom : Le raisonnement qui sous-tend la conception de la hauteur de la barre d'outils SmartCom est qu'un smartphone de 5,8 pouces en mode paysage peut facilement accueillir 6 rangées d'icônes sans label. Nous avons donc conçu une barre d'outils dont la hauteur correspond à un sixième de la hauteur du smartphone (1 cm - 142 px) en mode paysage. En ce qui concerne le nombre d'éléments dans une rangée, comme nous l'avons dit précédemment, la barre d'outils peut facilement contenir 10 ou 12 icônes, nous avons donc décidé d'évaluer 5 barres d'outils composées de 8, 10, 12, 14 et 16 éléments : 8 étant une largeur de menu fréquente dans la littérature, nous avons ajouté jusqu'à 16 éléments pour tester les limites de performance. La largeur des éléments pour **SCT** variait entre 1.5 cm (213 px) (8 éléments) et 0.7 cm (100 px) (16 éléments), tandis que pour **SCM**, en raison de l'espace vide à gauche, la largeur variait entre 1.2 cm (170 px) et 0.6 cm (85 px). Les barres d'outils de la souris avaient une hauteur de 0.8 cm (30 px) et une largeur variante entre 6.4 cm (242 px) (8 éléments) et 3.2 cm (121 px) (16 éléments).

Participants : Nous avons recruté 12 participants (2 femmes), âgés en moyenne de 27.1 ans (ET=2.01), avec une taille moyenne de main de 9.5 cm et une taille moyenne de pouce de 5.7 cm (mesurée comme dans l'étude préliminaire). Ils utilisent en moyenne le clavier et la souris pendant 52.75 heures/semaine et étaient tous des étudiants du laboratoire d'informatique local. Aucun d'entre eux n'a participé à l'étude précédente.

Apparatus : Nous avons utilisé le même apparatus et le même environnement matériel que dans l'étude préliminaire, à l'exception de l'écran tactile : nous avons utilisé un Samsung Galaxy S8 avec un écran de 5.8" (1440x2960, 360 dpi). Le smartphone était centré entre les deux touches Ctrl et collé au clavier. Les participants ne portaient pas de gants, contrairement à ce qui était prévu dans l'étude préliminaire.

Conception de l'étude : Notre étude a suivi une conception 3x5x4 intra-sujet avec comme facteurs la technique d'interaction (souris, SmartCom Touch - **SCT**, et SmartCom Magnet - **SCM**), le nombre d'éléments dans la barre d'outils (8, 10, 12, 14 et 16) et la position de l'élément à sélectionner dans la barre de menu (gauche, milieu gauche, milieu droit et droite). Les trois techniques ont été contrebalancées par un carré latin 3x3, et pour chaque technique, le nombre d'éléments a été présenté par ordre croissant de complexité. Un bloc consistait en la combinaison d'une technique d'interaction avec un nombre d'éléments et a été répété 4 fois. Chaque bloc comprenait les quatre positions de la cible (gauche, milieu gauche, milieu droit et droite) et ordonnées de manière aléatoire. Au total, chaque participant a effectué 240 essais (3 TI x 5 nombre d'éléments x 4 positions cibles x 4 répétitions). Avant de commencer chaque bloc, les participants ont fait une série d'essais de formation, où ils devaient successivement et avec succès sélectionner les quatre cibles à deux reprises.

Données collectées et mesurées : Pour chaque essai, nous avons enregistré tous les événements de saisie de la souris, du clavier et du smartphone. Le temps de réalisation des essais de la souris a été calculé à partir du clic du bouton de démarrage jusqu'au clic du bouton de fin. Pour les essais de SmartCom, le temps d'achèvement a été mesuré à partir du clic du bouton de départ avec la souris, jusqu'au relâchement de la touche Ctrl. Si la sélection de l'élément sur la barre d'outils simple était incorrecte, une erreur était ajoutée aux logs et les participants devaient continuer à essayer jusqu'à ce qu'ils sélectionnent le bon élément. Pour les mesures subjectives, les participants ont rempli un questionnaire SUS [9] après chaque technique et ont classé les techniques par ordre de préférence à la fin de la session d'expérimentation.

Analyse des données : Pour l'analyse des données, nous nous sommes appuyés sur des techniques d'estimation avec des intervalles de confiance de 95% et sur l'analyse des ratios, comme le recommande l'APA [36]. De plus, à l'aide de R, nous avons appliqué une transformation logarithmique aux données sur le temps de réalisation pour obtenir une distribution normale. Nous avons ensuite effectué un t test et généré la moyenne et les IC à 95% comme recommandé par l'APA [15]. Cette approche nous a permis de calculer une moyenne géométrique et de réduire l'effet des valeurs aberrantes [15]. Pour les lecteurs plus familiers des p-values, un parallèle peut être établi avec notre présentation des résultats (voir la figure 3 dans [24]). Les scripts utilisés pour calculer la moyenne géométrique et les intervalles de confiance proviennent de [6].

6.2 Résultats

Durée de réalisation : Les résultats montrent que la sélection d'un élément avec **SCT** prend moins de temps (1281ms, IC [1140, 1440]) qu'avec la souris (1804ms, IC [1662, 1958]) et **SCM** (1826ms IC [1666, 2001])(cf. Figure 4.). L'analyse intra-sujet, basée sur les rapports de temps Souris/SCT et **SCM/SCT**, permet de mieux quantifier cette différence pour chaque participant (Figure 4 - bas) et confirme qu'il y a une nette différence entre le temps de réalisation avec **SCT** et les autres techniques (ratios > 1) : l'utilisation de la Souris ou de **SCM** prend environ 40% de temps de plus que l'utilisation de **SCT** (Souris/SCT ~ SCM / SCT ~ 1,4).

Ces résultats demeurent vrais quel que soit la taille de la barre d'outils (Figure 5 - gauche) : Pour toutes les tailles de barre d'outils, la sélection d'un élément avec **SCT** nécessite clairement moins de temps qu'avec la souris ou **SCM**. De plus pour une technique donnée, les résultats ne montrent pas de différence en termes de temps d'exécution entre les différentes tailles de barre d'outils.

En outre, nous avons analysé le temps d'achèvement de la tâche par rapport à la position cible sur la barre d'outils (gauche, milieu gauche, milieu droit ou droite). Les avantages de **SCT** par rapport à la souris et **SCM** demeurent valables pour toutes les positions cibles, sauf entre **SCT** et **SCM** pour lesquelles il n'existe pas une différence aussi nette pour la cible de droite (Figure 5 - droite). Il semble également y avoir une autre tendance intéressante, bien qu'elle ne soit pas clairement établie par les mesures : avec **SCT**, les cibles du milieu semblent prendre moins de temps à sélectionner que celles de gauche, et même moins que celles de droite. Dans une moindre mesure, la même tendance est également observée pour la souris. Il est intéressant de noter qu'avec le **SCM**, il semble y avoir une tendance opposée (c'est-à-dire que la sélection des éléments à gauche ou à droite prend moins de temps que celle des éléments du milieu). Ce résultat n'est pas surprenant pour le **SCT**, car nous avons observé que pour atteindre les cibles de gauche, il faut rétracter le pouce, tandis que pour atteindre les cibles de droite, il faut étirer le pouce ; il en va de même pour la souris, car les cibles de gauche et de droite sont plus éloignées de la position de départ de la souris. En ce qui concerne le **SCM**, ce résultat tend à valider notre raisonnement de conception car il semble faciliter l'atteinte de cibles éloignées.

Précision : Les taux de précision varient entre 99.9% (IC [99.69, 100]) pour la souris, 98.5% (IC [97.81, 99.27]) pour **SCT** et 94.6% (IC [92.08, 96.56]) pour **SCM**, qui est clairement moins précis que la souris et **SCT**.

Mesures subjectives : En ce qui concerne les scores SUS, la souris a obtenu un score moyen de 80, **SCT** un score de 76 et **SCM** un score de 71. Cela correspond à un "bon" score d'utilisabilité pour la souris et **SCT**, et à "OK" pour **SCM** [4]. Six participants ont classé la souris comme la meilleure technique, tandis que **SCT** a été classé quatre fois en tête, et **SCM** seulement deux fois. La préférence pour la souris, malgré l'efficacité du **SCT**, peut s'expliquer par la familiarité des participants avec la souris.

6.3 Résumé

Notre étude a révélé que **SCT** est la meilleure technique en termes de temps de réalisation, et qu'elle est aussi bonne que la souris en termes de précision : sur une barre d'outils, **SCT** permet de

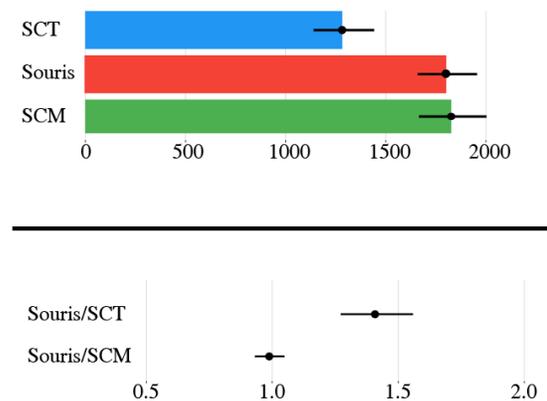


Figure 4: Temps moyen de sélection d'un élément de la barre d'outils (ms) pour les trois techniques (Haut). Ratios entre le temps de la souris et les techniques SmartCom (Gauche-bas). Toutes les barres représentent des IC à 95 %.

sélectionner jusqu'à 16 éléments en moins de 1.5s, offre une précision supérieure à 98% et qu'avec la souris le temps requis serait supérieur de 40%. Le temps de réalisation et la précision de **SCM** sont nettement moins bons que ceux de **SCT**. Enfin, il apparaît que les performances de **SCT** ne sont pas affectées par le nombre d'éléments.

7 DISCUSSION ET PERSPECTIVES

Dans ce travail, nous avons établi que la sélection de commandes dans une barre d'outils à l'aide de la souris nécessite 40 % de temps en plus qu'avec les techniques d'interaction SmartCom. **SCT** et **SCM** ont montré les mêmes niveaux de précision que la souris. Compte tenu de l'importance des barres d'outils dans les applications quotidiennes, et du fait que les interactions avec les barres d'outils sont assez fréquentes, cela pourrait représenter un gain de temps important pour les utilisateurs, et donc améliorer leur flux d'interaction global. En outre, des travaux antérieurs ont montré que les techniques exploitant la mémoire spatiale et kinesthésique de l'utilisateur [28] aident à mémoriser les commandes : cela pourrait s'appliquer à notre cas, étant donné le mapping parfait entre la disposition des cibles affichée et les gestes à accomplir pour les atteindre : en effet les positions des cibles sur l'écran tactile sont placées à l'endroit où la saisie directe par le toucher doit avoir lieu. À l'avenir, nous prévoyons d'étudier la mémorisation et la découverte des commandes lors de l'utilisation de SmartCom, maintenant que ses avantages en termes de performances sont établis.

Dans notre étude, 2 participants ont suggéré que l'écran tactile pourrait être intégré sur une surface plane, comme les trackpads des ordinateurs portables. Nous envisageons qu'un simple coussin repose-poignets autour de l'écran tactile permettrait de réaliser cette intégration à faible coût. Pour éviter toute interruption (appel/sms) lors de l'utilisation de SmartCom avec un smartphone personnel, les utilisateurs peuvent dévier les appels et les messages vers le PC (cette option est déjà intégrée dans MacOS et iOS et disponible pour Android et Windows via des applications dédiées). Notre implémentation de SmartCom utilise un socket TCP standard

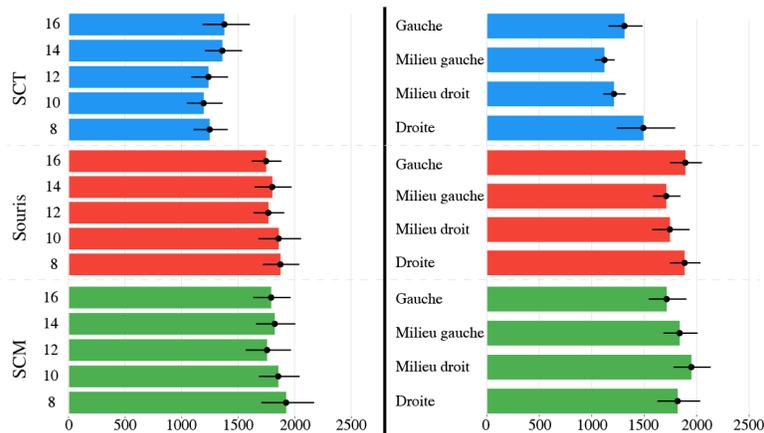


Figure 5: Affinement par technique, en fonction du nombre d'éléments (Gauche) et de la position de la cible (Droite). Toutes les barres représentent des IC à 95 %.

(sur Wi-Fi) pour la communication entre les deux appareils. L'usage du NFC ou Bluetooth peut facilement être envisagé : une application dédiée permettrait un appairage des deux appareils et la consommation de la batterie n'est pas un problème car le smartphone pourrait simplement être branché en USB sur le PC.

Dans notre choix de design des techniques SmartCom, nous avons adopté l'utilisation de quasimode pour éviter l'activation involontaire de commandes sur l'écran tactile lors de l'utilisation du clavier. De plus, l'utilisation de quasimode ne compromet pas l'utilisation de raccourcis clavier classiques (CTRL C, CTRL V, . . .). Par extension, l'utilisation de SmartCom pourrait consister en une combinaison de raccourcis clavier (pour le premier niveau de menu) et de 'taps' sur l'écran tactile (pour le second niveau) : c'est pourquoi nous avons inclus dans l'étude préliminaire l'utilisation d'une seule touche ou de touches multiples. Enfin l'utilisation du quasimode peut servir à appairer l'écran tactile au clavier uniquement au moment souhaité, laissant possible l'utilisation de l'écran tactile pour autre chose entre temps (e.g. répondre au téléphone). Cependant, SmartCom sans quasimode est également intéressant car il réduirait le temps de sélection des commandes.

Au-delà de l'exploration déjà mentionnée des questions d'apprentissage et de découverte, nous étudierons l'utilisation de SmartCom avec les deux mains en parallèle, ce qui pourrait avoir un sens lors de la saisie de texte sur le clavier. Dans ce cas, les avantages de SmartCom pourraient être encore plus importants : non seulement SmartCom faciliterait la sélection des commandes de la barre d'outils, mais il éviterait également tout effet de "homing" (i.e. le déplacement de la main droite du clavier vers la souris). Nous prévoyons d'étudier SmartCom avec des barres d'outils plus complexes en particuliers des barres d'outils à deux niveaux, i.e. un premier niveau dont chaque élément permet l'accès à une barre d'outils distincte : c'est typiquement le ruban de Windows. Nous analyserons aussi l'usage de barres d'outils contenant des listes, des curseurs et d'autres widgets courants. Enfin, nous prévoyons d'étudier aussi l'impact de l'utilisation de SmartCom sur une longue période et des solutions alternatives pour l'ancrage et la connexion

de SmartCom avec le clavier, comme un connecteur magnétique, similaire aux cordons Mac par exemple.

8 CONCLUSION

Dans ce travail, nous avons présenté SmartCom, une approche alternative à la sélection régulière de commandes sur les barres d'outils, qui exploite la combinaison unique d'un clavier et d'un écran tactile. Notre approche minimise la transition entre l'objet de travail et la commande en déplaçant la barre d'outils traditionnelle sur un écran tactile, dont les éléments peuvent ensuite être sélectionnés par une combinaison de touches clavier et d'écran tactile ou par simple touchés sur l'écran tactile. Nous avons d'abord validé expérimentalement la capacité de l'utilisateur à atteindre l'écran tactile placé en-dessous de l'écran, en montrant que la plus grande partie de la surface d'un écran tactile de 5.8 pouces placé sous un clavier peut être balayée avec le pouce tout en maintenant la touche Ctrl enfoncée. Ensuite, nous avons comparé la souris avec deux approches SmartCom (toucher direct ou indirect) pour sélectionner des éléments sur une barre d'outils. Notre étude a révélé que le toucher direct est la meilleure technique en termes de temps de réalisation, et aussi bonne que la souris en termes de précision. Au-delà de ces résultats en termes de performance, combiner un écran tactile avec un clavier permet d'éviter l'occultation d'une partie de l'écran principal par des menus, évite d'avoir à mémoriser l'existence et les moyens d'activation des commandes comme avec les raccourcis clavier, et permet une personnalisation de l'affichage des menus en fonction du niveau d'expertise des utilisateurs avec chaque application : par exemple, un utilisateur peut vouloir afficher les éléments de menu sous forme d'icônes uniquement dans Microsoft Word, mais sous forme d'icônes + labels dans Adobe Illustrator.

Enfin, nous pensons qu'au-delà de la mise en œuvre concrète des techniques SmartCom, notre article apporte une contribution au domaine de l'utilisation combinée d'écrans tactiles et de claviers, domaine qu'il conviendra d'étudier de manière plus approfondie pour d'autres usages et tâches. Nous espérons que nos résultats contribueront à développer les futures recherches sur ce sujet.

REMERCIEMENTS

Ce travail a été partiellement financé par l'ANRT sous la convention Cifre 2018/0246 et par un contrat de collaboration entre OKTAL SYDAC et l'IRIT.

RÉFÉRENCES

- [1] Jonathan Aceituno and Nicolas Roussel. 2014. The hotkey palette: Flexible contextual retrieval of chosen documents and windows. In *IHM 2014 - Actes de la 26ième Conference Francophone sur l'Interaction Homme-Machine*, 55–59. <https://doi.org/10.1145/2670444.2670452>
- [2] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual menu techniques. *ACM Computing Surveys* 49, 4. <https://doi.org/10.1145/3002171>
- [3] Gilles Bailly, Thomas Pietrzak, Jonathan Deber, and Daniel Wigdor. 2013. Métamorphe: Augmenting hotkey usage with actuated keys. In *Conference on Human Factors in Computing Systems - Proceedings*, 563–572. <https://doi.org/10.1145/2470654.2470734>
- [4] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction* 24, 6: 574–594. <https://doi.org/10.1080/10447310802205776>
- [5] Mathieu Berthelémy, Elodie Cayez, Marwan Ajem, Gilles Bailly, Sylvain Malacria, and Eric Lecolinet. 2015. SpotPad, LociPad, ChordPad & InOutPad: Exploration de l'interaction gestuelle sur pavé tactile. In *IHM 2015 - Actes de la 27ème Conference Francophone sur l'Interaction Homme-Machine*, 309–310. <https://doi.org/10.1145/2820619.2820623>
- [6] Lonni Besançon. 2017. An interaction Continuum for 3D Dataset Visualization. Université Paris-Saclay.
- [7] Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Magic desk. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, 2511. <https://doi.org/10.1145/1978942.1979309>
- [8] Florian Block, Hans Gellersen, and Nicolas Villar. 2010. Touch-display keyboards. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, 1145. <https://doi.org/10.1145/1753326.1753498>
- [9] John Brooke. 1996. SUS: a “quick and dirty” usability scale. In *Usability evaluation in industry*. 189.
- [10] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. 1988. An empirical comparison of Pie vs. Linear menus. In *Conference on Human Factors in Computing Systems - Proceedings*, 95–100. <https://doi.org/10.1145/57167.57182>
- [11] Stuart K. Card, Thomas P. Moran, and Allen Newell. 1980. The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM* 23, 7: 396–410. <https://doi.org/10.1145/358886.358895>
- [12] Olivier Chapuis and Nicolas Roussel. 2010. UIMarks: Quick graphical interaction with specific targets. In *UIST 2010 - 23rd ACM Symposium on User Interface Software and Technology*, 173–182. <https://doi.org/10.1145/1866029.1866057>
- [13] Wenzhe Cui, Jingjie Zheng, Blaine Lewis, Daniel Vogel, and Xiaojun Bi. 2019. Hotstrokes: Word-gesture shortcuts on a trackpad. In *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3290605.3300395>
- [14] Richard F. Dillon, Jeff D. Edey, and Jo W. Tombaugh. 1990. Measuring the true cost of command selection: Techniques and results. In *Conference on Human Factors in Computing Systems - Proceedings*, 19–25. <https://doi.org/10.1145/97243.97247>
- [15] Pierre Dragicevic. 2016. *Fair Statistical Communication in HCI*. Springer, Cham, 291–330. https://doi.org/10.1007/978-3-319-26633-6_13
- [16] Emmanuel Dubois, Marcos Serrano, and Mathieu Raynal. 2018. Rolling-menu: Rapid command selection in toolbars using Roll gestures with a multi-dof Mouse. In *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3173574.3173941>
- [17] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. 2016. How we type: Movement strategies and performance in everyday typing. In *Conference on Human Factors in Computing Systems - Proceedings*, 4262–4273. <https://doi.org/10.1145/2858036.2858233>
- [18] Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2017. MarkPad. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, 5630–5642. <https://doi.org/10.1145/3025453.3025486>
- [19] Emmanouil Giannidakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. 2017. IconHK. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, 4715–4726. <https://doi.org/10.1145/3025453.3025595>
- [20] Antonio Gomes, Tristan Trutna, and Roel Vertegaal. 2015. DisplayCover. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15*, 531–535. <https://doi.org/10.1145/2785830.2785843>
- [21] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. 2007. Strategies for accelerating on-line learning of hotkeys. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, 1591. <https://doi.org/10.1145/1240624.1240865>
- [22] François Guimbretière and Terry Winograd. 2000. FlowMenu. In *Proceedings of the 13th annual ACM symposium on User interface software and technology - UIST '00*, 213–216. <https://doi.org/10.1145/354401.354778>
- [23] Jay Henderson, Sylvain Malacria, Mathieu Nancel, and Edward Lank. 2020. Investigating the Necessity of Delay in Marking Menu Invocation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*, 1–13. <https://doi.org/10.1145/3313831.3376296>
- [24] Martin Krzywinski and Naomi Altman. 2013. Points of Significance: Error bars. *Nature Methods* 10, 10: 921–922. <https://doi.org/10.1038/nmeth.2659>
- [25] Jianwei Lai and Dongsong Zhang. 2014. ExtendedThumb: A motion-based virtual thumb for improving one-handed target acquisition on touch-screen mobile devices. In *Conference on Human Factors in Computing Systems - Proceedings*, 1825–1830. <https://doi.org/10.1145/2559206.2581158>
- [26] David M. Lane, H. Albert Napier, S. Camille Peres, and Anikó Sándor. 2005. Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. *International Journal of Human-Computer Interaction* 18, 2: 133–144. https://doi.org/10.1207/s15327590ijhc1802_1
- [27] Sangyoon Lee, Youn-kyung Lim, and Geehyuk Lee. 2020. MirrorPad: Mirror on Touchpad for Direct Pen Interaction in the Laptop Environment. 1–9. <https://doi.org/10.1145/3313831.3376212>
- [28] Blaine Lewis, Greg D'Eon, Andy Cockburn, and Daniel Vogel. 2020. KeyMap: Improving Keyboard Shortcut Vocabulary Using Norman's Mapping. In *CHI '20*.
- [29] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. 2013. Promoting Hotkey use through rehearsal with ExposeHK. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 573. <https://doi.org/10.1145/2470654.2470735>
- [30] Mathieu Pecchioli, Emmanuel Dubois, Pourang Irani, and Marcos Serrano. 2019. Investigating Screen Reachability on an Articulated Dual-Display Smartphone. Springer, Cham, 476–485. https://doi.org/10.1007/978-3-030-29387-1_27
- [31] Gary Perelman, Marcos Serrano, Mathieu Raynal, Celia Picard, Mustapha Derras, and Emmanuel Dubois. 2015. The Roly-Poly Mouse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 327–336. <https://doi.org/10.1145/2702123.2702244>
- [32] F. P. Preparata and S. J. Hong. 1977. Convex Hulls of Finite Sets of Points in Two and Three Dimensions. *Communications of the ACM* 20, 2: 87–93. <https://doi.org/10.1145/359423.359430>
- [33] Jef Raskin. 2000. *The humane interface: new directions for designing interactive systems*. Addison-Wesley.
- [34] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. 2008. TapTap and MagStick. In *Proceedings of the working conference on Advanced visual interfaces - AVI '08*, 146. <https://doi.org/10.1145/1385569.1385594>
- [35] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012. Improving command selection with CommandMaps. In *Conference on Human Factors in Computing Systems - Proceedings*, 257–266. <https://doi.org/10.1145/2207676.2207713>
- [36] Gary R. VandenBos and American Psychological Association. 2015. *APA dictionary of psychology*. American Psychological Association.
- [37] Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. 2010. LensMouse. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, 2431. <https://doi.org/10.1145/1753326.1753695>
- [38] Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual ACM symposium on User interface software and technology - UIST '04*, 33. <https://doi.org/10.1145/1029632.1029639>
- [39] Jingjie Zheng, Blaine Lewis, Jeff Avery, and Daniel Vogel. 2018. FingerArc and FingerChord. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*, 347–363. <https://doi.org/10.1145/3242587.3242589>
- [40] Jingjie Zheng and Daniel Vogel. 2016. Finger-aware shortcuts. In *Conference on Human Factors in Computing Systems - Proceedings*, 4274–4285. <https://doi.org/10.1145/2858036.2858355>
- [41] 2020. <https://support.apple.com/en-us/HT207055>.
- [42] 2020. <https://www.asus.com/us/site/zenbook/ux581.html>.
- [43] 2020. <https://www.razer.com/projectlinda>.
- [44] 2020. [https://en.wikipedia.org/wiki/Hand_\(unit\)](https://en.wikipedia.org/wiki/Hand_(unit)).