

Interactions multi-points indirectes sur grands écrans

THÈSE

présentée et soutenue publiquement le 26 février 2014

pour l'obtention du

Doctorat de l'Université Lille 1
(spécialité informatique)

par

Jérémie Gilliot

Composition du jury

Président : Jean-Stéphane Varré (Université Lille 1)
Rapporteurs : Martin Hachet (Inria Bordeaux – Sud-Ouest)
Stéphane Huot (Université Paris-Sud)
Examineurs : Renaud Blanch (Université Joseph-Fourier – Grenoble 1)
Yves Guiard (Télécom ParisTech)
Directeurs de thèse : Géry Casiez (Université Lille 1)
Nicolas Roussel (Inria Lille – Nord Europe)

Mis en page avec la classe thloria.

Résumé

L'interaction tactile multi-points co-localisée procure un fort sentiment de contrôle direct mais introduit des problèmes d'occultation, de précision et de fatigue. Ces problèmes sont exacerbés sur les écrans de grandes dimensions, qui supportent la manipulation de grandes quantités de données, souvent en collaboration avec d'autres utilisateurs. Pour résoudre ces problèmes, cette thèse s'intéresse à l'introduction d'indirections dans l'interaction multi-points sur grands écrans.

Nos travaux portent sur l'étude des facteurs qui influencent la précision dans les tâches de sélection sans prévisualisation du contact. Les résultats de deux expériences contrôlées ont ainsi permis d'estimer la précision attendue selon les caractéristiques des périphériques d'entrée et de sortie et des conditions d'utilisation. Sur la base de ces connaissances, nos travaux portent également sur le développement de nouvelles techniques d'interaction pour écrans de grandes dimensions. Nous avons d'abord proposé une technique permettant d'interagir indirectement avec une tablette tactile, en introduisant la gestion de plusieurs curseurs contrôlés relativement. Nous avons ensuite conçu une interaction reposant sur la création de plusieurs périphériques virtuels qui supportent une interaction à la fois directe et indirecte tout en permettant de réorganiser les éléments d'une interface. Parallèlement à cela, nous avons transformé un écran mural de 6×2,4m en une surface tactile multi-points afin de pouvoir tester les techniques et scénarios discutés dans ces travaux.

Mots-clés: interaction homme-machine ; interactions indirectes ; interactions multi-points ; curseurs multi-doigts ; périphériques tactiles

Indirect multi-touch interactions on large displays

Abstract

Co-localized multi-touch interaction affords a high feeling of directness but introduces problems of occlusion, precision and fatigue. These problems are even more acute with large displays, supporting the interaction with large data sets, often in collaboration with other users. To solve these problems this thesis focuses on the introduction of indirectness in multi-touch interaction on large displays.

We first study factors impacting accuracy in selection tasks without contact preview. Results of two controlled experiments allowed to estimate the expected accuracy depending on input and output device characteristics and input conditions. Based on this knowledge, we developed new interaction techniques for large displays. We first introduced a technique to interact indirectly using a tactile tablet by introducing the control of several cursors using a relative mapping. Then we proposed an interaction technique based on the creation of multiple soft-devices that supports both direct and indirect interaction and allows to re-arrange the interface of legacy applications. In order to design, test and evaluate these techniques, we turned a back projected wall display (6×2,4m) into an interactive multi-touch surface.

Keywords: human-computer interaction ; indirect interactions ; multi-touch interactions ; multi-touch cursors ; tactile devices

Remerciements

Pour commencer, je tiens à remercier chaudement Laurent Grisoni qui m'a accueilli dans son équipe en janvier 2009 pour réaliser un projet de Master 1, puis m'a recommandé un an plus tard à Géry Casiez pour le stage de Master 2. Il contribue à la bonne humeur au sein de l'équipe et sa curiosité naturelle nous pousse à découvrir de nouvelles choses, parfois très éloignées de l'interaction homme-machine mais tout aussi intéressantes.

Un grand merci à Nicolas Roussel et Géry Casiez pour m'avoir proposé d'effectuer cette thèse. Leurs conseils, remarques avisées et critiques m'ont grandement aidé à mener cette barque à bon port, sans boire la tasse. Leur confiance, leur patience sans faille et leur grande disponibilité malgré les nombreuses contraintes m'ont encouragé durant ces trois années passées à leurs côtés. Ils ont rendu cette thèse possible et ont contribué à développer mon goût pour la culture scientifique et l'envie de découvrir une abondante bibliographie.

J'adresse mes remerciements à Martin Hachet et Stéphane Huot pour m'accorder un peu de leur temps en acceptant d'être rapporteurs de ce mémoire. Merci également à Jean-Stéphane Varré d'avoir accepté de présider le jury et à Renaud Blanch et Yves Guiard d'avoir accepté d'y participer.

J'aurais aimé remercier l'ensemble des enseignants qui ont contribué à mon savoir et qui m'ont porté à ce si haut niveau. En particulier, merci à Anne-Françoise Le Meur qui s'investissait tant pour les étudiants et essayait l'amour des sciences et de la recherche.

Je remercie le projet ANR InSTInCT et le Conseil Régional Nord – Pas de Calais pour m'avoir permis matériellement et financièrement de réaliser cette thèse et le projet européen Shiva qui m'a permis de la terminer dans de bonnes conditions.

Je pense également à tous les collègues de l'équipe Mint : ceux qui sont déjà partis, ceux qui sont revenus, les nouvelles têtes et les permanents. Dans le microcosme d'un non-permanent, les quatre années passées entre le début du stage de Master 2 et la fin de la thèse, c'est très long ! L'ambiance change au fil des arrivées et des départs, mais je voudrais remercier chacune de ces personnes, pour les discussions à propos de rien mais qui sont indispensables, pour les discussions autour du travail et de la recherche qui font avancer, et pour les discussions qui amènent à réfléchir. Merci à tous.

Enfin, je termine par mes amis et ma famille qui m'ont aidé à décompresser durant ses trois années, ils étaient toujours disponibles et ravis de pouvoir m'aider à tenir le coup. Leur présence et leurs efforts pour se rappeler de mon sujet de recherche ont facilité la concrétisation de cette thèse. Merci à ceux qui ont gentiment proposé de corriger des mots qui ont parfois pu leur paraître abscons.

Et merci à Margot de m'avoir soutenu, encouragé, motivé, supporté et aimé durant cette longue aventure, et qui continue à m'inspirer.

Table des matières

Introduction	1
1 Motivations	2
2 Contributions et démarche	3
3 Structure du mémoire	3
Chapitre 1	
État de l'art	
1.1 Introduction	6
1.2 Taxonomie des périphériques	6
1.2.1 Degré de résistance et type de contrôle	6
1.2.2 Périphériques isotoniques absolus et relatifs	7
1.2.3 Périphériques directs et indirects	7
Direct : espaces co-localisés	8
Indirect : espaces séparés	8
1.2.4 Contrôle absolu et relatif, fonction de transfert	9
1.2.5 Tableau récapitulatif	10
1.3 Périphériques tactiles multi-points	10
1.3.1 Historique	11
1.3.2 Technologies optiques	13
1.3.3 Technologies électroniques	15
1.3.4 Technologies acoustiques	16
1.3.5 Interaction mono-point, multi-doigts et multi-points	16
1.4 Fondements de l'interaction directe absolue	18
1.4.1 Capacités sensorielles	18
Le sens du toucher	18
Les sens proprioceptifs	19
1.4.2 Évaluation de la précision du pointage	20
1.4.3 Problèmes de précision et de sélection	21

1.4.4	Facteurs influençant la performance	22
1.4.5	Manipulation indirecte et directe	24
1.5	Évolution de l'interaction tactile multi-points	25
1.5.1	Interaction co-localisée	25
1.5.2	Interaction dé-localisée absolue	29
1.5.3	Interaction quasi-absolue	30
1.5.4	Interaction relative multi-points	32
1.5.5	Espace de solution : Continuum d'indirection	34
1.6	Conclusion	35

<p>Chapitre 2</p> <p>Impacts du facteur de forme et des conditions d'entrée</p>

2.1	Introduction	38
2.2	Expérience 1 : taille du périphérique d'entrée et conditions d'uti- lisation	39
2.2.1	Matériel	40
2.2.2	Tâche et procédure	40
2.2.3	Participants	41
2.2.4	Plan expérimental	42
2.3	Résultats	43
2.3.1	Taux de succès et nombre de tentatives ratées	43
2.3.2	Erreur de visée	44
2.3.3	Taille minimale de cible dans l'espace moteur	47
2.3.4	Distribution spatiale des contacts à la première tentative	47
2.4	Expérience 2 : effets d'échelle et rapport de cadre	50
2.4.1	Matériel, tâche et procédure	50
2.4.2	Participants	50
2.4.3	Plan expérimental	50
2.5	Résultats	52
2.5.1	Taux de succès et nombre de tentatives ratées	52
2.5.2	Erreur de visée	53
2.5.3	Taille minimale de cible dans l'espace moteur	55
2.5.4	Distribution spatiale des contacts à la première tentative	56
2.6	Discussion	57
2.7	Conclusion	60

Chapitre 3

Technique multi-points indirecte relative pour l'interaction distante

3.1	Introduction	63
3.2	Principes de conception	65
3.3	Scénario d'illustration	65
3.4	Proposition	66
3.4.1	Un curseur...	66
	Déplacement	66
	Gestion de la sélection d'objets	67
3.4.2	... multi-doigts	67
	Cursels	67
	Multi-sélection et regroupement de doigts	69
3.4.3	Des curseurs : multi-points	69
	Création et destruction de curseurs	70
	Association doigts et curseurs	70
3.5	Évaluation préliminaire	72
3.5.1	Matériel, tâche et procédure	72
3.5.2	Résultats et limitations	73
3.6	Discussion	74
3.7	Conclusion	75

Chapitre 4

Interaction multi-points directe et indirecte sur murs interactifs

4.1	Introduction	77
4.2	Tactilisation d'un mur de réalité virtuelle	80
4.2.1	Matériel	81
4.2.2	Traitement d'image	82
4.3	WallPad	85
4.3.1	Un pavé tactile	85
4.3.2	Interaction directe délocalisée	86
4.4	Cycle de vie des WallPads	87
4.4.1	Création	87
4.4.2	Gestion dynamique et reconfigurable	88
4.5	Implémentation	89
4.6	Cas d'usage	90
4.6.1	Accès rapide à des fonctions distantes	90

Table des matières

4.6.2	Personnalisation de l'espace de travail	91
4.6.3	Aspects collaboratifs	92
4.6.4	Compatibilité et apports des WallPads avec les applica- tions régulières	93
4.6.5	Au delà du mur	93
4.7	Discussion	94
4.8	Conclusion	96
	Conclusion	97
	Bibliographie	101

Table des figures

1	Schéma d'interaction	2
1.1	Interaction co-localisée et séparée	8
1.2	Exemples d'interaction relative co-localisée	10
1.3	Saqueboute : le premier périphérique tactile	11
1.4	Fonctionnement des technologies optiques	14
1.5	Fonctionnement des technologies électroniques	15
1.6	Fonctionnement des technologies sonores	16
1.7	Organisation somatotopique : aires somesthésique et motrice	20
1.8	Point d'entrée perçu et estimé	21
1.9	Parents physiques des toolglass	25
1.10	<i>Shift</i> , <i>ThumbSpace</i> et <i>MagStick</i>	26
1.11	Conception de widgets de sélection	27
1.12	<i>SurfaceMouse</i> , <i>FluidDTMouse</i> et <i>SDMouse</i>	28
1.13	<i>Rizzos</i>	29
1.14	Interactions absolues délocalisées	29
1.15	<i>ARC-Pad</i>	30
1.16	<i>Overlay</i> et <i>VisualTouchpad</i>	31
1.17	<i>iLoupe</i> et <i>FingerGlass</i>	31
1.18	<i>LensMouse</i>	32
1.19	Design de souris multi-doigts et design de curseurs multi-doigts	33
1.20	<i>HandCursor</i> , <i>SimilarityCursor</i> et <i>AdjustableAreaCursor</i>	34
1.21	Critères d'interaction indirecte	34
2.1	Cache en carton et lunettes œillères	40
2.2	Scénario de l'expérience	41
2.3	Positions des cibles	42
2.4	Conditions d'utilisation	43
2.5	Expérience 1 : nombre de tentatives ratées	45
2.6	Expérience 1 : erreur de visée	46
2.7	Expérience 1 : erreur de visée	46
2.8	Expérience 1 : distribution des premières tentatives	48
2.9	Expérience 2 : nombre de tentatives ratées	53
2.10	Expérience 2 : erreur de visée	54
2.11	Expérience 2 : erreur de visée	55
2.12	Expérience 2 : erreur de visée	55
2.13	Expérience 2 : distribution des premières tentatives	57

Table des figures

3.1	Automate de fonctionnement	66
3.2	Placement des cursels	68
3.3	Regroupement de doigts	69
3.4	Création de curseur	70
3.5	Points d'attraction	71
3.6	Association des cursels	71
3.7	Évaluation informelle	72
4.1	Salle de réalité virtuelle de l'IRCICA	80
4.2	Capture brute des caméras	82
4.3	Traitement d'image	82
4.4	Éclairage de la salle	83
4.5	Vidéoprojecteurs et caméras	83
4.6	Calibration des projections	84
4.7	Un pavé tactile...	85
4.8	... et un peu plus	86
4.9	Geste de création	88
4.10	Gestion des WallPads	89
4.11	Interception des événements tactiles	90
4.12	Exemples d'utilisation des WallPads	91
4.13	Système de cache en carton	95

Liste des tableaux

1.1	Taxonomie des périphériques	11
2.1	Expérience 1 : tailles minimales de cibles	47
2.2	Dimensions de l'espace de travail	51
2.3	Taille des cibles dans l'espace moteur	52
2.4	Expérience 2 : tailles minimales de cible	56

Introduction

« *A map that encourages study encourages use, so that people are less likely to give up and ask for help, and the more they use the map the more they are going to learn about the network, making them more likely to be self-sufficient and resourceful.* »

—Max Roberts (*Lecturer* en Psychologie
à l'Université d'Essex et créateur de cartes)

L'outil informatique, puisqu'il est bien souvent perçu comme tel, que ce soit un outil scientifique pour atteindre d'autres domaines de sciences, ou un outil de médiation et de communication interpersonnel, requiert de l'utilisateur une planification et une organisation des buts. Depuis un objectif à atteindre à son découpage en tâches, de l'intention à l'action, les utilisateurs doivent en premier lieu connaître les possibilités du système, puis effectuer les actions nécessaires à la réalisation des tâches.

L'interaction homme-machine se place dans ce cadre de publication et de découverte de possibilités et d'assistance à la réalisation. Pour réaliser une tâche, l'utilisateur doit être représenté au sein du système informatique au travers d'un curseur. Que ce soit le curseur de la souris, pour sélectionner des objets à l'écran, le point représentant le doigt dans les interfaces tactiles, ou encore la barre clignotante (*carret*) du clavier, qui matérialise le lieu de saisie pour chaque application manipulant du texte. Le curseur représente l'utilisateur par un point unique, un pixel, avec potentiellement des états de sélection. Parfois, le curseur n'a pas besoin d'être affiché comme c'est le cas pour les tablettes et téléphones tactiles. L'utilisateur utilise ensuite le curseur pour agir sur un objet.

La figure 1 représente une partie de ce processus d'interaction. L'utilisateur réalise des gestes qui sont captés par un périphérique. Le périphérique transforme et transmet les gestes au système qui en extrait un ou plusieurs point(s) d'interaction et états. Ce sont ces curseurs, affichés ou non, qui manipulent le contenu, les objets via une technique d'interaction. Beaudouin-Lafon distinguait les instruments physiques et les instruments logiques dans la définition de *l'interaction instrumentale* [7, 8], ces deux types d'outils se retrouvent dans notre illustration qui distingue les *hard-devices* des *soft-devices*. Nous ajoutons le curseur, représentant de l'utilisateur entre ces deux notions.

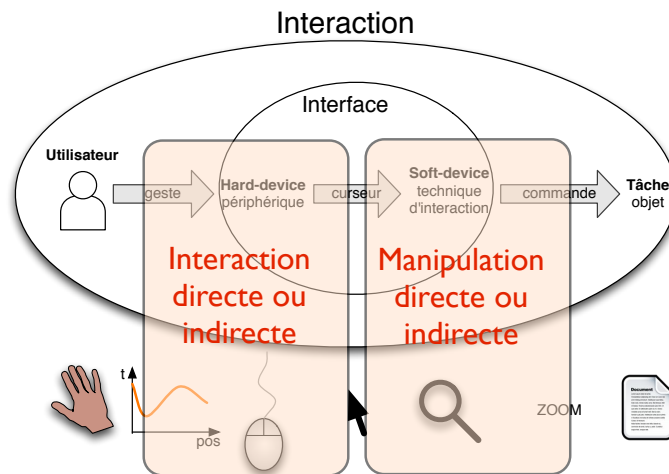


FIGURE 1 – Schéma d'interaction.

1 Motivations

L'interaction tactile existe depuis plusieurs décennies au travers des pavés tactiles d'ordinateurs portables et des tablettes graphiques des dessinateurs qui permettent de contrôler un curseur à l'écran. Dans les laboratoires, les écrans tactiles existent depuis presque aussi longtemps, mais ceux-ci connaissent un regain d'intérêt scientifique et commercial depuis un peu plus de dix ans. En effet avec la démocratisation des périphériques tactiles multi-points, les utilisateurs se sont pris au jeu d'aller au contact physique des objets virtuels.

Pourtant l'interaction tactile directe soulève plusieurs problèmes qui peuvent rendre l'expérience frustrante en raison d'une précision qui laisse à désirer, de sélections erronées, d'informations cachées par les doigts et de la fatigue engendrée par un usage prolongé.

Cette thèse s'inscrit dans le cadre du projet ANR InSTInCT¹ qui vise à concevoir, développer et évaluer de nouvelles interfaces utilisateur basées sur des surfaces tactiles pour favoriser l'interaction 3D, notamment pour des applications 3D à destination du grand public. L'interaction avec les grands écrans nous a paru un contexte d'utilisation intéressant dans le cadre des interfaces tactiles. Ces grandes surfaces d'affichage permettent de présenter une grande quantité de données et sont l'occasion de collaboration ou au moins de coexistence entre plusieurs utilisateurs [9].

Dans ce contexte d'interaction tactile avec de très grands écrans, l'accès indirect aux données par l'intermédiaire d'un périphérique tactile peut être un cas d'usage avantageux. En plus des problèmes classiques de l'interaction tactile, la grande surface d'affichage implique de résoudre l'accès à des objets distants et hors de portée. Les travaux réalisés dans cette étude s'attachent donc au concept de curseur là où il a plutôt tendance à disparaître.

1. <http://anr-instinct.cap-sciences.net/>

2 Contributions et démarche

Les travaux réalisés pendant le stage de Master 2 ont porté sur l'amélioration de la précision accessible en montrant du doigt. Le geste de désignation est un geste de pointage absolu, les premières pistes explorées ont donc consisté à porter ce travail pour l'interaction tactile. En établissant une correspondance absolue entre la surface d'entrée (une tablette tactile) et un écran, l'utilisateur peut accéder au contenu avec tous ses doigts de manière simple et rapide. Les problèmes de précision inhérents au facteur d'échelle entre les deux surfaces ont orienté les travaux vers la proposition d'une mise en correspondance moins naïve entre les deux espaces.

Cette approche a abouti à une technique reposant sur le contrôle de plusieurs curseurs via une seule surface tactile. La position des curseurs à l'écran est prise en compte afin de déterminer avec quel curseur l'utilisateur veut agir, cela fournit un moyen compréhensible pour choisir un des curseurs. En plus de la gestion de plusieurs curseurs, chaque curseur est multi-doigts : l'utilisateur peut engager plusieurs doigts et manipuler plusieurs objets. Cette contribution a fait l'objet d'un article publié à Ergo'IHM 2012 [30].

Dans ce travail, la sélection absolue sans prévisualisation tient une part importante dans la sélection des curseurs, il a donc paru intéressant de quantifier la précision de pointage absolu et les facteurs influant sur cette précision. L'objectif étant de mesurer la capacité à passer de l'espace d'entrée à l'espace d'affichage sans survol et sans technique d'amélioration de la précision afin d'établir une base de performance brute pour évaluer l'intérêt de ce type d'interaction. Les expériences et résultats sont regroupés dans un article publié à CHI 2014 [32].

Le contexte de l'interaction tactile directement au contact de l'écran est apparu comme favorable pour tirer parti de l'utilisation de plusieurs curseurs et ainsi autoriser l'accès à l'ensemble du contenu, aussi loin soit-il. Afin de faciliter la gestion de plusieurs curseurs et d'enrichir les cas d'usages observés avec la technique publiée à Ergo'IHM 12, le contrôle de chaque curseur est accessible par un pavé tactile virtuel créé par l'utilisateur. Ces travaux ne sont pas encore publiés [31].

3 Structure du mémoire

Dans le premier chapitre, nous allons introduire une classification des différentes familles de périphériques d'entrée, puis détailler les raisons du succès de l'interaction tactile et en exposer les limitations. À partir de là, nous passerons en revue les travaux de l'état de l'art qui introduisent une indirection entre le toucher et le lieu d'interaction.

Le deuxième chapitre s'intéressera particulièrement à l'étude du pointage indirect sans prévisualisation du contact. Les impacts de divers facteurs intrinsèques ou extrinsèques aux périphériques seront quantifiés afin d'évaluer les

possibilités d'interaction reposant sur la sélection absolue indirecte. En particulier, nous montrerons que la précision dépend de la taille du périphérique d'entrée et que la vision des deux espaces permet d'améliorer les performances.

Afin d'améliorer la précision tout en conservant la possibilité d'accéder à la totalité de l'écran, le troisième chapitre détaillera une technique multi-points autorisant la manipulation multi-doigts sur un grand écran en utilisant une tablette tactile. Après une évaluation préliminaire, nous discuterons de la difficulté à utiliser plusieurs curseurs simultanément, mais aussi du succès de la sélection des curseurs par pointage indirect absolu et de l'intérêt apporté par la possibilité de gérer plusieurs curseurs séquentiellement.

Enfin, le dernier chapitre se focalisera sur l'interaction au contact de murs tactiles interactifs et sur l'aspect multi-utilisateurs. Nous présenterons la tactilisation d'un écran mural d'une salle de réalité virtuelle. Ensuite nous proposerons l'introduction de périphériques logiques créés dynamiquement et contrôlant un curseur multi-doigt afin de supporter une interaction multi-points locale, distante et délocalisée. Avec la gestion de plusieurs périphériques logiques, l'utilisateur pourra bénéficier d'autant de curseurs que nécessaire pour accéder à la totalité de l'écran.

Chapitre 1

État de l'art

« Almost nobody, looking at a computer system for the first time, has the slightest idea what it will do or how it should work. What people call an "intuitive interface" is generally one which becomes obvious as soon as it is demonstrated. But before the demo there was no intuition of what it would be like. Therefore the real first sense of "intuitive" is retroactively obvious. »

—Ted Nelson²

Sommaire

1.1	Introduction	6
1.2	Taxonomie des périphériques	6
1.2.1	Degré de résistance et type de contrôle	6
1.2.2	Périphériques isotoniques absolus et relatifs	7
1.2.3	Périphériques directs et indirects	7
1.2.4	Contrôle absolu et relatif, fonction de transfert	9
1.2.5	Tableau récapitulatif	10
1.3	Périphériques tactiles multi-points	10
1.3.1	Historique	11
1.3.2	Technologies optiques	13
1.3.3	Technologies électroniques	15
1.3.4	Technologies acoustiques	16
1.3.5	Interaction mono-point, multi-doigts et multi-points	16
1.4	Fondements de l'interaction directe absolue	18
1.4.1	Capacités sensorielles	18
1.4.2	Évaluation de la précision du pointage	20
1.4.3	Problèmes de précision et de sélection	21
1.4.4	Facteurs influençant la performance	22
1.4.5	Manipulation indirecte et directe	24
1.5	Évolution de l'interaction tactile multi-points	25
1.5.1	Interaction co-localisée	25
1.5.2	Interaction dé-localisée absolue	29
1.5.3	Interaction quasi-absolue	30

2. <http://hyperland.com/TedCompOneLiners>

1.5.4	Interaction relative multi-points	32
1.5.5	Espace de solution : Continuum d'indirection	34
1.6	Conclusion	35

1.1 Introduction

L'interaction tactile regroupe une large famille de périphériques et de techniques, on peut citer le pavé tactile qui contrôle un curseur, la tablette graphique qui permet de dessiner en déplaçant un curseur de manière absolue et les écrans tactiles. Ces exemples sont aujourd'hui largement utilisés par le grand public, qui apprécie la précision et l'ergonomie d'utilisation du curseur et la facilité d'utilisation des interfaces tactiles. Afin de préciser les concepts relatifs de ces différents cas et comprendre l'intérêt de chacun de ces exemples, nous allons organiser ce chapitre de la manière suivante. En premier lieu, nous introduirons la terminologie liée aux périphériques d'interaction et aux concepts d'interaction, puis nous aborderons le fonctionnement des périphériques d'entrée tactile multi-points. Ensuite nous étudierons les avantages et inconvénients de l'interaction directe absolue. Finalement nous présenterons les travaux existants essayant de résoudre les problèmes posés par l'interaction directe en réintroduisant des concepts d'indirection entre les doigts et les objets.

1.2 Taxonomie des périphériques

Les périphériques d'interaction peuvent être classés en fonction de leurs caractéristiques, nous allons présenter une taxonomie des périphériques permettant de différencier les différentes familles et de restreindre le focus de cette étude.

1.2.1 Degré de résistance et type de contrôle

Il convient en premier lieu de distinguer les périphériques selon le type de déplacement de l'effecteur. La première catégorie mesure des positions ou des déplacements avec une résistance constante, voire nulle, on parle de périphériques *isotoniques*. Dans cette catégorie, on trouve les souris, les trackballs, mais aussi on trouve les surfaces tactiles et les systèmes de capture 3D. À l'opposé, se trouve la famille composée des périphériques *élastiques* et *isométriques* qui mesurent une force ou une variation de force en opposant au déplacement une résistance respectivement croissante ou infinie. Les joysticks font correspondre la force exercée au déplacement du manche, la force de rappel au centre, s'opposant au déplacement, augmente à mesure que le ressort se comprime. Les trackpoints et SpaceMouse (souris 3D), quand à eux ne se déplacent quasiment pas, l'utilisateur peut toutefois exercer une force variable pour accélérer plus ou moins le curseur, par exemple.

La mesure du périphérique est utilisée pour contrôler le curseur, différents types de contrôle peuvent être appliqués, on peut agir sur la position du curseur ou sur sa vitesse de déplacement. Une fonction de transfert d'ordre 0 est appelée *contrôle en position* car la position du périphérique contrôle la position du curseur. Un écran tactile ou une trackball sont des exemples de périphériques implémentant un contrôle en position. Une fonction de transfert d'ordre 1 définit un *contrôle en vitesse* : la vitesse de déplacement du curseur dépend de la force appliquée sur le périphérique ou de sa position, par exemple le trackpoint présent sur certains ordinateurs portables permet de contrôler la vitesse du curseur.

Zhai et Milgram ont étudié les liens entre les types de périphériques et les fonctions de transfert. Ils ont montré que les interfaces isométriques et élastiques ont des performances sensiblement identiques même si l'apprentissage des périphériques élastiques est plus rapide grâce aux retours d'informations fournis par le ressort et le déplacement [106]. Zhai a également établi que les associations contrôle en position pour les périphériques isotoniques et contrôle en vitesse pour les périphériques isométriques et élastiques présentaient les meilleures performances [107].

1.2.2 Périphériques isotoniques absolus et relatifs

Les surfaces tactiles sont donc des périphériques isotoniques et sont souvent associés à un contrôle en position. De plus ce sont des périphériques *absolus*, ils retournent comme information la position des contacts avec la surface. Les coordonnées des contacts sont dans un seul et même repère tout au long de l'interaction. Quand un doigt est pressé sur la surface, le périphérique mesure sa position absolue. Lorsque ce doigt est retiré puis reposé à un autre endroit, la nouvelle position est immédiatement placée dans le même repère que précédemment et l'on peut mesurer le déplacement qui a eu lieu au-dessus du périphérique.

Les périphériques *relatifs* ont ceci de différent qu'ils mesurent un déplacement par rapport à une référence temporaire. La souris par exemple ne connaît pas l'endroit où elle est posée, mais ne transmet que les déplacements qui ont été effectués. Lorsque la souris est soulevée, elle ne capte plus les déplacements, elle est dans un état débrayé. Les périphériques relatifs nécessitent une dimension supplémentaire pour permettre à l'utilisateur de débrayer et de définir un nouveau point de référence. La souris quitte le plan où elle repose, les périphériques de suivi 3D dispose d'un bouton. Les surfaces tactiles présentent également un état débrayé dans lequel les doigts ne sont plus suivis, appelé *Out Of Range (OOR)* [18], mais lorsqu'ils sont reposés, leur position absolue est à nouveau calculée.

1.2.3 Périphériques directs et indirects

L'utilisateur manipule le périphérique d'entrée dans l'espace de contrôle ou espace d'entrée ou encore espace moteur ; le contenu et les objets numériques

sont affichés quant à eux dans l'espace d'affichage ou espace de travail visuel. Pour la plupart des périphériques ces deux espaces sont clairement identifiables et distincts, il s'agit alors de périphériques indirects. Dans le cas des surfaces tactiles ou de l'interaction avec un stylet, ces deux espaces peuvent être confondus ou disjoints (figure 1.1).

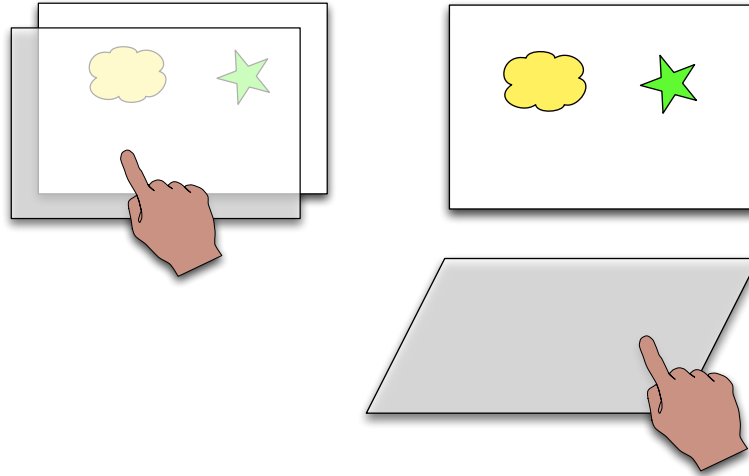


FIGURE 1.1 – À gauche, interaction co-localisée : les deux espaces d'entrée et d'affichage sont situés au même endroit. À droite, interaction séparée : le périphérique d'entrée est distinct de la surface d'affichage. De plus, les deux espaces peuvent avoir des orientations différentes, ici le plan d'entrée est horizontal alors que l'affichage est dans le plan vertical.

Direct : espaces co-localisés

Dans le cas où les deux espaces sont co-localisés, l'utilisateur a l'impression d'aller directement au contact du contenu. Le périphérique d'entrée est positionné directement sur l'écran et tend à disparaître : on parle d'écran tactile, l'utilisateur agit alors avec ses doigts, un stylet ou des objets tangibles selon les possibilités de la technologie employée. L'affichage d'un curseur pour représenter le contact n'est alors pas nécessaire, le point d'entrée extrait par le système est co-localisé avec le contact. Ils sont au même endroit, superposés.

Indirect : espaces séparés

Lorsque les espaces sont séparés, l'utilisateur perçoit clairement la distinction entre le périphérique d'entrée et l'espace d'affichage. Il effectue des gestes sur/avec le périphérique physique pour agir dans un environnement virtuel, il contrôle un curseur et l'amène sur un objet pour le sélectionner ou le modifier.

Quand les deux espaces sont disjoints, ils peuvent présenter des caractéristiques propres comme des dimensions différentes ou des rapports de cadre (*aspect-ratio*) différents. De plus les positions et orientations relatives peuvent être

changées et avoir un impact sur l'interaction et les capacités des utilisateurs à rapidement assimiler le contrôle. Nous reviendrons un peu plus tard sur ces considérations de performances et de prises en compte des configurations.

1.2.4 Contrôle absolu et relatif, fonction de transfert

L'introduction d'un curseur dans le cadre des interfaces indirectes amène à compléter la définition de la fonction de transfert établissant la relation entre le périphérique et le curseur. Les écrans tactiles, tout comme les souris et les trackballs, sont des périphériques isotoniques assurant un contrôle en position du curseur, c'est-à-dire qu'ils font correspondre un déplacement du curseur à un déplacement du périphérique ou sur le périphérique. La fonction de transfert calcule le déplacement du curseur en fonction du déplacement du périphérique d'entrée, c'est une fonction au sens algorithmique, avec des dépendances externes et des effets de bord éventuels, tels que le point de départ, l'emplacement des cibles ou des retenues de calculs.

Si on note dP le déplacement du périphérique (en mètres) pendant dt secondes et dC le déplacement du curseur correspondant, alors l'équation 1.1 donne la forme générique d'une fonction de transfert dans notre cas, et il faut comprendre qu'elle peut accéder et modifier d'autres informations (dépendances, effets de bord). L'équation 1.2 correspond à une fonction de gain. L'équation 1.3 correspond à un gain constant. Pour les systèmes de fenêtrage habituels, il a été montré que F_G est une fonction non linéaire qui tient compte de la vitesse $\frac{dP}{dt}$ et éventuellement des caractéristiques des périphériques d'entrée/sortie [20].

$$dC = F(dP, dt, \dots) \quad (1.1)$$

$$dC = F_G(dP, dt, \dots) \times dP \quad (1.2)$$

$$dC = k \times dP \quad (1.3)$$

Avec les interfaces directes, les points d'entrée correspondent aux contacts, ce gain est donc égal à 1. La position du curseur (point d'interaction retenu par le système) est la même que celle du contact. Dans le cas où les deux espaces sont séparés, le gain peut être constant : un facteur multiplicatif. Par exemple, sur une tablette graphique, le stylet déplace un curseur mais les déplacements sont multipliés pour passer d'un espace à l'autre, la tablette étant plus petite que l'écran, le facteur multiplicatif assure le passage à l'échelle de l'écran et est déterminé par le rapport des tailles.

Au lieu d'une constante, ce gain peut être une fonction dépendante du déplacement ou de la vitesse en entrée, on parle alors de fonctions de transfert non-linéaires. L'objectif est par exemple d'avoir pour des déplacements rapides, un gain élevé (quelques unités) qui va démultiplier le mouvement du curseur, augmentant encore plus la vitesse et la portée. A l'opposé, pour des déplacements lents, un faible gain (proche de 1 ou légèrement inférieur) apportera une précision accrue [20].

Les périphériques relatifs, ayant une référence temporaire, ne sont capables, par définition, que d'implémenter un contrôle relatif : le curseur n'est pas déplacé lorsque le périphérique est débrayé. Par exemple on peut replacer une souris sur le bureau sans déplacer le curseur. Une surface tactile est un périphérique absolu, mais peut fonctionner en mode *relatif*. Seuls les déplacements relatifs sont alors utilisés pour contrôler le(s) curseur(s), on utilise comme référence temporaire le lieu du début du contact. L'emplacement du doigt sur la surface n'est pas utilisé directement, seul compte ses mouvements. Les pavés tactiles d'ordinateurs sont des périphériques tactiles indirects absolus utilisés pour assurer un contrôle relatif du curseur. Les interactions relatives co-localisées sont par contre beaucoup plus rares. Dans la littérature on peut citer *HybridPointing* [27] ou *MagStick* [73] (figure 1.2).

Les surfaces tactiles peuvent également être exploitées en tant que périphérique absolu et implémenter une interaction *absolue* : la position des curseurs va alors dépendre directement de la position des contacts, potentiellement après application d'un gain multiplicatif. Les écrans tactiles fournissent une interaction absolue co-localisée, les tablettes graphiques sont des exemples d'interaction absolue séparée.

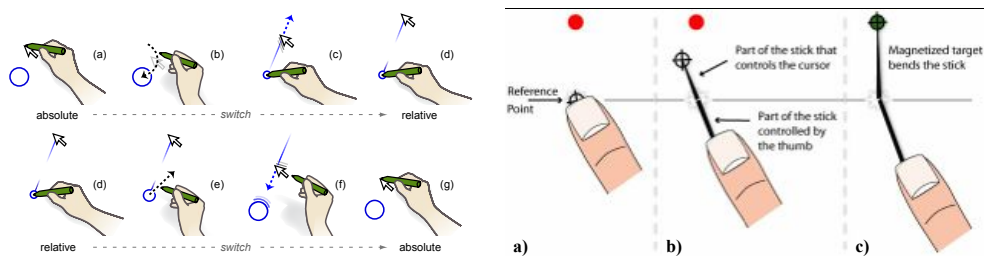


FIGURE 1.2 – Exemples d'interaction relative co-localisée (issus respectivement de [27] et [73]).

1.2.5 Tableau récapitulatif

Nous avons reporté les différentes caractéristiques évoquées plus haut dans le tableau 1.1 afin de synthétiser les familles et de montrer les relations et les configurations possibles.

Abordons maintenant l'aspect technique des périphériques d'entrée tactile multi-points au travers d'un bref historique et du passage en revue des diverses technologies employées.

1.3 Périphériques tactiles multi-points

Les périphériques d'entrée tactile multi-points ont pour particularité d'offrir un suivi continu de plusieurs points simultanément. Ces informations sur les doigts sont le plus souvent la position et la forme des contacts permettant ainsi

Degré de résistance	Type de contrôle	Mesure physique	Position	Contrôle du curseur	Exemple
Isotonique	Contrôle en position	Relatif	Indirect	Relatif	Souris
		Absolu (surfaces tactiles)	Direct	Absolu	Écran tactile
			Indirect	Relatif	<i>MagStick</i>
				Absolu	Tablette graphique
Relatif	Pavé tactile				
Élastique	Contrôle en vitesse	Absolu	Indirect	Relatif	Joystick
Isométrique					TrackPoint

TABLE 1.1 – Taxonomie des périphériques.

de déduire l'orientation des doigts [93] ou de rejeter la paume de la main, mais elles peuvent aussi être la pression normale du contact, l'angle d'incidence ou encore les forces tangentielles appliquées par chaque doigt, la différenciation des parties du doigt [40], l'identification des doigts [60], la posture des mains [103] et l'identité des personnes [23].

1.3.1 Historique

Les premiers périphériques à pouvoir détecter une position de manière absolue ont commencé à être développés au milieu du 20^e siècle. Ainsi le musicien Hugh Le Caine créait en 1953 un synthétiseur couplé à un contrôleur sensible à la pression des doigts³. Sa saqueboute électronique se jouait de la main droite sur un clavier et de la main gauche sur un périphérique atypique représenté sur la figure 1.3. Les doigts sont positionnés sur des surfaces sensibles à la pression, l'index déplace un petit coussinet pour sélectionner les instruments.

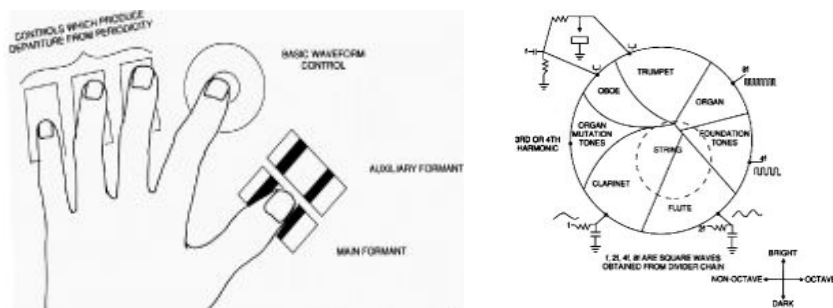


FIGURE 1.3 – Saqueboute : le premier périphérique tactile (issu de ³).

3. <http://www.hughlecaine.com/en/sackbut.html>

La volonté de pouvoir désigner une position ou un objet directement sur un écran informatique est tout d'abord passée par l'invention des crayons optiques : le *LightGun* en 1949 [25] et le *lightPen* en 1957 [34]. Le périphérique détectait le faisceau d'électrons du tube cathodique pour en déduire sa position. La possibilité d'aller directement au contact du contenu va amener Sutherland à concevoir *Sketchpad* [84], un précurseur des logiciels de CAO (conception assistée par ordinateur) employant un crayon optique, une interface graphique, de la manipulation directe et une grammaire graphique et gestuelle. Malgré les capacités de calcul limitées du système, il effectue déjà un rendu immédiat (mais pas instantané) des modifications comme le formalisera plus tard la manipulation directe. À partir de 1965, les premiers écrans tactiles sont créés en laboratoire [47] et en 1972 un premier terminal à écran tactile est distribué, le *Plato IV*⁴. Ces écrans sont toutefois mono-point et ne détectent pas le changement de pression

Au début des années 80, les premiers périphériques indirects multi-points sont créés : un dispositif électronique voit le jour en 1981 [102], un dispositif optique composé d'une vitre translucide et d'une caméra est mis au point en 1982 [63], à l'université de Toronto. Le *Bell Labs* travaille sur l'aspect dynamique apporté par ces périphériques tactiles [67] et crée en 1984 le premier écran multi-points en employant la technologie capacitive superposée à un écran cathodique (Bob Boie⁵), basée sur les travaux décrits dès 1965.

Pendant 20 ans les technologies se perfectionnent, des études sont conduites sur les techniques d'interaction, les performances et les facteurs humains. Les idées de bureaux digitaux [96], de téléphones tactiles et de gestes multi-doigts commencent à émerger. En 2001, la table interactive *Diamond Touch* est capable de distinguer les différents utilisateurs en y associant les doigts en contact avec la surface [23]. Vers 2004, 2005 plusieurs écrans tactiles sont créés en couplant des caméras et des vidéoprojecteurs [99, 100, 39] et commencent à démocratiser et à baisser les coûts de construction des surfaces multi-points de grandes tailles. En 2007, l'iPhone d'Apple⁶ et la table Microsoft Surface⁷ sont commercialisés. Depuis 2010 les tablettes tactiles se vendent en grandes quantités et aujourd'hui les trois systèmes d'exploitation standards (Windows, OS X et Linux) existent en version tablette tactile.

Il existe aujourd'hui plusieurs moyens de réaliser des surfaces tactiles. Les technologies les plus courantes peuvent être séparées en trois grandes catégories : les technologies optiques qui réalisent un traitement d'image pour localiser les contacts, les technologies électroniques qui détectent des changements de caractéristiques électriques ayant lieu sur la surface, et les technologies acoustiques qui reposent sur la spatialisation du son et les bruits de surface. Nous allons maintenant détailler le fonctionnement et les limitations de ces solutions.

4. http://en.wikipedia.org/wiki/Plato_computer

5. <http://www.billbuxton.com/multitouchOverview.html>

6. <http://www.apple.com/fr/iphone/>

7. renommée depuis PixelSense <http://www.microsoft.com/en-us/pixelsense/default.aspx>

1.3.2 Technologies optiques

Les technologies optiques se basent sur une ou plusieurs caméras pour filmer la surface d'interaction. Dans le cas d'écrans tactiles, un vidéoprojecteur affiche le contenu virtuel sur cette même surface d'interaction. Afin que le système de captation et de vidéoprojection ne se gênent pas mutuellement, l'acquisition des contacts a lieu en lumière infrarouge. Un vidéoprojecteur d'ordinateur et une caméra infrarouge sont placés sous une plaque en acrylique dépolie (figure 1.4). Cette plaque est utilisée comme surface de projection et comme surface d'interaction. La diffusion de la lumière au travers de la plaque dépolie permet de segmenter l'espace : les objets au contact de la plaque sont clairement visibles mais deviennent flous lorsqu'ils s'éloignent. Les doigts sont ainsi captés par la caméra infrarouge seulement lorsqu'ils sont en contact avec la surface. Les performances en termes de rapidité et nombre de contacts maximum dépendent donc des algorithmes de traitement d'images et des ressources disponibles. Ces technologies permettent de détecter les doigts (avec ou sans gants), les objets tangibles et les stylets et parfois fournissent des informations légèrement au-dessus de la surface. De plus, elles sont relativement peu coûteuses et fonctionnent bien sur de grandes surfaces. Par contre elles sont très sensibles à la lumière extérieure et prennent beaucoup de place derrière la surface.

Les différences entre les technologies proviennent de la manière d'apporter l'éclairage infrarouge. Pour les surfaces FTIR (*frustrated total internal reflection*), des diodes-électroluminescentes (DEL) placées sur la tranche de la plaque vont émettre la lumière à l'intérieur de la plaque. Cette lumière va se diffuser et rester dans la plaque par réflexion totale (comme dans une fibre optique). Lorsqu'un doigt est posé sur la surface, il modifie l'indice de réfraction et la lumière sort alors de la plaque et illumine le doigt en contact (figure 1.4 a). Le principe est similaire pour la technologie DSI (*diffused surface illumination*) pour lequel la lumière se réfléchit dans la plaque en acrylique. Une surface diffuseuse est superposée afin de diffuser la lumière infrarouge vers le haut et éclairer les mains à proximité et en contact (figure 1.4 b). Parfois, la plaque en acrylique elle-même contient de petites aspérités ou des billes ayant pour objectif de diffuser une partie de la lumière infrarouge. Des lasers infrarouges peuvent être également employés afin de créer un plan (*laser light plane*) juste au-dessus de la surface. Lorsque la main coupe ce plan, elle réfléchit la lumière qui devient visible par les caméras (figure 1.4 c). La lumière infrarouge peut également provenir de projecteurs placés derrière la surface (*rear diffused illumination*) ou devant la surface (*front diffused illumination*). Dans le premier cas, les infrarouges passent au travers de la plaque dépolie et lorsque la main se trouve au contact de l'écran, elle réfléchit une grande quantité de lumière qui peut alors être détectée. Dans le second cas, la plaque est illuminée par le rayonnement infrarouge ambiant, les doigts vont créer des ombres en venant au contact de la plaque.

Enfin il existe deux autres approches optiques pour tactiliser une surface, la première utilise une caméra 3D (type Kinect) ou de capture 3D (type Vicon) pour modéliser les mains ou objets à proximité de l'écran. Les contacts sont

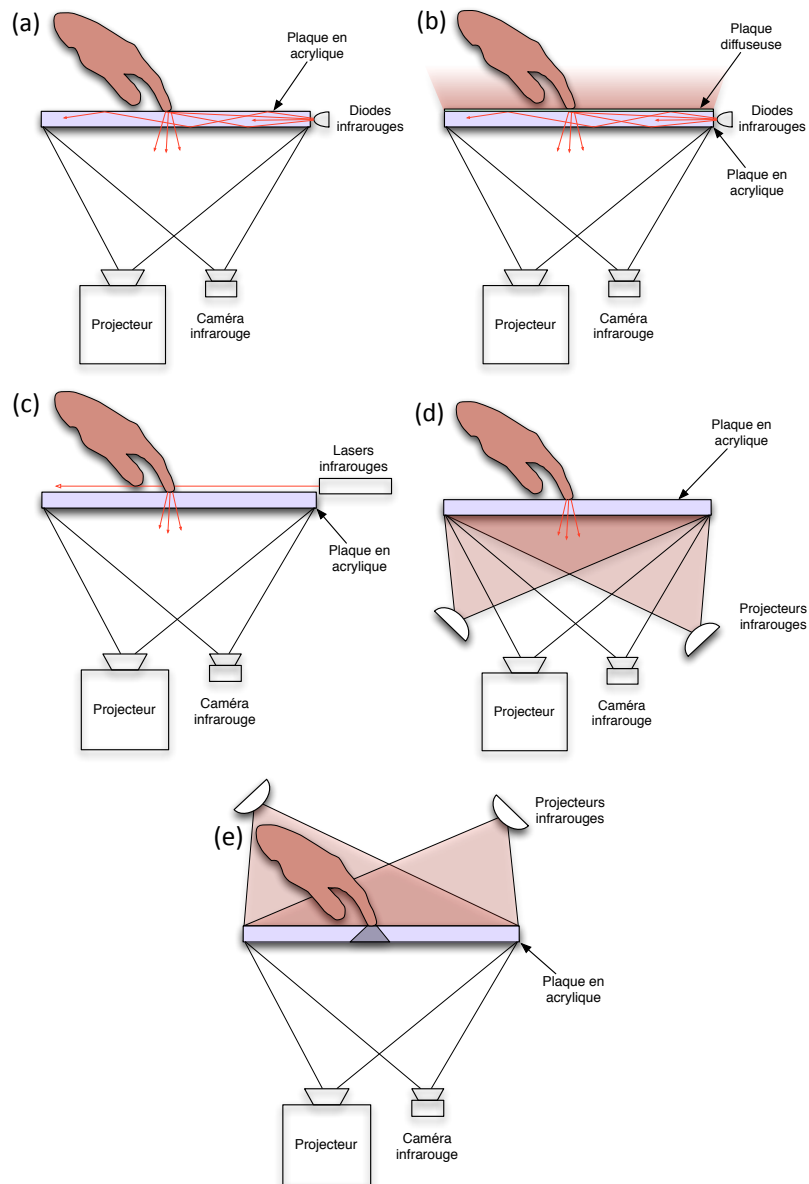


FIGURE 1.4 – Fonctionnement des différentes technologies optiques : FTIR (a), DSI (b), LLP (c), RDI (d) et FDI (e).

détectés par un algorithme de détection de contact entre les entités. Cette approche permet de rendre tactile n'importe quelle surface mais n'est cependant pas aussi précise et souffre de problèmes d'occultation. Enfin la table tactile de Microsoft propose une technologie nommée *PixelSense*, qui comprend des capteurs optiques intégrés dans la dalle LCD. Ces capteurs sont capables de renvoyer une vue des objets en contact avec la surface sans utiliser de caméra : la dalle se comporte comme une caméra et est en mesure de scanner le texte d'une feuille⁸.

1.3.3 Technologies électroniques

Les technologies électroniques ont l'avantage de se présenter sous la forme de couches additionnelles superposées à l'écran. Elles n'occupent donc que très peu de place mais elles nécessitent de disposer des circuits électroniques transparents sur les écrans, ce qui rend leur intégration coûteuse sur des écrans de grandes dimensions. Longtemps limitées à la détection d'un ou deux contacts, les améliorations de traitements de signaux ont permis de détecter plusieurs dizaines de contacts.

Les écrans tactiles résistifs sont composés de deux couches conductrices et transparentes (figure 1.5 gauche), elles sont traversées par un courant électrique, l'une est alimentée par les bords verticaux et l'autre par les bords horizontaux. L'espace entre ces deux couches est très mince, si bien que la pression exercée par un doigt ou un stylet suffit à les mettre localement en contact. Le courant passant entre les deux couches est alors mesuré afin de déduire la position du contact. Ce type d'écran fonctionne aussi bien avec les doigts gantés ou non, les stylets et autres objets conducteurs ou non, il faut néanmoins exercer une légère pression qui peut être ressentie comme un manque de réactivité par l'utilisateur. De plus, la souplesse de la couche supérieure peut la rendre sensible aux rayures et nécessiter un recalibrage⁹.

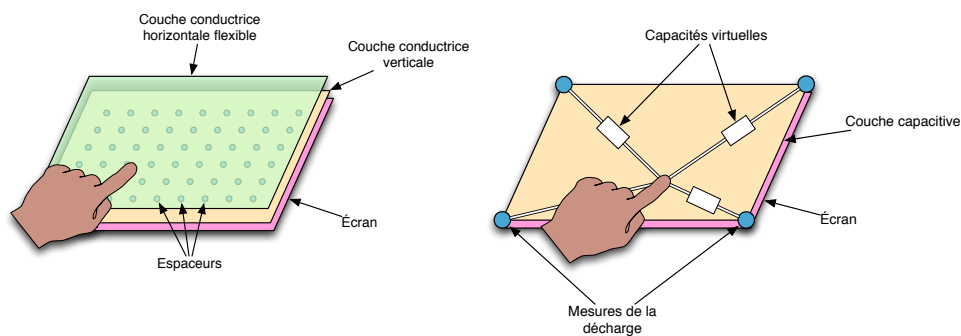


FIGURE 1.5 – Fonctionnement des technologies électroniques. À gauche, écran résistif. À droite, écran capacitif.

Les écrans capacitifs sont constitués d'une couche diélectrique en indium (figure 1.5 droite). Cette plaque possède une charge électrostatique et lorsque

8. <http://www.microsoft.com/en-us/pixelsense/pixelsense.aspx>

9. http://globaldisplayengineering.com/analog-touch_screens.php

la main entre en contact avec la surface, des charges sont transférées de la plaque vers la main. La mesure des décharges ayant lieu sur la plaque permet de calculer la position des contacts. Cette technologie est donc capable de reconnaître les objets conducteurs reliés à la terre, les doigts gantés ne sont donc pas reconnus (sauf dispositif particulier), les stylets doivent être spécifiquement compatibles avec les écrans capacitifs. La détection des contacts a lieu dès effleurement de la surface réduisant ainsi la latence du système et augmentant la réactivité ressentie. Une amélioration des écrans capacitifs se nomme capacité projetée : les capteurs sont répartis sur l'écran pour former une grille. La mesure de la décharge est donc effectuée au travers de ces capacités, permettant de détecter plusieurs contacts simultanément et plus précisément.

1.3.4 Technologies acoustiques

Les technologies acoustiques permettent de rendre tactile toute surface, des microphones ou des piézoélectriques sont disposés autour de la surface à tactiliser (figure 1.6) afin de capter et localiser les sons produits par le contact et le glissement avec la surface. Une variante nommée SAW (*surface acoustic wave*) repose sur l'émission d'une onde ultrasonique, les contacts absorbent et modifient le signal. La comparaison de l'acquisition avec le signal de référence et le décalage entre les microphones permettent de localiser la position des contacts. Cependant ces technologies demandent une calibration précise de chaque point pour construire une carte de référence. Elles sont très limitées en nombre de contacts détectés et l'altération de la surface tactile, comme une rayure ou la poussière, affecte grandement la précision [14].

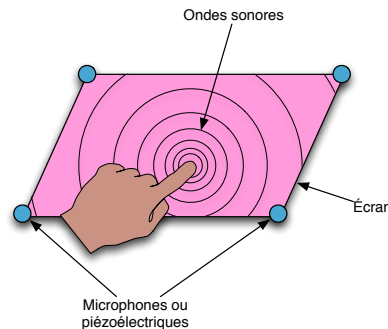


FIGURE 1.6 – Fonctionnement des technologies sonores.

1.3.5 Interaction mono-point, multi-doigts et multi-points

Il convient de distinguer les possibilités offertes par le périphérique des fonctionnalités de la technique d'interaction. Une surface tactile peut être mono-point ou multi-points, il s'agit ici d'une caractéristique matérielle du périphérique physique : soit plusieurs points d'entrée peuvent être détectés et suivis, soit il ne peut y avoir qu'un seul point d'entrée. Ainsi la plupart des pavés

tactiles modernes peuvent détecter plusieurs doigts (au moins deux) pour différencier le contrôle du curseur et le défilement du document (émulation de la molette) ; pour autant l'interaction n'en demeure pas moins *mono-point*. Les systèmes d'exploitation modernes introduisent quelques gestes tirant parti des capacités multi-points, par exemple pour changer de bureau ou d'application. Généralement, ces gestes déclenchent des commandes discrètes, la plupart du temps sans paramètres. Toutefois le contrôle du niveau de zoom avec deux doigts est un exemple de manipulation multi-doigts indirecte d'un paramètre continu.

Ainsi malgré les capacités multi-points des périphériques, l'interaction peut demeurer essentiellement *mono-point*. L'introduction de plusieurs doigts permet d'augmenter la bande passante entre l'utilisateur et la machine, puisqu'il devient possible de sélectionner ou manipuler plusieurs objets ou propriétés simultanément. Dans ce mémoire, nous distinguerons les interactions multi-doigts des interactions multi-points.

- Les interactions *multi-doigts* proposent d'utiliser plusieurs doigts dans un même lieu d'interaction : par exemple pour tourner un objet avec deux doigts, ou dessiner plusieurs lignes simultanément. L'utilisation de plusieurs doigts, d'une main ou des deux mains, de manière conjointe et synchronisée apporte déjà une plus grande expressivité à l'interaction.
- Les interactions *multi-points* regroupent les cas où l'utilisateur interagit dans divers endroits de l'écran, simultanément ou non, pour effectuer différentes tâches. Par exemple dans la plupart des logiciels, on sélectionne consécutivement des outils puis on édite l'objet central. L'usage de la souris implique de faire sans cesse des allers-retours, un utilisateur avancé utilisera à la place des raccourcis claviers pour gagner du temps. Une interaction tactile multi-points permettra à l'utilisateur d'utiliser ses deux mains ou de déplacer sa main physiquement vers les deux lieux d'interaction. L'utilisateur va effectuer un déplacement physique vers une cible *située dans l'espace* : la position de la cible est consistante dans le monde physique, il peut ainsi utiliser sa mémoire spatiale et musculaire au lieu d'amener un curseur vers le lieu d'interaction.

Moscovich et Hughes ainsi que Kin *et al.* ont tous établi la même distinction entre interaction multi-doigts uni-manuelle et interaction multi-points bi-manuelle en étudiant des tâches de positionnement 2D [66] et de sélection multiple [51]. Ainsi, ils recommandent l'usage de plusieurs doigts de la même main pour manipuler aisément plusieurs degrés de liberté simultanément, tel que l'orientation ou l'échelle d'un objet, alors que l'engagement des deux mains est à réserver pour effectuer des tâches qui requièrent le contrôle indépendant de deux points ou objets. Par ailleurs, dans la cadre de sélection multiple co-localisée, Kin *et al.* ont montré que l'interaction multi-points permet de diviser par deux le temps de pointage par rapport à une souris pour une tâche de sélection multiple [51].

Maintenant que les bases des périphériques d'entrée tactile multi-points sont posées, intéressons nous plus en détail à l'interaction et voyons pourquoi l'interaction directe absolue est attrayante mais soulève de nombreux problèmes.

1.4 Fondements de l'interaction directe absolue

L'interaction tactile absolue repose sur divers processus moteurs et cognitifs qui peuvent expliquer son succès, nous allons tout d'abord étudier comment nos sens nous permettent d'appréhender ce type d'interaction et pourquoi l'interaction absolue est facile à prendre en main. Nous présenterons ensuite les performances de l'interaction tactile absolue et verrons les limitations inhérentes.

1.4.1 Capacités sensorielles

Pour comprendre en quoi l'interaction tactile relève du sens du toucher, adoptons une approche physiologique du toucher. La somesthésie désigne la sensibilité du corps à lui-même et à l'environnement extérieur au travers du corps, cette capacité sensorielle repose sur l'ensemble des organes ayant une fonction de perception introspective : comme la peau, les tendons, les articulations, les os, les muscles. La somesthésie regroupe le toucher, les sens proprioceptifs et les sens intéroceptifs. Le toucher, qui fait partie des cinq sens extéroceptifs (vue, ouïe, odorat, goût, toucher), fournit des informations sur l'environnement extérieur. Les sens proprioceptifs perçoivent la position, les mouvements et l'état du corps. Les sens intéroceptifs sont en relation avec le fonctionnement interne des organes et de l'organisme. Nous ne nous attarderons pas sur cette catégorie sensorielle car elle sort du cadre de l'interaction homme-machine et concerne davantage la médecine (réflexe de déglutition, satiété, informations vasculaires).

Le sens du toucher

Le toucher est une sensibilité de surface, qui est perçue par la peau sur l'ensemble du corps. Plusieurs types de capteurs répartis dans le derme (couche inférieure) de la peau compose l'information. Les déplacements, mouvements, compressions, étirements et vibrations de la partie superficielle de la peau sont intégrés pour ressentir l'état des géométries, textures et rugosités [41].

Les interfaces tactiles sont donc d'un point de vue biologique extrêmement pauvres en informations tactiles : la surface plate et inerte du périphérique ne retourne qu'une information haptique passive. L'espace d'entrée présente un état de surface constant, une certaine rugosité et un niveau de friction déterminé, là où un bouton ou une touche de clavier délivrent un retour haptique quand on les presse. De multiples travaux ont été menés pour rendre la richesse de l'exploration tactile : modification de géométrie de surface via des matrices de points ou de picots [70], ajustement de l'orientation de la surface pour simuler des géométries macroscopiques [105], ou modulation du coefficient de friction par électro-vibration [5] ou par effet film d'air [2]. Ces travaux sortent de notre domaine d'étude et font partie des interfaces haptiques.

L'intérêt des interfaces tactiles ne provient donc pas du toucher au sens premier, mais elles tirent plutôt profit du second aspect de la sensibilité somesthésique : la capacité à situer ses membres dans l'espace et le rapport de coordination amené par un asservissement et une connaissance inconsciente du corps.

Les sens proprioceptifs

Les sens proprioceptifs regroupent le sens de l'équilibre, la détection de la température corporelle, la sensibilité à la douleur et la proprioception. La proprioception est composée de la kinesthésie qui rend compte de la dynamique et des mouvements des parties du corps, et la statesthésie qui rend compte de la statique et de la position des membres. Ces informations sont captées par les muscles, les tendons et les articulations. Les fuseaux neuro-musculaires captent la longueur du muscle, les organes tendineux de Golgi sont sensibles à la tension du muscle et les récepteurs articulaires reportent la position des articulations [75].

Ces informations permettent par exemple de pouvoir toucher notre nez avec notre index tout en fermant les yeux, mais également de situer notre main dans l'espace sans faire appel à la vision lorsque quelqu'un la déplace. Le cortex cérébral comprend deux aires dédiées au contrôle du corps : le cortex somesthésique primaire et le cortex moteur (figure 1.7). Les capteurs somesthésiques sont projetés dans le cerveau en fonction de leur densité. Ainsi les mains occupent une place plus importante que l'avant bras ou que la totalité du dos. Il en est de même pour le cortex moteur en fonction de l'importance et de la liberté musculaire des différentes parties du corps. La conjonction de ces deux aires apporte une dextérité plus ou moins remarquable et le retour d'informations continu des parties du corps contribue à réaliser un asservissement qui permet un ajustement de la tension et de la force des muscles afin de réaliser précisément un mouvement. Cette variation du tonus musculaire s'appuie sur la sensibilité proprioceptive inconsciente. Nous ne modulons heureusement pas consciemment nos muscles agonistes et antagonistes pour réaliser le moindre mouvement et nous ne nous concentrons pas perpétuellement sur la position de chacun de nos membres pour ajuster notre équilibre [72]. Par contre, nous pouvons prendre pleinement conscience du mouvement et de la position de nos membres.

Les membres étant complètement incarnés par la proprioception, nous bénéficions d'une mémoire musculaire, qui permet de réaliser des déplacements précis et mesurés sans recourir à la vision. Lorsque nous passons du clavier à la souris pour la saisir sans la regarder, nous nous servons de notre mémoire spatiale et de la proprioception. Lorsque nous entrons du texte au clavier sans regarder les touches, nous nous appuyons sur la mémoire musculaire qui permet d'automatiser des déplacements et d'atteindre facilement des positions répétées plusieurs fois.

Quand un utilisateur déplace un curseur au travers d'une fonction de transfert non-linéaire, c'est le retour d'informations visuelles qui fournit la plupart des

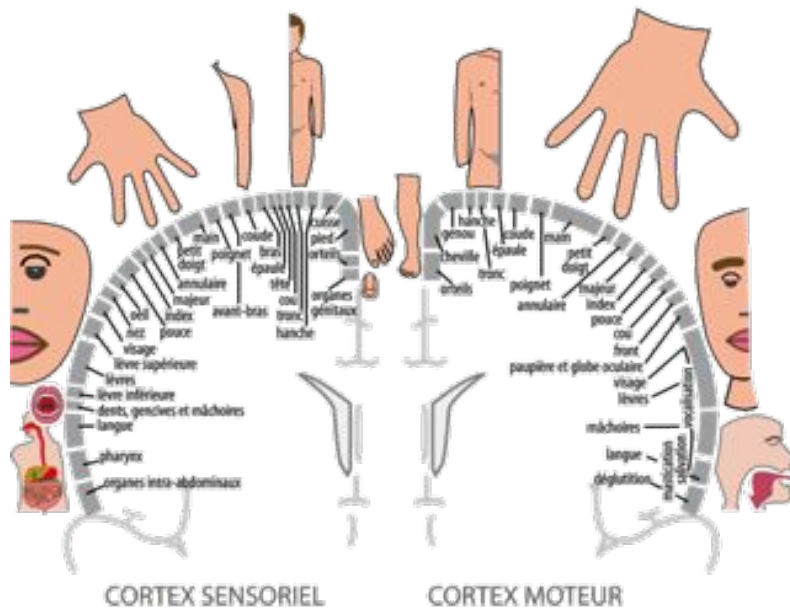


FIGURE 1.7 – Aires somesthésique (à gauche) et motrice (à droite) (issu de¹⁰).

données nécessaires à la coordination de la main et du bras. Cependant quand l'utilisateur contrôle un système absolu co-localisé ou séparé, le haut niveau de similarité entre espace d'entrée et espace visuel ainsi que le couplage entre la position des doigts et des curseurs [66] font que les déplacements dans l'espace physique peuvent s'appuyer sur la mémoire spatiale et musculaire.

1.4.2 Évaluation de la précision du pointage

Rapidement, l'intérêt pour les surfaces tactiles a amené à se poser la question de la précision accessible avec ce type de périphérique et a motivé un grand nombre d'études. Celles-ci commencent dès les années 80, Hall *et al.* reportent avec un écran tactile IBM InfoWindow des taux de réussite variant de 66,7% pour des cibles de 10 mm à 99,2% pour des cibles de 26 mm [38]. Plus récemment, dans une étude s'intéressant à l'impact de l'âge sur les performances de l'interaction tactile, Siek *et al.* ont relevé que la sélection de petites cibles est difficile en dessous de 10-15 mm [80]. En évaluant leur technique *Shift* sur un PDA (Personal Digital Assistant), Vogel et Baudisch ont trouvé une taille minimale de cibles de 10,5 mm [89]. Avec une surface plus grande : une table interactive FTIR, Wang et Ren ont trouvé une taille légèrement plus grande de 11,5 mm [94]. Holz et Baudisch qui étudiaient la répartition et les caractéristiques des touches autour d'un cible avec une surface tactile FingerWorks iGesture, ont reporté une taille de 15 mm [43]. Les auteurs de cet étude introduisent le *General Input Perceived Model* et suggèrent que la précision en entrée peut être améliorée en identifiant l'utilisateur et en prenant en compte sa façon de pointer. En effet les systèmes tactiles utilisent le centre du contact pour extraire un unique point de contact, or l'utilisateur ne peut pas savoir

10. https://interstices.info/jcms/i_62192/quand-le-cerveau-parle-aux-machines

exactement où va se situer ce point sous son doigt (figure 1.8. De plus chaque utilisateur pointe avec une partie différente de son doigt : par exemple le bout du doigt, l'extrémité de l'ongle ou encore le milieu de l'ongle. Cette incertitude entre le point d'entrée estimé et le point d'entrée perçu par le système est complémentaire au problème de gros doigt [80, 64, 43].

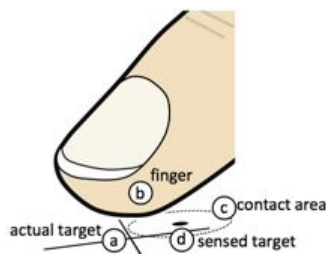


FIGURE 1.8 – Point d'entrée perçu (d) et estimé (a) (issu de [44]).

Dans le contexte spécifique de saisie sur claviers virtuels, les études ont montré que les participants avaient de meilleures performances avec des touches légèrement plus grandes que celles des claviers physiques. La taille minimale recommandée est de 23 mm [78, 26] alors que la taille standard de 19 mm est apparue comme trop petite lors d'essais préliminaires [26].

1.4.3 Problèmes de précision et de sélection

Les travaux que nous venons de citer montrent tous que l'interaction tactile absolue manque de précision. Un objet est facilement sélectionnable s'il mesure au moins 10 à 15 mm, or les systèmes actuels comportent beaucoup de cibles bien plus petites : menu, boîtes à outils, sélection libre. Plusieurs causes peuvent expliquer ce manque de précision, tout d'abord les écrans tactiles incitent à aller au contact du contenu, mais dès que le doigt entre en contact avec la surface, la sélection commence. Il n'existe pas d'état intermédiaire entre « non-suivi » et « posé ». Cette immédiateté de sélection empêche l'utilisateur de poser son doigt sans réaliser d'action et est parfois qualifiée de doigt en or ou toucher de Midas [71, 29] : l'utilisateur ne peut ni montrer quelque chose à l'écran à quelqu'un, ni bénéficier du survol pour se renseigner sur les possibilités d'interaction grâce aux *tooltips* (infobulles).

Le manque de l'état de survol a été formalisée par Buxton [18] et pose un autre problème désormais fort connu sous le nom de *fat finger* (gros doigt) [80, 64]. Lorsque le doigt entre en contact avec l'écran, il occulte la cible pointée, l'utilisateur se retrouve dans l'impossibilité d'ajuster finement la sélection car il ne voit pas la sélection effectuée sous son doigt. De plus, comme nous venons de le voir, Holz et Baudisch ont montré que l'utilisateur ne sait pas quel point du doigt est considéré comme le point d'entrée par le système [43], ce qui entraîne un décalage supplémentaire entre la cible visée et le contact réel.

De très nombreux travaux ont essayé de pallier à ce manque de précision : en proposant des stratégies de sélection au sortir de la cible [71], en interagissant

derrière l'écran [97], en décalant le toucher et un curseur [89, 48, 73], en détectant et en s'adaptant à l'utilisateur [43], ou plus récemment en ajoutant un petit écran sur le doigt [83]. Par ailleurs, des occultations sont aussi causées par l'ensemble main-poignet-bras, ce qui peut cacher certains éléments utiles à l'interaction comme des menus contextuels ou des options. Vogel *et al.* ont étudié ce phénomène et ont proposé de modéliser la position de l'ensemble main-poignet-bras en fonction de la position des contacts afin de pouvoir déplacer les éléments occultés [88, 91].

Dans le cas des interfaces indirectes, la question de la prévisualisation du contact est encore plus présente, ainsi le survol est souvent réintroduit au moyen d'un bouton ou de la détection du survol d'un stylet. Cependant, les caractéristiques des espaces d'entrée et d'affichage pouvant différer, des travaux se sont donc naturellement portés sur la définition des facteurs influençant les performances et sur la mesure de leur impact.

1.4.4 Facteurs influençant la performance

De nombreux travaux ont donc étudié les impacts de divers facteurs affectant les tâches de pointage indirect et ont essayé de quantifier la capacité à passer de l'espace d'affichage à l'espace de contrôle avec différentes transformations.

Cunningham [22] a étudié la mise en correspondance visuelle-motrice en plaçant un panneau occultant entre l'utilisateur et la tablette graphique. Elle a mesuré l'impact de la rotation et d'une symétrie sur le temps, la trajectoire et l'orientation initiale du mouvement en capturant la trajectoire du stylet allant vers une cible. L'approche psychologique de cette étude sur un tâche de positionnement indirect absolu avec stylet a permis d'établir un modèle reliant les performances à la rotation entre la tablette et l'écran. Elle a trouvé de meilleurs résultats lorsque l'orientation de l'espace d'entrée se trouve entre -45° et 45° par rapport à l'écran vertical. Les faibles rotations ont peu d'influence et sont rapidement assimilées par les participants.

Wigdor *et al.* ont confirmé par la suite ces résultats en étudiant les effets de la position d'écran et de l'orientation de l'espace d'entrée sur les préférences et les performances pour les tâches de positionnement absolu indirect [98]. Les participants ont majoritairement préféré une position d'écran décalée à 45° en face d'eux (position Nord-Ouest et Nord-Est plutôt que Nord). L'orientation la plus appropriée pour l'espace d'entrée dépend de la position de l'écran. Pour les positions d'écrans face à l'utilisateur (NO, N, NE), les meilleures performances étaient obtenues avec une orientation de 45° par rapport au Nord.

Schmidt *et al.* ont comparé les performances de mises en correspondance absolues directe et indirecte pour des tâches de positionnement multi-points sur une table interactive de 100 cm de diagonale [77]. Dans la condition indirecte, les contours des bras étaient affichés dans l'espace visuel afin de fournir la position de survol, les contacts étaient symbolisés par des cercles. Le temps nécessaire pour réaliser la tâche et le taux d'erreur observés ont été meilleurs dans le cas d'interaction directe. Les auteurs avancent que les performances

moindres de l'interaction indirecte peuvent avoir été causées par l'affichage des contours : la moitié des participants se sont plaints de la difficulté à coordonner les déplacements de leurs mains avec le retour vidéo. En particulier, la plupart des participants positionnaient l'ombre de leur main au-dessus d'une cible, puis tâtonnaient pour aller au contact de la surface et sélectionner la cible. La gestion de la hauteur de la main était séparée du déplacement en deux dimensions de la main dans un plan parallèle à la table.

La précision a également été étudiée dans le contexte de mises en correspondance absolues indirectes sans informations de survol présentée à l'écran. Lin *et al.* ont étudié la sélection de cibles en 1 dimension le long de l'avant-bras [57]. Les participants devaient toucher un ensemble de 5 à 9 points entre le poignet et le coude de leur avant-bras non dominant (22 cm de longueur moyenne), répartissant eux-même les cibles sur zone autorisée. Le dispositif expérimental empêchait les participants de voir leur avant-bras et la main dominante. Ils ont été capables de sélectionner correctement un ensemble comprenant 6 points, certains ont pu acquérir de manière sûre jusqu'à 8 points. Gustafson *et al.* ont évalué la précision de mémorisation pour une tâche de pointage 2D [35]. Les participants devaient réaliser des sélections dans leur paume non dominante et dans les airs, ils apprenaient auparavant les positions des cibles en les sélectionnant plusieurs fois directement sur un iPod Touch. Les auteurs ont ensuite calculé une taille minimum de cible qu'un utilisateur peut atteindre dans 95% des cas, ils reportent des tailles de 27,9 mm pour le pointage dans les airs et de 17,7 mm dans la paume. À la suite de ces travaux, les auteurs ont complété leur étude sur les interfaces imaginaires avec une tâche d'exploration aveugle [36]. Les participants devaient chercher et sélectionner une cible particulière en déplaçant le doigt au travers de la surface d'entrée, le nom des cibles était annoncé par le système lorsque le doigt passait dessus. Les différentes surfaces d'entrée évaluées étaient la paume non dominante, une main en caoutchouc, des écrans recouverts d'une grille en relief. Les temps nécessaires à la sélection ont montré que lorsque les participants avaient la possibilité de regarder la surface d'entrée, ils réalisaient leurs meilleures performances. Dans les conditions où les participants n'étaient pas autorisés à regarder la surface d'entrée, les informations retournées par la main non dominante – via la peau – étaient plus importantes et aidaient davantage que les informations retournées par l'index qui se déplace sur les grilles ou dans la paume en caoutchouc. Dans ces dernières études, les espaces d'entrée et d'affichage ont des rapports de cadre similaires et des tailles similaires, il n'est pas détaillé comment ces facteurs peuvent affecter la mémorisation spatiale et les performances.

Au-delà de la similarité entre les espaces d'entrée et de sortie qui favorise la disparition du périphérique au profit de l'interaction, de nouveaux concepts d'interaction ont été développés parallèlement offrant un engagement plus important de l'utilisateur [45].

1.4.5 Manipulation indirecte et directe

L'introduction de plusieurs points d'interaction augmente la bande passante entre l'utilisateur et la machine [65, 51], mais cela ne suffit pas à rendre les objets et concepts virtuels plus tangibles. En 1997, Beaudouin-Lafon nomme ceci la *réification* : processus qui transforme un concept ou une commande en objet manipulable [7, 8, 10] et introduit en même temps la notion de *manipulation indirecte* en opposition à la *manipulation directe* définie par Shneiderman en 1983 [79].

Les interactions à manipulation directe respectent les principes suivants :

- Représentation continue des objets d'intérêt ;
- Actions physiques sur les objets plutôt que syntaxe complexe ;
- Opérations rapides, incrémentales et réversibles dont l'effet sur les objets d'intérêt est immédiatement visible ;
- Approche incrémentale qui permet d'utiliser un système sans en connaître toutes les fonctionnalités et de les découvrir au fur et à mesure.

La réalisation d'une tâche doit se rapprocher du modèle mental de l'utilisateur et de ses connaissances du monde réel et des liens de causalité. Il faut montrer et agir sur des objets, incarner plutôt qu'abstraire les notions. Rutkowski et Hutchins se placent plutôt du côté de l'utilisateur pour qualifier la manipulation directe. Ainsi Rutkowski parle du *principe de transparence* où l'utilisateur ressent un engagement direct avec les objets qu'il manipule : « The user is able to apply intellect directly to the task ; the tool itself seems to disappear » [74]. Les concepts d'interruption et d'annulation sont matérialisés par des touches spécifiques sur le clavier. Hutchins *et al.* définissent également la manipulation directe par le sentiment de contrôle direct, sans intermédiaire et la réduction des distances sémantiques, l'utilisateur perçoit le système informatique comme un environnement réel ou ayant des caractéristiques semblables [45].

L'expression *manipulation indirecte* désigne la plupart des interfaces actuelles où l'utilisateur manipule des concepts au travers de menus et de boîtes de propriétés, où les actions ne sont pas toujours répercutées instantanément mais après validation. L'objet d'intérêt central est entouré d'une multitude de concepts entravant le chemin de la manipulation directe. Par exemple, l'utilisation de boîtes à outils peut être remplacée par l'introduction de *toolglass* [16]. Une boîte à outils contenant plusieurs outils est matérialisée et est manipulée par la main non dominante, l'interaction au travers de cette palette permet de désigner à la fois l'outil et l'objet d'interaction. Ce type d'interaction est similaire à l'utilisation d'une règle, d'un trace-cercle ou d'un pistolet de dessinateur où l'utilisateur manipule un même outil lui permettant de réaliser plusieurs fonctions (figure 1.9).

Si nous reprenons l'exemple de zoom de la figure 1 de l'introduction, des boutons « Loupe +/- » sont une manipulation indirecte pour agrandir le document. Un représentant physique de la loupe serait un lentille déplacée par le périphérique et affichant temporairement une vue magnifiée du document. Une manipulation directe et permanente de l'échelle d'affichage peut être de tirer le document vers soi (ou d'approcher le point de vue) à l'aide de la molette ou en écartant/resserrant deux doigts (pinch-to-zoom [54]).



FIGURE 1.9 – Parents physiques des toolglass. À gauche, un trace-cercle (MINERVA Trace-écrous et boulons n°2) permettant de tracer des cercles de différentes tailles et diverses autres formes. À droite, une collection de pistolets de dessinateur servant à dessiner toutes sortes de courbes (issu de ¹¹).

Afin d'améliorer les performances et de dépasser les problèmes d'occultation, de portée et de précision, de nombreuses techniques d'interaction ont été proposées. Ces techniques s'affranchissent du dogme de l'interaction absolue en découplant, à des degrés divers, le curseur de la position du doigt.

1.5 Évolution de l'interaction tactile multi-points

Lorsque l'on sélectionne un objet dans une interface tactile, on met le doigt dessus, nous avons vu que l'occultation des éléments et particulièrement le problème du « gros doigt » rendent incertain le positionnement précis du doigt.

L'interaction directe absolue implique une relation d'équivalence entre le doigt et le curseur, ceci crée une contrainte forte qui peut réduire les possibilités des techniques d'interaction alors qu'elles pourraient tirer parti des capacités dynamiques d'affichage pour supporter un accès moins « direct » mais plus riche. Les premiers travaux abordés vont introduire une légère indirection entre le toucher physique et l'action : les lieux d'action et d'interaction ne seront plus exactement les mêmes.

1.5.1 Interaction co-localisée

Tout d'abord des études s'intéressant aux surfaces tactiles de petites dimensions, comme un téléphone, proposent de décaler le point d'entrée du doigt. Potter *et al.* proposent très tôt de créer un curseur décalé (*Offset cursor*) au-dessus du doigt [71]. Ceci permet d'indiquer clairement le point d'interaction mais nécessite de différencier le survol de la sélection. Il ont comparé cette technique à l'interaction classique, les résultats ont montré un temps de sélection légèrement plus long mais une réduction du nombre d'erreurs de moitié. La sélection sur les bords peut rester compliquée, particulièrement en bas de l'écran. De plus, l'utilisateur ne peut plus placer directement son doigt sur une cible : il doit toujours penser à interagir avec un décalage, ce qui augmente la

11. <http://commons.wikimedia.org/wiki/File:Tekenschaats.JPG>

charge cognitive. Vogel et Baudisch proposent avec *Shift* de placer à côté du doigt une vue de ce qu'il y a sous le doigt [89]. Cette vue affiche le curseur et son environnement immédiat, l'utilisateur peut donc positionner précisément le curseur au travers d'une fonction de transfert non-linéaire avant de valider la sélection en retirant le doigt.

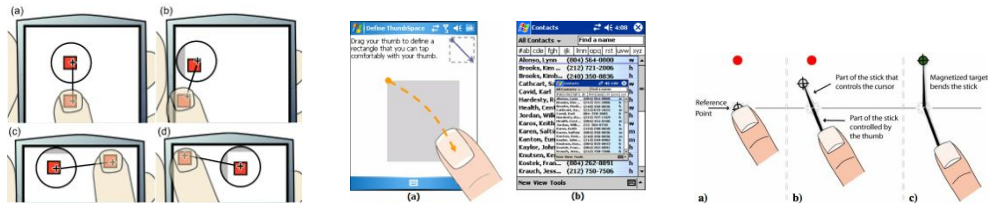


FIGURE 1.10 – *Shift*, *ThumbSpace* et *MagStick* (issus respectivement de [89, 48, 73]).

Les contraintes des petits appareils tactiles soulèvent déjà la question de l'accessibilité à l'ensemble de l'écran : en effet les mouvements du pouce sont limités car la main doit tenir l'appareil en même temps. Karlson et Bederson choisissent de créer un pavé tactile miniature absolu par dessus l'écran [48], les dimensions de la zone tactile sont ajustées afin de la contenir à l'intérieur de la zone atteignable par le pouce (*ThumbSpace*). Lorsque l'utilisateur pose son pouce, un curseur est créé à la position absolue, une mise à l'échelle assure le passage du pavé tactile vers la totalité de l'écran. L'interaction résultante est précise grâce à un curseur se déplaçant uniquement vers les cibles potentielles et sautant par dessus les espaces vides (pointage objet [33]). Cette technique se révèle assez lente, mais surtout le centre est difficile à atteindre en raison des occultations dus au pouce et au pavé tactile. Avec *MagStick* [73], Roudaut *et al.* réduisent les occultations et augmentent la portée du curseur en proposant de le déplacer comme un pantographe inversé. De façon similaire aux travaux précédents, l'utilisateur touche l'écran pour créer un curseur à proximité du doigt. Les mouvements du curseur sont inversés autour d'un pivot : ceci permet de déplacer un curseur distant tout en éloignant le pouce des objets d'intérêt. La précision est obtenue par l'attraction du curseur par les cibles, le pointage sémantique agit sur la fonction de transfert pour réduire les espaces vides [17]. Une baguette brisée au point de pivot relie le doigt au curseur, cet artifice assure que le toucher et le curseur sont clairement liés pour l'utilisateur et facilite la compréhension de la technique. *MagStick* obtient de meilleurs résultats que *Shift* malgré le caractère plus indirecte de l'interaction, le doigt ne touche plus réellement la cible.

Sur des écrans tactiles de plus grandes dimensions, le curseur est encore plus affirmé et on se rapproche de l'interaction souris de manière à utiliser des applications classiques et à augmenter la précision. Albinsson et Zhai transforment le curseur en widget d'interaction : ils créent des interfaces pour déplacer précisément un curseur [1]. Ils conçoivent différents curseurs qui restent affichés à l'écran et sont déplaçables de diverses manières (figure 1.11). Dans un processus itératif, les auteurs aboutissent au *Precision Handle* : dans un premier temps, l'utilisateur positionne le curseur à proximité de la cible, puis le doigt contrôle le curseur en déplaçant une petite zone situé à coté, les déplacements

de la zone sont répercutés au curseur au travers d'une fonction de transfert. Ils obtiennent de bons résultats pour *Precision Handle*, qui s'avère efficace pour sélectionner de petites cibles mais requiert un peu plus de temps que la sélection avec agrandissement.

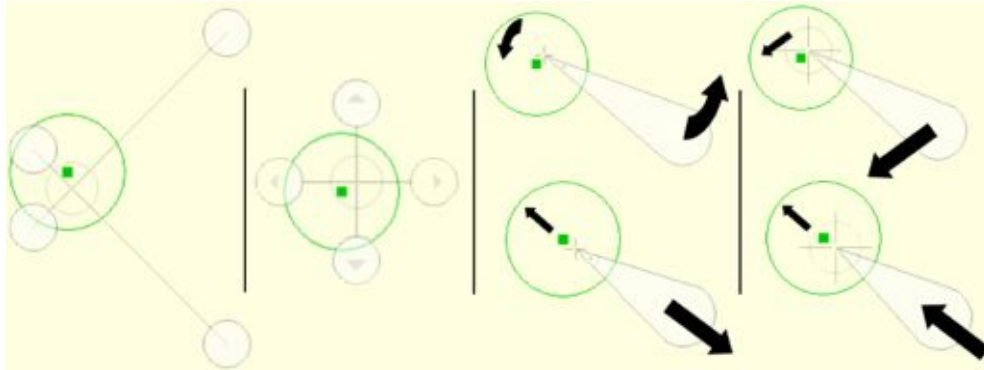


FIGURE 1.11 – Conception de widgets de sélection. *Cross-Lever*, le curseur est à l'intersection des deux lignes déplaçables par leurs extrémités. *Cross-Keys*, le curseur est déplacé avec les flèches. *2D Lever*, la zone de contrôle agit sur le curseur avec une symétrie autour d'un pivot pour réduire les déplacements. *Precision Handle*, les mouvements de la zone de contrôle sont répercutés sur le curseur après mise à l'échelle. (issu de [1]).

Avec *Surface Mouse*, Bartindale *et al.* affichent une souris virtuelle sous la main de l'utilisateur. En détectant la paume et les cinq doigts de l'utilisateur, l'image d'une souris apparaît pour prendre le contrôle du curseur [4]. L'utilisateur peut agir comme s'il tenait une souris réelle : contrôle relatif avec fonction de transfert, sélection en posant ses doigts sur les boutons et défilement de documents en utilisant la molette. D'autres travaux comparables proposent d'émuler l'interaction souris pour contrôler le curseur, mais utilisent un ensemble de gestes multi-doigts. Esenther et Ryall proposent une technique très simple *Fluid DTMouse* [24], ils affichent un curseur entre le majeur et le pouce, laissant à l'index le rôle d'activer la sélection. La position moyenne des deux doigts détermine l'emplacement du curseur, la stabilité et la précision sont donc améliorées sans avoir recours à une fonction de transfert, cependant la technique se révèle incapable d'atteindre les bords de l'écran. Matejka *et al.* étudient et comparent différents types de gestes multi-doigts pour émuler le survol, la sélection et le défilement [61]. La technique *SDMouse* (*Side+Distance*) émerge de leur travaux, lorsque l'utilisateur pose un doigt : un curseur en état de survol apparaît légèrement au-dessus du doigt. Un contact à gauche du doigt déclenche la sélection (clic gauche), un contact proche à droite est associé au clic de la molette, un contact un peu plus loin permet de simuler le clic contextuel (ou clic droit). Ces deux derniers travaux n'abordent toutefois pas les problèmes de portée et de fatigue et se positionnent davantage sur les problématiques de précision et de survol.

Pour faciliter l'accès à des cibles distantes sur des murs d'écrans interactifs, Forlines *et al.* permettent à l'utilisateur de combiner le pointage direct et le déplacement relatif d'un curseur [27]. Un widget suit le lieu d'interaction,

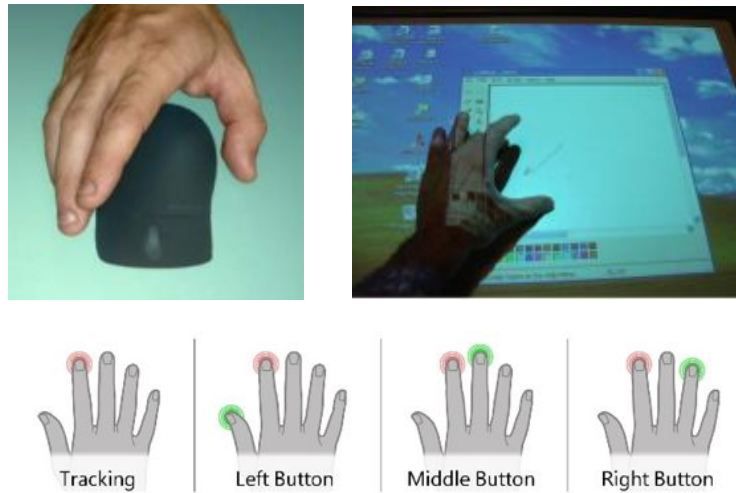
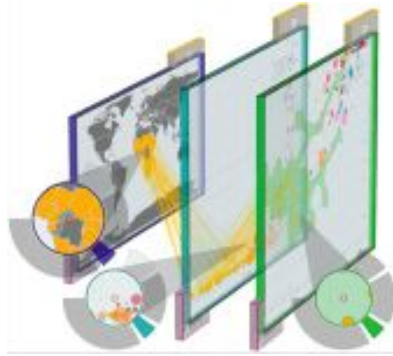


FIGURE 1.12 – *SurfaceMouse*, *FluidDTMouse* et *SDMouse* (issus respectivement de [4, 24, 61]).

en le sélectionnant l'utilisateur va créer un curseur qui est contrôlé avec le stylet au travers d'une fonction de transfert. L'interaction directe reste plus rapide pour les cibles proches, mais le contrôle relatif se montre plus rapide pour les cibles distantes. La technique n'est pas perçue comme compliquée par les utilisateurs et assure un bon moyen d'accéder à distance malgré quelques erreurs en changeant de mode.

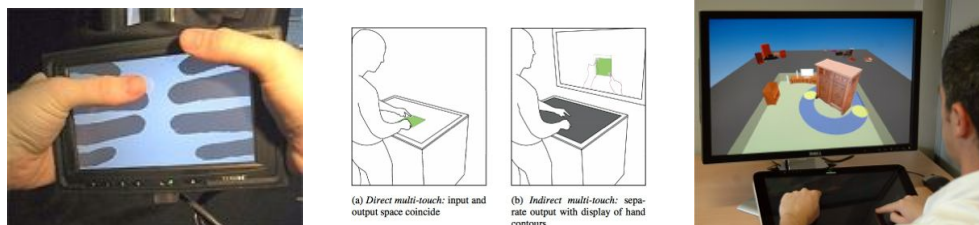
Vlaming *et al.* proposent eux aussi de réintégrer un curseur de souris au sein de l'interaction tactile afin d'offrir une plus grande précision et une meilleure lisibilité des informations géographiques affichés dans un environnement 3D [87]. Leur logiciel de visualisation de données représente plusieurs couches d'informations par plusieurs feuilles de classeur en 3D (figure 1.13), pour chacune de ces feuilles l'utilisateur peut manipuler un widget permanent, appelé *Rizzo*, qui contrôle un curseur. Ce widget combine les fonctionnalités d'une souris virtuelle et d'une lentille de visualisation : les déplacements relatifs d'un doigt contrôlent le curseur au travers d'une fonction de transfert, la vue assure une vision précise et non déformée par la perspective des objets autour du curseur et peut être magnifiée pour augmenter le niveau de détail.

Dans tous ces travaux, le doigt et le curseur commencent à être séparés. Un curseur apparaît plus ou moins loin du doigt et indique la position du point d'interaction compris par le système, l'utilisateur peut corriger l'emplacement du curseur de différentes manières pour affiner la sélection. Bien souvent l'indirection est créée spécialement pour l'occasion, une cible difficile par exemple, et disparaît juste après la sélection. Les problèmes de précision et d'accès distants trouvent donc une solution mais qui réduit l'interaction multi-points à un seul point d'interaction dans le curseur.

FIGURE 1.13 – *Rizzos* (issu de [87]).

1.5.2 Interaction dé-localisée absolue

Au delà de la séparation du curseur et du doigt, d'autres travaux ont proposé de séparer l'espace d'entrée de l'espace d'affichage toujours pour contourner le problème du gros doigt. Wigdor *et al.* placent une surface tactile directement derrière l'écran : l'utilisateur pose ses doigts au verso de l'écran [97]. La colocalisation des deux espaces pourtant distincts permet d'obtenir de bons résultats pour sélectionner des cibles, cependant leurs conclusions indiquent que certains participants souhaitent désactiver l'affichage des contours de la main et ne conserver que les points indiquant les contacts et le survol. La difficulté à trouver un retour vidéo adapté avait également été rencontré par Schmidt *et al.* dans leurs expériences comparant l'interaction directe sur une table et indirecte avec une table et un écran déporté [77]. L'affichage des contours avait impliqué une séparation du positionnement et de la sélection, dégradant grandement les temps nécessaire à la réalisation de la tâche de positionnement 2D. Knoedel et Hachet ont comparé l'interaction directe et indirecte dans une tâche de positionnement visualisée en 2D et en 3D [52]. Leurs résultats montrent que l'interaction directe est plus rapide mais que l'interaction indirecte est plus précise que l'interaction directe, la trajectoire était également meilleure. Dans le cas indirect les curseurs étaient limités aux déplacements dans le plan du sol, la surface du périphérique d'entrée était mise en correspondance avec le sol visible en vert dans la figure 1.14 à droite.

FIGURE 1.14 – *LucidTouch*, interaction indirect avec affichage des contours, interaction indirecte pour une tâche visualisé en 3D (issus respectivement de [97, 77, 52]).

1.5.3 Interaction quasi-absolue

En séparant la surface d'entrée de l'espace visuel, les tailles des ces deux espaces n'ont plus besoin d'être nécessairement identiques ; ainsi un périphérique de dimensions modestes peut être utilisé pour accéder à un écran sensiblement plus grand. Le facteur de mise à l'échelle ayant alors un impact négatif sur la précision de l'interaction : les gestes sont multipliés pour couvrir l'ensemble de l'écran, ce qui induit une perte de précision. Pour remédier à ce problème, McCallum et Irani proposent d'indiquer tout d'abord l'emplacement grossier de la cible, puis d'ajuster finement la position du curseur [62]. Un geste de tap assure un contrôle absolu du curseur le téléportant ainsi directement à l'emplacement désigné. Les gestes de glisser contrôlent le curseur de manière relative comme un pavé tactile. Grâce à cette technique, les auteurs permettent d'interagir sur un écran de 132 cm avec un téléphone tactile de 7,1 cm. Une étude préliminaire montre que ARC-Pad permet de se déplacer plus rapidement vers les cibles que l'interaction relative classique et est appréciée des utilisateurs, cependant cette technique n'autorise que le contrôle d'un unique curseur.



FIGURE 1.15 – *ARC-Pad* (issus de [62]).

Dans l'objectif de garder une mise en correspondance absolue et pour autoriser la manipulation multi-doigts, plusieurs travaux ont proposé de créer un sous-espace de travail à l'écran. Ce sous-espace est contenu dans l'écran et une mise en correspondance absolue entre ce sous-espace et le périphérique d'entrée assure une prise en main aisée de l'interaction. Satyanarayan *et al.* ont ainsi adressé leur très grande surface d'affichage *HIPerSpace* avec différents périphériques comme des écrans tactiles, des téléphones tactiles et des stylos digitaux afin de supporter la collaboration de plusieurs utilisateurs [76]. L'usage de sous-espace implique de pouvoir modifier la taille de l'espace de travail pour ajuster la précision et déplacer ce sous-espace pour accéder à différentes parties de l'écran. Le contrôle des *Overlays* est réalisé par des gestes à cinq doigts, facilement distinguables des sélections et manipulations engageant moins de doigts. Les contacts sont projetés dans l'espace de travail et représentés par des disques colorés, l'utilisateur agit de la même façon qu'en interaction directe. Malik et Laszlo ont conçu un périphérique tactile indirect filmant les mains (*VisualTouchpad* visible dans la figure 1.16 à droite) et permettant une interaction indirecte absolue vers l'ensemble de l'écran [58]. Malik *et al.* ont ensuite complété l'interaction en proposant de distinguer les deux mains pour déplacer un espace de travail avec la main non dominante et une main virtuelle à l'intérieur de ce sous-espace grâce à la main dominante [59]. Le périphérique avait une dimension de 60×20 cm et était divisé

en deux parties, chaque main évoluant dans sa moitié. La partie associée à la main non dominante était mise en correspondance avec la surface de l'écran, l'autre moitié était mise en correspondance avec l'espace de travail. L'acquisition de l'image de la main leur permettait d'afficher les mains en transparence à l'écran et de reconnaître divers gestes pour contrôler et modifier l'espace de travail, atteindre ou sélectionner des objets.

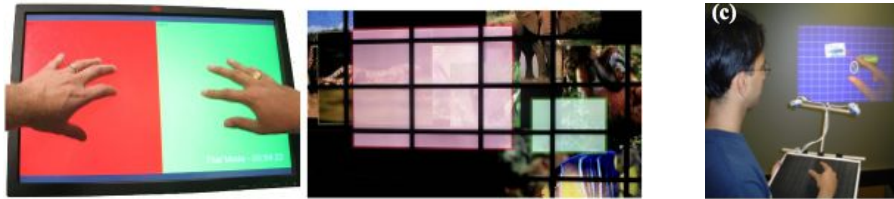


FIGURE 1.16 – *Overlay* et *VisualTouchpad* (issus respectivement de [76, 58]).

D'autres travaux reposent sur les mêmes principes de séparation de l'espace d'entrée et de l'espace de travail mais en restant sur la même surface. Ces travaux, s'intéressent davantage à la précision qu'à l'accès distant. En effet, la haute résolution des écrans permet d'afficher des objets très détaillés et des cibles très petites, or nous avons vu précédemment que la taille de cibles minimale aisément sélectionnable est située entre 10 et 15 mm. Volda *et al.* introduisent l'*i-Loupe* dans l'interaction directe pour résoudre ces problèmes d'incompatibilité de résolutions[92] (figure 1.17 à gauche). Sur une table interactive, un espace de contrôle et un espace de travail sont créés au-dessus d'une application multi-points, la position absolue des doigts entrant dans l'espace de contrôle est reportée dans l'espace de travail pour interagir avec les objets se situant sous les curseurs. Ici les positions et dimensions des espaces sont directement modifiables via les bordures de ceux-ci. Selon les tailles relatives des deux espaces, l'utilisateur peut créer un agrandissement (comme visible dans la figure 1.17) ou une vue radar permettant de prendre du recul ou d'accomplir de grand déplacements. Avec le *FingerGlass* de Kaser *et al.*, l'utilisateur crée et positionne un sous-espace de travail avec une main, cela fait apparaître une lentille déportée agrandissant la vue de l'espace de travail et rendant possible une interaction précise en sélectionnant directement les objets au travers de la lentille [49]. L'utilisateur peut déplacer le sous-espace en maintenant des objets ou déplacer de manière relative un curseur à l'intérieur de l'espace de travail avec la seconde main.

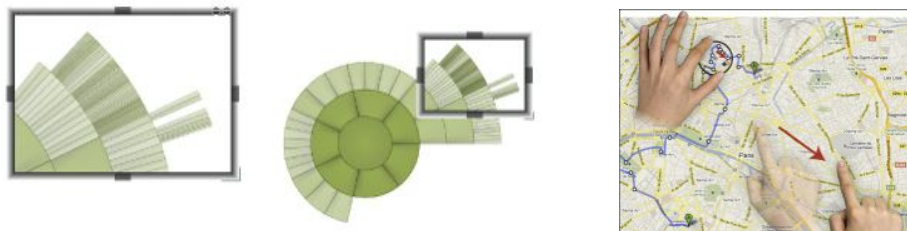


FIGURE 1.17 – *iLoupe* et *FingerGlass* (issus respectivement de [92, 49]).

Ces techniques permettent d'améliorer la précision tout en supportant une interaction multi-doigts facile à prendre en main. Cependant, il y a un prix à payer : l'utilisateur ne peut accéder qu'à une zone restreinte de l'espace visuel. Il doit en plus de l'interaction avec les objets gérer les déplacements de l'espace de travail et en adapter la taille en fonction de ses besoins.

1.5.4 Interaction relative multi-points

Afin de regrouper la gestion de l'espace de travail avec l'interaction sur les objets, plusieurs travaux se tournent vers le contrôle relatif de l'espace de travail. Yang *et al.* placent un écran tactile affichant la zone autour du curseur souris, permettant d'introduire l'interaction directe au sein de l'interaction souris [104]. Ils étudient les avantages de placer une boîte à outils sur l'écran tactile et montrent que cela permet de réduire les déplacements et le temps nécessaires par rapport à la souris classique. Ils envisagent également différentes interactions dont une similaire à ARC-Pad [62] permettant de déplacer le curseur de manière absolue en plaçant le doigt sur l'écran tactile et une technique où l'écran affiche les objets autour du curseur. L'utilisateur pouvant ainsi sélectionner directement les objets avec l'écran tactile.

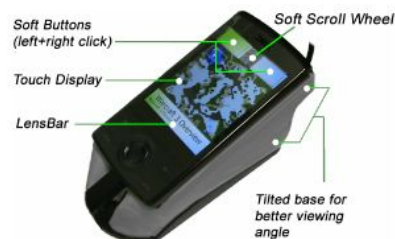


FIGURE 1.18 – *LensMouse* (issu de [104]).

Villar *et al.* ont aussi créé plusieurs prototypes de souris augmentées d'une surface tactile mais sans écran [86] (figure 1.19 à gauche). Les mouvements de la souris assurent un contrôle relatif de l'espace de travail représenté par un curseur. Les doigts posés sur la surface tactile sont affichés autour du curseur principal en fonction de leur position absolue sur la souris, le curseur principal en forme de flèche est accompagné de petits cercles : le *MT Cloud* (figure 1.19 à droite). Leurs travaux sur les souris multi-doigts ont été continués par Benko *et al.*, qui étudient les différentes possibilités du focus du curseur principal et des doigts, ainsi que l'activation de la sélection des contacts par un bouton [11]. En comparant diverses combinaisons, ils concluent que dans le cadre d'une interaction avec une souris augmentée, il est préférable de différencier explicitement la sélection du survol avec un bouton plutôt que de sélectionner dès que le doigt entre en contact. En effet les techniques ne proposant pas de survol pour les doigts ont nécessité plus de temps et ont entraîné plus d'erreurs. Parallèlement les techniques utilisant le curseur principal pour déterminer un objet-cible ont également présenté de meilleures performances que les techniques distribuant le focus et permettant de sélectionner les objets

sous chacun des doigts. Les auteurs argumentent que les souris multi-doigts sont considérées par les participants comme des souris augmentées et ne sont pas des substituts d'écrans tactiles. Ils ont observé que les participants prenaient la souris multi-doigts en main comme s'il s'agissait d'une souris classique, limitant ainsi la liberté de déplacements des doigts sur la surface. Le contexte d'utilisation d'une souris implique de garder des concepts similaires : la sélection des objets sous le curseur et la gestion de l'état de sélection avec un bouton.

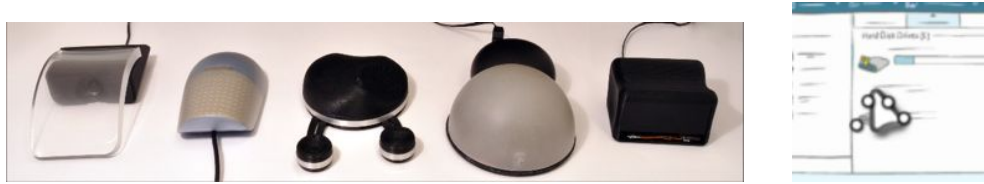


FIGURE 1.19 – Design de souris multi-doigts et design de curseurs multi-doigts (issus de [86]).

Auparavant, Moscovich et Hughes avaient déjà introduit le concept de curseur multi-doigts contrôlé entièrement par une surface tactile [65]. La position des doigts sur une surface tactile n'étant pas la même qu'avec une souris, leur technique a pu bénéficier de toute la liberté des doigts pour sélectionner plusieurs objets simultanément. Ils ont intégré l'utilisation de plusieurs doigts au sein d'un seul curseur en proposant le *HandCursor* [65], regroupant ainsi la manipulation d'objet et le déplacement du curseur. Tous les doigts posés sur la surface tactile apparaissent à l'intérieur d'un espace de travail représenté par un cercle à l'écran. Le premier doigt est placé au centre du cercle, les doigts suivants prennent place autour du premier doigt en respectant leur position absolue. L'espace de travail est déplacé au moyen d'une surface tactile multi-points et d'une fonction de transfert non linéaire. Le contrôle de celui-ci repose sur le déplacement du centre de gravité de l'ensemble des doigts pour que l'utilisateur puisse manipuler à la fois des objets et bénéficier de la fonction de transfert. Il peut prendre le contrôle de deux *HandCursor* avec deux périphériques distincts, un pour chaque main. Afin d'activer la sélection du curseur, un interrupteur était placé sous le dispositif tactile : la totalité de la surface s'enfonce légèrement en exerçant une pression raisonnable sur celle-ci, ce qui permet à l'utilisateur de cliquer et maintenir la sélection. 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 dynamique du nombre de curseurs délicat.

Les approches relatives s'appuient sur le curseur pour autoriser l'utilisation des interfaces classiques, la sélection de petites cibles et conserver le confort apporté par la souris. Mais, si l'aspect multi-doigts est bien pris en compte et permet d'augmenter le curseur, il n'y a souvent qu'un seul lieu d'interaction, la manipulation de plusieurs souris est rare et difficile et l'usage de plusieurs périphérique n'est pas des plus pratique.



FIGURE 1.20 – *HandCursor*, *SimilarityCursor* et *AdjustableAreaCursor* (issus de [65]).

1.5.5 Espace de solution : Continuum d'indirection

L'interaction directe absolue permet de poser les doigts sur les objets, mais restreint les possibilités et introduit de nombreux problèmes. Nous avons vu différentes techniques d'interaction tactiles qui sont donc rarement complètement directes et ont recours à diverses indirections pour résoudre les problèmes inhérents au concept de base. Se pose alors la question de qualifier cette indirection : « à quel point le curseur est-il physiquement et sémantiquement séparé du doigt ? » Nous avons reporté les travaux abordés dans cet état de l'art dans la figure 1.21 afin de construire un espace de solution. Nous y avons déjà placé nos travaux qui seront abordés dans les chapitres suivants, que nous avons entourés d'un halo vert.

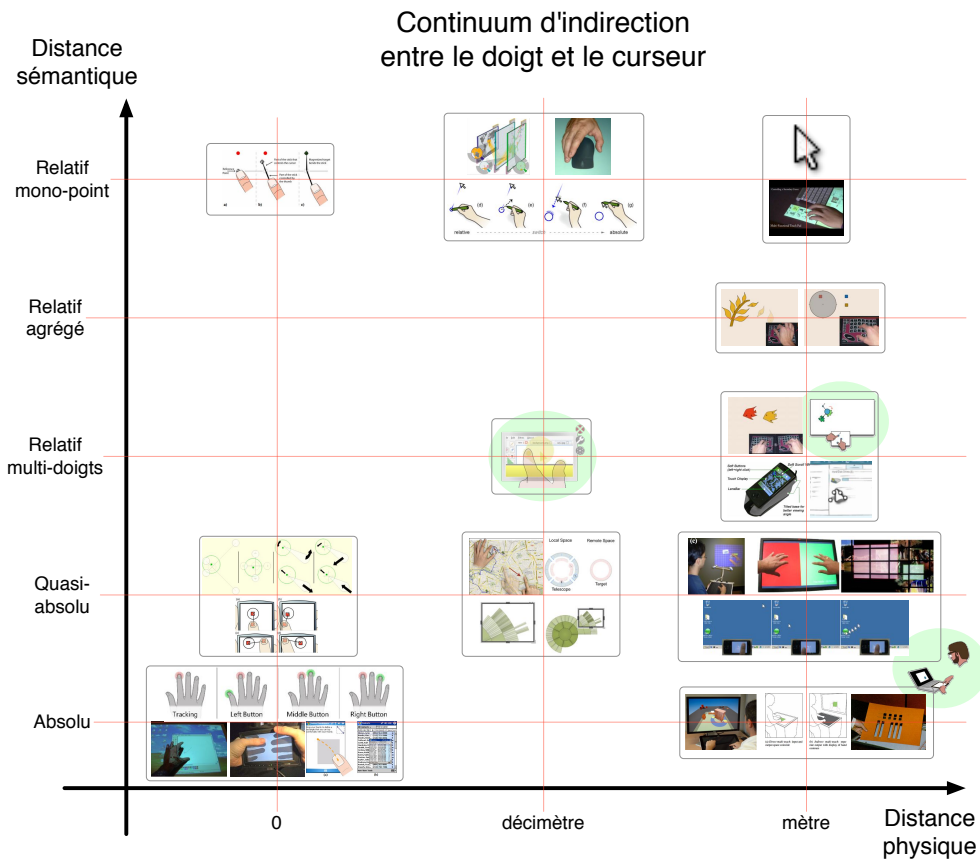


FIGURE 1.21 – Critères d'interaction indirecte.

En abscisse, nous distinguons les travaux selon la distance physique entre le lieu du contact et l'endroit où les objets sont manipulés. Ainsi, une première catégorie place le curseur juste à côté du doigt, le curseur n'ayant d'existence que pour la durée du contact. Ces travaux s'intéressent plus particulièrement aux problèmes de précision et d'état de survol. En augmentant un peu la séparation, nous trouvons les techniques introduisant un espace d'entrée distinct des objets d'intérêt, mais co-localisé avec l'affichage. Il s'agit ici d'autoriser un accès distant occasionnel, d'améliorer la précision ou de modifier le niveau de zoom. Enfin, les derniers travaux utilisent un périphérique d'entrée séparé de l'affichage. Ce peut être une surface tactile, une souris ou la combinaison des deux. L'objectif est alors principalement l'accès distant aux données tout en réintroduisant des degrés de libertés supplémentaires pour enrichir l'interaction.

En ordonnée, nous classons le lien sémantique entre le doigt et le curseur. Il peut s'agir d'une équivalence parfaite, comme c'est le cas de l'interaction absolue : la position et l'existence des curseurs sont identiques à celles des doigts. Les curseurs peuvent être placés dans un sous-espace en interaction quasi-absolue. Le concept supplémentaire d'espace de travail est introduit pour résoudre des problèmes de précision ou de facteurs d'échelles, mais la position des curseurs reste directement liée à la position des doigts. Ensuite nous entrons dans le domaine des curseurs relatifs. Ceux-ci peuvent être des espaces de travail : ils sont alors nécessaires au placement des doigts dans un sous-espace dont la position est définie de manière relative. Mais dans le même temps, le curseur représente, en l'absence de doigt, le lieu de survol et de sélection potentielle. Les curseurs relatifs peuvent également n'être qu'un unique point d'interaction, les gestes multi-doigts ne servant qu'à assurer le contrôle du curseur et sa sélection.

Cet espace de solution présente des zones plus denses que d'autres, signes qu'elles ont été davantage explorées. Ainsi lorsque l'utilisateur est co-localisé avec les objets qu'il manipule, l'interaction absolue s'impose. En effet il est intéressant d'utiliser la position absolue de la main comme indication approximative de la position de l'interaction et d'améliorer la précision au moyen d'un artefact graphique placé à proximité. Lorsque l'on s'éloigne des objets virtuels en prenant en main un périphérique tactile, les types d'interaction sont plus variées. La distance physique existante induit une plus grande liberté de placements et de contrôle des curseurs. Réussir à tirer parti des avantages de l'interaction multi-doigts et de l'interaction absolue pour maintenir une manipulation directe et un fort sentiment d'engagement devient alors un réel défi.

1.6 Conclusion

Dans ce chapitre, nous avons défini une classification des périphériques d'entrée. Plus particulièrement, nous avons qualifié les surfaces tactiles de périphériques isotoniques absolus, c'est-à-dire qu'elles mesurent des positions et des déplacements dans un référentiel constant et persistant. Après avoir détaillé le

fonctionnement de la plupart des périphériques tactiles, nous avons avancé et expliqué les qualités et avantages de l'interaction absolue et évalué la précision de l'utilisateur pour divers cas d'usage.

Néanmoins, ce type d'interaction soulève de multiples problèmes auxquels de nombreux travaux ont essayé de répondre en suivant des approches variées mais ayant pour caractéristique commune d'introduire une indirection entre le doigt et l'objet d'intérêt.

Nous placerons dans un premier temps nos travaux dans le cadre de l'interaction absolue indirecte au travers d'une étude systématique. Puis nous proposerons une technique similaire aux travaux de curseurs relatifs multi-doigts tout en essayant de supporter l'interaction multi-points, à plusieurs endroits. Enfin, nous explorerons l'interaction co-localisée en combinant les accès directs et indirects, relatifs et absolus.

Chapitre 2

Impacts du facteur de forme et des conditions d'entrée sur les tâches de pointage tactile indirect absolu

« [...] monkey tool use involves the incorporation of tools into the body schema, literally as extensions of the body »

—Dietrich Stout in [3] (Archéologue spécialisé dans l'utilisation des outils à l'*University College* de Londres)

Sommaire

2.1	Introduction	38
2.2	Expérience 1 : taille du périphérique d'entrée et conditions d'utilisation	39
2.2.1	Matériel	40
2.2.2	Tâche et procédure	40
2.2.3	Participants	41
2.2.4	Plan expérimental	42
2.3	Résultats	43
2.3.1	Taux de succès et nombre de tentatives ratées	43
2.3.2	Erreur de visée	44
2.3.3	Taille minimale de cible dans l'espace moteur	47
2.3.4	Distribution spatiale des contacts à la première tentative	47
2.4	Expérience 2 : effets d'échelle et rapport de cadre	50
2.4.1	Matériel, tâche et procédure	50
2.4.2	Participants	50
2.4.3	Plan expérimental	50
2.5	Résultats	52
2.5.1	Taux de succès et nombre de tentatives ratées	52
2.5.2	Erreur de visée	53
2.5.3	Taille minimale de cible dans l'espace moteur	55

2.5.4	Distribution spatiale des contacts à la première tentative	56
2.6	Discussion	57
2.7	Conclusion	60

2.1 Introduction

L'état de l'art présente une grande variété d'interactions tactiles, basées sur des mises en correspondance absolues ou relatives. Les mises en correspondance relatives s'appuient sur des fonctions de transfert non-linéaires pour apporter vitesse et précision. Les mises en correspondance absolues co-localisées fournissent un fort sentiment de contrôle direct, sans intermédiaire [45] car elles amènent l'utilisateur à toucher le contenu mais elles peuvent causer des problèmes de précision et d'occultation et fatiguer l'utilisateur qui doit parcourir la totalité de la surface d'entrée.

L'interaction multi-points relative indirecte résout ces problèmes mais la manipulation multi-points peut requérir de contrôler plusieurs curseurs en même temps. Malgré plusieurs tentatives pour contrôler plusieurs curseurs relatifs [65, 28, 11] cela reste une tâche difficile. Les mises en correspondance absolues indirectes supportent plus facilement la manipulation multi-points à distance, plus particulièrement quand un retour permet une pré-visualisation du contact. Les tablettes graphiques commencent par détecter le stylet avant qu'il n'entre en contact avec la surface, autorisant l'utilisateur à se positionner très précisément avant de commencer l'interaction. Néanmoins la plupart des périphériques tactiles ne fournissent pas cette information.

Une mise en correspondance absolue indirecte a été proposée par McCallum et Irani pour le positionnement grossier d'un curseur sur un grand écran en utilisant un téléphone portable tactile [62]. Leur technique ARC-Pad combine des fonctionnalités absolues et relatives : l'utilisateur réalise un *tap* sur la surface tactile pour téléporter le curseur à la position absolue projetée à l'écran, les gestes de *glissés* contrôlent le curseur de manière relative pour ajuster la sélection. Cette technique permet de réduire les gestes de débrayage de moitié par rapport à l'utilisation d'une souris. Ce travail suggère ainsi un bon niveau de performance atteignable en positionnement absolu discret, mais ces performances peuvent fortement dépendre de la taille d'écran choisi et du périphérique d'entrée.

Afin d'éviter les problèmes de précision dus à une mise en correspondance d'un petit espace d'entrée vers un large écran, nous avons pu voir précédemment dans la section 1.5.3 des travaux proposant d'interagir dans des sous-parties de l'espace d'affichage. Malik *et al.* ont adopté ce type d'approche en contrôlant un écran de 500×180 cm avec un système multi-points de 60×20 cm [59]. L'espace de travail contrôlé par la main non dominante a été dimensionné « de telle sorte que chaque pixel à l'écran peut être atteint en combinant le positionnement

grossier et fin »¹². Malgré cela, certains utilisateurs ont jugé que la précision fournie par défaut était trop grossière. Encore une fois, les tailles d'entrée, d'affichage et d'espace de travail ont été déterminés empiriquement et leur influence n'a pas été évaluée. On retrouve la même indétermination dans le dimensionnement des espaces de travail entourant les curseurs multi-doigts [65, 11] ou dans les travaux de pointage sur le corps [57, 35, 36].

Nous avons vu dans la section 1.4.4 que les impacts de divers facteurs ont été étudiés : les performances brutes de sélection co-localisée absolue, l'orientation entre les espaces d'entrée et d'affichage, leur position relative, la capacité à segmenter une partie du corps en zones d'interaction et l'importance de retours kinesthésiques et visuels lors d'une tâche d'exploration. Malgré tous les enseignements apportés par ces études à propos des tâches de pointage, positionnement continu ou d'exploration avec ou sans état de survol, les caractéristiques et les dimensionnements des surfaces d'entrée et d'affichage restent arbitrairement fixées et sans explications. Une approche systématique pourrait permettre de quantifier le niveau de précision accessible pour les tâches de positionnement discret absolu indirect.

Nous allons reporter dans ce chapitre deux expériences ayant pour objectif d'évaluer les effets du facteur de forme des espaces d'entrée et d'affichage et des conditions d'utilisation sur les performances des utilisateurs dans une tâche de pointage discrète indirecte absolue. Nous allons étudier les impacts de la taille du périphérique, la taille et le rapport de cadre de l'affichage ainsi que la taille et la position des cibles. Les conditions d'utilisation désignent la possibilité de tenir le périphérique et la possibilité de le regarder pendant l'interaction. Forts du résultat de ces expériences, nous formulerons des recommandations pour concevoir des interactions basées sur le pointage absolu indirect.

2.2 Expérience 1 : influence de la taille du périphérique d'entrée et des conditions d'utilisation

Le but de cette expérience était d'évaluer comment la taille du périphérique d'entrée influence la taille de la cible minimale qu'un utilisateur peut sélectionner au premier essai. Les hypothèses sont que :

- malgré l'absence de retour d'informations visuelles sur le périphérique d'entrée, la vue de celui-ci aide à positionner l'index à l'endroit voulu. Le participant voit l'espace d'entrée et l'espace d'affichage et peut comparer ces deux espaces pour faciliter la projection de la cible d'un repère à l'autre.
- comme dans le scénario d'exploration aveugle de Gustafson [36], la main non dominante peut agir comme une référence kinesthésique quand l'utilisateur ne peut pas voir la surface d'entrée, et aide au positionnement de l'index.

12. La citation originale est : « such that every pixel on the display can be reached using a combination of coarse and fine positioning ».

2.2.1 Matériel

Nous avons utilisé deux périphériques d'entrée différents. Le premier était un iPad 1 présentant une surface d'entrée de 196×147 mm. Le second était un iPod Touch 3 dont la surface d'entrée a été réduite à une zone rectangulaire de 66×50 mm au moyen d'un cache en carton épais (figure 2.1, à gauche) afin de présenter le même rapport de cadre (4:3) que l'iPad. Les écrans de ces périphériques affichaient un rectangle blanc tout au long de l'expérience, délimitant ainsi clairement la surface d'entrée des bords du périphérique.

Selon les conditions d'utilisation, des œillères artisanales étaient utilisées pour empêcher le participant de voir le périphérique d'entrée et ses mains mais sans gêner la vue de l'écran (figure 2.1, à droite).

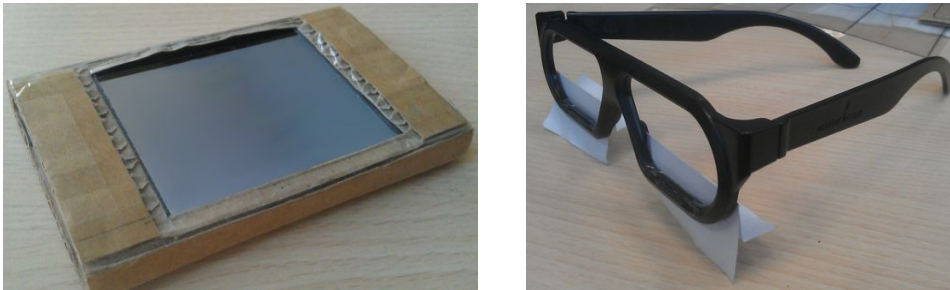


FIGURE 2.1 – À gauche, le cache en carton épais recouvrant l'iPod Touch empêche la détection des événements en dehors de la zone visible. À droite, les œillères ont été fabriquées à partir de lunettes en plastique auxquelles sont scotchés des caches en papier cachant la moitié basse du champ de vision.

Nous avons utilisé un ordinateur portable MacBook Pro (OS X 10.7, 2,4 GHz, 4 Go de RAM) et l'écran LCD intégré de 17,2" de diagonale (367×229 mm, 1680×1050 pixels). L'ordinateur portable était installé sur un bureau en face du participant, l'écran étant incliné de 30° en arrière et distant d'environ 60 cm des yeux du participant. Le périphérique d'entrée était placé par dessus le pavé tactile et au bord de l'ordinateur portable et la distance des yeux à l'écran était d'environ 50 cm. La position du périphérique permettait de le saisir avec la main non dominante sans le déplacer ni le soulever.

Le protocole d'expérimentation était codé en C++ en utilisant la librairie Qt. Les événements tactiles étaient transmis au moyen du protocole TUIO en mode UDP vers l'ordinateur portable. Les deux périphériques d'entrée et l'ordinateur communiquaient par un réseau WIFI ad-hoc.

2.2.2 Tâche et procédure

Nous avons utilisé une tâche de pointage discret en deux dimensions. Chaque essai débutait avec l'apparition d'une nouvelle cible à l'écran et terminait par la sélection de cette cible, en touchant la surface d'entrée à la position correspondante et en soulevant le doigt, ou après 5 tentatives de sélection

infructueuses. Chaque fois que le participant manquait une cible, c'est à dire que le centre du doigt était en dehors de la cible, une erreur était enregistrée.

L'écran de l'ordinateur affichait un rectangle noir sur toute la surface et un rectangle blanc représentant l'espace de travail au centre, ce dernier était de même dimension physique que la surface d'entrée qui affiche également un rectangle blanc. A l'intérieur de cet espace de travail, la cible à sélectionner était représentée par un disque bleu avec une croix noire au centre (figure 2.2, a). La consigne donnée aux participants était de pointer la position correspondant à cette croix sur le périphérique d'entrée. Lorsque le participant touche le périphérique d'entrée un disque bleu transparent avec une croix blanche en son centre sont affichés dans l'espace de travail à la position correspondante (b). Cette marque restait visible tant que le contact avec le périphérique tactile était maintenu, mais ne pouvait être déplacé pour corriger la tentative. En cas de tentative ratée, le disque entourant la cible courante se colorait alors en rouge sans autre moyen de correction que de retirer le doigt (c) et tenter une nouvelle fois. La cible correctement sélectionnée virait au vert pour informer de la fin de l'essai (d) en attendant la fin du contact pour passer à la cible suivante (e).

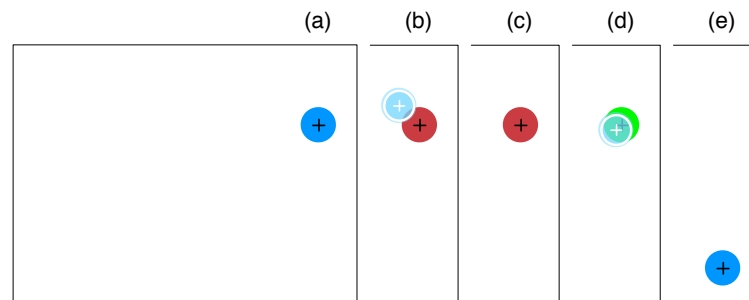


FIGURE 2.2 – Séquence possible de vues lors d'un même essai : (a) l'essai commence avec la cible à sélectionner ; (b) un contact est détecté et affiché, mais la tentative est ratée ; (c) le contact est terminé, l'essai continue avec une nouvelle tentative ; (d) un nouveau contact est détecté et la sélection est valide ; (e) le contact est à nouveau terminé, l'essai se termine, le suivant commence avec l'affichage de la nouvelle cible. Cet essai sera marqué comme erroné avec 1 tentative ratée.

Après chaque bloc d'essais, les participants étaient invités à prendre une courte pause avant de continuer en appuyant sur un bouton. Les participants étaient encouragés à sélectionner les cibles assez rapidement en visant au maximum le centre tout en conservant une attitude et des gestes détendus et relaxés.

2.2.3 Participants

12 volontaires non payés ont participé à cette expérience : 4 femmes et 8 hommes, tous droitiers. L'âge moyen des participants était de 25,5 ans (écart-type de 1,9), ils avaient déclaré utiliser souvent un pavé tactile et un téléphone tactile, mais quasiment aucun (sauf un) n'avait déjà utilisé de tablette graphique.

2.2.4 Plan expérimental

Un plan expérimental intra-sujets et une analyse à mesures répétées ont été utilisés. Les variables indépendantes étaient la taille du périphérique d'entrée (TAILLE D'ENTRÉE), les conditions d'utilisation (CONDITION D'UTILISATION), la position de la cible (POSITION DE LA CIBLE) et sa taille (TAILLE DE LA CIBLE). La variable TAILLE D'ENTRÉE était évaluée avec 2 niveaux : PETITE pour l'iPod Touch (3,26" de diagonale), et GRANDE pour l'iPad (9,65" de diagonale). La variable POSITION DE LA CIBLE avait 9 positions différentes correspondantes à un sous-ensemble des intersections d'une grille imaginaire 5×5 centrée sur l'écran et numérotée en colonne à partir du coin supérieur gauche. Les positions (1, 3, 9, 10, 11, 13, 19, 22, 25 sur la figure 2.3) ont été retenues de manière à couvrir l'ensemble de l'écran et des situations. La sélection d'un sous-ensemble parmi le 25 positions possibles a permis de raccourcir la durée de l'expérience. La TAILLE DE LA CIBLE présentait 3 niveaux autour des valeurs de tailles minimales trouvées dans la littérature [38, 89, 94, 43, 35, 78, 26]. Les tailles étaient les suivantes : $W_S = 10$ mm, $W_M = 20$ mm et $W_L = 40$ mm.

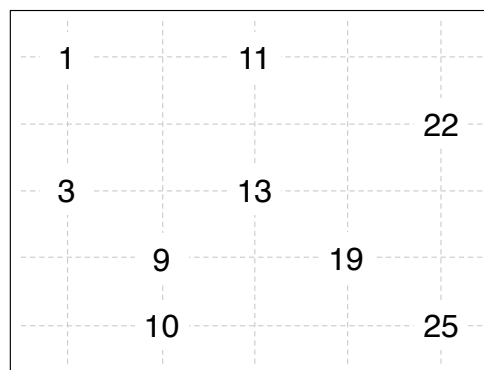


FIGURE 2.3 – Positions retenues pour la variable POSITION DE LA CIBLE.

Enfin la variable CONDITION D'UTILISATION comprenait 3 modalités : 1MAIN, 1MAINOEILLÈRES et 2MAINSOEILLÈRES (Figure 2.4). Dans le cas 1MAIN, les participants pouvaient regarder la surface d'entrée quand ils le désiraient mais n'étaient pas obligés. Au contraire dans les deux autres conditions (1MAINOEILLÈRES et 2MAINSOEILLÈRES), le port des œillères (figure 2.1, à droite) privait le participant de la perception de ses mains et du périphérique d'entrée.

Dans les conditions 1MAIN et 1MAINOEILLÈRES, la main non dominante devait rester sur les genoux ou l'accoudoir, le participant avait interdiction de la poser près du périphérique d'entrée. Dans la condition bimanuelle (2MAINSOEILLÈRES), l'utilisateur devait tenir le périphérique avec sa main non dominante tout en laissant le périphérique en place. Gustafson *et al.* ont montré que la possibilité de voir sa main agir sur la surface fournit une aide plus importante que les aides tactiles ou kinesthésiques [36]. Pour cette raison nous n'avons pas jugé utile d'ajouter une quatrième modalité où le participant agit avec ses deux mains sans œillères.

En résumé, l'écran de l'ordinateur portable est toujours visible et le participant

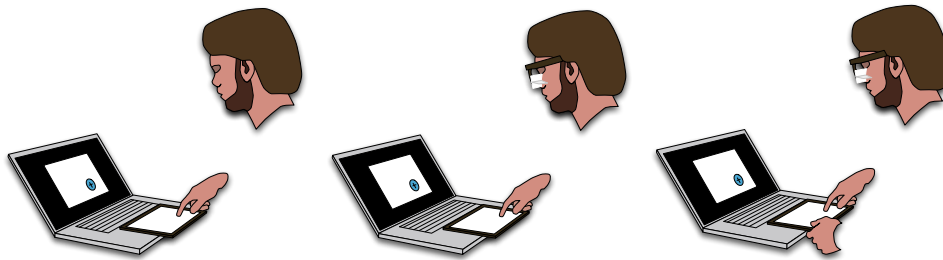


FIGURE 2.4 – Les trois modalités de la variable CONDITION D’UTILISATION : 1MAIN, 1MAINOEILLÈRES et 2MAINSOEILLÈRES.

ne touche la surface tactile qu’avec l’index de sa main dominante. La main non dominante n’intervient que dans la condition 2MAINSOEILLÈRES pour tenir le périphérique sans le déplacer, et il ne peut voir la surface d’entrée que dans le cas 1MAIN.

L’ordre de présentation des variables CONDITION D’UTILISATION et TAILLE D’ENTRÉE pour les participants était contre-balancé suivant un carré latin balancé. La variable TAILLE DE LA CIBLE était présentée en ordre décroissant afin d’augmenter progressivement la difficulté. Chaque BLOC était constitué de 3 répétitions des 9 POSITIONS DES CIBLES mélangées de façon pseudo-aléatoire. L’expérience durait en moyenne 50 minutes pour chaque participant.

Le plan expérimental était donc le suivant :

12 participants ×
 3 CONDITION D’UTILISATION ×
 2 TAILLE D’ENTRÉE ×
 3 BLOC ×
 3 TAILLE DE LA CIBLE ×
 9 POSITION DE LA CIBLE ×
 3 Répétitions
 = 17 496 essais au total.

Pour chaque essai, nous avons enregistré la position de la première tentative et le nombre de tentatives ratées.

2.3 Résultats

Les variables dépendantes étaient le taux de succès, le nombre de tentatives ratées et l’erreur de visée.

2.3.1 Taux de succès et nombre de tentatives ratées

Les cibles qui n’ont pas été sélectionnées avec succès à la première tentative étaient marquées comme erronées. Dans la suite, le *taux de succès* est le pourcentage de cibles sélectionnées à la première tentative. Le *nombre de tentatives*

ratées est le nombre moyen de fois où les participants avaient raté une cible (5 fois au maximum). Par exemple, pour la GRANDE taille de périphérique illustré dans la figure 2.5, le taux de succès est directement lisible, il est de 54% et le nombre moyen de tentatives ratées est de 0,66 ($0,54 \times 0 + 0,33 \times 1 + 0,08 \times 2 + 0,03 \times 3 + 0,01 \times 4 + 0,01 \times 5$). En moyenne les participants avaient dû effectuer 1,66 tentatives par essai pour sélectionner la cible.

Un test de Friedman a montré un effet significatif de la variable BLOC à la fois sur le taux de succès ($\chi^2=13,2$, $df=2$, $p=0,001$) et sur le nombre de tentatives ratées ($\chi^2=10,2$, $df=2$, $p<0,01$). Les comparaisons deux à deux¹³ ont montré une différence significative entre le premier bloc et les deux suivants pour le taux de succès ($p<0,05$), et entre le premier et le second bloc pour le nombre de tentatives ratées ($p<0,05$), montrant ainsi un effet d'apprentissage. Ainsi, le premier bloc a été supprimé des analyses suivantes. Nous avons également ignoré les essais déviant de la position moyenne de plus de 3 écart-types, car non représentatifs dans le cadre d'une expérience, cela représentait 0,9% du total des essais.

Le test de Friedman a montré les effets significatifs des facteurs CONDITION D'UTILISATION ($\chi^2=18,7$, $df=2$, $p<0,001$), TAILLE D'ENTRÉE ($\chi^2=12$, $df=1$, $p<0,001$), TAILLE DE LA CIBLE ($\chi^2=24$, $df=2$, $p<0,001$) et POSITION DE LA CIBLE ($\chi^2=30,4$, $df=8$, $p<0,001$) sur le nombre de tentatives ratées. Les comparaisons deux à deux ont révélé que la condition 1MAIN est significativement différente des deux autres conditions d'entrée ($p<0,01$, 1MAIN : 0,26, 1MAINOEILLÈRES : 0,56, 2MAINSOEILLÈRES : 0,51). Le nombre de tentatives ratées chute de 0,66 pour la GRANDE surface d'entrée à 0,23 pour la PETITE surface d'entrée. Les comparaisons deux à deux ont montré une différence significative ($p<0,001$) entre les tailles de cibles Ws (0,93) et WL (0,07). Par contre, bien que le facteur POSITION DE LA CIBLE a un effet, aucune différence significative n'a pu être observée.

Le nombre de tentatives ratées pour chaque variable est illustré sur la figure 2.5. En général, les participants avaient plus de succès quand ils pouvaient voir le périphérique d'entrée, utilisaient le plus petit périphérique ou sélectionnaient des cibles plus grandes.

2.3.2 Erreur de visée

L'erreur de visée est calculée pour tous les essais, qu'il soit erronés ou pas, il s'agit de la distance (en mm) entre la position de la première tentative et le centre de la cible. À une faible erreur de visée correspond une haute précision. Un test de Friedman a, ici aussi, montré un effet d'apprentissage significatif entre les BLOCS ($\chi^2=15,2$, $df=2$, $p<0,001$) sur l'erreur de visée. Les comparaisons deux à deux ont encore montré que le premier bloc est significativement différent des deux autres ($p<0,01$) avec pour conséquence de retirer ce bloc des analyses suivantes.

L'erreur de visée ne suit pas une distribution normale, nous avons donc utilisé une méthode de classement des valeurs « Aligned Rank Transform » [101] et

13. Toutes les analyses post-hoc de ce chapitre ont été ajustées par la méthode Bonferroni.

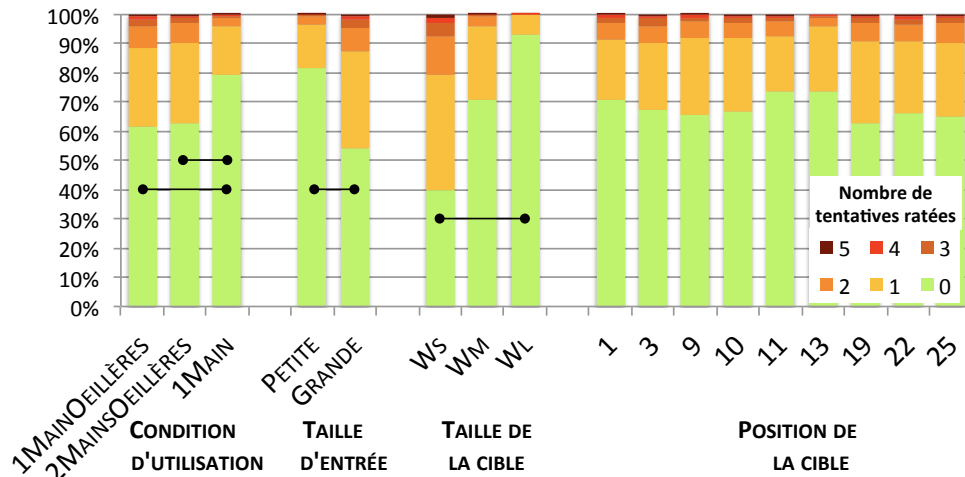


FIGURE 2.5 – Nombre de tentatives ratées en fonction des variables CONDITION D'UTILISATION, TAILLE D'ENTRÉE, TAILLE DE LA CIBLE et POSITION DE LA CIBLE (de gauche à droite). Les connections entre les barres indiquent une différence statistique significative.

poursuivi l'analyse sur ces rangs par des analyses de variance (ANOVA) pour détecter les effets et interactions significatifs.

Une ANOVA a montré l'effet significatif des CONDITION D'UTILISATION ($F_{2,22}=72,1$, $p<0,001$) sur l'erreur de visée, les comparaisons deux à deux ont révélé des différences significatives entre tous les niveaux (1MAINOEILLÈRES : 9,8 mm, 2MAINSOEILLÈRES : 9,4 mm, 1MAIN : 6,2 mm ; $p<0,001$). La possibilité de voir la surface d'entrée, même si celle-ci n'affiche rien, réduit clairement l'erreur de visée. Globalement, l'introduction de la deuxième main (2MAINSOEILLÈRES) diminue l'erreur de visée de 4% par rapport à 1MAINOEILLÈRES, alors que la condition 1MAIN améliore la précision de 37%.

L'ANOVA a mis en évidence l'effet significatif de la TAILLE D'ENTRÉE ($F_{1,11}=489$, $p<0,001$) et l'interaction significative entre les deux facteurs CONDITION D'UTILISATION \times TAILLE D'ENTRÉE ($F_{2,22}=28,4$, $p<0,001$) sur l'erreur de visée. L'erreur de visée moyenne pour les tailles PETITE et GRANDE sont respectivement de 5,9 mm et 11 mm. Nous supposons que l'erreur de visée moindre obtenue sur la plus petite taille peut être expliquée par des déplacements moins importants dans un espace plus restreint : les participants peuvent effectuer de petits gestes plus précisément que de grands gestes. L'interaction trouvée entre ces deux facteurs montre que pour la PETITE taille d'entrée, il n'y a pas de différence significative entre les conditions d'utilisation 1MAINOEILLÈRES et 2MAINSOEILLÈRES contrairement à ce qui se passe pour la plus grande taille d'entrée (figure 2.6).

Nous avons trouvé un effet significatif de la TAILLE DE LA CIBLE ($F_{2,22}=12,7$, $p<0,001$). Les comparaisons deux à deux ont indiqué que l'erreur de visée est significativement plus haute pour les plus grandes cibles WL (9,1 mm) que pour les autres WM (8,2 mm) et WS (8,0 mm). Les participants ont pu être moins

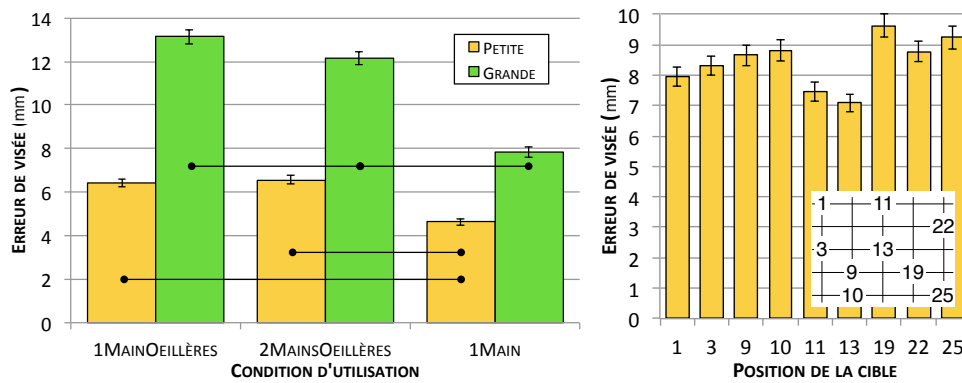


FIGURE 2.6 – Erreur de visée en fonction des variables TAILLE D'ENTRÉE et CONDITION D'UTILISATION (à gauche), et POSITION DE LA CIBLE (à droite). Les connections entre les barres indiquent une différence statistique significative.

appliqués pour des cibles plus grandes, parce qu'elles ont semblé très facile à atteindre. L'ANOVA a également révélé une interaction entre les facteurs TAILLE DE LA CIBLE \times TAILLE D'ENTRÉE ($F_{2,22}=7,0$, $p=0,005$). Les comparaisons deux à deux ont montré que l'erreur de visée a arrêté de diminuer entre les tailles de cibles WM et Ws sur la GRANDE taille d'entrée alors qu'il y a une différence significative entre ces tailles de cibles sur le plus petit périphérique d'entrée (figure 2.7). Aucune interaction significative n'a pu être observée entre les facteurs TAILLE DE LA CIBLE \times CONDITION D'UTILISATION.

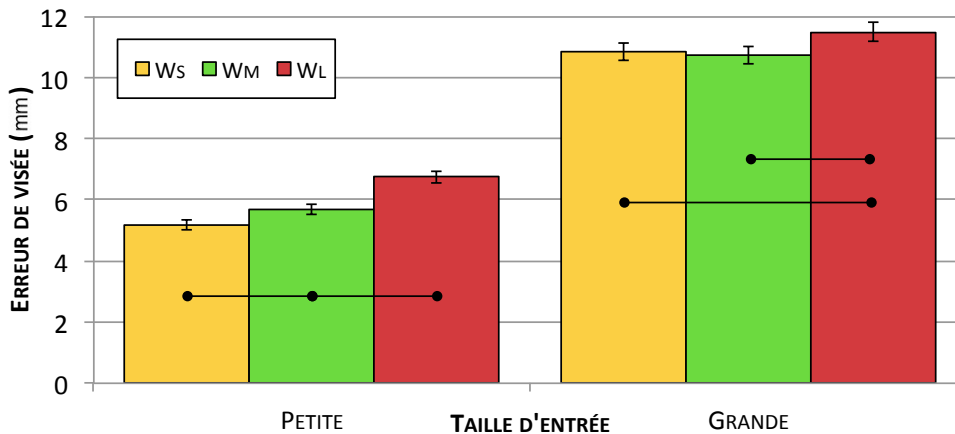


FIGURE 2.7 – Erreur de visée en fonction des variables TAILLE D'ENTRÉE et TAILLE DE LA CIBLE. Les connections entre les barres indiquent une différence statistique significative.

Le facteur POSITION DE LA CIBLE a également un effet significatif sur l'erreur de visée ($F_{8,88}=12,25$, $p<0,001$). L'ANOVA a mis en lumière l'interaction significative des facteurs POSITION DE LA CIBLE \times CONDITION D'UTILISATION ($F_{16,176}=5,39$, $p<0,001$). L'erreur de visée est minimale pour les cibles situées au centre, suivie par les cibles du quart supérieur gauche, puis viennent les cibles en diagonale 9, 19 et 22. Les participants laissaient leur index centré au-dessus du périphérique d'entrée pendant l'expérience, ce qui les obligeait

à plier le doigt pour atteindre les cibles du quart inférieur droit : ces cibles (19 et 25) présentent la plus haute erreur de visée. Les comparaisons deux à deux concernant l'interaction POSITION DE LA CIBLE \times CONDITION D'UTILISATION nous apprennent que l'erreur de visée est significativement plus faible pour la cible 25 quand le participant peut regarder la surface d'entrée que dans les conditions aveugles. À l'inverse, les cibles 9 et 19 présentent des erreurs de visée plus importantes dans la condition visuelle (1MAIN), probablement car ces cibles ne sont ni positionnées proches d'une bordure ou d'un coin, ni au centre, elles ne bénéficient ainsi pas d'indice visuel pour aider à estimer leur position.

2.3.3 Taille minimale de cible dans l'espace moteur

En se basant sur nos données expérimentales, nous pouvons estimer la taille minimale d'une cible dans l'espace moteur afin qu'un utilisateur voulant la sélectionner ait 95% de succès à la première tentative. Nous travaillons maintenant sur la dispersion de la position des premières tentatives autour des centres des cibles. Nous avons constaté une déviation moyenne globale de 0,4 mm vers la droite et 0,8 mm en dessus des centres.

Une analyse de variance multivariée (MANOVA) sur les coordonnées de chaque première tentative a permis de prouver l'effet significatif de la variable CONDITION D'UTILISATION ($F_{4,8}=5,8$, $p=0,019$) et de la TAILLE D'ENTRÉE ($F_{2,10}=9,1$, $p<0,01$) sur la taille minimale de cible. Ces tailles minimales de cibles dans l'espace moteur sont reportées dans le tableau 2.1. Ces résultats tendent à prouver que la vue du périphérique d'entrée amène une plus grande précision que le fait de le tenir avec la main non dominante. Cela confirme aussi que les participants sont presque deux fois plus précis (82%) sur la plus petite surface d'entrée.

CONDITION D'UTILISATION		1MAIN	2MAINS	1MAIN
		ŒILLÈRES	ŒILLÈRES	
TAILLE D'ENTRÉE	PETITE	22,3	23,2	16,8
	GRANDE	45,2	41,3	27,8

TABLE 2.1 – Taille minimale de cible (en mm) assurant une sélection sans erreur dans 95% des cas en fonction des variables CONDITION D'UTILISATION et TAILLE D'ENTRÉE.

2.3.4 Distribution spatiale des contacts à la première tentative

La figure 2.8 présente la distribution des premières tentatives pour les facteurs TAILLE D'ENTRÉE et CONDITION D'UTILISATION. Les ellipses illustrent la distribution normale bivariée de ces premiers contacts pour chaque position de cible, les positions réelles des centres des cibles sont représentées par les croix.

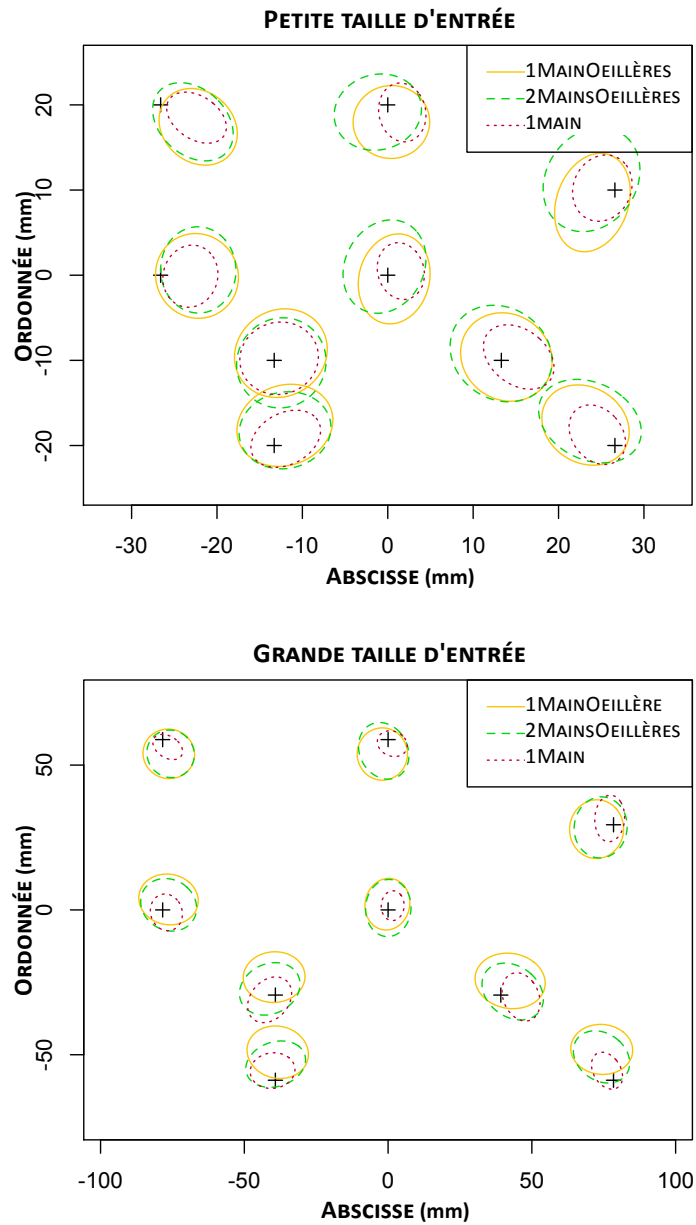


FIGURE 2.8 – Distribution de la position des premières tentatives en fonction des facteurs TAILLE D'ENTRÉE et CONDITION D'UTILISATION. Les croix représentent le centre des cibles à sélectionner. Les ellipses contiennent 68% des contacts pour chaque position de cible.

Toutes conditions confondues, les contacts semblent être attirés par le centre de la surface. Alors que la présentation des cibles était aléatoire, la stratégie dominante des participants consistait à garder un placement de la main au centre de la surface d'entrée, cela peut ainsi expliquer la déviation vers le centre. De plus, une seconde dérive cumulée à l'attraction du centre peut être observée dans la condition visuelle (1MAIN). La distribution des premiers contacts est décalée en bas à droite par rapport aux conditions aveugles. Le centre des ellipses pour 1MAIN se retrouve généralement dévié par rapport aux autres centres.

Vogel, Baudisch et Holz [89, 43] ont montré que la plupart des utilisateurs pensent que le point d'entrée en interaction tactile se trouve au centre de l'ongle ou à son extrémité supérieure : ils pointent avec le bout de leur doigt. Alors qu'en réalité, la machine utilise le centre de la surface en contact, qui est décalé au sud-est par rapport au bout du doigt pour la main droite. Ce décalage entre ce que l'utilisateur pense faire et ce que la machine interprète introduit une mauvaise compréhension du modèle d'entrée perçu (*perceived input point model* dans la littérature). Dans notre expérience, quand les participants étaient autorisés à regarder le périphérique d'entrée, nous pensons qu'ils projetaient mentalement sur la surface et essayaient de mettre le doigt à la position correspondante. Ce décalage supplémentaire que nous observons peut être causé par cette différence de perception. Comme tous nos participants étaient droitiers, leur index pointait vers le coin supérieur gauche, et la déviation est dirigée dans la direction opposée, au sud-est. Les deux conditions aveugles (1MAINOEILLÈRES et 2MAINOEILLÈRES) ne sont d'ailleurs pas concernées par ce décalage : étant dans l'impossibilité de voir le doigt sur la surface, les participants ne pouvaient se référer qu'à la position du contact perçu à l'écran, ils construisaient leur modèle d'entrée en accord avec les informations que la machine interprète.

En rappel de cette première expérience, le point le plus important est que l'erreur de visée n'est pas constante selon la taille du périphérique d'entrée : elle est plus faible avec la plus petite surface. Il faut également retenir que la possibilité de voir le périphérique d'entrée améliore grandement la sélection, alors que l'introduction de la main non dominante comme référence n'aide pas significativement. Nous avons montré que la position des cibles importe : les cibles situées à proximité du centre et au-dessus de la main sont plus faciles à atteindre que les cibles positionnées sous la paume, et que les coins facilitent la sélection de cibles.

Dans cette expérience, nous avons volontairement gardé un rapport 1:1 entre la surface d'entrée et l'espace de travail visuel : les rectangles blancs présentés par chacun des espaces avaient les mêmes dimensions et même rapport de cadre. Les effets d'échelle et de déformation vont être examinés lors de l'expérience suivante.

2.4 Expérience 2 : effets d'échelle et rapport de cadre

L'objectif de cette seconde expérience était d'évaluer l'influence des dimensions et des différences de rapport de cadre de l'écran sur les tâches de pointage tactile indirect absolu. Nous émettons les hypothèses suivantes :

- l'échelle d'affichage, c'est-à-dire la taille des cibles relativement à la taille de l'espace de travail visuel, n'a pas d'impact sur les performances ;
- des rapports de cadre similaires entre la surface d'entrée et l'espace de travail visuel amènent les meilleures performances.

2.4.1 Matériel, tâche et procédure

Afin de pouvoir tester une large plage de tailles et de rapports de cadre d'espace de travail, nous avons remplacé l'écran de la première expérience par un écran plasma de 50" (1106×622 mm, 1365×768 pixels). Nous voulions également avoir une condition où la surface d'affichage est plus petite que la surface d'entrée, nous avons ainsi conservé le plus grand des deux périphériques d'entrée : l'iPad 1 (la taille GRANDE). Le protocole de l'expérience était, pour cette deuxième expérience, exécuté sur un PC (Windows 7, 2,7 GHz, 4Go de RAM). L'écran était fixé à plat contre un mur à une distance de 90 cm des participants. L'iPad était toujours posé à plat sur un bureau, et se trouvait à 60 cm des yeux des participants.

La tâche et les procédures restaient les mêmes que lors de la première expérience, à ceci près que les dimensions et rapports de cadre de l'espace de travail affiché à l'écran variaient en fonction des facteurs. Nous autorisons les participants à regarder la surface d'entrée comme dans la modalité 1MAIN, et ils ne pouvaient pas se servir de leur main non dominante. Les cibles étaient toujours présentées comme des disques à l'écran mais étaient déformées dans l'espace moteur en fonction du rapport de cadre en cours. Encore une fois, aucune autre information qu'un rectangle blanc n'était affichée sur le périphérique d'entrée.

2.4.2 Participants

12 volontaires non payés et qui n'avaient pas participé à la première expérience ont participé à celle-ci : 5 femmes et 7 hommes. L'âge moyen était de 31,8 ans (écart-type de 8,4) et tous étaient droitiers. Ils ont déclaré être familiers avec les pavés tactiles et téléphones tactiles et certains avaient déjà utilisé une tablette graphique.

2.4.3 Plan expérimental

Un plan expérimental intra-sujets et une analyse à mesures répétées ont été utilisés. Les variables indépendantes étaient la hauteur (HAUTEUR DE L'ESPACE

2.4. Expérience 2 : effets d'échelle et rapport de cadre

DE TRAVAIL) et le rapport de cadre (RAPPORT DE CADRE) de l'espace de travail visuel, la position (POSITION DE LA CIBLE) et la taille (TAILLE DE LA CIBLE) de la cible.

La variable HAUTEUR DE L'ESPACE DE TRAVAIL était évaluée sur 3 niveaux : une petite hauteur $H_{\times 0,5} = 74$ mm, ce qui correspond à la moitié de la hauteur de l'iPad ; une hauteur moyenne de référence $H_{\times 1} = 147$ mm, soit autant que l'iPad ; et une grande hauteur $H_{\times 2} = 294$ mm, le double de la hauteur de l'iPad.

Le RAPPORT DE CADRE présentait 3 valeurs : $R_{4:3} = 4:3$, $R_{16:9} = 16:9$ et $R_{32:10} = 32:10$, les tailles d'espace de travail correspondantes sont calculées dans le tableau 2.2. Un rapport de cadre de 32:10 correspond à l'espace virtuellement créé en accolant deux écrans d'ordinateur larges modernes de rapport 16:10. Le rapport de cadre 16:9 est le rapport typique des téléviseurs actuels. Bien que de moins en moins populaire, le rapport de cadre 4:3 est identique à celui de notre périphérique d'entrée et permettra de comparer les résultats avec l'expérience 1.

HAUTEUR DE L'ESPACE DE TRAVAIL		$H_{\times 0,5}$	$H_{\times 1}$	$H_{\times 2}$
RAPPORT DE CADRE	$R_{4:3}$	98×74	196×147	392×294
	$R_{16:9}$	131×74	261×147	523×294
	$R_{32:10}$	235×74	470×147	941×294

TABLE 2.2 – Dimensions (en mm) de l'espace de travail en fonction de la HAUTEUR DE L'ESPACE DE TRAVAIL et du RAPPORT DE CADRE.

La TAILLE DE LA CIBLE avait 2 modalités : une taille moyenne $W_M = 20$ mm et une grande taille $W_L = 40$ mm. L'expérience 1 n'avait pas montré de différence significative entre les tailles W_S et W_M sur le plus grand périphérique, nous avons donc retiré W_S afin de raccourcir la durée de l'expérience. De plus, dans les conditions (W_S , $R_{32:10}$, $H_{\times 2}$), la cible aurait eu une taille de 1,05 mm dans l'espace moteur, taille sûrement trop petite pour être sélectionnée. Le tableau 2.3 présente les tailles de cibles dans l'espace moteur correspondantes aux tailles visibles dans l'espace de travail visuel en fonction des différentes combinaisons du RAPPORT DE CADRE et de la HAUTEUR DE L'ESPACE DE TRAVAIL.

L'effet de la POSITION DE LA CIBLE a déjà été examiné pour 9 emplacements dans l'expérience 1, nous avons décidé de continuer avec moins de positions afin de raccourcir la durée de l'expérience. Nous avons retenu 4 positions aux résultats contrastés : la cible du centre présentant les meilleurs résultats (la cible 13), une cible du coin supérieur gauche ayant également de bons résultats (1), une cible assez difficile située sur la diagonale (22), et la cible la plus difficile placée sous la main (19).

L'ordre de présentation des conditions HAUTEUR DE L'ESPACE DE TRAVAIL et RAPPORT DE CADRE a été contre-balançé pour les participants avec un carré latin balancé. La TAILLE DE LA CIBLE était présentée en ordre décroissant. Chaque BLOC était composé de 3 répétitions des 4 POSITIONS DES CIBLES mélangées par

WM	HAUTEUR DE L'ESPACE DE TRAVAIL	$H_{\times 0,5}$	$H_{\times 1}$	$H_{\times 2}$
RAPPORT DE CADRE	R _{4:3}	20×20	10×10	5×5
	R _{16:9}	15×20	17,5×10	3,7×5
	R _{32:10}	8,3×20	4,2×10	2,1×5
WL	HAUTEUR DE L'ESPACE DE TRAVAIL	$H_{\times 0,5}$	$H_{\times 1}$	$H_{\times 2}$
RAPPORT DE CADRE	R _{4:3}	40×40	20×20	10×10
	R _{16:9}	30×40	15×20	7,5×10
	R _{32:10}	16,7×40	8,3×20	4,2×10

TABLE 2.3 – Tailles des cibles dans l'espace moteur (en mm) correspondantes aux tailles visibles dans l'espace de travail en fonction des facteurs RAPPORT DE CADRE et HAUTEUR DE L'ESPACE DE TRAVAIL. Les cellules surlignées sont celles pour lesquelles différentes conditions visuelles conduisent à la même taille de cible dans l'espace d'entrée.

un algorithme pseudo-aléatoire. L'expérience avait une durée moyenne de 20 minutes.

Nous avons le plan expérimental suivant :

12 participants ×
 3 HAUTEUR DE L'ESPACE DE TRAVAIL ×
 3 RAPPORT DE CADRE ×
 3 BLOC ×
 2 TAILLE DE LA CIBLE ×
 4 POSITION DE LA CIBLE ×
 3 Repetitions
 = 7 776 essais au total.

Pour chaque essai, nous avons enregistré la position de la première tentative et le nombre de tentatives ratées.

2.5 Résultats

Les variables dépendantes étaient le taux de succès, le nombre de tentatives ratées et l'erreur de visée, précédemment définis.

Nous avons écarté des analyses les essais dont la position de la première tentative déviait de la position moyenne de plus de 3 écart-types, car considérés comme aberrants. Cela représentait 0,8% des essais.

2.5.1 Taux de succès et nombre de tentatives ratées

Le nombre de tentatives ratées pour chaque facteur est illustré sur la figure 2.9. Un test de Friedman n'a pas permis de montrer une différence significative

entre les BLOCS ni pour le taux de succès ($\chi^2=4,04$, $df = 2$, ns) ni pour le nombre de tentatives ratées ($\chi^2=3,43$, $df=2$, ns). Le test de Friedman a permis de montrer l'effet significatif de chaque facteur sur le nombre de tentatives ratées : HAUTEUR DE L'ESPACE DE TRAVAIL ($\chi^2=24$, $df=2$, $p<0,001$), RAPPORT DE CADRE ($\chi^2=22,2$, $df=2$, $p<0,001$), TAILLE DE LA CIBLE ($\chi^2=12$, $df=1$, $p<0,001$) et POSITION DE LA CIBLE ($\chi^2=24,2$, $df=3$, $p<0,001$). Les comparaisons deux à deux ont révélé des différences significatives entre les hauteurs d'espace de travail $H_{\times 0,5}$ et $H_{\times 2}$ (0,10 et 1,04, $p<0,001$) et entre les rapports de cadre $R_{4:3}$ et $R_{32:10}$ (0,30 et 0,80, $p<0,001$). Pour la POSITION DE LA CIBLE, les comparaisons deux à deux ont révélé des différences significatives entre les positions 13 et 19 (0,32 et 0,66, $p<0,05$), et entre les positions 13 et 22 (0,32 et 0,53, $p<0,05$).

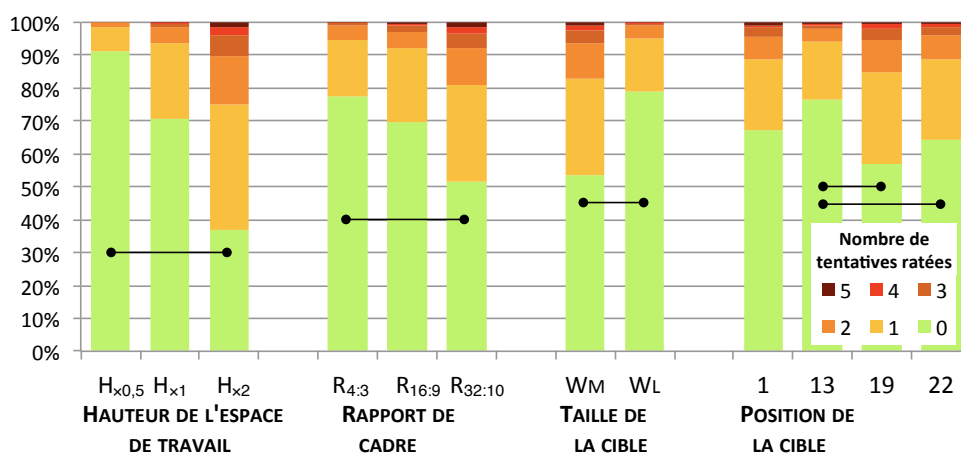


FIGURE 2.9 – Nombre de tentatives ratées en fonction des variables HAUTEUR DE L'ESPACE DE TRAVAIL, RAPPORT DE CADRE, TAILLE DE LA CIBLE et POSITION DE LA CIBLE (de gauche à droite). Les connections entre les barres indiquent une différence statistique significative.

2.5.2 Erreur de visée

Un test de Friedman a montré un effet significatif d'apprentissage au fil des BLOC ($\chi^2=8,17$, $df=2$, $p=0,017$) pour l'erreur de visée, mais les comparaisons deux à deux et l'ajustement de Bonferroni n'ont pas permis de trouver de différences significatives entre les blocs. L'ensemble des trois blocs sera donc conservé pour les analyses suivantes.

Une ANOVA a montré l'effet significatif des facteurs HAUTEUR DE L'ESPACE DE TRAVAIL ($F_{2,22}=7,06$, $p<0,01$) et RAPPORT DE CADRE ($F_{2,22}=11,72$, $p<0,001$) sur l'erreur de visée, mais aucune interaction n'a pu être observée entre ces facteurs. Les comparaisons deux à deux ont révélé que la hauteur d'espace de travail $H_{\times 0,5}$ présentait des résultats significativement différents des deux autres hauteurs ($H_{\times 0,5}$: 10,2 mm, $H_{\times 1}$: 8,9 mm et $H_{\times 2}$: 8,7 mm, $p<0,001$, figure 2.11). Des hauteurs d'espace de travail plus grandes impliquent, à cause de la projection, des cibles de tailles plus restreintes. L'erreur de visée réduite pour les hauteurs $H_{\times 1}$ et $H_{\times 2}$ a pu conduire les participants à faire plus attention lors de la sélection en réponse à un taux de succès plus faible (comme

précédemment vu dans la figure 2.9). Les comparaisons deux à deux ont mis en lumière des résultats significativement différents entre le rapport de cadre $R_{32:10}$ et les deux rapports de cadre $R_{4:3}$ et $R_{16:9}$ (9,8 mm, 8,9 mm et 9,0 mm, $p < 0,01$). L'erreur de visée augmente à mesure que le rapport de cadre devient plus large : l'étirement horizontal semble gêner l'estimation correcte de la position des cibles.

L'ANOVA a montré un effet significatif de la TAILLE DE LA CIBLE ($F_{1,11}=25,83$, $p < 0,001$, figure 2.11) sur l'erreur de visée. Comme dans l'expérience 1, des cibles plus petites ont amené une réduction de l'erreur de visée (WL : 9,6 mm, WM : 9,0 mm). Il semble, ici encore, que les participants étaient moins appliqués avec les cibles larges qui leur paraissaient plus faciles à acquérir. Une interaction significative a été trouvée pour les facteurs RAPPORT DE CADRE \times TAILLE DE LA CIBLE ($F_{2,22}=6,30$, $p < 0,01$) mais les comparaisons deux à deux ne permettent pas de localiser les différences. L'erreur de visée semble atteindre un plancher pour les petites cibles entre les rapports de cadre $R_{16:9}$ et $R_{4:3}$ (figure 2.10). Le gain de précision ne *semble* pas significatif (8,5 mm et 8,7 mm). Les participants ne pouvaient pas améliorer leur performances davantage.

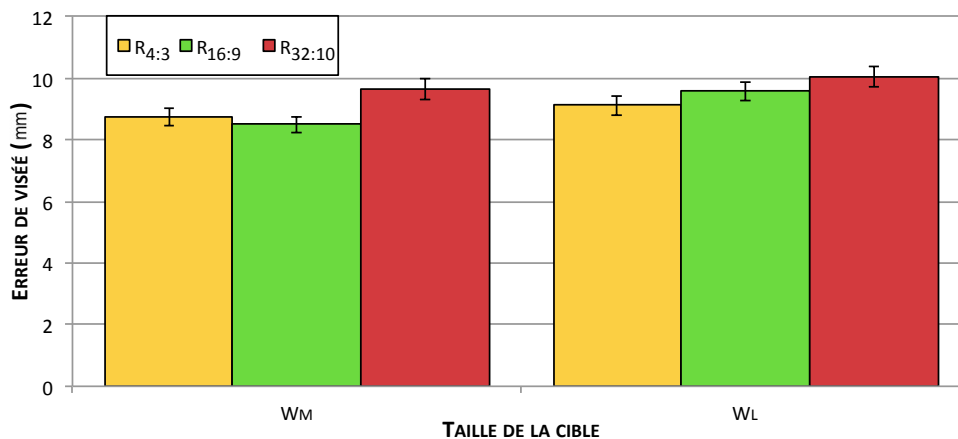


FIGURE 2.10 – Erreur de visée (en mm) en fonction des variables RAPPORT DE CADRE et TAILLE DE LA CIBLE.

Nous avons également trouvé un effet significatif de la POSITION DE LA CIBLE ($F_{3,33}=34,73$, $p < 0,001$) et une interaction entre les facteurs POSITION DE LA CIBLE \times RAPPORT DE CADRE ($F_{6,66}=4,44$, $p < 0,001$). Les comparaisons deux à deux des positions de cibles sont toutes significatives ($p < 0,001$) et l'ordre des résultats est conforme aux résultats de la première expérience (figure 2.11) : les cibles faciles sont restées faciles. Les comparaisons deux à deux ont révélé que l'interaction est due aux cibles 13 (le centre) et 22 (le bord droit). Pour la cible 13, les participants sont restés à peu près toujours aussi précis pour tous les rapports de cadre ($R_{4:3}$: 7,3 mm, $R_{16:9}$: 7,3 mm, $R_{32:10}$: 6,6 mm, $p < 0,05$). À contrario, la cible 22 présente une plus grande valeur d'erreur de visée pour le rapport de cadre $R_{32:10}$ que pour les autres rapports de cadre ($R_{4:3}$: 9,4 mm, $R_{16:9}$: 9,3 mm, $R_{32:10}$: 11,3 mm, $p < 0,05$, figure 2.12). Bien que non significatif, les cibles 1 et 19 présentent également une erreur de visée plus grande pour le rapport de cadre $R_{32:10}$, alors que le centre reste, malgré la déformation de l'espace moteur, une

cible facile à sélectionner.

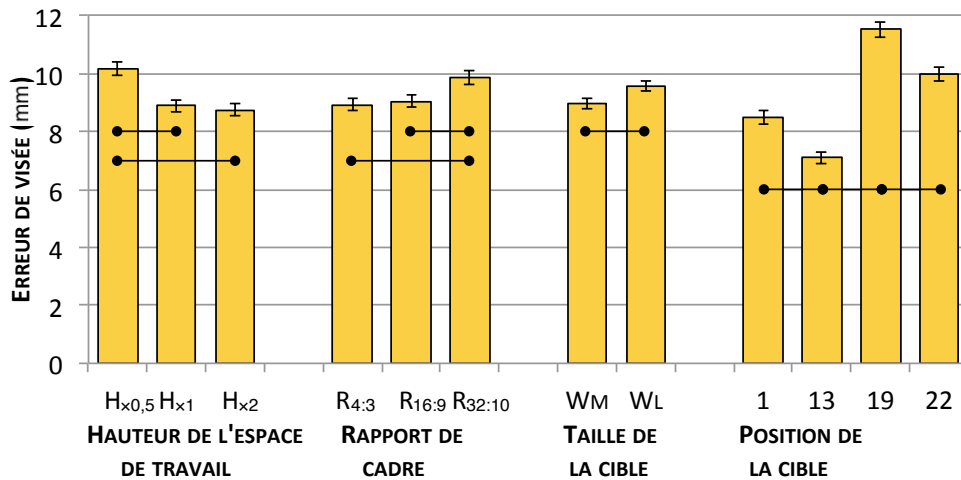


FIGURE 2.11 – Erreur de visée (en mm) en fonction des variables HAUTEUR DE L'ESPACE DE TRAVAIL, RAPPORT DE CADRE, TAILLE DE LA CIBLE et POSITION DE LA CIBLE. Les connexions entre les barres indiquent une différence statistique significative.

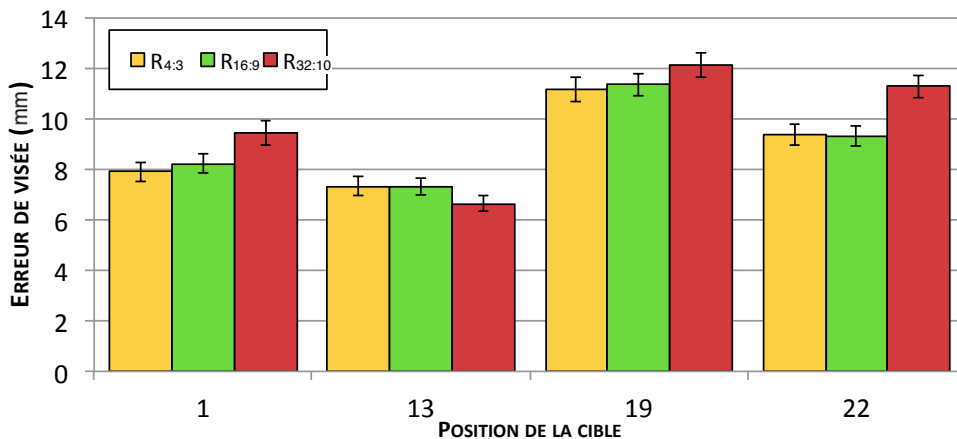


FIGURE 2.12 – Erreur de visée (en mm) en fonction des variables POSITION DE LA CIBLE et RAPPORT DE CADRE.

2.5.3 Taille minimale de cible dans l'espace moteur

Les tailles minimales de cibles pour qu'un utilisateur ait 95% de succès à la première tentative en fonction de facteurs HAUTEUR DE L'ESPACE DE TRAVAIL et RAPPORT DE CADRE sont reportées dans le tableau 2.4. Nous avons calculé un décalage moyen, tous facteurs confondus, orienté en bas à droite par rapport au centre des cibles (1,5 mm à droite, 1,1 mm en bas). La taille observée dans les conditions R_{4:3}, H_{x1} (30,2 mm) est comparable à celle obtenue lors de l'expérience 1 pour les conditions conduisant à la même tâche (GRANDE, 1MAIN), nous lisons ici 30,2 mm quand le tableau 2.4 rapportait 27,8 mm.

HAUTEUR DE L'ESPACE DE TRAVAIL	$H_{\times 0,5}$	$H_{\times 1}$	$H_{\times 2}$
$R_{4:3}$	35,5	30,2	28,2
RAPPORT DE CADRE	$R_{16:9}$	$R_{32:10}$	
	34,5	30,7	30,3
	37,6	32,1	35,1

TABLE 2.4 – Taille minimale de cible (en mm) assurant une sélection sans erreur dans 95% des cas en fonction des variables HAUTEUR DE L'ESPACE DE TRAVAIL et RAPPORT DE CADRE.

Une MANOVA sur les coordonnées du premier contact a mis en évidence une interaction significative entre la HAUTEUR DE L'ESPACE DE TRAVAIL et le RAPPORT DE CADRE ($F_{8,4}=8,1$, $p<0,05$) concernant l'erreur de visée. Les comparaisons deux à deux ont révélé que l'interaction est due à la plus petite hauteur d'espace de travail pour laquelle aucune différence significative n'a été trouvée entre les différents rapports de cadre alors que des différences existent pour les autres valeurs de HAUTEUR DE L'ESPACE DE TRAVAIL. Les tailles de cibles étaient les plus grandes quand l'espace de travail était réduit et il semblerait que leur grande taille ait rendu leur acquisition trop facile en dépit de l'étirement de l'espace de travail et de la déformation des cibles. Nous pouvons en déduire que les utilisateurs n'ont pas atteint leur limite de précision pour cette taille d'espace de travail.

L'effet du facteur d'échelle sur la précision peut être analysé en croisant les conditions qui aboutissent à la même tâche dans l'espace moteur. Par exemple, la hauteur moyenne $H_{\times 1}$ et les cibles moyennes W_M amènent à une hauteur de cible de 10 mm (selon le tableau 2.3), tout comme la plus grande hauteur $H_{\times 2}$ et les grandes cibles W_L . Mais dans le deuxième cas les cibles deux fois plus grandes sont affichées dans un espace de travail deux fois plus grand. Les tailles minimales des conditions présentent un écart de seulement 1,3 mm (29,9 mm et 31,2 mm respectivement), ce qui suggère que le facteur d'échelle n'a pas d'influence sur la précision.

2.5.4 Distribution spatiale des contacts à la première tentative

La figure 2.13 présente les distributions des premières tentatives séparées en fonction de RAPPORT DE CADRE. Les ellipses illustrent la distribution normale bivariée de ces premiers contacts pour chaque position de cible, les positions réelles des centres des cibles sont représentées par les croix. La cible du centre (13) présente une distribution homogène pour tous les rapports de cadre et la précision reste stable, confirmant encore qu'il s'agit d'une cible facile à pointer quel que soit le rapport de cadre. En analysant les autres positions, il est intéressant de remarquer comment les ellipses se décalent au fur et au mesure de l'élargissement de l'espace de travail. En particulier lorsque l'on considère le rapport de cadre $R_{32:10}$ qui présente la plus forte déviation, soutenant l'hypothèse de la difficulté à pointer le centre des cibles lorsqu'elles sont étirées par la projection dans l'espace moteur.

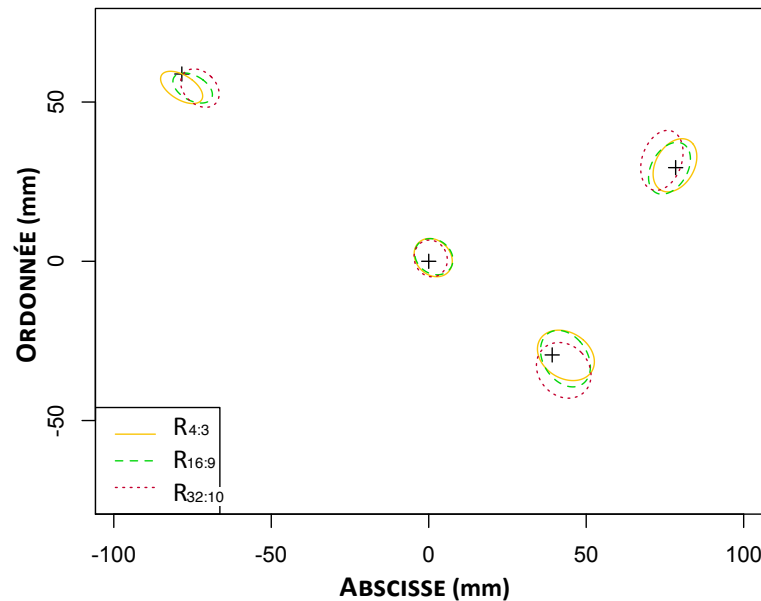


FIGURE 2.13 – Distribution de la position des premières tentatives en fonction du RAPPORT DE CADRE. Les croix représentent le centre des cibles à sélectionner. Les ellipses contiennent 68% des contacts pour chaque position de cible.

2.6 Discussion

Nous avons mené deux expériences afin d'estimer l'impact de différents facteurs sur la performance dans les tâches de pointage tactile indirect absolu. Nous allons maintenant essayer de dégager des recommandations de conception pour de telles tâches et discuter des limites de nos expériences. Enfin, en nous basant sur nos règles, nous présenterons quelques scénarios d'interaction où le pointage absolu indirect pourrait être utile.

L'expérience 1 a montré l'importance de pouvoir voir la surface d'entrée afin d'améliorer les performances par rapport aux conditions aveugles. Quand les participants ne peuvent s'aider que de la proprioception fournie par leur main non dominante, ils améliorent également leurs performances mais de manière moins significative : environ 20% de succès en moins et 3 mm d'erreur de visée en plus. Dans cette première expérience, nous ne pouvions pas contrôler dans quelle mesure les participants regardaient la surface d'entrée. Pour récolter des données quantitatives, il aurait fallu équiper les participants d'un système de suivi de regard. Nous pensons tout de même que le fait de regarder la surface d'entrée, au moins de temps en temps, aide au positionnement du doigt sur l'emplacement mentalement défini.

Recommandation 1 : *Même quand le périphérique d'entrée n'est pas un écran pourvu d'affichage, les utilisateurs devraient être en mesure de voir la surface d'entrée dans le cadre de sélection discrète tactile indirecte absolue. En conséquence, les frontières délimitant la surface active doivent être clairement*

définies et visibles.

La position des cibles a également un impact important sur les performances. Les cibles situées près du centre ou des coins sont faciles à sélectionner. Pour les utilisateurs droitiers, les cibles du quart nord-ouest présentent également une position privilégiée. Les quarts nord-est et sud-ouest sont équivalents. Les cibles les plus difficiles à atteindre sont celles du quart sud-est car elles requièrent de plier le doigt.

Recommandation 2 : *Les designers devraient prendre en compte la latéralité de l'utilisateur pour placer les cibles en conséquence. Si possible, les actions courantes devraient de préférence être placées au centre, dans les coins et dans le quart le plus facile d'accès (supérieur gauche pour les utilisateurs droitiers). Les cibles moins courantes pourraient être disposées dans les quarts de la diagonale, sur les cotés de la main (nord-est et sud-ouest). Le quart difficile d'accès (sud-est) pourrait être réservé aux actions irréversibles, par exemple : suppression, fermeture.*

L'expérience 2 suggère que les performances ne sont pas affectées par le facteur d'échelle : pour une même taille de surface d'entrée, une cible affichée deux fois plus grande dans un espace plus grand n'est pas plus facile ou plus difficile à sélectionner. La première expérience a mis en évidence une déviation des sélections vers le centre de la surface d'entrée, la deuxième expérience explique que cette déviation est d'autant plus prononcée que le rapport de cadre de l'espace de travail visuel diffère de celui du périphérique d'entrée.

Recommandation 3 : *Dans les situations de pointage indirect absolu, le facteur d'échelle n'importe pas, mais la différence de rapport de cadre impacte négativement les résultats. Pour améliorer le taux de succès et diminuer l'erreur de visée, les designers devraient utiliser des rapports de cadre semblables entre l'espace d'entrée et l'espace de travail à l'écran.*

Les participants ont été plus précis avec les plus petites cibles, la taille minimale qu'ils ont pu sélectionner de manière fiable en voyant la surface d'entrée est de 16,8 mm, soit environ la taille d'un doigt (d'après le tableau 2.1). Cette valeur est plus importante que les 10,5 mm trouvés par Vogel et Baudisch pour du pointage direct sur PDA [89], mais ce résultat était prédictible. La co-localisation des espaces d'entrée et d'affichage expliquent sans doute la plus grande précision observée dans leur évaluation. Notre valeur est toutefois très proche des résultats de Holz et Baudisch qui rapportent 15 mm pour du pointage direct sur un *iGesture touchpad* [43] et Gustafson *et al.* qui trouvent une précision de 17,7 mm en pointage indirect sur la paume [35]. Quand nous avons interdit aux participants de regarder leurs mains et le périphérique d'entrée, nous avons trouvé une taille minimale de cible de 23 mm (toujours d'après le tableau 2.1), cette taille est comparable avec les valeurs trouvées dans les travaux évaluant les performances de saisie sur claviers tactiles [78, 26].

Nous avons surtout montré que l'erreur de visée dépend des dimensions de la surface d'entrée. Dans la première expérience, la précision était meilleure sur le petit périphérique d'entrée (tableau 2.1). Nous supposons que la précision

moindre observée avec le plus grand périphérique est due aux plus grandes distances à parcourir dans l'espace moteur. En l'absence de retour visuel pendant la tâche, le mouvement exécuté peut être considéré comme purement balistique. À cause de l'imprécision du mouvement moteur, la destination du mouvement balistique peut ne pas coïncider avec l'emplacement estimé au départ. Cette erreur de destination évolue de manière linéaire avec l'amplitude du mouvement [56].

D'après nos résultats, nous pouvons, en première prévision, supposer que la taille minimale de cible est constante et vaut environ 17 mm en dessous d'une taille de périphérique de 50 mm, et augmente linéairement pour arriver à 28 mm pour une taille de surface d'entrée de 200 mm (d'après le tableau 2.1). Le domaine exploré par nos expériences ne permet pas de statuer sur le comportement au delà de cette taille. Cette hypothèse est résumée sous la forme de l'équation suivante.

Entre 0 et 50 : 17

Entre 50 et 200 : Taille minimale_{moteur} = 13,2 + 0,075 × Taille_{entrée}

Après 200 : ??

Des travaux complémentaires sont nécessaires pour valider cette hypothèse d'évolution linéaire de la taille minimale de cible en fonction de la taille du périphérique d'entrée, ainsi que pour comprendre et chiffrer la relation exacte entre ces deux tailles. Cela pourrait avoir d'importantes conséquences car la taille de cible accessible dans l'espace moteur est directement liée à la taille des objets à l'écran : la taille des cibles à l'écran doit être supérieure à la taille minimale de cible dans l'espace moteur multipliée par le rapport des diagonales des espaces d'affichage et d'entrée (Équation 2.1).

$$Taille\ cible_{\text{écran}} \geq Taille\ minimale_{\text{moteur}} \times \frac{Diagonale\ surface\ de\ travail}{Diagonale\ surface\ d'entrée} \quad (2.1)$$

Recommandation 4 : *Les designers d'interfaces de pointage indirect absolu devraient prendre en compte la taille des cibles dans l'espace moteur et vérifier que celle-ci est facilement sélectionnable. Des recherches permettraient d'en savoir davantage sur la précision de larges mouvements et de valider une approche systématique pour dimensionner correctement de telles interfaces.*

Pour finir, nous proposons quelques scénarios où les résultats de nos expériences peuvent se révéler utiles.

Considérons qu'un propriétaire d'un ordinateur portable avec un écran de 15" (323 × 202 mm) désire utiliser le pavé tactile de dimension 105 × 76 mm pour faire du pointage absolu tel que défini précédemment. Dans sa configuration, une taille minimum de cibles de 21 mm dans l'espace moteur (extrapolée d'après nos résultats et les équations précédentes) conduit à une taille minimale à l'écran de 62 mm. La taille des pavés tactiles actuels ne joue donc pas

en faveur de l'interaction absolue. Néanmoins, en examinant la taille des objets classiquement affichés sur un écran, on peut considérer que la réorganisation des fenêtres est possible. L'utilisateur pourrait entrer dans un mode spécial d'interaction absolue, par exemple en maintenant une touche du clavier enfoncée. Dans ce mode, il serait capable de sélectionner et déplacer les fenêtres au moyen de touchés absolus et de glissement sur le pavé tactile. En utilisant ce mode absolu au lieu de l'interaction classique, il serait en mesure de manipuler plusieurs fenêtres simultanément.

Cette même personne peut avoir accès à un mur d'affichage composé de plusieurs écrans. Dans cette situation, elle pourrait désigner un des écrans en indiquant sa position de manière absolue sur le pavé tactile pour envoyer une fenêtre, un objet ou le pointeur souris directement sur cet écran. Elle entre dans un mode absolu local pour sélectionner une fenêtre sur l'écran courant en tapant à la position correspondante, puis passe en mode absolu distant pour désigner l'écran de destination. Avec la taille de cible minimale de 21 mm, le pavé tactile peut adresser une grille de 5×4 écrans.

Dans un dernier scénario, imaginons une surface tactile localisée sur l'avant-bras permettant de l'utiliser avec une montre interactive. Avec des tailles de 30×30 mm pour l'écran de la montre et 70×70 mm pour la surface tactile, la taille de cible minimale devra être d'au moins 18 mm dans l'espace moteur, ce qui équivaut à une taille de cible de 8 mm sur l'écran de la montre.

2.7 Conclusion

En partant du manque de connaissance concernant les facteurs influant les tâches de pointage tactile indirect absolu, nous avons conduit deux expériences pour évaluer l'impact de ces conditions. La première expérience portait sur les conditions d'utilisation et la taille de surface d'entrée et a révélé que les utilisateurs améliorent leurs performances lorsqu'ils peuvent voir la surface d'entrée, même si rien n'y est affiché. Nous avons également montré que la taille de cible minimale qu'un utilisateur peut sélectionner n'est pas constante en fonction des dimensions de la surface d'entrée mais augmente quand la surface s'agrandit. La seconde expérience centrée sur les facteurs d'échelle et la différence de rapports de cadre a démontré que le facteur d'échelle n'influe pas sur la tâche mais que des différences importantes de rapports de cadre entre l'entrée et l'affichage affectent négativement les performances. Ces résultats nous ont permis de formuler des règles de conception pour le design d'interactions tactiles indirectes absolues. De plus, nos résultats plaident pour l'hypothèse d'une augmentation *linéaire* de la taille minimale de cible dans l'espace moteur en fonction de la taille du périphérique d'entrée. Une campagne d'expérience orientée sur l'étude systématique de l'erreur de visée en variant les tailles d'entrée devrait permettre de répondre de manière définitive à cette question.

Un des enseignements que nous avons tirés de ces résultats est que le pointage absolu indirect se limite à la sélection de cibles assez grandes. Dans la section 1.4.4 nous avons abordé l'expérience de Schmidt *et al.* [77] qui, pour

rappel, a comparé la sélection directe et indirecte en affichant sur une table interactive les contours des bras. Cette expérience montrait que les utilisateurs séparent leurs mouvements entre d'une part le pointage en deux dimensions et d'autre part la sélection en recherchant le contact avec la surface tactile. Dans nos expériences nous n'avons pas observé d'hésitations pour rechercher la tablette tactile, mais la tâche consistait justement à pointer cette surface et n'incluait pas de manipulation sur cette surface. Pourtant la perte de contact avec la surface peut obliger l'utilisateur à rediriger son attention de la tâche sur l'écran vers le périphérique tactile indirect. L'interaction indirecte tendrait donc plutôt à privilégier le contact avec la surface, donc à faire du contrôle relatif pour ne pas perturber la réalisation de tâches plus complexes. Dans le chapitre suivant, nous allons nous intéresser à l'introduction du concept de contrôle relatif dans l'interaction multi-doigts pour améliorer la précision accessible.

Chapitre 3

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

« *Great opportunities also pose great challenges* »

—Donald Norman in [69]

Sommaire

3.1	Introduction	63
3.2	Principes de conception	65
3.3	Scénario d'illustration	65
3.4	Proposition	66
3.4.1	Un curseur...	66
3.4.2	... multi-doigts	67
3.4.3	Des curseurs : multi-points	69
3.5	Évaluation préliminaire	72
3.5.1	Matériel, tâche et procédure	72
3.5.2	Résultats et limitations	73
3.6	Discussion	74
3.7	Conclusion	75

3.1 Introduction

Le chapitre précédent étudiait la mise en correspondance absolue entre la surface tactile et l'écran qui est le moyen le plus simple pour introduire plusieurs curseurs en interaction multi-points indirecte. Cette approche pose cependant des problèmes de pré-visualisation des contacts dans l'espace visuel et de précision de manipulation.

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 [20, 68]. Les travaux antérieurs ont notamment proposé d'augmenter la souris d'une surface tactile pour manipuler des éléments situés sous le curseur [104, 86, 11]. Des produits mixant l'interaction souris et les commandes multi-doigts sont déjà commercialisés, comme le *Magic Trackpad*¹⁴ d'Apple et la *Touch Mouse*¹⁵ de Microsoft. 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, comme le zoom ou la rotation d'objets, ou d'autres commandes liées au système de fenêtrage.

Moscovich et Hughes ont introduit des curseurs multi-doigts permettant de manipuler de manière continue des objets et regroupant dans le même geste le déplacement de l'espace de travail et la manipulation [65]. Le *HandCursor* repose sur un curseur relatif représentant l'espace de travail et sur le placement absolu des doigts à l'intérieur de cet espace. Cette technique présente l'avantage de proposer une manière élégante de supporter une interaction multi-doigts indirecte et précise. Cependant le contrôle de plusieurs curseurs requiert l'usage de plusieurs surfaces d'entrée. Ainsi l'interaction multi-points ne peut être obtenue qu'au travers une segmentation statique de l'espace d'entrée vers les curseurs.

Après avoir étudié la sélection indirecte absolue comme mode d'interaction exclusif, qui impose alors l'utilisation de grandes tailles de cibles, nous allons utiliser le pointage absolu pour sélectionner un curseur à l'écran. Chaque curseur a une zone d'influence autorisant ainsi la sélection du curseur le plus proche du doigt. D'autres gestes assureront le contrôle relatif des curseurs, permettant d'interagir précisément sur de petits objets. Pour tirer parti de la précision amenée par les curseurs, nous nous sommes intéressés à l'interaction sur écran de grandes dimensions, maximisant la disparité de taille entre les espaces d'entrée et d'affichage. Classiquement, pour interagir avec ce type d'écran, l'utilisateur utilise un clavier et une souris, ou encore des périphériques 3D. Nous retenons ici l'usage d'une tablette tactile multi-points permettant d'aller plus loin qu'une simple souris tout en restant moins contraignante qu'un système de suivi 3D.

Nous présentons dans ce chapitre une technique d'interaction indirecte combinant les aspects absolus et relatifs pour contrôler plusieurs curseurs multi-doigts en utilisant un seul périphérique d'interaction multi-points. Le positionnement absolu sert au positionnement et à la sélection discrète de curseurs, ainsi qu'au positionnement des doigts autour des curseurs pour la manipulation multi-doigts. Une présentation de la technique est disponible à l'adresse

14. <http://www.apple.com/magictrackpad/>

15. <http://www.microsoft.com/hardware/fr-fr/p/touch-mouse/3KJ-00002>

suivante : <http://bit.ly/1c0Y07K>. Cette vidéo peut fournir un support visuel à la lecture des paragraphes qui suivent.

3.2 Principes de conception

En nous basant sur les travaux et analyses de la littérature, nous désirons que notre technique respecte les principes suivants : L'interaction relative au travers d'un curseur a été retenue afin d'interagir de manière rapide et précise par le biais de fonctions de transfert. L'interaction à un doigt doit rester « classique » et ne demander aucun apprentissage : la prise en main doit être instantanée et s'appuyer sur les connaissances de l'utilisateur. Pour cela l'objectif est de mettre à profit l'expérience des utilisateurs avec les pavés tactiles. L'engagement de doigts supplémentaires doit permettre de manipuler les objets avec plusieurs points d'interaction. L'utilisateur doit pouvoir accéder à plusieurs parties de l'écran simultanément grâce à plusieurs curseurs. Un unique périphérique multi-points doit être employé pour potentiellement contrôler plusieurs curseurs. La totalité de la surface pourra ainsi être employée pour effectuer de grands déplacements, tout en permettant une distribution dynamique du périphérique vers les curseurs.

3.3 Scénario d'illustration

Afin d'offrir dès à présent au lecteur une vision globale de l'utilisation de notre technique et avant d'expliquer en détails les divers aspects et choix, nous proposons d'illustrer un exemple concret d'utilisation au travers du scénario suivant :

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, leurs implémentations et détails techniques seront détaillées dans les paragraphes suivants.

3.4 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 [65], adapté pour permettre la manipulation de petits objets et enrichi d'un mécanisme de création, manipulation et destruction de curseurs.

La Figure 3.1 présente l'automate reprenant l'ensemble des actions qui ont lieu à chaque fois qu'un doigt est posé. Chaque doigt possède sa propre machine à état qui évolue indépendamment, mais les listes des curseurs, doigts et des positions mémorisées (Débrayage, *Tap*) sont communes à toutes les machines à états. Les transitions sont écrites sous la forme « Événement [& condition] [/ actions] ».

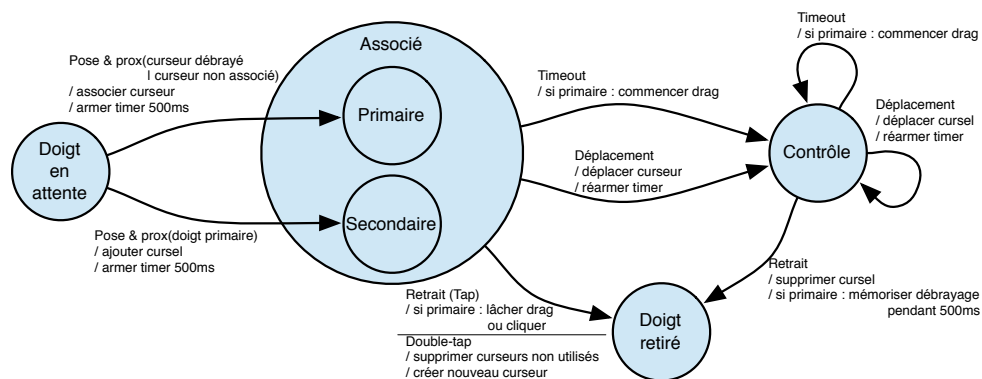


FIGURE 3.1 – Automate de fonctionnement. Chaque doigt est associé à un curseur dès l'instant où il est posé sur le périphérique, il devient doigt primaire s'il est plus proche d'un curseur non utilisé ; secondaire s'il est plus proche d'un autre doigt primaire.

3.4.1 Un curseur...

Afin de permettre une prise en main instantanée de la technique, l'interaction avec un doigt est similaire à l'utilisation d'un curseur d'ordinateur portable contrôlé par un pavé tactile.

Déplacement

Le déplacement d'un doigt sur la surface tactile contrôle un curseur de façon relative au travers d'une fonction de transfert. L'utilisateur interagit ainsi avec un curseur qui lui permet de sélectionner de petites cibles en prévisualisant le point d'interaction et la fonction de transfert non linéaire assure un équilibre entre précision et rapidité du déplacement. La fonction de transfert choisie est celle utilisée par OS X pour les pavés tactiles [20], ce système d'exploitation est le seul ayant une fonction qui ait été spécifiquement conçue pour une configuration tactile. Enfin pour limiter les débrayages (*clutch*) nécessaires

pour parcourir la totalité d'un écran mural de 6 mètres, tout en proposant un niveau de précision adapté à la résolution relativement basse de cet écran (10.6 dpi), nous multiplions les déplacements et positions par 7 après calcul du gain.

Gestion de la sélection d'objets

La génération d'un événement discret permet de distinguer le survol de la sélection. Notre technique a pour objectif de proposer le contrôle de plusieurs curseurs depuis un seul périphérique tactile et implique donc de gérer plusieurs états de sélection, un par curseur. Moscovich et Hughes avaient ajouté un bouton sur la surface tactile rendant la surface cliquable [65], mais cette solution est inappropriée à la création et au contrôle dynamique de plusieurs curseurs. De plus, notre périphérique tactile ne fournit pas d'informations permettant de détecter les doigts en survol comme cela est possible avec les technologies reposant sur la vision [59]. L'utilisation d'une seule surface tactile nous a donc conduit à implémenter un clic logiciel. Ainsi, nous avons ainsi considéré la détection d'un événement de *tap* pour générer un *clic*, qui est une solution simple et déjà courante sur les pavés tactiles classiques.

Cet événement de *tap* est détecté si un doigt est posé puis retiré avant un délai de 500 ms sans avoir été déplacé (Figure 3.1, transition « Tap »), temps qui correspond à celui utilisé par défaut pour effectuer un clic ou un double-clic sur la plupart des systèmes. Il est également possible de déplacer le curseur tout en maintenant la sélection activée (*drag*). Il faut pour cela conserver le doigt immobile pendant 500 ms (Figure 3.1, transition « Timeout »). La sélection reste maintenue tant qu'il n'y a pas de nouveau *tap* et l'utilisateur peut alors soulever et reposer le doigt (débrayage) plusieurs fois afin d'effectuer des gestes de plus grande amplitude sans perdre les objets sélectionnés.

3.4.2 ... multi-doigts

L'interaction avec un curseur permet de sélectionner un objet à la fois. La manipulation de plusieurs propriétés ou de plusieurs objets simultanément demande l'engagement de plusieurs doigts dans l'interaction.

Cursels

Un curseur représente un point d'interaction mais aussi un sous-espace de travail pouvant accueillir un ensemble de doigts agissant dans une même zone. Nous avons repris les principes posés par *HandCursor* [65] pour le placement absolu des doigts et le déplacement du curseur. Le premier doigt posé est utilisé comme origine du repère de l'espace de travail et les doigts suivants sont placés en respectant leur position absolue dans ce repère. L'utilisateur peut ainsi facilement associer les doigts aux curseurs et inversement. Le centre de gravité des doigts contrôle le curseur par le biais d'une fonction de transfert

conçue pour assurer rapidité et précision. Ainsi un déplacement d'un seul doigt implique le déplacement du repère de l'espace de travail et donc de l'ensemble des curseurs. Cependant à la différence du *HandCursor* qui change d'origine quand le premier doigt disparaît et passe au doigt suivant, nous enregistrons le premier doigt comme référence permanente (tant qu'il y a des doigts). En effet, le changement de doigts (ajout, suppression) au cours de l'interaction peut impliquer des déplacements brusques des curseurs à l'écran. Dans notre technique, le premier doigt posé reste au centre du curseur et réapparaît au même endroit en cas de retrait, repose.

Le premier doigt posé sur le périphérique sert de référence (Figure 3.1, état *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 (figure 3.2). De cette façon, l'utilisateur peut facilement identifier quel doigt bouger en fonction du curseur secondaire qu'il souhaite déplacer. Ces curseurs secondaires apparaissent autour du curseur comme des éléments satellites. Nous les appelons par la suite *cursel* pour « *cursor element* ».

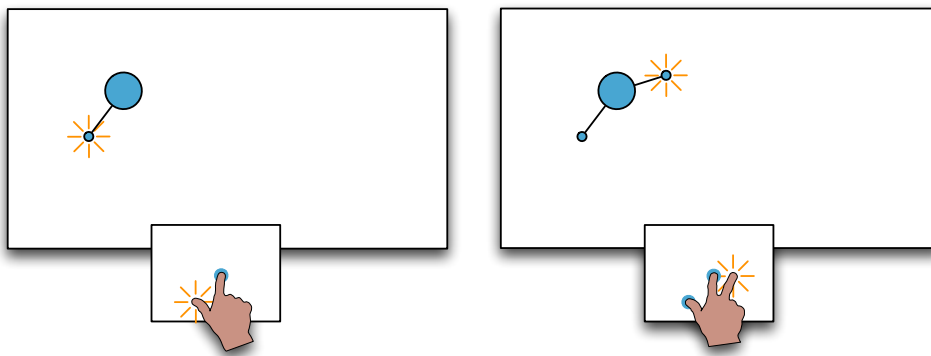


FIGURE 3.2 – Le doigt primaire est utilisé comme référence absolue pour le placement des cursels autour du curseur. À gauche, l'utilisateur pose le pouce en dessous de l'index qui contrôle déjà le curseur. Un cursel apparaît donc en dessous du curseur. À droite, lorsque qu'un troisième doigt est ajouté, un nouveau cursel est positionné au même endroit, ici à droite et légèrement au-dessus du curseur.

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 avec le périphérique. La gestion de l'état de sélection (*clac*) de l'ensemble curseur/cursels est assurée par le doigt primaire.

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. 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-doigts permet soit de manipuler un objet ou d'en sélectionner plusieurs selon que les cursels sont positionnés ou non sur d'autres objets. Mais la manipulation de petits objets peut toujours poser un problème, car nécessitant de poser deux cursels (ou un curseur et un cursel) sur un même objet. Lorsque deux doigts sont posés sur le périphérique d'entrée, il reste un écart entre le centre de contact des doigts, que l'on peut estimer à 15 mm (environ la taille d'un doigt). Dans notre configuration, cette distance implique que deux points d'un curseur ne pourront jamais se rapprocher de plus de 105 mm. Les objets inférieurs à cette taille ne pourront donc pas être manipulés avec plusieurs doigts s'il n'y a pas de mécanisme aidant à leur manipulation.

Pour pouvoir manipuler un objet, le curseur doit nécessairement être positionné dessus. Les cursels peuvent par contre être positionnés en dehors de l'objet (figure 3.3). Les cursels directement situés sur d'autres objets ne sont pas pris en compte pour la manipulation de l'objet situé 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 3.3 à gauche).

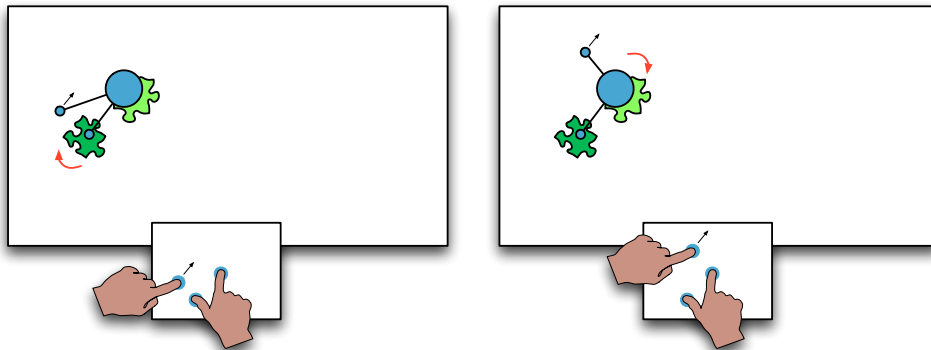


FIGURE 3.3 – Regroupement des doigts positionnés dans l'espace vide pour faciliter la manipulation de petits objets. À gauche, le 3^e doigt est regroupé avec l'objet de gauche déjà sélectionné par le pouce et permet une manipulation à deux doigts. À droite, le 3^e doigt est désormais plus proche de l'objet sous le curseur et regroupé avec ce dernier.

3.4.3 Des curseurs : multi-points

Lorsque l'utilisateur désire accéder à plusieurs parties de l'écran en même temps ou accéder à un endroit éloigné de là où se trouve le curseur, il peut manifester l'envie d'engager sa deuxième main et d'utiliser un autre curseur.

Création et destruction de curseurs

La création d'un nouveau curseur se fait par un double tap sur la surface (Figure 3.1, transition « Double-tap »). Un nouveau curseur apparaît alors à l'écran à la position absolue correspondante à l'emplacement du *double tap* dans l'espace d'entrée (figure 3.4). 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.

L'utilisateur peut ainsi supprimer les curseurs devenus inutiles. Grâce à ce mécanisme de positionnement absolu, l'utilisateur peut, dès la création, définir approximativement l'endroit où il souhaite interagir de façon similaire à ARC-Pad [62]. La sélection et la manipulation d'objets se poursuivent ensuite en déplaçant le curseur de manière relative et en bénéficiant de la précision apportée par la fonction de transfert.

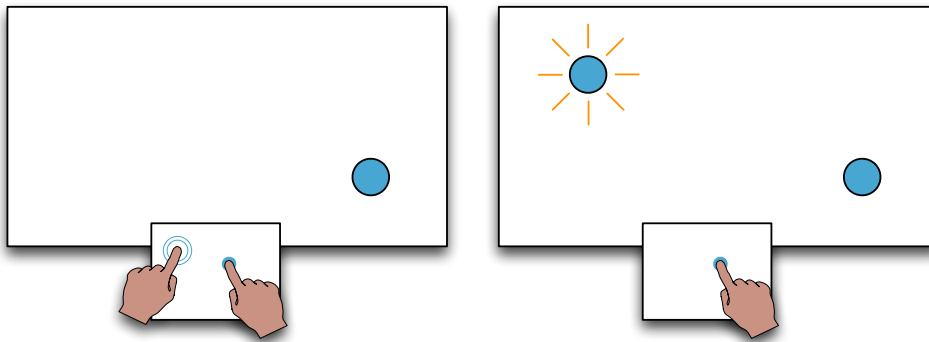


FIGURE 3.4 – Création d'un curseur supplémentaire. À gauche, l'utilisateur effectue un *double tap* sur la surface d'entrée. À droite, un nouveau curseur est créé à la position absolue correspondante.

Association doigts et curseurs

L'association entre les doigts et les différents curseurs créés est réalisée en considérant les positions absolues des doigts et des curseurs. Un doigt est associé au curseur le plus proche dans les repères absolus (figure 3.5).

Lorsqu'un doigt est posé sur la surface, il est étiqueté doigt primaire ou doigt secondaire. Si le doigt est plus proche d'un curseur non associé, il est associé au centre du curseur (figure 3.5) et est considéré comme primaire (Figure 3.1, transition « Pose & prox(curseur non associé) »). S'il est plus proche d'un doigt primaire (figure 3.6), il est considéré comme secondaire et ajouté en tant que curseur au curseur correspondant au doigt primaire (Figure 3.1, transition « Pose & prox(doigt primaire) »).

Lors d'un débrayage du doigt primaire, les curseurs débrayés deviennent attractifs et prioritaires pendant 500 ms : l'association est faite avec le plus proche de ces curseurs en priorité (Figure 3.1, transition « Pose & prox(curseur débrayé) »).

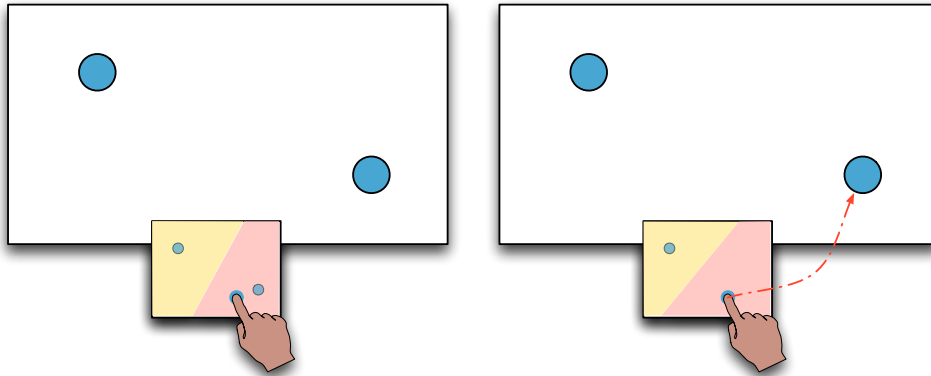


FIGURE 3.5 – Illustration de l'association et des points d'attraction. À gauche, la position des curseurs non utilisés à l'écran est reportée dans l'espace de contrôle et chacune des positions est un centre d'attraction. Le doigt va donc être associé au curseur le plus près. À droite, le doigt prend le contrôle du curseur de droite, il devient point d'attraction à la place du curseur.

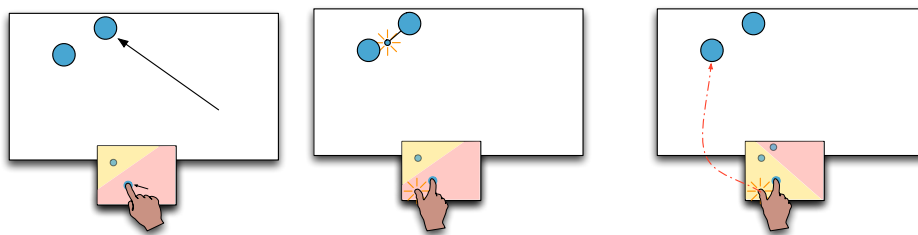


FIGURE 3.6 – Association des doigts secondaires. À gauche, le doigt primaire déplace le curseur à proximité d'un autre curseur, la zone d'attraction se déplace et suit le doigt. Au centre, un deuxième doigt est posé dans la zone d'influence du doigt primaire : l'utilisateur ajoute ainsi un curseur au curseur qu'il contrôlait déjà. À droite, si l'on n'utilisait que les curseurs comme centres d'attraction, cela conduirait à une erreur d'association : le pouce apparaît en dessous de l'index mais le curseur associé est assez loin, l'association se ferait ici avec un autre curseur dont la position est plus proche.

Cette méthode d'association au plus proche permet de débrayer un curseur, passer d'un curseur à un autre et faire apparaître des curseurs à volonté. Cette méthode présente l'avantage d'avoir un sens pour l'utilisateur et d'être prédictive; sans cela il serait impossible de savoir comment prendre le contrôle d'un curseur particulier ou ajouter un deuxième doigt au curseur voulu. L'attractivité des doigts primaires à la place des curseurs qu'ils contrôlent permet d'ajouter des doigts supplémentaires au bon curseur même si celui-ci a été déplacé (figure 3.6 à droite). Quand l'utilisateur désire ajouter un doigt à un curseur, ce nouveau doigt va être placé à proximité du premier doigt de la même main. Dans le cas où les curseurs sont croisés au cours de l'interaction, l'ajout de curseurs s'effectue sans erreur, quand les mains seront retirées de la surface tactile, chaque curseur sera accessible par le côté du périphérique où il est situé. L'organisation dynamique des zones d'attraction permet de refléter la position des curseurs : un curseur à gauche de l'écran sera accessible

par la partie gauche du périphérique d'entrée. Dans le cas d'une segmentation statique du périphérique (par exemple en utilisant la partie gauche pour contrôler un pointeur et la partie droite pour en contrôler un autre) ou en utilisant plusieurs périphériques comme Moscovich et Hughes [65], le contrôle resterait inversé : la surface gauche contrôlant le curseur droit, et la surface droite contrôlant le curseur gauche, ce qui est déstabilisant pour l'utilisateur [42, 55] et l'oblige à mémoriser l'inversion.

3.5 Évaluation préliminaire

Notre objectif était d'évaluer l'effet de notre technique sur une tâche de positionnement 2D en la comparant à la technique HandCursor et à l'interaction indirecte absolue. Nous avons conçu et développé un protocole d'expérience en essayant d'encourager l'usage des deux mains afin de tirer parti de la manipulation bimanuelle pour gagner en rapidité et de voir si les utilisateurs arrivent à sélectionner et utiliser plusieurs curseurs.

3.5.1 Matériel, tâche et procédure

La tâche consistait à replacer au centre de l'écran, les quatre morceaux d'une image (figure 3.7). Les pièces étaient colorées en rouge tant que leur position n'était pas correcte, puis en vert lorsque l'image était correctement placée, orientée et à l'échelle. Les pièces validées étaient ensuite automatiquement verrouillées à leur emplacement et retrouvaient leur teintes normales. Les quatre pièces à replacer étaient présentées simultanément afin d'autoriser la manipulation de plusieurs pièces.

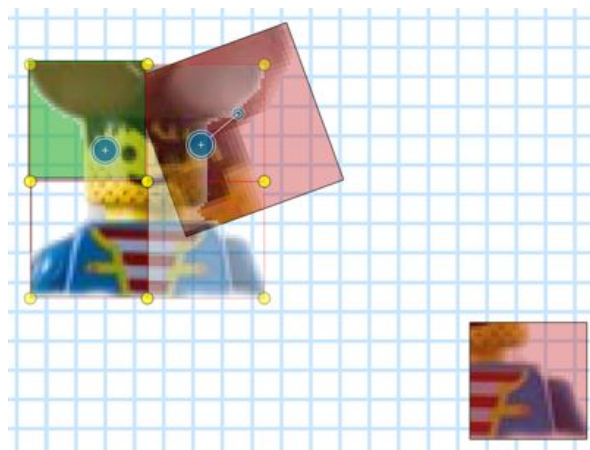


FIGURE 3.7 – Aperçu de l'expérience. Manipulation simultanée de deux objets. Le curseur droit positionne une pièce à son emplacement, la pièce se colore en vert. Le curseur droit manipule une autre pièce afin de la rétrécir et de l'orienter correctement, elle est pour l'instant colorée en rouge.

Le périphérique tactile utilisé était un iPad de première génération, la surface tactile était de dimension 196×147 mm. L'écran utilisé pour l'expérimentation était un écran mural de $5,96 \times 2,43$ mètres présentant une résolution de 2392×1050 pixels. Les participants étaient assis face à l'écran, la tablette posée devant eux pouvait être manipulée des deux mains. Une application exécutée sur l'iPad envoyait les événements tactiles de la tablette vers l'ordinateur contrôlant le mur via un réseau sans fil et le protocole TUIO.

Les variables indépendantes étaient la TECHNIQUE utilisée, la TAILLE des pièces, et les TRANSFORMATIONS nécessaires pour atteindre la position finale. Les TECHNIQUES étaient l'interaction indirecte absolue, la technique HandCursor et notre technique. La distance entre la position initiale de la pièce et sa destination était fixée à 1,20 m. La variable TAILLE avait deux valeurs possibles $W_L = 400$ mm et $W_M = 200$ mm. La marge d'erreur tolérée pour le positionnement était représentée par des disques jaunes de rayon égal à 10% de la taille de la pièce. Les TRANSFORMATIONS nécessaires étaient combinées de manière à former 4 niveaux de difficultés progressifs :

- T : Translation seule ;
- TS+ : Translation + agrandissement (Scale) des pièces d'un facteur 2, il fallait donc la réduire pendant la manipulation ;
- TRS+ : Translation + Scale + Rotation de la pièce de $\pm 45^\circ$;
- TRS- : Translation + Rotation + Réduction (Scale) des pièces d'un facteur 2, la pièce pouvait donc être difficile à sélectionner avec deux doigts.

3.5.2 Résultats et limitations

L'expérience n'a pas été conduite car la réalisation de la tâche avec deux curseurs imposait une trop forte charge cognitive pour être répétée à de multiples reprises. Nous ne reportons donc que les observations qualitatives faites pendant les essais préliminaires concernant la découverte et l'utilisation de notre technique par quatre participants.

Les utilisateurs n'ont eu aucun mal à prendre en main notre technique, ils ont pu contrôler un curseur avec un doigt sans aucun apprentissage, puis ont très rapidement utilisé plusieurs doigts pour manipuler les objets. Les participants ont apprécié de retrouver le fonctionnement d'un pavé tactile classique. Par ailleurs la transition vers la manipulation multi-doigts est considérée aisée et permet de sélectionner facilement plusieurs objets. Le regroupement des doigts pour faciliter la manipulation de petits objets est apparu comme simple et efficace. La gestion de la sélection a requis un léger temps d'apprentissage, en effet, l'utilisateur doit réaliser un geste de *tap* avec le premier doigt posé tout en laissant les autres doigts immobiles. Notre technique favorise l'interaction à un doigt pour déplacer le curseur, puis l'engagement des doigts supplémentaires, dans la plupart des cas le premier doigt posé était l'index. Le choix de déléguer l'état de sélection de l'ensemble des curseurs au premier doigt (donc à l'index) est ainsi cohérent avec l'utilisation que nous avons pu observer.

Dans un premier temps, le maintien de la sélection était réalisée par un « tap et demi » avec le doigt primaire. Ce geste a été jugé trop contraignant car requérant de soulever plusieurs fois un doigt alors que les autres ne doivent pas être déplacés. Nous avons alors privilégié l'utilisation d'un clic long car plus aisé à réaliser même si cela implique une légère temporisation.

Les utilisateurs ont pu facilement créer de nouveaux curseurs et les détruire. Ils n'ont pas ressenti de difficulté pour sélectionner les curseurs et ajouter des doigts au curseur souhaité. L'association correspond à leur attente et ne les a pas surpris. 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.

Toutefois, la destruction des curseurs non utilisés lors de la création d'un nouveau curseur impose aux utilisateurs de désigner explicitement les curseurs qu'ils désirent garder, sinon les curseurs vont être involontairement détruits. Cela peut poser quelques problèmes, notamment lorsque l'utilisateur désire conserver plusieurs curseurs. La grammaire de gestion du cycle de vie des curseurs est limitée et mériterait d'être étendue.

Malgré la familiarisation rapide à la création de nouveaux curseurs et à leur sélection, les gestes bimanuels ont été perçus comme très difficile. La manipulation relative de deux curseurs demande un haut niveau de coordination et de séparation des mains, les déplacements relatifs couplés à l'engagement potentiel de plusieurs doigts de chaque main a généré une forte charge cognitive.

3.6 Discussion

La technique proposée permet un engagement progressif de l'utilisateur au travers du contrôle d'un seul curseur avec un doigt, 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. Les premiers résultats sont encourageants : les utilisateurs arrivent facilement à créer et détruire des curseurs, la méthode d'association doigts / curseurs est efficace et compréhensible. Sans explication de la méthode utilisée, les utilisateurs arrivent facilement à contrôler et à ajouter des curseurs aux curseurs. Contrairement aux techniques de l'état de l'art, la répartition dynamique de la surface tactile permet de profiter de toute la surface pour chacun des curseurs et désambiguïse l'accès aux curseurs dans le cas où ceux-ci seraient croisés. La capacité à sélectionner une cible sans prévisualisation, que nous avons évaluée dans le chapitre précédent, explique cette facilité d'utilisation.

En dépit d'une grammaire de création limitée, la technique supporte des usages secondaires. Le positionnement absolu des nouveaux curseurs permet de téléporter le curseur directement à l'endroit voulu. L'utilisateur peut ainsi éviter

de parcourir de grandes distances et interagir avec notre technique d'une manière semblable à ARC-Pad [62]. Il a la possibilité de créer plusieurs curseurs, et potentiellement plus de deux curseurs, il peut donc positionner des curseurs là où il interagit régulièrement, interagir ensuite avec de nouveaux curseurs, puis revenir aux curseurs anciennement créés pour éviter des déplacements supplémentaires. Le jonglage entre les curseurs recourt donc à l'utilisation de plusieurs curseurs existants simultanément, mais utilisés à des instants différents. Dans ce cas, l'interaction est bien multi-points, dans le sens où l'on peut accéder à divers parties de l'écran en se basant sur un déplacement physique dans l'espace d'entrée.

Ces résultats pourraient être validés en repensant complètement le protocole d'expérimentation. Nous pourrions segmenter l'étude en une tâche de positionnement avec un seul curseur, afin de valider le curseur multi-doigt et la possibilité de coordonner les manipulations. Une seconde expérience de sélection multiple de cibles permettrait d'évaluer les performances du jonglage entre plusieurs curseurs déjà créés et de comparer cette approche à l'interaction absolue – et ses problèmes de précision – et à la répartition statique de deux périphériques vers deux *HandCursors*. Enfin l'interaction bimanuelle simultanée pourrait probablement être facilitée en améliorant la fonction de transfert et en réduisant sensiblement la différence de taille entre l'espace d'entrée et l'espace d'affichage (30 fois plus large dans notre cas). Le paramétrage de ces deux facteurs réduirait le nombre de débrayage et l'impression de découplage entre les mains et les curseurs. Moscovich et Hughes ont montré dans leur étude sur les mises en correspondance indirectes multi-doigts [66] que les manipulations à une seule main étaient compatibles avec des transformations entre les espaces d'entrée et d'affichage. Dans leur expérience, une rotation de l'espace de travail n'entraînait qu'une légère hausse du taux d'erreur pour la manipulation à une main. Dans le même temps, la même modification augmentait significativement le taux d'erreur lors de manipulation faisant intervenir un doigt de chaque main. Le découplage entre les doigts et les points manipulés dû à la fonction de transfert dégrade les performances des manipulations bimanuelles coordonnées.

3.7 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.

Ce chapitre a montré l'intérêt de pouvoir créer et contrôler plusieurs curseurs, même si leur manipulation simultanée reste difficile. Il est intéressant d'avoir

des curseurs indépendants pour accéder à différentes parties de l'écran et effectuer diverses tâches. Cela peut aider à réintroduire une interaction multi-points avec un périphérique indirect de taille réduite. Nous allons maintenant tirer parti de l'usage de plusieurs curseurs sur une plus grande surface tactile en créant des espaces de contrôle pour chacun de ces curseurs.

Chapitre 4

Interaction multi-points directe et indirecte sur murs interactifs

*« Any problem in computer science can be solved with another level of
indirection »*

—David Wheeler

Sommaire

4.1	Introduction	77
4.2	Tactilisation d'un mur de réalité virtuelle	80
4.2.1	Matériel	81
4.2.2	Traitement d'image	82
4.3	WallPad	85
4.3.1	Un pavé tactile	85
4.3.2	Interaction directe délocalisée	86
4.4	Cycle de vie des WallPads	87
4.4.1	Création	87
4.4.2	Gestion dynamique et reconfigurable	88
4.5	Implémentation	89
4.6	Cas d'usage	90
4.6.1	Accès rapide à des fonctions distantes	90
4.6.2	Personnalisation de l'espace de travail	91
4.6.3	Aspects collaboratifs	92
4.6.4	Compatibilité et apports des WallPads avec les applications régulières	93
4.6.5	Au delà du mur	93
4.7	Discussion	94
4.8	Conclusion	96

4.1 Introduction

Le chapitre précédent présentait une technique multi-points indirecte d'interaction à distance. Cette technique, malgré les limitations inhérentes au contrôle

de plusieurs curseurs relatifs, introduisait la gestion d'un ensemble dynamique de curseurs. Dans ce chapitre, nous allons amener le concept de gestion de plusieurs curseurs un peu plus loin. Pour cela nous nous sommes intéressés aux écrans tactiles de beaucoup plus grandes dimensions et particulièrement aux murs interactifs qui proposent une très grande surface d'affichage et d'interaction et permettent à plusieurs utilisateurs de collaborer avec de grandes quantités de données.

L'interaction tactile co-localisée avec le contenu contribue à créer un fort sentiment de contrôle direct immédiat, mais l'interaction avec du contenu distant peut rapidement devenir fatigante et inefficace à cause des déplacements répétés du corps. De plus les objets d'intérêt peuvent être difficiles à percevoir et/ou à atteindre s'ils sont situés, par exemple, à l'autre bout du mur interactif. Si se déplacer physiquement pour s'en rapprocher permet de les rendre plus lisibles, il est toutefois possible qu'ils soient toujours hors de portée du bras. Par ailleurs, un autre utilisateur peut se trouver sur le chemin de la zone à atteindre, le déplacement physique causant alors le dérangement des collaborateurs, et il est probable qu'on ne préférerait tout simplement pas avoir à se déplacer. Ces problèmes sont encore plus marqués lorsque l'on utilise des applications classiques, conçues pour être utilisées sur des écrans de taille normale avec une souris et un clavier. Les applications de bureaux reposent fortement sur des éléments qui ne peuvent que rarement être déplacés : les menus, les barres d'outils, les barres de défilements, les palettes de propriétés. De plus ces éléments sont souvent placés automatiquement sur les bords de l'écran, loin de la position appropriée pour les murs interactifs. L'interaction multi-points incite l'utilisateur à se rapprocher du mur, or à cette échelle, tous ces éléments d'interaction sont hors de portée : la barre de menu est trop haute et les barres d'outils sur les côtés sont trop loin. Des techniques d'interaction spécifiques sont alors nécessaires pour supporter et tirer parti de l'interaction multi-points avec les applications classiques et modernes sur de telles surfaces tactiles.

Une solution classique aux problèmes de portée dans l'interaction directe est d'accéder aux objets distants au travers d'un curseur et d'un contrôle relatif de celui-ci. Par exemple *HybridPointing* [27] introduit le changement de mode de l'interaction directe avec un stylet vers l'interaction indirecte sur très grand écran. L'utilisateur attrape et relâche un widget glissant à côté du stylet pour passer en contrôle relatif et atteindre des objets situés hors de portée. Bartindale *et al.* ont proposé d'émuler une souris classique en posant la main sur une table interactive [4]. Lorsque l'utilisateur pose ses cinq doigts et la base de la paume, une *SurfaceMouse* est créée ; ce geste nécessite une technologie tactile optique pour être reconnu. Les outils tangibles ont également été mis à contribution pour assurer le contrôle relatif d'un curseur, ainsi Vogel et Casiez avaient doté leur dispositif *Conté* [90] d'un mode d'émulation de souris ; ces approches ne sont toutefois pas compatibles avec les surfaces d'entrée verticales. Bien que toutes ces techniques introduisent un changement de mode entre interaction directe et indirecte, elles ne supportent qu'un seul point d'interaction en mode indirect et ne fournissent aucune assistance pour percevoir le contenu distant. Les souris virtuelles *Rizzo* proposées par Vlaming *et al.*

dans leur système de visualisation de données géographiques [87], contrôlent un curseur relatif et ont la particularité d'afficher une vue agrandie (ou réduite) du curseur distant et du contenu environnant : elles combinent pavé tactile virtuel et lentille de visée. Cependant malgré l'usage de gestes multi-doigts pour contrôler les widgets, l'interaction au travers des curseurs reste mono-doigt et demeure en tout point similaire à une souris classique.

Au lieu d'augmenter la portée de l'interaction avec un curseur, d'autres travaux proposent de rapprocher ces objets de l'utilisateur. La technique *Drag&Pop* [6] et le widget *Vacuum* [13] permettent d'amener temporairement les cibles à portée de la main pour interagir avec. De telles techniques requièrent de désigner les cibles d'intérêt : dans le *Drag&Pop*, l'orientation du déplacement initial du curseur indique la direction des cibles potentielles, le système teste alors ces cibles et rapproche celles pour lesquelles il existe une action autorisée. Avec le *Vacuum*, l'utilisateur définit sur le widget le secteur angulaire l'intéressant. Il est ainsi impossible de sélectionner l'espace vide comme cela peut être nécessaire pour dessiner ou éditer une image à distance par exemple. Stoakley *et al.*, dans un contexte de réalité virtuelle en 3D, ont proposé non pas d'interagir sur quelques objets distants mais de créer une copie miniature (*World in Miniature*) [81], et donc à portée de main, de l'ensemble de la scène. Cette approche qui est à rapprocher d'une vue radar pour les environnements 2D, pose d'évidents problèmes de précision lorsque la scène est grande et les objets de la réplique miniature petits. C'est pourquoi, le widget *Frisbee* de Khan *et al.* ne duplique qu'une sous-partie de l'écran pour la mettre à la portée de l'utilisateur [50]. Il peut ainsi interagir avec le *Frisbee* pour déplacer ou agrandir la scène distante et peut directement sélectionner les objets distants au travers de leur représentation locale avec un stylet. Cependant le déplacement de la zone distante est à distinguer du déplacement d'un curseur. Cela requiert de pointer une zone particulière du widget, ce qui peut avoir pour effet de scinder l'interaction en deux temps : la sélection d'objet dans l'espace local et le déplacement de la zone d'interaction distante. De plus le contrôle de la zone est réalisé par une mise en correspondance 1:1, sans fonction de transfert non linéaire donc, ce qui implique un grand nombre de débrayages pour couvrir de grandes distances.

En résumé, bien que les techniques présentées plus haut peuvent faciliter l'interaction, elles souffrent de plusieurs limitations. D'abord, ces techniques sont conçues pour une interaction mono-utilisateur alors que le contexte de mur interactif implique souvent plusieurs collaborateurs. Le support de plusieurs utilisateurs requiert un moyen de créer et détruire dynamiquement des widgets au moins pour chaque utilisateur. Seulement deux techniques, *Frisbee* et *Rizzo*, abordent brièvement l'usage de plusieurs widgets, mais ne définissent pas complètement l'utilisation. Deuxièmement, ces techniques proposent soit d'accéder aux objets distants avec un curseur, soit d'interagir sur des proxys locaux en abandonnant la fonction de transfert ; aucune de ces techniques ne propose la combinaison de ces deux modes d'interaction distante. Enfin la manipulation multi-doigts d'objets distants est absente de toutes ces techniques.

Afin d'enrichir la gestion d'une interface basée sur les curseurs, nous proposons

d'introduire des périphériques logiques d'entrée multi-points et de laisser tout loisir à l'utilisateur de créer et manipuler ces périphériques virtuels. Par la réification du concept du curseur dans nos WallPad, nous désirons compléter la grammaire associée au contrôle des curseurs. Pour concevoir et tester notre technique WallPad, nous avons transformé à moindre coût et sans modification permanente un écran mural déjà existant dans notre laboratoire. Nous reporterons en premier lieu comment nous avons supporté l'interaction multi-points sur un large écran courbe. Nous présenterons ensuite les WallPads, une interface conçue pour adresser les problèmes posés par l'interaction sur mur tactile. Les WallPads introduisent l'accès facile et précis à du contenu distant et supportent aussi bien l'interaction directe et indirecte que la manipulation multi-doigts à distance. Nous détaillerons enfin la manière dont ils peuvent être utilisés pour répondre à divers scénarios d'interaction.

4.2 Tactilisation d'un mur de réalité virtuelle

Inaugurée en 2006, la salle de réalité virtuelle de l'IRCICA (figure 4.1) a été conçue avec un budget d'environ 400 k€ et comprend un large écran mural courbe. Cet écran est composé d'une plaque d'acrylique de 5,96 m × 2,43 m contrainte par un cadre métallique à suivre une courbe avec un arc de 76 cm. La plaque est recouverte sur la face avant d'un diffuseur permettant la projection de l'image. Deux vidéoprojecteurs stéréoscopiques sont placés derrière l'écran dans une salle annexe, la calibration particulière de ces vidéoprojecteurs permet d'afficher une image sans frontière visible sur l'écran d'une résolution de 2392×1050 pixels.



FIGURE 4.1 – Vue d'ensemble de la salle de réalité virtuelle de l'IRCICA.

Pour supporter l'entrée multi-points sur une surface d'une telle dimension, la détection optique semble être la solution la plus appropriée et la plus économique. Les autres technologies que nous avons abordées dans l'état de l'art sont difficilement réalisables sur un écran aussi grand (électroniques) ou manquent

de précision (sonores). Plusieurs technologies optiques ont également rapidement été écartées : les cloisons de la salle ont été refermées autour de l'écran, il est donc inenvisageable de remplacer la plaque d'acrylique par une plaque diffusant la lumière infrarouge telle que celle nécessaire à la technologie DSI (*diffused surface illumination*). Il est impensable de séparer la plaque de son cadre pour y placer des DELs infrarouges, empêchant ainsi l'illumination par la tranche nécessaire à la mise en place de la technologie FTIR. De plus la forme non-plane de l'écran ne permet pas d'installer de manière satisfaisante un plan laser juste au-dessus de la surface (LLP).

Finalement l'illumination diffuse semble être la solution à retenir. L'université de Groningue a utilisé la technologie *rear diffuse illumination* avec un ensemble de 1000 DELs pour réaliser le *Reality Touch Theatre*¹⁶. Nous avons essayé une approche similaire en plaçant des projecteurs infrarouges derrière l'écran, mais les résultats n'ont pas été des plus convaincants. Il est en effet très difficile d'obtenir un éclairage homogène de l'ensemble de l'écran. L'aspect brillant de l'acrylique cause de nombreux reflets et plusieurs zones sont saturées d'infrarouges, la détermination du nombre optimal de projecteurs et de leur position est délicate et ne doit pas entrer en conflit avec les cônes de projection des vidéoprojecteurs.

Toutefois, pendant ces tests préliminaires, nous avons observé que le système d'éclairage de la salle émettait suffisamment de lumière dans le spectre infrarouge pour fournir un éclairage de l'ensemble de l'écran et être perçu par les caméras (figure 4.2). La *front diffuse illumination* est habituellement utilisée sur des surfaces tactiles de dimensions restreintes¹⁷ et avec une source lumineuse directionnelle et unique – les rayons lumineux sont parallèles entre eux – ce qui rend difficile la distinction entre les ombres et les contacts. Dans le cas d'un grand écran, plusieurs sources lumineuses vont éclairer la surface sous plusieurs angles, créant des zones de pénombre et des zones d'ombre totale. Comme dans un stade où les joueurs ont plusieurs ombres dans des directions différentes, la pluralité des sources d'infrarouges permet de discriminer les points de contacts, là où les ombres convergent, des zones de pénombre créées par les objets ou personnes à distance de l'écran (figure 4.3). En analysant les ombres projetées, le système est capable de générer les événements tactiles.

4.2.1 Matériel

Le système d'éclairage de la salle est composé de quinze spots encastrés dans le plafond, quatre appliques murales omnidirectionnelles et quatre appliques au bas des murs (figure 4.4). Toutes ces lampes sont des ampoules halogènes classiques réglées au maximum de leur intensité afin de maximiser l'émission

16. <http://www.rug.nl/science-and-society/centre-for-information-technology/hpcv/nieuws/touchscreen1>

17. Seth Sandler, designer impliqué dans la création de CCV et dans le NUIGroup, utilise ce principe pour construire une surface tactile à bas coût : le MTmini <http://sethsandler.com/multitouch/mtmini>

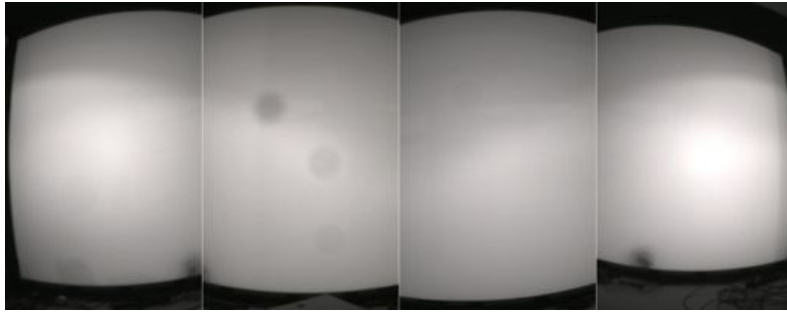


FIGURE 4.2 – Capture brute de l’ensemble de l’écran par les quatre caméras (les taches sombres de la deuxième caméra sont causées par des poussières sur le filtre infrarouge).

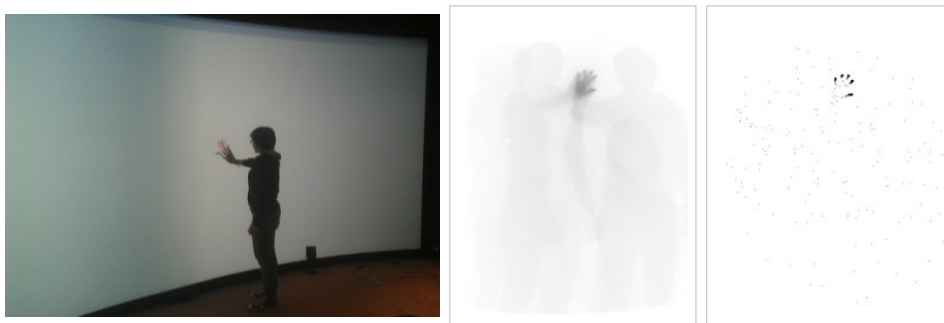


FIGURE 4.3 – À gauche, un utilisateur touche l’écran. Au centre, vue de la caméra après soustraction du fond. À droite, extraction des contacts par le seuillage adaptatif.

d’infrarouges. Il s’agit donc d’un système d’éclairage puissant mais tout à fait classique et courant pour ce type de salle.

Nous avons utilisé quatre caméras IDS UI-1220LE équipées de lentilles légèrement fisheye Tamron 13VM2811ASIR et de filtres bloquant la lumière visible (filtres passe-haut, fréquence de coupure à 750 nm). Le coût total pour chaque caméra était d’environ 340 €.

Ces caméras sont positionnées en mode portrait derrière l’écran sur des structures construites spécialement à cet usage, en dehors du volume de projection de vidéoprojecteurs (figure 4.5), découpant ainsi la surface d’entrée en quatre rectangles disposés côte à côte (figure 4.2). Les zones filmées par les caméras se chevauchent sur 30 cm en moyenne. Les quatre caméras sont connectées à un ordinateur (2,66 GHz Intel Xeon, 4 Go RAM, Windows 7) exécutant notre logiciel en charge du traitement des données. Les quatre images de 752×480 pixels sont reçues et traitées 60 fois par seconde, les contacts sont envoyés à la machine responsable de l’affichage par UDP avec le protocole TUIO.

4.2.2 Traitement d’image

Le système acquiert des images brutes similaires à celle présentée dans la figure 4.2, le logiciel doit en extraire les points de contacts et écarter les ombres.

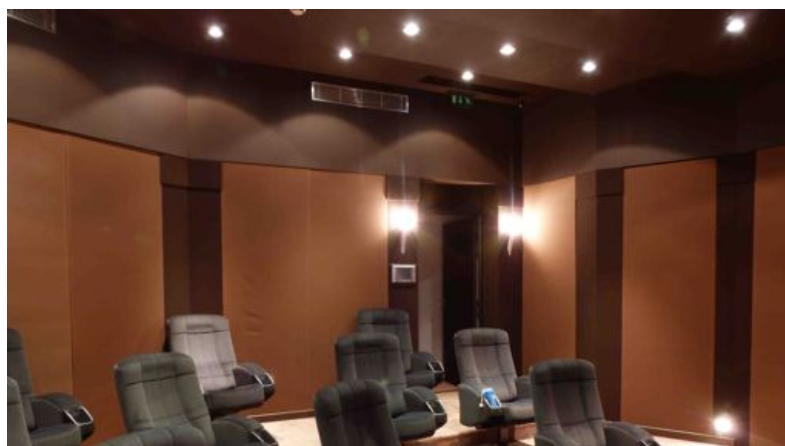


FIGURE 4.4 – Éclairage de la salle.

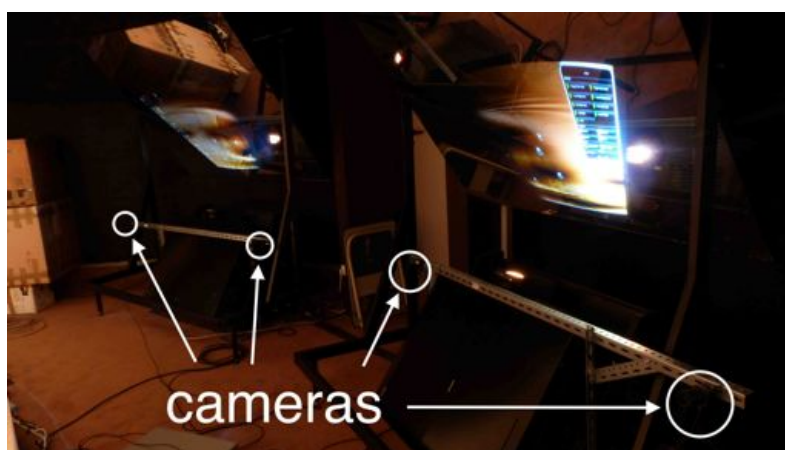


FIGURE 4.5 – Vidéoprojecteurs et caméras.

Les paramètres des caméras (temps d'exposition, gains...) sont fixés avec les pilotes des caméras de manière à réduire les zones saturées et à avoir un contraste suffisant sur l'ensemble de l'image. Le traitement d'image est ensuite réalisé en utilisant la librairie OpenCV, la distorsion en barillet due à la lentille (effet *fish-eye*) est corrigée en calibrant les caméras sur un échiquier en noir et blanc. Une image de l'arrière plan est enregistrée au lancement de l'application, quand la salle est vide (de personnes); cette image est soustraite à l'image courante, cela a pour effet de supprimer le cadre de l'écran, les ombres de l'éclairage et ne laisse que les ombres des utilisateurs (figure 4.3 au centre). Les zones correspondantes aux contacts sont isolées grâce à un filtre adaptatif avec un voisinage gaussien assez large – de 13×13 voisins –, les teintes claires des doigts et des membres sont supprimées, la paume de la main est rejetée grâce à l'effet de filtre passe-haut du filtre gaussien (figure 4.3 à droite). À partir des zones sombres restantes, les contacts sont extraits en filtrant selon la taille (bornes minimale et maximale) et selon la forme qui ne doit pas être trop allongée. Les contacts sont ensuite géométriquement projetés dans le repère de l'écran pour corriger la déformation due à l'écran. Le passage dans le repère

écran, qui est commun à toutes les caméras, permet de regrouper l'ensemble des contacts, et de supprimer les doublons grâce à un seuil de distance. Les doublons interviennent dans les zones de recouvrement des caméras qui vont alors chacune filmer le même doigt. Ces contacts sont finalement envoyés sur le réseau en utilisant le protocole TUIO.

La correction géométrique utilisée lors de la projection compense la forte déformation non-linéaire causée par la forme courbe de l'écran et la position proche du sol des caméras. En effet, la scène est vue du dessous afin de ne pas entrer dans le cône de projection des vidéoprojecteurs. La courbe de l'écran est alors approchée par 10×6 rectangles plats et une projection homographique classique (ou interpolation linéaire) est appliquée dans chacun de ces rectangles afin de définir l'application projective des caméras vers l'écran. Une étape de calibration est donc nécessaire lors de la mise en place du système pour enregistrer les coordonnées de ces rectangles dans les repères des caméras. La grille des rectangles est progressivement affichée, une personne doit alors pointer chacun des points de la grille (figure 4.6). Les caméras détectent le contact et le système enregistre la correspondance de coordonnées et la(les) caméra(s) associée(s).



FIGURE 4.6 – Calibration des projections.

Notre système peut détecter plus de 50 points simultanément avec une précision variant de 3 à 5 mm selon la hauteur du toucher. Nous avons mesuré grâce à un dispositif externe une latence globale de 140 ms, dont 30 ms pour le traitement d'image. Avec les réglages de caméras et seuils que nous avons utilisés, les doigts en contact imminents sont détectés à un peu moins de 1 cm au-dessus de la surface, ce qui est difficilement perceptible quand on touche un objet et satisfait grandement les utilisateurs. Toutefois ce survol peut être gênant pour effectuer des clics ou des "tap" car il faut reculer suffisamment le doigt. Le résultat de la calibration est satisfaisant : les utilisateurs ne perçoivent pas d'écart entre leur toucher et le contact détecté. Les résultats obtenus sont réellement surprenants au regard du peu de matériel nécessaire et du peu de modifications apportées à la salle : 4 caméras bon marché, deux petites structures de 60 cm de haut, pour un coût total de matériel d'environ 0,5%

du budget de la salle d'origine.

4.3 WallPad

Nous allons maintenant décrire le widget WallPad, de son utilisation comme un simple pavé tactile jusqu'à l'interaction plus sophistiquée sur du contenu distant, en incluant la gestion du cycle de vie des WallPad : à savoir la création, le déplacement, le redimensionnement et la destruction. Une vidéo a été réalisée pour illustrer dynamiquement les différentes façons d'utiliser WallPad et montre quelques scénarios d'utilisation. Cette présentation est disponible sur internet à l'adresse suivante : <http://bit.ly/1crHbc8>.

4.3.1 Un pavé tactile

En ayant toujours dans l'optique de faciliter l'apprentissage des utilisateurs novices, le WallPad a été conçu pour fonctionner par défaut comme un pavé tactile classique. Sa représentation visuelle est similaire à un pavé tactile : il comprend une surface tactile rectangulaire qui surmonte un bouton (figure 4.7).



FIGURE 4.7 – WallPad dans une application de dessin. Le widget comporte une zone tactile rectangulaire (a) avec un bouton en dessous (b). Les doigts posés dans la zone tactile assurent un contrôle relatif du curseur (c). La région autour du curseur est affichée dans la zone centrale. Une couleur unique est attribuée à chaque couple WallPad (le bouton) et curseur afin de les identifier.

Les WallPads apparaissent par dessus les applications existantes, l'ensemble du visuel est semi-transparent afin de réduire les occultations et l'encombrement visuel. Les gestes de glisser à l'intérieur du rectangle sensible contrôlent de manière relative les déplacements d'un curseur distant en forme classique de flèche. Ce contrôle a lieu au travers d'une fonction de transfert non-linéaire similaire à celle de OS X pour les pavés tactiles, déjà utilisée précédemment [20]. Les déplacements du curseur sont réduits lorsque le doigt se déplace lentement. À l'inverse les déplacements du curseur sont multipliés pour des vitesses de

doigts plus importantes. Pour réduire les débrayages nécessaires aux longs déplacements, les doigts sortant de la zone sensitive continuent à être suivis et contrôlent toujours le curseur. La sélection (clic) est réalisée avec le bouton qui permet de cliquer et de maintenir la sélection. Lorsque le curseur est dans un état de sélection, il est entouré d'un disque coloré afin de fournir un retour visuel (figure 4.8 a).

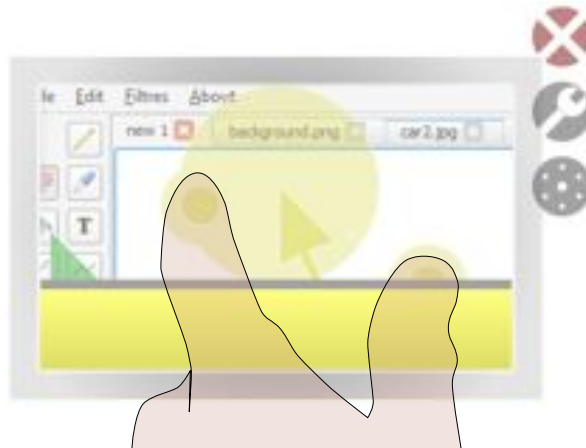


FIGURE 4.8 – Capture d'écran d'un WallPad en mode d'interaction directe délocalisé : la vue centrale est opaque, l'utilisateur a engagé deux doigts.

Dans le contexte des murs interactifs, les objets distants peuvent être difficiles à percevoir, l'angle de vue ou la distance peut gêner la distinction d'éléments et l'interaction avec ceux-ci [12]. Afin de résoudre ces problèmes de lisibilité de l'information, et grâce aux capacités dynamiques des périphériques virtuels (*soft-devices*) [67], la zone centrale présente une vue déportée à l'échelle de la région entourant le curseur. Cet affichage local permet de positionner précisément le curseur pour sélectionner de petits objets ou lire du texte.

4.3.2 Interaction directe délocalisée

Les WallPads autorisent également une manipulation directe des objets affichés dans la vue locale : l'*interaction directe délocalisée*. Les taps et double-taps sur les objets de la vue locale sont redirigés vers l'emplacement distant. L'utilisateur peut ainsi interagir directement avec les boutons d'un menu par exemple.

Le quasi-mode de manipulation directe multi-doigts est déclenché en maintenant un doigt immobile pendant 0,5 s ou en engageant plusieurs doigts dans la zone centrale. La vue semi-transparente devient alors opaque pour informer l'utilisateur du passage dans ce quasi-mode d'interaction directe délocalisée et suggère ainsi que les objets peuvent être manipulés directement et avec plusieurs doigts (figure 4.8). Les doigts sont alors affichés autour du curseur distant comme des petits cercles colorés (similaires au visuel des cursels du chapitre précédent), cette affichage indique à l'utilisateur que les doigts sont bien reconnus et permet en plus d'informer les autres utilisateurs éventuels que

les objets sont en train d'être manipulés par quelqu'un. Les WallPads restent dans ce quasi-mode tant qu'il y a un doigt dans la zone centrale et pendant 0,5 s après la fin du dernier contact, temps durant lequel l'utilisateur peut débrayer sans quitter le mode.

4.4 Cycle de vie des WallPads

L'aspect virtuel de nos périphériques WallPads autorise à en créer autant que nécessaire ainsi qu'à les doter de propriétés et caractéristiques modifiables et personnalisables.

4.4.1 Création

L'utilisateur peut invoquer plusieurs WallPads pour adapter l'usage à la réalisation de la tâche. La création doit reposer sur un geste non-ambigu et compatible avec les commandes courantes déjà existantes afin de prévenir toute apparition inopinée et non souhaitée. Après avoir considéré plusieurs alternatives, nous en sommes arrivés à l'idée d'extraire le curseur depuis un doigt, de manière à "déposer un curseur sur l'écran". La description du geste correspondant est la suivante :

1. l'utilisateur pose un premier doigt sur l'écran tactile et reste immobile ;
2. un doigt de la seconde main glisse le long de ce premier doigt en direction de l'écran ;
3. le second doigt entre en contact avec l'écran et continue son mouvement, cet enchaînement est reconnu par le système comme le début de mouvement de création : le retour visuel est aussitôt affiché sous la forme de deux disques sous chaque doigt et d'une ligne indiquant le seuil de distance à dépasser pour valider la création ;
4. le second doigt dépasse le seuil de distance, le disque associé au premier doigt quitte ce doigt et rejoint le second doigt pour indiquer à l'utilisateur que le geste de création est valide (*feed-forward*) ;
5. le curseur sera créé à l'emplacement du retrait du second doigt.

L'ensemble de ces étapes est illustré dans la figure 4.9. À tout moment, le geste peut être annulé en ramenant le second doigt en deçà du seuil de distance ou en retirant ou déplaçant le premier doigt. Pendant les différentes phases de tests et lors de son utilisation courante, nous n'avons jamais eu à déplorer un conflit avec d'autres gestes aboutissant à une création non désirée. Ce geste peut être facilement décliné pour former une famille de nouvelles commandes : par exemple, un clavier virtuel peut être créé en glissant deux doigts au lieu d'un seul.

Une couleur unique est attribuée à chaque couple WallPad (coloration du bouton) et curseur afin d'identifier rapidement les couples en présence de plusieurs WallPads. Le geste de création permet à la fois de définir l'endroit où l'utilisateur souhaite créer le périphérique mais aussi d'indiquer où placer le curseur

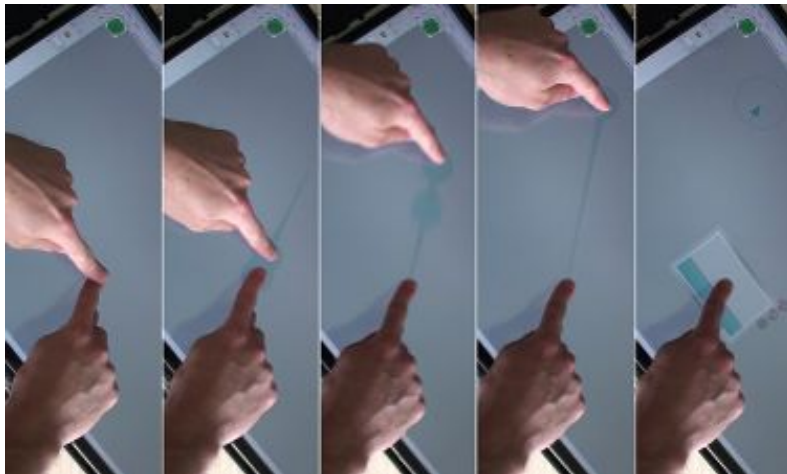


FIGURE 4.9 – Création d’un WallPad, capturé ici sur un écran multi-points 3M pour plus de visibilité. De gauche à droite : un doigt glisse le long d’un autre doigt qui reste immobile et continue le mouvement sur l’écran jusqu’à dépasser un seuil pour créer un WallPad. Le *feedback* et le *feed-forward* signale la reconnaissance du geste et informe de la validité de la création et du lieu d’apparition du curseur.

dans un premier temps : directement sur une cible ou juste dans la direction de la cible. Dans un contexte de surface interactive horizontale (table), l’orientation du nouveau périphérique, et donc de l’utilisateur, peut être estimée par la droite joignant les deux doigts. Ainsi les WallPads pourraient être créés directement dans le bon sens quel que soit la position de l’utilisateur.

4.4.2 Gestion dynamique et reconfigurable

Une fois créés, les WallPads arborent trois icônes sur le coin supérieur droit, ces boutons permettent de modifier les propriétés de chaque WallPad.

Les WallPads peuvent être déplacés et redimensionnés après avoir activé le mode de réglage. En pressant sur le bouton « clé à molette », l’utilisateur peut afficher les bordures autour du WallPad (figure 4.10 a). En posant un doigt sur la bordure, on peut déplacer le widget pour accéder au contenu situé en dessous ou simplement utiliser le widget ailleurs. En sélectionnant deux côtés opposés, l’utilisateur peut modifier la taille du widget. Le redimensionnement peut être particulièrement appréciable pour adapter la taille de la zone centrale au contenu d’un menu ou d’une barre d’outils et interagir avec une copie locale de ceux-ci. Cela permet aussi d’agrandir la zone tactile pour accéder à un plus grand espace de travail en interaction directe délocalisée.

La taille des WallPads peut également être réduite grâce à un bouton situé juste en dessous du bouton « clé à molette ». Le WallPad minimisé apparaît comme un petit rectangle de 20 mm de largeur adossé au bouton de minimisation (figure 4.10 b), cette taille a été montrée suffisante pour être facilement atteinte [38, 80, 89, 94, 43]. Une fois minimisé, le curseur ne peut plus être déplacé : les gestes de glisser sur le widget servent à déplacer le WallPad afin



FIGURE 4.10 – (a) Affichage étendu d’un WallPad, les bordures sont agrandies après avoir pressé le bouton « clé à molette ». Les bordures permettent de déplacer et redimensionner le widget. (b) Vue réduite de WallPad avec le bouton de minimisation.

de pouvoir le bouger rapidement au cas où celui-ci se trouverait sur une cible. Toutefois les taps et double-taps continuent à être redirigés vers le curseur distant. La minimisation des WallPads peut être utile pour réduire au maximum l’espace visuel occupé par un WallPad sans le détruire, ou bien pour accéder occasionnellement à un bouton situé un peu loin du lieu général d’interaction comme les boutons « Défaire/Refaire » ou « Copier/Coller ». Finalement un WallPad peut être détruit en sélectionnant le bouton en forme de croix rouge.

4.5 Implémentation

Les WallPads s’affichent en superposition par rapport aux fenêtres applicatives (figure 4.11). Ces applications ne subissent que peu ou pas de modifications, notre approche peut s’intégrer comme une couche composée d’outils pour interagir avec tous types d’applications. Si le système d’exploitation ou l’application comporte des événements multi-points, l’utilisateur pourra poser plusieurs doigts dans la zone tactile et réaliser des manipulations multi-doigts, sinon les WallPads généreront des événements souris en direction des applications.

Lorsqu’un contact débute, notre couche intercepte l’événement, si celui-ci a lieu dans un des WallPads ou sur un curseur, il est utilisé par le WallPad concerné comme nous avons pu le voir dans la section précédente. Dans le cas contraire l’événement est redistribué aux applications. Les curseurs sont eux-mêmes générateurs d’événements envoyés à la couche applicative lorsque l’utilisateur clique, ou manipule directement des objets délocalisés.

Les WallPads apparaissent comme des objets virtuels, l’utilisateur agit dessus comme il le ferait avec d’autres objets autorisant la manipulation multi-doigts. Les widgets peuvent être déplacés, agrandis, réduits ou détruits, les curseurs

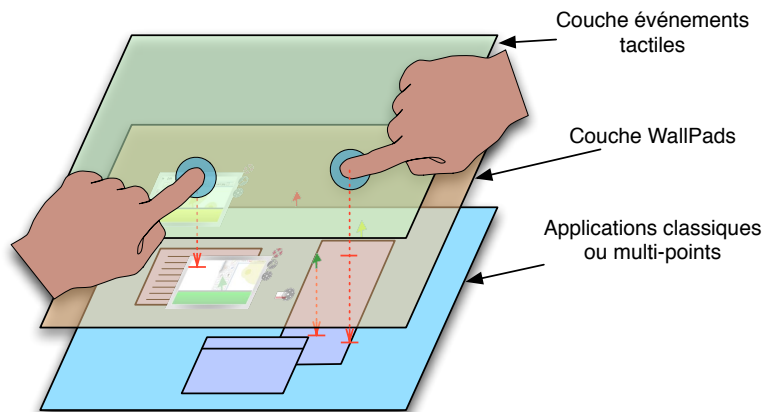


FIGURE 4.11 – Interception des événements tactiles : les WallPads composent une couche intercalée entre l'entrée tactile et les applications.

sont aussi des objets qui peuvent être déplacés en les sélectionnant directement dans le cas où ils se trouveraient au-dessus d'informations. Mais les WallPads sont également des concepts de plus haut niveau permettant d'interagir avec le contenu affiché et générant des événements.

4.6 Cas d'usage

La conception des WallPads a été raffinée et corrigée par plusieurs tests informels pour arriver à un outil facile à prendre en main tout en regroupant plusieurs modalités d'interaction. La version décrite ici ne nécessite que peu d'apprentissage, une fois que l'utilisateur a appris comment créer un WallPad et sait passer dans le mode d'interaction directe délocalisée, il peut facilement découvrir les autres fonctionnalités.

Ces fonctions détaillées plus haut permettent des usages plus complexes et la combinaison de plusieurs WallPad autorise des scénarios d'utilisation intéressants (figure 4.12). Voyons comment WallPad peut être utilisé pour répondre à divers problèmes au travers des quelques scénarios ou contextes suivants.

4.6.1 Accès rapide à des fonctions distantes

Les utilisateurs peuvent créer dynamiquement des curseurs là où cela semble nécessaire, chacun de ces curseurs est contrôlé à distance et indépendamment par un widget. Les WallPads sont permanents (jusqu'à leur destruction) et librement déplaçables, on peut donc créer plusieurs WallPads et placer les curseurs à divers endroits de l'écran, proches de fonctionnalités disséminées dans l'espace d'affichage. Si le curseur n'a plus à être déplacé, car il est positionné au-dessus d'un bouton par exemple, le WallPad peut même être minimisé afin de réduire l'empreinte visuelle.

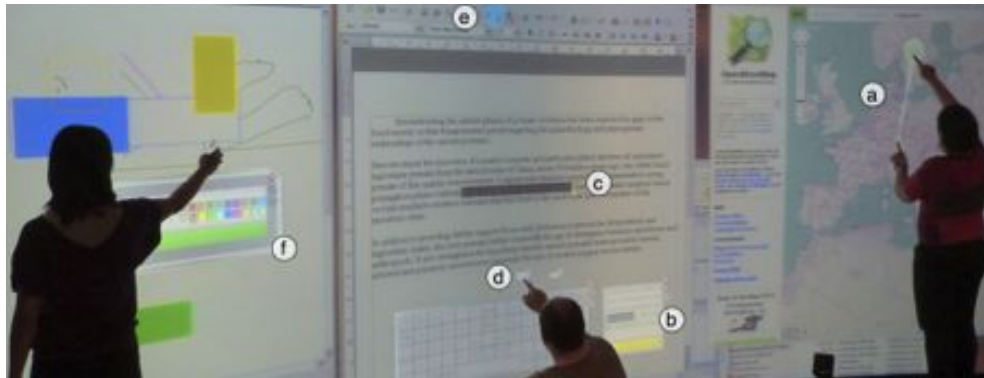


FIGURE 4.12 – Exemples d'utilisation des WallPads. Un utilisateur crée un WallPad (a) pour accéder à une barre d'outils hors de portée. Un autre utilisateur utilise un WallPad comme un pavé tactile (b) pour sélectionner du texte (c), et a positionné deux WallPads minimisés (d) sur les boutons Copier/Coller régulièrement utilisés (e). Un troisième utilisateur rapproche une palette de couleur grâce à un WallPad (f).

En changeant de périphérique virtuel, l'utilisateur passe d'un curseur à l'autre et change la position de l'interaction. Au-delà de la vitesse et de la précision apportées par la fonction de transfert, il peut se déplacer (au travers du changement de curseur) instantanément à un emplacement exact, défini au préalable. La création de « signets » ou de raccourcis s'apparente aux fonctionnalités proposées par UIMarks [21] où les auteurs proposent de marquer les cibles et actions récurrentes et où l'utilisateur peut ensuite accéder rapidement à ces marques par des raccourcis clavier ou créer des proxys aimantés attirant le curseur de la souris. Notre approche n'ayant pas été conçue pour enregistrer des actions associées au changement de curseur propose moins de possibilités (programmation d'actions notamment), mais elle permet cependant de réintroduire ce concept de signets au sein d'une proposition plus générique. L'utilisateur expert peut ainsi réduire drastiquement la distance à parcourir entre les éléments d'interface qu'il utilise selon la tâche et ses habitudes.

4.6.2 Personnalisation de l'espace de travail

La personnalisation possible de l'espace de travail va encore un peu plus loin grâce à l'interaction directe délocalisée. Les WallPads peuvent servir à recopier des fragments de l'écran et ainsi réorganiser une interface et sélectionner des outils. Dans la plupart des applications, l'utilisateur va agir en priorité sur ce qui est affiché au centre. La zone centrale contient l'objet d'intérêt et les outils permettant de modifier cet objet sont accessibles sur les côtés de l'application dans des menus, des barres d'outils, des palettes, des propriétés [10].

Par exemple, dans une application de peinture ou de retouche d'image, l'utilisateur doit se déplacer constamment pour changer d'outils d'édition ou de couleurs, les WallPads permettent de créer des proxys à portée de main. L'utilisateur peut choisir de copier la barre d'outils et une palette de couleurs en plaçant les curseurs de deux WallPads au-dessus. Il ne lui reste plus qu'à redimensionner les périphériques virtuels aux dimensions de chaque outil. Chaque

WallPad affiche une vue distante de la zone autour du curseur, l'utilisateur peut directement interagir avec le contenu distant sans parcourir la distance le séparant du curseur.

La spécification d'une région d'intérêt pour recomposer une interface est similaire à ce qui était offert, dans un contexte de bureau classique, par les systèmes *WinCuts* [85] et *UIFaçades* [82]. Dans ces travaux, les utilisateurs peuvent définir une région avec la souris et des raccourcis clavier afin de copier des sous-parties de fenêtres applicatives ou de menus. Ces copies peuvent ensuite être agencées afin de créer une nouvelle façade de présentation et d'interaction. Les fenêtres originales peuvent être cachées grâce à un système de rendu hors de l'écran assurant la mise à jour des vues. Les WallPads permettent également de réaliser une composition d'interfaces sans toutefois autoriser à cacher les objets distants. Notre approche transforme l'espace en créant des raccourcis mais ne copie pas réellement les objets et reste flexible en pouvant changer la façon dont est utilisé un WallPad à tout moment.

4.6.3 Aspects collaboratifs

Les murs interactifs permettent à plusieurs utilisateurs d'agir simultanément, soit pour effectuer une tâche commune en collaborant, soit pour réaliser des actions indépendantes des autres utilisateurs. Les utilisateurs vont ainsi par moment être répartis sur l'écran, et à d'autres moments peuvent vouloir accéder à une zone commune simultanément. Un utilisateur peut entrer dans l'espace de travail d'un autre utilisateur pour un instant : sélectionner un objet, donner ou prendre un objet, atteindre un menu situé à proximité ou de l'autre côté d'une personne.

Ces différentes configurations nécessitent de se déplacer, de communiquer avec les autres usagers, parfois de les déranger ou les contourner. L'accès simultané à une partie de l'écran requiert une interaction à tour de rôle pour ne pas se gêner mutuellement.

Avec les WallPads, plusieurs utilisateurs peuvent chacun créer plusieurs périphériques virtuels et ainsi avoir une collection privée de curseurs. L'interaction à distance, que ce soit avec le curseur ou en interaction directe délocalisée autorise l'accès à la totalité de l'écran en restant dans un espace restreint. Les collaborateurs peuvent se partager l'espace de l'écran et interagir sans se gêner physiquement. Les WallPads restent à portée de main, la couleur des curseurs et les vues locales de l'environnement des curseurs assurent aux utilisateurs de retrouver facilement leurs curseurs parmi ceux existants.

Le cercle entourant le curseur lorsqu'il est en état de sélection (par rapport à l'état de survol) et l'affichage des doigts autour des curseurs en mode d'interaction délocalisé expose à tout le monde les manipulations ayant cours. Les utilisateurs peuvent donc prendre conscience de ce que les collaborateurs font et ne seront pas surpris lors du déplacement d'objets ou lors du changement de propriétés. Pour collaborer dans une même zone, il est évidemment toujours nécessaire de s'appuyer sur la communication inter-personnelle pour

se synchroniser, mais la vision des actions de chaque utilisateur peut aider à l'accomplissement de la tâche. Si un curseur entre dans l'espace péri-personnel d'un autre utilisateur, celui-ci va immédiatement le percevoir et peut deviner les moments où le curseur réalise une action. S'il ne désire pas être gêné par d'autres curseurs, il peut les repousser en les sélectionnant directement et en les écartant. Il peut donc joindre la parole au geste pour signifier au collaborateur de partir.

4.6.4 Compatibilité et apports des WallPads avec les applications régulières

Nous avons conçu les WallPads comme des périphériques autorisant la manipulation multi-points. S'il est nécessaire de disposer d'un écran multi-points, l'utilisation des WallPads peut aussi être profitable aux applications et systèmes mono-points.

En effet, en mode curseur, il n'y a qu'un seul point d'interaction envoyé à la couche applicative. L'utilisateur peut manipuler le périphérique avec plusieurs doigts, par exemple pour réaliser un *drag* avec le curseur, mais finalement ces gestes sont intégrés pour créer un événement souris mono-point avec un état de sélection. Lorsque l'utilisateur va agir dans une barre d'outils en mode d'interaction directe délocalisée, il ne sélectionne qu'un outil à la fois. Le toucher est redirigé à l'emplacement absolu autour du curseur mais le centre du curseur n'est pas activé, il s'agit donc bien, encore une fois, d'une entrée mono-point vers les applications.

L'usage des WallPads avec des applications classiques n'est donc pas limitée. Bien au contraire les avantages apportés par notre approche améliorent l'utilisabilité des anciennes applications. L'utilisateur peut créer des raccourcis et accéder à distance à diverses parties de l'écran. La création de plusieurs WallPads reste d'actualité, même dans ce contexte qui apparaît comme peu favorable. La multiplicité des périphériques virtuels autorise à passer d'un lieu à l'autre toujours aussi facilement et tant que l'utilisation des widgets est séquentielle, il n'y a pas de conflit. Notre approche rend ainsi un système mono-point compatible avec l'interaction multi-points séquentielle mais pas multi-doigts.

Ceci est la principale limitation, les applications ne comprenant pas l'interaction multi-doigts, la manipulation avec plusieurs doigts via la zone tactile est rendue inopérante. De même, des conflits peuvent surgir lorsqu'on interagit avec plusieurs WallPads en même temps. Le support multi-utilisateurs est donc limité si le système n'autorise pas la création de plusieurs curseurs souris.

4.6.5 Au delà du mur

Le contexte d'utilisation dans lequel nous nous sommes placés concerne les murs interactifs, ceux-ci offrent en effet une très grande surface d'affichage

et invitent plusieurs utilisateurs à agir simultanément. Toutefois l'approche proposée par les WallPads pourrait se prêter à d'autres contextes, une table interactive revêt les mêmes aspects de collaboration et possède une taille d'affichage respectable. Les utilisateurs sont répartis autour de la table, ayant chacun leur orientation propre ; nous avons vu dans la section 4.4.1 que le geste de création peut définir l'orientation des périphériques. Chaque utilisateur créant ses périphériques peut accéder à l'ensemble de la table avec des outils correctement orientés dans la vue délocalisée.

Les environnements de bureau digitaux peuvent également être propices à l'utilisation des WallPads : composés de surfaces d'affichage horizontales et verticales, mixant écrans normaux et tactiles, ces espaces de travail proposent une grande richesse de configurations. Récemment Weiss *et al.* ont conçu un bureau composé d'un écran multi-points courbe en forme de « L » [95] créant une transition sans frontière entre la table et l'écran. Bi *et al.* avaient proposé et étudié des interactions autour de leur dispositif couplant un écran vertical et une table interactive utilisée comme bureau [15]. Avec de tels agencements, les WallPads pourraient être mis à profit pour accéder au contenu des écrans verticaux depuis l'espace horizontal du bureau. Le contenu affiché dans des surfaces non tactiles pourrait tout de même être manipulé en délocalisé. L'utilisateur pourrait composer à l'envie son interface en sélectionnant les outils nécessaires et en créant des proxys dans la surface horizontale, combinant ainsi plusieurs WallPads, la souris et le clavier.

4.7 Discussion

WallPad est une approche unique apportant plusieurs types de solutions pour répondre à des cas d'usages variés. Leur conception permet de les utiliser de diverses manières selon les besoins des utilisateurs. Les WallPads réduisent non seulement la fatigue et les déplacements pour accéder à des objets distants, mais ils améliorent également la précision et la visibilité. De plus l'interaction délocalisée et les caractéristiques dynamiques telles que le redimensionnement et l'ajustement de la position permettent aux utilisateurs experts de recomposer l'interface d'un logiciel en mettant à portée de main les éléments qu'ils utilisent, y compris pour des applications non conçues pour les très grands écrans.

Cette approche peut être généralisée et déclinée autrement qu'au travers des WallPads. En effet les WallPads sont une implémentation du concept de périphérique logique de contrôle. Les périphériques logiques sont des interfaces virtuelles permettant, par exemple, de contrôler un curseur, un espace de travail, un objet [37], ou encore un bouton [19], chacun des widgets est une sous-partie de l'espace d'entrée global. La réintroduction des curseurs rend l'interaction précise, rapide et distante, tandis que la création de plusieurs sous-espaces permet d'utiliser plusieurs curseurs pour accéder rapidement à différents endroits de manière récurrente. L'utilisateur ne va pas forcément interagir avec plusieurs objets simultanément mais il peut passer séquentiellement d'un endroit

à l'autre. Lui proposer de créer plus de deux couples curseur/contrôleur qu'il peut placer à ces différents endroits peut alors s'avérer pertinent.

Le découpage de l'espace d'entrée en plusieurs périphériques avait déjà été proposé par Buxton dès 1985 sur une surface tactile indirecte [19]. En plaçant des caches en carton par dessus une surface tactile (figure 4.13), il créait des *sliders* et des boutons pour contrôler plusieurs propriétés. Son approche, reposant sur des caches physiques, était statique et la disposition des périphériques logiques ne pouvait être modifiée qu'en changeant de carton.

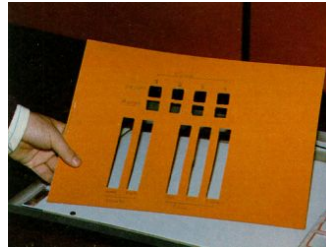


FIGURE 4.13 – Système de cache en carton sur une surface tactile. Les trous dans le carton créent des périphériques logiques, ici des *sliders* et des boutons. Les bords des ouvertures fournissent un retour tactile facilitant l'utilisation de plusieurs éléments simultanément (issu de [19]).

Le développement des interfaces graphiques, et en particulier des écrans tactiles, peut permettre de rendre ces périphériques logiques dynamiques. En 1976, Knowlton projette un affichage par dessus une matrice de touches grâce à un miroir sans tain [53]. De cette manière les fonctionnalités accessibles peuvent être modifiées dynamiquement et ainsi dépasser le caractère statique des boutons physiques : il superpose les fonctions accessibles sur les boutons et l'affichage des fonctions disparaît lorsqu'elle ne sont pas disponibles. Il peut aussi grouper les boutons en affichant des cadres ou changer l'affectation des boutons pour adapter l'interface au contexte d'usage. Nakatani et Rohrlich formaliseront cette approche avec les *Soft Machines* [67]. Ils détaillent les caractéristiques intéressantes des contrôles physiques (*hard controls*). Par exemple dans une voiture, la disposition des commandes des vitres électriques permet d'y associer la vitre commandée. Le type d'actionneur est également une information importante : un interrupteur ne comporte que deux états, on peut donc allumer ou éteindre, à l'opposé un variateur rotatif possède une suite continue de positions renseignant donc sur la possibilité d'une commande continue. Les auteurs les comparent ensuite aux caractéristiques des contrôles virtuels (*soft controls*) qui sont flexibles, modulables, adaptables.

De ces travaux fondateurs, nous pouvons voir les interactions tangibles et tactiles comme héritières de ces concepts. Ce sont, en effet, dans ces situations que l'utilisateur va interagir avec une collection d'outils.

Nous pourrions concevoir une instanciation minimaliste des périphériques logiques de contrôle pour de plus petits écrans. Le pavé tactile virtuel garderait la taille minimale de 15 mm, les gestes de glisser et de tap à un doigt contrôlant respectivement le déplacement du curseur et la sélection, les gestes à

deux doigts contrôlant respectivement le déplacement du widget et la création/suppression. Cette technique minimaliste garderait la précision et la portée introduite par les curseurs, et permettrait à l’instar de WallPad et UI-Marks [21] de marquer des éléments pour y accéder rapidement.

4.8 Conclusion

Ce chapitre a présenté comment nous avons apporté le support multi-points à un écran mural non-conçu à cet effet. La technologie de *front diffuse illumination* habituellement réservée aux surfaces de petites dimensions a pu être employée sur une très grande surface grâce aux sources infrarouge multiples. Celles-ci ont créé des zones d’ombres et de pénombre, permettant de distinguer les corps proches de l’écran des doigts en contact. Cette solution doit pouvoir facilement s’appliquer à d’autres écrans basés sur la vidéoprojection arrière et se place comme une alternative efficace et bon marché.

Considérant ensuite les techniques facilitant l’interaction sur très grand écran, nous avons relevé le manque de support d’une interaction à la fois directe et indirecte, et tirant partie du caractère multi-points du périphérique. Nous avons alors introduit WallPad une technique permettant, au travers de la réintroduction des curseurs, de supporter une interaction riche et puissante. Nous avons conçu un geste de création non-ambigu permettant de désigner à la fois le lieu de création du curseur et du widget. Les WallPads peuvent être utilisés comme des pavés tactiles traditionnels pour passer de l’interaction directe à l’interaction indirecte. La vue locale de l’environnement du curseur améliore la visibilité des éléments distants et autorise une sélection précise, mais elle introduit aussi la possibilité d’agir directement sur les objets distants en plaçant plusieurs doigts dessus.

Les WallPads supportent une interaction multi-utilisateurs en permettant d’accéder à toute la surface sans déranger les autres. De plus, l’aspect dynamique des WallPads est mis à profit pour proposer la recomposition d’une interface en amenant à portée de main des menus et boîtes à outils situés trop loin. Enfin, cette approche n’est pas spécifique à notre contexte de grand écran et des études pourraient valider leur utilisation sur d’autres configurations, telles que tables interactives et bureaux digitaux.

Conclusion

« reality-based is typically not sufficient. A useful interface will rarely entirely mimic the real world, but will necessarily include some “unrealistic” or artificial features and commands. In fact much of the power of using computers comes from this “multiplier” effect, the ability to abstract from or go beyond a precise imitation of the real world. »

—Robert Jacob in [46]

Dans cette thèse, nous avons étudié l’interaction multi-points indirecte pour manipuler des données présentées sur de grands écrans. En étudiant l’interaction absolue multi-points, nous avons noté que ce type d’interaction repose sur les capacités proprioceptives pour accéder facilement et rapidement à des cibles situées dans un espace physique. Pourtant cette immédiateté de sélection n’est pas sans contrepartie. En effet les problèmes de précision, d’occultation et de portée sont omniprésents. En passant en revue les travaux de l’état de l’art, nous avons mis en lumière que le concept de curseur est central à l’ensemble des réponses proposées. L’introduction d’une indirection entre les doigts et les objets manipulés apporte des solutions à ces problèmes. Cependant, les caractères multi-doigts et multi-points de l’interaction disparaissent souvent au profit de la précision.

Dans un premier temps, nous avons évalué la précision accessible pour une tâche de pointage absolu indirect sans information de survol pour mieux comprendre les cas où ce type d’interaction peut être employé. Au travers de deux expériences, nous avons mesuré l’effet des conditions d’utilisation et de la taille du périphérique d’entrée puis des ratios de taille entre l’espace d’affichage et l’espace d’entrée, ainsi que l’échelle d’affichage. Les résultats de la première expérience ont montré que la vision de l’espace d’entrée améliore les performances même si rien n’y est affiché. De plus, la taille minimale de cible accessible dépend de la taille de la surface d’entrée. Nos résultats tendent vers l’hypothèse d’une augmentation linéaire de la taille minimale avec la taille du périphérique. Notre seconde expérience a permis de démontrer que le facteur d’échelle de l’affichage n’a pas d’influence sur les performances, mais qu’il est préférable de conserver un rapport de cadre similaire entre les espaces d’entrée et d’affichage. Nous avons également établi quatre recommandations portant sur la conception des interfaces et le placement des cibles en fonction des conditions d’utilisation.

Nous avons ensuite conçu une technique d’interaction mettant à profit cette sélection sans survol. En reprenant les principes des curseurs multi-points au-

torisant l'interaction distante, nous avons donné la possibilité aux utilisateurs de contrôler plusieurs curseurs relatifs pour accéder à un grand écran. L'utilisation de plusieurs curseurs depuis un seul périphérique tactile de taille modeste permet d'interagir à différents endroits avec les deux mains. Nous avons proposé une méthode de sélection des curseurs basée sur leur position absolue ainsi qu'une association des doigts vers les curseurs supportant l'engagement de plusieurs doigts vers les mains correspondantes. Les méthodes proposées ont été jugées facilement compréhensibles par les participants et ont montré leur efficacité pour interagir avec les curseurs désirés tout en ajoutant des doigts sans difficulté particulière. Notre approche de placer plusieurs curseurs et de les utiliser séquentiellement a montré l'intérêt de passer d'un curseur à l'autre et de revenir à un curseur précédemment créé.

Enfin, devant la difficulté d'utiliser plusieurs curseurs relatifs simultanément et pour améliorer la gestion d'une collection de curseurs, nous avons exploré un contexte autorisant la création de plusieurs périphériques logiques dédiés au contrôle des curseurs. Pour cela nous avons montré la réalisation d'un mur tactile en prouvant la faisabilité d'une illumination ambiante de face (FDI). Nous avons conçu un widget introduisant une interaction riche, distante et locale. Notre approche reposant sur les WallPads met en œuvre une technique complètement intégrée et compatible avec les interfaces actuelles. Nous avons combiné les avantages de l'interaction au travers d'un curseur avec l'instantanéité de la manipulation multi-doigts. Nous avons également proposé un usage convaincant et utilisable de plusieurs curseurs mettant à profit l'engagement des deux mains. Plus largement cette approche de périphériques logiques multiples pourrait être déclinée, et apporter ainsi des réponses intéressantes à plusieurs contextes d'usages.

Limites et perspectives

À l'issue de ces travaux, de nouvelles pistes de recherches s'ouvrent et des études complémentaires seraient à mener pour consolider les axes abordés dans ce mémoire.

En particulier, fort des résultats des expériences menées sur le pointage absolu sans survol, on pourrait adopter une approche plus systématique des facteurs tels que la taille du périphérique ou le rapport de cadre de celui-ci. Cela permettrait de préciser les performances des utilisateurs, statuerait du caractère linéaire de l'évolution de la précision et étendrait le domaine d'application pour des dimensions d'entrée supérieures à 200 mm.

La technique d'interaction sur grands écrans avec une tablette tactile est limitée par la difficulté à manipuler plusieurs curseurs en même temps. Nous n'avons pas étudié plus en détail les capacités des utilisateurs à agir avec deux mains dans des espaces de travail se recouvrant et parfois se confondant. Des tâches plus ciblées pourraient mettre en lumière les cas où l'utilisateur arrive à déplacer plusieurs curseurs et les paramètres facilitant ce type d'interaction tels que la fonction de transfert et l'échelle de l'espace de travail du curseur multi-doigts.

Les WallPads ont introduit une nouvelle interaction autorisant différents modes de fonctionnement. Il serait intéressant d'étudier comment la technique est utilisée et comment elle peut modifier les usages. Dans l'objectif de concevoir une étude longitudinale, une adaptation de l'approche de périphériques logiques de commande pour écrans de tailles plus modestes pourrait être envisagée. Nous avons très brièvement décrit une variante possible à la fin de la discussion du chapitre dédié aux WallPads. Ces connaissances théoriques supplémentaires amélioreraient le design des WallPads et permettrait de généraliser notre approche et de la rendre plus facilement intégrable en environnement de bureau.

Dans une autre optique, les WallPads illustrent que les techniques d'interaction multi-points doivent cohabiter avec des techniques mono-point, plus à même de répondre à certains problèmes. Les techniques multi-points devraient s'insérer, dès leur conception, dans une utilisation segmentée où l'utilisateur passe sans cesse de la manipulation mono-point simple, à une interaction plus complète, plus engagée.

Nous avons tiré parti de propriétés absolues ayant lieu sur le périphérique pour améliorer l'interaction dans l'espace de travail distant. Il reste cependant de nombreux paramètres non utilisés, comme la forme et la taille de contacts ou encore la distance séparant les contours des contacts, qui peuvent avoir un sens supplémentaire à apporter à l'interaction. La combinaison d'événements absolus et de concepts relatifs contribue à créer des interactions très riches, car si le curseur apporte une réponse convaincante, il convient de préciser que toute l'interaction n'est pas dans le curseur, mais aussi sur la surface tactile.

Bibliographie

- [1] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *Proceedings of CHI'03*, pages 105–112. ACM, 2003.
- [2] Michel Amberg, Frédéric Giraud, Betty Semail, Paolo Olivo, Géry Casiez, and Nicolas Roussel. Stimtac: a tactile input device with programmable friction. In *Adjunct proceedings of UIST'11*, pages 7–8. ACM, 2011.
- [3] Michael Balter. Tool use is just a trick of the mind. *ScienceNOW Daily News*. <http://news.sciencemag.org/2008/01/tool-use-just-trick-mind>, Janvier 2008.
- [4] Tom Bartindale, Chris Harrison, Patrick Olivier, and Scott E. Hudson. SurfaceMouse: supplementing multi-touch interaction with a virtual mouse. In *Proceedings of TEI '11*, pages 293–296. ACM, 2011.
- [5] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. Teslatouch: electrovibration for touch surfaces. In *Proceedings of UIST'10*, pages 283–292. ACM, 2010.
- [6] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederon, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proceedings of INTERACT '03*, pages 57–64, 2003.
- [7] Michel Beaudouin-Lafon. Interaction instrumentale : de la manipulation directe à la réalité augmentée. In *Actes Neuvièmes Journées sur l'Interaction Homme-Machine*, IHM'97. Cépaduès-Éditions, 1997.
- [8] Michel Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 446–453, New York, NY, USA, 2000. ACM.
- [9] Michel Beaudouin-Lafon, Olivier Chapuis, James Eagan, Tony Gjerlufsen, Stéphane Huot, Clemens Klokmoose, Wendy E. Mackay, Mathieu Nancel, Emmanuel Pietriga, Clément Pillias, Romain Primet, and Julie Wagner. Multisurface Interaction in the WILD Room. *IEEE COMPUTER*, 45(4) :48–56, April 2012.
- [10] Michel Beaudouin-Lafon and Wendy E. Mackay. Reification, polymorphism and reuse: three principles for designing visual interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '00, pages 102–109. ACM, 2000.

- [11] Hrvoje Benko, Shahram Izadi, Andrew D. Wilson, Xiang Cao, Dan Rosenfeld, and Ken Hinckley. Design and evaluation of interaction models for multi-touch mice. In *Proceedings of Graphics Interface 2010*, GI '10, pages 253–260. Canadian Information Processing Society, 2010.
- [12] A. Bezerianos and P. Isenberg. Perception of visual variables on tiled wall-sized displays for information visualization applications. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12) :2516–2525, Dec 2012.
- [13] Anastasia Bezerianos and Ravin Balakrishnan. The vacuum: facilitating the manipulation of distant objects. In *Proceedings of CHI '05*, pages 361–370. ACM, 2005.
- [14] Mudit Ratana Bhalla and Anand Vardhan Bhalla. Article:comparative study of various touchscreen technologies. *International Journal of Computer Applications*, 6(8) :12–18, September 2010. Published By Foundation of Computer Science.
- [15] Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Magic desk: bringing multi-touch surfaces into desktop work. In *Proceedings of CHI '11*, pages 2511–2520. ACM, 2011.
- [16] Eric A. Bier, Maureen C. Stone, Ken Pier, Ken Fishkin, Thomas Baudel, Matt Conway, William Buxton, and Tony DeRose. Toolglass and magic lenses: The see-through interface. In *Conference Companion of CHI'94*, pages 445–446. ACM, 1994.
- [17] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of CHI'04*, pages 519–526. ACM, 2004.
- [18] William Buxton. A three-state model of graphical input. In *INTERACT '90 : Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*, pages 449–456. North-Holland Publishing Co., 1990.
- [19] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. In *Proceedings of SIGGRAPH'85*, pages 215–224. ACM, 1985.
- [20] Géry Casiez and Nicolas Roussel. No more bricolage! Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of UIST'11*, pages 603–614. ACM, 2011.
- [21] Olivier Chapuis and Nicolas Roussel. UIMarks: quick graphical interaction with specific targets. In *Proceedings of UIST '10*, pages 173–182. ACM, 2010.
- [22] Hellen. A. Cunningham. Aiming error under transformed spatial mappings suggests a structure for visual-motor maps. In *Journal of Experimental Psychology : Human Perception and Performance*, pages 493–506. American Psychological Association, 1989.
- [23] Paul Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *Proceedings of UIST'01*, pages 219–226. ACM, 2001.

-
- [24] Alan Esenther and Kathy Ryall. Fluid dtmouse: better mouse support for touch-based interactions. In *Proceedings of AVI '06*, pages 112–115. ACM, 2006.
- [25] R. Everett. Lightgun. Technical report, MIT Lincoln Lab, Lexington, MA, USA., 1949.
- [26] Leah Findlater, Jacob O. Wobbrock, and Daniel Wigdor. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of CHI'11*, pages 2453–2462. ACM, 2011.
- [27] Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. Hybridpointing: fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of UIST '06*, pages 211–220. ACM, 2006.
- [28] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of CHI'07*, pages 647–656. ACM, 2007.
- [29] Jens Gerken, Hans-Christian Jetter, Toni Schmidt, and Harald Reiterer. Can "touch" get annoying? In *Proceedings of ITS'10*, pages 257–258, 2010.
- [30] Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. Technique multi-points indirecte relative pour l'interaction avec des écrans de grandes dimensions. In *Proceedings of IHM'12*, pages 63–66, 2012.
- [31] Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. Direct and indirect multi-touch interaction on an immersive wall display. Research report, Inria, June 2013.
- [32] Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. Impact of form factors and input conditions on absolute indirect-touch pointing tasks. In *Proceedings of CHI'14*. ACM, 2014. To be published.
- [33] Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. Object pointing: A complement to bitmap pointing in guis. In *Proceedings of GI'04*, pages 9–16. Canadian Human-Computer Communications Society, 2004.
- [34] B. Gurley. Lightpen. Technical report, MIT Lincoln Lab, Lexington, MA, USA., 1957.
- [35] Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary phone: learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of UIST'11*, pages 283–292. ACM, 2011.
- [36] Sean G. Gustafson, Bernhard Rabe, and Patrick M. Baudisch. Understanding palm-based imaginary interfaces: the role of visual and tactile cues when browsing. In *Proc. CHI '13*, pages 889–898. ACM, 2013.
- [37] Martin Hachet, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste de la Rivière. Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of UIST'11*, pages 587–592. ACM, 2011.
- [38] Anthony D. Hall, James B. Cunningham, Richard P. Roache, and Julie W. Cox. Factors affecting performance using touch-entry systems: Tactual recognition fields and system accuracy. *Journal of Applied Psychology*, 73(4) :711–720, November 1988.

- [39] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of UIST'05*, pages 115–118. ACM, 2005.
- [40] Chris Harrison, Julia Schwarz, and Scott E. Hudson. Tapsense: Enhancing finger interaction on touch surfaces. In *Proceedings of UIST'11*, pages 627–636. ACM, 2011.
- [41] Vincent Hayward, Oliver R. Astley, Manuel Cruz-Hernandez, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles de-la Torre. Haptic interfaces and devices, 2004.
- [42] Ken Hinckley, Mary Czerwinski, and Mike Sinclair. Interaction and modeling techniques for desktop two-handed input. In *Proceedings of UIST'98*, pages 49–58. ACM, 1998.
- [43] Christian Holz and Patrick Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of CHI'10*, pages 581–590. ACM, 2010.
- [44] Christian Holz and Patrick Baudisch. Understanding touch. In *Proceedings of CHI'11*, pages 2501–2510. ACM, 2011.
- [45] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *HCI*, 1(4) :311–338, December 1985.
- [46] Robert J. K. Jacob. What is the next generation of human-computer interaction? In *Extended abstracts of CHI'06*, CHI EA '06, pages 1707–1710. ACM, 2006.
- [47] E.A. Johnson. Touch display – a novel input/output device for computers. *Electronics Letters*, 1(8) :219–220, 1965.
- [48] Amy K. Karlson and Benjamin B. Bederson. Thumbspace: generalized one-handed input for touchscreen-based mobile devices. In *Proceedings of INTERACT'07*, pages 324–338. Springer-Verlag, 2007.
- [49] Dominik P. Käser, Maneesh Agrawala, and Mark Pauly. Fingerglass: efficient multiscale interaction on multitouch screens. In *Proceedings of CHI'11*, pages 1601–1610. ACM, 2011.
- [50] Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, and Gordon Kurtenbach. A remote control interface for large displays. In *Proceedings of UIST '04*, pages 127–136. ACM, 2004.
- [51] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics Interface 2009*, GI '09, pages 119–124. Canadian Information Processing Society, 2009.
- [52] Sebastian Knoedel and Martin Hachet. Multi-touch rst in 2d and 3d spaces: Studying the impact of directness on user performance. In *Proceedings of 3DUI'11*, pages 75–78. IEEE Computer Society, 2011.
- [53] K. C. Knowlton. Computer displays optically superimposed on input devices. *Bell System Technical Journal*, 56(3) :367–383, 1977.
- [54] M.W. Krueger. *Artificial Reality*. Addison-Wesley, 1983.

-
- [55] Celine Latulipe, Stephen Mann, Craig S. Kaplan, and Charlie L. A. Clarke. sym spline: Symmetric two-handed spline manipulation. In *Proceedings of CHI'06*, pages 349–358. ACM, 2006.
- [56] Ray F. Lin and Colin G. Drury. Verification of models for ballistic movement time and endpoint variability. *Ergonomics*, 56(4) :623–636, 2013.
- [57] Shu-Yang Lin, Chao-Huai Su, Kai-Yin Cheng, Rong-Hao Liang, Tzu-Hao Kuo, and Bing-Yu Chen. Pub - point upon body: exploring eyes-free interaction and methods on an arm. In *Proceedings of UIST'11*, pages 481–488. ACM, 2011.
- [58] Shahzad Malik and Joe Laszlo. Visual touchpad: a two-handed gestural input device. In *Proceedings of ICMI'04*, pages 289–296. ACM, 2004.
- [59] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proceedings of UIST'05*, pages 43–52. ACM, 2005.
- [60] Nicolai Marquardt, Johannes Kiemer, and Saul Greenberg. What caused that touch?: Expressive interaction with a surface through fiduciary-tagged gloves. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 139–142. ACM, 2010.
- [61] Justin Matejka, Tovi Grossman, Jessica Lo, and George Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. In *Proceedings of CHI '09*, pages 1073–1082. ACM, 2009.
- [62] David C. McCallum and Pourang Irani. Arc-pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of UIST'09*, pages 153–156. ACM, 2009.
- [63] Nimish Mehta. A flexible machine interface. Master's thesis, Department of Electrical Engineering, University of Toronto, 1982.
- [64] Tomer Moscovich. Contact area interaction with sliding widgets. In *Proceedings of UIST'09*, pages 13–22. ACM, 2009.
- [65] Tomer Moscovich and John F. Hughes. Multi-finger cursor techniques. In *Proc. GI '06*, pages 1–7. Canadian Information Processing Society, 2006.
- [66] Tomer Moscovich and John F. Hughes. Indirect mappings of multi-touch input using one and two hands. In *Proceedings of CHI'08*, pages 1275–1284. ACM, 2008.
- [67] Lloyd H. Nakatani and John A. Rohrlich. Soft machines: A philosophy of user-computer interface design. In *Proceedings of CHI'83*, pages 19–23. ACM, 1983.
- [68] Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 831–840, New York, NY, USA, 2013. ACM.

- [69] Donald Norman and Bahar Wadia. The next touch evolution advancing the consumer experience in other realms: Tasks and tough environments. *SID Symposium Digest of Technical Papers*, 44(1) :541–543, 2013.
- [70] Thomas Pietrzak, Andrew Crossan, Stephen A. Brewster, Benoit Martin, and Isabelle Pecci. Creating usable pin array tactons for nonvisual information. *EEE Trans. Haptics*, 2(2) :61–72, April 2009.
- [71] R. L. Potter, L. J. Weldon, and B. Shneiderman. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of CHI'88*, pages 27–32. ACM, 1988.
- [72] Gabriel Robles-De-La-Torre. The importance of the sense of touch in virtual and real environments. *IEEE MultiMedia*, 13(3) :24–30, July 2006.
- [73] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Taptap and magstick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces, AVI '08*, pages 146–153. ACM, 2008.
- [74] Chris Rutkowski. An introduction to the human applications standard computer interface. *BYTE*, 1982.
- [75] G.H. Sage. *Introduction to motor behavior : a neuropsychological approach*. Addison-Wesley series in physical education. Addison-Wesley Pub. Co., 1977.
- [76] Arvind Satyanarayan, Nadir Weibel, and James Hollan. Using overlays to support collaborative interaction with display walls. In *Proceedings of IUI'12*, pages 105–108. ACM, 2012.
- [77] Dominik Schmidt, Florian Block, and Hans Gellersen. A comparison of direct and indirect multi-touch input for large surfaces. In *Proceedings of INTERACT'09*, pages 582–594, 2009.
- [78] Andrew Sears, Doreen Revis, Janet Swatski, Rob Crittenden, and Ben Shneiderman. Investigating touchscreen typing: The effect of keyboard size on typing speed. *Behaviour & Information Technology*, 12 :17–22, 1993.
- [79] B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8) :57–69, August 1983.
- [80] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat finger worries: How older and younger users physically interact with PDAs. In *Proceedings of INTERACT'2005*, pages 267–280, 2005.
- [81] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of CHI '95*, pages 265–272. ACM, 1995.
- [82] Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. User interface façades: towards fully adaptable user interfaces. In *Proceedings of UIST '06*, pages 309–318. ACM, 2006.
- [83] Chao-Huai Su, Liwei Chan, Chien-Ting Weng, Rong-Hao Liang, Kai-Yin Cheng, and Bing-Yu Chen. Naildisplay: bringing an always available

-
- visual display to fingertips. In *Proceedings of CHI'13*, pages 1461–1464. ACM, 2013.
- [84] Ivan E. Sutherland. Sketchpad: a man-machine graphical communication system. In *Proceedings of the May 21-23, 1963, spring joint computer conference*, AFIPS '63 (Spring), pages 329–346. ACM, 1963.
- [85] Desney S. Tan, Brian Meyers, and Mary Czerwinski. Wincuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI '04 EA*, pages 1525–1528. ACM, 2004.
- [86] Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao, and Billy Chen. Mouse 2.0: multi-touch meets the mouse. In *Proceedings of UIST '09*, pages 33–42. ACM, 2009.
- [87] Luc Vlaming, Christopher Collins, Mark Hancock, Miguel Nacenta, Tobias Isenberg, and Sheelagh Carpendale. Integrating 2d mouse emulation with 3d manipulation for visualizations on a multi-touch table. In *Proceedings of ITS'10*, pages 221–230. ACM, 2010.
- [88] Daniel Vogel and Ravin Balakrishnan. Occlusion-aware interfaces. In *Proceedings of CHI'10*, pages 263–272. ACM, 2010.
- [89] Daniel Vogel and Patrick Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of CHI'07*, pages 657–666. ACM, 2007.
- [90] Daniel Vogel and Géry Casiez. Conté: multimodal input inspired by an artist's crayon. In *Proceedings of UIST '11*, pages 357–366, 2011.
- [91] Daniel Vogel and Géry Casiez. Hand occlusion on a multi-touch tabletop. In *Proceedings of CHI'12*, pages 2307–2316. ACM, 2012.
- [92] Stephen Vaida, Matthew Tobiasz, Julie Stromer, Petra Isenberg, and Sheelagh Carpendale. Getting practical with interactive tabletop displays: designing for dense data, "fat fingers," diverse interactions, and face-to-face collaboration. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 109–116. ACM, 2009.
- [93] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proceedings of UIST'09*, pages 23–32. ACM, 2009.
- [94] Feng Wang and Xiangshi Ren. Empirical evaluation for finger input properties in multi-touch interaction. In *Proc. CHI '09*, pages 1063–1072. ACM, 2009.
- [95] Malte Weiss, Simon Voelker, Christine Sutter, and Jan Borchers. Bendesk: dragging across the curve. In *Proceedings of ITS '10*, pages 1–10. ACM, 2010.
- [96] Pierre Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*, UIST '91, pages 27–33, New York, NY, USA, 1991. ACM.

- [97] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: a see-through mobile device. In *Proceedings of UIST'07*, pages 269–278. ACM, 2007.
- [98] Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Effects of display position and control space orientation on user preference and performance. In *Proceedings of CHI'06*, pages 309–318. ACM, 2006.
- [99] Andrew D. Wilson. Touchlight: An imaging touch screen and display for gesture-based interaction. In *ICMI '04 : Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76. ACM Press, 2004.
- [100] Andrew D. Wilson. Playanywhere: a compact interactive tabletop projection-vision system. In *Proceedings of UIST'05*, pages 83–92. ACM, 2005.
- [101] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of CHI'11*, pages 143–146. ACM, 2011.
- [102] Jeffrey A. Wolfeld. Real time control of a robot tactile sensor. Master's thesis, Philadelphia : Moore School of Electrical Engineering, 1981.
- [103] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of UIST'03*, pages 193–202. ACM, 2003.
- [104] Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. Lensmouse: augmenting the mouse with an interactive touch display. In *Proceedings of CHI'10*, pages 2431–2440. ACM, 2010.
- [105] Tao Zeng, Frédéric Giraud, Betty Lemaire-Semail, and Michel Amberg. Analysis of a new haptic display coupling tactile and kinesthetic feedback to render texture and shape. In *Haptics : Generating and Perceiving Tangible Sensations*, volume 6192 of *LNCS*, pages 87–93. Springer Berlin Heidelberg, 2010.
- [106] S. Zhai and P. Milgram. Human performance evaluation of manipulation schemes in virtual environments. In *Proceedings of VRAIS'93*, pages 155–161. IEEE Computer Society, 1993.
- [107] Shumin Zhai. *Human Performance in Six Degree of Freedom Input Control*. PhD thesis, University of Toronto, 1995.

Interactions multi-points indirectes sur grands écrans

Résumé

L'interaction tactile multi-points co-localisée procure un fort sentiment de contrôle direct mais introduit des problèmes d'occultation, de précision et de fatigue. Ces problèmes sont exacerbés sur les écrans de grandes dimensions, qui supportent la manipulation de grandes quantités de données, souvent en collaboration avec d'autres utilisateurs. Pour résoudre ces problèmes, cette thèse s'intéresse à l'introduction d'indirections dans l'interaction multi-points sur grands écrans.

Nos travaux portent sur l'étude des facteurs qui influencent la précision dans les tâches de sélection sans prévisualisation du contact. Les résultats de deux expériences contrôlées ont ainsi permis d'estimer la précision attendue selon les caractéristiques des périphériques d'entrée et de sortie et des conditions d'utilisation. Sur la base de ces connaissances, nos travaux portent également sur le développement de nouvelles techniques d'interaction pour écrans de grandes dimensions. Nous avons d'abord proposé une technique permettant d'interagir indirectement avec une tablette tactile, en introduisant la gestion de plusieurs curseurs contrôlés relativement. Nous avons ensuite conçu une interaction reposant sur la création de plusieurs périphériques virtuels qui supportent une interaction à la fois directe et indirecte tout en permettant de réorganiser les éléments d'une interface. Parallèlement à cela, nous avons transformé un écran mural de 6×2,4m en une surface tactile multi-points afin de pouvoir tester les techniques et scénarios discutés dans ces travaux.

Mots-clés: interaction homme-machine ; interactions indirectes ; interactions multi-points ; curseurs multi-doigts ; périphériques tactiles

Indirect multi-touch interactions on large displays

Abstract

Co-localized multi-touch interaction affords a high feeling of directness but introduces problems of occlusion, precision and fatigue. These problems are even more acute with large displays, supporting the interaction with large data sets, often in collaboration with other users. To solve these problems this thesis focuses on the introduction of indirections in multi-touch interaction on large displays.

We first study factors impacting accuracy in selection tasks without contact preview. Results of two controlled experiments allowed to estimate the expected accuracy depending on input and output device characteristics and input conditions. Based on this knowledge, we developed new interaction techniques for large displays. We first introduced a technique to interact indirectly using a tactile tablet by introducing the control of several cursors using a relative mapping. Then we proposed an interaction technique based on the creation of multiple soft-devices that supports both direct and indirect interaction and allows to re-arrange the interface of legacy applications. In order to design, test and evaluate these techniques, we turned a back projected wall display (6×2,4m) into an interactive multi-touch surface.

Keywords: human-computer interaction ; indirect interactions ; multi-touch interactions ; multi-touch cursors ; tactile devices