

Comprendre et concevoir l'interaction tactile avec identification des doigts

Alix **GOGUEY**

Thèse dirigée par Géry **CASIEZ**

Thèse présentée et soutenue publiquement le 10 octobre 2016
pour obtenir le grade de Docteur de l'Université de Lille
dans la spécialité Informatique.
École Doctorale Sciences pour l'Ingénieur Lille Nord-de-France

Composition du JURY

PRÉSIDENT	:	Laurence DUCHIEN	Université de Lille
RAPPORTEURS	:	Michel BEAUDOUIN-LAFON	Université Paris-Sud
		Laurence NIGAY	Université Grenoble Alpes
EXAMINATEURS	:	Edward LANK	University of Waterloo
		Éric LECOLINET	Télécom ParisTech

Inria Lille - Nord Europe
40 avenue Halley
59650 Villeneuve-d'Ascq
FRANCE

Equipe **MJOLNIR**

Université Lille 1
Cité Scientifique
59650 Villeneuve-d'Ascq
FRANCE



Remerciements

La thèse est une aventure qui se vit en famille. Mon aventure avait cette particularité d'en avoir trois qui ont eu tendance à n'en former qu'une.

Je voudrais tout d'abord remercier ma famille Recherche et tout particulièrement mon directeur de thèse Géry Casiez. Grâce à lui j'ai appris un métier qui me passionne. Son encadrement, ses conseils et son expérience m'ont énormément appris. J'ai eu un plaisir fou à travailler avec lui. Géry merci ! Je voudrais aussi remercier mes rapporteurs Michel Beaudouin-Lafon et Laurence Nigay, et mes examinateurs Laurence Duchien, Edward Lank et Éric Lecolinet, pour avoir pris le temps de lire et évaluer ce travail. C'est un immense honneur pour moi qu'ils aient accepté de faire parti de mon jury. Je tiens aussi à remercier François Bérard et Gaëlle Calvary sans qui je ne connaîtrais pas l'acronyme IHM. C'est grâce à leurs cours que je me suis passionné pour ce domaine et que j'ai construit mon engouement pour la Recherche et l'enseignement. Je remercie aussi toutes les personnes qui ont fait le bonheur de mon quotidien au labo : Hanaë, Jonathan, Bruno, Fanny, Sylvain, Stéphane, Nicolas, Thomas, Damien, Amira, Thibaut, Justin, Christian, Izzat, Karine, Ludo, Marc-Antoine Mathieu(s), Sébastien, Yosra, Laurent, Marie-Agnès, Ben Soussan et tous ceux que j'ai croisé pendant ces 3 années. Je vous dois beaucoup et notamment un lien vers ma seconde famille : mes amis.

Merci pour votre soutien, merci pour les moments de détente, merci, mille fois merci. Que ce soit au panda au Tir na nog, au Morrisson ou au café disquaire, j'ai aimé nos discussions. Je complète donc ma liste avec : Gaël, Eulalie, Sara, Philippe, Vincent, Olivier, Djoude, Mathix, François, Clément, Thê-Minh, Hugues, Céline, Jérôme, Célia, Libouille, Jean, Soucia, Audrey, Piepierre, la trêve, les peaux d'renards et tous ceux que j'oublie... Non seulement j'ai des cheveux blancs mais j'ai la mémoire qui décaroche, pardon.

Et enfin merci à LA famille ! On la choisi pas mais qu'est-ce que le monde est bien fait. Effectuer cette thèse à Lille, dans mon Nord natal, m'a permis de me rapprocher d'eux et d'encore plus les aimer. Momon, Popo, Piepierre, Totom, Biquette, MM, Professeur Xavier, les Blésois c'est aussi votre thèse.

J'oublie un truc... Ah oui... Je dédie cette thèse à ma reuss : Léa #tesmonone #lovecoeursetpailletesdanstaface.

Résumés

Comprendre et concevoir l'interaction tactile avec identification des doigts

FR La dernière décennie a vu s'établir la démocratisation des interfaces tactiles. De nombreux logiciels jusque là réservés aux ordinateurs de bureau offrent désormais une version qui se contrôle du bout des doigts. Cependant, l'expressivité limitée de la modalité tactile restreint drastiquement le nombre de fonctionnalités disponibles. La recherche explore donc différentes pistes pour augmenter cette expressivité notamment par l'identification des doigts. Alors que la littérature se focalise principalement sur les méthodes d'identification des doigts, cette thèse vise à mieux en comprendre l'utilisation afin de guider la conception de techniques d'interaction.

Nous réalisons d'abord une revue des technologies existantes, présentons nos prototypes et évaluons l'utilisation de la reconnaissance des empreintes digitales, qui nous semble la solution la plus prometteuse en vue d'une intégration dans des systèmes commerciaux. Par la réalisation d'expériences contrôlées, nous étudions ensuite les différences de performances et de préférences entre les doigts, l'influence de l'identification des doigts sur la stratégie de réalisation de tâches. Nous explorons également comment intégrer cette information au sein des techniques d'interaction existantes et aider les utilisateurs à appréhender le riche vocabulaire offert. Nos travaux aboutissent à des recommandations pour la conception des techniques d'interaction basées sur l'identification des doigts que nous mettons en œuvre dans trois contextes d'interaction (grands écrans, tablettes et smartphones) au travers d'*Adoiraccourcix*, une technique d'interaction qui combine sélection de commandes et contrôle continu de paramètres.

Mots-clefs Interaction homme-machine ; techniques d'interaction ; interaction tactile ; identification des doigts

Understanding and designing touch interaction using finger identification

EN During the last decade, touch interfaces have become more and more ubiquitous. A lot of software applications initially designed for desktop computers have now a tactile counterpart. However, the limited expressiveness of the touch modality restricts drastically the amount of features available on touch applications. Researchers have been exploring different tracks on how to augment this expressiveness, notably through finger identification. While the literature mainly focuses on tackling the technological challenge, this dissertation aims at better understanding its use in order to provide guidelines for designing interaction techniques.

We first summarize existing technologies, present our prototypes and evaluate the use of fingerprint recognition, that we perceive as the most promising solution with the aim of being integrated in consumer devices. Through controlled experiments, we study the differences of performance and preference between fingers, the influence of finger identification on user strategies to complete tasks. We also explore how to integrate this new information along existing interaction techniques and help users grasp the rich vocabulary provided. Our works lead to guidelines for designing interaction techniques leveraging finger identification that we implement in three different interaction contexts (tabletops, tablets and smartphones) through *Adoiraccourcix*, an interaction technique combining command selection and parameter manipulation.

Keywords Human-computer interaction ; interaction technique ; touch interaction ; finger identification

Table des matières

Remerciements.....	iii
Résumés.....	v
Table des matières.....	vii
Table des figures	x
Liste des tableaux.....	xiii
Introduction.....	1
Objectif de la thèse	3
Contributions	4
Structure du document	4
Notes de lecture	5
1 Technologie.....	9
1.1 État de l'art	10
1.1.1 L'ère du <i>Rich Touch</i>	14
1.1.2 L'identification des doigts	19
1.1.3 Des technologies pas encore matures	25
1.2 Prototypage	25
1.2.1 Architecture	26
1.2.2 Sur grandes surfaces	27
1.2.3 Sur tablettes et smartphones	29
1.2.4 Les autres pistes explorées	29
1.3 Utilisation des empreintes digitales	30
1.3.1 Partie utile de la pulpe	32
1.3.2 Utilisation des empreintes digitales	39
1.3.3 Conclusion et travaux futurs	49
1.4 Conclusion	50
2 Facteurs d'intégration.....	51
2.1 Dextérité des doigts et des accords	52
2.1.1 État de l'art	53
2.1.2 Sélection de cible	58

2.1.3	Performance et préférence	63
2.1.4	Performance et préférence : pointage de cible	65
2.1.5	Performance et préférence : déplacement d'objets	68
2.1.6	Performance et préférence : changement d'échelle	71
2.1.7	Performance et préférence : rotation d'objet	73
2.1.8	Discussion et recommandations de conception	76
2.1.9	Conclusion et travaux futurs	79
2.2	Cognition	81
2.2.1	État de l'art	82
2.2.2	Stratégie	84
2.2.3	Mapping	92
2.3	Conclusion	97
3	Technique d'interaction	99
3.1	État de l'art	100
3.1.1	Du <i>Touch</i> au <i>Rich Touch</i> : aperçu de la sélection de commande	101
3.1.2	Les techniques basées sur l'identification des doigts	103
3.2	<i>Adoiraccourcir</i>	106
3.2.1	Espace des entrées qui utilisent l'identification des doigts	107
3.2.2	Une technique d'interaction en trois étapes	108
3.2.3	Justification de la conception, variations, et analyse	109
3.3	Exemples de mises en œuvre	115
3.3.1	Sur grands écrans	115
3.3.2	Sur tablette	120
3.3.3	Sur <i>smartphone</i>	123
3.4	Évaluation et validation	124
3.4.1	Étude qualitative	125
3.4.2	Étude quantitative	127
3.4.3	Analyse basée sur le <i>Fingerstroke-Level Model</i>	129
3.5	Discussion	134
3.5.1	Ergonomie de l'appareil	134
3.5.2	Cohérence des associations	134
3.5.3	Choix des associations commandes-accords	135
3.5.4	Efficacité de l'intégration de la sélection de commandes et du contrôle continu de paramètres	135
3.5.5	Limites	136
3.6	Conclusion	137
	Conclusion	139
	Résumé	139
	Perspectives de recherche	141
	Glossaire	145
	A Publications	147

B	Détail de l'observation des accords	149
C	Détails des taux de succès de reconnaissance	153
	C.1 Reconnaissance des doigts	153
	C.2 Reconnaissance des doigts et des orientations	153
D	Bibliographie	161

Table des figures

1	Les différents facteurs de forme utilisés au cours de cette thèse. .	4
1.1	Technologies d'identification des doigts utilisées durant notre phase d'exploration	10
1.2	Principe de fonctionnement d'un écran tactile capacitif.	11
1.3	Principe de fonctionnement d'un écran tactile résistif.	12
1.4	Principe de fonctionnement d'un écran tactile <i>FTIR</i> infrarouge.	12
1.5	Événements tactiles transmis au système.	13
1.6	Illustration du mécanisme de copier coller sur <i>Android messenger</i>	15
1.7	Gestes reconnus par <i>TouchTools</i> inspirés d'objets du quotidien.	16
1.8	Description de l'orientation du doigt à l'aide des angles de tangage, de roulis et de lacet.	18
1.9	Illustration de <i>TapSense</i>	18
1.10	Relations géométriques utilisées par Wagner <i>et al.</i> pour identifier des accords à 3 doigts.	20
1.11	Exemple de boîte englobant la main utilisée par Lepinski <i>et al.</i> pour identifier 14 accords.	21
1.12	Dispositif utilisé par Zheng <i>et al.</i> pour identifier les doigts.	22
1.13	Dispositif utilisé par Gupta <i>et al.</i> pour identifier les doigts.	23
1.14	Gants utilisés par Marquardt <i>et al.</i> pour identifier les doigts.	24
1.15	Prototype d'identification des doigts sur grande surface.	27
1.16	Schéma de l'architecture fonctionnelle des prototypes développés.	28
1.17	Prototype d'identification des doigts pour tablette et <i>smartphone</i>	29
1.18	Dispositif <i>Fiberio</i>	31
1.19	Exemples de minuties.	32
1.20	Différentes parties de la pulpe.	33
1.21	Influence du tangage et du roulis sur la partie de la pulpe en contact avec une surface.	33
1.22	Dispositif utilisé pour l'expérience visant à déterminer la partie utile de la pulpe.	36
1.23	Distributions des données d'orientation au cours des 1 500 premières millisecondes des contacts.	38
1.24	Distributions des orientations en degrés de la première phalange de chaque doigt.	40
1.25	Dispositif d'acquisition <i>FTIR</i> d'empreintes digitales.	41
1.26	Exemple d'image acquise par notre dispositif.	42

1.27	Contrôle de l'orientation d'un doigt lors de l'acquisition d'une empreinte digitale.	43
1.28	Exemple de calcul des taux de succès, d'échec et d'incertitude pour une empreinte e avec $n = 3$	44
1.29	Taux de succès moyen de reconnaissance du doigt pour $n = 1$	47
1.30	Taux de succès moyen de reconnaissance du doigt pour $n = 2$	48
2.1	Exemples de tâche de l'étude <i>Sélection de cible</i>	59
2.2	Taux d'erreur par CONTACT en fonction de la TOLÉRANCE.	60
2.3	Distances moyennes entre les deux doigts composant un ACCORD pour chacune des TOLÉRANCES.	62
2.4	Tâche de pointage.	66
2.5	Tâche de déplacement d'objet.	69
2.6	Tâche de changement d'échelle.	72
2.7	Tâche de rotation.	74
2.8	<i>Throughput</i> des différentes tâches (en <i>bps</i>) et notes moyennes des préférences (sur 5).	77
2.9	Exemples d'associations entre commandes et accords basés sur les scores de performances et de préférences.	80
2.10	Exemples d'instances pour les TÂCHES T1, T2 et T3.	86
2.11	Illustration de l'utilisation de la <i>Palette d'Outils Fixe</i> , la <i>Toolglass</i> et la <i>Palette Flottante avec Identification des Doigts</i>	88
2.12	Dispositif de l'expérience.	89
2.13	Taux <i>OriObj</i> et <i>OriOut</i> moyens pour chaque TECHNIQUE.	90
2.14	Miroir fixé devant la caméra intégrée qui dévie le champs de capture vers le pavé tactile.	94
2.15	Application d'annotation des gestes collectés.	95
3.1	Exemple bi-manuel illustrant la logique d' <i>Adoiraccourcix</i>	100
3.2	<i>Adoiraccourcix</i> au service d'une application de dessin vectoriel.	116
3.3	Exemple d'interaction sur grande surface avec <i>Adoiraccourcix</i>	119
3.4	RST, effectué à l'aide de l'accord pouce + index, contraint par le nd-index (désactivation du changement d'échelle) et le nd-majeur (ancrage du centre) n'autorisant que la rotation de l'objet.	120
3.5	Mode d'aide explicite accessible en appuyant avec les 10 doigts en même temps.	121
3.6	Le <i>sélecteur</i> pouce sur tablette.	122
3.7	Exemples d'utilisation des doigts <i>déclencheurs</i> du groupe d'associations par défaut.	122
3.8	Autres fonctionnalités.	122
3.9	<i>Sélecteur</i> et <i>déclencheurs</i> sur <i>smartphone</i>	123
3.10	<i>Déclencheurs</i> sur <i>smartphone</i>	124
3.11	Tâches de manipulation de la deuxième partie de l'étude qualitative.	126
3.12	La <i>GUI</i> montrant la tâche de COPIER.	128
3.13	TÂCHES de l'expérience quantitative.	129

3.14 Temps moyens de réalisation pour <i>Adoiraccourcix</i> et la <i>GUI</i> sur les différentes TÂCHES.	130
C.1 Taux de succès moyen de reconnaissance du doigt pour $n = 1$. . .	154
C.2 Taux de succès moyen de reconnaissance du doigt pour $n = 2$. . .	155
C.3 Taux de succès moyen de reconnaissance du doigt pour $n = 3$. . .	156
C.4 Taux de succès moyen de reconnaissance du doigt et de l'orientation pour $n = 1$	158
C.5 Taux de succès moyen de reconnaissance du doigt et de l'orientation pour $n = 2$	159
C.6 Taux de succès moyen de reconnaissance du doigt et de l'orientation pour $n = 3$	160

Liste des tableaux

1.1	Résumé des notes données par les 15 participants aux dix propositions.	37
1.2	Taux de reconnaissance des empreintes digitales des index, majeurs et annulaires droits (<i>i.e.</i> I , M et A) à plat.	45
1.3	Taux de reconnaissance des empreintes digitales à plat.	46
2.1	Résumé des critères évalués dans les travaux précédents.	56
2.2	TOLÉRANCE à laquelle le taux d'erreur pour un CONTACT donné commence à être inférieur à 5%.	61
2.3	Résumé pour chaque DOIGT des différents paramètres des modélisations pour l'expérience de pointage.	67
2.4	Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de pointage.	68
2.5	Résumé pour chaque DOIGT des différents paramètres des modélisations pour l'expérience de déplacement d'objet.	70
2.6	Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de déplacement d'objet.	71
2.7	Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de changement d'échelle.	74
2.8	Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de rotation.	76
2.9	Recommandations des CONTACTS à privilégier ($C_{préf}$) et à éviter ($C_{préf}$).	76
2.10	Recommandations des CONTACTS à privilégier ($C_{préf}$) et à éviter ($C_{préf}$) en fonctions des performances ou des préférences.	79
2.11	Gestes multi-points du pavé tactile <i>Apple</i>	93
3.1	Nombre de k -combinaisons de n doigts pour différentes valeurs de n .107	
3.2	Groupes d'associations commande-accord.	118
3.3	Analyse comparative basée sur le <i>FLM</i> entre la <i>GUI</i> et <i>Adoirac-courci</i>	132
B.1	Détails des observations des accords sur pavé tactile par geste (1/3).150	
B.2	Détails des observations des accords sur pavé tactile par geste (2/3).151	
B.3	Détails des observations des accords sur pavé tactile par geste (3/3).152	

Introduction

Communiquer, ou transmettre de l'information, est une compétence que nous acquérons très rapidement. Pour évoluer dans notre monde bien réel, il nous faut maîtriser les outils pour interagir avec. Sans surprise, l'humain excelle dans ce domaine. Il suffit de se pencher sur la communication entre deux personnes pour s'en rendre compte. Nous connaissons et exploitons nombre de moyens pour nous faire comprendre et communiquer nos idées, nos pensées : l'oral, l'écrit, les gestes ou encore les dessins. Il faut cependant bien se rendre compte que le processus de communication est à double sens. Si nous excellons à transmettre de l'information à d'autres humains, c'est aussi parce que nos interlocuteurs sont capables de déchiffrer l'information transmise. La méthode de communication doit être partagée entre les deux parties : lors d'une conversation avec une personne de langue maternelle différente, nous nous adaptons et communiquons par exemple en anglais, voire à l'aide de gestes.

La communication avec une machine est, elle, plus compliquée. Pour transmettre des instructions, nous devons nous plier aux protocoles de communication que celle-ci connaît et la flexibilité, que les humains ont entre eux, n'existe généralement pas. En schématisant, les ordinateurs ne sont, par exemple, capables que de recevoir ou transmettre une information binaire qu'il nous est *a priori* difficile de maîtriser. Fort heureusement, les concepteurs de ces machines rivalisent d'ingéniosité pour augmenter l'*expressivité* (*i.e.* les différents canaux qui servent à transmettre une instruction à la machine). Ainsi les dispositifs tels que la souris, le clavier, l'écran, les boutons (physiques ou virtuels), les menus contextuels, et autres mécanismes d'interaction, nous permettent de communiquer relativement facilement et efficacement avec un ordinateur et d'effectuer des tâches complexes. De plus, à l'instar d'une communication entre humains, une même transmission d'informations peut-être effectuée suivant différentes modalités améliorant potentiellement l'efficacité de la communication. La notion de *novice* et d'*expert* entre alors en jeu. Par exemple, un utilisateur *novice* préférera l'usage de la souris et de boutons virtuels pour accomplir une tâche, alors qu'un utilisateur *expert* utilisera des raccourcis clavier, une alternative efficace, qui lui fera gagner du temps [GDB07].

Les ordinateurs sont un parfait exemple de machines dont l'expressivité est conséquente. Toutefois, ce n'est pas le cas de toutes : les interfaces tactiles (*ex.* tablette, *smartphone**, *etc.*) sont par exemple beaucoup plus limitées. Bien qu'avec leur démocratisation, la plupart des logiciels, jusque là réservés aux

ordinateurs, offrent aux utilisateurs une version qui se contrôle depuis le bout des doigts, les métaphores d'interaction ont dû être adaptées. À cause du vocabulaire d'entrées relativement restreint sur les interfaces multi-points, la plupart des commandes sont représentées visuellement. Elles se trouvent plus ou moins loin de l'objet d'intérêt dans des barres d'outils, des palettes ou autres menus qui occupent de la place à l'écran et cassent la fluidité de l'interaction à cause de mouvements d'allers-retours complexes et chronophages induits par l'interface graphique [BL00]. L'espace restreint des écrans ainsi que le vocabulaire d'entrées succinct tendent à rendre les interfaces tactiles d'aujourd'hui bien moins riches en fonctionnalités que leur équivalent sur ordinateur¹. Les concepteurs d'interfaces ont aussi recours aux techniques prenant en compte le nombre de doigts, aux gestes effectués ou encore à la temporisation, pour déclencher différentes commandes. Par exemple, dans l'éditeur de texte *iPad Pages*, il est possible de déplacer le curseur d'un caractère avec un glissé à un doigt, d'un mot avec un glissé à deux doigts et d'un paragraphe avec un glissé à trois doigts [App]. Néanmoins le vocabulaire disponible reste beaucoup moins expressif que celui des ordinateurs. Par conséquent, l'élargissement de l'espace des entrées multi-points est devenu une question de recherche importante. Les chercheurs s'appuient sur de nouvelles informations comme l'utilisation des caractéristiques physiques de l'écran [SLG13], de séquences rythmiques [GFH⁺12], de l'orientation [XSH15] ou encore des forces normales et tangentielles [WR09a]. La multiplication des informations exploitées augmente petit à petit cet espace d'entrées multi-points.

La plupart de ces nouvelles informations trouvent leur inspiration de nos aptitudes dans le monde réel. Certains gestes, comme la rotation ou le *pan*^{*}, sont des traductions proches sémantiquement et physiquement [HHN85] des mouvements réels que nous pouvons faire lors de la manipulation de documents papiers. L'utilisation de différents niveaux de pression pour déplacer un ou plusieurs objets superposés [CWB⁺08] est aussi un exemple inspiré de notre dextérité naturelle. Toutefois, avant d'aller jusqu'à s'inspirer des mouvements que nous savons faire, nous pourrions utiliser les outils que nous avons à notre disposition, et en particulier nos dix doigts. Jusqu'ici tous les contacts sur un écran tactile sont traités de façon équivalente, *i.e.* un glissé à un doigt déclenche la même action qu'il soit effectué avec l'index, le majeur ou le pouce. Or nous différencions naturellement nos doigts en fonction de l'information que nous voulons véhiculer : nous utilisons par exemple le pouce pour approuver ou non, ou encore l'index pour indiquer une direction. De plus, nous sommes capables de les combiner pour augmenter nos moyens d'expression. La langue des signes en est un exemple flagrant. Identifier les doigts est donc un moyen simple et efficace d'augmenter l'expressivité de la modalité tactile.

Identifier *quels* doigts sont en contact avec une surface tactile offre un espace des entrées multi-points bien plus large. Des études récentes ont montré que pour la sélection de commandes à un doigt, l'identification des doigts est une solution prometteuse pour augmenter l'expressivité [RGB⁺15]. Les cher-

1. *Adobe Photoshop (CS6)* propose 648 commandes sur la version PC et seulement 35 sur la version tablette (Express) [WLS14].

cheurs s'emploient déjà à explorer différentes technologies pour développer une identification des doigts fiable. Ils utilisent les relations géométriques entre les différents points de contact [AT10, EKF12, MVER12, Wes99, LGF10, WLS14], des caméras surplombant la surface de travail [MRB05, WC04], des gants augmentés [MKL⁺11], de la reconnaissance d'empreintes digitales [SK98, HB13], des électromyogrammes placés sur l'avant-bras [BSMT09] ou encore de faux ongles augmentés de marqueurs Radio-Frequency IDentification (*RFID*)* [VF13]. Pourtant peu de travaux s'intéressent à la compréhension et à l'utilisation de cette information. Certains fournissent quelques illustrations de scénarios possibles, mais aucun ne fournit un ensemble cohérent, n'étudie ses limitations, ni n'inscrit cette nouvelle information au sein de celles déjà existantes.

C'est dans ce contexte que s'inscrivent nos recherches. Avec des solutions technologiques viables qui se dessinent, l'identification des doigts sera vraisemblablement une information fournie par les futures générations d'écran. Nos travaux cherchent donc à mieux la comprendre afin de guider la conception des nouvelles techniques d'interaction.

Objectif de la thèse

L'objectif de cette thèse est de mieux appréhender l'interaction tactile avec identification des doigts. Pour atteindre cet objectif, nos recherches se structurent en deux parties. Dans un premier temps, nous avons souhaité étudier l'utilisateur face à cette nouvelle information : identifier les problèmes d'ergonomie liés à l'utilisation de l'identification des doigts ; identifier les points de conception favorisant une utilisation optimale des techniques d'interaction qui se basent sur l'identification des doigts ; analyser l'impact de l'intégration de ces nouvelles techniques dans le cosme déjà existant des interactions multi-points. Le but de ces études est de constituer un corpus de nouvelles connaissances scientifiques sur l'identification des doigts afin de proposer des recommandations de conception et guider l'implémentation de ces nouvelles interactions. La deuxième partie a pour but d'explorer et de développer des techniques d'interaction tout en suivant les recommandations engendrées dans la première partie. Nous nous sommes intéressés à trois contextes (*cf.* Figure 1) : les grands écrans tactiles où la surface d'affichage n'est pas limitée et l'utilisation des deux mains est envisageable ; les *smartphones** où la surface d'affichage est restreinte et l'appareil est tenu par l'utilisateur ; les tablettes qui ont un contexte d'utilisation proche du *smartphone** mais dont la surface d'affichage est moins restreinte. Un objectif transversal s'est défini au cours de cette thèse : étudier et développer des prototypes d'identification des doigts. Pour atteindre l'objectif fixé initialement, il était impératif d'aborder l'aspect technologique du problème. Nous avons donc souhaité étudier les technologies existantes et ce pour deux raisons : s'en inspirer pour établir les prototypes que nous utilisons dans nos propres travaux ; identifier les solutions technologiques viables pour une utilisation dans la vie courante.

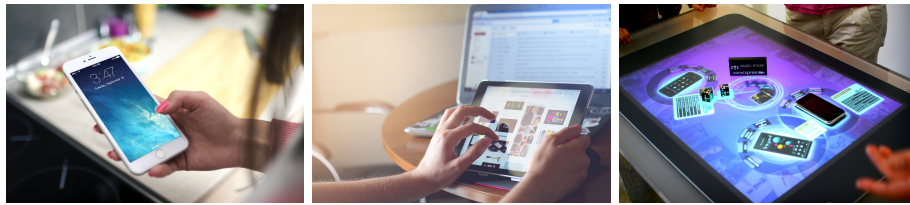


FIGURE 1: Les différents facteurs de forme utilisés au cours de cette thèse : *smartphone**, tablette et grande surface tactile.

Contributions

Nos recherches ont produit les contributions suivantes :

1. L'évaluation de l'utilisation des empreintes digitales comme solution technologique viable d'identification des doigts.
2. L'étude des différences de performances et de préférences entre les doigts et entre certains accords. Les différences trouvées permettent d'établir un panel de recommandations ergonomiques sur la conception de techniques avec identification des doigts.
3. La création de métriques quantifiant automatiquement et objectivement les stratégies utilisateurs. Ces métriques permettent l'étude de l'influence de l'identification des doigts sur ces stratégies et d'identifier des points à ne pas négliger lors de la conception des techniques avec identification des doigts (*ex. feedforward**, *feedback**).
4. L'observation des associations entre commandes et accords actuellement utilisées. Ces observations visent à comprendre l'impact de l'intégration des techniques avec identification des doigts aux côtés des techniques déjà existantes.
5. La conception et l'évaluation d'*Adoiraccourcix*, une technique de sélection de commandes intégrant le contrôle continu de paramètres. Cette technique est inspirée des raccourcis clavier et peut se décliner sur les différents facteurs de forme. Nous l'avons d'ailleurs mise en œuvre et évaluée sur grands écrans, tablettes et *smartphones**.

Nous regroupons en annexe A, la liste des publications produites durant cette thèse.

Structure du document

Cette thèse est divisée en trois chapitres.

Le chapitre 1 aborde la partie technologique de l'identification des doigts. Ce chapitre commence par un état de l'art des méthodes et prototypes existants pour résoudre le défi technique. Il se poursuit par la description de nos propres solutions. Le chapitre se termine par une étude qui établit la viabilité, sous forme

de preuve de concept, de l'utilisation des empreintes digitales pour discriminer les doigts sur un écran tactile.

Le chapitre 2 aborde la partie facteurs humains de l'identification des doigts. Ce chapitre est divisé en deux parties : une partie dextérité et une partie cognition. La partie dextérité présente une étude sur les habilités motrices des utilisateurs à utiliser des accords spécifiques. La partie cognition présente deux études utilisateurs qui s'intéressent à l'impact de l'identification des doigts sur le raisonnement utilisateur ainsi que leurs habitudes. Chaque étude établit une série de recommandations qui guide la conception de techniques d'interaction.

Le chapitre 3 aborde la partie conception de l'identification des doigts. Ce chapitre présente *Adoiraccourcix*, une technique d'interaction basée sur l'identification des doigts qui suit les recommandations établies au chapitre précédent. Après une description détaillée de la technique et de différentes mises en œuvre sur grands écrans, tablettes et *smartphones*^{*}, il se poursuit par une évaluation de la technique.

Enfin, nous concluons cette thèse par un résumé et une discussion des résultats de nos travaux, avant d'ouvrir sur les différentes directions des travaux futurs.

Notes de lecture

À l'attention du lecteur

Comme l'indique le titre de cette thèse, le contenu se focalisera essentiellement sur l'*identification des doigts*. Les premières mentions de cette expression apparaissent à la fin des années 70 avec la publication de brevets comme [Sch77, Sch82, Sch85]. Dans ces brevets, l'*identification des doigts* signifie *identification d'un individu* grâce à ses empreintes digitales. L'appropriation de cette expression en interaction homme-machine vient en 1998, lorsque Sugirura *et al.* [SK98] font état d'une technique d'interaction basée sur l'*identification des doigts*. Cette technique est aussi basée sur la reconnaissance des empreintes digitales d'un individu. C'est l'année suivante, en 1999, que l'expression est associée à la modalité tactile. Avec sa thèse *Hand tracking, finger identification, and chordic manipulation on a multi-touch surface* [Wes99], Westerman identifie les doigts en se basant sur la géométrie des contacts, mais ne propose aucune technique d'interaction. Celles-ci viendront en 2005 [MRB05] puis chaque année (sauf 2015) depuis 2009 [BSMT09, AT10, MKL⁺11, MVER12, GHB⁺13, CH14, GB16].

Mais qu'est-ce que l'*identification des doigts* ? Dans cette thèse et les travaux présentés, cette expression se définit par *la connaissance par le système du doigt à l'origine d'une interaction*. Sur un écran tactile, la majorité des interactions se font du bout des doigts. Ouvrir une application ou un menu se fait par simple pression du doigt sur l'écran et ce, peu importe le doigt utilisé. L'utilisation de l'index droit ou du pouce gauche n'affectera en rien le résultat de l'interaction. Avec l'*identification des doigts*, les écrans tactiles ne sont plus agnostiques aux doigts utilisés : nos dix doigts ne sont plus équivalents. À noter que cette définition ne s'applique pas uniquement à la modalité tactile. Identifier les doigts sur clavier

revient à ce que le système connaisse le doigt ayant appuyé sur une touche. Enfin, dans cette thèse, l'*identification des doigts* n'implique pas nécessairement la connaissance de l'individu qui interagit. Par exemple, les index gauches de deux utilisateurs peuvent être équivalents pour le système.

Deux autres expressions sont régulièrement utilisées tout au long du manuscrit : *multi-point* et *accord*. Un dispositif tactile *multi-point* est un dispositif qui est capable de détecter plusieurs contacts en même temps. Les écrans tactiles grand public sont très majoritairement multi-points (*ex. iPhones*, pavés tactiles *MacBook*, *etc.*). Un accord est une combinaison spécifique de doigts en contact simultanés sur un écran tactile. Un accord *pouce + index droit* signifie que le pouce droit et l'index droit sont en contact avec l'écran. Un accord peut être composé de deux doigts, trois doigts *etc.* jusqu'à dix doigts. Plus largement nous englobons aussi les accords à un doigt (*ex. l'index droit seul en contact avec l'écran*). L'expression *accord nul* est sporadiquement utilisée pour signifier qu'aucun doigt n'est en contact avec la surface tactile. Un dispositif tactile capable de distinguer les dix doigts d'un utilisateur est donc à même de reconnaître 1023 accords différents (1024 en comptant l'*accord nul*), chaque doigt ayant deux états possibles (en contact avec l'écran ou non). Pour plus de détails sur ces calculs, le lecteur peut se référer à la section 3.2.1.

Enfin, avant d'entrer dans le vif du sujet, il est pertinent d'avertir le lecteur sur le choix de vocabulaire qui a été fait. Bien que cette thèse soit écrite en français, nombre de termes ou d'expressions *classiques* dans le domaine de l'interaction homme-machine n'ont pas été traduits. Ce choix a été fait afin de ne pas alourdir la lecture. La traduction de ces termes aurait nécessité l'utilisation de formules longues, peu habituelles et/ou obscurcissant le sens des phrases. Ainsi les termes *smartphone*^{*} ou encore *feedback*^{*} ne sont pas traduits. Cependant tous sont indiqués par une astérisque (*) et sont répertoriés et définis dans un glossaire se trouvant à la fin du manuscrit.

Notation des configurations de doigts

Pour faciliter la lecture de cette thèse, nous utilisons la notation suivante pour décrire les différentes configurations de doigts. La syntaxe peut être décrite simplement avec les trois exemples représentatifs suivants : 1) *index* est la configuration utilisant uniquement l'index de la *main dominante* ; 2) *index + majeur* est l'*accord* formé par l'index et le majeur de la main dominante ; 3) *nd-index + nd-majeur* est l'accord formé par l'index et le majeur de la main *non-dominante*.

Dans la même optique, nous utilisons une notation plus concise dans les parties listant un grand nombre d'accords à la suite. Les notations P_d , I_d , M_d , A_d , L_d font références au pouce, à l'index, au majeur, à l'annulaire et à l'auriculaire² de la main dominante. Les notations P_n , I_n , M_n , A_n , L_n font références au pouce, à l'index, au majeur, à l'annulaire et à l'auriculaire de la main non-dominante. Les notations utilisées pour les accords ont une structure similaire : la notation

2. Nous utilisons L pour *little finger*, A et P étant déjà utilisées.

PI_nPM_d est utilisée pour décrire l'accord impliquant le pouce et l'index de la main non-dominante et le pouce et le majeur de la main dominante; la notation $LA_{\leftarrow I_{\rightarrow}}$ est utilisée pour décrire l'accord impliquant l'auriculaire et l'annulaire de la main gauche et l'index de la main droite.

Chapitre 1

Technologie

La technologie tient un rôle essentiel dans nos travaux. Bien qu'elle ne soit pas notre objet d'étude principal, il nous faut élaborer des prototypes afin d'explorer et évaluer l'utilisation de l'identification des doigts. Les solutions actuelles ne sont pas encore viables, suffisamment robustes et nécessitent pour la plupart d'équiper l'utilisateur. Cependant, nos prototypes souffrent des mêmes problèmes.

Pour nos expérimentations sur grandes surfaces, nous utilisons 5 *Gametraks*¹ (cf. Figure 1.1-gauche). Ces périphériques de jeu composés de rotules et de fils rétractables permettent de suivre le bout de chaque doigt dans un repère en trois dimensions. Sur tablette et *smartphone*^{*}, nous utilisons une webcam et des anneaux de couleurs (cf. Figure 1.1-droite). En combinant les informations de ces capteurs externes avec l'information tactile de leur écrans respectifs, nous sommes en mesure de fournir une identification des doigts relativement fiable.

Cependant, une technologie avec peu de latence et fiable en terme de reconnaissance est un point critique dans l'avancement de nos recherches. En effet, sans une telle technologie les comparaisons avec les techniques de l'état de l'art ne sont pas envisageables. Nous nous penchons donc aussi sur le développement de prototypes robustes et temps réel. Leur développement est encore en cours. Des premières versions peu coûteuses ont été réalisées et bien que nécessitant d'équiper l'utilisateur ils permettraient de répondre aux besoins identifiés, en attendant l'arrivée d'une technologie plus efficiente.

Cette technologie pourrait d'ailleurs avoir déjà vu le jour. Avec *Fiberio* [HB13], Holz *et al.*, identifient les utilisateurs interagissant avec une table tactile à l'aide de leurs empreintes digitales. En partant du principe que chaque doigt a une empreinte digitale unique, cette technologie pourrait fournir une solution d'identification des doigts qui s'intégrerait aux dalles tactiles d'aujourd'hui. *Fiberio* a cependant été développée à des fins de sécurité et nécessite donc que l'utilisateur pose son doigt bien à plat pour extraire les minuties d'une empreinte digitale large et claire. Or la partie des doigts interagissant avec les écrans tactiles n'entre pas dans ce cas idéal (*ex.* doigt orthogonal à la surface). Afin de vérifier si cette

1. <http://fr.wikipedia.org/wiki/Gametrak>

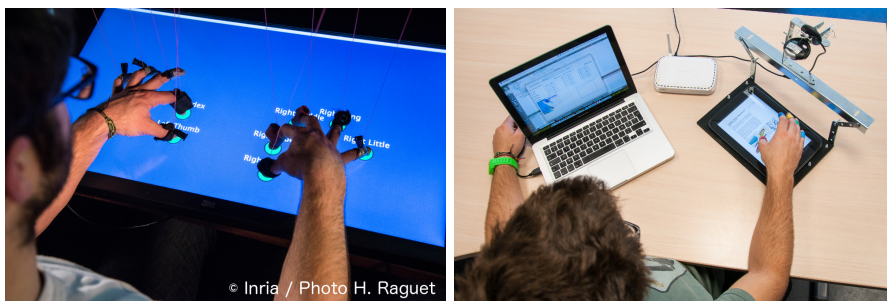


FIGURE 1.1: Technologies d'identification des doigts utilisées durant notre phase d'exploration : (gauche) sur grande surface utilisant des Gametraks ; (droite) sur tablette utilisant une webcam et des anneaux de couleur.

technologie peut être théoriquement utilisée, nous avons mené une étude visant à déterminer les différentes parties de la surface du bout de chaque doigt qui seraient susceptibles de porter suffisamment d'information pour les identifier de manière robuste.

Dans un premier temps, nous reviendrons sur le principe de fonctionnement des écrans tactiles et les informations couramment utilisées par les différentes applications. Nous proposerons ensuite un état de l'art sur les nouvelles informations visant à enrichir le vocabulaire tactile. Nous focaliserons cet état de l'art sur les solutions technologiques permettant d'obtenir ces nouvelles informations. Nous porterons une attention particulière à celles rendant possible l'identification des doigts. Nous détaillerons par la suite les différents prototypes implémentés au cours de nos recherches. Enfin nous nous proposerons d'étudier la viabilité de l'utilisation des empreintes digitales pour identifier les doigts sur les surfaces tactiles.

1.1 État de l'art

Dans cet état de l'art, nous nous intéressons aux différentes informations fournies par les écrans tactiles. Notre but est de donner une vue d'ensemble ainsi que de décrire les différents moyens technologiques utilisés pour les obtenir.

Un écran tactile est un périphérique informatique combinant à la fois dispositif de pointage, détectant les contacts physiques avec la surface, et moniteur. Le dispositif de pointage est superposé sur le moniteur ce qui permet la colocalisation des contacts et de leurs effets sur les éléments affichés par le système. Il existe différentes technologies pour détecter les contacts physiques comme : la technologie *capacitive*, la technologie *résistive* ou encore la technologie Frustrated Total Internal Reflection (*FTIR*)* infrarouge.

La plus répandue est la technologie *capacitive*. Elle est notamment utilisée par

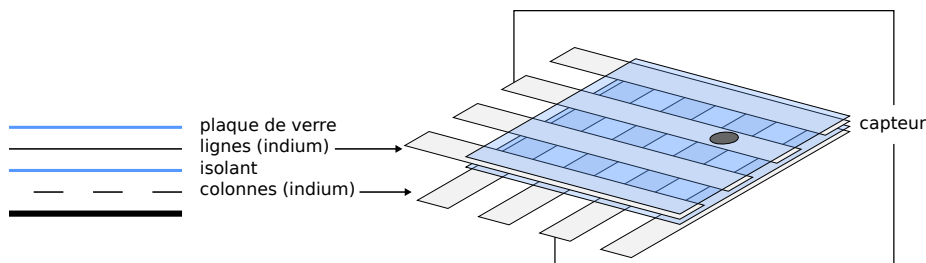


FIGURE 1.2: Principe de fonctionnement d’un écran tactile capacitif. Le dispositif est composé d’une couche de lignes et d’une couche de colonnes de métal transparent (*ex.* indium) électriquement chargées. Un isolant transparent est inséré entre les deux et le tout est recouvert d’une plaque de verre. Lorsqu’un contact est effectué sur la surface, les électrons des lignes et colonnes sont attirés et s’accumulent au niveau du contact. Des capteurs mesurent le changement de capacité au cours du temps de chaque croisement et déterminent s’il y a eu contact ou non.

la majorité des *smartphones** et des tablettes. Les écrans capacitifs sont constitués d’une couche de lignes et d’une couche de colonnes de métal transparent (*ex.* indium) électriquement chargées, isolées les unes des autres et placées sous la plaque de verre de l’écran. Lorsqu’il y a un contact (*ex.* surface du doigt) sur l’écran, une accumulation d’électrons sur la plaque de verre s’opère créant un déficit localisé dans les couches de métal transparent. Différents capteurs placés sur les bords de chaque ligne et colonne mesurent les changements de capacités aux différents croisements et localisent ainsi les points de contact (*cf.* Figure 1.2) [Wal12].

La technologie *résistive* est aussi basée sur la mesure de phénomènes électriques. Une couche de lignes et une couche de colonnes de métal transparent tenues distantes par de microscopiques cales d’espacement sont placées sur l’écran puis recouvertes d’un film isolant souple. Lorsqu’il y a un contact sur la surface, la pression exercée met en contact les couches de métal fermant ainsi un circuit électrique. Différents capteurs placés sur les bords de chaque ligne et colonne détectent quand le circuit se ferme et localisent ainsi les points de contact (*cf.* Figure 1.3) [Wal12].

Des technologies optiques comme la technologie *FTIR** infrarouge existent aussi. Cette dernière consiste à émettre des rayons infrarouges dans l’épaisseur de l’écran. Le milieu de l’écran agit comme un guide d’onde qui réfléchit les rayons tout le long de celui-ci. Lorsqu’il y a un contact sur l’écran, la réfraction des rayons se transforme en diffusion et une partie de l’infrarouge est émise vers l’extérieur. Cette lumière est captée par des caméras infrarouges. L’extraction des zones lumineuses de l’image permet ensuite de localiser les points de contact (*cf.* Figure 1.4) [Han05].

À intervalle de temps régulier, et ce quelque soit la technologie utilisée, le dispositif de pointage liste l’ensemble des points de contact actuels. Une mise en

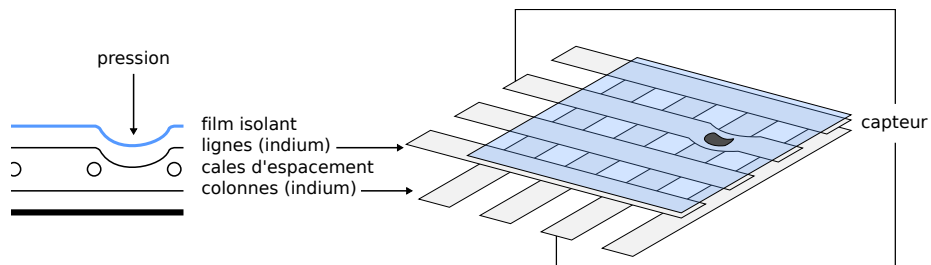


FIGURE 1.3: Principe de fonctionnement d'un écran tactile résistif. Le dispositif est composé d'une couche de lignes et d'une couche de colonnes de métal transparent (*ex. indium*). Des cales d'isolement tiennent à distance les deux couches et le tout est recouvert d'un film isolant souple. Lorsqu'une pression est effectuée sur la surface, les couches de métal entrent en contact et ferment le circuit. Des capteurs détectent quand le circuit se ferme et localisent les points de contact.

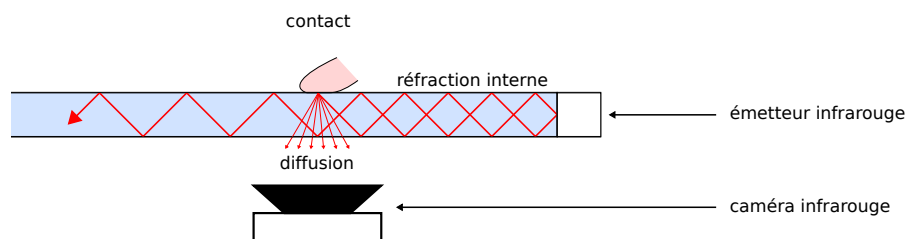


FIGURE 1.4: Principe de fonctionnement d'un écran tactile *FTIR** infrarouge. Le dispositif est composé de caméras infrarouges et d'émetteurs infrarouges. Ces derniers émettent des rayons qui se réfléchissent sur les parois internes de l'écran. Lorsqu'un contact est effectué sur l'écran, les rayons se diffusent et sont captés par les caméras infrarouges. L'analyse des images permet de déduire la positions du contact sur l'écran.

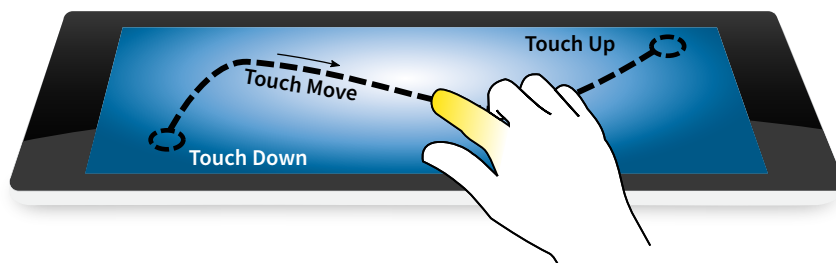


FIGURE 1.5: Événements tactiles transmis au système. Lorsqu'un contact vient d'être détecté par l'écran tactile, le système reçoit un événement *Touch Down**. À intervalle de temps régulier, le système reçoit des événements *Touch Move** qui décrivent la dernière position connue du contact. Lorsque le contact disparaît de la surface, le système reçoit un événement *Touch Up**.

relation est faite avec la précédente liste dans le but de détecter (1) les nouveaux contacts, (2) les déplacements ainsi que (3) les contacts disparus. L'ensemble des informations est ensuite transmise par le système aux différentes applications aux travers d'événements (1) *Touch Down**, (2) *Touch Move** et (3) *Touch Up**. Ces événements servent de briques de base à l'interaction. Par exemple, un événement *Touch Down** sur un bouton engendrerait un changement de couleur (*ex.* le bouton devient bleu pour simuler un enfoncement) et un événement *Touch Up** déclencherait la commande associée.

Aujourd'hui les interactions dans les produits commerciaux sont essentiellement basées sur trois types d'informations : un identifiant unique de contact permettant de suivre le mouvement au court du temps ; la position absolue de chaque contact sur l'écran ; le nombre de contacts simultanés sur l'écran. L'expressivité, *i.e.* les différents canaux physiques pour communiquer de l'information au système, d'un dispositif tactile est donc très limité. En faisant le parallèle avec la souris sur ordinateur de bureau, un dispositif tactile offre au plus dix curseurs. À noter que ces curseurs ne permettent pas le survol (*ex.* un contact physique sur l'écran tactile est un curseur cliquant en permanence) et ne peuvent pas être interprétés différemment (*ex.* une souris possède plusieurs boutons, le clavier peut aussi altérer l'interprétation des actions de l'utilisateur).

Les recherches se tournent donc vers l'augmentation de l'expressivité des écrans tactiles multi-points. Chris Harrison, chercheur à *Carnegie Mellon University* et co-fondateur de [Qeexo](http://www.qeexo.com/)², parle d'une nouvelle ère pour ces écrans : l'ère du *Rich Touch* (traduisible par *Contact Riche*). Ce domaine de recherche vise à collecter de nouvelles informations caractérisant plus finement les contacts physiques sur les dispositifs tactiles. Ces nouvelles informations ont pour avantage d'être orthogonales les unes aux autres. Ainsi l'orientation et la pression peuvent se cumuler sans entrer en conflit (*ex.* un doigt vertical peut soit exercer une pression forte soit une pression faible et cela suivant différentes orientations).

2. <http://www.qeexo.com/>

La suite de cet état de l'art a pour but de présenter une partie de ces nouvelles informations en mettant l'accent sur celle au cœur de cette thèse : l'identification des doigts.

1.1.1 L'ère du *Rich Touch*

Le *Rich Touch*, comme son nom l'indique, enrichit l'information transmise par le contact en fonction de ses caractéristiques temporelles, spatiales ou encore physiques.

1.1.1.1 Caractéristiques temporelles

Le caractère temporel d'un contact physique sur une surface tactile est depuis longtemps utilisé dans les produits commerciaux.

L'appui long est l'un des moyens les plus répandus pour différencier les contacts. Un appui long consiste à maintenir à la même position un contact physique avec la surface. La durée d'activation d'un appui long (*i.e.* temps écoulé entre le début du contact et la détection de l'appui long) varie selon le système d'exploitation. Sous *Android* et *iOS* la durée par défaut est de 0,5 s. Cette technique permet donc de doubler le cardinal de l'ensemble des entrées tactiles (*i.e.* expressivité) : chaque contact peut avoir deux états différents *court* ou *long*. Dans les systèmes commerciaux une utilisation typique de l'appui long permet d'afficher un menu contextuel (*i.e.* menu permettant d'accéder aux commandes copier et coller de l'application *Messenger* d'*Android*, cf. Figure 1.6). Cette nouvelle modalité souffre néanmoins de son principe même : l'appui long nécessite par définition d'interrompre le flux d'interaction de l'utilisateur. De même l'expressivité reste relativement limitée.

Une autre modalité basée sur le temps est aussi couramment utilisée : le double *tap*^{*}. Celui-ci consiste à effectuer rapidement deux contacts avec le même doigt et au même endroit à la manière d'un double clic. Dans les systèmes commerciaux une utilisation typique du double *tap*^{*} permet d'amorcer un zoom à un doigt (*i.e.* changement du niveau de zoom dans l'application *Maps* d'*Android*). De même que l'appui long, le double *tap*^{*} double l'expressivité qui reste encore une fois relativement limitée.

Ghomi *et al.* [GFH⁺12] introduisent eux le concept de séquence rythmique. À l'instar du *Morse*, cette technique propose à l'utilisateur d'effectuer des phrasés composés d'appuis courts et/ou d'appuis longs. Il est aussi possible de regrouper deux appuis longs consécutifs ainsi que d'effectuer des pauses. Les différentes *notes* rythmiques composant un phrasé sont jouées à un certain tempo. La longueur du phrasé définit donc le nombre de phrasés différents : 5 phrasés de longueur 2 ; 16 phrasés de longueur 3 ; 53 phrasés de longueur 4 ; 171 phrasés de longueur 5 ; 554 phrasés de longueur 6. Ici l'expressivité explose. En utilisant des phrasés de longueur 4 ou plus, cette modalité semble donc suffisante pour couvrir les commandes offertes par la plupart des applications³. Cependant, tout

3. Malacria *et al.* [MBH⁺13] ont trouvé entre 15 et 112 raccourcis claviers en étudiant 30 des applications les plus utilisées sous Mac OSX ($\mu = 55, 31, \sigma = 20, 53$)

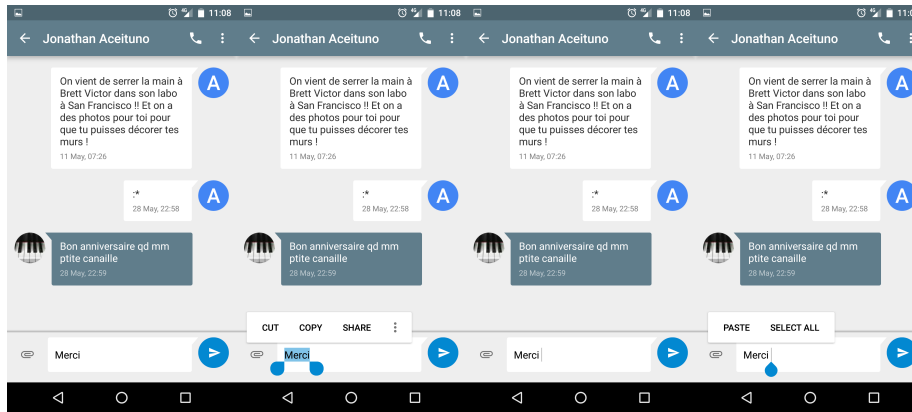


FIGURE 1.6: Illustration du mécanisme de copier-coller sur *Android messenger* : l'utilisateur effectue un appui long sur le mot à copier ; un menu contextuel s'affiche comprenant la commande *copier* (*copy* dans la figure) ; après avoir sélectionné la commande, l'utilisateur effectue de nouveau un appui long à la position où il souhaite coller le texte ; un autre menu contextuel s'affiche comprenant la commande *coller* (*paste* dans la figure).

comme l'appui long, le flux d'interaction de l'utilisateur se trouve interrompu.

1.1.1.2 Caractéristiques spatiales

Le caractère spatial d'un contact est lui aussi couramment utilisé sur surface tactile. Au travers du mot *spatial*, nous englobons à la fois la position sur l'écran ainsi que les mouvements des contacts. Pour être plus précis, les mouvements, puisque effectués au cours du temps, devraient être une caractéristique *spatio-temporelle*.

La position des contacts sur l'écran est l'information principale utilisée pour augmenter l'expressivité. En effet, la position du contact détermine avec quel *widget** affiché à l'écran l'utilisateur va interagir. Par exemple, lorsque celui-ci veut appuyer sur un bouton il lui suffit d'entrer en contact avec la surface à la position *ad hoc*. Plus il y a de *widgets**, plus l'expressivité augmente. Cependant plus il y a de *widgets** à l'écran, moins il y a de place disponible : les *widgets** doivent donc rétrécir (ce qui augmente le nombre d'erreurs) ou doivent être cachés (ce qui interrompt le flux d'interaction).

L'utilisation du mouvement des contacts est très répandue dans les systèmes commerciaux. Il existe d'ailleurs une liste de *gestes* populaires commune aux différents systèmes d'exploitation (*ex. flicks**, *pinch-and-expand**, *etc.*). Cette liste de gestes est un vocabulaire à part entière. La plupart peuvent s'effectuer aussi bien avec un doigt qu'avec 10. Cependant plus il y a de doigts qui prennent part à l'interaction plus elle devient problématique. En effet outre l'occultation de l'affichage par les doigts et les mains, ces gestes nécessitent bien souvent

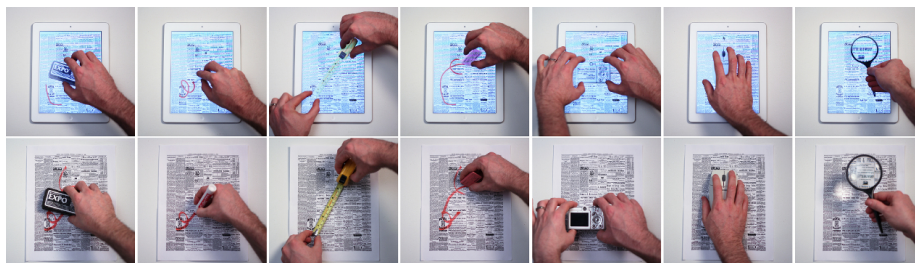


FIGURE 1.7: Gestes reconnus par *TouchTools* inspirés d'objets du quotidien [HXSH14].

que tous les contacts soient sur l'objet d'intérêt. Par conséquent, rarement les applications tactiles implémentent l'utilisation de gestes à trois doigts et plus.

En imitant sur une surface tactile la forme de la main manipulant certains objets du quotidien (*ex.* feutre, mètre, loupe, *etc.*), Harrison *et al.* [HXSH14] sont parvenus à augmenter la liste des gestes reconnus sur les écrans tactiles (*cf.* Figure 1.7). En analysant les caractéristiques géométriques entre différents contacts simultanés, *TouchTools* simule l'utilisation de l'objet *tenu* par l'utilisateur comme entrée du système. L'expressivité est augmentée d'autant de nouveaux gestes. Cependant le nombre de prises sur les outils physiques est limité, cette méthode ne pourra donc pas faire la différence entre des outils similaires.

Roudaut *et al.* [RLG09] introduisent aussi de nouveaux gestes les *MicroRolls*. Contrairement aux *flicks** nécessitant le déplacement du contact sur la surface, les *MicroRolls* s'effectuent en roulant le bout du doigt sur la surface. La position du doigt ne change quasiment pas. Cette différenciation possible entre les *flicks** et les *MicroRolls* vient augmenter le vocabulaire tactile de huit nouveaux gestes. Ces gestes ne sont néanmoins faisables facilement qu'à un doigt. Plus de contact rend difficile l'utilisation de la technique même si tous les doigts effectuent le même *MicroRoll*.

1.1.1.3 Caractéristiques physiques

Actuellement, c'est en utilisant certaines informations issues des caractéristiques physiques que l'expressivité est la plus augmentée. La plupart de ces informations sont inhérentes à chaque contact individuel et ne souffrent d'aucun délai d'activation. Par conséquent le vocabulaire se retrouve avec de nouvelles dimensions : son cardinal se retrouve multiplié et non simplement augmenté d'un nombre fixe d'éléments.

Dans le même esprit que *TouchTools* [HXSH14], Wigdor *et al.* [WBP⁺11] avec *Rock & Rails* utilisent la forme de la surface de contact pour activer différents quasi-modes d'interaction. Dans ce travail, les contacts ne sont plus simplement les doigts mais aussi le côté de la main sous trois formes différentes : poing fermé, main tendue et main courbée. Les quasi-modes sont activés par contact de la

main. Un contact effectué par un doigt est ensuite interprété différemment en fonction de la forme de la main. L’utilisation de la main n’est cependant pas adaptée à toutes les surfaces tactiles : l’écran devant être assez large, l’utilisation sur tablette et *smartphone*^{*} n’est donc pas indiquée.

Cao *et al.* [CWB⁺08] avec *ShapeTouch* utilisent l’aire de la surface de contact pour simuler la pression exercée par l’utilisateur. Ils distinguent deux types de pseudo-pression : faible et élevée. La pseudo-pression n’étant pas tout le temps corrélée à l’aire de la surface de contact (*ex.* la surface du pouce, plus grande par nature, engendrera une forte pseudo-pression) l’utilisateur se retrouvera contraint de faire attention à l’interprétation qu’aura le système.

Néanmoins la pression est une information riche et continue. Qu’elle soit tangentielle ou normale à la surface, elle ouvre de nouvelles possibilités d’interaction et commence à arriver sur les produits commerciaux (*ex.* iPhone⁴). Dans [HL11], Heo *et al.* entourent un *smartphone*^{*} d’un cadre équipé de capteurs de force. Le déplacement relatif entre le cadre et le *smartphone*^{*} leur permet de détecter la pression exercée par l’ensemble des contacts. Harrison *et al.* [HH12] utilisent eux deux joysticks entre le moniteur et le capteur capacitif et mesurent le déplacement entre les deux couches. Ce déplacement leur permet d’en déduire la force tangentielle. Malheureusement, ces technologies ne permettent pas de distinguer la pression associée à chaque contact quand ceux-ci sont effectués en même temps, mais les solutions à ce problème existent déjà, tout du moins pour des dispositifs de pointage indirect (*ex.* Sensel Morph⁵).

En étudiant l’image produite par le capteur capacitif d’un écran tactile, Xiao *et al.* [XSH15] sont à même d’estimer le tangage (*i.e.* *pitch*, *cf.* Figure 1.8) et le lacet (*i.e.* *yaw*, *cf.* Figure 1.8) du doigt lorsque celui-ci est en contact avec la surface. Chaque contact ayant la forme d’une ellipse, les calculs du ratio de la longueur des deux axes et l’orientation des axes leurs permettent d’en déduire l’orientation du doigt sans ajout de capteur particulier. L’orientation du doigt, caractéristique physique continue à deux dimensions, rejoint donc les informations augmentant l’expressivité tactile.

Avec *TapSense* [HSH11], Harrison *et al.* détectent la partie du doigt qui a causé un contact physique : la pulpe, le bout du doigt, l’ongle ou la jointure (*cf.* Figure 1.9). Leur technologie consiste à analyser le son produit par l’impact et trouver le meilleur candidat grâce à de l’apprentissage supervisé. Marquardt *et al.* [MKL⁺11] identifient aussi différentes parties de la main comme le bout des doigts, la paume, le dos de la main, le poignet et autres phalanges, en utilisant un gant bardé de marqueurs fiduciaux. Mais contrairement aux travaux précédents, son gant est aussi capable de différencier les doigts.

Bien que la pression, l’orientation et les différentes parties du doigt soient des caractéristiques faciles à contrôler pour un contact unique, il semble difficile de les contrôler séparément pour plusieurs contacts simultanés. Par exemple, placer un doigt à la verticale et un autre doigt à plat n’est pas chose aisée. À l’inverse, l’identification des doigts semble plus propice au contrôle fin. Notre

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

5. <http://www.sensel.com/>

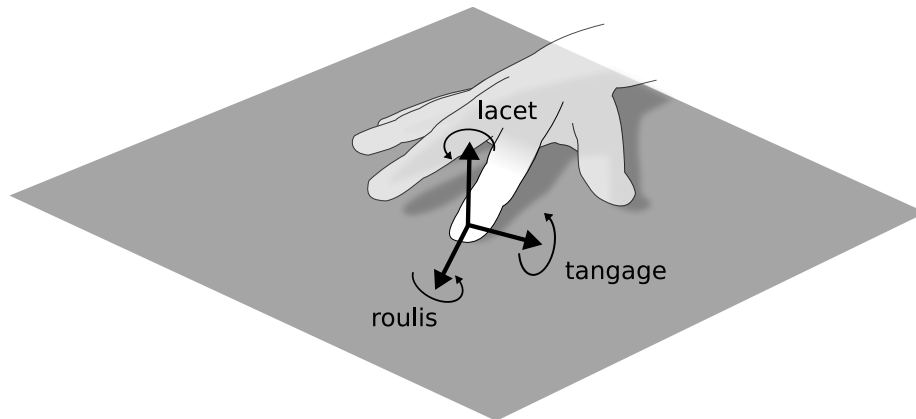


FIGURE 1.8: Description de l'orientation du doigt à l'aide des angles de tangage, de roulis et de lacet. En anglais, le tangage, le roulis et le lacet correspondent respectivement au *pitch*, au *roll* et au *yaw*.

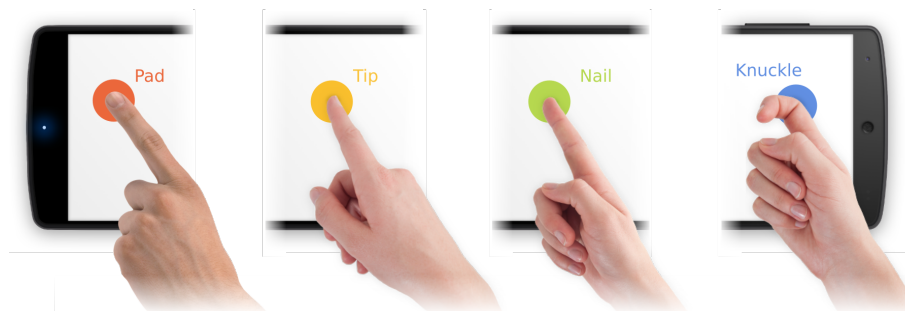


FIGURE 1.9: Illustration de *TapSense* [HSH11]. De gauche à droite, les différentes parties du doigt reconnues sont : la pulpe, le bout du doigt, l'ongle et la jointure.

dextérité nous permet d’utiliser n’importe quelle combinaison de doigts sans difficulté particulière.

1.1.2 L’identification des doigts

L’identification des doigts s’inscrit clairement dans l’ère du *Rich Touch*. En effet, en identifiant les doigts, l’expressivité explose : 1023 combinaisons de doigts étant possibles, 1023 éléments de vocabulaire tactile sont donc disponibles. De plus, cette information peut se combiner avec toutes les caractéristiques précédemment listées (sauf peut-être le nombre de contacts simultanés, cf. Section 3.2.3.4).

Les travaux qui se sont attachés à franchir l’obstacle technique de l’identification des doigts, ont utilisé des approches se basant sur : la simulation d’identifiants [GHB⁺13] ; l’analyse des relations géométriques entre les points de contact [GHB⁺13, AT10, EKF12, Wes99, LGF10, WLS14] ; l’utilisation de caméras traquant les mains [MRB05, MVER12, CH14, ZV16, Ibr16] ou les doigts à l’aide d’anneaux de couleur [WC04] ; l’utilisation de gants avec marqueurs fiduciaires [MKL⁺11] ; la mesure de distance entre les doigts et l’écran [GB16] ; la reconnaissance d’empreintes digitales [SK98] ; l’utilisation d’électromyogramme placés sur les avant-bras [BSMT09] ; ou encore l’utilisation de marqueurs Radio-Frequency IDentification (*RFID*)^{*} attachés aux doigts [MLH⁺16, VF13].

1.1.2.1 Émulation et géométrie des contacts

L’étude des caractéristiques géométriques entre les différents contacts est de loin la famille de solutions technologiques la plus explorée.

Ghomi *et al.* ont développé *Arpège* [GHB⁺13] : une technique de sélection de commandes invitant l’utilisateur à construire son accord progressivement. Pour sélectionner une commande, l’utilisateur invoque *Arpège* en effectuant un double *tap*^{*} sur l’écran, en appuyant sur un bouton spécifique ou en appuyant sur l’écran avec une main détendue. Une fois *Arpège* invoqué, des cercles représentants les différents doigts sont affichés. À l’aide de *feedforward*^{*}, l’utilisateur n’a plus qu’à placer les doigts composant l’accord correspondant à la commande voulue un par un dans les cercles adéquats. À noter que quand l’utilisateur place la main détendue sur l’écran, le système place les cercles sous les doigts et déduit une identification de chacun d’entre eux. Cette identification est basée sur les résultats d’une phase de calibration où l’utilisateur place sa main détendue sur l’écran offrant au système la possibilité de s’adapter à sa morphologie.

Dans sa thèse, Westerman [Wes99] propose d’identifier les doigts et les mains analysant les relations géométriques entre les différents points de contacts. Son but est de segmenter les différents doigts et les contacts causés par la paume des mains. L’algorithme utilisé s’appuie sur des diagrammes de *Voronoi*. Dans [AT10], Au *et al.* identifient les doigts lorsque cinq contacts ont été détectés. Après avoir calculé le barycentre des contacts, ils comparent les angles entre les différents segments contact/barycentre. Les deux angles les plus grands ayant pour contact commun le pouce, leur algorithme identifie le pouce en premier.

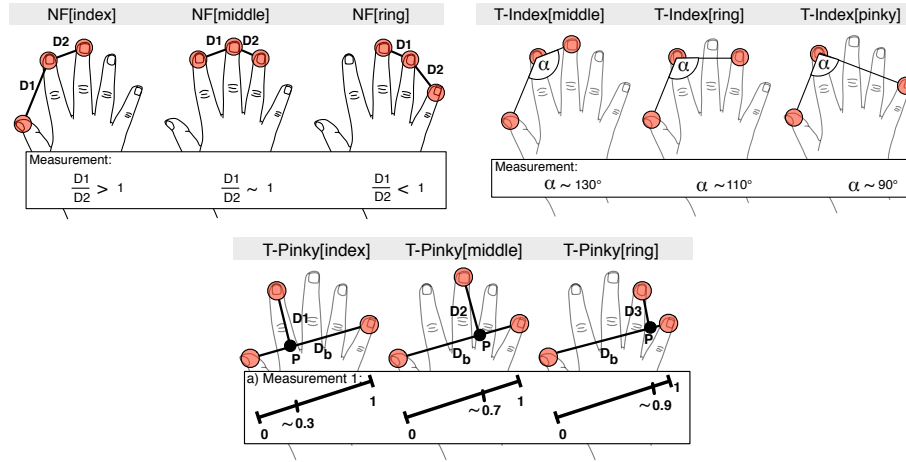


FIGURE 1.10: Relations géométriques utilisées par Wagner *et al.* pour identifier des accords à 3 doigts [WLS14]. Dans l'ordre de lecture, les différentes familles identifiées sont : la famille des *doigts voisins*, la famille *pouce index* et la famille *pouce auriculaire*.

Les doigts restant sont identifiés en suivant simplement l'ordre des doigts de la main. Ces deux méthodes nécessitent le placement sur l'écran des cinq doigts en même temps.

Wagner *et al.* [WLS14] se basent aussi sur les relations géométriques entre les points de contact pour identifier des accords à 3 doigts (*cf.* Figure 1.10). En tout, ils identifient 3 familles d'accord à 3 doigts : la famille des *doigts voisins* (pouce + index + majeur, index + majeur + annulaire, majeur + annulaire + auriculaire) en analysant le ratio des distances entre les contacts ; la famille *pouce index* (pouce + index + majeur, pouce + index + annulaire, pouce + index + auriculaire) en analysant l'angle formé par les contacts ; la famille *pouce auriculaire* (pouce + index + auriculaire, pouce + majeur + auriculaire, pouce + annulaire + auriculaire) en analysant les projetés orthogonaux. Bien que très robuste, cette méthode ne permet de détecter que des accords à 3 doigts.

Ewerling *et al.* [EKF12] utilisent l'image infrarouge fournit par certaines surfaces tactiles. En effet, l'image infrarouge ne comporte pas uniquement une information binaire (contact ou pas), elle encode aussi une information sur les objets à proximité de la surface. Ainsi quand une main est proche, ses contours peuvent être visibles. Grâce à cette *ombre*, Ewerling *et al.* extraient les contours de la main et en déduisent le bout de chaque doigt. Enfin en analysant les caractéristiques géométriques à la manière des travaux précédents, ils en déduisent l'identification de chaque doigt. Lepinski *et al.* [LGF10] utilisent une méthode similaire. En se basant sur l'image infrarouge capturant la main à proximité de la surface, ils déduisent la boîte englobant la main (*cf.* Figure 1.11). En plaçant ensuite les contacts dans cette boîte, ils sont capables de reconnaître

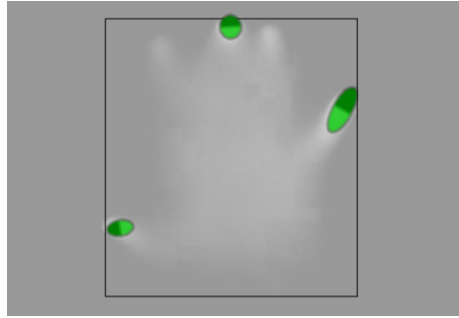


FIGURE 1.11: Exemple de boîte englobant la main utilisée par Lepinski *et al.* dans [LGF10] pour identifier 14 accords.

de façon robuste 14 accords : 3 accords à un doigt ; 5 accords à deux doigts ; 3 accords à trois doigts ; 2 accords à quatre doigts ; 1 accord à cinq doigts. Ces méthodes nécessitent cependant que la main soit relativement plate pour que le contour ou la boîte englobante soient utilisables.

1.1.2.2 Caméra

Une autre approche populaire consiste à utiliser des caméras externes filmant les mains des utilisateurs.

En 2005, Malik *et al.* [MRB05] proposent un système de contrôle à distance pour mur d’écrans. Ils utilisent une caméra RGB pour filmer, en vue de dessus, la surface d’un bureau où les utilisateurs effectuent différentes postures de main. Leur système segmente les images de la caméra pour reconnaître des postures pré-enregistrées. Une fois la posture courante reconnue, ils sont capables d’identifier les différents doigts visibles. Murugappan *et al.* [MVER12] reprennent le même principe. Grâce à une caméra de profondeur, ils reconnaissent non seulement différentes postures de mains mais aussi les points de contact avec la surface filmée. Leur système n’est pas un contrôle à distance mais une solution de surface tactile. En plus de la caméra de profondeur, un projecteur surplombe la surface de travail. De même, Kung *et al.* [KKS⁺12] utilisent une caméra de profondeur *Kinect* et un algorithme cherchant à satisfaire un modèle articulaire de la main. La caméra est placée au dessus d’un grand écran tactile. Ils utilisent ce système pour interagir dans une application de construction de scènes en trois dimensions pour film d’animation. Dans ces trois travaux, il faut cependant que la posture soit reconnue, dans le cas contraire aucun doigt n’est identifié.

Dans [Ibr16], Ibraheem identifie les doigts à l’aide d’apprentissage supervisé. Après avoir segmenté le contour de la main placée devant un fond noir, il extrait différentes caractéristiques pour chacun des doigts telles que : la distance entre le doigt et l’axe de la main, l’angle formé entre le doigt et l’axe de la main ou encore les angles formés avec les autres doigts. La reconnaissance s’effectue ensuite à l’aide d’un modèle gaussien mixte. Dans ce travail, la main n’a pas obligation

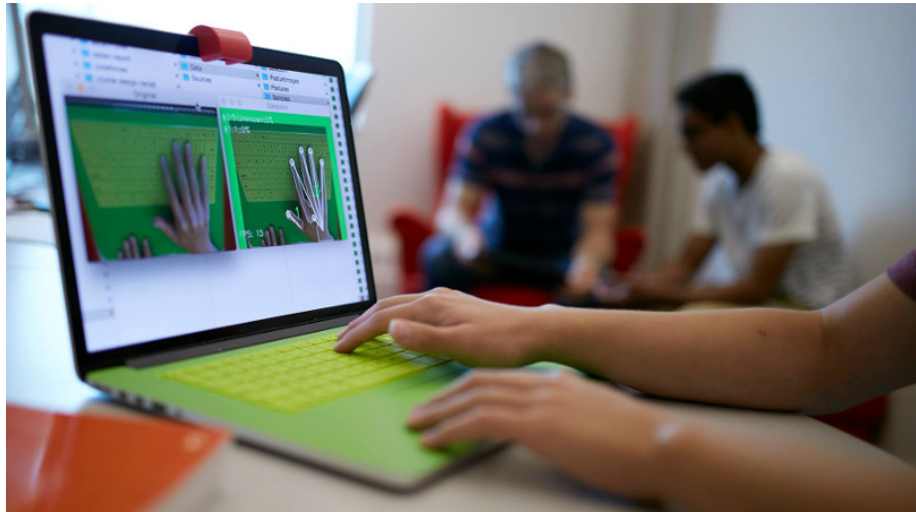


FIGURE 1.12: Dispositif utilisé par Zheng *et al.* dans [ZV16] pour identifier les doigts. Un miroir dévie le champ de vue de la caméra vers le pavé tactile. Le flux vidéo est ensuite analysé en temps réel pour déduire l'identification des différents doigts.

d'être plate, cependant chaque pose à reconnaître doit être préalablement apprise par le système.

Colley *et al.* [CH14] utilisent eux un *Leap Motion*⁶ pour identifier les doigts. Le *Leap Motion* est un appareil de la taille d'une clef USB intégrant deux caméras infrarouges. Grâce aux deux flux d'images et à des résolutions de contraintes structurelles, ce dispositif reconstruit un modèle en trois dimensions de la main. Chaque doigt est donc identifié. En plaçant le *Leap Motion* sur le bord d'un *smartphone*^{*}, Colley *et al.* identifient le doigt à l'origine du contact sur l'écran. À noter que leur système ne permet d'interagir qu'avec un seul doigt à la fois.

Enfin, bien que leur technique d'identification des doigts soit utilisée pour augmenter les raccourcis claviers, Zheng *et al.* [ZV16] utilisent un miroir déviant le champ de la caméra intégrée d'un ordinateur portable (*cf.* Figure 1.12). La caméra capture la zone du clavier et du pavé tactile toutes deux recouvertes d'un revêtement vert facilitant l'extraction des mains de l'image. Après avoir extrait le contour des mains, ils repèrent la position du centre du poignet et calculent les maximums locaux entre ce centre et les points du contour. Une fois ces maximums trouvés, ils identifient les doigts. Contrairement aux techniques précédentes, il n'y a pas d'apprentissage de posture et les doigts peuvent être identifiés même avec le poing partiellement fermé. Cependant, les mains doivent rester relativement plates.

6. <https://www.leapmotion.com/>

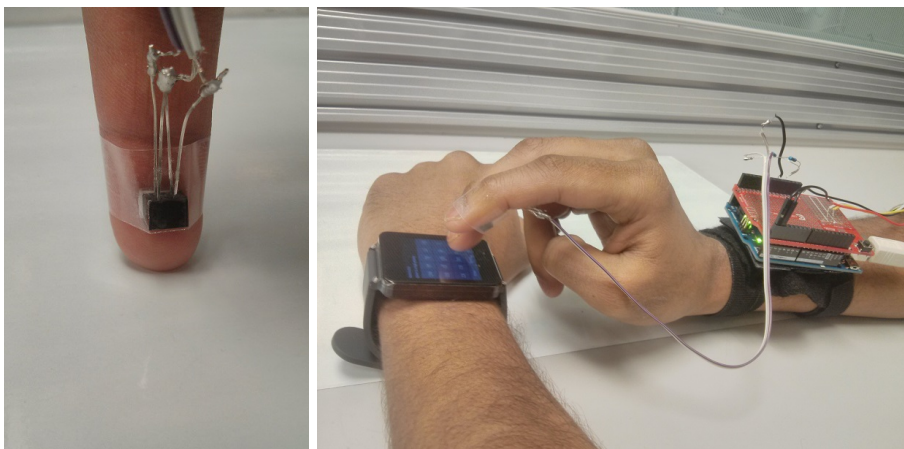


FIGURE 1.13: Dispositif utilisé par Gupta *et al.* dans [GB16] pour identifier les doigts. Des caméras *temps de vol* sont attachées aux différents doigts. La distance entre le doigt et l'écran permet de déduire quel doigt interagit avec l'écran.

1.1.2.3 Instrumentation de la main

Équiper les utilisateurs est une approche plus robuste que les solutions précédentes mais aussi plus contraignante.

Gupta *et al.* [GB16] utilisent des caméras *temps de vol* pour mesurer la distance entre les doigts et l'écran (*cf.* Figure 1.13). Afin d'améliorer la saisie de texte sur montre tactile, ils attachent sous la pulpe de l'index et du majeur des capteurs capables de mesurer la distance entre la caméra et le premier obstacle. Lorsqu'un contact sur l'écran est effectué, le doigt le plus proche de l'écran est associé. Dans leurs travaux, seuls deux doigts sont identifiables mais le concept pourrait être généralisé. Par contre, en plus d'équiper l'utilisateur, l'orientation du doigt sur l'écran devient (raisonnablement) contraint (*ex.* impossibilité d'avoir le doigt à plat).

Avec *FingerSense*, Wang *et al.* [WC04] placent des marqueurs de couleurs différentes sous la pulpe de chacun des doigts. Chaque couleur correspond à un doigt différent. Leur approche est utilisée pour des boutons physiques équipés de capteur RGB. En fonction du doigt reconnu, le bouton déclenche une commande distincte. Bien que simple, le principe est difficilement réalisable sur des surfaces tactiles classiques qui ne peuvent reconnaître les couleurs. Marquart *et al.* [MKL⁺11] s'emparent de la méthode et l'adaptent aux surfaces capables de fournir une image infrarouge. En lieu et place de marqueurs de couleurs, ils utilisent des marqueurs comportant des images binaires à la manière des codes *Quick Response* (*i.e.* *QR Code*). Ces marqueurs sont attachés à des gants que l'utilisateur doit revêtir (*cf.* Figure 1.14). Comme expliqué précédemment, non seulement leur système identifie les doigts mais aussi différentes parties de la main. Les aspects robustesse et fiabilité de ces méthodes sont indéniables. Cependant

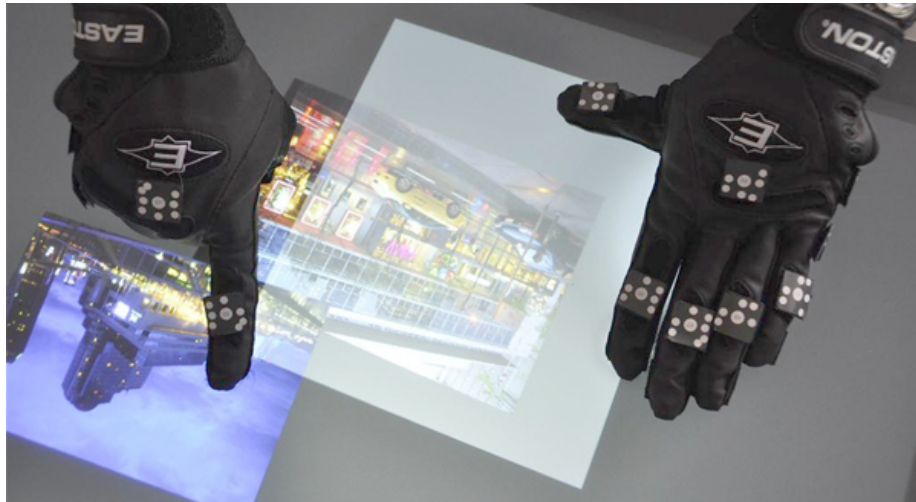


FIGURE 1.14: Gants utilisés par Marquardt *et al.* dans [MKL⁺11] pour identifier les doigts.

ils nécessitent un capteur tactile adapté (*ex.* l'utilisation sur les *smartphones** et tablettes grand public est impossible) et surtout ils recouvrent la pulpe du doigt ce qui altère la perception de l'utilisateur lors de l'interaction.

Dans [MLH⁺16], Ma *et al.* utilisent des marqueurs *RFID** dissimulés dans des gants pour doigts. En utilisant un écran intégrant lecteurs et antennes *RFID** sous la surface d'interaction, ils sont capables d'identifier les différents doigts en contact. Avec cette méthode, les doigts sont entièrement recouverts. De plus, un lecteur *RFID** renvoie une information de présence et non de position. Le placement et la taille des antennes doit être étudié précisément pour éviter : une trop grande profondeur de champ de captation pouvant détecter les doigts adjacents même si ceux-ci ne sont pas en contact ; une trop grande coupe de champ de captation rendant impossible la désambiguïsation de contacts simultanés. Sur le même principe, Vega *et al.* utilisent eux des marqueurs *RFID** dissimulés dans de faux ongles en plastique [VF13]. Contrairement au travail précédent, ils utilisent cette technologie pour interagir avec des vêtements intelligents dans lesquels sont intégrés les lecteurs et antennes *RFID**. Bien que leur méthode a pour avantage de ne pas recouvrir la pulpe, équiper l'utilisateur de faux ongles est une contrainte forte.

1.1.2.4 Capteurs biométriques

Une dernière grande famille de solutions technologiques porte sur l'analyse de données biométriques.

En plaçant un électromyogramme (*i.e.* capteur sensible aux impulsions électriques du système nerveux) sur l'avant bras des utilisateurs, Benko *et al.* [BSMT09] sont capables de reconnaître les doigts qui ont été déplacés par les

muscles. La solution demande d'équiper les utilisateurs d'un bracelet non invasif et ne recouvre pas la pulpe des doigts. Cependant, la méthode est fiable si un seul doigt est déplacé à la fois.

Sugiura *et al.* [SK98] exploitent une autre information biométrique unique à chaque doigt : les empreintes digitales. À la manière de Wang *et al.*, ils utilisent un lecteur d'empreinte digital extérieur comme bouton physique et déclenchent une action différente en fonction de l'empreinte. Cette solution élégante ne nécessite pas d'équiper l'utilisateur. Cependant l'adaptation de la technique sur des écrans tactiles reste un problème ouvert (*cf.* Partie 1.3).

1.1.3 Des technologies pas encore matures

Le *Rich Touch* est en passe de révolutionner l'ère des écrans tactiles multi-points. Le vocabulaire d'interaction, jusque là limité à quelques dizaines de lemmes (*i.e.* élément constituant le vocabulaire), se retrouve démultiplié par la plupart des nouvelles informations prises en compte (*ex.* pression, orientation, partie du doigt en contact, *etc.*). De plus, le point fort de ces nouvelles informations est qu'elles ajoutent des dimensions à l'espace des entrées : ces informations peuvent être combinées et n'entrent pas nécessairement en compétition les unes avec les autres. Un contact pourra par exemple s'effectuer avec un doigt à la vertical et exerçant une forte pression. Un autre contact pourra véhiculer au système une information différente si celui-ci est à l'horizontal et exerce une pression tout aussi forte que dans l'exemple précédent.

L'identification des doigts fait partie de l'ère *Rich Touch*. En discriminant les doigts, le vocabulaire passe d'une dizaine de lemmes à 1023. Toutes les combinaisons de doigts peuvent transmettre une information différente. Cependant à l'inverse des autres informations, les solutions technologiques ne sont pas encore matures. Nous estimons néanmoins que cette information deviendra une fonctionnalité standard des écrans multi-points grand public.

Dans la suite, nous détaillerons les prototypes implémentés au cours de nos recherches. Ces prototypes souffrent des mêmes problèmes que ceux de l'état de l'art. Ils ont cependant l'avantage d'être robustes et fiables nous permettant ainsi d'explorer les possibilités offertes par l'identification des doigts. Nous présenterons de même la solution que nous estimons la mieux placée pour résoudre le défi technologique posé par l'identification des doigts.

1.2 Prototypage

Le but de nos recherches est d'étudier l'impact sur l'utilisateur de l'identification des doigts ainsi que d'explorer cette information au travers de nouvelles interactions tactiles. Nous ne nous attachons pas à résoudre le défi technique posé par l'identification des doigts. Cependant, nos recherches ont engendré le développement de prototypes permettant une identification des doigts fiable sur

surfaces tactiles de différentes tailles : un prototype pour grande surface et un prototype pour tablette et *smartphone*^{*}.

Tous nos prototypes fonctionnent sur le même principe que certaines technologies de l'état de l'art : instrumenter les différents doigts de l'utilisateur pour faciliter le suivi de leur position. Les marqueurs fiduciaux utilisés par Marquardt *et al.* [MKL⁺11] sont un exemple d'instrumentation, bien qu'ils ne soient pas compatibles avec des écrans tactiles capacitifs.

1.2.1 Architecture

L'architecture fonctionnelle de tous nos prototypes est la même (*cf.* Figure 1.16). Au centre de l'architecture se trouve un serveur d'événements qui gère la mise en relation entre les événements tactiles (*i.e.* *Touch Down*^{*}, *Touch Move*^{*} et *Touch Up*^{*}), provenant de l'écran multi-points, et les informations de positions des différents doigts, provenant de notre (nos) capteur(s). À chaque instant, ce serveur met à jour la position de l'ensemble des doigts. Lorsqu'un événement tactile est reçu, l'ensemble des contacts est analysé en fonction de leur identifiant unique : (1) pour un nouveau contact (*i.e.* *Touch Down*^{*}), le serveur affecte au contact l'identifiant du doigt le plus (au sens de la distance euclidienne, *cf.* paragraphe suivant) parmi les doigts couramment disponibles et enregistre cette mise en relation ; (2) pour un contact qui a déjà été mis en relation avec un doigt (*i.e.* *Touch Move*^{*}), le serveur récupère l'identifiant du doigt précédemment associé et le réaffecte au contact ; (3) pour un contact en fin de vie (*i.e.* *Touch Up*^{*}), le serveur réaffecte l'identifiant au contact comme dans (2) et rend disponible l'identifiant du doigt pour la prochaine étape (1). Une fois l'ensemble des contacts augmentés avec le doigt correspondant, cette liste est transmise à l'application.

Pour estimer le doigt le plus proche du contact, il nous faut transformer les positions récupérées par les capteurs dans le référentiel de l'écran tactile. Si les capteurs fournissent une position en trois dimensions, nous associons une altitude arbitraire à l'écran tactile. Ce changement de repère est effectué à l'aide d'une matrice de passage M déterminée lors d'une phase de calibration. La phase de calibration demande à l'utilisateur de sélectionner des cibles dont les positions P_{ecran} à l'écran sont connues. À chaque acquisition de cible, la position du doigt P_{doigt} ayant effectué la sélection est enregistré. La forme d'une matrice de passage étant connue, ces acquisitions constituent les conditions initiales pour la résolution de l'équation. La résolution du système nous permet donc de trouver une matrice M satisfaisant $MP_{doigt} = P_{ecran}$ pour l'ensemble des acquisitions. Le calcul de la distance entre un contact et un doigt revient donc à calculer la distance entre $P_{contact}$ et MP_{doigt} . Une matrice de passage par capteur identifiant les doigts est nécessaire.

Cette architecture comporte deux grands avantages. Le premier avantage est qu'elle garantit que l'identifiant du doigt associé à un contact soit le même tout au long de la vie du contact. Lorsqu'un utilisateur interagit avec l'application, le comportement du système sera cohérent au cours du temps et non pas variable. Bien sûr si un contact est mal identifié, l'utilisateur est contraint de lever son

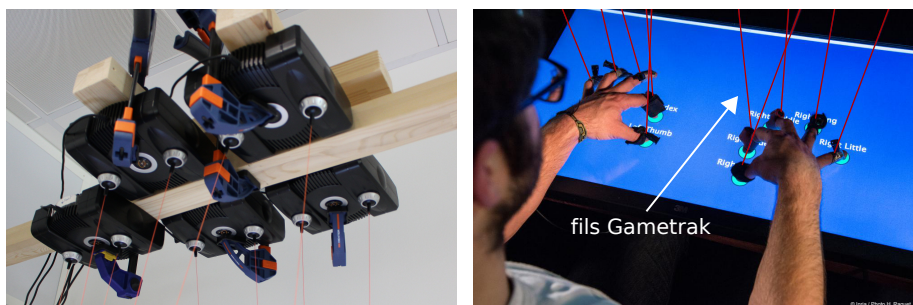


FIGURE 1.15: Prototype d'identification des doigts sur grandes surfaces : (gauche) cinq *Gametraks** placés à la verticale de l'écran ; (droite) fils des *Gametraks** attachés sur la première phalange de chaque doigt.

doigt et de le reposer afin de changer l'identifiant trouvé. Le deuxième avantage est que cette architecture rend l'application agnostique à la technologie utilisée. Les capteurs tactile et d'identification des doigts sont donc faciles à changer.

1.2.2 Sur grandes surfaces

Les capteurs d'identification des doigts sont cinq *Gametraks**. Ces dispositifs ont été à l'origine conçus pour suivre le mouvement de club de golf sur *Playstation 2*. Les *Gametraks** sont chacun composés de deux rotules et de deux fils rétractables. Chaque rotule possède trois potentiomètres : deux potentiomètres mesurent l'orientation de la rotule et un dernier mesure l'extension du fil sortant de la rotule. En combinant les trois mesures, il est possible de retrouver les coordonnées en trois dimensions du bout de chaque fil. Un *Gametraks** permet donc de suivre la position de deux points dans l'espace. La récupération des coordonnées s'effectue grâce à la bibliothèque *libgametrak*⁷ à une fréquence de 125 Hz.

Dans ce prototype, les *Gametraks** sont placés à la verticale de l'écran et leurs fils sont attachés sur la première phalange de chaque doigt à l'aide d'anneaux en velcro (cf. Figure 1.15). Il est à noter que les fils et les anneaux produisent une certaine gêne visuelle, mais qui reste cependant très minime.

Une fois captées, les positions dans l'espace des différents doigts sont transformées dans le référentiel d'un écran 32" 3MTM C3266PW multi-touch posé à l'horizontal sur un bureau. Cette transformation est réalisée à l'aide de matrices de passage calculées lors d'une courte phase de calibration, comme précédemment exposé. Ce système d'identification des doigts envoie aux applications clientes des événements *TUIO** augmentés d'identifiants correspondant aux doigts ayant causés chaque contact.

Le serveur d'événements, l'application et les différents capteurs (incluant l'écran tactile) s'exécutent sur un même ordinateur.

7. <https://github.com/casiez/libgametrak>

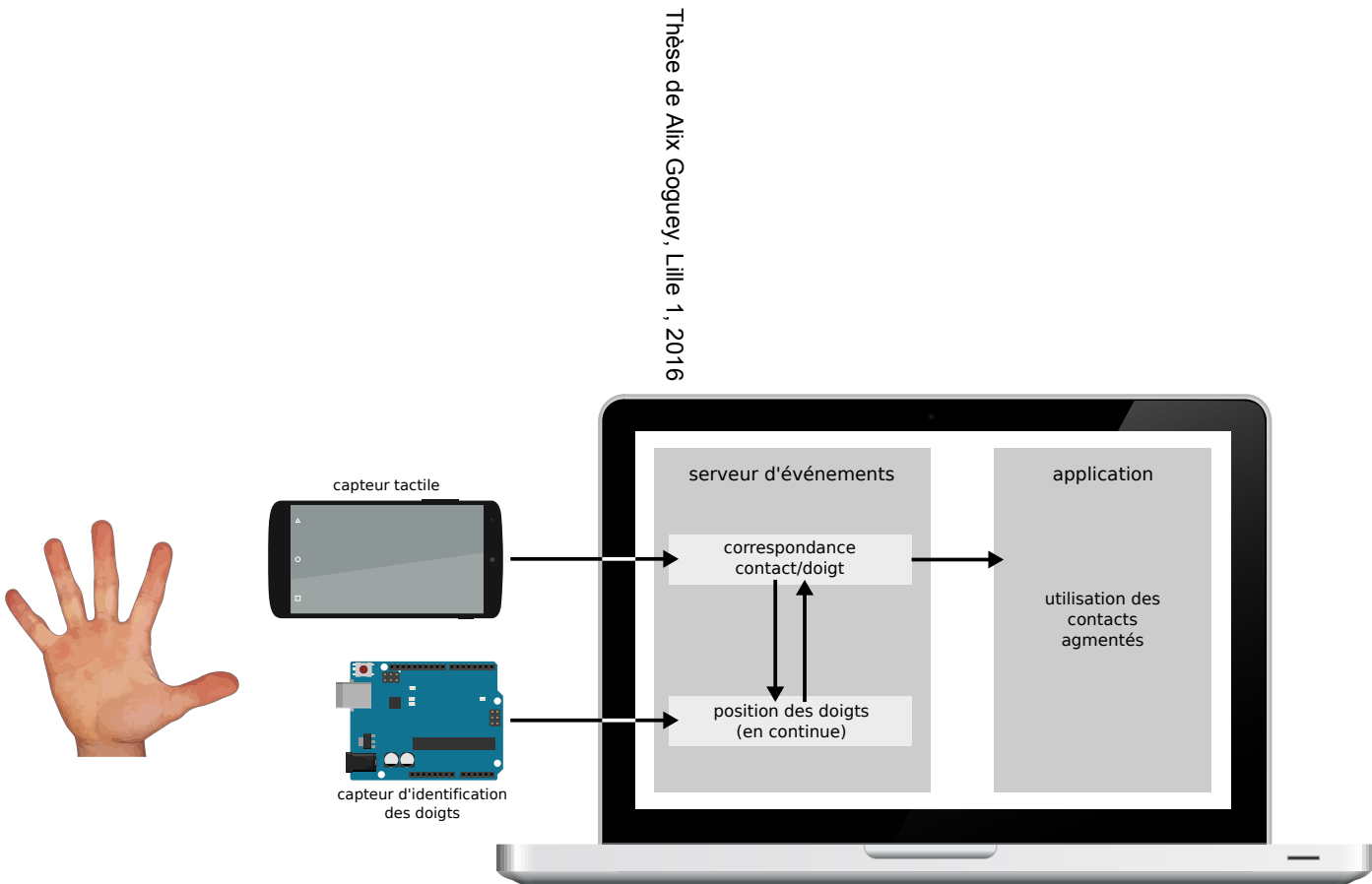


FIGURE 1.16: Schéma de l'architecture fonctionnelle des prototypes développés.

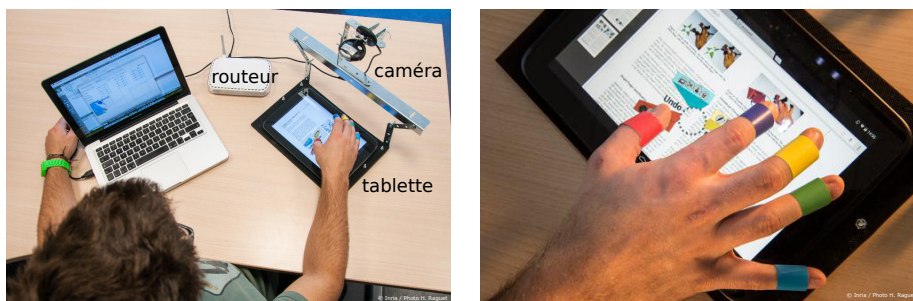


FIGURE 1.17: Prototype d'identification des doigts pour tablette et *smartphone** : (gauche) une caméra surplombe l'écran de la tablette ou du *smartphone** ; (droite) un anneau de couleur est attaché au bout de chaque doigt.

1.2.3 Sur tablettes et smartphones

Notre prototype d'identification des doigts pour grande surface utilisant des *Gametraks** supposait un écran fixe. Cette dernière condition n'est pas respectée lors de l'utilisation d'une tablette ou d'un *smartphone**. À la place de ces capteurs, nous utilisons des anneaux de couleur (ou comportant des *QR code*) et une caméra RGB 640 px × 480 px surplombant la surface (cf. Figure 1.17). Les anneaux sont placés sur la première phalange de chaque doigt. La position de chaque doigt est donnée dans le référentiel en deux dimensions de l'image RGB. La récupération des coordonnées s'effectue grâce à la bibliothèque *OpenCV*⁸ à une fréquence de 20 Hz. Cette méthode est similaire à celle de Wang et Canny [WC04]. Elle permet de repérer de façon fiable jusqu'à 6 doigts. Avec ce prototype, nous repérons les 5 doigts de la main dominante et le pouce de la main non-dominante.

Comme pour le prototype sur grand écran, les positions des doigts sont transformées dans le référentiel de l'écran de notre tablette 10" ou de notre *smartphone** 4". Cette transformation est réalisée à l'aide d'une matrice de passage calculée lors d'une courte phase de calibration, comme précédemment exposé. De même que pour notre prototype pour grande surface, ce système d'identification des doigts envoie aux applications clientes des événements *TUIO** augmentés des identifiants des doigts.

Le serveur d'événements et le capteur d'identification des doigts fonctionnent sur un même ordinateur. La transmission des informations de l'écran tactile vers le serveur et la transmission des événements *TUIO** augmentés vers l'application s'effectue *via* un routeur *Wi-Fi*.

1.2.4 Les autres pistes explorées

Les prototypes précédents ne sont pas les seules pistes que nous avons explorées. Nous nous sommes en effet penchés sur le développement de deux

8. <http://opencv.org/>

solutions que nous voulons économiques, fiables et temps réel. Ces solutions sont toutefois toujours en cours de développement et de test. Nous présentons donc ici leur principe général de fonctionnement.

La première solution, inspirée de [HSP12], repose sur la mesure de phénomènes électrostatiques. L'idée est de mesurer une capacité entre un capteur placé sur chacun des doigts et une grille conductrice placée sur l'écran. Les capacités mesurées sont proportionnelles à la distance entre les capteurs et la grille. Plus la distance est faible plus la capacité mesurée est grande. Cette propriété nous permet de distinguer de façon robuste la hauteur de chacun des doigts. À chaque nouveau contact (*i.e.* *Touch Down*^{*}) est associé le doigt libre le plus proche (*cf.* Section 1.2.1). Les avantages de ce système sont sa précision et son taux de rafraîchissement élevé (*i.e.* environ 100 Hz). De plus, il est utilisable sur *smartphone*^{*}, tablette et grand écran.

La seconde solution repose elle sur la mesure d'accélérations grâce à des accéléromètres placés sur chacun des doigts. Un accéléromètre fournit un vecteur accélération sur trois axes. Ces trois composantes nous permettent de déduire plusieurs caractéristiques comme la norme et l'orientation du doigt. En étudiant les signaux, nous cherchons à reconnaître la signature du contact d'un doigt sur une surface. Cette solution a pour avantage de ne pas modifier la surface et pourrait potentiellement être utilisée pour identifier les doigts dans d'autres contextes (*ex.* sur clavier).

Nos prototypes ont été développés pour nous permettre d'étudier et d'explorer l'utilisation de l'identification des doigts. Comme toutes les technologies présentées dans l'état de l'art, ce sont des outils de recherche et qui seraient difficiles à intégrer dans des produits commerciaux. Dans la suite nous présentons une solution technologique qui pourrait-elle être utilisée par le grand public.

1.3 Utilisation des empreintes digitales

Les travaux décrits dans cette section ont donné lieu à la soumission d'un papier long à la conférence CHI 2017.

Une condition impérative pour la démocratisation de l'identification des doigts sur nos écrans tactiles est d'avoir une technologie fiable et qui ne nécessite pas d'équiper l'utilisateur. Jusqu'à présent, l'ensemble des prototypes présentés ne satisfait pas cette condition : certains prototypes demandent à l'utilisateur d'enfiler des gants ou d'attacher des marqueurs aux doigts, alors que les autres se reposent sur l'utilisation de capteurs externes tels que des caméras rendant impossible une utilisation nomade. Il existe cependant une technologie qui nous apparaît à même de résoudre le défi technologique posé par l'identification des doigts : la reconnaissance des empreintes digitales.

En 2013, Holtz *et al.* ont conçu *Fiberio* [HB13], une table tactile capable d'identifier les utilisateurs interagissant sur la surface à l'aide de leurs empreintes digitales. *Fiberio* est composé d'une plaque de fibres optiques en dessous de

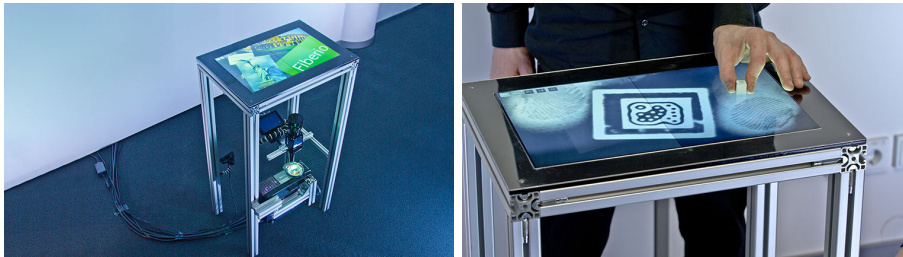


FIGURE 1.18: Dispositif *Fiberio* : cette table tactile est constituée d'une plaque de fibre optique, d'un vidéoprojecteur et d'une caméra haute résolution.

laquelle est située un vidéo-projecteur et une caméra haute résolution (cf. Figure 1.18). Lorsque l'utilisateur place son doigt à plat sur l'écran, les propriétés optiques de la plaque permettent à la caméra de capturer une image contrastée de l'empreinte digitale. Celle-ci est alors comparée à une base de données d'utilisateurs enregistrés afin d'identifier son propriétaire.

Les empreintes digitales sont uniques à chaque individu (*ex.* on estime que la probabilité⁹ que deux personnes aient la même empreinte digitale est de 1 sur 10^{24}). La formation des empreintes est influencée par les gènes et l'environnement dans lequel se développe le fœtus (*ex.* mouvement du liquide amniotique) : bien que la structure de base des empreintes ne soit pas complètement aléatoire, dû à l'influence de l'environnement, il est virtuellement impossible que deux empreintes soient identiques [MMJP09].

Dans [HB13], cette propriété est exploitée pour identifier les utilisateurs à des fins de sécurité (*ex.* autoriser un paiement uniquement si le propriétaire du compte qui le valide *via* un bouton). Mais les empreintes digitales sont aussi uniques à *chaque doigt* de chaque individu. Une technologie telle que *Fiberio* serait donc susceptible de permettre l'identification des doigts. Sugiura *et al.* ont d'ailleurs utilisé ce principe dans [SK98] : en associant un bouton physique à un scanner d'empreintes digitales, des commandes différentes étaient déclenchées en fonction du doigt pressant le bouton.

Il est à noter que la technologie capacitive (cf. Section 1.1) peut être utilisée pour numériser des empreintes digitales si la résolution du capteur est suffisamment élevée. C'est d'ailleurs cette technologie qui est utilisée sur les boutons *Touch ID*¹⁰ des téléphones *iPhone*¹¹. Il est donc plausible d'envisager que les prochaines générations d'écrans tactiles soient capables de numériser les empreintes digitales sur toute la surface. Certains constructeurs comme *Apple* ont d'ailleurs déposé des brevets sur ce sujet [YBL⁺15].

La reconnaissance d'empreintes digitales se base sur l'extraction de points caractéristiques appelés *minuties*. Les minuties sont les irrégularités jonchant les lignes formant les empreintes digitales (cf. Figure 1.19). La reconnaissance

9. http://fr.wikipedia.org/wiki/Empreinte_digitale

10. <https://support.apple.com/fr-fr/HT201371>

11. <http://essentialmac.co.uk/apple/fingerprint-scanner-5s-really-work/>

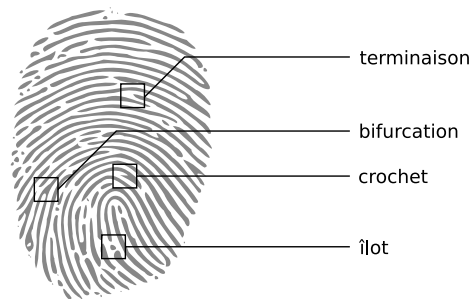


FIGURE 1.19: Exemples de minuties.

s'effectue en comparant la géométrie du réseau de minuties extraites à une base de données : un score de similarité est calculé entre l'empreinte courante et les différentes empreintes de la base, le score le plus élevé étant le meilleur candidat. Plus le réseau comporte de points caractéristiques, plus la reconnaissance sera fiable.

L'empreinte digitale recouvre la pulpe du doigt (*i.e.* partie charnue au bout de chaque doigt). La majorité des minuties se trouve sur la partie palmaire de la pulpe (*cf.* Figure 1.20). Cependant notre doigt est rarement bien à plat (*i.e.* cas idéal pour la reconnaissance d'empreintes digitales) : la plupart du temps nous interagissons avec les parties distale, médiale et latérale de la pulpe (*cf.* Figure 1.20) qui contiennent moins d'irrégularités. L'enjeu est donc de savoir si les différentes parties de la pulpe avec lesquelles nous interagissons comportent suffisamment d'information pour permettre une identification des doigts.

Afin de répondre à cette question nous proposons deux études complémentaires. La première étude identifie les différentes parties de la pulpe avec lesquelles nous interagissons sur un écran tactile. La seconde établit si la reconnaissance d'empreintes digitales est efficace pour identifier les doigts en fonction de la partie de la pulpe en contact.

1.3.1 Partie utile de la pulpe

La partie de la pulpe en contact avec une surface tactile est étroitement liée à l'orientation de la première phalange lors de l'interaction. En fonction du tangage (*i.e. pitch*, *cf.* Figure 1.21) la surface de la pulpe en contact passe de la partie palmaire à la partie distale. En fonction du roulis (*i.e. roll*, *cf.* Figure 1.21) la surface de la pulpe en contact passe de la partie palmaire aux parties latérales ou médiales. Afin d'établir les différentes orientations utilisées, nous avons mené une étude enregistrant le tangage et le roulis, par rapport à la surface de l'écran, des différents doigts lors d'une interaction sur tablette. Cette étude nous permettra par la suite d'identifier les orientations à tester dans la seconde étude.

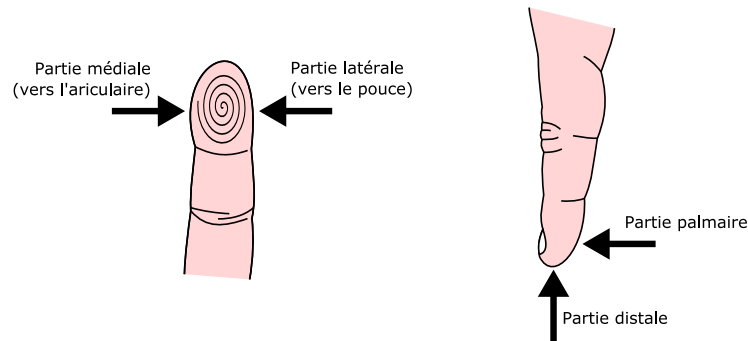


FIGURE 1.20: Différentes parties de la pulpe. La partie palmaire de la pulpe correspond au côté parallèle à l'ongle. La partie distale de la pulpe correspond au bout du doigt. La partie médiale de la pulpe correspond au côté tourné vers l'auriculaire. La partie latérale de la pulpe correspond au côté tourné vers le pouce.

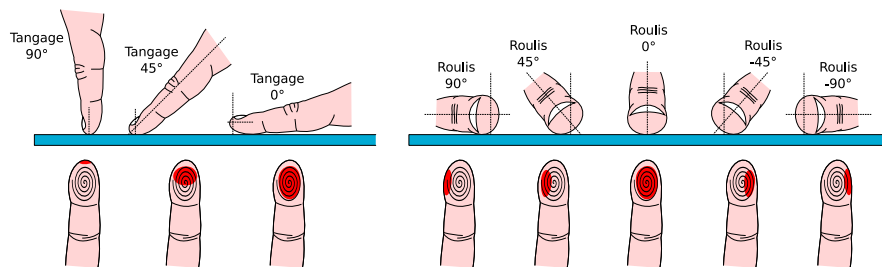


FIGURE 1.21: Influence du tangage et du roulis sur la partie de la pulpe en contact avec une surface : à gauche l'influence du tangage (*i.e. pitch*) ; à droite l'influence du roulis (*i.e. roll*). En rouge est schématisé la partie de la pulpe en contact avec la surface.

1.3.1.1 Participants

Nous avons recruté 15 volontaires (26,1 ans en moyenne, 4,3 ans de déviation standard, 4 femmes, 1 gaucher). Tous sauf un étaient utilisateurs quotidiens d'écrans tactiles (9 ayant un *smartphone** ou une tablette *Android*, 4 utilisant *iOS* et 1 utilisant *Windows mobile*). 8 participants n'utilisaient jamais le clavier *swipe*, le reste utilisaient ce clavier occasionnellement voire quotidiennement. 11 participants utilisaient une tablette très régulièrement.

1.3.1.2 Protocole et tâches

Après une courte introduction visant à collecter les différentes données démographiques, l'expérimentateur présentait le dispositif utilisé ainsi que le déroulement de l'expérience. Les participants ont eu pour instruction de suivre un scénario décrivant une interaction de quelques minutes. Le scénario était un enchaînement d'actions dans une application développée par nos soins, simulant des situations réalistes :

- Effectuer 4 *flicks** vers la gauche et 4 *flicks** vers la droite pour simuler des changements de bureau de la page d'accueil *Android*.
- Effectuer 4 glissés/déposés (un dans chaque direction cardinale) d'un objet de la taille d'une icône pour simuler la réorganisation d'applications sur la page d'accueil *Android*.
- Écrire une adresse email (*john.doe@gmail.com*) et le sujet d'un email pour simuler l'écriture d'un message dans *Gmail*.
- Dessiner des formes schématiques simples (une maison, un arbre et un humain) pour simuler l'écriture sur un clavier *swipe* sans perturber les utilisateurs novices.
- Se déplacer d'un point A à un point B puis retourner vers A sur *Google Map* en utilisant des gestes de changement d'échelle et de rotation.
- Faire défiler une page *Wikipédia* vers le bas puis vers le haut pour simuler la navigation dans un navigateur internet.

Pour passer d'une tâche à une autre, les participants appuyaient sur les onglets d'un menu coulissant. Cette navigation inter-tâche faisait partie du scénario. Le DOIGT utilisé durant l'interaction était imposé. Le scénario a été répété 5 fois afin de tester les 5 DOIGTS de la main dominante. Pour l'utilisation des gestes à deux doigts (*i.e.* navigation dans *Google Map*), les utilisateurs devaient utiliser le pouce et le doigt imposé. Si le doigt imposé était le pouce, l'utilisateur pouvait utiliser n'importe quel autre doigt. L'ordre de présentation des DOIGTS a été tiré aléatoirement au sort pour chaque participant.

L'expérimentateur effectuait le scénario devant le participant en lui expliquant chaque tâche de la liste précédente. Les participants étaient encouragés à s'entraîner avant de commencer. Durant l'expérience, l'expérimentateur veillait à ce que le scénario soit suivi et le bon doigt soit utilisé. Aucune erreur était

possible et toutes les tâches de la liste étaient effectuées. Lorsque les participants effectuaient plus d'actions que nécessaire, plus de données étaient tout simplement produites et analysées.

Après l'expérience, les participants étaient invités à noter dix propositions sur une échelle continue de 1 à 5 (1 - d'accord, 5 - pas d'accord) : P1) *J'ai eu le même niveau de **contrôle** entre tous les doigts* ; P2) *L'interaction avec certains doigts était moins **naturelle*** ; P3) *Les marqueurs n'ont pas réduit mon **amplitude** de mouvements possibles* ; P4) *Porter les marqueurs était **inconfortable*** ; P5) *Porter les marqueurs m'a fait agir de manière **non-naturelle*** ; P6) *Porter les marqueurs était **distrayant*** ; P7) *Les marqueurs ont rendu la **vision** de ce que je faisais difficile* ; P8) *Je me suis **habitué** aux marqueurs* ; P9) *Porter les marqueurs a **ralenti** mes actions* ; P10) *J'aurais agi **différemment** sans les marqueurs*.

1.3.1.3 Dispositif

L'expérience utilisait une tablette *Nexus 10 Android*. La tablette était posée à plat sur une table et entourée d'un système de *Motion Capture OptiTrack* comportant 4 caméras *Prime 13*. Le dispositif a été calibré pour suivre la position et l'orientation dans l'espace de deux squelettes rigides comportant chacun trois marqueurs réfléchissants : un squelette permettant le suivi de la tablette et un autre permettant le suivi de la première phalange d'un doigt. Les informations relatives aux marqueurs étaient fournies par le logiciel *Motive* et étaient envoyées *via* le réseau à une application dédiée enregistrant le tout dans un fichier de données. Côté tablette, tous les événements tactiles étaient enregistrés dans un fichier local.

Une fois l'expérience terminée, les deux fichiers de données (*i.e.* données d'orientation et données d'événements tactiles) étaient fusionnés en un seul et même fichier *via* une application dédiée permettant : visualisation des données et ré-alignement des horodatages le cas échéant.

1.3.1.4 Résultats et discussion

Nous avons collecté un total de 253 289 événements tactiles ainsi que l'orientation des doigts correspondants. Ces données sont réparties au travers de 2 166 gestes (*i.e.* *tap**, glissé, *flick**, *etc.*). L'analyse de la distribution nous permettra d'établir une liste d'orientations à tester dans la seconde partie de l'étude. Cependant, nous devons préalablement déterminer si les données que nous avons collecté sont écologiquement valides. En effet, les doigts sont équipés de marqueurs qui pourraient potentiellement altérer la façon dont les participants interagissent avec une surface tactile. C'est précisément l'objectif derrière la notation des différentes propositions. Dans un premier temps, nous nous proposons donc d'analyser les questionnaires remplis par les participants à la fin de chaque session. Nous passerons ensuite à l'analyse des orientations.

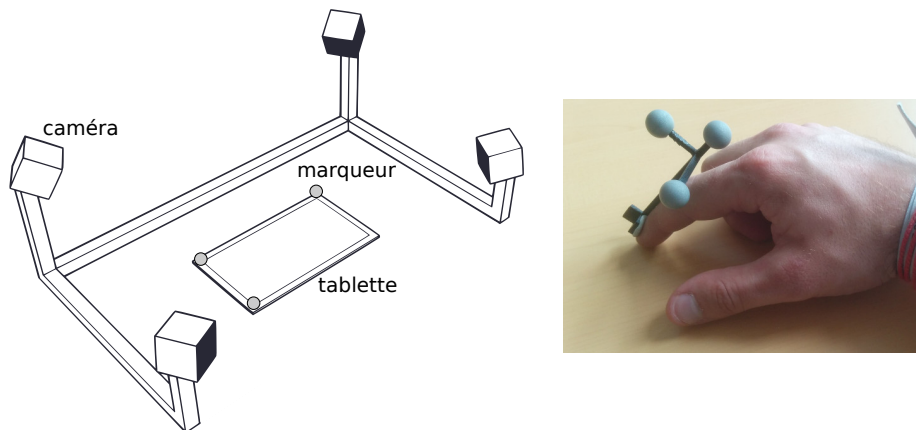


FIGURE 1.22: Dispositif utilisé pour l'expérience visant à déterminer la partie utile de la pulpe : (gauche) la tablette posée à plat, les trois marqueurs qui permettent de suivre l'orientation et la position de la tablette et les quatre caméras du système de *Motion Capture* ; (droite) le squelette rigide et les trois marqueurs qui permettent de suivre l'orientation et la position de la première phalange du doigt.

Analyse des propositions La table 1.1 regroupe les notes moyennes ainsi que les déviations standards pour chacune des propositions. Comme attendu, les participants n'ont pas eu le même niveau de contrôle entre les différents doigts (P1 : moyenne 4,1), certains leur paraissant moins naturels pour l'interaction (P2 : moyenne 1,7). 7 participants sur 15 ont trouvé l'annulaire et l'auriculaire plus difficiles à utiliser. Les participants n'ont en revanche pas trouvé que les marqueurs réduisaient l'amplitude de leurs mouvements (P3 : moyenne 1,8). Pour les 3 participants qui ont perçu une réduction, la gêne apparaissait lors de l'extension des doigts. Dans leur cas, les doigts se retrouvaient moins à plat et donc éloignés du cas idéal pour la reconnaissance d'empreintes digitales. Il n'y a pas eu de ressenti particulier sur l'inconfort des marqueurs (P4 : moyenne 3,2) et lorsque celui-ci était mentionné, il était lié au léger surpoids engendré. De même, les mouvements effectués par les participants n'ont pas été jugés anormaux (P5 : moyenne 3,4). Les aspects du mouvement sur lesquels les participants restaient prudents étaient la vitesse et la précision. De même, pour les doigts n'étant jamais utilisés *naturellement* les participants ont préféré rester neutre. Aucun des participants n'a trouvé les marqueurs distrayants (P6 : moyenne 4,4) et un seul a ressenti une gêne visuelle (P7 : moyenne 4,6). À titre de comparaison, un participant, pour qui les marqueurs n'ont pas provoqué de gêne, a trouvé l'effet visuelle similaire au fait d'avoir "un stylo entre les doigts". Il y a eu un léger temps d'adaptation aux marqueurs (P8 : moyenne 2,3). Pour certains, passé quelques interactions avec le premier doigt imposé de l'expérience, les marqueurs étaient rapidement oubliés. Pour d'autres, les marqueurs n'étant pas une distraction ou une gêne au départ, il n'y a pas eu besoin d'adaptation. Un

Proposition	P1	P2	P3	P4	P5
	contrôle	naturelle	amplitude	inconfortable	non-naturelle
Moyenne	4,1	1,7	1,8	3,2	3,4
Dév. std	1,3	1,4	1,1	1,2	1,2
Proposition	P6	P7	P8	P9	P10
	distrayant	vision	habitué	ralenti	différemment
Moyenne	4,4	4,6	2,3	3,9	3,8
Dév. std	0,7	0,8	1,3	1,3	1,0

TABLE 1.1: Résumé des notes données par les 15 participants aux dix propositions. Pour chaque proposition est donné la moyenne et la déviation standard. Le mot clef de chaque proposition est rappelé en dessous de l'identifiant.

participant nous a dit s'être adapté mais que les marqueurs restaient tout de même désagréables. En général, les marqueurs n'ont pas spécialement ralenti l'interaction (P9 : moyenne 3,9). Selon un participant, c'est plus le "manque de dextérité que les marqueurs qui a ralenti l'interaction" et ce pour les doigts dont nous n'avons pas l'habitude de se servir. Pour finir, les participants n'auraient pas particulièrement agi différemment (P10 : moyenne 3,8). Encore une fois, les différences mentionnées sont surtout liées à la vitesse et à la précision plutôt qu'à l'orientation.

L'analyse des notations nous permet donc d'établir une certaine confiance dans notre jeu de données. Les orientations enregistrées lors de l'expérience semblent être proches de celles qui auraient été obtenues dans un contexte réel. Les différences qui pourraient exister viendraient des gênes ressenties lors de l'extension des doigts affectant ainsi la distribution des données proches du cas idéal (*i.e.* doigt bien a plat). Ces altérations éventuelles rendraient donc la tâche de l'identification des doigts plus difficile faisant des résultats de la seconde étude une évaluation pessimiste de la réalité.

Analyse des distributions L'analyse des 253 289 événements nous a révélé une perte de 7 460 éléments dû à l'occultation des marqueurs par le reste de la main. Cette perte correspond à 2,9% des données récoltées (11,3% de pertes pour le pouce, 1,8% pour l'auriculaire, 0,7% pour l'index et l'annulaire et 0,3% pour le majeur). La perte des données intervient le plus souvent pour le pouce lors de la rotation. Le taux de perte des données de rotation du pouce représente 20,7% (75,5% des pertes totales, 71,6% des pertes du pouce). Lorsque que ce mouvement commence ou finit avec le pouce en dessous de la main, les marqueurs, bien que décalés de la première phalange, peuvent se retrouver cachés par le poignet. Dans une moindre mesure, cette situation se produit aussi avec l'auriculaire. Le taux de perte des données de rotation de l'auriculaire représente 2,9% (12,9% des pertes totales, 65,5% des pertes de l'auriculaire). Ces doigts étant aux extrêmes de la main, ils sont plus enclins à ce problème. La figure 1.23 montre la distribution

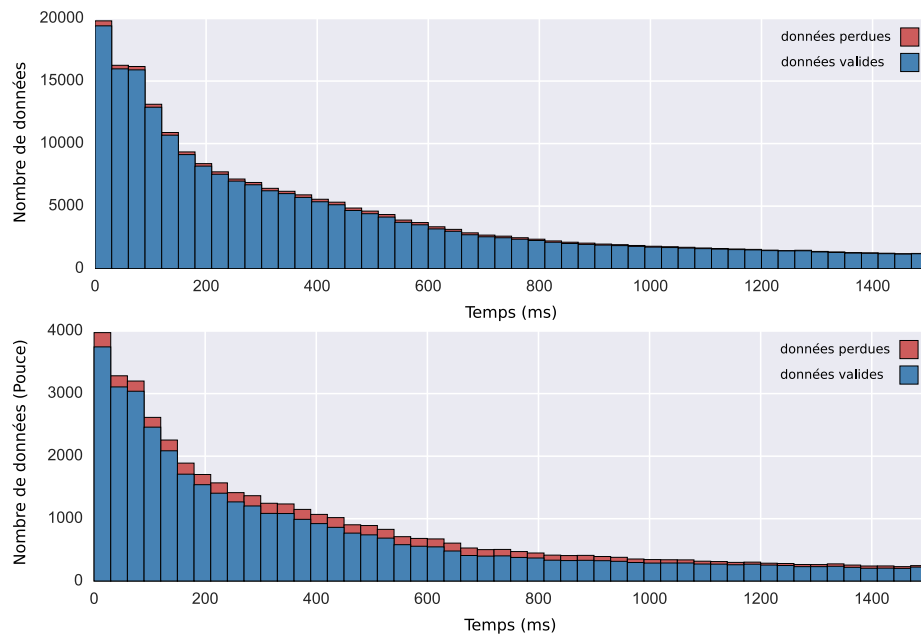


FIGURE 1.23: Distributions des données d'orientation au cours des 1 500 premières millisecondes des contacts. La distribution du haut représente tous les doigts cumulés, celle du bas uniquement les données liées au pouce. Les données perdues correspondent à l'occultation des marqueurs qui engendre une perte de suivi par le système de *Motion Capture*

des données (valides et perdues) au cours des 1 500 premières millisecondes des contacts pour tous les doigts cumulés et pour le pouce seul.

La Figure 1.24 représente la distribution des orientations¹² (tangage et roulis) pour chacun des doigts. Ces distributions représentent les répartitions de 95% des données. Les différences d'altitude entre les lignes de niveau sont de 5% et la ligne des 75% est indiquée en pointillé. Chaque ligne représente un doigt différent de la main. Les colonnes correspondent aux distributions des 50 premières millisecondes, 200 premières millisecondes et à l'ensemble des données. On peut remarquer des distributions similaires entre les colonnes.

Parmi les doigts, trois comportements se distinguent : l'index qui interagit avec toutes les parties de la pulpe ; le majeur, l'annulaire et l'auriculaire qui interagissent majoritairement avec la partie médiale de la pulpe (*cf.* Figure 1.20) ; le pouce qui interagit avec la partie latérale de la pulpe (*cf.* Figure 1.20). Cette différence de comportement s'explique par la morphologie de la main. L'index est suffisamment indépendant pour pouvoir interagir sans contrainte (*ex.* en fermant le poing). En revanche, avec le majeur, l'annulaire et l'auriculaire, il est

12. Pour l'utilisateur gaucher, les angles correspondant au roulis ont été inversés afin de normaliser les données entre droitier et gaucher.

difficile d'interagir en utilisant la partie de la pulpe qui fait face au pouce sans que le reste de la main vienne buter sur la surface d'interaction. Pour le pouce, qui est opposable au reste des doigts, le problème est analogue mais de manière symétrique. De plus, le pouce peut difficilement interagir à plat ce qui façonne la ligne nette formée par la distribution (*cf.* première ligne de la Figure 1.24). Enfin on peut remarquer que seul le pouce et l'auriculaire interagissent avec un tantage en deçà de 30° .

Choix des orientations à tester Pour déterminer les orientations à tester dans la seconde étude, nous nous sommes imposés trois règles : tester chaque doigt à plat (*i.e.* tantage 0° et roulis 0°) afin d'établir une condition témoin ; tester les contours des distributions afin d'expérimenter les conditions *a priori* les moins favorables ; tester les points de forte concentration des données afin d'expérimenter les conditions les plus fréquentes. Nous souhaitions de même avoir un nombre raisonnable d'orientations à tester pour pouvoir effectuer l'expérience en un temps acceptable. Les points rouges de la Figure 1.24 correspondent aux orientations choisies à l'aide de ces règles.

1.3.2 Utilisation des empreintes digitales

Afin d'évaluer si la reconnaissance d'empreintes digitales peut servir à l'identification des doigts, nous avons constitué une base de données de 2 400 empreintes digitales. Ces empreintes proviennent de 12 participants¹³ et ont été acquises grâce à un dispositif qui utilise le principe Frustrated Total Internal Reflection (*FTIR*)* [HS84, BC96]. Ce principe d'acquisition est le plus ancien et le plus communément utilisé [MMJP09].

1.3.2.1 Dispositif

Une empreinte digitale est composée de creux (*i.e.* les sillons visibles sur la pulpe) et de bosses (*i.e.* les lignes qui composent l'empreinte). Le principe d'acquisition d'une empreinte digitale est d'obtenir une image contrastée qui fasse ressortir les bosses du reste de l'anatomie du doigt. Notre dispositif (*cf.* Figure 1.25-gauche) est composé d'un prisme, d'un système de lumière diffuse et d'un appareil photo (*Reflex Canon EOS 700D* équipé d'un objectif *Canon EF 50 mm f/2.5 Macro*). Le prisme est placé tel que l'une des pointes soit dirigée vers le bas. Le système de lumière et l'appareil photo sont placés de façon à ce que chacun soit perpendiculaire à l'une des faces inférieures du prisme (*cf.* Figure 1.25-droite). Dans cette configuration, la lumière émise vers le prisme est entièrement réfléchi sur la face supérieure vers le dispositif d'acquisition. Lorsque le doigt est posé sur le prisme, seules les bosses sont en contact direct avec la surface. La lumière arrivant sur ces points de contact est alors diffusée dans toutes les directions (*cf.* Figure 1.25-droite). La part de cette lumière réfléchi

13. Étant donné le caractère sensible de ces données, celles-ci ont été anonymisées et nous n'avons récolté aucune information démographique.

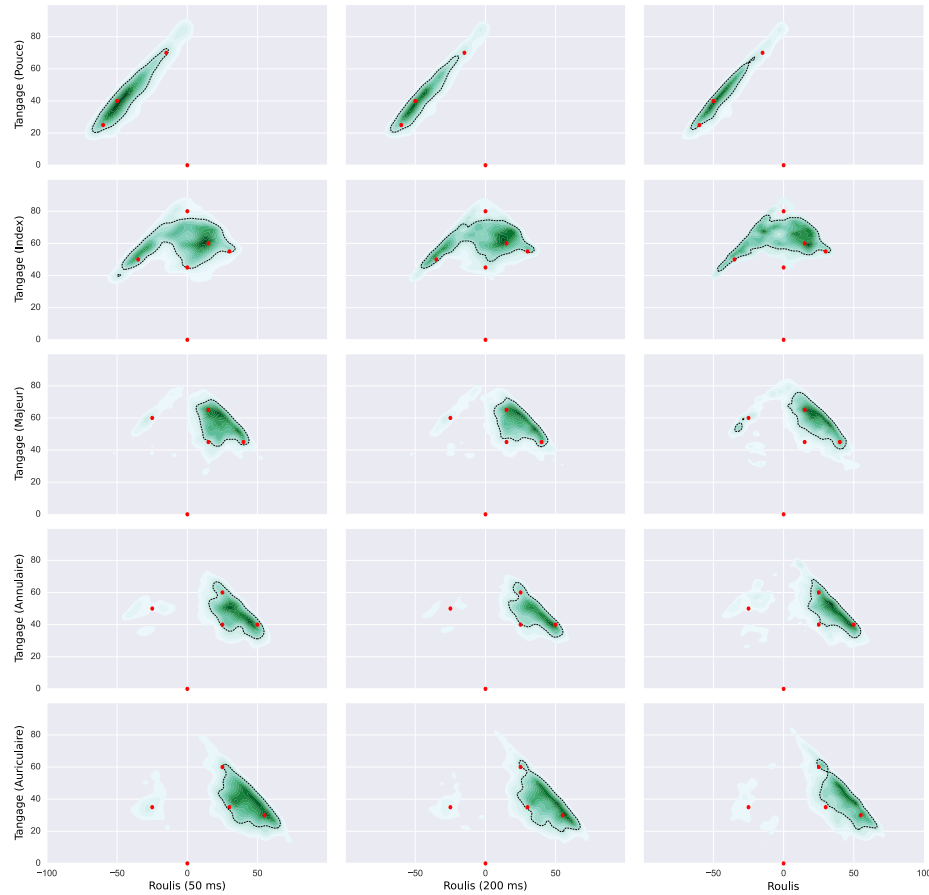


FIGURE 1.24: Distributions des orientations en degrés de la première phalange de chaque doigt pour les premières 50 ms de l'interaction, 200 ms et toute la durée du contact. Le premier niveau de couleur représente 95% des distributions. La ligne de niveau en pointillée représente 75% des distributions. Chaque couleur représente une ligne de niveau à 5% d'écart de la précédente. Les valeurs positives du roulis correspondent à la partie médiale de la pulpe (*i.e.* côté tourné vers l'auriculaire). À l'inverse les valeurs négatives correspondent à la partie latérale de la pulpe (*i.e.* côté tourné vers le pouce). En rouge sont représentées les orientations testées par la suite.

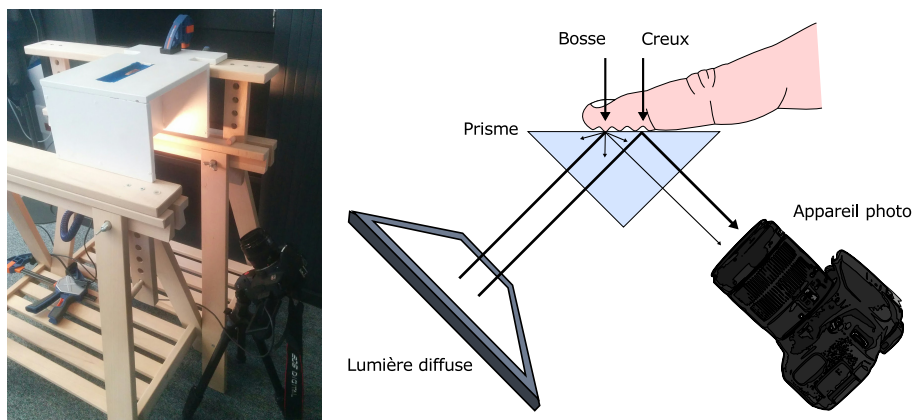


FIGURE 1.25: Dispositif d'acquisition *FTIR** d'empreintes digitales : (gauche) image du dispositif ; (droite) principe de fonctionnement. Dans le schéma, les flèches représentent différents rayons de lumière. L'épaisseur de ces flèches correspond à l'intensité lumineuse : plus une flèche est fine moins le rayon est lumineux.

vers l'appareil photo est donc moins lumineuse ce qui engendre un contraste sur l'image acquise (cf. Figure 1.26).

1.3.2.2 Protocole

L'expérimentateur commençait chaque session par expliquer au participant l'objectif de l'étude ainsi que la manière dont étaient stockées les acquisitions des empreintes digitales¹⁴. La session d'acquisition des empreintes se poursuivait uniquement si le participant acceptait explicitement d'y participer. L'expérimentateur présentait ensuite au participant le dispositif et son fonctionnement. Chaque session était divisée en 10 parties. Une partie correspondait à l'acquisition d'images de l'un des 10 DOIGTS sous différentes orientations (cf. Section 1.3.1.4) : 4 orientations pour les pouces, 6 pour les index, 5 pour les majeurs, 5 pour les annulaires et 5 pour les auriculaires. Au début d'une partie, l'expérimentateur équipait le DOIGT courant d'un accéléromètre qui permettait de mesurer l'orientation en continue (cf. Figure 1.27-gauche). Pour ne pas couvrir l'empreinte digitale, celui-ci était placé sur la dernière phalange. Le participant avait donc pour instruction de garder le DOIGT tendu lors des acquisitions afin que la première et la dernière phalanges soient alignées. Le participant plaçait ensuite son DOIGT à plat sur le prisme et l'expérimentateur réinitialisait le système pour que la position courante corresponde à un angle de tangage et de roulis nuls. Une fenêtre de visualisation permettait de contrôler

14. Les images des empreintes digitales étaient stockées sur un disque dur externe crypté dans un bureau à accès restreint. Aucune autre copie des acquisitions n'a été réalisée et celles-ci n'ont jamais transité par un quelconque réseau. Une fois l'analyse effectuée, les acquisitions ont été effacées.

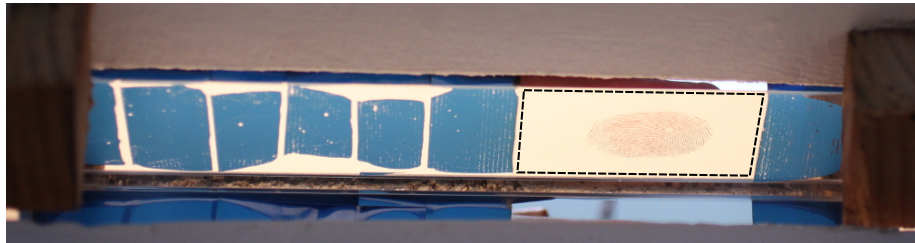


FIGURE 1.26: Exemple d'image acquise par notre dispositif. Afin de rendre l'empreinte (de l'auteur) illisible, la qualité de l'image a volontairement été dégradée. Cette image illustre néanmoins le contraste engendré par le dispositif entre les bosses de l'empreinte et le reste de l'anatomie du doigt. Le contour pointillé correspond à la zone de l'image utilisée pour la comparaison d'empreintes digitales.

l'orientation courante ainsi que de voir celle à atteindre (*cf.* Figure 1.27-droite). Une fois l'orientation cible atteinte, le participant posait son DOIGT sur le prisme, l'expérimentateur déclenchait l'appareil photo *via* l'application, puis le DOIGT était enlevé du prisme. En tout 4 acquisitions étaient effectuées par orientation ce qui donne un total de 200 acquisitions par participant.

1.3.2.3 Comparaison d'empreintes

Pour chaque acquisition, nous effectuons un recadrage de l'image sur la zone utile qui contient l'empreinte (*cf.* Figure 1.26). Nous utilisons ensuite l'algorithme de détection de caractéristiques Speeded Up Robust Features (*SURF*)^{*} [BTVG06] pour créer un descripteur de l'image. Les caractéristiques extraites sont indépendantes de l'échelle et de la rotation de l'image ce qui nous affranchit du contrôle de l'angle de lacet du doigt (*i.e. yaw*, *cf.* Figure 1.8). Pour comparer deux empreintes, nous comparons leur descripteurs respectifs à l'aide de l'algorithme de recherche des plus proches voisins Fast Approximate Nearest Neighbors (*FANN*)^{*} [ML09]. Nous prenons la distance minimum entre deux caractéristiques du voisinage trouvé comme score de similarité entre les empreintes. Plus ce score approche de 0, plus les empreintes sont similaires. L'avantage d'utiliser un score indépendant du cardinal du voisinage trouvé, nous assure qu'une empreinte partielle ait autant de chance qu'une empreinte complète : seule la qualité compte et pas la surface en commun. Nous utilisons les implémentations des algorithmes fournis par la bibliothèque *OpenCV*¹⁵.

Notre méthode de comparaison d'empreintes digitales est différente de celle habituellement utilisée par l'état de l'art [MMJP09, HB13]. Dans la méthode classique, une série de filtres qui améliorent la qualité de l'image sont d'abord appliqués avant d'extraire les positions et orientations des minuties. La principale raison est liée au manque de bibliothèques *open source* disponibles et fiables. Les

15. <http://opencv.org/>

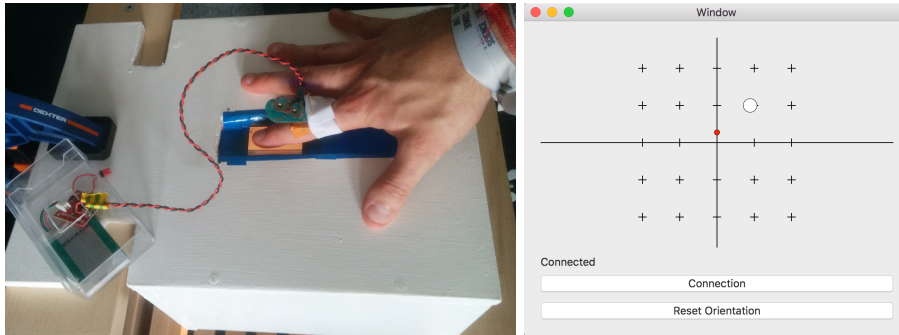


FIGURE 1.27: Contrôle de l'orientation d'un doigt lors de l'acquisition d'une empreinte digitale : (gauche) doigt d'un utilisateur équipé de l'accéléromètre ; (droite) fenêtre de visualisation de l'orientation courante. Dans la fenêtre de visualisation, le point rouge correspond à l'orientation courante du doigt et la cible blanche correspond à la position à atteindre.

tests préliminaires effectués ont révélé des anomalies lors de l'extraction des minuties et des calculs des scores de similarité. Nous avons donc décidé d'utiliser une méthode alternative mais fiable, nous permettant d'établir rapidement une borne inférieure des résultats. Nous sommes actuellement en cours d'implémentation de l'algorithme classique et prévoyons de recommencer l'analyse par la suite.

Il est à noter que des solutions commerciales existent, cependant : elles sont opaques quant aux algorithmes utilisés ; elles imposent un nombre de minuties minimum pour effectuer une comparaison, ce qui interdit leur utilisation sur une large proportion de notre base de données (*ex.* orientations aux angles de tangage ou de roulis élevés). Nous avons donc écarté leur utilisation.

1.3.2.4 Résultats et discussion

Nous séparons l'analyse des données en deux parties. Tout d'abord, nous évaluons notre dispositif en utilisant la méthode utilisée par Holtz *et al.*. La comparaison de nos résultats à ceux de [HB13], nous permet ainsi d'établir la fiabilité de notre processus d'acquisition et de comparaison. Nous analysons ensuite nos données afin d'apporter des éléments de réponse à la problématique qui motive cette étude.

Dans la suite, nous présentons des taux de reconnaissance d'empreintes digitales. Ces taux sont calculés de la façon suivante :

1. Nous constituons un ensemble E d'empreintes selon différents critères (*ex.* toutes les empreintes d'un même utilisateur, toutes les empreintes correspondant à un index droit, *etc.*).
2. Pour toute empreinte e de E , nous calculons le score de similarité avec chacune des autres empreintes de E .
3. Pour toute empreinte e de E , nous calculons trois taux de reconnaissance

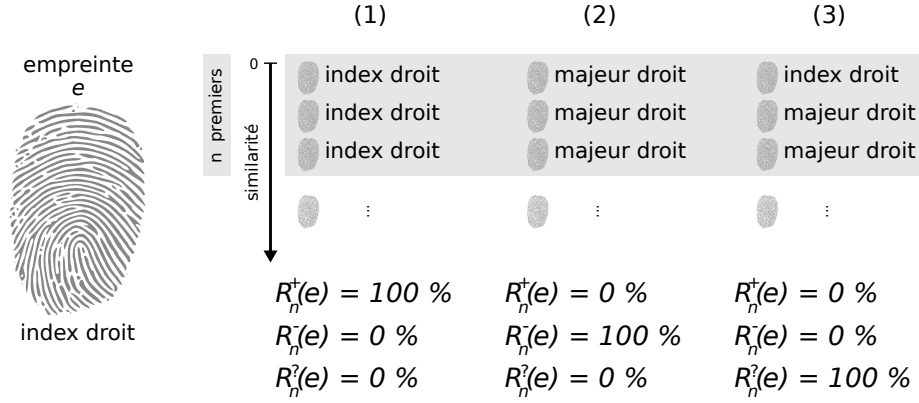


FIGURE 1.28: Exemple de calcul des taux de succès, d'échec et d'incertitude pour une empreinte e avec $n = 3$: (1) les n empreintes les plus similaires correspondent à l'empreinte e donc $R_n^+(e) = 100\%$ et $R_n^-(e) = R_n^?(e) = 0\%$; (2) les n empreintes les plus similaires correspondent entre elles mais pas à l'empreinte e donc $R_n^-(e) = 100\%$ et $R_n^+(e) = R_n^?(e) = 0\%$; (3) les n empreintes les plus similaires ne correspondent pas entre elles donc $R_n^+(e) = 100\%$ et $R_n^-(e) = R_n^?(e) = 0\%$.

$R_n^+(e)$ (*i.e.* taux de succès), $R_n^-(e)$ (*i.e.* taux d'échec) et $R_n^?(e)$ (*i.e.* taux d'incertitude) qui valent soit 100% soit 0% (*cf.* Figure 1.28).

- $R_n^+(e)$ vaut 100% si les n empreintes les plus similaires à e correspondent au même doigt : taux de succès.
- $R_n^-(e)$ vaut 100% si les n empreintes les plus similaires correspondent au même doigt entre elles mais pas à celui de e : taux d'échec.
- $R_n^?(e)$ vaut 100% si ni $R_n^+(e)$ ni $R_n^-(e)$ ne valent 100% : taux d'incertitude.

n correspond à la tolérance du système : plus n est grand plus il est difficile de reconnaître une empreinte avec succès, de même qu'il est plus difficile d'obtenir un faux positif. Dans notre étude, n peut valoir 1, 2 ou 3 étant donné que notre protocole prévoyait 4 répétitions par orientation et par doigt. À noter que lorsque n vaut 1, il n'y a pas de taux d'incertitude.

4. Les taux de reconnaissance globaux R_n^+ , R_n^- et $R_n^?$ sont les moyennes des $R_n(e)$ respectifs.

Évaluation du dispositif Pour évaluer leur prototype, Holtz *et al.* ont collecté les empreintes des index, majeur et annulaire droits de 10 participants. Cinq acquisitions par doigt ont été réalisées. La portion de l'empreinte capturée correspond à la partie palmaire de la pulpe (*i.e.* doigt à plat, *cf.* Figure 1.20). Dans [HB13], l'ensemble E est composé des 150 empreintes digitales. Le taux de

$E = IMA$	R_n^+	$R_n^?$	R_n^-
$n = 1$	94%	0%	6%
$n = 2$	90%	9%	1%
$n = 3$	74%	26%	0%

TABLE 1.2: Taux de reconnaissance des empreintes digitales des index, majeurs et annulaires droits (*i.e.* I , M et A) à plat.

succès R_4^+ est de 99% (148 empreintes sur 150), le taux d'incertitude $R_4^?$ est de 1% (2 empreintes sur 150) et le taux d'échec R_4^- est de 0%.

Nous reprenons la méthode précédente et l'appliquons à notre base de données. Nous constituons l'ensemble E des empreintes des index, majeurs et annulaires droits à plat de tous les participants. Les taux de reconnaissance obtenus sont résumés dans la table 1.2. Nous avons aussi calculé les taux de reconnaissance en faisant varier E (*cf.* Tables 1.3).

Sans surprise, les taux de succès et d'échec diminuent quand n augmente. Les taux trouvés confirment qu'une amélioration du système est possible et que les résultats des analyses suivantes constituent une estimation basse du pouvoir discriminant des empreintes digitales.

Taux de succès Pour savoir s'il est possible de discriminer les doigts d'un utilisateur grâce aux empreintes digitales, nous avons calculé différents taux pour chaque participant. Pour un participant donné, nous définissons l'ensemble E comme l'ensemble de ses empreintes (200 empreintes au total). Les taux calculés étaient : (1) les taux de succès pour reconnaître un doigt particulier à une orientation particulière (*ex.* reconnaître le majeur gauche lorsque celui-ci est à plat), au total 50 taux calculés par participant ; (2) les taux de succès pour reconnaître un doigt particulier (*ex.* reconnaître l'annulaire droit), au total 10 taux calculés par participant ; (3) le taux de succès R_n^+ pour reconnaître un doigt, au total 1 taux calculé par participant. Nous avons ensuite calculé les taux moyens des participants. Le processus a été effectué pour $n = 1, 2$ et 3 .

Le taux de succès R_1^+ moyen est de 78%, R_2^+ vaut 61% et R_3^+ vaut 43%. Dans les figures 1.29 et 1.30 nous résumons les autres taux de succès moyens pour $n = 1$ et $n = 2$. La colonne de gauche correspond aux doigts de la main gauche et celle de droite à ceux de la main droite. Les lignes correspondent aux différents doigts des pouces (en haut) aux auriculaires (en bas). Les points rouges correspondent aux taux (1) et en haut de chaque graphique est affiché le taux (2) du doigt associé. La figure correspondant à $n = 3$ est donnée en annexe (*cf.* Annexe C.1).

Par souci d'exhaustivité, nous avons aussi calculé les taux de succès pour reconnaître à la fois doigts et orientations. Nous résumons ces résultats dans des figures similaires (*cf.* Annexe C.2).

$E = \text{tous}$	R_n^+	$R_n^?$	R_n^-	$E = I_g M_g A_g$	R_n^+	$R_n^?$	R_n^-
$n = 1$	90%	0%	10%	$n = 1$	92%	0%	8%
$n = 2$	77%	21%	2%	$n = 2$	83%	12%	4%
$n = 3$	54%	45%	1%	$n = 3$	69%	30%	1%
$E = TIM$	R_n^+	$R_n^?$	R_n^-	$E = T_g I_g M_g$	R_n^+	$R_n^?$	R_n^-
$n = 1$	99%	0%	1%	$n = 1$	91%	0%	9%
$n = 2$	97%	2%	1%	$n = 2$	85%	11%	3%
$n = 3$	80%	20%	0%	$n = 3$	73%	25%	2%
$E = I_g \text{ ou } I$	R_n^+	$R_n^?$	R_n^-	$E = I_g I$	R_n^+	$R_n^?$	R_n^-
$n = 1$	100%	0%	0%	$n = 1$	97%	0%	3%
$n = 2$	100%	0%	0%	$n = 2$	93%	6%	1%
$n = 3$	100%	0%	0%	$n = 3$	82%	18%	0%
$E = T_g I_g M_g A_g L_g$	R_n^+	$R_n^?$	R_n^-	$E = TIMAL$	R_n^+	$R_n^?$	R_n^-
$n = 1$	91%	0%	9%	$n = 1$	93%	0%	7%
$n = 2$	82%	15%	3%	$n = 2$	85%	14%	1%
$n = 3$	62%	37%	1%	$n = 3$	67%	32%	1%

TABLE 1.3: Taux de reconnaissance des empreintes digitales à plat pour d'autres combinaisons de doigts : tous les doigts ; index, majeurs et annulaires gauches (*i.e.* I_g , M_g et A_g) ; pouces, index et majeurs droits (*i.e.* P , I et M) ; pouces, index et majeurs gauches (*i.e.* P_g , I_g et M_g) ; index gauches ou droits (*i.e.* I_g ou I) ; index gauches et droits (*i.e.* $I_g I$) ; les doigts de la main gauche (*i.e.* $T_g I_g M_g A_g L_g$) ; les doigts de la main droite (*i.e.* $TIMAL$).

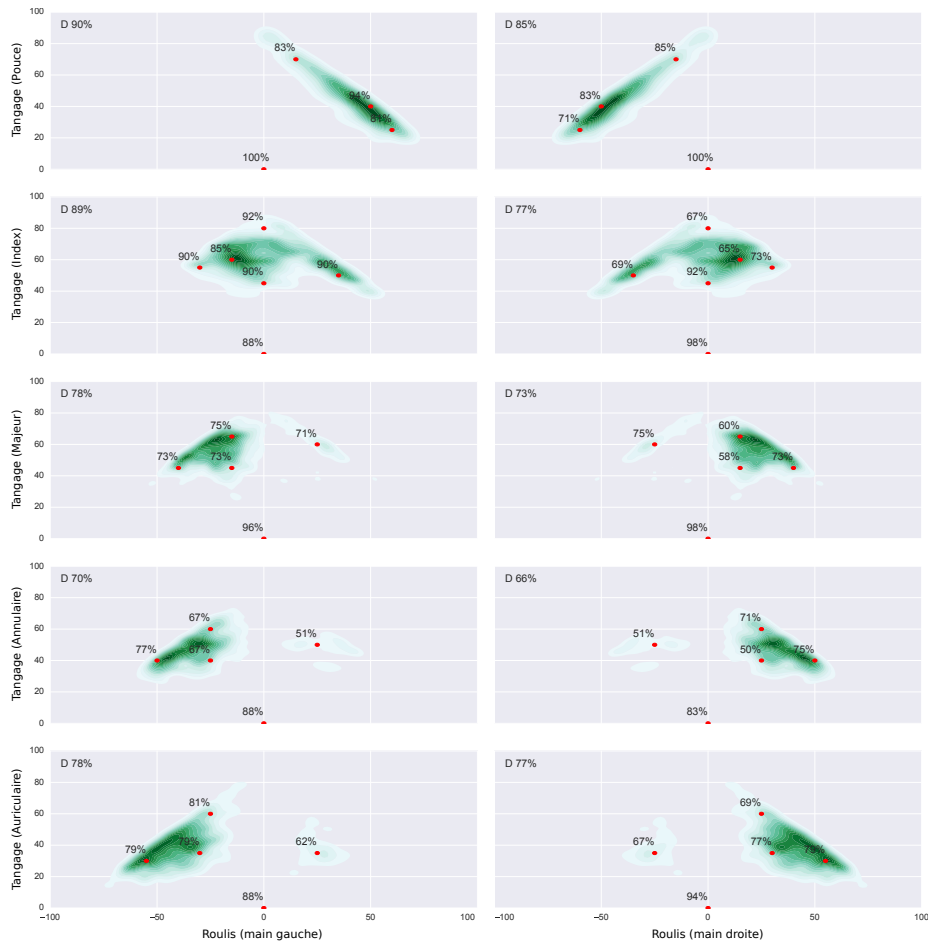


FIGURE 1.29: Taux de succès moyen de reconnaissance du doigt pour $n = 1$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt pour l'orientation courante. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

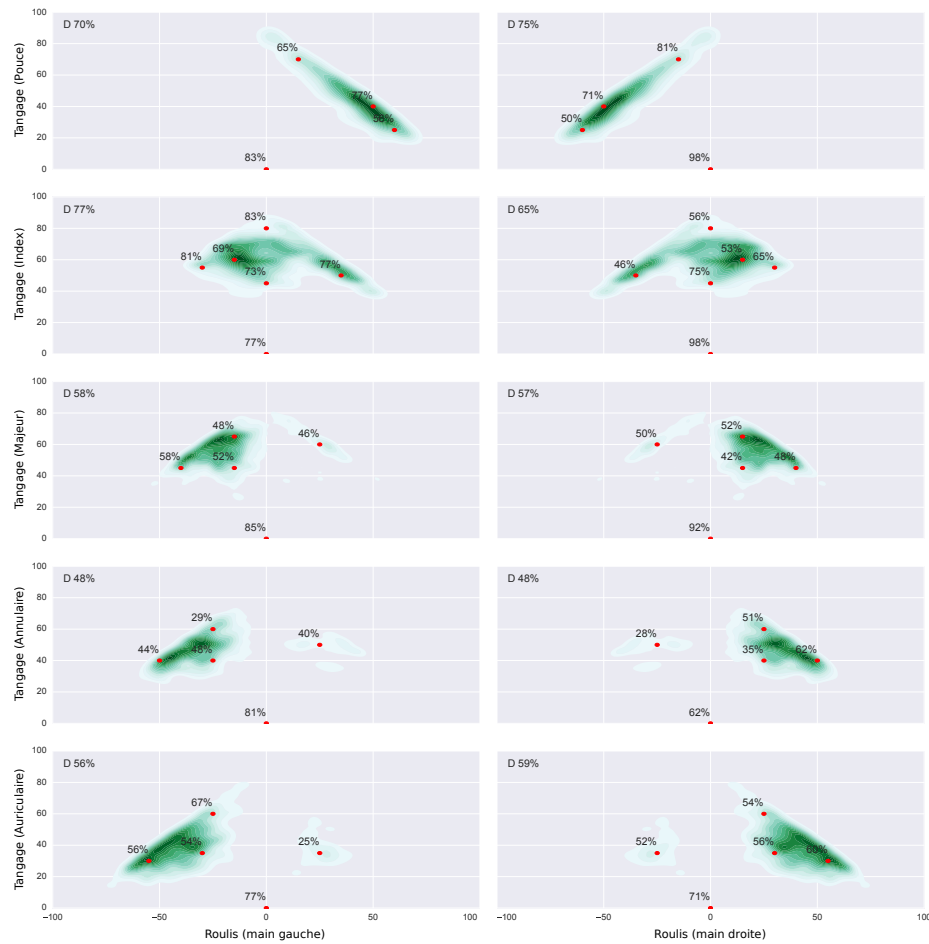


FIGURE 1.30: Taux de succès moyen de reconnaissance du doigt pour $n = 2$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt pour l'orientation courante. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

Identifier les doigts d’un utilisateur L’évaluation de notre dispositif *via* la méthode décrite dans *Fiberio* suggère qu’une amélioration des résultats est possible (en adoptant par exemple l’algorithme classique de reconnaissance d’empreintes digitales). Cependant, en se basant sur les résultats obtenus avec $n = 1$ (*i.e.* les plus proches d’un dispositif idéal) il est possible d’extraire des observations intéressantes.

Bien que la surface de contact diminue fortement lorsqu’un doigt n’est pas à plat, les taux de succès restent bien supérieurs à une identification aléatoire (*i.e.* 10%); 4 des 50 orientations testées ont un taux de succès entre 50% et 60% (4 sur 40, en excluant les doigts à plat); 9 des 50 orientations testées ont un taux de succès entre 60% et 70% (8 sur 40); 15 des 50 orientations testées ont un taux de succès entre 70% et 80% (12 sur 40); 10 des 50 orientations testées ont un taux de succès entre 80% et 90% (7 sur 40); 10 des 50 orientations testées ont un taux de succès entre 90% et 100% (7 sur 40). De plus, les orientations les plus utilisées (*i.e.* zones foncées de la figure 1.29) ont des taux proches de 70% (voire plus) à part pour le majeur et l’annulaire droits.

Il est aussi important de noter, que 3 participants ont eu des taux de succès R_1^+ moyens relativement faibles (62%, 66% et 68%) comparé aux autres participants (76%, 76%, 76%, 78%, 78%, 80%, 82%, 84%, 92% et 96%) alors que leur taux de succès R_1^+ moyens des doigts à plat sont élevés (88%, 88% et 98%). L’analyse manuelle des images a pourtant révélé des images de très bonne qualité. Cette observation laisse donc à penser qu’il pourrait y avoir une inégalité entre les personnes quant à la quantité d’information contenue sur les parties périphériques d’une empreinte digitale.

1.3.3 Conclusion et travaux futurs

Les empreintes digitales semblent être une information pertinente pour identifier les doigts. Bien que les orientations prises par les différents doigts soient éloignées du cas classique et idéal de la reconnaissance d’empreintes, nos résultats tendent à montrer que les différentes parties de la pulpe contiennent une quantité d’information non négligeable. Il est cependant important de noter qu’avec l’état actuel de notre dispositif, cette information n’est pas suffisante pour avoir une identification des doigts robuste. Elle peut néanmoins être combinée à d’autres pour aider à la désambiguïsation. Par exemple, l’utilisation du travail de Xiao *et al.* [XSH15], qui permet entre autre d’estimer le tangage du doigt en contact, pourrait permettre de réduire la base de données de comparaison et d’augmenter les chances de succès d’identification *via* les empreintes digitales. De même, certaines heuristiques pourraient aussi être utilisées : dans notre étude nous nous basons sur une image unique du doigt à identifier, or lors d’une interaction, plusieurs acquisitions peuvent être effectuées et ainsi augmenter la fiabilité de la reconnaissance.

Des travaux complémentaires sont nécessaires pour étudier les pistes évoquées ci-dessus. Pour se faire, il est impératif d’améliorer la méthode de comparaison des empreintes digitales ainsi que d’utiliser un dispositif d’acquisition temps réel et interactif, tel que *Fiberio*.

1.4 Conclusion

Dans ce chapitre, nous avons présenté les différentes technologies utilisées dans la littérature et dans nos travaux de recherche pour identifier les doigts. Ces technologies se divisent en trois familles, chacune souffrant de ses propres limitations. La première famille se base sur l'analyse de la géométrie des points de contact. Elle ne peut identifier qu'une sous partie des accords mais surtout elle ne peut pas identifier des doigts seuls. La seconde est basée sur l'utilisation de capteurs externes (*ex.* caméra). L'identification des doigts est fiable mais nécessite un certain contrôle des conditions de captation (*ex.* pas d'utilisation nomade). La dernière famille se base sur l'utilisation de marqueurs ou capteurs placés sur l'utilisateur. Une utilisation nomade est envisageable mais contraignante puisqu'elle implique d'équiper l'utilisateur. Nous avons de même décrit les prototypes développés et en cours de développement, qui nous ont permis d'effectuer nos recherches. Ces prototypes font partie des deux dernières familles.

Les solutions technologiques actuelles ne sont pas envisageables dans un contexte d'utilisation grand public. Une technologie candidate doit être fiable, temps réel et surtout doit pouvoir être utilisée n'importe où, n'importe quand. La reconnaissance des empreintes digitales est une candidate potentielle. Nous avons donc présenté une étude en deux parties analysant la viabilité de son utilisation. Les résultats tendent à montrer que cette caractéristique peut répondre au besoin. Cependant, en l'état actuel la reconnaissance des empreintes digitales seules n'est pas suffisante. Il lui faudrait être couplée à d'autres technologies pour permettre pleinement l'identification des doigts.

Développer des outils qui permettent l'identification des doigts est un problème très répandu dans la littérature. Cependant, peu de travaux s'attardent sur les implications de cette nouvelle information sur l'utilisateur. Sommes nous capables de l'appréhender tant sur le plan cognitif que sur le plan physique ? Dans le chapitre suivant, nous présentons une série d'études visant à mieux cerner l'impact de l'identification des doigts sur l'utilisateur.

Chapitre 2

Facteurs d'intégration

En tant que nouvelle information, l'impact de l'identification des doigts sur l'utilisateur est peu, voire pas connue. Or contraindre l'utilisation d'un accord ou d'un doigt en particulier change la façon d'aborder l'interaction. L'accord à utiliser doit être choisi sciemment et ne peut être remplacé par un autre. Y-a-t-il des contraintes physiques? Y-a-t-il un effort intellectuel particulier à faire? L'utilisation de l'identification des doigts pose donc plusieurs questions tant sur le plan moteur que sur le plan cognitif. Au cours de nos recherches, nous avons étudié l'aspect dextérité de la motricité, ainsi que deux aspects de la cognition (le raisonnement et le (ré-)apprentissage). Le but de ces trois études est de mieux cerner les facteurs favorisant ou non l'utilisation de l'identification des doigts dans un contexte grand public.

La première étude porte sur la différence de performance et de préférence entre les différents doigts et accords. La plupart des techniques d'interaction basées sur l'identification des doigts associent une commande à un accord. Mais ces choix de conception sont arbitraires ou reposent sur des hypothèses sur l'habileté des utilisateurs. Par exemple : préférer utiliser l'index ou le majeur pour la manipulation précise de paramètres plutôt que l'auriculaire. Ces hypothèses semblent *a priori* justifiées, pourtant aucune étude ne les supportent. Notre étude a donc pour but d'établir une partie de la connaissance nécessaire pour former de telles hypothèses. En comparant les différents doigts et accords *via* des tâches de pointage, nous souhaitons exposer les éventuelles différences de performances et de préférences, dans le but de discuter les choix d'association entre commandes et accords. Grâce aux résultats de ces expériences, nous proposons une série de recommandations.

Dans la seconde étude, nous nous intéressons à l'influence de ces techniques sur la stratégie des utilisateurs. Pour résoudre une tâche, un utilisateur accomplit une séquence d'actions canoniques. L'ordre des actions varie en fonction de la stratégie choisie : l'utilisateur peut choisir d'effectuer toutes les actions canoniques utilisant un outil particulier avant de passer à l'outil suivant (*i.e.* stratégie centrée-outil) ou bien encore d'accomplir toutes les actions nécessaires à la réalisation d'une entité avant de passer à l'entité suivante (*i.e.* stratégie centrée-objet). Une stratégie

est une combinaison de ces deux extrêmes et est influencée par la technique d'interaction : une technique ayant un coût élevé en temps de changement d'outils, favorisera le choix d'une stratégie centrée-outil. Nous proposons deux métriques quantifiant la composante centrée-outil et la composante centrée-objet d'une stratégie. Après avoir validé ces métriques, nous nous sommes intéressés à l'influence de deux techniques basées sur l'identification des doigts minimisant le coût moteur de changement d'outils sur la stratégie utilisateur. Les résultats montrent que la réduction de ce coût ne mène pas nécessairement à des stratégies centrées-objet. En effet, le coût cognitif de ces techniques n'est pas négligeable démontrant ainsi l'importance de guider l'utilisateur au travers de l'expressivité de l'identification des doigts.

La dernière étude porte sur l'intégration des techniques utilisant l'identification des doigts. Les utilisateurs sont déjà habitués à de nombreuses interactions tactiles bien ancrées dans les systèmes commerciaux (*ex. pinch-and-expand**, défilement, *etc.*). Introduire l'identification des doigts signifie restreindre l'utilisation de ces gestes à certains accords. À l'inverse, réserver un nombre trop important d'accords pour ces interactions communes signifie réduire le potentiel de l'identification des doigts. Afin d'évaluer cette possible limitation, nous menons une étude longitudinale visant à observer quels accords sont utilisés pour les interactions communes : par exemple, le *pinch-and-expand** est-il très majoritairement déclenché par un accord pouce + index ? Si oui, les autres accords pouce + doigt seraient libres d'être associés à d'autres commandes. Les résultats indiquent un juste milieu entre les deux extrêmes. L'utilisation de l'identification des doigts et l'adaptation aux habitudes des utilisateurs ne semblent donc pas incompatibles.

La suite du chapitre est séparée en deux parties distinctes : une partie sur la dextérité qui présente la première étude et une partie sur la cognition qui regroupe les deux dernières études. Dans chacune des parties, nous proposons un état de l'art sur les travaux liés aux différents aspects abordés. Nous décrivons ensuite les protocoles expérimentaux utilisés avant d'analyser les résultats. Enfin, nous discutons les implications de ces résultats sur l'identification des doigts.

Dans la suite, nous commençons par aborder l'aspect moteur de l'identification des doigts, en parlant de la dextérité intrinsèque à chacun de nos doigts. En règle générale, la modalité tactile est comparée à d'autres modalités. Puisque chacun des doigts était perçu de la même manière par nos écrans, l'étude des différences entre les doigts n'était pas nécessaire. L'identification des doigts ayant changé la donne, nous nous intéressons à ces différences.

2.1 Dextérité des doigts et des accords

Les travaux décrits dans cette section ont été publiés à la conférence CHI 2016 [GNCV16].

Tout d'abord, nous abordons l'identification des doigts d'un point de vue moteur. Dans cette section, nous présentons une étude analysant la dextérité

des différents doigts et leurs conséquences sur le développement de techniques d'interaction basées sur l'identification des doigts.

La plupart des travaux de recherche qui se sont intéressés aux performances des entrées tactiles imposaient l'utilisation d'un seul doigt de la main dominante [HB11, MDLB13, CH04, SM15, BLZ13, BPO⁺15, PH10, PKB06, SMS09, PFB04] ou de l'index et du pouce de la main dominante pour des transformations d'objets [ZSB15, ACVL14]. Peu d'études se concentrent sur la précision et la vitesse de chaque doigt pris individuellement ou combiné en accord [GHB⁺13]. La précision et la vitesse de pointage de l'index de la main dominante, du pouce de la main dominante ou des deux pouces ont été étudiées pour des appareils tenus en main [NBW14, PH08, LGK15, AZ12]. Le pouce et certains accords ont été indirectement évalués au travers de techniques d'interaction [BLD⁺11, GCS⁺14a]. Les cinq doigts de la main dominante ont été évalués combinés en accord dans des tâches demandant d'effectuer des *flicks*^{*} dans différentes directions [LGF10] et individuellement dans des tâches de pointages [WR09b, CH14, RGB⁺15].

À notre connaissance, aucune étude n'a examiné la précision et la vitesse des dix doigts pris individuellement sur des tâches de pointage et de déplacement d'objets. De même, aucune étude n'a examiné la précision et la vitesse d'accords à deux doigts spécifiques sur des tâches de pointage et de transformation d'objets (avec déplacement, rotation et changement de taille). Or cette connaissance est un point clé pour la conception de techniques d'interaction utilisant l'identification des doigts. En effet, elle permettrait d'établir un cadre de réflexion aux concepteurs pour choisir les différentes associations entre commande et accord. Nous souhaitons donc améliorer la connaissance des différences de performance et de préférence entre les différents doigts et accords, pour mieux informer les concepteurs et chercheurs développant des techniques d'interactions basées sur l'identification des doigts.

Dans un premier temps, nous proposerons un état de l'art sur les différents travaux étudiant les performances des utilisateurs sur la modalité tactile. Nous focaliserons cet état de l'art sur les travaux ayant reporté les doigts et/ou accords utilisés. Nous présentons ensuite des résultats empiriques sur la précision, le *throughput*^{*} et la préférence subjective des 10 doigts et de 7 accords à deux doigts, collectés auprès de 48 participants au travers de 5 expériences réparties en 2 études. En s'appuyant sur ces résultats, nous proposons une série de recommandations sur la taille des cibles pour chacun des doigts et des accords, ainsi qu'un classement relatif de la pertinence d'utilisation des différents doigts et accords pour les actions multi-points usuelles.

2.1.1 État de l'art

Dans cet état de l'art, nous nous intéressons aux différentes études de performance et de préférence réalisées sur écran tactile. Contrairement à la section 1.1 qui abordait l'aspect technologique de la modalité tactile, nous nous concentrons ici sur l'aspect dextérité du facteur humain.

Il existe de nombreuses études analysant une multitude d'aspects de l'interaction tactile, comme la performance *via* des expériences de *Fitts* de pointage [SS91, MSY07, FWSB07], de glissé [CAG12] et *crossing** [LV14] ainsi que la préférence subjective [SS91, FWSB07] et la précision [PWS88, HCRC88]. Ces études ont permis de d'émettre des recommandations comme avoir une taille de cible entre 11.5 mm [WR09b, SS91] et 26 mm [HCRC88] et d'établir le *throughput** du pointage entre 3 *bps** [MSY07] et 8 *bps** [FWSB07]. Cependant, aucune de ces études ne contrôle explicitement le doigt utilisé lors des expériences. Elles laissent implicitement penser que c'est l'index de la main dominante qui a été utilisé.

Nous concentrons notre état de l'art sur les études qui imposent ou tout du moins rapportent quels doigts ont été testés (*cf.* Table 2.1).

2.1.1.1 Tâches spécifiques

Tâches liées au pointage McManus *et al.* ont mesuré le temps entre deux *tap** consécutifs d'un même doigt (*i.e.* *inter-tap interval*) pour chacun des dix doigts [MKG86]. Leur expérience a montré que l'index de la main dominante est le doigt le plus rapide alors que l'annulaire de la main dominante est le plus lent. De même, la main dominante est plus rapide que la main non-dominante. Lepinski *et al.* ont comparé des *flicks** effectués dans différentes directions avec trois doigts seuls, deux accords à deux doigts, un à trois doigts, un à quatre et un à cinq [LGF10]. Ils ont ensuite testé l'effet de l'accord utilisé sur la précision et la vitesse angulaire. Ils ont trouvé que l'accord affecte significativement à la fois le temps et l'erreur : plus l'accord comporte de doigts plus le mouvement prend du temps et est enclin aux erreurs (à noter que les erreurs étaient souvent dues à une mauvaise détection). Banovic *et al.* ont mené une étude sur le pointage en utilisant des accords et ont proposé des recommandations pour la conception de menus circulaires multi-points [BLD⁺11]. Dans cette étude, ils comparent des accords à deux doigts de la main dominante qui incluent toujours le pouce ou l'index (appelé ancre) afin d'étudier le temps de sélection d'une cible secondaire. Le temps de sélection démarre une fois l'ancre placée. Dans le cas où l'ancre est le pouce, ils reportent que l'accord pouce + auriculaire est significativement plus lent que les autres accords pouce + doigt. Ils reportent aussi que dans le cas où le pouce est l'ancre, les sélections engendrent plus d'erreurs.

Tâches de transformations Zhao *et al.* se sont concentrés sur la transformation Rotate-Scale-Translate (*RST*)* [ZSB15]. Bien que l'accord à utiliser ne soit pas exactement imposé, les participants devaient soit utiliser l'accord pouce + index soit l'accord pouce + majeur de la main dominante. À l'aide des temps et des *throughput** trouvés empiriquement, leur travail a permis d'étendre le calcul de l'indice de difficulté de la loi de *Fitts* aux tâches de *RST**. Hoggan *et al.* ont étudié la rotation utilisant l'accord pouce + index de la main dominante [HWO⁺13]. Leur plan expérimental faisait varier l'angle, le diamètre et la position de la rotation sur l'écran. Les résultats soulignent les contraintes ergonomiques de la main : les rotations dans le sens horaire sont plus lentes

(respectivement plus rapides) quand l'angle initial est plus petit (respectivement plus grand) que 120° .

2.1.1.2 Pointage

Le pointage est la tâche la plus évaluée pour les interactions tactiles. Dans cette partie, nous nous intéressons aux études imposant un doigt seul (l'index ou le pouce de la main dominante), plusieurs doigts seuls et tous les doigts de la main dominante.

Index de la main dominante Holtz *et al.* ont exploré les *modèles mentaux* qu'ont les utilisateurs quand ils pointent le centre d'une croix [HB11]. Ils montrent que les utilisateurs ont généralement un décalage entre la position du contact et la position perçue de 4 mm. En se basant sur les modèles trouvés, ils ont pu réduire ce décalage à 1,6 mm. Également en étudiant le décalage, Möllers *et al.* ont trouvé qu'en tenant compte de l'orientation d'approche du doigt sur la surface la précision du pointage sur écran tactile pouvait être augmentée [MDLB13]. À l'aide de tâche de pointage sur un clavier numérique pour lequel la taille et l'espacement des boutons variaient, Herbert *et al.* recommandent une taille de touches de 20 mm [CH04]. MacKenzie *et al.* ont eux comparé différentes façons de tenir un *smartphone** sur des tâches de pointage à une et deux dimensions [SM15]. Ils ont trouvé un *throughput** 15% supérieur pour le pointage à une dimension et aucun effet de la prise. Bi *et al.* ont proposé une amélioration de la loi de Fitts pour l'acquisition de petites cibles sur écran tactile en introduisant le modèle *FFitts* [BLZ13]. En faisant varier les facteurs de forme (tablette, ordinateur portable, grand écran, large écran public, *smartphone** tenu à deux mains, *smartphone** tenu à une main), Bachynskyi *et al.* ont mesuré le *throughput** et l'activité musculaire [BPO+15]. Ils rapportent qu'en moyenne le *throughput** est de 6,55 *bps**, que sur grand écran et *smartphone** tenu à deux mains il est supérieur d'environ 20% et que sur tablette et ordinateur portable il est inférieur d'environ 11%. Sasangohar *et al.* ont comparé le *throughput** sur grand écran avec une souris et avec la modalité tactile : avec une souris, il est de 5,53 *bps** alors qu'avec la modalité tactile il est de 3,83 *bps** [SMS09]. Ils rapportent aussi un plus grand taux d'erreur avec la condition tactile. Po *et al.* ont comparé les champs visuels haut et bas lors de l'utilisation d'écran tactile [PFB04]. Lorsque l'écran est dans le champ visuel bas la sélection de cibles engendre de meilleures performances.

Pouce de la main dominante Park *et al.* ont étudié la précision sur *smartphone** tenu à une main [PH10]. Après analyse des points de contact pour différentes tailles et positions de cibles, ils proposent une solution aidant la conception de boutons qui améliorent le taux de succès. Parhi *et al.* ont montré que des cibles ayant une taille entre 9 et 10 mm peuvent être sélectionnées avec succès en utilisant le pouce de la main dominante [PKB06].

	Doigts seuls	Plusieurs doigts seuls	Accords
Pointage	I_d Holtz <i>et al.</i> [HB11] (D) Möllers <i>et al.</i> [MDLB13] (D) Herbert <i>et al.</i> [CH04] (T, E) MacKenzie <i>et al.</i> [SM15] (T, E, T _h) Bi <i>et al.</i> [BLZ13] (T, E, P _r) Bachynskyi <i>et al.</i> [BPO ⁺ 15] (T _h , AM) Sasangohar <i>et al.</i> [SMS09] (T, E, T _h , P _f) Po <i>et al.</i> [PFB04] (T, E) P_d Park <i>et al.</i> [PH10] (E, D) Parhi <i>et al.</i> [PKB06] (T, E)	Pouces Perry <i>et al.</i> [PH08] (T, E, IP) P_d, I_d Ljubic <i>et al.</i> [LGK15] (T) Pouces, P_d, I_d Ng <i>et al.</i> [NBW14] (T, E, P _r) Azenkot <i>et al.</i> [AZ12] (V, P _r , D) Main dominante Wang <i>et al.</i> [WR09b] (P _r , D) Colley <i>et al.</i> [CH14] (T, E, P _f) Roy <i>et al.</i> [RGB ⁺ 15] (T, E, T _h)	Accords à 2 doigts Banovic <i>et al.</i> [BLD ⁺ 11] (T, E)
Glissé	Rien	Rien	Rien
Transformations	Rien	Rien	PI_d ou PM_d Zhao <i>et al.</i> [ZSB15] (T) Hoggan <i>et al.</i> [HWO ⁺ 13] (T)
Autres		Tous les doigts McManus <i>et al.</i> [MKG86] (<i>inter-tap</i>)	Accords de 1 à 5 doigts Lepinski <i>et al.</i> [LGF10] (<i>flück*</i>)

TABLE 2.1: Résumé des critères évalués dans les travaux précédents pour un doigt seul, plusieurs doigts seuls et des accords au travers de tâches de pointage, de glissé et de transformation. Explications des critères : D pour *décalage*, T pour *temps*, E pour *erreur*, T_h pour *throughput**, P_r pour *précision*, AM pour *activité musculaire*, P_f pour *préférence*, IP pour *indice de performance*, V pour *vitesse*

Pouce de la main dominante et pouce de la main non-dominante

Perry *et al.* ont comparé les performances sur *smartphone*^{*} entre les deux pouces lorsque l'utilisateur est debout immobile et marche [PH08]. Ils ont montré une performance supérieure pour le pouce de la main dominante qui, en moyenne, est plus rapide de 100 ms et plus précis de 5%.

Pouce de la main dominante et index de la main dominante

Ljubic *et al.* ont réalisé une synthèse des modèles existants de la loi de Fitts [LGK15]. Ils ont aussi montré que cette loi permet de correctement prédire le temps de pointage pour des cibles rectangulaires sur écrans tactiles. Dans leur étude, ils ont varié le facteur de forme (tablette et *smartphone*^{*}) et la façon d'interagir (à une ou à deux mains).

Deux pouces, pouce de la main dominante et index de la main dominante

Ng *et al.* ont montré que lorsque les utilisateurs marchent et sont encombrés, l'interaction sur *smartphone*^{*} devient significativement plus mauvaise [NBW14]. Azenkot *et al.* ont eux exploré la distribution des contacts tactiles sur clavier *QWERTY* et ont étudié la précision des utilisateurs [AZ12]. En se basant sur l'analyse des données collectées, ils donnent des points importants à prendre en compte dans la conception de ces claviers. Par exemple, en fonction de la prise et des doigts utilisés, la forme et la position des touches du clavier doivent être adaptées. De même, en fonction de ces mêmes variables, des corrections sur la position des contacts peuvent être effectuées pour faciliter l'action de pointage de l'utilisateur.

Doigts de la main dominante

Colley *et al.* ont comparé les performances sur des tâches de pointage des cinq doigts de la main dominante [CH14]. Ils ont trouvé des différences significatives pour les taux d'erreur et les temps de réalisation entre l'index (le meilleur) et l'auriculaire (le moins bon). Les notes subjectives données par les participants confirment leurs résultats : l'index est perçu comme le plus confortable et le plus rapide comparé à tous les autres doigts, alors que l'auriculaire est perçu comme le moins confortable et le plus lent. Roy *et al.* ont obtenu des résultats similaires pour les doigts de la main dominante : le temps de réaction, le temps d'exécution, le temps de réalisation et le taux d'erreur étaient significativement meilleurs pour l'index de la main dominante et moins bon pour l'auriculaire [RGB⁺15]. Dans ces deux dernières études, aucune différence significative n'a été trouvée entre les autres paires de doigts. Wang *et al.* ont évalué la précision du *tap*^{*} pour tous les doigts de la main dominante et ont trouvé que l'index, le majeur, et l'annulaire sont plus précis que le pouce et l'auriculaire [WR09b]. Ils rapportent aussi que l'auriculaire a été perçu comme le plus difficile à utiliser.

2.1.1.3 Les informations manquantes

Comme suggéré dans la table 2.1, les travaux de l'état de l'art qui ont imposé un certain contrôle sur les doigts utilisés se sont principalement focalisés sur le

pointage et peu ont examiné plus de un ou deux doigts utilisés individuellement. Les études les plus poussées n'ont établi des comparaisons qu'entre les différents doigts de la main dominante. De plus, à notre connaissance, aucun travail n'a été effectué sur le glissé-déposé.

Pour les accords, le constat est similaire : à l'exception de Zhao *et al.* et Hoggan *et al.*, aucune recherche ne s'est penchée sur la précision et la vitesse d'un ensemble d'accords dans des tâches de pointage ou de déplacement.

Nous nous proposons donc dans la suite, d'étudier l'ensemble des doigts et certains accords au travers des tâches usuelles de la modalité tactile.

2.1.2 Sélection de cible

L'objectif de cette première étude est de déterminer la taille optimale des cibles pour la sélection en fonction du doigt ou de l'accord utilisé. Nous utilisons des cibles circulaires de différentes tailles plutôt que de simples croix telles qu'utilisées par Bi *et al.* dans leur étude de la loi de *FFitts* [BLZ13]. En effet, après avoir effectué une étude préliminaire, nous avons observé que les participants tendaient à fortement pincer leurs doigts entre eux lors de la sélection d'une cible en forme de croix avec un accord à deux doigts, alors qu'une cible circulaire qui devient de plus en plus large encourage une sélection avec une position des doigts plus *confortable*. Les résultats de cette étude motivent nos choix de tailles de cibles pour notre étude principale sur les différences de performance (*cf.* Section 2.1.3).

2.1.2.1 Participants

Nous avons recruté 12 volontaires (23,1 ans en moyenne, 3,5 ans de déviation standard, 5 hommes, 1 gaucher, tous utilisateurs d'écran tactile). Chaque participant était rémunéré \$10.

2.1.2.2 Dispositif

L'expérience a été implémentée en *C++ QT 5.3* et a été effectuée sur une tablette *Nexus 10*. Les stimuli visuels et les *feedback** ont été réduits au minimum pour limiter la latence. En utilisant la méthode de Ng *et al.* [NLW⁺12] et la caméra à 240 *fps** d'un *iPhone 6*, la latence de bout-en-bout, mesurée entre le déplacement du doigt sur l'écran et la mise à jour de l'image affichée, a été estimée autour de 100 ms. Cette latence, qui est dans la moyenne des applications réelles, (*i.e.* les écrans tactiles actuels ont une latence qui varie entre 50 et 200 ms [DAJ⁺16, NLW⁺12]) fait de notre application un test écologiquement valide. À noter que même si une latence plus faible augmenterait la performance absolue, nous sommes intéressés par la performance relative entre les différents doigts. L'écran de 264 mm × 178 mm a une résolution de 2560 px × 1600 px et une densité de pixels de 11,8 px/mm (300 *ppp**). Les participants étaient assis sur une chaise face à un bureau où ils utilisaient la tablette posée à plat dans le sens paysage. Poser la tablette à plat permet d'éviter que la tenue de l'appareil soit une variable de confusion en augmentant la fatigue des participants dans certaines

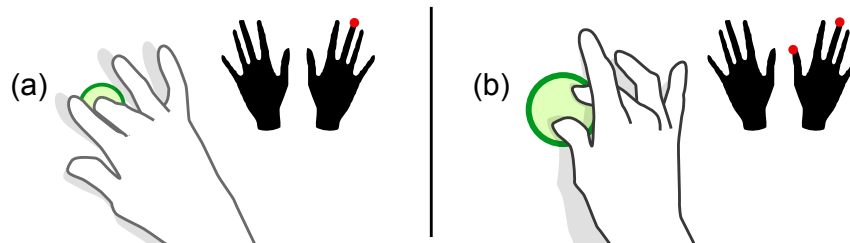


FIGURE 2.1: Exemples de tâche de l'étude *Sélection de cible*. Les participants effectuent des *tap** sur des cibles circulaires (ici montrées en vert) en utilisant un doigt ou un accord imposé : (a) exemple avec le doigt M_d ; (b) exemple avec l'accord PM_d . Le doigt ou l'accord utilisé est indiqué à l'aide d'une icône représentant les mains (ici montrée en noir).

conditions. De plus, nous gardons une cohérence avec des études précédentes, ce qui permet une comparaison plus aisée [BLD⁺11, LV14]. Nos résultats se limitent à l'utilisation d'une tablette à plat sur un bureau, mais en pratique l'angle formé par le bras et la tablette est similaire à celui formé quand la tablette est posée sur les genoux ou tenue à une main.

2.1.2.3 Tâche

Pour cette tâche, les participants devaient effectuer un *tap** sur une cible circulaire en utilisant un doigt ou un accord spécifique (*cf.* Figure 2.1). Les cibles étaient affichées une par une à des positions aléatoires sur l'écran. Si les contacts physiques sur l'écran tactile étaient détectés en dehors de la cible, aucun *feedback** d'erreur n'était affiché. Étant donné que notre objectif est de mesurer une taille de cible confortable, aucune instruction particulière concernant la vitesse ni la précision n'a été donnée aux participants.

2.1.2.4 Protocole

Les participants étaient conviés à répondre à un court questionnaire collectant des informations démographiques, puis l'expérimentateur présentait la tâche ainsi que le fait que les doigts et les accords à utiliser étaient imposés. Les participants pouvaient ensuite commencer l'expérience. Le plan expérimental groupait les ESSAIS d'un même doigt ou accord et séparait l'expérience en une section pour les doigts et une section pour les accords. Avant de commencer chaque section, les participants avaient la possibilité de s'exercer sans limite de temps. Une icône représentant les mains (en noir dans la figure 2.1) rappelait aux participants le doigt ou l'accord en cours d'utilisation. Ce même graphique était affiché avant chaque groupe d'ESSAIS. Lorsque le doigt ou l'accord changeait entre deux groupes d'ESSAIS, un message indiquant ce changement était affiché durant 5 secondes pour prévenir les participants du changement et les inciter à prendre une pause. En moyenne, l'expérience durait 40 minutes.

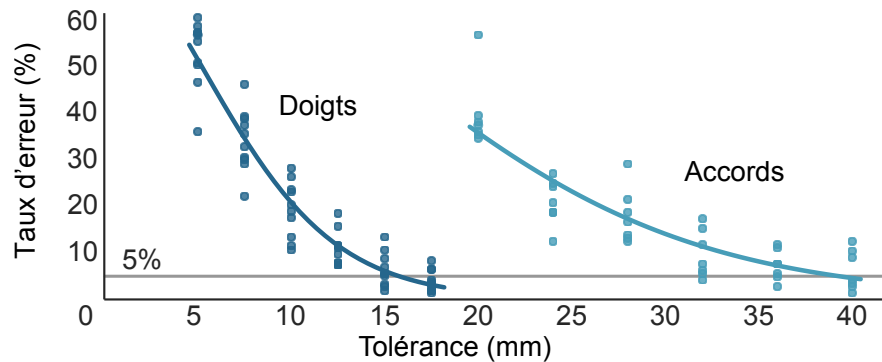


FIGURE 2.2: Taux d'erreur par CONTACT en fonction de la TOLÉRANCE.

2.1.2.5 Plan expérimental

La principale variable indépendante était le doigt ou l'accord utilisé pour effectuer les CONTACTS sur l'écran. En tout nous avons testé 17 CONTACTS différents couvrant l'ensemble des 10 DOIGTS (P_d , I_d , M_d , A_d , L_d pour la main dominante et P_n , I_n , M_n , A_n , L_n pour la main non-dominante, cf. exemple de la figure 2.1-a) et 7 ACCORDS à 2 doigts (les accords *pincés* de la main dominante PI_d , PM_d , PA_d , PL_d et les accords *adjacents* de la main dominante IM_d , MA_d , AL_d , cf. exemple de la figure 2.1-b). L'utilisation des termes *dominante* et *non-dominante* permet de normaliser l'appellation des mains pour les participants *droitiers* et *gauchers*. Certains des ACCORDS comme PI_d , PM_d , IM_d et MA_d sont déjà couramment utilisés sur les interfaces multi-points pour le zoom, la rotation et le défilement. Nous avons donc ajouté d'autres ACCORDS similaires dans la limite d'une durée d'expérience acceptable.

Une deuxième variable indépendante est la TOLÉRANCE de la sélection, qui dans cette expérience revient au diamètre de la cible circulaire. Six TOLÉRANCES ont été testées. Pour les CONTACTS DOIGTS : 17,5, 15, 12,5, 10, 7,5 et 5 mm. Pour les CONTACTS ACCORDS : 40, 36, 32, 28, 24 et 20 mm. Une expérience préliminaire nous a permis de déterminer ces tailles de cibles, allant de la sélection difficile et inconfortable à la sélection facile et plus que confortable.

L'expérience comporte 3 BLOCS, chaque BLOC comprenant 6 répétitions de chaque combinaison de facteurs. Tous les ESSAIS d'un même CONTACT sont regroupés ce qui signifie que les participants utilisaient le même DOIGT ou ACCORD durant une série d'ESSAIS couvrant toutes les TOLÉRANCES. Ce choix permet de réduire l'effort cognitif nécessaire pendant l'expérience et aide à se concentrer sur la précision et non sur la vitesse de réactivité lors de changement de CONTACT. L'ordre des CONTACTS était aléatoire mais respectait les contraintes suivantes : les ACCORDS et les DOIGTS étaient regroupés dans deux sections séparées ; l'ordre des sections était contrebalancé entre les participants ; dans la section DOIGTS, les CONTACTS alternés d'une main à l'autre ; dans la section ACCORDS, les CONTACTS alternaient entre les types *pincés* et *adjacents*. Le

	L _n	A _n	M _n	I _n	P _n	P _d	I _d	M _d	A _d	L _d
Tol (mm)	17,5	17,5	17,5	17,5	17,5	17,5	17,5	15	17,5	17,5
Err %	6,5%	6,5%	1,4%	1,9%	3,2%	3,2%	1,9%	1,9%	4,2%	8,3%
	PI _d	PM _d	PA _d	PL _d	IM _d	MA _d	AL _d			
Tol (mm)	36	40	40	40	36	40	40			
Err %	1,4%	4,2%	9,0%	12,5%	2,8%	3,5%	10,4%			

TABLE 2.2: TOLÉRANCE à laquelle le taux d’erreur pour un CONTACT donné commence à être inférieur à 5%. Les valeurs en gras correspondent aux CONTACTS n’étant jamais descendu sous les 5% (les valeurs rapportées sont les taux d’erreur pour la TOLÉRANCE maximale).

plan expérimental était donc : 17 CONTACTS $\times$ 3 BLOCS $\times$ 6 TOLÉRANCES $\times$ 6 RÉPÉTITIONS = 1 836 ESSAIS par participant. En moyenne, l’expérience durait 30 minutes par participant.

2.1.2.6 Analyse

Toutes les analyses sont des *ANOVA* multi-variées : le participant est considéré comme une variable aléatoire utilisant la procédure *REML* du paquet *SAS* de *JMP*. Des tests de *Tukey* sont effectués *post hoc* quand des effets significatifs sont trouvés.

2.1.2.7 Résultats

Dans cette partie, nous rapportons séparément les analyses statistiques pour DOIGTS et ACCORDS. Les résultats statistiques sont présentés de manière brefs et concis, nous encourageons le lecteur à examiner la figure 2.2 pour un aperçu des tendances générales des taux d’erreur et consulter la table 2.2 pour les recommandations de tailles basées sur cette analyse.

Doigts *Apprentissage et/ou Fatigue* — Aucun effet significatif de BLOC sur le taux d’erreur, tous les BLOCS sont inclus dans les futures analyses.

Erreur — Effet significatif de DOIGT sur le taux d’erreur ($F_{9,99} = 4,5, p < 0,0001$). Les analyses *post hoc* montrent : A_n (28%) a un taux d’erreur significativement plus important ($p < 0,05$) que M_d (17%), I_n (17%) et I_d (14%); I_d (14%) a un taux d’erreur significativement plus bas ($p < 0,05$) que A_n, L_d, L_n, P_n et P_d (moyennes entre 24% et 28%). Un effet significatif de la TOLÉRANCE sur le taux d’erreur, respectant le comportement escompté, a été trouvé pour chaque DOIGT (tous les $p < 0,0001$). Les analyses *post hoc* révèlent que les TOLÉRANCES de 5 et 7,5 mm ont des taux d’erreur plus importants ($p < 0,05$). En traitant les TOLÉRANCES séparément, il y a un effet significatif de DOIGT sur le taux d’erreur sauf pour la TOLÉRANCE 12,5 mm où l’erreur moyenne est de 7%.

Accords *Apprentissage et/ou Fatigue* — Un effet significatif de BLOC sur le taux d’erreur a été trouvé ($F_{2,22} = 6,3, p < 0,007$). Les analyses *post hoc* révèlent

que le BLOC 3 a un taux d'erreur plus important que le BLOC 1 et 2 ($p < 0,05$) ce qui suggère un effet de fatigue. Nous enlevons le BLOC 3 pour le reste de nos analyses.

Erreur — Aucun effet significatif d'ACCORD sur le taux d'erreur (moyennes allant de 12% à 25%). Cependant un effet significatif de la TOLÉRANCE sur le taux d'erreur a été trouvé pour chaque ACCORD (tous les $p < 0,0001$). Les analyses *post hoc* révèlent que la TOLÉRANCE de 20 mm a un taux d'erreur plus important ($p < 0,05$) que toutes les autres TOLÉRANCES.

Caractéristiques des contacts — Pour examiner le confort, nous avons calculé la distance moyenne (en mm) entre les deux doigts composant un ACCORD au moment de la sélection. Nous interprétons cette distance comme un indicateur du confort de l'ACCORD lors de l'acquisition de la cible. Lorsque la distance moyenne est affichée en fonction de la taille de la cible, (*cf.* Figure 2.3), nous considérons qu'un plateau indique que les tailles de cibles correspondantes sont confortables à sélectionner.

Un effet significatif de TOLÉRANCE sur cette distance a été trouvé ($F_{6,451} = 7,8$, $p < 0,0001$). Des analyses *post hoc* révèlent que la distance entre les doigts pour les cibles de 40 et 36 mm de diamètre (moyenne de la distance de 16,8 mm) est significativement plus élevée que pour les cibles de 24 et 20 mm de diamètre (moyenne de la distance de 14,4 mm) ($p < 0,01$). Une analyse séparée pour chaque ACCORD confirme ce résultat pour IM_d, AL_d, PM_d et PA_d (tous les $p < 0,05$). Pour l'accord MA_d, seules les cibles de 40 et 20 mm de diamètre sont significativement différentes ($p < 0,05$). Pour l'accord PL_d, la distance était significativement plus grande pour la cible de 40 mm que pour les cibles de 20 et 24 mm ($p < 0,01$). Enfin, malgré un effet significatif, des analyses *post hoc* n'ont montré aucune différence entre les TOLÉRANCES pour PL_d.

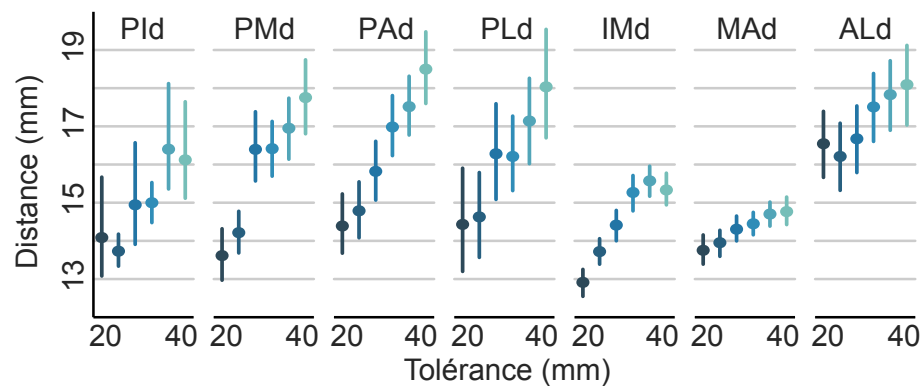


FIGURE 2.3: Distances moyennes entre les deux doigts composant un ACCORD pour chacune des TOLÉRANCES.

2.1.2.8 Discussion

Pour la plupart des doigts, à l'exception de L_n , A_n et L_d , les cibles dont le diamètre dépasse 17,5 mm permettent de confortablement descendre le taux d'erreur sous la barre des 5%. Ce résultat est conforme aux travaux de l'état de l'art qui recommandent des tailles de cibles entre 11,5 mm [WR09b, SS91] et 26 mm [HCRC88]. Pour les accords à deux doigts, la taille minimale d'une cible doit être d'au moins 36 mm voire souvent 40 mm. À noter que dans notre étude, les participants n'ont reçu aucune instruction concernant la précision et n'ont pas été exposé à un quelconque *feedback** les informant de leurs erreurs. Pour mettre l'accent sur la précision dans les expériences suivantes, nous testons intentionnellement des TOLÉRANCES plus petites que les tailles recommandées.

2.1.3 Performance et préférence

L'objectif de cette deuxième étude est de mesurer les performances et les préférences des différents doigts et de différents accords au travers de quatre tâches courantes : le pointage, le déplacement d'objet, le changement d'échelle et la rotation. L'étude se compose de quatre expériences s'apparentant à des expériences de *Fitts*, chacune dédiée à une unique tâche. Toutes ces expériences ont été menées simultanément au cours d'une seule et même étude. Les participants recrutés étaient associés aux expériences de façon aléatoire. La performance est mesurée à l'aide du temps de réalisation, du taux d'erreur, du *throughput** et des paramètres de régression du modèle de la loi de *Fitts*. La préférence est mesurée grâce à une notation subjective.

Dans cette section, nous décrivons les aspects communs entre les quatre expériences. Les descriptions spécifiques aux tâches et les résultats seront donnés dans les sections suivantes pour chacune des expériences (*cf.* Sections 2.1.4, 2.1.5, 2.1.6 et 2.1.7).

2.1.3.1 Dispositif

Toutes les expériences utilisent le même dispositif que celui de la première étude (*cf.* Section 2.1.2.2).

2.1.3.2 Participants

Au total, 36 participants ont été recrutés. Tous étaient utilisateurs d'écrans tactiles multi-points. Les participants ont été répartis comme suit : 12 pour l'expérience de pointage ; 12 pour l'expérience de déplacement d'objet ; 12 pour l'expérience de changement d'échelle et l'expérience de rotation. Les détails démographiques seront donnés dans les sections suivantes (*cf.* Sections 2.1.4, 2.1.5, 2.1.6 et 2.1.7). Chaque participant était rémunéré \$10.

2.1.3.3 Protocole

La procédure suivie pour toutes les expériences est similaire à celle de la première étude, à l'exception des tâches et de l'ajout de *feedback** pour les erreurs ainsi que du taux d'erreur. Le taux d'erreur correspond au nombre d'ESSAIS erronés divisé par le nombre d'ESSAIS courants. Il est remis à zéro à chaque changement de CONTACT. Lors de la présentation de la tâche, l'expérimentateur explicitait aussi les actions causant une erreur. Cette fois-ci, une instruction supplémentaire était donnée aux participants : effectuer la tâche aussi rapidement que possible tout en gardant un taux d'erreur d'environ 4%. Pour aider les participants à atteindre cet objectif, l'application affichait le taux d'erreur correspondant au CONTACT courant. À la fin de l'expérience, les participants ont noté, pour chaque CONTACT, la facilité d'utilisation, la rapidité, la précision et la fatigue (ou plutôt le manque de fatigue) perçues. Cette notation s'est effectuée à l'aide d'une échelle continue allant de 1 (le plus bas) à 5 (le plus haut) : *i.e.* les notations, telles que 3,5, étaient donc permises. Chaque expérience durait en moyenne entre 30 et 60 minutes en tout.

2.1.3.4 Plan expérimental

Pour chaque expérience, les variables indépendantes étaient similaires.

Tout comme la première étude, la principale variable indépendante était CONTACT. Les 17 CONTACTS différents couvrant l'ensemble des 10 DOIGTS et les 7 ACCORDS à 2 doigts ont été évalués pour les expériences de pointage et de déplacement d'objets. Seuls les ACCORDS ont été évalués pour les tâches de changement d'échelle et de rotation. Parmi les ACCORDS, les 7 ACCORDS ont été évalués pour la rotation mais seuls les 4 ACCORDS *pincés* l'ont été pour le changement d'échelle à cause du type de mouvement requis.

La difficulté de la tâche est définie par deux variables indépendantes : TOLÉRANCE et AMPLITUDE. Les définitions précises seront données dans les sections suivantes (*cf.* Sections 2.1.4, 2.1.5, 2.1.6 et 2.1.7).

Toutes les expériences comportent 3 BLOCS, chaque BLOC comprenant 6 répétitions de chaque combinaison de facteur. Comme pour la première étude, tous les ESSAIS d'un même CONTACT sont regroupés (*i.e.* les participants utilisaient le même DOIGT ou ACCORD durant une série d'ESSAIS couvrant toutes les combinaisons de TOLÉRANCE et d'AMPLITUDE). De même, l'ordre des CONTACTS était aléatoire mais respectait les contraintes de la première étude (*cf.* Section 2.1.2.5). Pour chaque CONTACT, tous les ESSAIS d'une même TOLÉRANCE étaient regroupés et l'ordre des AMPLITUDES était aléatoire. L'ordre des TOLÉRANCES était contrebalancé entre les BLOCS.

Modélisation de la loi de Fitts Pour chaque expérience, nous modélisons la performance de chacun des CONTACTS à l'aide de la loi de Fitts [Mac92] en utilisant la formulation de Shannon pour calculer l'*Indice de Difficulté*

(ID) [Mac89, Mac13] :

$$Temps = a + b \times \log_2 \left(\frac{AMPLITUDE}{TOLÉRANCE} + 1 \right) \quad (2.1)$$

Chacune des expériences possède 3 niveaux de TOLÉRANCE et 2 niveaux d'AMPLITUDE. Ces niveaux ont été choisis de sorte que la plage des ID testés soit raisonnable. Nous rapportons l'Indice de Performance ($IP = \frac{1}{b}$) par exhaustivité, mais utilisons le *throughput** (TP) comme mesure principale de la performance de chacun des CONTACTS. Pour la mesure de TP , nous utilisons la formulation utilisant la *largeur de cible effective* (L_e) recommandée par la norme ISO 9241 [ISO] :

$$TP = \frac{ID_e}{Temps} = \frac{\log_2 \left(\frac{AMPLITUDE}{L_e} + 1 \right)}{Temps} \quad (2.2)$$

Erreur dans l'enregistrement des données Une erreur dans la programmation du module d'enregistrement des données a provoqué un écrasement d'un ESSAI sur six (16%). Fort heureusement, les pertes engendrées sont directement liées à l'indexation de chaque ESSAI dans notre base de donnée. Cette indexation étant réalisée *via* de l'arithmétique modulaire, les pertes sont contrebalancées entre les conditions. Aucun ESSAI n'étant partiellement corrompu, aucun bruit n'a été introduit dans les données. Dans toutes nos analyses, nous agrégeons les multiples répétitions des ESSAIS, ce qui réduit encore plus l'impact de cette erreur.

2.1.3.5 Analyses

Toutes les analyses sont des *ANOVA* multi-variées : participant est considéré comme une variable aléatoire utilisant la procédure *REML* du paquet *SAS* de *JMP*. Des tests de *Tukey* sont effectués *post hoc* quand des effets significatifs sont trouvés. Puisque les temps de réalisation de tâche n'ont généralement pas une distribution normale, les temps médians sont utilisés lors des agrégations.

2.1.4 Performance et préférence : pointage de cible

L'objectif de cette expérience est de mesurer la performance des différents DOIGTS et ACCORDS lors de l'acquisition de cibles. 12 participants ont été recrutés pour cette expérience (22,2 ans en moyenne, 2,2 ans de déviation standard, 7 femmes). Le dispositif, le protocole, le plan expérimental et l'analyse sont décrits dans la section 2.1.3.

2.1.4.1 Tâche

La tâche consistait à pointer une cible aussi rapidement que possible en utilisant un CONTACT imposé, tout en gardant un taux d'erreur d'environ 4%. Chaque ESSAI commençait au moment où le participant effectuait un *Touch*

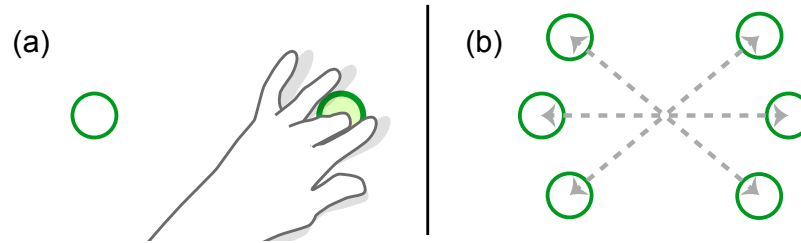


FIGURE 2.4: Tâche de pointage. Les participants effectuent un *tap*^{*} dans une cible circulaire (disque vert) après avoir effectué un *tap*^{*} dans une zone circulaire de départ (cercle vert) en utilisant un CONTACT imposé : (a) exemple utilisant le doigt M_n ; (b) les différentes directions utilisées.

Down^{*} dans une zone de départ circulaire (2 cm de diamètre pour les DOIGTS, 4 cm de diamètre pour les ACCORDS). Un ESSAI se terminait lorsque le participant effectuait un *Touch Down*^{*} et un *Touch Up*^{*} dans une cible circulaire dont la position et la géométrie étaient contrôlées (cf. Figure 2.4). Lorsqu'un point de contact était détecté en dehors de la zone cible, l'ESSAI était considéré comme erroné et l'arrière plan de l'application devenait orange.

2.1.4.2 Plan expérimental

Dans cette tâche, la TOLÉRANCE correspond au diamètre de la cible et l'AMPLITUDE correspond à la distance entre le centre de la zone de départ et le centre de la cible. Des TOLÉRANCES et AMPLITUDES différentes ont été utilisées pour les DOIGTS et les ACCORDS. Pour les DOIGTS, les TOLÉRANCES étaient de 1, 1,5, et 2 cm et les AMPLITUDES étaient de 4 et 19 cm. Les six combinaisons de TOLÉRANCE et d'AMPLITUDE produisaient des *ID* s'étalant de 1,6 à 4,3 *bits*^{*}. Pour les ACCORDS, les TOLÉRANCES étaient de 3, 3,5, et 4 cm et les AMPLITUDES étaient de 5 et 17 cm. Les six combinaisons de TOLÉRANCE et d'AMPLITUDE produisaient des *ID* s'étalant de 1,2 à 2,7 *bits*^{*}. Il est à noter que nos plages d'*ID* sont contraintes par les tailles de cibles et les distances sur la tablette qui doivent rester raisonnables.

La direction entre la zone de départ et la zone cible variait avec les 6 RÉPÉTITIONS : de gauche à droite, du coin haut gauche au coin bas droit, du coin bas gauche au coin haut gauche et leur symétrie, de droite à gauche. L'utilisation de toutes les directions, comme recommandé par l'ISO 9241 [ISO], aurait drastiquement réduit l'AMPLITUDE maximale et donc la plage des *ID*. Le plan expérimental était donc : 17 CONTACTS $\times$ 3 BLOCS $\times$ 3 TOLÉRANCES $\times$ 2 AMPLITUDES $\times$ 6 RÉPÉTITIONS par participant.

2.1.4.3 Résultats

Dans cette partie, nous rapportons séparément les analyses statistiques et les modélisations de la loi de *Fitts* pour DOIGTS et ACCORDS. Les résultats clés

	L _n	A _n	M _n	I _n	P _n	P _d	I _d	M _d	A _d	L _d
a	0,22	0,22	0,23	0,19	0,30	0,18	0,14	0,20	0,22	0,27
b	0,16	0,15	0,13	0,15	0,14	0,14	0,13	0,12	0,13	0,13
r ²	0,99	0,99	1,00	1,00	1,00	1,00	0,99	0,99	1,00	1,00
Temps	0,69	0,67	0,61	0,61	0,70	0,59	0,52	0,54	0,59	0,64
TP	4,28	4,43	4,86	4,91	4,25	5,16	5,85	5,57	5,13	4,65
IP	6,13	6,47	7,55	6,87	7,13	7,09	7,63	8,65	7,87	7,72

TABLE 2.3: Résumé pour chaque DOIGT des différents paramètres des modélisations pour l'expérience de pointage. Les valeurs extrêmes sont surlignées en vert (meilleures) et en gris (moins bonnes).

sont résumés dans les tables 2.3 et 2.4 et les tendances pour le *throughput** et les préférences sont résumées dans les figures 2.8-ab de la section 2.1.8 où toutes les expériences peuvent être comparées visuellement.

Doigts *Apprentissage et/ou Fatigue* — Aucun effet significatif de BLOC sur le temps de réalisation ni sur le taux d'erreur, tous les BLOCS sont inclus dans les futures analyses.

Erreur — Effet significatif de DOIGT sur le taux d'erreur ($F_{9,649} = 2,1$, $p < 0,05$). Les analyses *post hoc* ne montrent aucune différence par paires. Les taux d'erreurs moyens varient entre 2,07% et 4,08%. Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le taux d'erreur sont significatifs.

Temps — Effet significatif de DOIGT sur le temps de réalisation ($F_{9,649} = 33,5$, $p < 0,0001$). Les analyses *post hoc* montrent : M_d et I_d sont plus rapides que tous les autres doigts (tous les $p < 0,0001$) ; P_n est plus lent que tous les autres doigts sauf A_n et L_n (tous les $p < 0,0002$) ; L_d est plus lent que tous les autres doigts de la main dominante ($p < 0,0001$). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le temps de réalisation sont significatifs.

Modélisation de la loi de Fitts — Pour tous les DOIGTS, les modélisations ont un coefficient de régression $r^2 > 0,98$. Les différents paramètres des modélisations sont résumés dans la table 2.3.

*throughput** — Effet significatif de DOIGT sur le TP ($F_{9,649} = 32,4$, $p < 0,0001$). Les analyses *post hoc* montrent : tous les doigts de la main dominante ont un TP plus élevé que L_n et P_n ($p < 0,01$) ; I_d, M_d, P_d et A_d ont un TP plus élevé que A_n, L_n et P_n ($p < 0,01$) ; I_d, M_d et P_d ont un TP plus élevé que L_d et tous les doigts de la main non-dominante sauf I_n ($p < 0,05$) ; I_d et M_d ont un TP plus élevé que tous les autres doigts sauf P_d ($p < 0,01$).

Notes — Effet significatif de DOIGT sur la facilité d'utilisation, la rapidité, la précision et la fatigue ($F_{9,99} > 4,9$, $p < 0,0001$). Les analyses *post hoc* révèlent une tendance similaire pour toutes les notations (tous les $p < 0,0001$) : I_d, M_d, I_n et P_d ont des notes plus élevées que L_n et A_n (facilité d'utilisation, rapidité), A_n (précision) et L_n, A_d (fatigue).

Accords *Apprentissage et/ou Fatigue* — Effet significatif de BLOC sur le temps de réalisation ($F_{2,22} = 5,4$, $p < 0,05$) mais pas sur le taux d'erreur. Le BLOC 1 est

	PI _d	PM _d	PA _d	PL _d	IM _d	MA _d	AL _d
a	0,28	0,23	0,32	0,33	0,27	0,32	0,30
b	0,12	0,13	0,13	0,12	0,13	0,11	0,12
r^2	0,99	1,00	0,99	0,99	0,99	0,98	0,98
Temps	0,51	0,48	0,57	0,57	0,51	0,52	0,54
TP	5,09	5,36	4,46	4,24	5,02	5,17	4,73
IP	8,33	7,44	7,66	8,11	7,88	9,43	8,19

TABLE 2.4: Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de pointage. Les valeurs extrêmes sont surlignées en vert (meilleures) et en gris (moins bonnes).

significativement plus lent que les 2 autres BLOCS. Nous enlevons donc le BLOC 1 pour le reste de nos analyses.

Erreur — Effet significatif d'ACCORD sur le taux d'erreur ($F_{6,451} = 2,4$, $p < 0,05$). Les analyses *post hoc* montrent que PL_d cause plus d'erreurs que MA_d (les taux d'erreur s'étalent de 1,2% à 4,0%). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le taux d'erreur sont significatifs.

Temps — Effet significatif d'ACCORD sur le temps de réalisation ($F_{6,451} = 6,7$, $p < 0,0001$). Les analyses *post hoc* montrent : PI_d, IM_d, et PM_d sont significativement plus rapides que PA_d (tous les $p < 0,02$) ; IM_d et PM_d sont plus rapides que PA_d et PL_d (tous les $p < 0,01$). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le temps de réalisation sont significatifs.

Modélisation de la loi de Fitts — Pour tous les ACCORDS, les modélisations ont un coefficient de régression $r^2 \geq 0,97$. Les différents paramètres des modélisations sont résumés dans la table 2.4.

*throughput** — Effet significatif d'ACCORD sur le TP ($F_{6,451} = 6,5$, $p < 0,0001$). Les analyses *post hoc* montrent : PM_d a un TP plus élevé que AL_d, PA_d et PL_d ($p < 0,01$) ; PM_d, PI_d et MA_d ont un TP plus élevé que PL_d ($p < 0,05$).

Notes — Effet significatif d'ACCORD sur la facilité d'utilisation, la rapidité, la précision et la fatigue ($F_{9,99} > 4,6$, $p < 0,001$). Les analyses *post hoc* révèlent une tendance quasi similaire pour toutes les notations (tous les $p < .001$) : IM_d, MA_d, PI_d et PM_d ont des notes plus élevées que AL_d et PA_d (facilité d'utilisation, fatigue) et PL_d (rapidité) ; IM_d et PI_d ont des notes plus élevées que PL_d pour la précision.

2.1.5 Performance et préférence : déplacement d'objets

L'objectif de cette expérience est de mesurer la performance des différents DOIGTS et ACCORDS lors du déplacement d'objets. 12 participants ont été recrutés pour cette expérience (23,8 ans en moyenne, 3,5 ans de déviation standard, 6 femmes). Le dispositif, le protocole, le plan expérimental et l'analyse sont décrits dans la section 2.1.3.

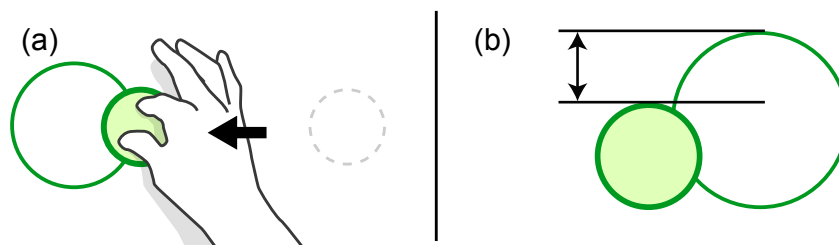


FIGURE 2.5: Tâche de déplacement d'objet. Les participants déplacent un objet circulaire (disque vert) vers une zone cible circulaire (cercle vert) en utilisant un CONTACT imposé : (a) exemple utilisant l'accord PI_d ; (b) TOLÉRANCE de la tâche.

2.1.5.1 Tâche

La tâche consistait à déplacer un objet circulaire dans une cible circulaire aussi rapidement que possible en utilisant un CONTACT imposé, tout en gardant un taux d'erreur d'environ 4%. Chaque ESSAI commençait au moment où le participant effectuait un *Touch Down** dans l'objet circulaire (1,75 cm de diamètre pour les DOIGTS, 3 cm de diamètre pour les ACCORDS). Sans lever son(s) doigt(s) de la surface, il déplaçait l'objet dans une zone circulaire dont la position et la géométrie étaient contrôlées (cf. Figure 2.5). Un ESSAI se terminait lorsque le participant relâchait l'objet dans la zone circulaire. Si une quelconque partie de l'objet se retrouvait en dehors de la zone cible lorsque celui-ci était relâché, l'ESSAI était considéré comme erroné et l'arrière plan de l'application devenait orange.

2.1.5.2 Plan expérimental

Dans cette tâche, la TOLÉRANCE correspond à la différence entre le diamètre de l'objet et celui de la zone cible (cf. Figure 2.5-b). L'AMPLITUDE correspond à la distance entre le centre de l'objet à sa position initiale et le centre de la zone cible. Des TOLÉRANCES et AMPLITUDES différentes ont été utilisées pour les DOIGTS et les ACCORDS. Pour les DOIGTS, les TOLÉRANCES étaient de 0,5, 1, et 1,5 cm et les AMPLITUDES étaient de 3 et 17 cm. Les six combinaisons de TOLÉRANCE et d'AMPLITUDE produisaient des *ID* s'étalant de 1,6 à 5,1 *bits**. Pour les ACCORDS, les TOLÉRANCES étaient de 0,5, 1, et 1,5 cm et les AMPLITUDES étaient de 4 et 15,5 cm. Les six combinaisons de TOLÉRANCE et d'AMPLITUDE produisaient des *ID* s'étalant de 1,9 à 5,0 *bits**. La direction entre la zone de départ et la zone cible variait avec les 6 RÉPÉTITIONS, à l'instar de l'expérience de pointage (cf. Section 2.1.4.2). Le plan expérimental était donc : 17 CONTACTS $\times$ 3 BLOCS $\times$ 3 TOLÉRANCES $\times$ 2 AMPLITUDES $\times$ 6 RÉPÉTITIONS par participant.

	L_n	A_n	M_n	I_n	P_n	P_d	I_d	M_d	A_d	L_d
a	0,46	0,45	0,43	0,55	0,51	0,40	0,39	0,39	0,41	0,38
b	0,35	0,32	0,30	0,28	0,34	0,32	0,25	0,27	0,27	0,31
r^2	0,86	0,85	0,85	0,80	0,80	0,87	0,84	0,81	0,83	0,86
Temps	1,14	1,08	1,01	1,11	1,17	1,03	0,88	0,93	0,94	1,00
TP	1,58	1,57	2,08	1,64	1,72	1,87	1,97	2,04	2,12	2,33
IP	2,89	3,15	3,35	3,53	2,96	3,14	3,96	3,69	3,72	3,18

TABLE 2.5: Résumé pour chaque DOIGT des différents paramètres des modélisations pour l'expérience de déplacement d'objet. Les valeurs extrêmes sont surlignées en vert (meilleures) et en gris (moins bonnes).

2.1.5.3 Résultats

Comme précédemment, nous rapportons séparément les analyses statistiques et les modélisations de la loi de *Fitts* pour DOIGTS et ACCORDS. Les résultats clefs sont résumés dans les tables 2.5 et 2.6 et les tendances pour le *throughput** et les préférences sont résumées dans les figures 2.8-de de la section 2.1.8.

Doigts *Apprentissage et/ou Fatigue* — Aucun effet significatif de BLOC sur le temps de réalisation ni sur le taux d'erreur, tous les BLOCS sont inclus dans les futures analyses.

Erreur — Aucun effet significatif de DOIGT sur le taux d'erreur. Les taux d'erreurs moyens s'étalent de 1,77% à 3,56%. Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le taux d'erreur sont significatifs.

Temps — Effet significatif de DOIGT sur le temps de réalisation ($F_{9,649} = 29,5$, $p < 0,0001$). Les analyses *post hoc* montrent : I_d et M_d sont plus rapides que L_d et tous les doigts de la main non-dominante (tous les $p < 0,0001$) ; P_n est plus lent que tous les autres doigts sauf L_n (tous les $p < 0,01$) ; P_n , L_n et A_n sont plus lents que tous les doigts de la main dominante (tous les $p < 0,0001$). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le temps de réalisation sont significatifs.

Modélisation de la loi de Fitts — Pour tous les DOIGTS, les modélisations ont un coefficient de régression r^2 entre 0,67 et 0,75. Les différents paramètres des modélisations sont résumés dans la table 2.5.

*throughput** — Effet significatif de DOIGT sur le *TP* ($F_{9,649} = 7,5$, $p < 0,0001$). Les analyses *post hoc* montrent : M_n et tous les doigts de la main dominante ont un *TP* plus élevé que P_n ($p < 0,05$) ; I_d , M_d et A_d ont un *TP* plus élevé que A_n , L_n et P_n ($p < 0,05$).

Notes — Effet significatif de DOIGT sur la facilité d'utilisation, la rapidité, la précision et la fatigue ($F_{9,99} > 4,0$, $p < 0,001$). Les analyses *post hoc* révèlent une tendance similaire pour toutes les notations (tous les $p < 0,001$) : I_d , M_d et I_n ont des notes plus élevées que L_d , A_n et L_n (facilité d'utilisation, rapidité) et A_n et L_d (fatigue). I_d et M_d ont des notes plus élevées que A_n and L_n pour la précision.

	PI _d	PM _d	PA _d	PL _d	IM _d	MA _d	AL _d
a	0,55	0,59	0,48	0,46	0,39	0,50	0,60
b	0,35	0,39	0,44	0,50	0,59	0,45	0,48
r^2	0,77	0,78	0,83	0,85	0,90	0,82	0,85
Temps	1,12	1,21	1,20	1,27	1,35	1,23	1,38
TP	2,17	1,84	1,86	1,70	1,68	1,91	1,63
IP	2,85	2,59	2,27	2,01	1,70	2,24	2,09

TABLE 2.6: Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de déplacement d'objet. Les valeurs extrêmes sont surlignées en vert (meilleures) et en gris (moins bonnes).

Accords *Apprentissage et/ou Fatigue* — Aucun effet significatif de BLOC sur le temps de réalisation ni sur le taux d'erreur, tous les BLOCS sont inclus dans les analyses ci-dessous.

Erreur — Effet significatif d'ACCORD sur le taux d'erreur ($F_{6,451} = 5,0$, $p < 0,0001$). Les analyses *post hoc* montrent que PL_d cause plus d'erreurs que MA_d ou PI_d (tous les $p < 0,001$) ; PL_d et AL_d causent plus d'erreurs que PI_d (respectivement $p < 0,001$ et $p < 0,05$). Les taux d'erreurs moyens s'étalent de 2,48% à 7,87%. Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le taux d'erreur sont significatifs.

Temps — Effet significatif d'ACCORD sur le temps de réalisation ($F_{6,451} = 7,3$, $p < 0,0001$). Les analyses *post hoc* montrent : PA_d, PM_d et PI_d sont plus rapides que AL_d et IM_d (tous les $p < 0,05$) ; PL_d est plus lent que PI_d ($p < 0,05$). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le temps de réalisation sont significatifs.

Modélisation de la loi de Fitts — Pour tous les ACCORDS, les modélisations ont un coefficient de régression r^2 entre 0,59 et 0,81. Les différents paramètres des modélisations sont résumés dans la table 2.6.

*throughput** — Effet significatif d'ACCORD sur le TP ($F_{6,451} = 5,0$, $p < 0,0001$). Les analyses *post hoc* montrent : PI_d et MA_d ont un TP plus élevé que PL_d ($p < 0,05$) ; PI_d a un TP plus élevé que PM_d, IM_d, AL_d et PL_d ($p < 0,05$).

Notes — Effet significatif d'ACCORD sur la facilité d'utilisation, la rapidité, la précision et la fatigue ($F_{6,66} > 5,2$, $p < 0,001$). Les analyses *post hoc* révèlent une tendance quasi similaire pour toutes les notations (tous les $p < 0,001$) : PI_d, IM_d et MA_d ont des notes plus élevées que AL_d, PL_d et PA_d (facilité d'utilisation, fatigue), AL_d et PL_d (rapidité) et PL_d (précision).

2.1.6 Performance et préférence : changement d'échelle

L'objectif de cette expérience est de mesurer la performance des différents ACCORDS lors du changement d'échelle d'objet *via* manipulation directe. 12 participants ont été recrutés pour cette expérience (23,6 ans en moyenne, 2,5 ans de déviation standard, 7 femmes). Le dispositif, le protocole, le plan expérimental et l'analyse sont décrits dans la section 2.1.3.

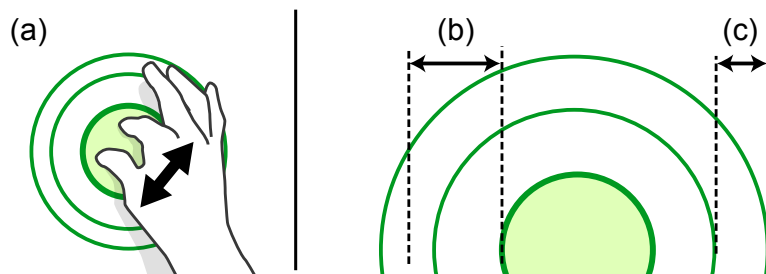


FIGURE 2.6: Tâche de changement d'échelle. Les ACCORDS *pinces* sont utilisés pour ajuster la taille d'un objet circulaire (ici en vert) jusqu'à ce que son bord soit contenu dans une zone cible en forme d'anneau : (a) exemple utilisant l'accord PI_d ; (b) demi-AMPLITUDE ; (c) TOLÉRANCE.

2.1.6.1 Tâche

La tâche consistait à ajuster la taille d'un objet circulaire jusqu'à ce que son bord soit contenu dans une zone cible en forme d'anneau (*cf.* Figure 2.6). Les participants recevaient la consigne d'être aussi rapide que possible en utilisant un ACCORD imposé et tout en gardant un taux d'erreur d'environ 4%. Chaque ESSAI commençait au moment où les deux doigts de l'ACCORD étaient en contact avec l'objet circulaire. La position de l'objet était fixe et toujours la même. Son diamètre était de 4, 7 ou 10 cm en fonction de la direction du changement d'échelle et de la TOLÉRANCE. Les deux doigts de l'ACCORD s'écartaient ou se rapprochaient sans être levés de l'écran, jusqu'à ce que le bord soit contenu dans une zone cible en forme d'anneau (*cf.* Figure 2.6). Un ESSAI se terminait lorsque le participant relâchait l'objet. Si le bord de l'objet se retrouvait en dehors de la zone cible lorsque celui-ci était relâché, l'ESSAI était considéré comme erroné et l'arrière plan de l'application devenait orange.

2.1.6.2 Plan expérimental

Dans cette tâche, la TOLÉRANCE correspond à l'épaisseur de l'anneau cible (*cf.* Figure 2.6-c). L'AMPLITUDE correspond à la distance moyenne à parcourir par les deux doigts : deux fois la distance entre la bordure de l'objet et le cercle moyen contenu dans l'anneau (*cf.* Figure 2.6-b). Seuls les ACCORDS *pinces* ont été évalués. Les TOLÉRANCES étaient de 1, 1,5, et 2 cm et les AMPLITUDES étaient de 3 et 6 cm.

Nous utilisons la formulation de Shannon pour calculer l' ID de la tâche de changement d'échelle plutôt que celle proposée par Zhao *et al.* [ZSB15]. En effet, le comportement du changement d'échelle est caractérisé par la distance physique parcourue par chacun des doigts : dans notre implémentation, les doigts restent en permanence à la même distance de la bordure de l'objet. Zhao *et al.* ont utilisé, en tant que facteur d'échelle, le ratio de la distance entre les doigts de deux positions consécutives. Cette implémentation induit un grossissement (ou rétrécissement)

exponentiel de l'objet expliquant les termes logarithmiques de leur formulation de l'*ID*. Notre approche permet d'étudier le mouvement des accords dans l'espace moteur et notre implémentation du changement d'échelle correspond à celle de travaux précédents comme [MH06]. Les six combinaisons de TOLÉRANCE et d'AMPLITUDE produisaient des *ID* s'étalant de 1,3 à 2,8 *bits*^{*}. La direction du changement d'échelle variait avec les 6 RÉPÉTITIONS : agrandissement et rétrécissement. Le plan expérimental était donc : 4 CONTACTS $\times$ 3 BLOCS $\times$ 3 TOLÉRANCES $\times$ 2 AMPLITUDES $\times$ 6 RÉPÉTITIONS par participant.

2.1.6.3 Résultats

Comme précédemment, nous rapportons les analyses statistiques et les modélisations de la loi de *Fitts*. Les résultats clefs sont résumés dans la table 2.7 et les tendances pour le *throughput*^{*} et les préférences sont résumées dans la figure 2.8-c de la section 2.1.8.

Apprentissage et/ou Fatigue — Effet significatif de BLOC sur le temps de réalisation ($F_{2,22} = 12,2$, $p < 0,001$) mais pas sur le taux d'erreur. Le BLOC 1 est significativement plus lent que les 2 autres BLOCS. Nous enlevons le BLOC 1 pour le reste de nos analyses.

Erreur — Aucun effet significatif d'ACCORD sur le taux d'erreur. Les taux d'erreurs moyens s'étalent de 3,77% à 4,13%. Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le taux d'erreur sont significatifs.

Temps — Effet significatif d'ACCORD sur le temps de réalisation ($F_{3,429} = 4,2$, $p < 0,01$). Les analyses *post hoc* montrent : PA_d est plus lent que tous les autres accords sauf PL_d (tous les $p < 0,05$). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le temps de réalisation sont significatifs.

Modélisation de la loi de Fitts — Pour tous les ACCORDS, les modélisations ont un coefficient de régression $r^2 \geq 0,98$. Les différents paramètres des modélisations sont résumés dans la table 2.7.

throughput^{*} — Aucun effet significatif d'ACCORD sur le *TP*.

Notes — Effet significatif d'ACCORD sur la facilité d'utilisation, la rapidité, la précision et la fatigue ($F_{3,33} > 9,7$, $p < 0,0001$). Les analyses *post hoc* révèlent une tendance similaire pour toutes les notations (tous les $p < 0,0001$) : PI_d et PM_d ont des notes plus élevées que PA_d et PL_d (facilité d'utilisation, rapidité) et PL_d (précision) ; PI_d a des notes plus élevées que tous les autres accords pour la fatigue.

2.1.7 Performance et préférence : rotation d'objet

L'objectif de cette expérience est de mesurer la performance des différents ACCORDS lors de rotation d'objet *via* manipulation directe. Les mêmes 12 participants de l'expérience de changement d'échelle ont été recrutés pour celle-ci (23,6 ans en moyenne, 2,5 ans de déviation standard, 7 femmes). L'ordre des deux expériences était contrebalancé entre les participants. Le dispositif, le protocole, le plan expérimental et l'analyse sont décrits dans la section 2.1.3.

	PI _d	PM _d	PA _d	PL _d
a	0,30	0,30	0,15	0,26
b	0,42	0,43	0,53	0,46
r^2	1,00	1,00	0,99	0,99
Temps	1,15	1,17	1,22	1,19
TP	0,81	1,05	0,86	0,83
IP	2,37	2,32	1,88	2,17

TABLE 2.7: Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de changement d'échelle. Les valeurs extrêmes sont surlignées en vert (meilleures) et en gris (moins bonnes).

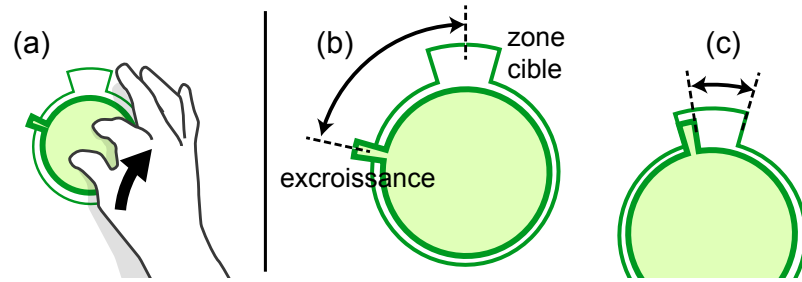


FIGURE 2.7: Tâche de rotation. Les ACCORDS sont utilisés pour faire pivoter un objet circulaire (*disque vert*) jusqu'à ce qu'une excroissance de l'objet soit contenue dans une zone cible (*cercle vert*) : (a) exemple utilisant l'accord PI_d ; (b) AMPLITUDE ; (c) TOLÉRANCE.

2.1.7.1 Tâche

La tâche consistait à faire pivoter un objet circulaire jusqu'à ce qu'il soit aligné avec une cible. Les participants recevaient la consigne d'être aussi rapide que possible en utilisant un ACCORD imposé et tout en gardant un taux d'erreur d'environ 4%. Chaque ESSAI commençait au moment où les deux doigts de l'ACCORD étaient en contact avec l'objet circulaire. La position de l'objet était fixe et toujours la même, tout comme son diamètre (8 cm). L'objet pivotait suivant la rotation effectuée par les doigts jusqu'à ce qu'une excroissance de l'objet soit contenue dans une zone cible (*cf.* Figure 2.7). Un ESSAI se terminait lorsque le participant relâchait l'objet. Si une quelconque partie de l'excroissance se retrouvait en dehors de la zone cible lorsque l'objet était relâché, l'ESSAI était considéré comme erroné et l'arrière plan de l'application devenait orange.

2.1.7.2 Plan expérimental

Dans cette tâche, la TOLÉRANCE correspond à l'ouverture en degré de la zone cible moins l'ouverture de l'excroissance qui est de 5° (*cf.* Figure 2.7-c). L'AMPLITUDE correspond à l'angle entre la bissectrice de l'excroissance à sa position initiale et la bissectrice de la zone cible (*cf.* Figure 2.7-b).

Les 7 ACCORDS ont été évalués. Les TOLÉRANCES étaient de 5, 10 et 15° et les AMPLITUDES étaient de 25 et 90°. Les six combinaisons de TOLÉRANCE et d'AMPLITUDE produisaient des *ID* s'étalant de 1,4 à 4,2 *bits**. Le sens de la rotation variait avec les 6 RÉPÉTITIONS : sens horaire et sens anti-horaire. Le plan expérimental était donc : 7 CONTACTS $\times$ 3 BLOCS $\times$ 3 TOLÉRANCES $\times$ 2 AMPLITUDES $\times$ 6 RÉPÉTITIONS par participant.

2.1.7.3 Résultats

Comme précédemment, nous rapportons les analyses statistiques et les modélisations de la loi de *Fitts*. Les résultats clés sont résumés dans la table 2.8 et les tendances pour le *throughput** et les préférences sont résumées dans la figure 2.8-f de la section 2.1.8.

Apprentissage et/ou Fatigue — Effet significatif de BLOC sur le temps de réalisation ($F_{2,22} = 4,5$, $p < 0,05$) mais pas sur le taux d'erreur. Le BLOC 1 est significativement plus lent que le BLOC 3. Les analyses *post hoc* ont révélé que le BLOC 1 ne pouvait être enlevé sans aussi enlever le BLOC 2 puis le BLOC 3. Par conséquent, tous les BLOCS sont inclus dans les futures analyses

Erreur — Effet significatif d'ACCORD sur le taux d'erreur ($F_{6,451} = 6,7$, $p < 0,0001$). Les analyses *post hoc* montrent : MA_d a un taux d'erreur moins élevé que AL_d et PL_d (respectivement $p < 0,0001$ et $p < 0,05$) ; AL_d a un taux d'erreur plus élevé que tous les autres accords sauf PL_d (tous les $p < 0,01$). Les taux d'erreurs moyens s'évaluent de 2,29% à 7,54%. Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur l'Erreur sont significatifs.

Temps — Effet significatif d'ACCORD sur le temps de réalisation ($F_{6,451} = 26,0$, $p < 0,0001$). Les analyses *post hoc* montrent : PM_d est plus rapide que tous les autres accords sauf PA_d et PI_d (tous les $p < 0,05$) ; AL_d est plus lent que tous les autres accords (tous les $p < 0,0001$). Sans surprise, les effets de TOLÉRANCE et d'AMPLITUDE sur le temps de réalisation sont significatifs.

Modélisation de la loi de Fitts — Pour tous les ACCORDS, les modélisations ont un coefficient de régression r^2 entre 0,97 et 0,98. Les différents paramètres des modélisations sont résumés dans la table 2.8.

*throughput** — Effet significatif d'ACCORD sur le *TP* ($F_{6,451} = 21,0$, $p < 0,0001$). Les analyses *post hoc* montrent : AL_d a un *TP* moins élevé que tous les autres accords ($p < 0,001$) ; PM_d, PI_d et PA_d ont un *TP* plus élevé que PL_d et AL_d ($p < 0,05$).

Notes — Effet significatif d'ACCORD sur la facilité d'utilisation, la rapidité, la précision et la fatigue ($F_{6,66} > 8,5$, $p < 0,0001$). Les analyses *post hoc* révèlent une tendance similaire pour toutes les notations (tous les $p < 0,0001$) : PI_d, PM_d et IM_d ont des notes plus élevées que PL_d et AL_d pour la facilité d'utilisation, la rapidité et la précision ; PI_d, IM_d et PA_d ont des notes plus élevées que PL_d et AL_d pour la fatigue.

	PI _d	PM _d	PA _d	PL _d	IM _d	MA _d	AL _d
a	0,31	0,22	0,15	0,23	0,18	0,21	0,29
b	0,38	0,39	0,44	0,43	0,46	0,46	0,53
r^2	0,99	0,99	0,99	1,00	0,98	0,99	0,99
Temps	1,33	1,27	1,34	1,39	1,41	1,45	1,72
TP	1,56	1,86	1,53	1,36	1,64	1,69	1,03
IP	2,63	2,55	2,25	2,32	2,18	2,17	1,89

TABLE 2.8: Résumé pour chaque ACCORD des différents paramètres des modélisations pour l'expérience de rotation. Les valeurs extrêmes sont surlignées en vert (meilleures) et en gris (moins bonnes).

	C_{pref}	C_{pref}
Pointage	DOIGTS ACCORDS	I _n , P _d , I _d , M _d PI _d , PM _d , IM _d , MA _d
Déplacement d'objet	DOIGTS ACCORDS	I _n , P _d , I _d , M _d PI _d , IM _d , MA _d
Changement d'échelle	ACCORDS	PI _d
Rotation	ACCORDS	PI _d , PM _d , IM _d

TABLE 2.9: Recommandations des CONTACTS à privilégier (C_{pref}) et à éviter (C_{pref}).

2.1.8 Discussion et recommandations de conception

En s'appuyant sur nos résultats, nous proposons une série de recommandations pour la conception de techniques basées sur l'identification des doigts incorporant les actions atomiques que sont : le pointage, le déplacement d'objets, le changement d'échelle et la rotation. Ces recommandations tirent partie de l'analyse des performances et des préférences de chacune des tâches.

Nous résumons nos résultats par tâche et par CONTACT au travers de trois groupes de niveau : C_{pref} pour les doigts ou accords à privilégier, C_{pref} pour ceux à éviter et C_{neutre} pour les contacts restants. Pour chaque CONTACT et chaque tâche, nous avons calculé un score basé sur les différences *post hoc* trouvées sur les performances ainsi que les préférences. Chaque CONTACT commençait avec un score initial de 0. Nous ajoutons 1 lorsque celui-ci était significativement meilleur que d'autres CONTACTS et enlevions 1 si à l'inverse s'il était significativement moins bon. Les groupes de niveaux sont basés sur les similitudes produites. Ils sont résumés dans la table 2.9 et dans la figure 2.8.

2.1.8.1 Exemples d'applications

Avec *Adoiraccourcix* (cf. Section 3.2), nous nous basons sur un ensemble d'accords utilisant l'identification des doigts pour activer des quasi-modes et un ensemble de doigts pour déclencher les commandes. L'activation de ces quasi-modes ne nécessite pas d'être effectuée à une position particulière de

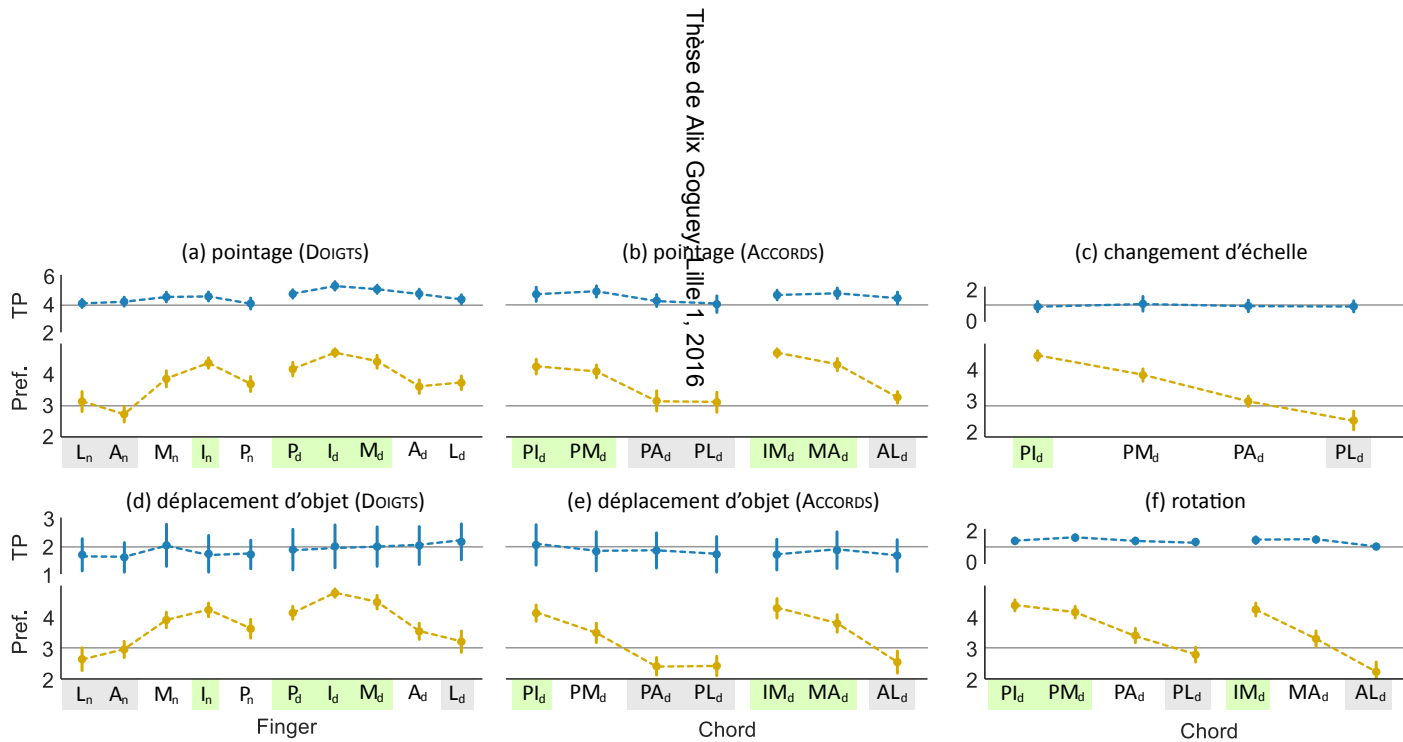


FIGURE 2.8: $throughput^*$ des différentes tâches (en bps^*) et notes moyennes des préférences (sur 5). Les barres d'erreur indiquent les intervalles de confiance à 95%. Les contacts du groupe C_{pref} sont indiqués en vert et ceux de $C_{nonpref}$ en gris.

l'écran et est temporaire (*i.e.* une fois la commande déclenchée, le doigt ou l'accord activant le quasi-mode peut être retiré). En fonction du facteur de forme, cette activation s'effectue en appuyant un doigt/accord sur la surface (*i.e.* pointage sans précision) ou *via* un *flick** (*i.e.* glissé court dans une direction particulière). En suivant donc les recommandations de cette étude, la première façon de faire pourrait s'effectuer à l'aide de n'importe quel CONTACT. Pour la seconde, à l'aide des CONTACT C_{neutre} ou C_{pref} (lorsque qu'il y a beaucoup de directions possibles) ou n'importe quel CONTACT (lorsque qu'il y a peu de directions possibles). Le déclenchement de commandes pourrait lui utiliser les doigts/accords C_{pref} lorsqu'une position précise est requise (*ex.* *copier* ou *coller*) ou les CONTACTS C_{neutre} lorsque peu de précision est requise (*ex.* *sélection de paragraphe*). Les doigts/accords C_{pref} pourraient être associés aux commandes non contextuelles, comme *sauvegarder*, ou à des quasi-modes contraignant les actions des CONTACTS C_{pref} comme le magnétisme lors de dessins ou la discrétisation lors de rotations.

Marquardt *et al.* associent eux différents outils de dessin aux différents doigts et accords [MKL⁺11]. Nos recommandations seraient que les outils nécessitant de la précision (*ex.* *stylo*, *création de forme*, *positionnement de formes*) soient associés au groupe C_{pref} et que les outils nécessitant une précision grossière (*ex.* *création de zone de texte* ou des actions de *pan**) soient associés au groupe C_{neutre} . Le groupe C_{pref} serait approprié pour les mouvements avec très peu de précision comme des *flicks** (comme suggéré par Lepinski *et al.* [LGF10]). Par exemple pour l'utilisation de *flicks** gauches/droites pour naviguer entre différents calques d'un dessin.

Dans une application de manipulation de contenu, la transformation d'objets (*i.e.* changement d'échelle et rotation d'un objet) et le zoom sont la plupart du temps effectués au cours d'un seul et même geste. Puisque que le changement d'échelle et la rotation nécessitent un contrôle plus fin, il serait préférable de les associer à des accords de C_{pref} , voire même deux accords différents pour donner plus de possibilités à l'utilisateur. Le contrôle du niveau de zoom et la rotation d'une vue (*ex.* navigation dans *Google Map*) pourraient être associés à des accords du groupe C_{neutre} puisque moins de précision est requise.

2.1.8.2 Recommandations fines

Dans les exemples précédents, nous avons choisi de regrouper les CONTACTS en se basant à la fois sur les performances et sur les préférences. Mais il est aussi possible de discuter les associations avec un niveau de granularité plus fin. En reprenant le système de score décrit précédemment et en calculant un score pour les performances et un autre pour les préférences, nous proposons des groupes de niveaux dans un espace à deux dimensions (*cf.* Table 2.10).

En se basant sur ce nouvel espace, nous proposons : l'utilisation des CONTACTS efficaces pour des commandes localisées (*i.e.* nécessitant une certaine précision pour être déclenchées) ; l'utilisation des CONTACTS moins efficaces pour des commandes générales (*i.e.* ne nécessitant pas d'être déclenchées à une position particulière de l'écran) ; l'utilisation des CONTACTS appréciés pour des com-

Performances		C_{pref}	C_{pref}
Pointage	DOIGTS ACCORDS	P_d, I_d, M_d PI_d, PM_d, IM_d	L_n, A_n, P_n PA_d, PL_d
Déplacement d'objet	DOIGTS ACCORDS	P_d, I_d, M_d PI_d	L_n, A_n, P_n PL_d, IM_d, AL_d
Changement d'échelle	ACCORDS	PI_d, PM_d	PA_d
Rotation	ACCORDS	PI_d, PM_d, PA_d, MA_d	AL_d
Préférences		C_{pref}	C_{pref}
Pointage	DOIGTS ACCORDS	I_n, P_d, I_d, M_d PI_d, PM_d, IM_d, MA_d	L_n, A_n PA_d, PL_d, AL_d
Déplacement d'objet	DOIGTS ACCORDS	I_n, I_d, M_d PI_d, IM_d, MA_d	L_n, A_n, L_d PA_d, PL_d, AL_d
Changement d'échelle	ACCORDS	PI_d	PA_d, PL_d
Rotation	ACCORDS	PI_d, PM_d, IM_d	PL_d, AL_d

TABLE 2.10: Recommandations des CONTACTS à privilégier (C_{pref}) et à éviter (C_{pref}) en fonctions des performances ou des préférences.

mandes non-critiques ; l'utilisation des CONTACTS les moins appréciés pour des commandes critiques, voire irréversible, afin d'éviter quelles soient déclenchées par inadvertance. En suivant ces règles, nous pourrions recommander d'associer (cf. Figure 2.9) : la commande *dessiner* aux CONTACTS efficaces (ex. commande localisée) et appréciés (ex. commande non-critique) ; la commande *sauvegarder* aux CONTACTS moins efficaces (ex. commande générale) et appréciés (ex. commande non-critique) ; la commande *supprimer* aux CONTACTS efficaces (ex. commande localisée) et moins appréciés (ex. commande critique) ; la commande *fermer* aux CONTACTS moins efficaces (ex. commande générale) et moins appréciés (ex. commande critique).

2.1.9 Conclusion et travaux futurs

Afin de mieux concevoir les techniques d'interaction basées sur l'identification des doigts, notre étude vise à mieux comprendre les différences de performance et de préférence entre les doigts et les accords. Au travers de cinq expériences de *Fitts*, reprenant chacune une action classique des interactions tactiles, nous avons comparé les dix doigts ainsi que sept accords à deux doigts. Nous proposons des recommandations pour la conception de ces techniques en nous basant sur les résultats des différentes expériences. Ces recommandations sont résumées dans les tables 2.9 et 2.10.

Afin de réduire les temps des phases de recrutement et des sessions, certains choix de notre plan expérimental souffrent de quelques limitations. Tout d'abord, les participants recrutés pour notre étude sont principalement des

	contacts efficaces	contacts neutres	contacts moins efficaces	
contacts appréciés	 dessiner		 sauvegarder	commandes non-critiques
contacts neutres				
contacts moins appréciés	 supprimer		 fermer	commandes critiques
	commandes localisées		commandes générales	

FIGURE 2.9: Exemples d'associations entre commandes et accords basés sur les scores de performances et de préférences.

étudiants d'université. Cette population relativement jeune n'est pas forcément représentative de l'ensemble des utilisateurs d'écrans tactiles. De même, nous n'avons pas voulu faire varier la position de la tablette. Diversifier ces deux facteurs pourrait permettre d'élargir la validité de nos résultats empiriques. De plus, des travaux complémentaires sont nécessaires pour compléter ces résultats. Notre étude ne compare que sept accords à deux doigts, mais le reste des accords à deux doigts de la main dominante et de la main non-dominante restent à être évalués. Enfin, d'autres tâches pourraient être explorées, comme l'exécution de *flicks** dans différentes directions à la manière de [LGF10].

Dans la partie précédente, nous nous sommes intéressés à l'aspect moteur de l'identification des doigts. Nous nous intéressons maintenant à deux aspects de la cognition : le raisonnement et le (ré-)apprentissage. Le but de ces études est d'identifier des facteurs qui peuvent influencer sur l'appréhension de cette nouvelle information par l'utilisateur.

2.2 Cognition

À chaque technique d'interaction est associé un *coût cognitif*. Elle peut par exemple nécessiter de l'utilisateur qu'il retienne un grand nombre de règles, de commandes, *etc.*, auquel cas le coût cognitif sera élevé, due à la mémorisation, mais peut diminuer avec l'expertise de l'utilisateur. Dans la même veine, une technique peut influencer la stratégie, et donc le coût cognitif, qu'un utilisateur a pour résoudre une tâche : si changer d'outils est coûteux en temps, l'utilisateur essaiera d'optimiser l'ordre de ses actions pour minimiser les changements d'outils. Le *coût cognitif* peut être influencé par une multitude de facteurs. L'identification des doigts est une information relativement nouvelle et aucune étude n'a pour l'instant été effectuée sur l'aspect cognitif de son utilisation (les études portant surtout sur l'expressivité [RGB⁺15]). Nous souhaitons donc au travers de deux études apporter des briques à cet édifice en construction

Pour comparer différentes techniques entre elles ou comprendre ce qui fait qu'une technique est plus rapide qu'une autre dans un contexte donné, la comparaison en terme de temps n'est pas forcément la métrique la plus indiquée (*ex.* le temps peut dépendre de la fiabilité du prototype). L'étude des stratégies suivies par les utilisateurs pour résoudre une tâche offre une alternative à l'étude du temps. Mackay [Mac02] a par exemple mené une comparaison entre palette d'outils, *marking menus* [KB91] et *Toolglass* [BSP⁺93] au travers de deux tâches consistant à copier ou modifier des réseaux de *Petri*. Ses recherches montrent que la technique d'interaction optimisant les performances varie en fonction de la tâche, du contexte cognitif de l'utilisateur et de ses préférences. Elle a de plus observé que la palette d'outils et les *marking menus* favorisent des actions *outil par outil* (*ex.* création de tous les *triangles* en premier, puis de tous les *cercles*) alors que la *Toolglass* favorise un changement fréquent d'outil. Ces enchaînements d'actions sont appelés *stratégies*. Nous proposons des métriques qui aident à catégoriser les techniques d'interaction en annotant automatiquement

et objectivement les stratégies utilisateurs. Puis à l'aide de ces métriques nous étudions l'influence de techniques basées sur l'identification des doigts sur les stratégies utilisateurs.

Le coût mémoire est un autre aspect important dans l'étude de nouvelles techniques d'interaction, surtout lorsque celles-ci cohabitent avec des interactions déjà existantes. Dans le cas des interactions basées sur l'identification des doigts, un accord est associé à une commande, or le nombre de doigts en contact simultanés sur une surface est aussi un déclencheur de commande. Les utilisateurs sont déjà habitués à de nombreuses interactions tactiles bien ancrées dans les systèmes commerciaux (*ex. pinch-and-expand**, défilement, *etc.*). Introduire de l'identification des doigts signifierait restreindre l'utilisation de ces gestes à certains accords et prendre le risque de perturber l'utilisateur. À l'inverse, réserver un nombre trop important d'accords pour ces interactions communes signifierait réduire le potentiel de l'identification des doigts. Nous nous intéressons donc à un ré-apprentissage éventuel en étudiant quels accords sont utilisés pour les interactions communes : par exemple, le *pinch-and-expand** est-il très majoritairement déclenché par un accord pouce + index ? Auquel cas, les autres accords à deux doigts impliquant le pouce seraient libres d'être associés à de nouvelles commandes sans que l'utilisateur ait d'effort de ré-apprentissage particulier à faire.

Dans un premier temps, nous proposerons un état de l'art sur les différentes méthodes utilisées pour qualifier les stratégies utilisateurs. Nous présenterons ensuite deux métriques qui permettent la quantification automatique et objective de stratégies. Nous apporterons une validation de ces métriques au travers de la comparaison de deux techniques de la littérature (une *Palette d'Outils Fixe* [ABLM05] et une *Toolglass* [BSP⁺93]). Nous nous proposerons aussi d'étudier deux nouvelles techniques basées sur l'identification des doigts (une *Palette d'Outils Fixe avec Identification des Doigts* et une *Palette Flottante avec Identification des Doigts*). Puis nous discuterons les raisons cognitives des différences de stratégies entre toutes ces techniques. Dans un second temps, nous reportons une étude de terrain menée afin d'observer les différents accords utilisés par les utilisateurs pour effectuer des gestes tactiles usuels. Enfin nous discuterons l'enjeu cognitif qui découle d'une intégration de techniques basées sur l'identification des doigts.

2.2.1 État de l'art

Dans cet état de l'art, nous nous intéressons aux différentes études qui abordent l'influence des techniques d'interaction sur les stratégies et l'analyse des habitudes utilisateurs. Contrairement aux sections 1.1 et 2.1.1 qui abordaient respectivement l'aspect technologique de la modalité tactile et l'aspect dextérité du facteur humain, nous nous concentrons ici sur un l'aspect cognitif du facteur humain.

Pour résoudre une tâche, un utilisateur accomplit une séquence d'actions

canoniques. Appert *et al.* [ABLM05] et Mackay [Mac02] définissent une stratégie comme étant l'ordre de ces actions canoniques. La stratégie varie en fonction de différents facteurs comme le modèle mental de la tâche qu'a un utilisateur ou encore la technique utilisée. *Quantifier* une stratégie n'est pas chose aisée. Une stratégie peut être une combinaison de stratégies *centrées-objets* (*i.e.* effectuer toutes les actions nécessaires à la réalisation d'une entité avant de passer à l'entité suivante) et *centrées-outils* (*i.e.* effectuer toutes les actions canoniques utilisant un outil particulier avant de passer à l'outil suivant). Par manque de technique d'annotation, les chercheurs *qualifient* les séquences d'actions. Ce choix étant discret, il ne reflète pas forcément la vraie nature d'une stratégie : une stratégie n'est pas nécessairement purement centrée-objet ni centrée-outils.

Dans cet état de l'art nous présentons les différentes méthodes qui ont été utilisées pour qualifier les séquences d'actions d'un utilisateur.

2.2.1.1 Quantification de stratégies

Dans [ABLM05] et [Mac02], Appert *et al.* et Mackay étudiaient différentes techniques d'interaction dans différents contextes et identifiaient quel type de stratégie était le plus adapté dans chacun des cas. Avec le modèle *Complexity of Interaction Sequences* (CIS), Appert *et al.* analysent la structure d'une technique d'interaction et prédisent sa performance en terme de temps pour une stratégie donnée. Une stratégie doit donc être déterminée en amont. Mackay n'imposait pas de stratégie. À la place elle observait et annotait les séquences d'interaction. L'annotation est une tâche fastidieuse, subjective et sujette à des erreurs en considérant le fait qu'une séquence d'actions appartient rarement à une catégorie ou une autre. Les résultats de Appert *et al.* et de Mackay se recoupent : les palettes d'outils fixes sont orientées-outil alors que les *marking menus* [KB91] et les *Toolglass* [BSP⁺93] sont orientées-objet.

Mackay [Mac02] a aussi mesuré le nombre moyen d'actions identiques que nous effectuons avant de changer d'outil : un score élevé indique l'utilisation d'une stratégie orienté-outil, à l'inverse un score bas indique l'utilisation d'une stratégie orientée-objet. En plus d'être un choix subjectif, changer souvent d'outil n'indique pas forcément l'utilisation d'une stratégie orientée-objet (*ex.* dessiner un cercle, puis un triangle, puis colorer le cercle en bleu et finalement le triangle en rouge n'est ni une stratégie orienté-objet ni orientée-outil). Les stratégies *orientées-objet* et *orientées-outil* sont orthogonales entre elles. Les métriques que nous présentons ont pour but de mesurer automatiquement le degré auquel une séquence d'interaction est orientée-objet et orientée-outil. De plus, nos métriques nous permettent d'être plus écologique puisqu'aucune stratégie n'est imposée à l'utilisateur.

Bhavnani *et al.* [BJ98, BJ00] ont étudié les stratégies plus haut niveau (*i.e.* les stratégies qui différencient les utilisateurs novices des experts) et comment les utilisateurs gagnent de l'expertise. Ils défendent que les utilisateurs doivent apprendre les stratégies : la connaissance d'une tâche et d'un outil ne suffisent pas pour rendre un utilisateur plus efficace pour des applications complexes sur ordinateur. Il faut donc un moyen de montrer/d'apprendre à l'utilisateur de

nouvelles stratégies. *Skillometer* [MSC⁺13] est un exemple de système qui aide les utilisateurs à apprendre des stratégies plus efficaces en utilisant les raccourcis claviers à la place de menus chronophages. Nos métriques sont destinées à mesurer des stratégies comme celles étudiées par Appert, Beaudouin-Lafon et Mackay.

Avant d'entrer dans la description de nos métriques et dans l'étude de l'influence de l'identification des doigts sur la stratégie utilisateur, nous souhaitons aborder différents travaux liés à notre étude sur l'observation des accords. À notre connaissance il n'existe pas de travail similaire, nous reportons donc les études qui ont effectué dans le même contexte d'interaction, à savoir sur pavé tactile.

2.2.1.2 L'utilisation des accords

Notre étude observant les différents accords utilisés par les utilisateurs pour effectuer des gestes tactiles usuels, a été effectuée sur pavé tactile d'ordinateur *Macbook Pro* de la marque *Apple*. Bien que le pavé tactile soit un dispositif à la fois répandu et proposant maintenant un vocabulaire d'entrée riche, nous ne savons toujours pas comment les utilisateurs interagissent avec, ni quels gestes multi-points ils préfèrent utiliser. En effet, les pavés tactiles sont encore trop souvent considérés dans la littérature comme une alternative aux dispositifs de pointage et comparés soit en terme de performance [AZ99, HH10], ou en terme de préférences utilisateurs [Kot14]. Entre-temps, aucune étude de terrain n'a été effectuée pour analyser l'utilisation des pavés tactiles. Plusieurs études ont été menées pour examiner les mouvements du pointeur de souris en situation réelle, réalisés à l'aide d'une souris ou d'un pavé tactile [AHJ13, CBBL07, KSR06, MCPR13], mais sans analyse particulière des données brutes des périphériques de pointage. Enfin, l'utilisation d'accords [BML12] et la performance et l'ergonomie des gestes sur grandes surfaces tactiles (*cf.* Section 2.1) ont été étudiés, mais aucune information n'existe à ce jour sur l'utilisation des doigts et des mains sur pavés tactiles.

2.2.2 Stratégie

Les travaux décrits dans cette section ont été publiés à la conférence INTER-ACT 2015 [GWC15].

Avec une technique d'interaction donnée, les utilisateurs peuvent optimiser leur efficacité et effectuer une tâche composée en utilisant des stratégies qui varient entre purement *orientée-outil* et purement *orientée-objet*. Une stratégie (S) peut se décomposer en n actions élémentaires (a_i) effectuées sur des objets interactifs. $Obj(a_i)$ est l'objet modifié durant l'action a_i . Par exemple, dessiner deux rectangles bleus se compose des actions *créer un rectangle* (c_{rect}) et *peindre en bleu* (p_{bleu}), effectuées sur deux objets rectangles R_1 et R_2 . Avec une stratégie *orientée-outil*, les utilisateurs évitent de changer d'outil ce qui donne la séquence d'action suivante : $(c_{rect})_{R_1}(c_{rect})_{R_2}(p_{bleu})_{R_1}(p_{bleu})_{R_2}$. Avec une stratégie *orientée-objet*,

les utilisateurs préfèrent terminer un objet avant de passer au suivant ce qui donne la séquence d'action suivante : $(c_{rect})_{R_1}(p_{bleu})_{R_1}(c_{rect})_{R_2}(p_{bleu})_{R_2}$.

2.2.2.1 Métrique pour quantifier une stratégie orientée-objet

Avec une stratégie orientée-objet, les utilisateurs finissent toutes les actions qui s'appliquent à un même objet avant de passer au suivant. Nous pénalisons donc toute action s'effectuant sur un objet précédemment modifié ou créé. Pour une stratégie $S = \langle (a_1)_{Obj(a_1)}, \dots, (a_n)_{Obj(a_n)} \rangle$ à n actions, nous mesurons le taux $OrientéeObjet(S)$ comme suit :

$$P(S) = \sum_{i=3}^n \begin{cases} 1 & \text{si } Obj(a_i) \neq Obj(a_{i-1}) \\ & \text{et } \exists j \in \llbracket 1; i-2 \rrbracket \text{ tel que } Obj(a_i) = Obj(a_j) \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

$$OrientéeObjet(S) = OriObj(S) = 1 - \frac{P(S)}{n - m} \quad (2.4)$$

Si les utilisateurs effectuent toutes les actions sur un même objet avant de passer au suivant, $P(S) = 0$ et $OriObj(S)=1$. À l'inverse, s'il changent d'objet à chacune de leurs actions, $P(S) = n - m$ (avec m le nombre d'objet dans la zone de travail) et $OriObj(S)=0$.

2.2.2.2 Métrique pour quantifier une stratégie orientée-outil

Avec une stratégie orientée-outil, les utilisateurs gardent le même outil aussi longtemps que possible avant de passer au suivant. Nous pénalisons donc tout changement d'outil sélectionnant un outil précédemment utilisé. Pour une stratégie $S = \langle (a_1)_{Obj(a_1)}, \dots, (a_n)_{Obj(a_n)} \rangle$ à n actions, nous mesurons le taux $OrientéeOutil(S)$ comme suit :

$$P(S) = \sum_{i=3}^n \begin{cases} 1 & \text{si } a_i \neq a_{i-1} \\ & \text{et } \exists j \in \llbracket 1; i-2 \rrbracket \text{ tel que } a_i = a_j \\ 0 & \text{sinon} \end{cases} \quad (2.5)$$

$$OrientéeOutil(S) = OriOut(S) = 1 - \frac{P(S)}{n - c} \quad (2.6)$$

Si les utilisateurs gardent le même outil autant qu'ils peuvent avant de passer au suivant, $P(S) = 0$ et $OriOut(S)=1$. À l'inverse, s'il changent d'outil à chacune de leurs actions, $P(S) = n - c$ (avec c le nombre d'outils disponibles) et $OriOut(S)=0$.

2.2.2.3 Expérience

Pour évaluer nos métriques, nous avons comparé deux nouvelles techniques d'interaction, une *Palette d'Outils Fixe avec Identification des Doigts* et une *Palette Flottante avec Identification des Doigts*, avec une *Palette d'Outils Fixe* et une *Toolglass*.

Participants Nous avons recruté 12 volontaires (26 ans en moyenne, 3 femmes). Quatre d'entre eux ont qualifié leur dextérité manuelle de *bonne* et les huit autres de *normale*. Tous étaient utilisateurs réguliers d'écrans tactiles et d'applications de dessin.

Protocole et tâches Nous avons choisi un plan expérimental intra-sujets composé de 4 TECHNIQUES $\times$ 3 TÂCHES, contrebalancées entre les TECHNIQUES. Chacune des conditions uniques a été répétée 7 fois ($4 \times 3 \times 7 = 84$ données par participant) et l'ordre des conditions TÂCHE $\times$ RÉPÉTITION était pseudo-aléatoire. Les participants ont eu pour instruction de minimiser le temps de réalisation des tâches. Une TÂCHE consistait à reproduire la position, la forme et la couleur de différents objets affichés en transparence et taille réelle à l'écran. La figure 2.10 montre les 3 TÂCHES : chacune contenait 6 objets placés sur une grille de deux lignes et trois colonnes. Les objets étaient soit tous de la même forme et de la même couleur (T1), soit de trois formes et couleurs différentes groupées spatialement (T2) ou mélangées (T3). Toutes les TÂCHES nécessitaient le même nombre d'actions pour être achevées. Toutes les TECHNIQUES donnaient accès aux outils *rectangle*, *ellipse* et *triangle* ainsi qu'aux outils pour peindre en *rouge*, *vert* et *bleu*. Nous avons ajouté un outil de suppression pour corriger les erreurs. Nous avons intentionnellement laissé de côté les outils logiques tels que *copier*, *coller* ou *grouper* pour éviter d'introduire trop de variance dans les données liées à l'utilisation de stratégies très différentes.

Sur l'écran de la TÂCHE étaient affichées les différentes formes en transparence pour permettre aux participants de les dessiner sans avoir besoin de les repositionner : la création des formes s'effectuait en étirant une boîte englobante ; autour de chaque coin de la boîte englobante des différentes formes affichées se trouvait une zone de tolérance de 15 mm (environ la largeur d'un doigt) ce qui indiquait où commencer et terminer le geste de création ; le contour d'une forme devenait rouge lorsque la forme créée dépassait la tolérance autorisée. Les formes nouvellement créées ne possédaient aucune couleur de remplissage. Lorsque la bonne couleur était appliquée, le contour de la forme devenait vert indiquant l'achèvement de l'objet.

Nous avons implémenté deux techniques issue de la littérature : la *Palette d'Outils Fixe*, qui devrait favoriser les stratégies orientées-outils, et la *Toolglass*, qui devrait favoriser les stratégies orientées-objets [Mac02, ABLM05]. En outre, nous avons aussi implémenté deux nouvelles techniques (la *Palette d'Outils Fixe*

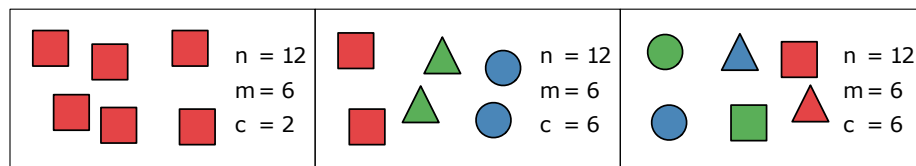


FIGURE 2.10: Exemples d'instances pour les TÂCHES T1, T2 et T3.

avec *Identification des Doigts* et la *Palette Flottante avec Identification des Doigts*) qui devraient, selon nous, favoriser des stratégies orientées-objets.

TECHNIQUE 1 : *Palette d'Outils Fixe* La *Palette d'Outils Fixe*, i.e. barre d'outils, est une technique répandue utilisant un seul pointeur (cf. Figure 2.11-a) [ABLM05]. Elle contient un ensemble d'outils que l'utilisateur sélectionne en appuyant sur le bouton approprié. L'utilisateur tient conceptuellement l'outil jusqu'à ce qu'un autre outil soit sélectionné. Étant donné que le changement d'outils nécessite de larges mouvements entre la palette et la feuille de dessin, nous supposons que les utilisateurs adopteront une stratégie orientée-outils. Nous avons implémenté la *Palette d'Outils Fixe* de telle sorte à ce qu'elle reste fixe sur la partie droite de l'écran.

TECHNIQUE 2 : *Palette d'Outils Fixe avec Identification des Doigts* Nous avons étendu la *Palette d'Outils Fixe* d'une technique à un seul pointeur à une technique multi-pointeurs. L'affichage à l'écran reste le même. Cependant, l'utilisateur peut associer temporairement un outil à chacun de ses doigts de la main dominante : par exemple, en appuyant sur le bouton *rectangle* avec l'index et le bouton *ellipse* avec le majeur, les deux outils peuvent être utilisés en utilisant le doigt *ad hoc*. Étant donné que le changement entre un nombre limité d'outils (5 doigts au maximum) est plus rapide qu'avec la *Palette d'Outils Fixe*, nous supposons que les utilisateurs adopteront une stratégie orientée-objet.

TECHNIQUE 3 : *Toolglass* La *Toolglass* est une technique bi-manuelle utilisant deux pointeurs. Un *widget** contenant des boutons semi-transparents [BSP⁺93] est positionné sur l'écran *via* la main non-dominante. La sélection de commandes s'effectue avec la main dominante (cf. Figure 2.11-b). L'index de la main non-dominante positionne la *Toolglass* principale contenant les six outils. Le majeur positionne lui une *Toolglass* secondaire contenant l'outil de suppression. Étant donné que réutiliser un même outil ou changer d'outil demandent un *effort* similaire, nous supposons que les utilisateurs adopteront une stratégie orientée-objet.

TECHNIQUE 4 : *Palette Flottante avec Identification des Doigts* La *Palette Flottante avec Identification des Doigts* est une technique bi-manuelle multi-pointeurs. La main non-dominante contrôle l'association temporaire mais pré-définie des outils aux doigts de la main dominante : pour un droitier, maintenir l'index gauche en contact avec l'écran associe les outils *rectangle*, *triangle* et *ellipse* à l'index, au majeur et à l'annulaire droit (cf. Figure 2.11-c). Les outils sont utilisés par les doigts de la main dominante indépendamment de la position de la main non-dominante. Pour indiquer les associations entre les doigts et les commandes, nous affichons une notice rappelant les associations près du doigt de la main dominante en contact. Nous avons regroupé les couleurs et les outils de dessin sur les palettes du pouce et de l'index. Une palette contenant l'outil de suppression a elle été associée au majeur. De nouveau, nous supposons

que les utilisateurs adopteront une stratégie orientée-objet puisque toutes les commandes sont directement utilisables de n'importe où sur l'écran.

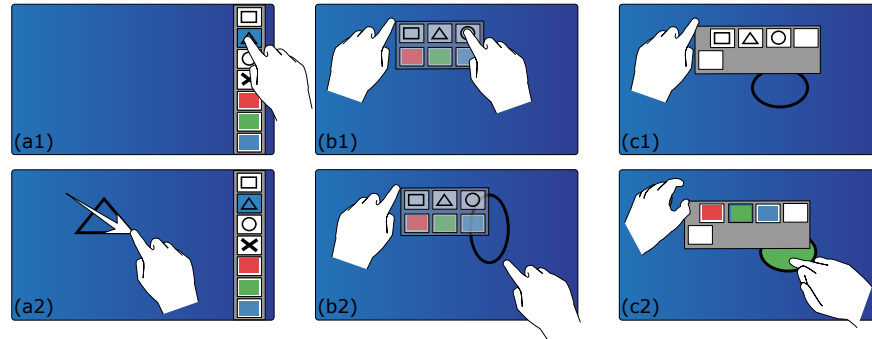


FIGURE 2.11: Illustration de l'utilisation de la *Palette d'Outils Fixe*, la *Toolglass* et la *Palette Flottante avec Identification des Doigts* (ici la main dominante est la main droite) : *Palette d'Outils Fixe*, l'utilisateur sélectionne l'outil triangle (a1) et crée un triangle en glissant son doigt (a2) ; *Toolglass*, l'utilisateur positionne le *widget** semi-transparent à l'aide de sa main non-dominante (b1) et commence à dessiner en appuyant puis glissant son doigt au travers du bouton ellipse (b2) ; *Palette Flottante avec Identification des Doigts*, la main non-dominante contrôle l'association des outils aux doigts de la main dominante (c1), l'utilisateur utilise les outils de coloration à l'aide du pouce de la main non-dominante et peint l'ellipse en vert avec le majeur de la main dominante (c2).

Dispositif Le dispositif utilisé est celui décrit à la section 1.2.2. La figure 2.12 montre un participant qui réalise la TÂCHE T1 avec la *Palette Flottante avec Identification des Doigts*.

2.2.2.4 Résultats et discussion

Les variables dépendantes étaient les taux *OriOut* (i.e. *OrientéeOutil(S)*) et *OriObj* (i.e. *OrientéeObjet(S)*). Une analyse *ANOVA* uni-variée n'a montré aucun effet de RÉPÉTITION sur les taux *OriOut* et *OriObj* ce qui suggère l'absence d'effet d'apprentissage. Une analyse *MANOVA* multi-variée a montré un effet significatif de TECHNIQUE ($F_{6,66} = 10,561, p < 0,0001$) et une interaction significative TECHNIQUE $\times$ TÂCHE ($F_{12,132} = 5.201, p < 0,0001$) sur les taux *OriOut* et *OriObj* (cf. Figure 2.13).

Évaluation des métriques Des analyses *post hoc* ont montré des différences significatives ($p < 0,03$) entre toutes les techniques sauf entre *Palette d'Outils Fixe* et *Palette Flottante avec Identification des Doigts*. La Figure 2.13 montre la distribution des deux taux pour chaque TECHNIQUE. Tout comme les résultats de l'état de l'art [ABLM05, Mac02], les participants ont utilisé des stratégies

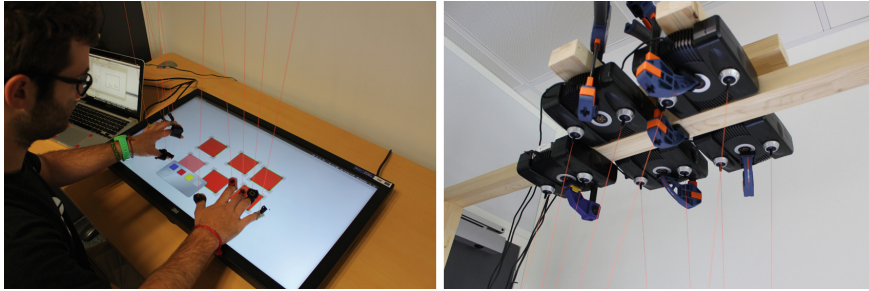


FIGURE 2.12: Dispositif de l'expérience : (gauche) participant complétant la TÂCHE T1 avec la *Palette Flottante avec Identification des Doigts* ; (droite) les 5 *Gametraks** situés au dessus de l'écran tactile multi-points.

orientées-outil lorsqu'ils utilisaient la *Palette d'Outils Fixe* (taux *OriOut* : $\bar{m} = 0,99$, $CI[0,99,1,00]$ et taux *OriObj* : $\bar{m} = 0,05$, $CI[0,04,0,07]$, $\bar{m}$ étant la moyenne) ou orientées-objet lorsqu'ils utilisaient la *Toolglass* (taux *OriOut* : $\bar{m} = 0,51$, $CI[0,46,0,55]$ et taux *OriObj* : $\bar{m} = 0,68$, $CI[0,63,0,74]$). Ces résultats apportent une première validation de nos métriques.

Stratégies orientées-outil pour la *Palette Flottante avec Identification des Doigts* Les taux *OriOut* et *OriObj* ne sont pas significativement différents entre la *Palette Flottante avec Identification des Doigts* (taux *OriOut* : $\bar{m} = 0,93$, $CI[0,91,0,96]$ et taux *OriObj* : $\bar{m} = 0,06$, $CI[0,04,0,08]$) et la *Palette d'Outils Fixe* : bien que le mouvement physique nécessaire soit moins important en utilisant la *Palette Flottante avec Identification des Doigts*, celui-ci n'affecte pas le choix des utilisateurs qui n'adoptent pas des stratégies orientées-objets. Nous supposons que cela est dû au groupement des outils de la technique qui encourage l'utilisation de stratégies orientées-outils.

Stratégies dépendantes de la tâche pour la *Palette d'Outils Fixe avec Identification des Doigts* Pour la *Palette d'Outils Fixe avec Identification des Doigts*, nous avons trouvé des différences significatives ($p < 0,05$) entre les TÂCHES : les utilisateurs ont adopté des stratégies plus orientées-objet pour T1 (taux *OriOut* : $\bar{m} = 0,61$, $CI[0,51,0,71]$ et taux *OriObj* : $\bar{m} = 0,44$, $CI[0,35,0,53]$, cf. point jaune de la figure 2.13) que pour les deux autres tâches T2 et T3 (taux *OriOut* : $\bar{m} = 0,92$, $CI[0,90,0,95]$ et taux *OriObj* : $\bar{m} = 0,14$, $CI[0,10,0,18]$, cf. carré et losange jaunes de la figure 2.13). Dans T1, où la tâche consistait à dessiner des rectangles rouges, les participants nous ont dit que la personnalisation des associations outil-doigt facilitait la mémorisation. Lorsque la diversité des formes et des couleurs augmentait, la mémorisation devenait plus difficile ce qui les faisait basculer vers une stratégie orientée-outils.

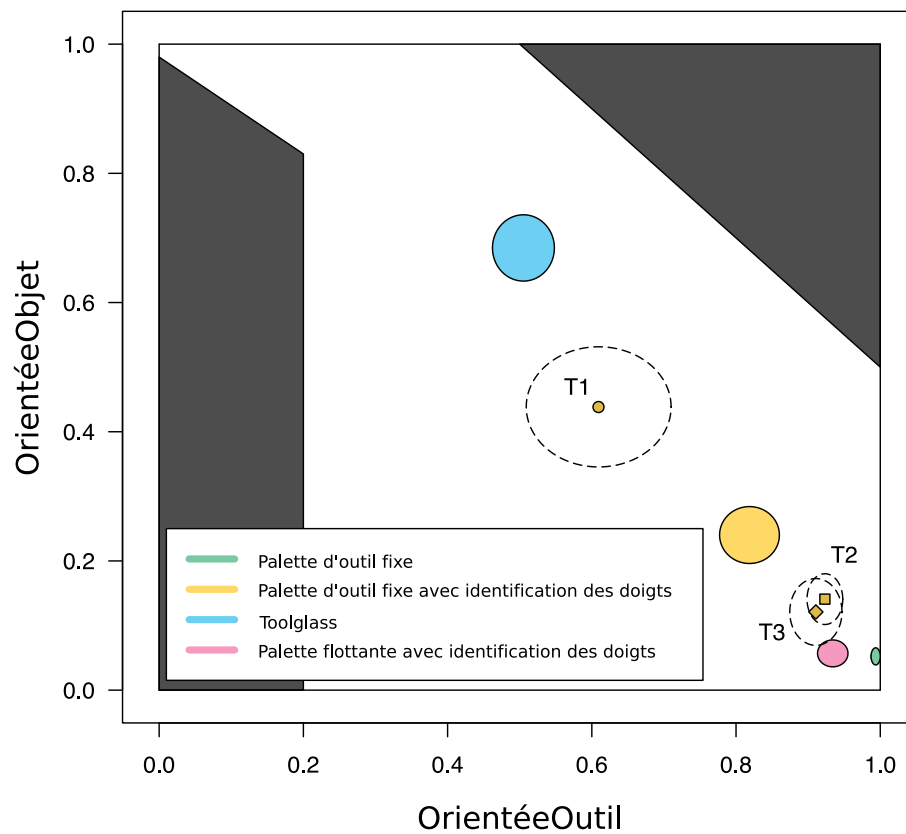


FIGURE 2.13: Taux *OriObj* et *OriOut* moyens pour chaque TECHNIQUE. Les ellipses représentent les intervalles de confiance à 95% pour les moyennes. Les zones grisées représentent les valeurs des taux inaccessibles pour les tâches considérées. Le point jaune représente la stratégie moyenne utilisée pour la tâche T1 avec la *Palette d'Outils Fixe avec Identification des Doigts* et le carré et losange jaunes correspondent aux valeurs moyennes pour T2 et T3, illustrant l'effet d'interaction.

2.2.2.5 Conclusion et travaux futurs

Nous avons introduit deux nouvelles mesures qui combinées quantifient les composantes centrée-objet et centrée-outil des stratégies utilisateurs utilisées lors de la réalisation de tâches. Nos métriques se basent sur la pénalisation du changement d’outil et du changement d’objet d’intérêt à l’écran. Nous avons comparé quatre techniques en mesurant les stratégies utilisateurs sur trois types de tâches de dessin. Nos résultats concordent avec ceux de l’état de l’art concernant la *Palette d’Outils Fixe* et la *Toolglass*, ce qui valide nos métriques [ABLM05, Mac02] : les utilisateurs suivent des stratégies orientées-outils avec la *Palette d’Outils Fixe* et orientées-objets avec la *Toolglass*.

Fusionner les deux métriques en une, avec une projection sur la diagonale, est un moyen élégant de fournir une métrique à une dimension où les deux extrêmes correspondent aux stratégies orientées-objet et -outil. Cependant, n’ayant aucun argument scientifique appuyant la fusion, nous avons explicitement décidé de garder les deux métriques séparées dans l’éventualité où celles-ci seraient effectivement indépendantes.

Pour la *Palette Flottante avec Identification des Doigts*, nous avons trouvé que les utilisateurs suivent des stratégies orientées-outils. Nous en concluons que les techniques minimisant le coût moteur de changement d’outil ne mènent pas nécessairement à l’utilisation de stratégies centrées-objets. Nous supposons que l’organisation et le groupement des outils dans l’interface a un effet sur le choix de la stratégie, ce qui demanderait une vérification dans le cadre de futurs travaux. Nous avons aussi trouvé, dans le cadre de nos tâches, que les utilisateurs favorisent de manière significative l’utilisation de stratégies orientées-objets pour la *Palette d’Outils Fixe avec Identification des Doigts* par rapport à la *Palette d’Outils Fixe* lors de tâches avec peu de diversité d’objets. Une plus grande diversité d’outils mène à une augmentation du coût cognitif à cause des différentes associations outils-doigts à mémoriser. Ce résultat suggère que pour promouvoir des outils d’interaction orientés-objets, il faut à la fois minimiser le coût physique mais aussi le coût cognitif.

Le coût cognitif des techniques basées sur l’identification des doigts ne semble donc pas négligeable démontrant ainsi l’importance de guider l’utilisateur au travers de l’expressivité de l’identification des doigts.

Dans de futures recherches, nous souhaitons adapter nos métriques pour supporter les deux limitations actuelles : (1) étudier nos métriques au travers de tâches réelles où les utilisateurs ne savent pas forcément à l’avance le résultat final de la tâche. (2) adapter nos métriques à des concepts d’outils plus haut niveau. Dans ces travaux, nous avons étudié l’effet de la technique d’interaction sur les séquences d’interaction. Mais les tâches peuvent être résolues en utilisant des concepts d’outils logiques plus haut niveau (*ex. copier puis coller*) comme étudié par Bhavnani *et al.* [BJ00].

L’étude précédente aborde l’influence de l’identification des doigts sur la cognition *via* l’aspect raisonnement. Dans la suite nous nous penchons sur un second aspect : l’apprentissage ou plutôt le ré-apprentissage. Les techniques

décrites jusqu'ici n'ont pas été conçues pour être intégrées aux côtés d'autres déjà existantes. Or la modalité tactile étant répandue, les utilisateurs sont déjà habitués à un certain nombre d'actions. Le but de la prochaine étude est de savoir si l'utilisation de l'identification des doigts doit entraîner une modification des habitudes des utilisateurs.

2.2.3 Mapping

Les travaux décrits dans cette section ont été publiés à la conférence IHM 2016 [MBC16].

Le pavé tactile est un dispositif de pointage indirect généralement intégré aux ordinateurs portables. Les dernières générations de pavés tactiles sont la plupart du temps multi-points. De plus, ces dispositifs sont souvent intégrés dans un écosystème tirant parti de cette dernière caractéristique en proposant la reconnaissance des gestes multi-points classiques (*ex. flicks**, *pinch-and-expand**, *etc.*). Le système d'exploitation *Mac OSX* et ses gestes *Multi-Touch*¹ offre un parfait exemple d'un tel écosystème.

Afin d'étudier quels accords sont couramment associés à quels gestes, nous avons effectué une étude de terrain ciblant les pavés tactiles. Nous nous sommes concentrés sur les pavés multi-points de marque *Apple* qui présentent les avantages suivants : fournir un ensemble conséquent de gestes élégamment intégrés ; donner accès à une Application Programmable Interface (*API*)* fournissant caractéristiques physiques, données brutes et préférences système de l'utilisateur ; offrir un écosystème propice aux études de terrain.

Bien que les pavés tactiles, contrairement aux écrans tactiles, soient des dispositifs de pointage indirects, les gestes multi-points supportés sur les deux dispositifs sont relativement similaires. Nous faisons l'hypothèse qu'une étude sur pavé tactile puisse donner un aperçu du comportement des utilisateurs en général. De plus, une telle étude est beaucoup plus facile à mettre en œuvre dans un contexte écologique sur pavé tactile que sur écran tactile.

2.2.3.1 Observation des accords

Pour observer les accords, nous avons développé une application qui écoute et collecte en permanence les points de contact et les gestes multi-points effectués sur le pavé tactile (*cf.* Table 2.11). En plus de ces données, l'application déclenche une prise de vue de la caméra intégrée de l'ordinateur portable pour chaque geste détecté. Cette prise de vue est rendue possible grâce à l'utilisation d'un miroir fixé devant la caméra pour dévier son champ de vision vers le pavé tactile, et ainsi cadrer la main de l'utilisateur (*cf.* Figure 2.14).

Cette application a été implémentée en *Objective-C* avec la bibliothèque *Cocoa*. Les points de contact sont obtenus grâce à l'*API* privée multi-points d'*Apple*. Aucune *API* publique ne permettant l'écoute des gestes multi-points au niveau système, un travail de rétro-ingénierie a été effectué pour intercepter

1. <https://support.apple.com/fr-fr/HT204895>

Geste	Abr.	Nombre de doigts	Direction	Commande
Défilement	<i>Scroll</i>	2	↕	Scroll
Pinch-and-expand 2-doigts	<i>P/E</i>	2	↖ ↗	Zoom
Rotation 2-doigts	<i>Rotation</i>	2	↻	Rotation
Double-taps 2-doigts	<i>2Taps</i>	2		Zoom intelligent
Scroll vers la gauche	<i>2←</i>	2	←	Notification
Tap 3-doigts	<i>3Tap</i>	3		Lookup
Drag 3-doigts	<i>3Drag</i>	3	↕	Drag
Flick Horiz. 3-doigts	<i>3Flick↔</i>	3	↔	Changer bureau/page
Flick Vert. 3-doigts	<i>3Flick↑↓</i>	3	↑↓	↑Mission Control ↓Expose
Flick Horiz. 4-doigts	<i>4Flick↔</i>	4	↔	Changer bureau
Flick Vert. 4-doigts	<i>4Flick↑↓</i>	4	↑↓	↑Mission Control ↓Expose
Pinch-and-expand 4-doigts	<i>4P/E</i>	4	↖ ↗	Launchpad/Bureau

TABLE 2.11: Gestes multi-points du pavé tactile *Apple*. Chaque ligne contient : le nom du geste, son abréviation, le nombre de doigts utilisés pour effectuer le geste, les directions possible du gestes et les commandes associées.

une partie des événements de l'*API Quartz*. Toutes les données collectées sont stockées dans une base de données *SQLite*. L'accès au flux de la caméra se fait à l'aide de la bibliothèque *AV Foundation* d'*Apple* et les images capturées sont stockées au format *Jpeg2000* pour minimiser l'utilisation de la mémoire disque. Dans la même optique, une seule photo est stockée lorsque deux gestes de même type sont effectués sur une période de moins de 3 secondes (*ex.* navigation dans un document). Un algorithme de détection de visages est utilisé pour mettre en pause le flux de la caméra lorsqu'un visage a été détecté (signifiant par conséquent que le miroir n'est pas en place). Le miroir est collé à un support en plastique imprimé en 3D et équipé d'aimants. En tirant parti des aimants intégrés au cadre de l'écran de l'ordinateur portable (utilisés pour la mise en veille de l'appareil), le miroir est placé à une position pré-définie sans nécessiter de précision particulière de la part de l'utilisateur. Il peut ainsi s'installer et se désinstaller en quelques secondes. Notre dispositif est similaire au miroir commercial *Osmo*², vendu 89€, utilisé par Zheng et al. pour proposer de nouveaux raccourcis clavier [ZV16].

2.2.3.2 Protocole et déploiement

Nous avons recruté 9 volontaires (30,8 ans en moyenne, 1 gaucher³) qui ont installé notre application sur leur ordinateur portable pour une période de deux semaines. Tous les ordinateurs étaient des ordinateurs *Macbook* de début 2011 ou plus récents tournant sous *Mac OS Yosemite*. Tous étaient équipés de pavé tactile multi-points ayant les mêmes caractéristiques de taille. Après les deux semaines, les participants étaient invités à supprimer les images non pertinentes et à nous envoyer l'ensemble des données.

2. <https://www.playosmo.com/fr/>

3. 100% des gestes effectués par l'utilisateur gaucher impliquaient uniquement les doigts de sa main droite. Pour les autres participants, ce taux était au minimum de 94,2%.

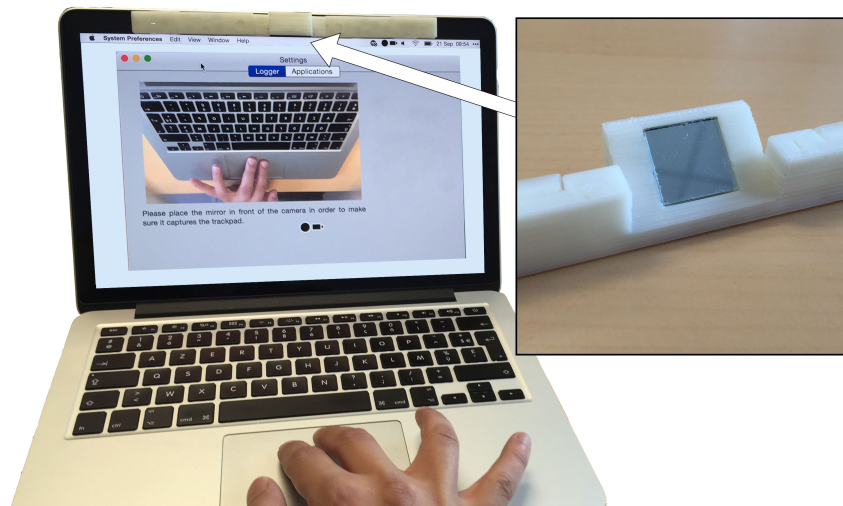


FIGURE 2.14: Miroir fixé devant la caméra intégrée qui dévie le champ de capture vers le pavé tactile. L'image capturée permet d'annoter les doigts à l'origine d'un geste.

2.2.3.3 Résultats

Nous avons collecté un total de 26 664 photos que nous avons annotées manuellement à l'aide d'une application dédiée pour associer chaque geste à l'accord correspondant (cf. Figure 2.15). Nous avons ensuite calculé le taux d'occurrence de chaque accord par geste et par utilisateur. 97,3% des accords utilisés impliquaient uniquement la main droite. Cinq participants avaient activé le *3Drag*. Six participants avaient activé les *flicks** à 4 doigts droite-gauche et haut-bas. Trois participants avaient activé les *flicks** à 3 doigts droite-gauche et haut-bas. Tous les autres gestes étaient activés pour tous les participants.

Le détail des observations est donné en annexe B.

Pointage L'index ($I_{\rightarrow}$) et le majeur ($M_{\rightarrow}$) étaient les deux doigts les plus utilisés pour le pointage (respectivement 48% et 39% du temps). Quatre participants étaient constants et ont utilisé le même doigt (l'index ou le majeur) au moins 79% du temps. Les cinq autres participants étaient moins constants mais ont tout de même utilisé ces deux doigts au moins 71% du temps.

Défilement Majeur + Annulaire ($MA_{\rightarrow}$) et Index + Majeur ($IM_{\rightarrow}$) étaient les deux accords les plus utilisés pour le défilement (respectivement 56% et 40% du temps). Sept utilisateurs ont utilisé le même accord ($MA_{\rightarrow}$ ou $IM_{\rightarrow}$) au moins 92% du temps. Les deux autres utilisateurs étaient moins constants mais ont tout de même utilisé ces deux accords au moins 94% du temps.

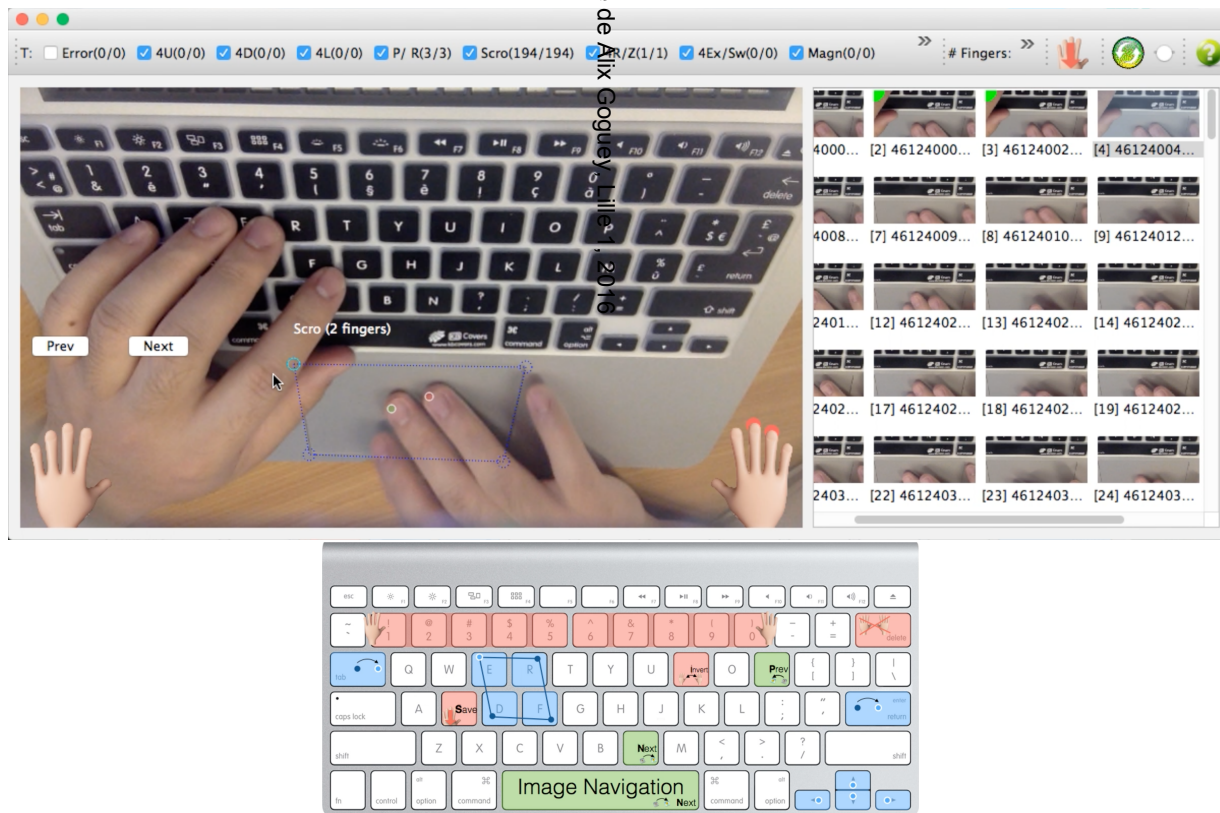


FIGURE 2.15: Application d'annotation des gestes collectés. Après avoir placé à la souris les coins du pavé tactile, les contacts sont projetés sur celui-ci soulignant les doigts utilisés. L'expérimentateur n'a plus qu'à appuyer sur la combinaison de touche du clavier correspondant à l'accord. Dans l'exemple, pour annoter l'accord majeur+annulaire, l'expérimentateur appuie sur les touches 8 et 9.

***Pinch-and-expand* à 2 doigts** $PI_{\rightarrow}$ et $IM_{\rightarrow}$ ont été utilisés respectivement 36% et 20% du temps. Nous n'avons observé aucun comportement concordant entre les différents utilisateurs. Cependant, trois utilisateurs ont été extrêmement constants en utilisant l'un de ces deux accords entre 80% et 98% du temps. Les autres utilisateurs avaient une plus grande variabilité utilisant entre 3 et 5 accords différents. Le manque de concordance pourrait s'expliquer par le faible nombre de données collectées pour ces gestes (107/26 664).

Rotation $PI_{\rightarrow}$ et $IM_{\rightarrow}$ ont été utilisés respectivement 33% et 24% du temps. Tout comme pour *Pincement/Étirement* à 2 doigts, nous n'avons observé aucun comportement concordant entre les différents utilisateurs. Cependant, cinq de nos neuf participants ont montré une grande constance en utilisant le même accord plus de 75% (voire 100%) du temps bien que l'accord ne faisait pas nécessairement partie du lot $PI_{\rightarrow}$ et $IM_{\rightarrow}$. Une fois de plus, le manque de concordance pourrait s'expliquer par le faible nombre de données collectées (40/26 664). Il est toutefois intéressant de noter que le participant qui a utilisé 5 accords différents pour le *Pincement/Étirement* à 2 doigts a utilisé les 5 mêmes pour la *Rotation*.

Glissement à 3 doigts $IMA_{\rightarrow}$ et $PIM_{\rightarrow}$ étaient les deux accords les plus utilisés (respectivement 68% et 28% du temps) avec cinq participants sur six utilisant l'un de ces deux accords au moins 94% du temps (quatre favorisant l'utilisation de $IMA_{\rightarrow}$) et le dernier participant utilisant $PIM_{\rightarrow}$ 73% et $IMA_{\rightarrow}$ 18% du temps.

***Flicks* à 3 doigts** $IMA_{\rightarrow}$ était l'accord le plus utilisé (81% du temps). Deux participants sur trois l'ont utilisé très souvent (respectivement 100% et 93,2% du temps) alors que le dernier participant l'a utilisé 50% du temps.

***Flicks* à 4 doigts** $IMAL_{\rightarrow}$ était de loin l'accord le plus utilisé (98% du temps). À noter que seul un autre accord a été utilisé $IMAL_{\leftarrow}$ (2%).

***Pinch-and-expand* à 4 ou 5 doigts** Seuls trois accords ont été utilisés : $IMAL_{\rightarrow}$ (54,7%), $PIMAL_{\rightarrow}$ (41,5%) et $PIMA_{\rightarrow}$ (3,8%).

2.2.3.4 Conclusion et travaux futurs

Notre étude montre que l'index et le majeur sont les doigts les plus utilisés, surtout pour le pointage et le défilement, et que les gestes fortement utilisés, tel que le défilement, sont effectués de façon constante avec les mêmes doigts. Pour les gestes rarement utilisés (comme *Pincement/Étirement* ou la *Rotation*), les utilisateurs semblent varier les accords.

La caractérisation des différents doigts utilisés pour chaque geste permet de guider les concepteurs de techniques d'interaction pour pavés tactiles utilisant l'identification des doigts. Comme nous avons pu l'observer, les utilisateurs changent d'accord pour effectuer un même geste. Cependant pour les gestes

s'effectuant à 3 doigts ou moins, les accords semblent se concentrer sur les premiers doigts de la main (pouce, index et majeur). L'utilisation de l'identification des doigts et l'adaptation aux habitudes des utilisateurs ne semble donc pas incompatibles. L'annulaire et l'auriculaire pourraient par exemple être utilisés comme déclencheurs des quasi-modes basés sur l'identification des doigts (*cf.* Section 3.2).

Une limitation possible de ce travail serait le faible nombre de participants à l'étude. Recruter des personnes ayant un ordinateur avec les caractéristiques techniques requises et acceptant d'installer une application enregistrant leurs actions n'est pas simple. Nous avons donc décidé de compenser notre faible nombre de participants en collectant un nombre conséquent de données. Enfin, notre étude ne s'intéresse qu'au pavé tactile de la marque *Apple* car ils offrent une grande variété de gestes qui sont élégamment intégrés dans un écosystème favorisant leur utilisation et rendant l'étude intéressante. Des travaux complémentaires sont nécessaires pour reproduire cette étude dans d'autres écosystèmes (*ex.* *Windows 10*) et sur des dispositifs de pointage direct (*ex.* *smartphones**, tablette, *etc.*) afin de confirmer les comportements utilisateurs ou en découvrir de nouveaux.

2.3 Conclusion

Dans ce chapitre, nous avons présenté trois études ayant pour but de cerner un peu plus les implications de l'identification des doigts sur l'utilisateur. La première étude est tournée vers la dextérité, les deux autres sont tournées vers la cognition.

Par définition, l'identification des doigts contraint les doigts et accords à utiliser pour effectuer une action là où l'utilisateur choisissait, sciemment ou non, ceux avec lesquels il était le plus à l'aise. Connaître les différences de performances et de préférences est donc primordial pour guider le choix des associations entre commandes et doigts/accords. Nous avons donc étudié ces différences aux travers de cinq expériences chacune dédiée à une tâche classique de l'interaction tactile. Grâce aux résultats obtenus, nous avons compilé une série de recommandations sur les doigts/accords afin d'aider les concepteurs de techniques d'interaction.

Pour étudier l'influence sur le raisonnement de l'utilisateur, nous avons présenté deux métriques qui permettent de quantifier les stratégies utilisateurs. En analysant les résultats nous avons remarqué une différence entre les stratégies trouvées et prédites pour les techniques qui utilisent l'identification des doigts. Ces différences suggèrent que la qualité de la conception de la technique est un facteur clef : bien que ces techniques permettent de réduire les mouvements de l'utilisateur pour accomplir une tâche, les stratégies choisies étaient loin d'être optimales d'un point de vue moteur. Étant donné l'effort de mémorisation nécessaire, les utilisateurs ont donc choisi un compromis entre réduction du coût moteur et réduction du coût cognitif.

L'intégration de l'identification des doigts pose un problème d'adaptation : les utilisateurs ont potentiellement déjà leurs habitudes et associent déjà une

commande à un accord. Pour étudier ces habitudes, nous avons déployé un système qui observe les accords utilisés pour un geste donné. Les résultats ont montré que les utilisateurs n'associent pas une commande à un unique accord. Cependant, cette variabilité est relative : la majorité des accords, à un, deux ou trois doigts, est une combinaison des premiers doigts de la main (pouce, index, majeur). En fonction du nombre de fonctionnalités à intégrer, imposer des accords pourrait ne pas être un changement des habitudes utilisateurs.

Fort de ces différents résultats et recommandations, nous nous attachons maintenant à les appliquer. Dans le chapitre suivant, nous présentons une technique d'interaction basée sur l'identification des doigts. Sa conception et sa mise en œuvre ont été émaillées de choix motivés par les conclusions des trois études précédentes.

Chapitre 3

Technique d'interaction

Les travaux décrits dans ce chapitre ont été publiés à la conférence IHM 2014 [GCP⁺14a, GCP⁺14b], à la conférence UIST 2014 [GCV⁺14] et ont donné lieu à la soumission d'un article au journal IJHCS.

Avec l'identification des doigts, le vocabulaire d'entrée pour les interactions multi-points se retrouve très fortement augmenté (1023 accords). Afin de tirer partie de cette expressivité, nous proposons *Adoiraccourcix*, une technique de sélection de commandes et de manipulation de paramètres s'inspirant des raccourcis clavier. À l'instar de ces derniers, nous proposons d'utiliser un sous-ensemble de doigts comme des *sélecteurs* (ex. **Ctrl**, **Alt**, **Shift**, ...) et un autre sous-ensemble comme des *déclencheurs* de commandes (ex. **C**, **V**, **X**, ...). L'utilisation d'*Adoiraccourcix* s'effectue en 3 étapes : (1) un accord de *sélecteurs* sélectionne un quasi-mode associant une commande à chaque *déclencheur* ; (2) l'interaction d'un *déclencheur* sur l'écran sélectionne la commande associée (une fois la commande déclenchée, l'accord *sélecteur* peut être retiré) ; (2-optionnel) si la commande possède un paramètre, le mouvement du *déclencheur* spécifie sa valeur ; (3-optionnel) un accord de *modificateurs* (i.e. sous ensemble de *sélecteurs*) joué pendant la manipulation d'un paramètre contraint ce dernier. La figure 3.1 illustre ces 3 étapes avec pour *sélecteurs* les doigts de la main gauche et pour *déclencheurs* ceux de la main droite.

Pour s'assurer que les nombreuses combinaisons ne soient pas un frein pour l'utilisateur, *Adoiraccourcix* intègre un système de *feedforward*^{*} (cf. Figure 3.1-ab) qui indique à l'utilisateur les commandes associées aux doigts *déclencheurs* pour le quasi-mode courant. De plus, contrairement aux raccourcis clavier, les actions effectuées par un utilisateur novice et un utilisateur expert sont les mêmes, ce qui favorise la transition de l'un vers l'autre.

Adoiraccourcix est adaptable à différents facteurs de forme (ex. grand écran, tablette, *smartphone*^{*}). Cependant le choix des *sélecteurs* et des *déclencheurs* ainsi que les associations commande/accord sont critiques. En effet, différents facteurs comme l'habilité motrice de chaque doigt (cf. Section 2.1) ou la compatibilité avec les interactions communément utilisées (cf. Section 2.2.3), sont à

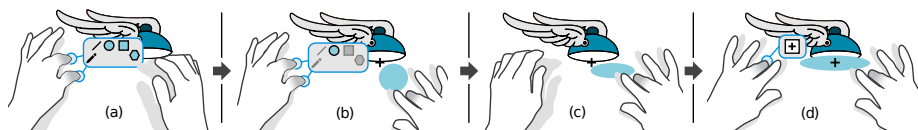


FIGURE 3.1: Exemple bi-manuel illustrant la logique d'*Adoiraccourcix* combinant sélection de commande avec manipulation continu de paramètre et *feedforward** : (a) un accord de doigts de la main non-dominante affiche du *feedforward** précisant les différentes commandes associées aux doigts de la main dominante, pour cet accord. Ici le pouce et l'index affichent les cinq outils de dessin associés aux doigts de la main dominante; (b) le majeur de la main dominante sélectionne l'outil de création d'ellipses; (c) le mouvement du majeur de la main dominante contrôle les paramètres de la commande, ici la taille et la forme de l'ellipse; (d) un autre accord de la main non-dominante définit une contrainte pour la commande en train d'être exécutée. Ici le majeur de la main non-dominante contraint la création d'ellipse suivant son centre plutôt que son coin supérieur gauche. Un exemple d'utilisation bi-manuel restreint sur tablette et un exemple d'utilisation uni-manuel sur *smartphone** sont donnés dans la section 3.3.

prendre en compte dans la mise en œuvre d'*Adoiraccourcix*.

Dans un premier temps, nous proposerons un état de l'art sur les différentes techniques de sélection de commande sur interfaces tactiles. Nous ne repréciserons pas les technologies utilisées, toutes ayant déjà été décrites à la section 1.1.2. Nous détaillerons ensuite : l'espace des entrées basées sur l'identification des doigts, le mécanisme d'interaction derrière *Adoiraccourcix* ainsi que les éléments que nous prenons en compte pour la conception de cette technique. Nous présenterons des exemples de mises en œuvre d'*Adoiraccourcix* sur trois facteurs de forme : grand écran, tablette et *smartphone**. Enfin nous nous proposerons d'évaluer *Adoiraccourcix* au travers de deux études et d'une analyse basée sur le modèle Fingerstroke-Level Model (*FLM**).

3.1 État de l'art

Dans cet état de l'art, nous nous intéressons aux techniques de sélection de commandes et de manipulation de paramètres de la modalité tactile. Contrairement aux sections 1.1, 2.1.1 et 2.2.1 qui abordaient respectivement l'aspect technologique de la modalité tactile, l'aspect dextérité du facteur humain et l'aspect cognitif du facteur humain, nous nous concentrons ici sur la description des mécanismes d'interaction.

Une interface à manipulation directe doit fournir la possibilité de spécifier une commande au système. Il existe des commandes effectuant des actions globales (*ex. annuler*) et des commandes effectuant des actions contextuelles liées à un objet

spécifique (*ex. supprimer ce rectangle*). Il existe aussi des commandes qui activent des *modes*, dans lesquels le système interprète les entrées de façon spécifique (*ex. commencer à dessiner un rectangle*). Les modes sont conceptuellement associés à des *outils* de la vie réelle. Les commandes peuvent aussi modifier le mode courant (*ex. continuer de dessiner un rectangle, mais le dessiner depuis le centre*). La plupart du temps, ces *modificateurs* sont maintenus temporairement *via* une action kinesthétique, par exemple maintenir une touche enfoncée. Au niveau des commandes, cela signifie qu’enfoncer une touche envoie une commande qui débute la modification du mode courant et que la relâcher envoie la commande d’arrêt. Ce principe d’action kinesthétique temporaire peut être appliqué aux modes et sont, pour l’occasion, renommés *quasi-modes* [Ras00].

Sur interface multi-points, fournir une façon efficace de spécifier ces commandes est un vrai défi. Le nombre de points de contact simultanés procure une certaine expressivité, mais celle-ci est loin d’être aussi pourvue que le vocabulaire des raccourcis clavier (*cf.* Section 1.1). Les techniques qui permettent d’augmenter le vocabulaire de l’espace multi-points se sont concentrées sur une nouvelle interprétation de l’information spatiale, la temporisation et les gestes effectués. Cependant, elles augmentent aussi le temps d’interaction (*ex.* appuis longs) et/ou la distance physique à parcourir (*ex.* geste) et détournent l’attention de l’utilisateur de l’objet d’intérêt [BL00].

Pour résoudre ce problème d’expressivité, nous avons vu que les chercheurs se tournent vers le *Rich Touch* et en particulier vers l’identification des doigts. Cependant peu de travaux tirent vraiment partie de la richesse de cette nouvelle information. La suite de cet état de l’art a pour but de donner un aperçu des différentes techniques de sélection de commandes sur écran tactile, puis de présenter celles basées sur l’utilisation d’accords et de l’identification des doigts. Étant l’un des avantages d’*Adoiraccourcir*, nous présentons aussi les travaux qui fusionnent la sélection de commande et la manipulation directe.

3.1.1 Du *Touch* au *Rich Touch* : aperçu de la sélection de commande

La sélection de commandes sur les interfaces tactiles s’effectue principalement à l’aide de boutons affichés à l’écran et de menus contextuels activés *via* des appuis longs. Outre le ralentissement de la fluidité d’interaction, lorsque l’on veut augmenter le nombre de commandes disponibles, le manque de place à l’écran implique de cacher la majorité des commandes dans des menus voire des sous-menus. Plus une commande est profonde dans l’arborescence des menus, plus le nombre d’actions requises et le temps nécessaire pour sa sélection augmente. Les chercheurs visent donc à développer de nouvelles techniques d’interaction de manière à de manière à réduire l’utilisation des menus et/ou réduire le temps de sélection.

En plus des boutons et des menus, les interfaces tactiles sont capables de reconnaître un ensemble de gestes tels que les *flicks** directionnels ou le *pinch-and-expand**. Dans la majorité des cas, ces gestes fusionnent à la fois sélection de commandes et manipulation de paramètres. Par exemple en effectuant un *pinch-*

*and-expand**, l'utilisateur sélectionne la commande de changement de niveau de zoom et contrôle celui-ci avec la distance entre les deux doigts. Le nombre de gestes est cependant limité ce qui réduit de fait le nombre de commandes disponibles. Avec *Scriboli* [HBRG05], Hinckley *et al.* proposent une solution à la limitation du nombre de gestes. Leur idée est de finir un geste de sélection au lasso par une queue en tire bouchon qui déclenche ainsi la sélection de commandes *via* un *marking menu* [KB94]. La sélection de la commandes dans les *marking menus* utilisés par *Scriboli* se fait à l'aide d'une temporisation. Une fois la commande sélectionnée, le contrôle de paramètres s'enchaîne directement et la commande s'applique à l'objet sélectionné au lasso. Bien que le tout s'effectue en un seul geste, les différentes étapes, dont la temporisation, ne réduisent pas le temps de sélection.

Kin *et al.* [KHA11] sépare l'écran en deux zones. Chaque zone active un *marking menu*. Ces deux *marking menus* s'utilisent en complémentarité : le premier contact sélectionne un élément du *marking menu* de plus haut niveau à l'aide d'un *flick** directionnel ; le second contact sélectionne un élément du sous menu activé toujours à l'aide d'un *flick** directionnel. Les deux *flicks** peuvent s'effectuer l'un après l'autre ou simultanément. Cette méthode à l'avantage de réduire le temps de navigation dans les menus et sous menus. Avec *FastTap*, Gutwin *et al.* [GCS⁺14b] utilisent toute la taille de l'écran pour afficher l'ensemble des commandes disponibles. Un bouton situé dans le coin en bas à gauche de l'écran active l'affichage d'une grille dont chaque case correspond à une des commandes. La sélection d'une ou plusieurs commandes s'effectue en appuyant sur les cases correspondantes. Une fois le bouton du coin relâché, la commande nouvellement sélectionnée est active. La grille est affichée avec un délai de 300 ms pour favoriser le passage de novice à expert. En mode expert, l'utilisateur peut directement effectuer un accord sur l'écran en plaçant ses doigts aux bonnes positions. Le *Bezel-Tap* [SLG13] est une technique de sélection de commandes sur tablette. Pour déclencher une commande, l'utilisateur effectue un *tap** sur le contour de l'écran puis sélectionne une zone sur la périphérie de l'écran. Le premier *tap** permet d'entrer dans un mode de sélection de commandes, y compris quand la tablette est en veille, en détectant un pic dans les données des accéléromètres. Une fois ce mode activé, la sélection d'une commande se fait *via* des boutons virtuels placés sur les bords de l'écran tactile. En mode expert, les boutons virtuels ne sont pas affichés et l'utilisateur peut déclencher une commande en un geste rapide.

En reconnaissant le nombre de contacts simultanés sur une surface tactile, l'espace des entrées peut être augmenté de manière significative. Appuyer sur une surface avec deux doigts quelconques pour ouvrir un menu contextuel peut être considéré comme étant un type de commande activé *via* un accord. Le nombre de doigts peut aussi permettre de sélectionner des éléments dans un menu [BLG10]. Une telle méthode permet d'associer jusqu'à 25 commandes. Cette technique est utilisée pour l'appel à des commandes globales, comme *annuler*, où les accords sont joués sans toucher les objets. Compte tenu de la surface requise pour effectuer ces accords, l'utilisation de cette technique sur des objets particuliers pour des commandes contextuelles semble difficile. À noter

que l’utilisation des méthodes précédentes pour des commandes modales avec paramètres (*ex. démarrer le dessin de rectangles, et dessiner le premier à cet endroit*) n’apparaît pas faisable à cause de l’imprévisibilité de l’endroit où sera exécutée la commande.

Avec leurs schémas rythmiques, Ghomi *et al.* [GFH⁺12] accèdent aux différentes commandes en jouant des phrasés à la manière du code *Morse*. Chaque commande est accessible en temps constant. La longueur des phrasés dépend du nombre de commandes accessibles (*cf.* Section 1.1.1.1). N’ayant qu’un seul point de contact à la fois, cette technique peut permettre l’enchaînement avec la sélection de commandes et la manipulation directe de paramètres. Cependant, elle souffre du même problème que les appuis longs.

3.1.2 Les techniques basées sur l’identification des doigts

Les exemples d’interaction basée sur l’identification des doigts de la littérature proposent principalement d’associer un doigt ou un accord à une commande. L’utilisation du doigt ou de l’accord sert ensuite à déclencher la commande *ad hoc*. Cependant, avant de présenter ces techniques d’interaction, il est intéressant de mentionner d’autres utilisations de l’identification des doigts.

Dans [MLH⁺16], Ma *et al.* proposent d’utiliser l’identification des doigts comme un nouveau niveau de sécurité lors de la saisie de code *PIN*. Leur code prend non seulement en compte la séquence de chiffres, mais aussi les doigts avec lesquels aura été saisie cette séquence.

Bien que Vega *et al.* [VF13] et Zheng *et al.* [ZV16] n’utilisent pas l’identification des doigts sur des interfaces tactiles, leurs techniques d’interaction respectives permettent la sélection de commandes voire le contrôle continu de paramètres. Dans [VF13], les différents doigts sont utilisés pour contrôler des pistes audio dans une application de mixage ainsi que les différentes notes d’un synthétiseur. Dans [ZV16], l’appui de certaines touches du clavier avec un doigt particulier déclenche un raccourci clavier différent. Zheng *et al.* proposent aussi la navigation dans un menu, déclenchée de manière similaire, en fonction de l’angle formé par le pouce.

3.1.2.1 Spécification de commandes à l’aide de doigts et d’accords

Indépendamment de la méthode de détection, une technique d’interaction répandue et qui démontre le potentiel de l’identification des doigts consiste à associer différentes commandes à chacun des doigts. Sugiura *et al.* utilisent un scanner d’empreintes digitales pour identifier l’index, le majeur, l’annulaire et l’auriculaire. Ils utilisent ces doigts pour contrôler un lecteur de musique et ajouter des favoris dans un navigateur Internet [SK98]. Sur le même principe, Wang *et al.* [WC04] déclenchent des actions différentes en fonction du doigt qui appuie sur un même bouton. Dans *Dualkey*, Gupta *et al.* [GB16] regroupent les lettres d’un clavier *Qwerty* par deux. Un bouton correspond donc à deux symboles différents : appuyer avec l’index, respectivement le majeur, permet de saisir le premier symbole associé à la touche et respectivement le second.

Grâce à cette technique, ils réduisent le problème d'acquisition de petites cibles sur l'écran des montres tactiles. En utilisant un gant équipé de marqueurs, Marquardt *et al.* identifient le bout des doigts, les jointures et d'autres parties de la main [MKL⁺11]. Les utilisateurs peuvent associer un outil ou une commande à chaque marqueur. Ils peuvent ainsi dessiner avec le bout de leur index ou avoir un aperçu de l'outil associé en posant la jointure de leur index sur la surface tactile. Colley *et al.* [CH14] ont implémenté deux applications pour *smartphone** qui utilisent l'identification des doigts : un répertoire téléphonique et une galerie d'image. Dans le répertoire, le pouce appelle un contact, l'index affiche les détails d'un contact, le majeur affiche un menu d'ouverture rapide d'application, l'annulaire supprime le contact et l'auriculaire affiche l'aide du système. Dans la galerie d'image, le pouce affiche le menu de partage, l'index déclenche le mode d'édition de l'image et l'annulaire supprime l'image. Le majeur et l'auriculaire ont la même fonction que précédemment. Cependant, associer un outil à un doigt ne permet d'accéder qu'à un nombre limité de commandes.

D'autres techniques basées sur les accords introduisent un certain niveau d'identification des doigts, mais optent pour la même approche basique : directement associer un accord à une commande. Benko *et al.* [BSMT09] associent les accords pouce + index et pouce + majeur aux commandes copier et couper. Le menu *Palm* [AT10] et *Arpège* [GHB⁺13] affichent tous deux des positions prédéfinies de doigts pour reconnaître des accords et invoquer des commandes. *Arpège* implémente un intéressant système de découverte des associations commandes-accords. Les accords sont formés progressivement à l'aide de *feedforward**, ce qui aide à sélectionner la commande voulue. Les noms des commandes possibles sont affichés près des doigts et sont mis à jour à chaque changement d'accord. Les commandes sont exécutées quand tous les doigts quittent la surface. Les auteurs ont aussi montré que les utilisateurs sont capables d'exécuter facilement jusqu'à 480 accords. Bien que ce système permette l'utilisation de *feedforward** en exécutant les commandes au lever des accords, il empêche d'effectuer dans un même geste une manipulation continue des paramètres. Sans donner d'exemple d'application concret, Wagner *et al.* [WLS14] ont associé 9 accords à trois doigts à différentes commandes. En effet, leur travail se focalise sur la mémorisation des associations. Il est toutefois intéressant de noter qu'un même accord peut déclencher différentes commandes, puisque la reconnaissance des accords est progressive. Ainsi les accords pouce + index + auriculaire et pouce + auriculaire + index (construits dans cet ordre) sont différents aux yeux de leur système. Lepinski *et al.* [LGF10] vont au delà d'une simple association commande-accord. Chaque commande est associée à un accord et une direction : les utilisateurs effectuent un accord puis déplacent leurs doigts dans une des huit directions reconnues pour exécuter une commande. Ces gestes rendent malheureusement les manipulations directes qui découlent d'un appel de commande difficile.

3.1.2.2 Fusionner sélection de commandes et manipulation directe

Fusionner la sélection de commandes et la manipulation directe permet de réduire à fois la distance articulaire¹ et la distance sémantique² [HHN85]. Spécifier une commande à l’endroit où l’on souhaite l’utiliser réduit la distance articulaire. De la même façon, l’utilisation d’actions explicites et directes, ne nécessitant pas la manipulation de concepts intermédiaires, réduit la distance sémantique. Guimbretière *et al.* [GMW05] défendent, au travers d’un autre exemple de sélection de commandes, que l’aspect le plus important de la *Toolglass* [BSP⁺93] est que l’appel d’une commande et la manipulation de ses paramètres s’effectuent en un seul mouvement. Avec les accords multi-points, la fusion de la sélection de commandes et de la manipulation directe n’est pas triviale. En effet, plusieurs doigts étant en contacts simultanés, l’endroit où sera exécutée la commande devient ambigu. De plus, l’occultation engendrée par la main rend difficile la mise en place de stratégies similaires à la *Toolglass*.

Les techniques d’interaction qui fusionnent spécification de commandes et manipulation directe sont similaires aux techniques vues précédemment : elles utilisent une association fixe entre commandes et doigts, voire accords. Cette implémentation permet l’enchaînement avec la manipulation directe. Benko *et al.* associent une couleur différente à chaque doigt de telle sorte que *dessiner une ligne rouge* puisse être réalisé en une action unique [BSMT09]. Malik *et al.* utilisent les index gauche et droit afin d’activer le mode de *pan** et de pointage et spécifier ainsi dans un même geste les paramètres [MRB05]. L’accord pouce-index est aussi utilisé pour activer le mode de redimensionnement des objets où la taille évolue proportionnellement avec la distance physique entre les doigts de l’accord. De même, un accord pouce-index-majeur permet d’activer le mode de zoom. Une fois l’accord effectué, lever l’index agrandit la vue et à l’inverse lever le pouce la rétrécit. Ces techniques souffrent cependant d’un problème : la découverte des associations commande/doigt(s) est peu évidente y compris en y intégrant du *feedforward** à la manière d’*Arpège*.

Marquardt *et al.* adressent le problème de la découverte des commandes en utilisant les jointures des doigts et non du *feedforward** [MKL⁺11]. Par exemple, toucher l’écran avec la jointure du majeur affiche une info-bulle montrant que la commande *couper* est associée au bout du doigt. Mais cette méthode n’est pas utilisée de manière constante : certaines jointures sont associées à la copie et au déplacement de piles d’objets. Il n’est pas non plus certain que les jointures soient indiquées pour révéler les info-bulles liées aux accords. En revanche, la façon dont

1. La distance articulaire représente l’effort physique nécessaire à l’utilisateur pour résoudre une tâche. Réduire la distance articulaire c’est minimiser les mouvements physiques par lesquels l’utilisateur doit passer (*ex.* l’utilisation de menu contextuels sur les interfaces Windows, Icons, Menus and Pointing device (*WIMP*)* au lieu d’utiliser les icônes des barres d’outils souvent loin de l’objet d’intérêt).

2. La distance sémantique représente la différence entre le niveau d’abstraction offert par une interface et la représentation mentale de la tâche qu’à l’utilisateur. Réduire la distance sémantique c’est réduire l’effort cognitif utile pour accomplir la tâche, en utilisant par exemple des notions familières (*ex.* la rotation à deux doigts d’une photo sur les interfaces tactiles commerciales utilise la notion familière de la rotation d’une feuille de papier dans le monde réel).

Marquardt *et al.* utilisent les accords de la main non-dominante pour mélanger les couleurs (ajout de bleu avec le pouce, de jaune avec l'index et de rouge avec le majeur) est particulièrement intéressante pour nos travaux. Kung *et al.* [KKS⁺12] proposent une extension de *Eden* [KMB⁺11] : une application de construction de scènes en trois dimensions pour films d'animation. L'index de la main dominante permet de déplacer les différents objets de la scène. Le majeur permet la rotation de ces objets. Le pouce et le majeur modifient la taille des objets. Quand le pouce de la main non-dominante est en contact avec la surface, l'index de la main dominante contrôle la rotation de la caméra, le majeur contrôle le niveau de zoom et l'index + majeur déplacent la caméra. Dans les deux exemples précédents, l'utilisation de la main non-dominante pour modifier le mode dans lequel interagit la main dominante est un exemple simple de l'étape de sélection dans notre schéma en trois étapes (*cf.* Section 3.2.2).

Les techniques d'interaction jusqu'ici proposées utilisant l'identification des doigts se sont concentrées sur l'utilisation de doigts seuls ou d'accords pour déclencher des commandes ou manipuler des outils. Ils ont démontré que l'utilisation de cette nouvelle information pour atteindre une fusion entre spécification de commandes et manipulation directe est possible, au moins pour l'utilisation des doigts de manière indépendante. Avec leurs accords de la main-dominante, Marquardt *et al.* entrouvrent même la porte pour généraliser cette fusion aux accords. Néanmoins, une difficulté persiste : les méthodes précédentes de découverte comme *Arpège* ou l'utilisation des jointures ne sont pas compatibles avec le duo sélection de commandes / manipulation directe.

Cette synthèse des travaux met en lumière le potentiel des techniques qui utilisent l'identification et les accords. Mais surtout, les limitations qui, si nous n'en sommes pas conscients, viennent rapidement amoindrir ce potentiel (*ex.* ne pas tirer partie du grand nombre d'accords disponibles, ambiguïté du point d'interaction, *etc.*). Dans la suite, nous listons et justifions les différentes règles que nous nous sommes fixées pour la conception d'*Adoiraccourcix*, et ce afin d'exploiter pleinement l'identification des doigts.

3.2 *Adoiraccourcix*

Adoiraccourcix est une technique d'interaction à manipulation directe qui utilise l'identification des doigts sur surfaces multi-points. L'objectif principal est à la fois de rendre la sélection de commandes efficace en termes de temps et d'espace, surtout quand les commandes peuvent être couplées avec du contrôle continu, et aussi de faciliter la découverte des différentes commandes. L'idée au centre d'*Adoiraccourcix* est d'utiliser l'identification des doigts pour transférer sur les écrans multi-points le mécanisme expert d'invocation de commandes des *modificateurs* et des raccourcis clavier conventionnels.

k	1	2	3	4	5	6	7	8	9	10	Total
$C(10, k)$	10	45	120	210	252	210	120	45	10	1	1023
$C(5, k)$	5	10	10	5	1						31
$C(4, k)$	4	6	4	1							15
$C(3, k)$	3	3	1								7
$C(2, k)$	2	1									3

TABLE 3.1: Nombre de k -combinaisons de n doigts pour différentes valeurs de n .

3.2.1 Espace des entrées qui utilisent l'identification des doigts

Avant d'entrer dans les détails de la technique d'interaction, nous listons l'espace des entrées qui utilisent l'identification des doigts. L'application la plus évidente de l'identification des doigts est de faire correspondre une commande à chaque doigt [MKL⁺11, BSMT09, SK98]. Avec seulement 5 doigts par main, l'utilisation d'un seul doigt à la fois limite fortement l'espace des entrées possibles. L'utilisation d'accords augmente les possibilités de manière significative. Par la suite, nous ferons référence aux accords multi-doigts ainsi qu'aux accords à un doigt comme étant l'espace combiné des *configurations de contacts*, ou plus simplement *configurations*.

Avec l'identification des doigts, tout accord de k doigts parmi 10 peut théoriquement se faire de $C(10, k)$ façons. Cela signifie aussi que $C(10, k)$ interactions à k doigts peuvent être sélectionnées en un geste simple³. Le nombre total d'accords disponibles avec 10 doigts est $\sum_{k=2}^{10} C(10, k) = 1023$. La première ligne de la table 3.1 montre la distribution de ce nombre en fonction de k . L'aspect temporel de la formation de l'accord peut aussi être pris en compte pour différencier les accords. En musique, un *arpège* est un accord dont les notes ne sont pas jouées simultanément. En anglais, un *arpège* est aussi parfois appelé *broken chord* (littéralement *accord cassé*). Permuter l'ordre des doigts de ces *accords cassés* augmenterait le nombre total d'accords disponibles avec 10 doigts à $\sum_{k=2}^{10} P(10, k)$, presque 10 millions⁴. En outre, toutes les configurations peuvent être complétées par des techniques déjà existantes comme les appuis longs, les menus contextuels, les gestes, la pression et ainsi de suite.

Cette limite est cependant purement théorique. En effet, le nombre des accords physiquement réalisables et le nombre d'associations commande-accord mémorisables est probablement bien moindre. Il a été montré que les utilisateurs sont capables de réaliser jusqu'à 480 accords de manière confortable [GHB⁺13].

3. $C(n, k)$ est le nombre de k -combinaisons, le nombre de sous-ensembles distincts de taille k dans un ensemble à n éléments. Il égal au coefficient binomial $\binom{n}{k}$.

4. $P(n, k)$ est le nombre de k -permutations, le nombre de permutations distinctes de taille k dans un ensemble à n éléments. Il égal à $\frac{n!}{(n-k)!}$.

Wagner *et al.* ont étudié comment aider les utilisateurs à mieux retenir les associations commandes-accords en améliorant la relation entre groupement de commandes et groupement d'accords [WLS14]. Pour 9 accords, ils ont montré que des associations thématiques sont mieux assimilées que des associations choisies au hasard.

Afin d'exploiter le potentiel de l'espace des entrées des interfaces à manipulation directe, l'utilisation de l'identification des doigts doit respecter un certain nombre de règles raisonnables qui prennent en compte des limitations physiques et cognitives, le tout organisé et unifié sous un même paradigme.

3.2.2 Une technique d'interaction en trois étapes

Adoiraccourcix est une technique d'interaction en trois étapes inspirée des raccourcis clavier. L'utilisation des raccourcis clavier pour activer des modes outils se passe en trois étapes :

1. la main non-dominante spécifie une commande ou active un mode outil à l'aide d'une touche (*ex.* R) ou d'une combinaison de touches (*ex.* Ctrl+R) ;
2. la main dominante manipule les paramètres de la commande ou de l'outil à l'aide d'un dispositif de pointage tel que la souris ;
3. la main non-dominante est de nouveau utilisée pour modifier l'interprétation des paramètres manipulés (*ex.* ajout de contraintes) en appuyant sur des modificateurs comme Ctrl, Alt, et Shift.

Le schéma d'interaction en trois étapes d'*Adoiraccourcix* diffère pour l'étape 1 : un ensemble d'associations commande-doigt est sélectionné, et dans l'étape 2 nous fusionnons l'appel d'une commande et la manipulation de ses paramètres. Notre étape 3 est similaire à celle des raccourcis clavier. Il est important de noter que les raccourcis clavier introduisent une séparation spatiale entre l'appel de la commande et la manipulation de ses paramètres au cours des étapes 1 et 2. Nous cherchons à éviter cette séparation indésirable. Par formalisme, nous définissons trois ensembles de configurations de doigts (*sélecteurs*, *déclencheurs*, *modificateurs*), un pour chaque étape :

1. une configuration ⁵ *sélecteur* sélectionne un groupe d'associations commande-doigt ;
2. une configuration *déclencheur* déclenche une commande du groupe d'associations commande-doigt actif et commence la manipulation directe de ses paramètres quand cela s'applique ;
3. une configuration *modificateur* peut être utilisée pour contraindre la manipulation en cours des paramètres.

L'ensemble qui regroupe toutes les configurations *sélecteur* possible est noté *sélecteurs*. De même, nous définissons les ensembles *déclencheurs* et *modificateurs*.

⁵ Pour rappel, nous faisons ici référence aux accords multi-doigts ainsi qu'aux accords à un doigt.

Reprenons la figure 3.1 pour illustrer ces trois étapes, avec la terminologie précédemment introduite, sur un grand écran tactile et avec un utilisateur droitier. La figure 3.1-a correspond à l'étape 1. L'accord pouce gauche et index gauche est une configuration *sélecteur* qui sélectionne un groupe d'associations commande-doigt. Grâce à *Lilanotix*, l'utilisateur peut voir le groupe d'associations commande-doigt sélectionné. Dans l'exemple, le groupe actif permet d'accéder aux différents modes de dessin comme dessiner des lignes, des ellipses, des polygones, *etc.*. Chaque commande, *i.e.* un des différents modes de dessin, est associée à un doigt de la main droite. Les cinq doigts de la main droite forment donc l'ensemble des *déclencheurs* disponibles. La figure 3.1-b correspond à l'étape 2. Le majeur droit est la configuration *déclencheur* qui simultanément active le mode *démarrer le dessin d'ellipse* et commence la manipulation directe de la géométrie de l'ellipse (la largeur et la hauteur étant les paramètres de la commande). À noter que la configuration *sélecteur* peut être librement enlevée sans conséquence dès que la commande a été sélectionnée avec succès : dans la figure 3.1-c, la main gauche est enlevée alors que l'ellipse continue à être dessinée. La figure 3.1-d correspond à l'étape 3. Le majeur gauche est la configuration *modificateur* qui permet de contraindre la création toujours en cours de l'ellipse : il contraint la création de l'ellipse à s'effectuer depuis son centre plutôt que de son coin supérieur gauche.

3.2.3 Justification de la conception, variations, et analyse

Dans l'exemple précédent, les ensembles de *sélecteurs* et de *modificateurs* sont tout naturellement associés à la main non-dominante et l'ensemble des *déclencheurs* à la main dominante. Pour de plus petits écrans tactiles ou dans un contexte d'utilisation uni-manuel, cette séparation devient moins stricte. Dans les sections 3.3.2 et 3.3.3, nous discutons la constitution et l'utilisation des ensembles *sélecteurs* et *déclencheurs* en fonction des caractéristiques de l'écran et du contexte d'utilisation. En pratique, nous recommandons que l'ensemble des *modificateurs* soit un sous-ensemble de celui des *sélecteurs* et que les ensembles *sélecteurs* et *déclencheurs* soient disjoints.

Pour mieux guider la conception d'*Adoiraccourcix*, nous avons identifié une série de points importants à prendre en compte (par ordre d'importance) :

- *Découverte et Apprentissage* : l'utilisateur doit pouvoir découvrir les groupes d'associations commande-doigt à l'aide d'un seul et même mécanisme, un *feedback** instantané est de même nécessaire pour permettre à l'utilisateur de s'améliorer.
- *Ergonomie de la main* : les limitations physiques des différents doigts, poignets, et jointures des bras doivent limiter la liste des accords, postures et gestes uni-manuels et bi-manuels possibles.
- *Ergonomie de l'appareil* : le nombre de mains, doigts et accords utilisables sur un écran doit dépendre des caractéristiques physiques de l'appareil comme sa taille (*ex. smartphone**, tablette, grand écran) et du contexte d'utilisation (*ex. tablette tenue, tablette à plat sur un bureau, etc.*).

- *Rétrocompatibilité* : les associations commande-accord usuelles des interactions déjà existantes telles que le *pinch-and-expand** et le Rotate-Scale-Translate (*RST*)* doivent être préservées (en suivant les observations tirées de l'étude décrite à la section 2.2.3).
- *Cohérence* : les nouvelles techniques d'interaction qui utilisent identifications des doigts et accords doivent être cohérentes entre les différents écrans tactiles et les applications.
- *Ambiguïté temporelle* : les techniques doivent être robustes aux contacts non simultanés sur la surface tactile lors de l'exécution d'accords à plusieurs doigts.
- *Ambiguïté spatiale* : dans le cas d'une manipulation à l'aide d'un contact unique (comme le dessin), les techniques se doivent de désambigüiser la caractéristique géométrique d'un accord à plusieurs doigts qui servira à cette manipulation.
- *Personnalisation* : l'utilisateur doit pouvoir personnaliser ses associations commande-accord afin de s'adapter à ses capacités physiques et cognitives ainsi qu'à celles de l'écran utilisé.

Au travers de cette liste, nous pouvons analyser et justifier les trois étapes.

- Étape 1, *Sélection d'un groupe d'associations commande-doigt* : L'utilisateur commence par sélectionner un groupe d'associations commande-doigt en effectuant une configuration *sélecteur*. Si le nombre de *sélecteurs* est petit dû à des compromis avec l'ergonomie de l'écran, il est possible d'utiliser d'autres stratégies comme la direction des mouvements. Ce qui est important, c'est qu'une configuration *sélecteur* doit toujours se comporter comme un quasi-mode. De cette manière, l'ordre et le timing avec lesquels un accord de *sélecteur* est exécuté n'importe pas, accélérant ainsi la reconnaissance des accords puis-qu'aucune temporisation n'est nécessaire pour résoudre l'*ambiguïté temporelle*. À noter que dans la plupart des applications, la configuration *sélecteur* nulle (pas de doigt) active un groupe d'associations commande-doigt par défaut.
- Étape 2, *Invocation de commande et manipulation de paramètres* : Après avoir sélectionné un groupe d'associations commande-doigt, l'utilisateur invoque la commande souhaitée au moyen d'une configuration *déclencheur*. Une fois la commande déclenchée, la configuration *sélecteur* n'a plus besoin d'être maintenue, ainsi les doigts formant cette configuration *sélecteur* peuvent être retirés de la surface. Cette séparation entre la sélection du groupe de commandes et l'invocation d'une commande réduit la crainte de déclencher accidentellement une opération involontaire, et les rectifications plus ou moins coûteuses qui y sont associées. De plus, ces deux premières étapes offre l'opportunité d'intégrer un système de *feedforward** et de *feedback** qui favorisent la *découverte et l'apprentissage* des configurations *déclencheur*. La fiche active de *Lilanotix* est affichée proche des doigts qui exécutent la configuration *sélecteur*. Elle permet de visualiser

les commandes associées aux configurations *déclencheur*. Une fois la commande initiée, les mouvements des doigts de la configuration *déclencheur* contrôlent les paramètres, procurant à l'utilisateur les bénéfices habituels de la fusion entre la sélection de commandes et le contrôle de paramètres avec manipulation directe [GMW05].

- Étape 3, *Manipulation de paramètres et contraintes* : Pendant la manipulation de paramètres, et dès lors que la configuration *sélecteur* a été enlevée après avoir invoqué la commande, l'utilisateur a la possibilité de contraindre leur contrôle à l'aide d'une configuration *modificateur*. Ces contraintes peuvent servir à interpréter la manipulation différemment. Comme dans l'exemple précédent où les rectangles sont dessinés depuis leur centre plutôt que depuis l'un de leurs coins, ou encore aimanter le point d'interaction sur une grille, contraindre le ratio, *etc.*. La plupart du temps, le nombre de contraintes disponibles est assez faible pour que chacune puisse être associée à un doigt seul, ce qui permet d'appliquer plusieurs contraintes en jouant l'accord *ad hoc*. L'utilisateur peut par exemple contraindre le ratio du rectangle tout en aimantant ses coins sur une grille.

3.2.3.1 Intégration de *feedback* et de *feedforward*

Pour les experts, ces trois étapes construisant progressivement un accord peuvent être effectuées plus rapidement [Bux86]. Pour les débutants, l'intégration de *feedback** et de *feedforward** offrent un mécanisme qui aide à la découverte et l'apprentissage des interactions au fur et à mesure qu'ils gagnent en expertise. Les différents groupes d'associations commande-doigt sont affichés *via* les fiches de *Lilantix* dès que la configuration *sélecteur* correspondante est effectuée sans nécessiter qu'une configuration *déclencheur* soit jouée. *Lilantix* n'a pour but que d'afficher des informations, aucun bouton n'est présent sur les fiches. De plus, ces fiches ont un double enjeu : apporter un *feedback** à l'utilisateur sur la configuration *sélecteur* qui vient d'être jouée et fournir du *feedforward** sur les configurations *déclencheurs* qui sont actuellement disponibles. De même, lors de la manipulation de paramètres, *Lilantix* s'affiche lorsqu'une configuration *modificateur* est posée sur la surface afin d'apporter un *feedback** sur les contraintes jouées par l'utilisateur.

3.2.3.2 Nombre de commandes disponibles

Le nombre de commandes pouvant être déclenchées en utilisant cette approche dépend du nombre N_{sel} de configurations *sélecteur* et du nombre N_{dec} de configurations *déclencheur*.

$$\begin{array}{ll}
 N_{dec} & + \quad N_{sel} \times N_{dec} \\
 \text{associations} & \text{associations} \\
 \text{commande-doigt} & \text{commande-doigt} \\
 \text{par défaut} & \text{alternatives}
 \end{array}$$

La partie basse de la table 3.1 présente le nombre de k -combinaisons disponibles pour 5 à 2 doigts. Jusqu'à 31 accords peuvent théoriquement être joués en utilisant les 5 doigts d'une main, cependant cette limite tombe à 15, 7 puis 3 en n'utilisant qu'une sélection fixe de 4, 3 ou 2 doigts. En utilisant la main non-dominante pour l'ensemble de *sélecteurs* et la main non-dominante pour l'ensemble de *déclencheurs*, le nombre théorique maximal de commandes qui peuvent être sélectionnées avec notre mécanisme est de $31 + 32 \times 31 = 1023$. Le choix des configurations *sélecteurs* et *déclencheurs* est donc crucial dans l'optique de réduire et d'organiser cet espace d'interaction. Il peut être préférable de vouloir un grand nombre de configurations *sélecteur* et peu de configurations *déclencheur*, ou l'inverse, ou bien encore préférer un compromis. Par exemple, en utilisant des combinaisons de 3 doigts ou moins parmi une population de 4 doigts de la main non-dominante et des combinaisons à un doigt sur la main dominante, il est possible de sélectionner $((4 + 6 + 4) + 1) \times 5 = 75$ commandes, ce qui est suffisant pour la plupart des applications⁶.

Même si les exemples précédents partent tous du principe que les configurations *sélecteurs* sont effectuées avec la main non-dominante et les *déclencheurs* avec la main dominante, cette répartition n'est en aucun cas imposée par notre technique. Dans la majorité des cas, le choix des ensembles *sélecteurs* et *déclencheurs* peut être fait de manière *ad hoc* en fonction des applications, du contexte d'utilisation ou de l'écran utilisé. Alors que les applications sur *smartphone** et tablette seraient plus enclines à favoriser une symétrie entre les deux mains et ainsi supporter une utilisation ambidextre, les applications sur grandes surfaces tactiles pourraient encourager une interaction bi-manuelle asymétrique (*ergonomie de l'appareil*).

Les commandes avec paramètres telles que le dessin de formes ou l'ajustement de leur taille sont plus adaptées pour des configurations *déclencheurs* à un doigt seul (*ex.* en utilisant l'index ou le majeur). À l'inverse, les commandes sans paramètres telles que *Enregistrer* sont plus adaptées pour des accords à plusieurs doigts (*ex.* pouce + index + majeur). Les configurations *déclencheurs* à un doigt seul sont intrinsèquement adaptées aux commandes ayant jusqu'à deux paramètres, *ex.* ajuster la position d'un des coins d'un objet (*ambiguïté spatiale*). Le pouce, l'index et le majeur de la main dominante doivent être préférés (*cf.* Section 2.1.8) quand une manipulation précise est requise (*ergonomie de la main*).

3.2.3.3 Supporter la transition d'utilisateur novice à expert

La visualisation de *Lilanotix* permet aux utilisateurs novices de se rappeler les configurations *déclencheurs* disponibles et les commandes qui leur sont associées sans risquer d'invoquer une commande accidentellement (*cf.* Figure 3.1-a). Pour supporter la transition d'utilisateur novice à expert, nous utilisons le principe de répétition [CGSM14] : utiliser les mêmes actions quelque soit le niveau d'expertise. À l'instar des *marking menus* [KB91], au fur et à mesure que les utilisateurs

6. Malacria *et al.* [MBH⁺13] ont trouvé entre 15 et 112 raccourcis claviers en étudiant 30 des applications les plus utilisées sous Mac OSX ($\mu = 55, 31, \sigma = 20, 53$)

novices se remémorent quelle configuration *sélecteur* et *déclencheur* déclenchent quelle commande, ils peuvent exécuter ces configurations rapidement au cours d'un même mouvement physique. Pour encourager un peu plus cette transition de novice à expert, *Lilanotix* peut-être affichée après un court délai.

Étant donné que le mécanisme d'*Adoiraccourcix* est inspiré des raccourcis clavier, nous nous attendons à ce que les utilisateurs comprennent les principes sous-jacents de la technique grâce à leur connaissance du fonctionnement des raccourcis clavier. La séparation des ensembles *sélecteurs* et *déclencheurs* permet la création de groupes de commandes similaires à ceux que l'on trouve dans les menus ou les barres d'outils. De plus, Wagner *et al.* [WLS14] ont montré que cette organisation par similarité sémantique aide les utilisateurs à mieux se rappeler les associations commande-accord.

Afin de découvrir et d'explorer les commandes pour lesquelles aucun *feed-forward** n'est disponible (comme celles associées à la configuration *sélecteur* nulle), un mode d'aide peut être explicitement invoqué en touchant la surface avec tous les doigts qui composent les ensembles *sélecteurs* et *déclencheurs*. Cette configuration d'aide spéciale est facilement mémorisable et s'apparente métaphoriquement à demander au système d'afficher l'ensemble des associations commande-accord. Une fois entré dans ce mode d'aide, l'application offre une vue d'ensemble interactive sur les configurations *sélecteurs*, *déclencheurs* et *modificateurs* disponibles ainsi que leurs relations.

3.2.3.4 Compatibilité avec les techniques existantes

L'identification des doigts est *retrocompatible* avec la plupart des techniques multi-points déjà existantes telles que les *tap**, les doubles *tap**, les gestes (*ex. flick**), les appuis longs et la pression. Cette caractéristique est due au fait que l'identification des doigts est orthogonale à ces techniques : cette information peut être utilisée en plus ou en combinaison de ces techniques. Par exemple, une configuration *déclencheur* à un doigt seul peut être surchargée afin qu'une commande différente soit déclenchée si ce doigt effectue un *tap** ou un *flick**. Il est possible d'imaginer combiner des configurations à doigt seul avec des *flick** directionnels pour accéder à plus de commandes, ou différencier les contacts selon la pression, *etc.* En pratique, il est préférable que ce type de surcharge soit utilisé pour des commandes sémantiquement liées. Par exemple, un *flick** du pouce dans une direction pour déclencher la commande *annuler* et un *flick** du pouce dans la direction opposée pour *refaire*.

Il est souvent bien commode d'associer une commande au nombre de doigts en contact simultanés sur l'écran, sans se préoccuper de leur identification. Par exemple, la rotation et le changement d'échelle d'un objet peut s'effectuer avec n'importe quel accord à deux doigts sur la plupart des interfaces multi-points. Associer la même commande à différentes configurations de *déclencheurs* est bien sûr possible en utilisant l'identification des doigts. Pour autant, associer toutes les configurations à *n* doigts à une unique commande restreint drastiquement les avantages de l'identification des doigts. En pratique, il est possible d'atteindre une *retrocompatibilité* raisonnable avec les techniques qui se basent sur le nombre

de doigts simultanément en contact en identifiant les combinaisons de doigts les plus utilisées pour déclencher une commande donnée. Il suffit par la suite d'associer ces combinaisons au déclenchement de cette commande. Par exemple, le pouce et l'index sont souvent utilisés pour la rotation et le changement d'échelle de petits objets [VC12]. De même, l'utilisation des deux index est assez courante sur grandes surfaces pour la rotation et le changement d'échelle de grands objets (*ergonomie de l'appareil*).

3.2.3.5 Configuration et personnalisation

Les choix d'associations entre les configurations de doigts et les commandes sont conséquents. La manière dont ces associations sont effectuées ne dépend pas uniquement de l'application : la taille de l'appareil et son contexte d'utilisation sont des facteurs importants à prendre en compte (*ergonomie de l'appareil*). Par exemple, une application d'édition de photo sur tablette a, la plupart du temps, un nombre de commandes plus petit que son pendant sur ordinateur de bureau. De plus, les utilisateurs auront généralement tendance à interagir d'une seule main avec les applications sur tablette, la seconde étant utilisée pour tenir la tablette [WHM12]. Pour ce type de scénarios, les configurations *sélecteurs*, *déclencheurs*, et *modificateurs* peuvent être choisies de telle sorte que tout soit associé sur une seule main et refléter ces associations sur l'autre main de manière symétrique. Avec cette symétrie, peu importe la main choisie pour interagir, l'utilisation restera la même. Dans une application plus complexe comme un éditeur de photos sur grandes surfaces, où le nombre de commandes accessibles est élevé et les deux mains sont disponibles, l'idéal serait d'opter pour une asymétrie des ensembles sur les mains : les ensembles *sélecteurs* et *modificateurs* sur la main non-dominante et l'ensemble *déclencheurs* sur la main dominante. Dans ce scénario, l'interface doit fournir à l'utilisateur un moyen de configurer le choix de sa main dominante (*personnalisation*).

Permettre la personnalisation des associations est une autre façon d'aider l'utilisateur à développer un comportement expert. Les utilisateurs pourraient définir leur ensembles *sélecteurs*, *déclencheurs* et *modificateurs*, les accords contenus dans ces ensembles, ainsi que les associations avec les commandes et paramètres. Les utilisateurs pourraient par exemple tirer partie de cette souplesse en associant les commandes qu'ils utilisent le plus à certains doigts (*personnalisation*). Nous nous attendons à ce que la personnalisation soit particulièrement importante dans le cas où les trois ensembles contiennent un faible nombre de doigts. Elle permettrait ainsi d'optimiser l'utilisation de l'espace limité des entrées.

Après avoir décrit le fonctionnement théorique d'*Adoiraccourcix*, nous nous proposons maintenant de fournir des exemples concrets qui implémentent la technique. Au travers de ces exemples, nous essayons de montrer les différentes stratégies qui peuvent être adoptées en fonction du facteur de forme (*ex. smart-phone**, tablette, grand écran) ou des propriétés de l'application (*ex. nombre de commandes*).

3.3 Exemples de mises en œuvre

Pour implémenter *Adoiraccourcix*, la tâche des concepteurs consiste à définir les trois ensembles et les associations commande-accord en prenant en compte l'ergonomie de l'appareil, le contexte d'utilisation et le type d'application en suivant les recommandations énoncées précédemment. Dans la suite, nous démontrons comment *Adoiraccourcix* peut être instancié au travers d'applications sur grands écrans, tablettes et *smartphones*^{*}. Une vidéo de démonstration illustre *Adoiraccourcix* sur ces trois facteurs de forme⁷.

Le dispositif utilisé pour la mise en œuvre sur grands écrans est celui décrit à la section 1.2.2. Sur tablette et *smartphone*^{*}, le dispositif utilisé est celui décrit à la section 1.2.3.

3.3.1 Sur grands écrans

Notre application principale qui implémente le système *Adoiraccourcix* est une application de dessin vectoriel sur grandes surfaces. Pour la création de dessins vectoriels, les utilisateurs doivent souvent effectuer des allers-retours entre les différents modes de dessin, comme la création de formes, les transformations géométriques, l'édition de courbes *etc.* Cette application sert d'exemple canonique dans un contexte où l'interaction bi-manuelle est possible et naturelle et où le potentiel d'*Adoiraccourcix* peut être pleinement exploré.

Adoiraccourcix au service d'une application de dessin vectoriel

Étant donné la taille du grand écran tactile utilisé, nous associons les doigts *sélecteurs* à ceux de la main non-dominante et les doigts *déclencheurs* à ceux de la main dominante. Il y a 7 groupes d'associations commande-accord différents, chacun associé à un accord de la main non-dominante (*cf.* Figure 3.2-1a-h et première colonne de la table 3.2). Les accords qui activent ces groupes d'associations sont tous une combinaison de trois doigts de la main non-dominante : nd-pouce, nd-index, nd-majeur, nd-pouce + nd-index, nd-pouce + nd-majeur, nd-index + nd-majeur, et nd-pouce + nd-index + nd-majeur. Un *groupe d'associations commande-accord par défaut* est actif pour la configuration de *sélecteur* nulle (*i.e.* aucun *sélecteur* en contact). Par choix, le groupe d'associations par défaut et le groupe activé par l'accord nd-pouce sont les mêmes.

Chaque groupe d'associations commande-accord comprend au plus 5 commandes. De cette façon, chaque commande est déclenchée à l'aide d'un doigt *déclencheur* seul de la main dominante. Comme indiqué précédemment, jusqu'à $7 \times 5 = 35$ commandes peuvent être offertes ($8 \times 5 = 40$ en incluant la configuration nulle des *sélecteurs*). Cependant pour des raisons de rétrocompatibilité et d'ergonomie, le système ne fournit que 29 commandes (*cf.* Table 3.2). Par exemple, le *sélecteur* nd-index sélectionne le groupe de création de formes comprenant 4 commandes (*cf.* Figure 3.2-1b), l'index déclenche la création de courbes

7. Lien de la vidéo de démonstration : <http://www.alixgoguy.fr/phd/adoiraccourcix-en.mp4>

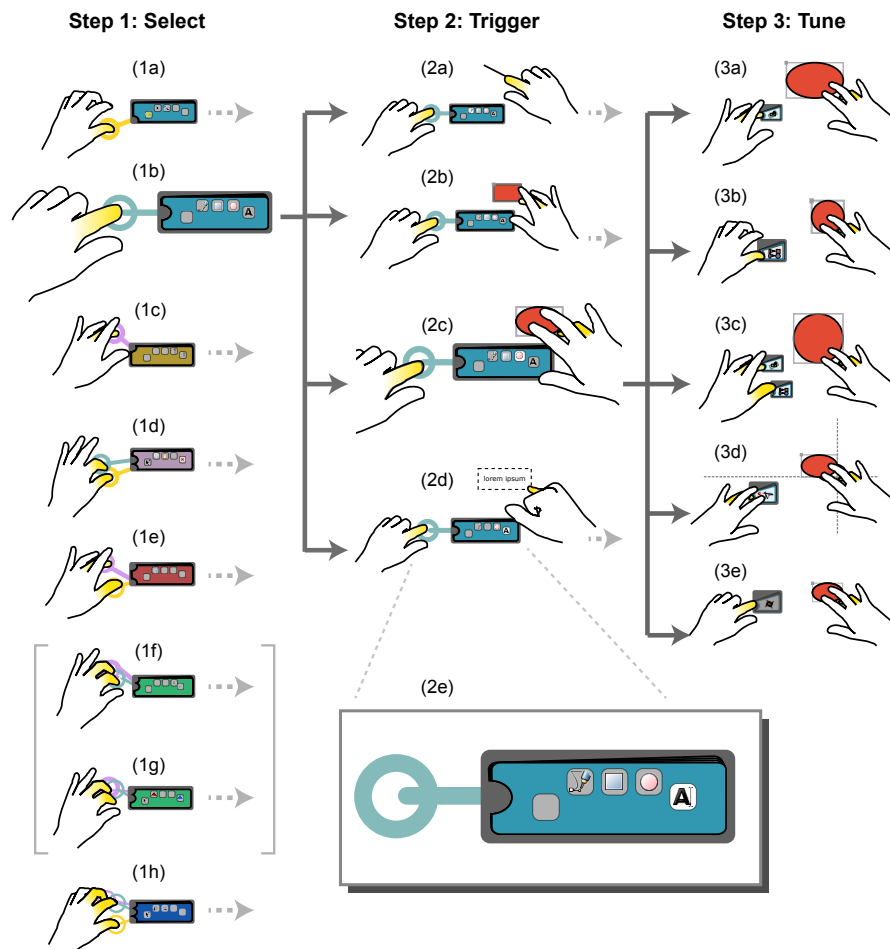


FIGURE 3.2: Description de la figure à la page suivante.

FIGURE 3.2: *Adoiraccourcix* au service d'une application de dessin vectoriel. À l'étape 1, l'utilisateur effectue une configuration *sélecteur* pour sélectionner un groupe d'associations accords-commandes : (1a) la configuration *sélecteur* nd-pouce (groupe *Annuler/Refaire, Sélection, Edition de points*) ; (1b) la configuration *sélecteur* nd-index (groupe *Courbe, Rectangle, Ellipse, Zone de Texte*) ; (1c-h) autres nd configurations *sélecteur* (cf. Table 3.2). À l'étape 2, l'utilisateur déclenche une commande et manipule ces paramètres en utilisant une configuration *déclencheur* et en maintenant la configuration *sélecteur* : avec la configuration *sélecteur* nd-index, l'utilisateur dessine (2a) une courbe avec l'index ; (2b) un rectangle avec le majeur ; (2c) une ellipse avec l'annulaire ; (2d) une zone de texte avec l'auriculaire. Lorsqu'une commande est déclenchée, l'icône de la fiche *Lilantix* correspondante est surlignée (2e). À l'étape 3, l'utilisateur peut optionnellement contraindre les paramètres de la commande grâce à une configuration *modificateur* : lorsqu'il dessine une ellipse, l'utilisateur peut (3a) la dessiner depuis son centre ; (3b) la contraindre en cercle ; (3c) ou encore combiner les deux contraintes précédentes pour dessiner un cercle depuis son centre ; (3d) dessiner l'ellipse avec le magnétisme activé. Lorsqu'une contrainte n'est pas disponible la fiche *Lilantix* correspondante est grisée : la contrainte désactivant le changement d'échelle est disponible lors d'un *RST**, elle ne l'est pas lors des phases de dessin (3e).

Accord	pouce	index	majeur	annulaire	auriculaire
○○○○○ ○○○○●	Annuler/Refaire	Sélection	Modification		
○○○●○		Courbe	Rectangle	Ellipse	Texte
○○●○○	S+	À l'avant	Vers l'avant	Vers l'arrière	À l'arrière
○○○●●	S+	Copier	Couper	Coller	Effacer
○○●○●	S+	Grouper	Dégrouper		
○○●●○	S+	←—— commandes contextuelles ———→			
○○●●●	S+	Vsym	Hsym		

TABLE 3.2: Groupes d'associations commande-accord. Chaque ligne de la table représente un groupe d'associations commandes-accords. L'accord effectué avec la main non-dominante est indiqué dans la colonne de gauche où les cercles représentent de gauche à droite les doigts de l'auriculaire au pouce. ● et ○ représentent respectivement des doigts en contact ou pas. Les commandes associées aux doigts de la main dominante sont listées dans les colonnes suivantes où S+ représente la multi-sélection.

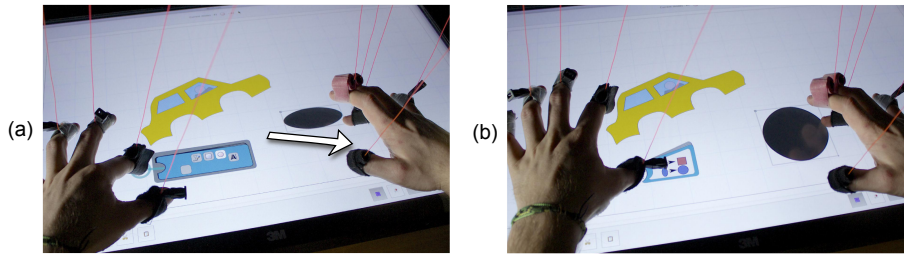


FIGURE 3.3: Exemple d'interaction sur grande surface avec *Adoiraccourcix* : (a) l'accord nd-pouce+nd-index active un groupe d'associations commande-accord (étape 1), puis l'annulaire déclenche la création d'une ellipse (étape 2) ; (b) l'accord nd-pouce est ensuite utilisé pour contraindre le ratio de l'ellipse (étape 3).

(cf. Figure 3.2-2a), le majeur la création de rectangles (cf. Figure 3.2-2b), l'annulaire la création d'ellipses (cf. Figure 3.2-2c), et l'auriculaire la création de zone de texte (cf. Figure 3.2-2d). Le pouce ne déclenche aucune commande dans ce groupe d'associations commande-accord. En suivant un schéma similaire pour les 6 autres configurations *sélecteur*, 25 autres commandes sont accessibles à l'étape 2.

La figure 3.2-2e montre une vue détaillée d'une fiche de *Lilanotix* : du *feedforward** correspondant aux accords *déclencheurs* disponibles est affiché à l'aide des 5 icônes qui représentent les commandes associées ; une absence d'icône indique les doigts *déclencheurs* pour lesquels aucune commande n'est associée, comme ici le pouce ; les icônes sont légèrement décalées rappelant les différences naturelles de longueur entre les doigts. *Lilanotix* est affichée après un délai de 0,33s (valeur similaire à celle utilisée par les *marking menus* [KB94]). Dès qu'une commande est déclenchée par un doigt *déclencheur*, un *feedback** visuel est affiché sur *Lilanotix* (ex. fond blanc derrière l'icône correspondant à la commande de création d'une zone de texte, cf. Figure 3.2-2e).

Lors de la manipulation des paramètres d'une commande, l'utilisateur peut retirer ses *sélecteurs* de l'écran pour ensuite utiliser les *modificateurs* pour modifier l'interprétation de la commande. Par exemple, durant la création d'une ellipse (cf. Figure 3.2-2c) des modifications possibles seraient de dessiner depuis le centre à l'aide du nd-majeur (cf. Figure 3.2-3a) et contraindre le ratio des ellipses (cf. Figure 3.2-3b). Ces modifications peuvent aussi être combinées en utilisant l'accord *modificateur ad hoc* (ex. Figure 3.2-3c). Lorsqu'une contrainte ne s'applique pas, la fiche correspondante dans *Lilanotix* est grisée (cf. Figure 3.2-3e).

Dans le groupe d'associations par défaut (première ligne de la table 3.2), certaines configurations de la main dominante sont réservées afin de préserver la rétrocompatibilité avec les techniques déjà existantes. Par exemple, les accords pouce+index et index+nd-index sont associés au geste *RST** et l'index est utilisé pour déplacer les formes. Ces accords sont déjà couramment utilisés par

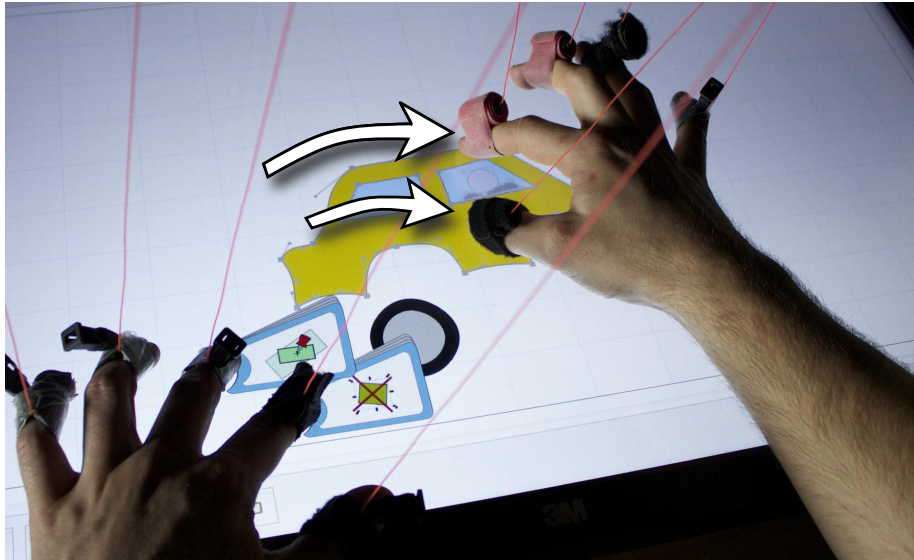


FIGURE 3.4: RST^* , effectué à l'aide de l'accord pouce + index, contraint par le nd-index (désactivation du changement d'échelle) et le nd-majeur (ancrage du centre) n'autorisant que la rotation de l'objet.

les utilisateurs, dans la plupart des applications, pour effectuer ces manipulations standards. Par ailleurs, *Adoiraccourcix* augmente ces gestes classiques. Les *sélecteurs* peuvent ainsi aussi contraindre le RST^* pour ne permettre qu'une rotation avec ou sans centre fixe (cf. Figure 3.4). D'autres commandes fréquemment utilisés, comme la modification des points de contrôle ou annuler/refaire une action, sont associées au majeur et à des *flicks*^{*} directionnels du pouce. À noter que la découverte des dernières commandes est possible grâce au *sélecteur* nd-pouce, qui affiche la fiche de *Lilanotix* correspondante au groupe d'associations commande-accord par défaut (première ligne de la table 3.2).

Pour mieux découvrir et apprendre les associations commande-accord, un mode d'aide explicite est disponible en appuyant sur l'écran avec les dix doigts en même temps (cf. Figure 3.5). Quand celui-ci est actif, une visualisation interactive de toutes les commandes est affichée. Elle permet d'explorer les différentes combinaisons de doigt sans affecter le travail en cours.

3.3.2 Sur tablette

Pour démontrer l'adaptabilité d'*Adoiraccourcix* à d'autres facteurs de forme et applications, nous avons créé une application d'annotation de PDF sur tablette.

En prenant en compte la petite taille des appareils et le type des applications, nous associons tous les *sélecteurs*, les *déclencheurs* et les *modificateurs* aux mêmes doigts des deux mains. Wagner *et al.* [WHM12] ont montré que le pouce

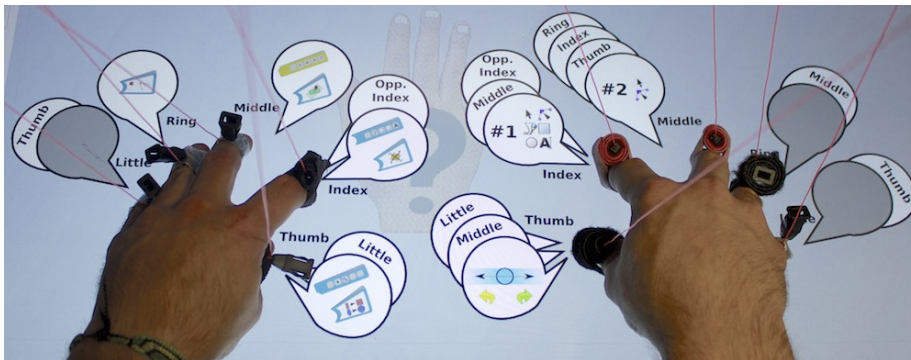


FIGURE 3.5: Mode d'aide explicite accessible en appuyant avec les 10 doigts en même temps.

de la main non-dominante peut-être utilisé pour interagir même lorsque la main non-dominante tient l'appareil. Nous nous appuyons sur ce fait pour utiliser un pouce, sans distinguer la main, comme l'unique configuration *sélecteur* (nous utilisons en plus le mouvement du pouce pour sélectionner différents groupes d'associations commandes-accords, les détails sont expliqués par la suite). Cela signifie que le pouce de la main non-dominante peut être utilisé pour sélectionner un groupe d'associations commandes-accords alors que cette même main tient l'appareil. Pour plus de clarté et puisque les associations commandes-accords sont symétriques par rapport aux mains, les noms des doigts sont donnés sans la spécification de la main.

Application d'annotation de PDF sur tablette

Les applications de lecture de PDF et d'annotations, comme *Adobe Acrobat*, nécessitent un certain nombre de commandes pour naviguer dans le document, annoter avec différents types d'encres, différents tampons et autres commentaires textuels. Notre application possède 18 commandes. 15 d'entre elles sont séparées en quatre groupes d'associations commande-accord (en incluant le groupe par défaut). Les 3 restantes sont accessibles *via* des *flicks**. L'un ou l'autre des deux pouces sélectionne le groupe des commandes associées aux doigts *déclencheurs* (index, majeur, annulaire et auriculaire). Quand aucun pouce n'est en contact, le groupe d'associations commandes-accords par défaut est actif : l'index fait défiler le document (*cf.* Figure 3.7-a), le majeur surligne (*cf.* Figure 3.7-b) et l'annulaire crée une zone de texte. De plus, un accord simultané pouce + index déclenche un *pinch-and-expand**. Les autres groupes sont sélectionnés en glissant un pouce dans l'une des trois directions possibles (*cf.* Figure 3.6-a). Lorsque l'un de ces groupes est sélectionné, *Lilantix* se met à jour et les icônes montrant les quatre commandes associées aux *déclencheurs* index, majeur, annulaire et auriculaire s'affichent autour du pouce (*cf.* Figure 3.6-b). Chacun des trois groupes à son code couleur : (bleu) pour copier, coller, couper et supprimer ; (jaune) pour des

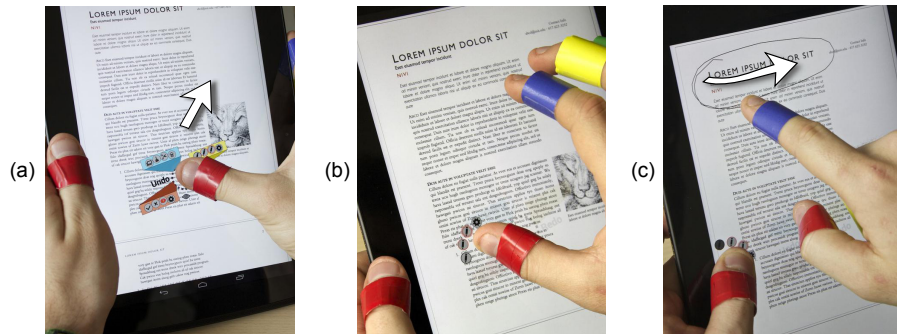


FIGURE 3.6: Le *sélecteur* pouce sur tablette : (a) le *sélecteur* pouce affichant *Lilanolix* avec trois groupes d'associations commandes-accords ; (b) utilisation du groupe jaune après avoir glissé le pouce en haut à droite pour sélectionner un groupe d'associations, les commandes associées aux doigts *déclencheurs* sont affichées autour du pouce ; (c) le *sélecteur* pouce peut aussi être celui de la main non-dominante pour faciliter l'interaction des doigts de la main dominante.

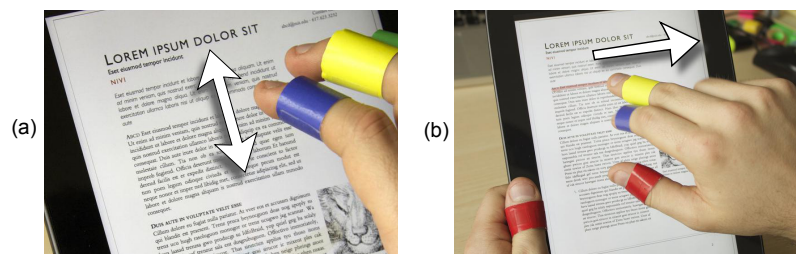


FIGURE 3.7: Exemples d'utilisation des doigts *déclencheurs* du groupe d'associations par défaut : (a) l'index fait défiler le document ; (a) le majeur surligne le document.

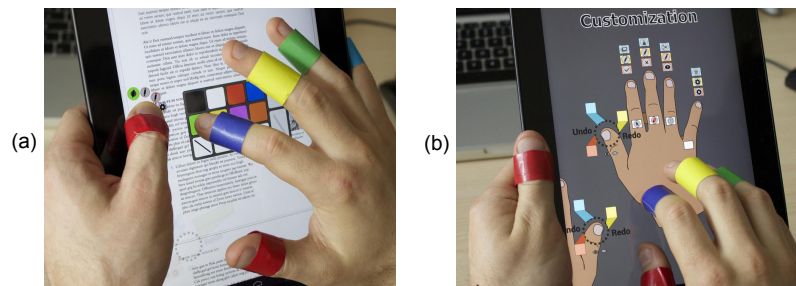


FIGURE 3.8: Autres fonctionnalités : (a) personnalisation du stylo associé à l'index ; (b) un accord à cinq doigts permet d'activer le mode de personnalisation (les commandes peuvent être ré-associées en faisant un glissé déposé de l'icône correspondante).



FIGURE 3.9: *Sélecteur et déclencheurs sur smartphone** : (a) le majeur corrige un mot ; (b) le *sélecteur* pouce affiche *Lilantotix* ; (c) l'index déclenche une *copie*.

stylos personnalisables ; (rouge) pour des tampons personnalisables. Dès qu'un stylo est sélectionné, le pouce devient un *modificateur* contraignant l'encre à suivre les lignes du texte.

Les groupes de stylos et de tampons incluent un mode de personnalisation activé par l'auriculaire. Dans ce mode, la couleur, l'épaisseur, la taille, l'opacité et la forme des trois stylos et trois tampons est personnalisable en appuyant avec le *déclencheur ad hoc* sur des couleurs, épaisseurs, tailles, opacités, et formes prédéfinies (cf. Figure 3.8-a).

Des commandes globales sont également disponibles. Les *flicks** gauche et droit du pouce déclenchent les commandes annuler et refaire. Un *flick** du pouce vers le bas affiche ou cache une barre latérale contenant les aperçus des pages. Ces mouvements du pouce n'entrent pas en conflit avec la sélection des groupes d'associations décrits précédemment. Enfin, un *tap** d'un accord à cinq doigts invoque un mode d'aide explicite. Cet écran décrit visuellement les différentes associations commandes-accords (cf. Figure 3.8-b). Il permet aussi la personnalisation de ces associations : les commandes peuvent être ré-associées en faisant un glissé-déposé d'une icône de *Lilantotix* sur une autre. L'échange des icônes engendre l'échange des commandes associées aux accords.

3.3.3 Sur *smartphone*

De même que sur tablette, nous avons créé une application d'envoi de SMS sur *smartphone** pour démontrer l'adaptabilité d'*Adoiraccourcix*. Les *sélecteurs*, les *déclencheurs* et les *modificateurs* sont les mêmes que pour tablette.

Application d'envoi de SMS sur *smartphone*

Écrire un SMS nécessite des commandes d'entrée de texte, de correction orthographique, pour copier, coller, couper et pour naviguer dans le fil de discussion (i.e. l'historique des messages). Notre application d'écriture de SMS sur *smartphone** utilise, de la même façon que notre application tablette, le pouce comme seul *sélecteur*. Quand le pouce entre en contact avec l'écran, *Lilantotix* s'affiche et indique à l'aide de rectangles adjacents les commandes associées aux quatre doigts *déclencheurs* (cf. Figure 3.9-b). Un *tap** de l'index, du majeur ou

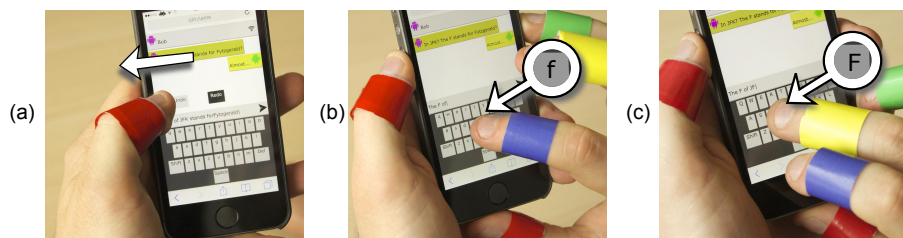


FIGURE 3.10: *Déclencheurs sur smartphone** : (a) un *flick** droit du pouce déclenche la commande annuler ; (b) l'index écrit une lettre en minuscule ; (c) le majeur écrit une lettre en majuscule.

de l'annulaire sur un mot seul déclenche respectivement la commande copier, coller ou couper (cf. Figure 3.9-c). Bien que non implémenté, glissé l'index sur du texte, sans que le pouce touche l'écran, pourrait sélectionner une zone de texte alternative à copier, coller ou couper. Quand le *sélecteur* pouce ne touche pas l'écran un groupe d'associations commandes-accords par défaut est actif. L'index permet de naviguer dans le fil de discussion et effectuer un *tap** du majeur sur un mot affiche une liste suggérant différentes orthographes (cf. Figure 3.9-a). Les commandes annuler et refaire existent et sont de nouveau déclenchées par des *flicks** gauche et droit (cf. Figure 3.10-a). Durant l'écriture, les minuscules sont écrites grâce à l'index (cf. Figure 3.10-b) et les majuscules grâce au majeur. Nous reprenons ici le principe de *DualKey* [GB16] (cf. Figure 3.10-c). L'annulaire peut aussi être utilisé pour accéder à la ponctuation. L'écriture *via* les pouces pourrait être supportée en désambiguïsant l'action voulue (*ex.* annuler, refaire, activer le groupe d'associations ou écrire) grâce à la position (*i.e.* sur le clavier ou non), le geste (*i.e.* *flick** ou *tap**) et les autres doigts impliqués dans un éventuel accord : nous discutons cette limitation dans la section 3.5.5.

Après avoir décrit la mise en œuvre d'*Adoiraccourcix* sur différents facteurs de forme, nous nous penchons sur leur évaluation. Nous proposons de le faire en trois étapes. Tout d'abord, nous souhaitons avoir un retour qualitatif sur l'utilisation de notre technique. Ensuite, nous nous focalisons sur les performances grâce à une étude faite au travers de l'application de dessin vectoriel sur grand écran. Enfin, nous analysons les performances sur tablette et *smartphone**.

3.4 Évaluation et validation

Nous avons mené deux études utilisateurs et une analyse basée sur le *FLM** pour évaluer l'application de dessin vectoriel décrite ci-dessus. L'objectif de la première étude, portant une attention particulière à la découverte et la transition de novice à expert, était principalement qualitative. L'objectif de la seconde étude, comparant au travers de tâches les performances entre *Adoiraccourcix* et une interface graphique traditionnelle, Graphical User Interface (*GUI**), était

quant à elle quantitative. L'analyse basée sur le *FLM** apporte une validation supplémentaire aux résultats de l'expérience quantitative. De plus, elle fournit une méthode d'analyse raisonnable des performances d'*Adoiraccourcix* sur d'autres tâches et facteurs de forme.

3.4.1 Étude qualitative

Nous avons recruté 8 volontaires (20 à 30 ans, 2 femmes, 2 non informaticiens, tous droitiers). Après une courte phase d'introduction (5 minutes) aux applications de dessin vectoriel et à l'identification des doigts, l'expérience s'est déroulée en deux parties.

Dans la première partie, les participants étaient conviés à explorer librement les différentes fonctionnalités durant 20 minutes. Ils étaient incités à dessiner des formes simples et complexes. Durant cette période d'exploration, les participants étaient invités à penser à voix haute afin de partager leurs réactions et autres mécanismes de pensée. Par la suite, l'expérimentateur leur montrait les fonctionnalités non découvertes les préparant ainsi à la deuxième partie de l'expérience.

Dans la deuxième partie, les participants devaient effectuer quatre tâches de manipulation simples et dessiner une scène en partant d'une feuille blanche. Ils étaient libres d'utiliser *Adoiraccourcix* et/ou la *GUT** pour accomplir les différentes tâches. Chacune des quatre tâches de manipulations simples nécessitait un type d'actions particulières : (Tâche A) créer des formes approchant des contours et tailles donnés ; (Tâche B) dupliquer des polygones pour créer un ciel étoilé ; (Tâche C) changer la couleur de formes données ; (Tâche D) manipulation de formes données. Pour chaque tâche, les participants commençaient avec une scène définie (cf. Figure 3.11-haut) et l'expérimentateur leur montrait la scène à reproduire (cf. Figure 3.11-bas). Aucune précision ni durée maximale n'ont été spécifiées. L'ordre des tâches ne variait pas d'un participant à l'autre. La dernière tâche était la plus complexe dans la mesure où elle nécessitait la création d'une scène simple à partir d'une feuille vierge. Les participants devaient dessiner une maison comprenant au moins les éléments suivants : une porte avec poignée, deux fenêtres et un toit triangulaire. Aucune limite de temps n'était imposée pour la deuxième partie. La durée moyenne était de 30 minutes.

Après chaque partie de l'étude, une pause était proposée. À la fin de l'étude, les participants ont complété un questionnaire listant 6 propositions qu'ils devaient noter sur une échelle de Likert à 5 points (1=*Pas du tout d'accord* ; 5=*Tout à fait d'accord*) :

- “L'application ressemble aux logiciels de dessin vectoriel que je connais.”
- “Les raccourcis étaient faciles à apprendre.”
- “J'ai été capable de faire ce que je souhaitais.”
- “Les raccourcis étaient faciles à découvrir.”
- “J'ai préféré utiliser la *GUI** plutôt que les raccourcis proposés.”
- “J'ai eu une meilleure sensation de fluidité de d'interaction avec les raccourcis qu'en utilisant la *GUI**.”

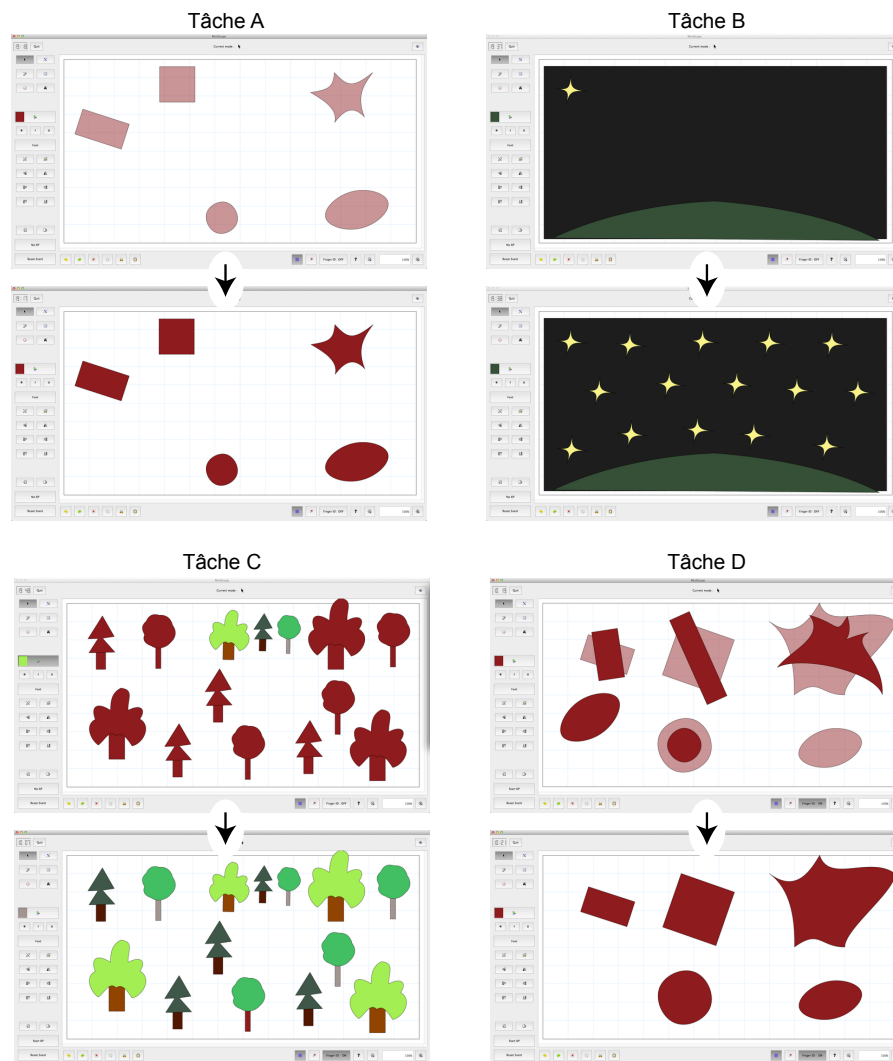


FIGURE 3.11: Tâches de manipulation de la deuxième partie de l'étude qualitative : Tâche A, création de formes; Tâche B, duplication de polygones; Tâche C, changement de couleurs; Tâche D, manipulation de formes. Pour chaque tâche, l'image du haut représente la scène de départ et celle du bas l'objectif à atteindre.

Résultats

Les participants n'ont pas eu de difficulté particulière pour compléter les deux parties de l'étude. Pour la deuxième partie, 1 participant sur 8 n'a utilisé que la *GUP**, 3 participants ont utilisé une combinaison de la *GUP** et d'*Adoiraccourcix* (la majorité des commandes ont cependant été déclenchées en utilisant *Adoiraccourcix*), et 4 participants n'ont utilisé qu'*Adoiraccourcix*. Tous les participants ont découvert *Lilanotix* par eux-mêmes. Cependant, quelques participants ont initialement cherché à presser les icônes de *Lilanotix* comme s'il s'agissait de boutons. L'affordance du *feedforward** visuel actuel pourrait être améliorée en enlevant la bordure des icônes ou en les fondant dans l'arrière-plan. Tous les participants ont utilisé les *modificateurs* pour contraindre le ratio des formes et créer des carrés et des cercles.

Après analyse des réponses au questionnaire, les participants ont confirmé la *similarité* entre notre application et les logiciels de dessins vectoriels qu'ils connaissaient (mode⁸ 5, médiane 5). Ils ont aussi eu l'impression d'avoir réussi à effectuer les opérations qu'ils souhaitaient réaliser avec *Adoiraccourcix* (mode 4, médiane 4,5). Il semblerait donc qu'*Adoiraccourcix* soit suffisamment expressif pour supporter l'ensemble des commandes d'une application relativement complexe. Dans l'ensemble, *Adoiraccourcix* est plutôt facile d'utilisation (mode 4, médiane 4). Les participants nous ont déclaré avoir besoin de plus de temps pour apprendre les différents raccourcis mais tous ont reconnu qu'ils pouvaient s'améliorer avec plus d'entraînement. Ils ont aussi affirmé que la fluidité d'interaction était plus continue avec *Adoiraccourcix* (mode 4, médiane 4).

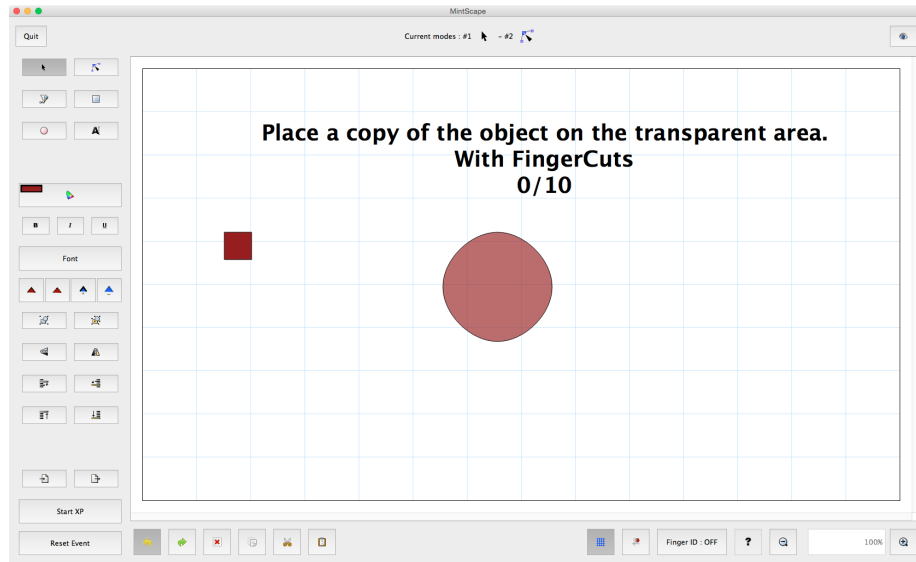
À l'instar des raccourcis clavier, il faut du temps et de l'entraînement pour devenir familier et efficace avec *Adoiraccourcix*. Nous n'étions pas étonnés de voir que les participants n'ont pas complètement réussi à maîtriser *Adoiraccourcix* après seulement une heure d'utilisation, cependant ils nous ont admis qu'ils apprenaient : P7 a commenté "Avec de l'entraînement, je serais plus rapide." Nous étions ravis de voir que les participants puissent découvrir rapidement les commandes, P3 a déclaré "Il y a beaucoup de raccourcis et je ne me rappelle pas de tous, par contre ils sont faciles à retrouver". Nous avons de même vu des signes émergents de transition novice-expert, P1 a indiqué "Les raccourcis que j'ai utilisés m'ont fait gagner du temps".

3.4.2 Étude quantitative

Nous avons recruté 12 volontaires pour la seconde étude (23 à 35 ans, 1 femme, 3 non informaticiens, tous droitiers). Nous avons choisi un plan de recherche expérimental intra-sujet avec pour variables indépendantes *TECHNIQUE* (*Adoiraccourcix* et la *GUP** comme affichée sur la figure 3.12) et *TÂCHE*. La variable *TÂCHE* était composée de 3 conditions :

- COPIER : créer la copie d'un objet et placer la copie dans une zone cible (*cf.* Figure 3.13-a).

8. Le mode correspond à la valeur la plus représentée dans la population des réponses.

FIGURE 3.12: La *GUI*^{*} montrant la tâche de COPIER.

- CRÉER : créer un rectangle en plaçant les sommets opposés dans des zones cibles (*cf.* Figure 3.13-b).
- ÉDITER : éditer le contour d'un triangle pour que l'un des sommets soit placé dans une zone cible (*cf.* Figure 3.13-c).

Les variables TECHNIQUE et TÂCHE ont été contrebalancées entre les participants en utilisant un carré Latin. Chaque condition TECHNIQUE $\times$ TÂCHE a été répétée 50 fois au travers de 5 BLOCS. Après chaque BLOC, les participants pouvaient prendre une pause. Pour pouvoir passer à l'ESSAI suivant, l'ESSAI courant devait être impérativement réussi. Les paramètres des TÂCHES (position des zones cibles, tailles, *etc.*) restaient constants. L'expérimentateur présentait systématiquement la manière la plus rapide d'effectuer chaque TÂCHE pour chacune des TECHNIQUES et imposait aux participants de l'utiliser pour réaliser la TÂCHE le plus rapidement possible. Pour la *GUI*^{*} les participants étaient contraints d'utiliser leur main non-dominante pour presser les boutons de la barre d'outils sur le panneau latéral et d'interagir avec leur main dominante. Pour *Adoiraccourcix* les participants étaient contraints d'utiliser la technique comme décrite précédemment. Les participants ont eu la possibilité de s'entraîner sur chacune des TÂCHES et chacune des TECHNIQUES.

Le plan expérimental était donc : 12 participants $\times$ 2 TECHNIQUES $\times$ 3 TÂCHES $\times$ 5 BLOCS $\times$ 10 ESSAIS = 3000 données.

Résultats

La variable dépendante principale est le temps de réalisation de tâche. Il est défini comme le temps écoulé entre le premier *Touch Down*^{*} et le dernier

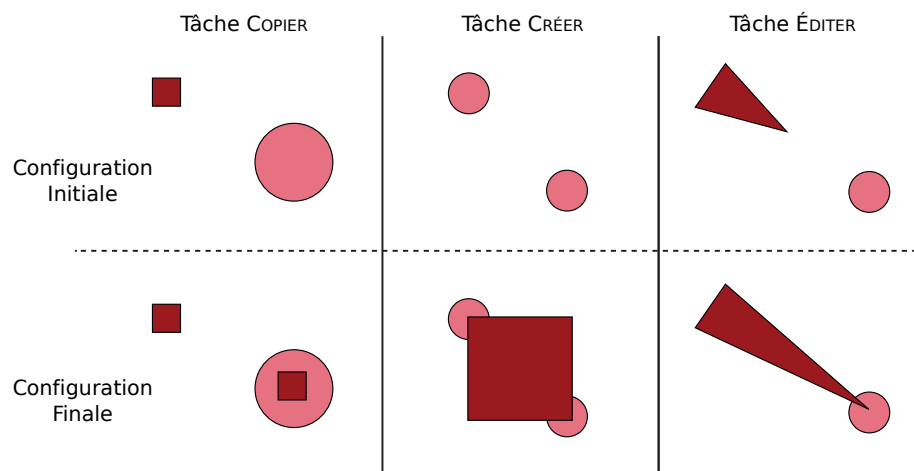


FIGURE 3.13: TÂCHES de l'expérience quantitative. Pour la tâche COPIER, l'utilisateur doit dupliquer un carré et le placer à l'intérieur d'une zone cible circulaire. Pour la tâche CRÉER, l'utilisateur doit dessiner un rectangle en partant d'une zone circulaire et en finissant dans une seconde zone circulaire. Pour la tâche ÉDITER, l'utilisateur doit déplacer le sommet bas droit d'un triangle à l'intérieur d'une zone circulaire.

*Touch Up** lors d'un ESSAI. La distribution des temps de réalisation n'étant pas normale, les données ont été pré-traitées en utilisant la méthode *Aligned Rank Transform* [WFGH11] avant d'être analysées à l'aide d'ANOVA à mesures répétées.

Les analyses ont montré un effet significatif de TECHNIQUE ($F_{1,11} = 38,7$, $p < 0,001$) et une interaction significative TECHNIQUE $\times$ TÂCHE ($F_{2,22} = 48,8$, $p < 0,001$) sur le temps de réalisation (cf. Figure 3.14). Les analyses *post hoc* ont révélé des différences significatives entre les deux TECHNIQUES pour la tâche COPIER ($p < 0,001$, GUT^* : 1,87 s, *Adoiraccourcix* : 1,03 s) et la tâche ÉDITER ($p < 0,001$, GUT^* : 1,14 s, *Adoiraccourcix* : 0,71 s). En revanche, aucune différence significative n'a été trouvée entre les deux TECHNIQUES pour la tâche CRÉER (GUT^* : 0,91 s, *Adoiraccourcix* : 0,92 s). Les différences significatives observées pour les tâches COPIER et ÉDITER peuvent être expliquées par le nombre plus important d'actions à réaliser lorsque l'on utilise la *GUT** (ex. appuyer sur les boutons pour copier, coller et changer de mode). Alors que les tâches COPIER et ÉDITER nécessitaient plusieurs changements de mode, la tâche CRÉER n'en nécessitait qu'un et ce quelque soit la TECHNIQUE utilisée.

3.4.3 Analyse basée sur le *Fingerstroke-Level Model*

En plus de nos études, nous avons aussi modélisé les tâches de l'expérience quantitative en utilisant le *Fingerstroke-Level Model (FLM)** décrit par Lee *et al.*

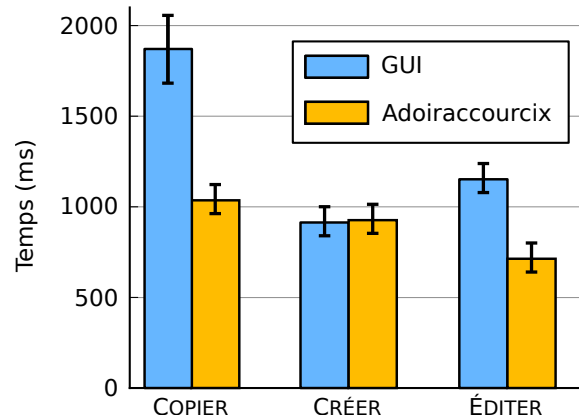


FIGURE 3.14: Temps moyens de réalisation pour *Adoiraccourcix* et la *GUI** sur les différentes TÂCHES (les barres d'erreurs sont les intervalles de confiance à 95% de la moyenne).

dans [LSR⁺15] (qui est une extension du *Keystroke-Level Model (KLM)*, décrit par Card *et al.* dans [CMN80], au vocabulaire d'entrées tactiles). Ce modèle montre comment chaque action contribuera aux différences de temps observées. Les temps prédits par le *FLM** ont été trouvés empiriquement par Lee *et al.* en utilisant des terminaux mobiles. Ces terminaux ont un facteur de forme différent du grand écran utilisé dans notre étude. Par conséquent, nous regardons le ratio des temps de réalisation entre *Adoiraccourcix* et la *GUI**. Si les ratios prédits par le *FLM** concordent avec ceux issus de nos expériences, nous utiliserons le *FLM** pour prédire les performances relatives entre la *GUI** et *Adoiraccourcix* pour d'autres tâches. Nous utilisons les opérateurs *FLM** suivants (le temps est indiqué quand celui-ci est disponible) :

- M pour *temps de préparation Mental* — temps nécessaire à la préparation des séquences d'actions.
- T pour *réalisation d'un tap** (0,31 s) — temps nécessaire pour sélectionner une cible virtuelle en appuyant dessus à une position donnée.
- P pour *Pointage* (0,52 s) — temps nécessaire pour se positionner et sélectionner une cible virtuelle en appuyant dessus.
- D pour *réalisation d'un Déplacement* (0,28 s) — temps nécessaire pour effectuer une translation d'un doigt ou d'un accord d'une position à une autre tout en maintenant le contact avec l'écran.

Nous étendons le *FLM** avec un opérateur supplémentaire nécessaire à notre analyse :

- A pour *Accord* (0,31 s) — le temps nécessaire pour toucher l'écran avec un accord *n'importe où sur l'écran* (nouveau). Après analyse de nos résultats

expérimentaux, nous approximations que $A = T$ quand les accords sont effectués par des experts⁹.

3.4.3.1 Analyse sur grand écran

La table 3.3 résume l'analyse basée sur le *FLM** qui compare *Adoiraccourcix* à la *GUI** pour les trois tâches de l'expérience quantitative.

Tâche COPIER. En utilisant la *GUI**, la séquence d'action est : préparation mentale ; *tap** pour sélectionner l'objet à copier ; *tap** sur le bouton copier ; pointer le bouton coller ; *tap** pour sélectionner l'objet copié ; déplacer l'objet copié à la position cible (séquence *fingerstroke* : MTTPD). En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale ; effectuer l'accord nd-pouce + nd-index ; *tap** sur l'objet avec l'index pour le copier ; pointer la position cible avec le majeur pour coller l'objet (séquence *fingerstroke* : MATP)

Tâche CRÉER. En utilisant la *GUI**, la séquence d'action est : préparation mentale ; *tap** pour sélectionner l'outil rectangle ; *tap** sur la zone de départ ; déplacement vers la zone de fin (séquence *fingerstroke* : MTTD). En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale ; effectuer l'accord nd-index ; *tap** sur la zone de départ avec le majeur ; déplacement vers la zone de fin (séquence *fingerstroke* : MATD).

Tâche ÉDITER. En utilisant la *GUI**, la séquence d'action est : préparation mentale ; *tap** pour sélectionner l'outil de modification des contours ; *tap** sur le sommet en bas droite du triangle ; déplacement vers la zone de fin (séquence *fingerstroke* : MTTD). En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale ; *tap** avec le majeur sur le sommet en bas droite du triangle ; déplacement vers la zone de fin (séquence *fingerstroke* : MTD).

Discussion Les participants ayant été entraînés, nous supposons un temps de préparation mental négligeable par rapport aux temps d'actions moteurs. Bien que les temps prédits et empiriques diffèrent, nous observons néanmoins une corrélation entre les ratios, prédits et empiriques, de temps *Adoiraccourcix* et *GUI** (les deux dernières colonnes de la table 3.3). Nous supposons que les différences absolues de temps entre les prédictions et les mesures sont dues aux différences de facteur de forme (*i.e.* appareils mobiles [LSR⁺15] *v.s.* grand écran).

Les tests effectués ne sont pas exhaustifs (*i.e.* seuls trois scénarios ont été évalués). Cependant, nous défendons qu'ils justifient quand même l'utilisation de *FLM** comme modèle théorique pour comparer les performances entre *Adoiraccourcix* et la *GUI**.

9. Des études complémentaires sont nécessaires pour mieux estimer les différents coûts (théoriques) de chaque opération, mais une explication possible pour l'équivalence observée entre *tap** et Accord peut être que le coût d'effectuer une combinaison de plusieurs doigts n'importe où sur l'écran est compensée par la distance à parcourir pour atteindre une cible.

Tâche	<i>GUI*</i>	<i>Adoiraccourcix</i>	Différence (<i>Adoiraccourcix</i> - <i>GUI*</i>)
COPIER	(M)TTPTD	(M)ATP	+ A - TTD
CRÉER	(M)TTD	(M)ATD	+ A - T
ÉDITER	(M)TTD	(M)TD	- T

Tâche	Ratio de temps Expérience	Ratio de temps <i>FLM*</i>
COPIER	0,55	0,66
CRÉER	1,01	1,00
ÉDITER	0,62	0,66

TABLE 3.3: Analyse comparative basée sur le *FLM** entre la *GUI** et *Adoiraccourcix*. Les colonnes de ratio de temps correspondent aux temps de réalisation d'*Adoiraccourcix* divisés par ceux de la *GUI** (mesurés au cours de l'expérience et prédits).

3.4.3.2 Analyse sur tablette

Nous utilisons les opérateurs *FLM** définis précédemment pour comparer *Adoiraccourcix* à une *GUI** traditionnelle. Nous ajoutons cependant un opérateur défini par Lee *et al.* dans [LSR⁺15] :

- F pour la *réalisation d'un flick** (0,11 s) — temps nécessaire réaliser un déplacement court et rapide.

Considérons une tâche simple mais courante : naviguer jusqu'à la deuxième page d'un document, surligner une phrase et ajouter une note textuelle. Dans *Adobe Acrobat 15.2.2* sur *Nexus 10*, la séquence d'action est : préparation mentale; *tap** sur la première page; déplacement jusqu'à la seconde page; pointer pour sélectionner l'outil surligneur; pointer la position du début de la phrase à surligner; déplacement pour surligner la phrase; pointer pour sélectionner l'outil de création de zone de texte; pointer la position où commence la zone de texte; déplacement pour définir la taille de la zone de texte (séquence *fingerstroke* : MTDPPDPPD). En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale; *tap** sur la première page avec l'index; déplacement jusqu'à la seconde page; pointer la position du début de la phrase à surligner avec le majeur; pointer la position où commence la zone de texte avec l'annulaire; déplacement pour définir la taille de la zone de texte (séquence *fingerstroke* : MTDPPDPPD). Dans cet exemple, *Adoiraccourcix* élimine un P par changement d'outil, deux P au total.

Considérons une autre tâche simple : surligner un paragraphe avec trois couleurs différentes. Dans *Adobe Acrobat 15.2.2* sur *Nexus 10*, la séquence d'action est : préparation mentale; pointer pour sélectionner l'outil surligneur; pointer la position du début de la phrase à surligner; déplacement pour surligner la phrase; *tap** optionnel pour sélectionner l'épaisseur du trait; *tap** pour sélectionner l'option couleur; *tap** sur la nouvelle couleur. Une seule couleur à la fois peut être

sélectionnée. La séquence *fingerstroke* correspondante à deux surlignage en jaune, un en rouge et en en vert est M-PPD-PPD-PPDTT-PPDTT. En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale ; *flick** avec le nd-pouce ; pointer la position du début de la phrase à surligner avec l'index, le majeur ou l'annulaire ; déplacement pour surligner la phrase. La séquence *fingerstroke* correspondante est M-F-PD-PD-PD-PD. Dans cet exemple, *Adoiraccourcix* élimine un P par surlignage et l'obligation de re-sélectionner une couleur déjà utilisée (trois T par sélection de couleur). Au total, c'est une réduction de quatre P et six T.

3.4.3.3 Analyse sur *smartphone*

Nous utilisons les opérateurs *FLM** définis précédemment pour comparer *Adoiraccourcix* à une *GUT** traditionnelle. Nous utilisons l'application *Messaging* sur *Android 5.1.1* comme *GUT** de référence. Comme dans de nombreuses *GUT** sur *smartphone**, les commandes telles que copier, coller et couper sont accessibles depuis un menu contextuel affiché après avoir effectué un appui long. Nous devons donc définir un nouvel opérateur *FLM** :

- L pour *réalisation d'un appui long* (entre 0,3 s and 0,5 s) — temps écoulé entre le début d'un contact et l'activation d'un appui long (nouveau). L'intervalle de temps donné correspond au 0,33 s utilisé par les *marking menus* [KB91] et du 0,5 s utilisé comme temps par défaut pour l'appui long dans le système *Android*.

Considérons une tâche simple mais courante : lors de la rédaction d'un message, l'utilisateur sélectionne un mot d'un message précédent du fil de discussion (*ex.* le nom d'un restaurant), le copie et le colle dans le message en cours d'écriture. Dans notre *GUT** de référence, la séquence d'action est : préparation mentale ; *tap** sur le mot à copier ; attendre l'activation de l'appui long ; *tap** sur le bouton copier ; *tap** à la position où le mot doit être coller ; attendre l'activation de l'appui long ; *tap** sur le bouton coller (séquence *fingerstroke* : M-TLT-TLT). En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale ; effectuer l'accord pouce ; *tap** sur le mot à copier avec l'index ; *tap** à la position où le mot doit être coller avec le majeur (séquence *fingerstroke* : M-AT-T). Dans cet exemple, *Adoiraccourcix* réduit le nombre d'opérations de deux LT pour le coût d'une opération A.

Considérons une autre tâche simple : écrire une phrase courte comprenant acronyme et ponctuation tel que "*The FLM model* :)". Dans notre *GUT** de référence, la séquence d'action est : préparation mentale ; *tap** sur la touche "T" ; pointer les touches "h", "e", "espace", "Maj", "F" etc. ; pointer les touches "Symboles", " ." et ")" (séquence *fingerstroke* : M-TPP-P-PPPPPP-P-PPPPPP-P-PPP). En utilisant *Adoiraccourcix*, la séquence d'action est : préparation mentale ; *tap** sur la touche "T" avec le majeur ; pointer les touches "h", "e" et "espace" avec l'index ; pointer les touches "F", "L" et "M" avec le majeur ; pointer la touche "espace" avec l'index ; pointer les touches " ." et ")" avec l'annulaire (séquence *fingerstroke* : M-TPP-P-PPP-P-PPPPPP-P-PP). Dans cet exemple,

Adoiraccourcix élimine tous les P correspondant aux touches modificatrices (trois touches “*Maj*” et une “*Symboles*”). Au total, c’est une réduction de quatre P.

Les résultats de nos analyses montrent un gain de performance comparé à des interfaces traditionnelles et indiquent aussi que les utilisateurs sont capables d’utiliser l’identification des doigts pour la sélection de commandes et le contrôle continu de paramètres. Dans la suite, nous discutons les choix derrière les implémentations concrètes d’*Adoiraccourcix*. Enfin, bien que le potentiel de l’identification des doigts semble se confirmer, nous décrivons aussi les limites que nous avons identifiées.

3.5 Discussion

Motivé par nos trois applications de démonstrations qui mettent en œuvre *Adoiraccourcix*, nous résumons ici les points clefs, donnons des éléments pour guider la conception d’interfaces basées sur *Adoiraccourcix*, donnons des recommandations et discutons les limitations actuelles et inhérentes d’*Adoiraccourcix*.

3.5.1 Ergonomie de l’appareil

L’ergonomie de l’appareil joue un rôle clef dans la détermination du nombre de doigts utilisables pour les accords. Les *smartphones** et les tablettes sont généralement tenus avec la main non-dominante, ce qui confine l’interaction avec l’écran à la main dominante. Cependant, le pouce de la main non-dominante reste la plupart du temps disponible pour interagir avec l’écran [WHM12]. Nos applications d’écriture de SMS et d’annotation de PDF utilisent les deux pouces comme un seul et même *sélecteur*, pour supporter cette aptitude et donner une certaine flexibilité à la façon dont les *sélecteurs* et *déclencheurs* sont effectués. Une unique configuration *sélecteur* était suffisante pour nos application de SMS mais pas pour notre application sur tablette. Nous illustrons donc une stratégie possible qui utilise les mouvements des doigts pour engendrer plus d’un groupe d’associations commandes-accords sur une seule et même configuration *sélecteur*. À l’inverse, les grands écrans, par définition plus large, permettent des applications avec plus de commandes. Dans ce contexte, notre application de dessin vectoriel associe les doigts *sélecteurs* à ceux de la main non-dominante et les doigts *déclencheurs* à ceux de la main dominante.

3.5.2 Cohérence des associations

Durant la conception des applications de démonstration, nous prenions en compte la rétrocompatibilité avec les techniques existantes qui se basent sur le nombre de doigts en contact ainsi qu’une cohérence au travers des différents appareils et applications pour optimiser le transfert de compétences. Par exemple, les commandes annuler et refaire sont systématiquement déclenchées par des *flicks** gauche et droit du pouce. Dans les applications PDF et dessin vectoriel,

nous réservons l'accord pouce + index pour le zoom et les transformations d'objet. De même, l'index est systématiquement utilisé pour déplacer des objets et défiler dans un document. Ces décisions se basent sur les résultats de l'étude d'observation des accords sur pavés tactiles (*cf.* Section 2.2.3.3).

Lorsqu'il ne nous était pas possible d'utiliser les mêmes associations commandes-accords sur tous les appareils, nous utilisons le même schéma d'interaction. Par exemple, bien que les commandes copier, coller et couper sont accessibles *via* différents accords *sélecteurs* sur les différents appareils, systématiquement l'index copie, le majeur colle et l'annulaire coupe.

3.5.3 Choix des associations commandes-accords

Pour combiner la sélection de commandes et la manipulation de paramètres, il est impératif de savoir quels sont les paramètres à contrôler et comment ceux-ci peuvent l'être. Nos démonstrations se focalisent sur le contrôle continu de la position d'un unique point. L'utilisation d'un doigt unique est donc la plus indiquée. Cependant d'autres paramètres peuvent requérir des degrés de liberté supplémentaires, ce qui nécessite l'emploi de doigts *déclencheurs* supplémentaires (*ex.* le *RST** est manipulé à l'aide d'un accord à deux doigts). Pourtant, n'utiliser qu'un seul doigt *déclencheur*, avec un nombre arbitraire de doigts *sélecteurs*, résout le problème de l'ambiguïté temporelle puisqu'aucune fenêtre de temps n'est requise pour reconnaître un accord. Par conséquent, les commandes sont exécutées sans délai. De plus, il n'y a pas d'ambiguïté spatiale puisque l'utilisateur s'attend à ce que l'action se passe sous son doigt *déclencheur*.

Les résultats de notre étude sur la dextérité des doigts et des accords (*cf.* Section 2.1.8) nous indiquent que l'index et le majeur sont précis et plus appréciés que l'annulaire et l'auriculaire. Une corrélation pourrait être faite avec l'indépendance physique de chaque doigt et la gêne physique ressentie lors de contacts engendrés par l'annulaire et l'auriculaire. Les commandes fréquentes et nécessitant de la précision doivent plutôt être associées aux doigts performants et appréciés. Par exemple dans l'application d'annotation de PDF, l'index est utilisé pour le défilement (commande fréquente), la copie ainsi que l'annotation.

Les commandes n'ayant aucun paramètre, telle que la sauvegarde, peuvent être associées à un accord de plusieurs *déclencheurs* vu qu'aucun point d'interaction particulier n'est nécessaire. Enfin, les accords précis mais moins appréciés (*cf.* Section 2.1.8) peuvent être associés à des commandes risquées. Par exemple, dans l'application de dessin vectoriel, la commande de suppression est associée à l'accord nd-pouce + nd-index + auriculaire.

3.5.4 Efficacité de l'intégration de la sélection de commandes et du contrôle continu de paramètres

En comparant trois techniques qui intègrent la sélection de commande à la manipulation directe, Guimbretière *et al.* ont établi que le facteur déterminant qui influence la performance des techniques réside dans la *fluidité* combinant la sélection et la manipulation [GMW05]. Les résultats de notre étude utilisateur,

montrant qu'*Adoiraccourcix* peut être plus rapide qu'une *GUF* pour des tâches canoniques, sont cohérents avec cette observation. Bien que les participants pouvaient utiliser leurs deux mains pour interagir avec la *GUF*, l'intégration de la sélection de commandes et la manipulation directe d'*Adoiraccourcix* résultaient en un nombre moins important d'actions à effectuer et donc des temps de réalisation plus courts. Notre expérience ciblait une utilisation experte de la technique *via* un grand nombre de répétitions d'une même tâche. Nous supposons que notre schéma d'interaction en trois étapes peut aider les utilisateurs novices à progresser pour devenir rapidement des experts efficaces.

L'analyse basée sur le *FLM* confirme cette réduction du nombre d'actions chronophages, particulièrement quand des changements de modes sont nécessaires. Elle permet aussi une première étape de validation pour généraliser les résultats de nos études sur grands écrans. Cependant des évaluations complémentaires doivent être réalisées pour quantifier empiriquement le gain de temps possible dans différents contextes.

3.5.5 Limites

La première limite d'*Adoiraccourcix* est le besoin qu'ont les nouveaux utilisateurs de découvrir quels doigts ou accords sont les *sélecteurs*. Une fois ceux-ci connus, ils peuvent découvrir les commandes et les *déclencheurs* associés en essayant les différents *sélecteurs* et en examinant *Lilanotix* (aucune commande n'est déclenchée tant qu'un *déclencheur* n'est pas utilisé). Bien que ceci constitue une barrière, notre étude qualitative montre que les utilisateurs peuvent découvrir et utiliser les *sélecteurs* avec un minimum d'instructions. Cela souligne aussi l'importance de la conception du *feedforward* pour éviter la confusion entre *Lilanotix* et une palette à outils avec boutons.

Étant donné que les *sélecteurs* ne requièrent pas de position particulière pour pouvoir être utilisés, l'utilisateur peut se concentrer sur les *déclencheurs*. L'inconvénient est que les configurations *sélecteurs* ne peuvent pas être utilisées pour interagir avec les éléments de l'interface. Par exemple, dans la version courante de notre application de rédaction de SMS, l'utilisateur ne peut plus utiliser ses pouces pour écrire. Une façon de contourner ce problème est d'interpréter les configurations *sélecteurs* comme des interactions directes en fonction du temps, de la position et de contacts simultanés de *déclencheurs*. À titre d'exemple, un appui court du pouce sur un bouton sans contact sur l'écran de *déclencheurs* signifierait l'appui du bouton. À l'inverse, un appui long afficherait *Lilanotix*.

Tous les accords qui incluent un doigt *sélecteur* bénéficient de *feedforward*. Néanmoins, par définition, *Lilanotix* ne peut être utilisée pour les accords qui n'impliquent que des *déclencheurs*. Par exemple, considérons l'accord index+majeur pour du défilement relatif de contenu (*i.e.* défilement classique sur un navigateur internet) et l'accord majeur+annulaire pour du défilement absolu (*i.e.* mise en correspondance de la position de l'accord sur l'écran et la position de l'ascenseur). Si l'index, le majeur et l'annulaire sont tous des *déclencheurs*, il devient compliqué d'informer l'utilisateur que de telles actions sont possibles. Le même problème s'applique aux doigts *déclencheurs* seuls. Il faut cependant

noter que le nombre de commandes qui entrent dans cette catégorie est limité et le coût mémoire associé reste faible. Pour y remédier, un mode d'aide explicite peut être conçu pour découvrir de telles commandes.

Notre exploration suppose une identification des doigts robuste sur écrans multi-points. Si les premiers prototypes commerciaux ne fournissent pas une identification parfaite, le *feedforward** et le *feedback** permettront une certaine tolérance aux faux positifs. Par exemple, si la notice attendue de *Lilanotix* ne s'affiche pas, l'utilisateur comprendrait qu'il lui faut réajuster légèrement la position de ses doigts et rejouer l'accord.

Pour finir, *Adoiraccourcix* suppose la connaissance de la main dominante de l'utilisateur quand le rôle des deux mains est asymétrique (*ex.* notre application de dessin vectoriel). Ceci peut faire partie d'un profil utilisateur, ou bien encore un bouton ou un geste pourrait inverser la dominance courante de la main.

3.6 Conclusion

Dans ce chapitre, nous avons présenté *Adoiraccourcix*, une technique d'interaction qui intègre sélection de commandes et contrôle continu de paramètres et tire partie de l'identification des doigts. Étant donné le vaste espace d'entrées offert par l'identification des doigts et les problèmes de découverte des accords associés, *Adoiraccourcix* définit un schéma d'interaction en trois étapes qui utilise des doigts *déclencheurs* pour déclencher des commandes, des doigts *sélecteurs* pour changer les commandes associées aux *déclencheurs* et des doigts *modificateurs* pour altérer l'exécution en cours d'une commande. Ce schéma en trois étapes aide les utilisateurs novices en introduisant *Lilanotix*, un système de *feedforward** et de *feedback**, tout en permettant aux experts d'exécuter les commandes rapidement. Nous avons illustré *Adoiraccourcix* au travers de trois applications chacune liée à un contexte particulier. Nous avons également détaillé comment maintenir une rétrocompatibilité avec les techniques multi-points existantes et nous avons guidé les concepteurs d'interactions pour mettre en œuvre *Adoiraccourcix* sur des périphériques avec différents facteurs de forme. Les résultats d'études préliminaires et d'une analyse basée sur le *FLM** montrent que les utilisateurs sont capables d'utiliser l'identification des doigts pour la sélection de commandes et le contrôle continu de paramètres. En outre, ils sont aussi capables d'être plus performants qu'avec une *GUT** traditionnelle sur des tâches qui nécessitent de fréquents changements de modes.

Conclusion

Avec la démocratisation des interfaces tactiles, les concepteurs d'applications se sont largement tournés vers cette modalité. L'offre du catalogue est conséquente mais l'expressivité limitée des interfaces multi-points borne la richesse des fonctionnalités. L'augmentation du vocabulaire d'entrée est donc primordiale et nombre de recherches s'y attachent. Ainsi l'ère du *Rich Touch* se dessine avec l'apparition de nouvelles informations enrichissant l'interaction comme la pression ou encore l'orientation. Parmi les nouvelles informations étudiées : l'identification des doigts.

La plupart des travaux de la littérature sur l'identification des doigts essaie de résoudre le défi technologique : comment discriminer les doigts d'un utilisateur. L'utilisation de l'identification des doigts est donc peu explorée et sa compréhension n'en est qu'à ses balbutiements. Les travaux de cette thèse ont donc eu pour but d'enrichir les connaissances du domaine. Nos contributions se sont faites selon trois axes : identifier une solution technologique pouvant supporter l'identification des doigts dans des systèmes grand public ; guider la conception des techniques d'interaction grâce à une meilleure compréhension de l'impact de l'identification des doigts sur l'utilisateur ; proposer une technique d'interaction mettant en œuvre les recommandations établies.

Résumé

Dans un premier temps (*cf.* Chapitre 1), nous nous sommes attachés à faire un tour d'horizon des différentes technologies utilisées dans la littérature pour discriminer les doigts. Cet état de l'art nous a permis d'identifier une technologie particulièrement intéressante pour supporter l'identification des doigts dans un contexte d'utilisation grand public : la reconnaissance des empreintes digitales. Afin d'étudier la viabilité de cette solution, nous avons présenté une étude en deux parties. La première partie établit les zones de la pulpe que nous utilisons lors de nos interactions sur des écrans tactiles. Les résultats montrent sans surprise que nos doigts sont rarement à plat (*i.e.* cas idéal de la reconnaissance d'empreintes digitales). Ils soulèvent donc le problème suivant : possède-t-on suffisamment d'informations sur la périphérie de la pulpe pour discriminer les doigts. Les résultats de la seconde partie de l'étude suggèrent que nous possédons une quantité d'informations non-négligeable, mais que la reconnaissance d'em-

preintes digitales seule ne suffit pas pour supporter l'identification des doigts.

Par la suite (*cf.* Chapitre 2), nous avons étudié trois aspects de l'identification des doigts pour mieux comprendre son impact sur l'utilisateur.

La première étude s'attache à établir les différences de performances et de préférences entre les doigts et certains accords pour les actions de pointage, de déplacement d'objets, de rotation et de changement d'échelle. Les résultats nous ont permis d'établir trois groupes de niveaux pour chacune de ces actions (*cf.* Tables 2.9 et 2.10). La connaissance de ces groupes de niveaux permet de motiver le choix des associations entre commandes et accords pour suivre au mieux les capacités des utilisateurs.

La seconde étude porte sur l'influence de cette nouvelle information sur la stratégie suivie par l'utilisateur pour réaliser une tâche. Après avoir présenté et validé deux métriques qui quantifient objectivement les stratégies utilisateurs, nous nous sommes intéressés aux stratégies obtenues lors de l'utilisation de techniques basées sur l'identification des doigts. Bien que nos techniques aient été conçues afin de réduire les mouvements requis pour accomplir une tâche et optimiser le temps de réalisation, les stratégies choisies ne s'attachaient pas toutes à réduire le coût moteur. Ce résultat laisse à penser que l'effort de mémorisation nécessaire doit être pris en compte dans la conception de ces nouvelles interactions. L'intégration de *feedback** et de *feedforward** semble donc essentielle pour augmenter l'efficacité des utilisateurs.

La dernière étude se concentre sur l'intégration des techniques d'interaction avec identification des doigts au sein de celles déjà communément utilisées. Étant donné le vocabulaire de gestes existants et leur disponibilité sur les interfaces tactiles grand public, introduire l'identification des doigts pourrait venir contrarier les habitudes des utilisateurs. Nous avons donc observé les accords utilisés pour effectuer les gestes multi-points classiques. Les résultats ont montré une importante variabilité dans l'association d'un geste à un accord. L'identification des doigts imposerait donc à l'utilisateur de s'adapter à cette nouvelle information. Cependant, en fonction du nombre de fonctionnalités à ajouter, l'adaptation pourrait être minime. En effet, la majorité des accords, à un, deux ou trois doigt(s), est une combinaison des premiers doigts de la main (pouce, index, majeur) ce qui laisse un nombre non-négligeable d'accords libres.

Ce chapitre établit donc une liste de recommandations utiles à la conception de techniques d'interaction utilisant l'identification des doigts.

Enfin (*cf.* Chapitre 3), en se basant sur les conclusions des études du chapitre 2 et sur une analyse méthodique de l'espace d'interaction utilisant l'identification des doigts, nous avons présenté *Adoiraccourcix*, une technique de sélection de commandes avec contrôle continu de paramètres, inspirée des raccourcis clavier. Cette technique fonctionne en trois étapes : 1) un accord de doigts *sélecteurs* sélectionne les commandes associées à un autre ensemble de doigts *déclencheurs* ; 2) chaque doigt de l'ensemble *déclencheurs* déclenche une commande et contrôle les paramètres de celle-ci dans un même mouvement — une fois la manipulation commencée, les doigts *sélecteurs* utilisés peuvent être retirés ; enfin 3) durant la

manipulation de paramètres, à chaque doigt *sélecteur* est associé une contrainte, un accord de *sélecteurs* peut donc modifier optionnellement la manipulation. Les associations entre les commandes et les *déclencheurs* suivent les recommandations du chapitre 2. De plus, l'ajout de *feedforward** et de *feedback** guide l'utilisateur dans la construction progressive de leurs accords. Pour les experts, ces trois étapes peuvent être effectuées plus rapidement.

Nous avons illustré *Adoiraccourcix* au travers de trois applications, chacune liée à un facteur de forme particulier : implémentation d'une application de dessin vectoriel sur grand écran, application d'annotation de documents sur tablettes et application de messagerie sur *smartphone**. Nous avons évalué la version grand écran à l'aide de deux études utilisateurs : une étude qualitative suggérant qu'avec de la pratique, *Adoiraccourcix* est expressive, facile à utiliser et améliore la sensation de fluidité de l'interaction ; une étude quantitative montrant qu'*Adoiraccourcix* peut être aussi rapide, voire plus rapide, qu'une interface graphique traditionnelle, principalement dû au fait que le nombre d'opérations requises est moins élevé. Enfin, une analyse basée sur le Fingerstroke-Level Model (*FLM*)* confirme les résultats de l'étude quantitative et nous permet de généraliser ces résultats sur tablette et *smartphone**.

Perspectives de recherche

Bien que les travaux présentés constituent un premier pas vers la compréhension et la conception méthodique de techniques d'interaction avec identification des doigts, les améliorations et aspects restant à étudier sont encore nombreux.

L'utilisation des empreintes digitales est *a priori* prometteuse. Cependant notre dispositif et protocole de comparaison des images peuvent être grandement améliorés. La mise en œuvre de ces améliorations ainsi que des heuristiques évoquées (*ex.* utiliser l'estimation du tangage pour réduire le nombre d'images dans la base de données) seront nécessaires pour s'assurer de la viabilité de l'utilisation des empreintes digitales.

Nous avons montré qu'il existe des différences de performances et de préférences entre les doigts et certains accords pour le pointage, le déplacement d'objets, la rotation et le changement d'échelle. Néanmoins, d'autres études restent à être menées pour compléter ces résultats. Un certain nombre d'accords à deux doigts n'a pas été testé, tout comme les accords à trois doigts et plus. De même, certaines tâches, comme les gestes de *flick** ou de défilement, et certains facteurs, comme la taille de l'écran ou la façon de tenir le dispositif tactile, peuvent aussi faire l'objet d'études complémentaires.

Notre étude sur l'observation des associations entre accords et gestes sur pavés tactiles nous permet d'établir des habitudes utilisateurs. Bien que nous les pensons transposables dans d'autres contextes (*ex.* *smartphones** et tablettes) au vu de la similarité entre les deux vocabulaires de gestes, il est indéniable que des études complémentaires soient nécessaires pour prouver la généralisation de nos résultats.

Enfin, nous avons proposé des mises en œuvre d'*Adoiraccourcix* dans trois

contextes différents. Mais d'autres contextes, comme les pavés tactiles qui sont des dispositifs de pointage indirects, méritent aussi d'être explorés. De même, notre technique qui est actuellement adaptée à des applications isolées pourrait être mise en œuvre au niveau d'un système d'exploitation. Cette perspective offrirait d'ailleurs la possibilité d'une étude de terrain, qui montrerait si nous utilisons et sommes capables de nous adapter aux techniques d'interaction tactiles avec identification des doigts.

Glossaire

API voir *Application Programmable Interface*. 92

Application Programmable Interface Interface de Programmation Appllicative. En programmation, c'est un ensemble de fonctions et méthodes permettant l'utilisation d'un logiciel au sein d'un autre. 92

bits Unité de mesure de la quantité d'information transmise par un message. 66, 69, 73, 75

bps Bit par seconde. Unité de débit binaire qui mesure la quantité de données numériques transmises par unité de temps.
[Wikipédia : http://fr.wikipedia.org/wiki/Debit_binaire]. 54, 55, 77

crossing Franchissement de cible par un pointeur. 54

FANN voir *Fast Approximate Nearest Neighbors*. 42

Fast Approximate Nearest Neighbors Algorithme de recherche des plus proches voisins présenté par Muja *et al.* dans [ML09]. 42

feedback Information fournie par le système pendant ou après une action de l'utilisateur. Terme anglais utilisé à la place de *retour d'information* pour faciliter la lecture. 4, 6, 58, 59, 63, 64, 109–111, 119, 137, 140, 141

feedforward Information fournie avant que l'action ne se déroule. Terme anglais utilisé à la place de *retour d'information par anticipation* pour faciliter la lecture. 4, 19, 99, 100, 104, 105, 110, 111, 113, 119, 127, 136, 137, 140, 141

Fingerstroke-Level Model Modèle prédisant la durée de tâches sur dispositif tactile. Ce modèle décompose chaque tâche en action élémentaire dont le temps est connu. Il est proposé par Lee *et al.* dans [LSR⁺15] et est inspiré du Keystroke-Level Model décrit par Card *et al.* dans [CMN80]. 100, 129, 141

flick Geste de balayage rapide d'un doigt sur un dispositif tactile. Il est souvent utilisé pour faire défiler du contenu verticalement ou horizontalement (*ex.* changement de bureau sous *Android*). 35, 56, 78, 102, 113, 123, 124, 132, 133, 141

flicks voir *flick*. 15, 16, 34, 53, 54, 78, 81, 92, 94, 101, 102, 120, 121, 123, 124, 134

- FLM** voir *Fingerstroke-Level Model*. 100, 124, 125, 129–133, 136, 137, 141
- fps** Image par secondes (*frame per second*). Unité de mesure correspondant au nombre d'images affichées ou enregistrées en une seconde par un dispositif. [Wikipédia : http://fr.wikipedia.org/wiki/Images_par_seconde]. 58
- Frustrated Total Internal Reflection** Réflexion totale. Phénomène optique qui survient lorsqu'un rayon lumineux arrive sur la surface de séparation de deux milieux d'indices optiques différents avec un angle d'incidence supérieur à une valeur critique : il n'y a alors plus de rayon réfracté transmis, et seul subsiste un rayon réfléchi. [Wikipédia : http://fr.wikipedia.org/wiki/Reflexion_totale]. 10, 39
- FTIR** voir *Frustrated Total Internal Reflection*. 10–12, 39, 41
- Gametraks** Contrôleur de jeu permettant de capter la position de l'extrémité d'un fil rétractable dans l'espace à l'aide de 3 potentiomètres. [Wikipédia : <http://fr.wikipedia.org/wiki/Gametrak>]. 27, 29, 89
- Graphical User Interface** Interface graphique utilisateur. 124
- GUI** voir *Graphical User Interface*. 124, 125, 127–133, 136, 137
- pan** Geste de navigation utilisé sur les dispositifs tactiles. Le *pan* déplace la vue d'une application pour afficher la sous-partie du contenu voulu (*ex.* déplacement de la carte dans *Google Map*). 2, 78, 105
- pinch-and-expand** Geste de transformation utilisé sur les dispositifs tactiles. Le *pinch-and-expand* permet en rapprochant ou éloignant des points de contact de contrôler le niveau de zoom (*ex.* agrandissement ou rétrécissement dans *Google Map*). 15, 52, 82, 92, 101, 110, 121
- ppp** Pixel par pouce. Unité de mesure de densité exprimant la résolution d'une image. Elle correspond au nombre de pixels utilisés pour afficher un pouce. [Wikipédia : http://fr.wikipedia.org/wiki/Pixel_par_pouce]. 58
- Radio-Frequency IDentification** Méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés *radio-étiquettes* contenant un identifiant voire des données complémentaires. [Wikipédia : <http://fr.wikipedia.org/wiki/Radio-identification>]. 3, 19
- RFID** voir *Radio-Frequency IDentification*. 3, 19, 24
- Rotate-Scale-Translate** Geste de transformation utilisé sur les dispositifs tactiles. Effectué en un seul et même mouvement, le *RST* permet de tourner un objet (ou un contenu), de changer son échelle et de le déplacer dans la vue de l'application. 54, 110
- RST** voir *Rotate-Scale-Translate*. 54, 110, 117, 119, 120, 135
- smartphone** Téléphone portable disposant d'un écran tactile (*ex.* *iPhone*, *Nexus 5*, *etc.*). 1, 3, 4, 6, 9, 17, 22, 26, 29, 30, 34, 55, 57, 99, 100, 104, 109, 112, 114, 115, 123, 124, 133, 141

smartphones voir *smartphone*. 3–5, 11, 24, 97, 115, 134, 141

Speeded Up Robust Features En traitement d'image, c'est un algorithme de détection de caractéristiques présenté par Bay *et al.* dans [BTVG06]. 42

SURF voir *Speeded Up Robust Features*. 42

tap Pression rapide sur un dispositif tactile. Il est souvent utilisé pour effectuer une sélection (*ex.* lancement d'une application sous *Android*). 14, 19, 35, 54, 57, 59, 66, 102, 113, 123, 124, 130–133

throughput Le throughput est une mesure de la performance humaine. Il correspond au débit d'information moyen généré par un mouvement. Le throughput est aussi appelé index de performance. 53–56, 63, 65, 67, 68, 70, 71, 73, 75, 77

Touch Down Événement tactile correspondant à la détection de nouveaux contacts sur le dispositif tactile. 13, 26, 30, 65, 66, 69, 128

Touch Move Événement tactile correspondant à la mise à jour (*ex.* changement de position, de pression, *etc.*) de contacts sur le dispositif tactile. 13, 26

Touch Up Événement tactile correspondant à la disparition de contacts sur le dispositif tactile. 13, 26, 66, 129

TUIO Protocole de communication d'événements tactiles. Il permet de séparer la partie captation des événements et la partie applicative qui utilise ces événements [KBBC05].
<http://www.tuio.org>. 27, 29

widget Composant de base des interfaces graphiques. 15, 87, 88

widgets voir *widget*. 15

WIMP voir *Windows, Icons, Menus and Pointing device*. 105

Windows, Icons, Menus and Pointing device Fenêtres, icônes, menus et dispositif de pointage. Paradigme d'interaction utilisé par la plupart des systèmes d'exploitations (*ex.* *Windows*, *OSX*, *Android*). 105

Annexe A

Publications

La liste des publications ci-dessous regroupe les travaux et activités liées à la recherche qui ont été produits durant ces trois années de thèse. Pour les travaux dont le contenu est en partie présenté dans cette thèse, le chapitre ou la section correspondante est mentionné(e) entre crochets. Les versions PDF des publications sont disponibles à l'adresse <http://www.alixgoguey.fr>.

CONFÉRENCE INTERNATIONALE AVEC COMITÉ DE LECTURE

1. **Alix Goguey**, Mathieu Nancel, Géry Casiez & Daniel Vogel. The Performance and Preference of Different Fingers and Chords for Pointing, Dragging, and Object Transformation. In the 33th Conference on Human Factors in Computing Systems (CHI 2016). 4250-4261. [Section 2.1]
2. **Alix Goguey**, Julie Wagner & Géry Casiez. Quantifying Object- and Command-Oriented Interaction. In the 15th IFIP TCI3 Conference on Human-Computer Interaction (INTERACT 2013). 231-239. [Section 2.2.2]

CONFÉRENCE NATIONALE AVEC COMITÉ DE LECTURE

3. Sylvain Malacria, **Alix Goguey**, Gilles Bailly & Géry Casiez. Étude de terrain sur l'utilisation des pavés tactiles. À la 28e conférence francophone sur l'Interaction Homme-Machine (IHM 2016). Accepté. [Section 2.2.3]
4. **Alix Goguey**, Géry Casiez, Thomas Pietrzak, Daniel Vogel & Nicolas Roussel. Adoiraccourcix : Sélection de Commandes sur Écrans Tactiles Multi-Points par Identification des Doigts. ★ Meilleur article. À la 26e conférence francophone sur l'Interaction Homme-Machine (IHM 2014). 28-37. [Chapitre 3]

JOURNAL

5. **Alix Goguey**, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak, Nicolas Roussel & Géry Casiez. Leveraging Finger Identification to Integrate Multi-touch Command Selection and Parameter Manipulation. In International Journal of Human-Computer Studies (IJHCS). Conditionnellement accepté. [Chapitre 3]

DÉMONSTRATIONS

6. **Alix Goguey**, Géry Casiez, Thomas Pietrzak, Daniel Vogel & Nicolas Roussel. Sélection de Commandes sur Tables Interactives Multi-Points par Identification des Doigts. À la 26e conférence francophone sur l'Interaction Homme-Machine (IHM 2014). 12-13. [Chapitre 3]
7. **Alix Goguey**, Géry Casiez, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak & Nicolas Roussel. A Three-Step Interaction Pattern for Improving Discoverability in Finger Identification Techniques. In the 27th ACM User Interface Software and Technology Symposium (UIST 2014). 33-34. [Chapitre 3]

RENCONTRES DOCTORALES

8. **Alix Goguey**. L'Identification des Doigts au service de Nouvelles Interactions Tactiles. Rencontres Doctorales. Annexes de la 27e conférence francophone sur l'Interaction Homme-Machine (IHM 2015).

ÉDITION D'ACTES DE CONFÉRENCE

9. **Alix Goguey**, Olivier Chapuis, Stéphane Conversy, José Rouillard, Nadine Vigouroux, Géry Casiez & Thomas Pietrzak. Éditeurs. Actes de la conférence IHM'14. ISBN 978-1-4503-2935-4. ACM.

MÉDIATION SCIENTIFIQUE

10. **Alix Goguey**. Participation au concours *Ma thèse en 180 secondes*. Qualifié pour la finale régionale Nord-Pas de Calais.
11. **Alix Goguey**. Adoiraccourcix : Sélection de Commandes sur Écrans Tactiles utilisant l'Identification des Doigts. Dans le cadre du programme *Chercheurs Itinérants* pour la fête de la Science (2015)
12. **Alix Goguey**. L'identification des doigts au service de raccourcis tactiles. Dans le cadre du programme *Chercheurs Itinérants* pour la fête de la Science (2014)

Annexe B

Détail de l'observation des accords

Les tables B.1, B.2 et B.3 montrent le détail des accords observés et décrits dans la section 2.2.3.3. Pour chaque geste, nous donnons la liste de l'ensemble des accords utilisés. Les accords sont représentés visuellement : les dix cercles symbolisent (dans l'ordre) les 5 doigts de la main gauche (de l'auriculaire au pouce) ainsi que les 5 doigts de la main droite (du pouce à l'auriculaire). Les cercles ● correspondent aux doigts impliqués dans l'accord. Par exemple, ○○○○○ ●●○○○ correspond à l'accord pouce droit + index droit. Pour chaque accord, nous donnons son nombre d'occurrence dans les données ainsi que ses taux d'occurrence pondérés (et non-pondérés) par utilisateur. Le taux d'occurrence est la proportion d'un accord dans le jeu de données. Le taux d'occurrence pondéré par utilisateur est la moyenne des taux d'occurrence par utilisateur. C'est ce taux que nous utilisons dans la section 2.2.3.3 car il permet de normaliser les participants ayant une fréquence d'utilisation différente de leur ordinateur.

Geste	Accord	Nb. Obs.	%	
Pointage	○○○○○ ○○○●	1	0.1	(0.1)
	○○○○○ ○○○●○	128	6.6	(10.8)
	○○○○○ ○○●○○	652	39.4	(55.0)
	○○○○○ ○●○○○	343	48.2	(28.9)
	○○○○○ ●○○○○	28	2.1	(2.4)
	○○○○● ○○○○○	1	0.1	(0.1)
	○○○●○ ○○○○○	9	0.8	(0.8)
	○○●○○ ○○○○○	22	2.6	(1.9)
	○●○○○ ○○○○○	1	0.2	(0.1)
Défilement	○○○○○ ○○○●●	16	0.6	(0.1)
	○○○○○ ○○●○●	2	0.0	(0.0)
	○○○○○ ○○●●○	13224	56.2	(54.6)
	○○○○○ ○●○○●	5	0.0	(0.0)
	○○○○○ ○●○●○	31	0.2	(0.1)
	○○○○○ ○●●○○	10268	40.4	(42.4)
	○○○○○ ●○○●○	1	0.0	(0.0)
	○○○○○ ●○●○○	1	0.0	(0.0)
	○○○○○ ●●○○○	11	0.1	(0.0)
	○○○○● ○●○○○	1	0.0	(0.0)
	○○○●○ ○○○●○	5	0.0	(0.0)
	○○○●○ ○○●○○	9	0.0	(0.0)
	○○○●○ ○●○○○	9	0.0	(0.0)
	○○○●● ○○○○○	2	0.0	(0.0)
	○○●○○ ○○●○○	3	0.0	(0.0)
	○○●○● ○○○○○	1	0.0	(0.0)
	○○●●○ ○○○○○	351	1.0	(1.4)
	○○●○○ ○○○○○	276	1.3	(1.1)

TABLE B.1: Détails des observations des accords sur pavé tactile par geste (1/3). Pour chaque geste sont regroupés l'ensemble des accords qui ont été utilisés au moins une fois. Les accords sont représentés par dix cercles correspondant (dans l'ordre) aux 5 doigts de la main gauche (de l'auriculaire au pouce) puis aux 5 doigts de la main droite (du pouce à l'auriculaire). Les cercles ● correspondent aux doigts impliqués dans l'accord. Nb. Obs. correspond au nombre d'observations. % correspond au taux d'occurrence pondéré par utilisateur. Nous donnons entre parenthèses le taux non-pondéré.

Geste	Accord	Nb. Obs.	%	
<i>Pinch-and-expand</i> à 2 doigts	○○○○○ ○○○●●	1	1.6	(0.9)
	○○○○○ ○○●●○	7	5.2	(6.5)
	○○○○○ ○●○○●	1	5.6	(0.9)
	○○○○○ ○●○●○	9	10.2	(8.4)
	○○○○○ ○●●○○	46	19.8	(43.0)
	○○○○○ ●○●○○	6	8.6	(5.6)
	○○○○○ ●●○○○	29	35.8	(27.1)
	○○○●○ ○○●○○	3	6.7	(2.8)
	○○○●○ ○●○○○	2	2.5	(1.9)
	○○○●● ○○○○○	3	4.2	(2.8)
Rotation	○○○○○ ○○●●○	1	1.1	(2.5)
	○○○○○ ○●○○●	1	2.5	(2.5)
	○○○○○ ○●○●○	4	8.9	(10.0)
	○○○○○ ○●●○○	6	23.5	(15.0)
	○○○○○ ●○●○○	3	9.5	(7.5)
	○○○○○ ●●○○○	16	32.9	(40.0)
	○○○●○ ○○●○○	4	10.0	(10.0)
	○○○●○ ○●○○○	1	1.6	(2.5)
	○○○●● ○○○○○	4	10.0	(10.0)

TABLE B.2: Détails des observations des accords sur pavé tactile par geste (2/3). Pour chaque geste sont regroupés l'ensemble des accords qui ont été utilisés au moins une fois. Les accords sont représentés par dix cercles correspondant (dans l'ordre) aux 5 doigts de la main gauche (de l'auriculaire au pouce) puis aux 5 doigts de la main droite (du pouce à l'auriculaire). Les cercles ● correspondent aux doigts impliqués dans l'accord. Nb. Obs. correspond au nombre d'observations. % correspond au taux d'occurrence pondéré par utilisateur. Nous donnons entre parenthèses le taux non-pondéré.

Geste	Accord	Nb. Obs.	%
Glissement à 3 doigts	○○○○○ ○●●●●	1	1.5 (1.5)
	○○○○○ ○●●●○	54	68.3 (81.8)
	○○○○○ ●●●○○	9	28.8 (13.6)
	○●●●○ ○○○○○	2	1.4 (3.0)
<i>Flicks</i> à 3 doigts	○○○○○ ○○●●○	2	0.8 (2.0)
	○○○○○ ○○●●●	3	10.0 (2.9)
	○○○○○ ○●●●○	91	81.1 (89.2)
	○○○●○ ○○●●○	2	6.7 (2.0)
	○●●●○ ○○○○○	4	1.5 (3.9)
<i>Flicks</i> à 4 doigts	○○○○○ ○●●●●	780	98.3 (98.5)
	●●●●○ ○○○○○	12	1.7 (1.5)
<i>Pinch-and-expand</i> à 4 ou 5 doigts	○○○○○ ○●●●●	29	59.2 (54.7)
	○○○○○ ●●●●○	2	16.7 (3.8)
	○○○○○ ●●●●●	22	24.1 (41.5)

TABLE B.3: Détails des observations des accords sur pavé tactile par geste (3/3). Pour chaque geste sont regroupés l'ensemble des accords qui ont été utilisés au moins une fois. Les accords sont représentés par dix cercles correspondant (dans l'ordre) aux 5 doigts de la main gauche (de l'auriculaire au pouce) puis aux 5 doigts de la main droite (du pouce à l'auriculaire). Les cercles ● correspondent aux doigts impliqués dans l'accord. Nb. Obs. correspond au nombre d'observations. % correspond au taux d'occurrence pondéré par utilisateur. Nous donnons entre parenthèses le taux non-pondéré.

Annexe C

Détails des taux de succès de reconnaissance

Dans cette annexe, nous présentons l'ensemble des taux de succès moyens de reconnaissance d'un doigt (*cf.* Section C.1) et d'un doigt et de son orientation (*cf.* Section C.2). Le calcul des taux de succès est présenté à la section 1.3.2.4.

C.1 Reconnaissance des doigts

Le taux de succès R_1^+ moyen est de 78%, R_2^+ vaut 61% et R_3^+ vaut 43%. La figure C.1 résume les taux de succès moyens pour $n = 1$, la figure C.2 pour $n = 2$ et la figure C.2 pour $n = 3$. La colonne de gauche correspond aux doigts de la main gauche et celle de droite à ceux de la main droite. Les lignes correspondent aux différents doigts des pouces (en haut) aux auriculaires (en bas). Les points rouges correspondent aux taux de succès pour reconnaître un doigt particulier à une orientation particulière (*ex.* reconnaître le majeur gauche lorsque celui-ci est à plat). En haut de chaque graphique est affiché D le taux de succès pour reconnaître le doigt associé (*ex.* reconnaître l'annulaire droit).

C.2 Reconnaissance des doigts et des orientations

Dans cette section, nous calculons des taux de succès pour à la fois reconnaître un doigt et son orientation avec l'empreinte digitale. Les calculs sont similaires, l'unique différence est que dans les calculs de $R_n^+(e)$, $R_n^-(e)$ et $R_n^?(e)$ (*cf.* Section C.1) les n empreintes les plus similaires doivent à la fois correspondre au même doigt mais aussi à la même orientation.

Le taux de succès R_1^+ moyen est de 72%, R_2^+ vaut 58% et R_3^+ vaut 40%. La figure C.4 résume les taux de succès moyens pour $n = 1$, la figure C.5 pour $n = 2$ et la figure C.5 pour $n = 3$. La colonne de gauche correspond aux

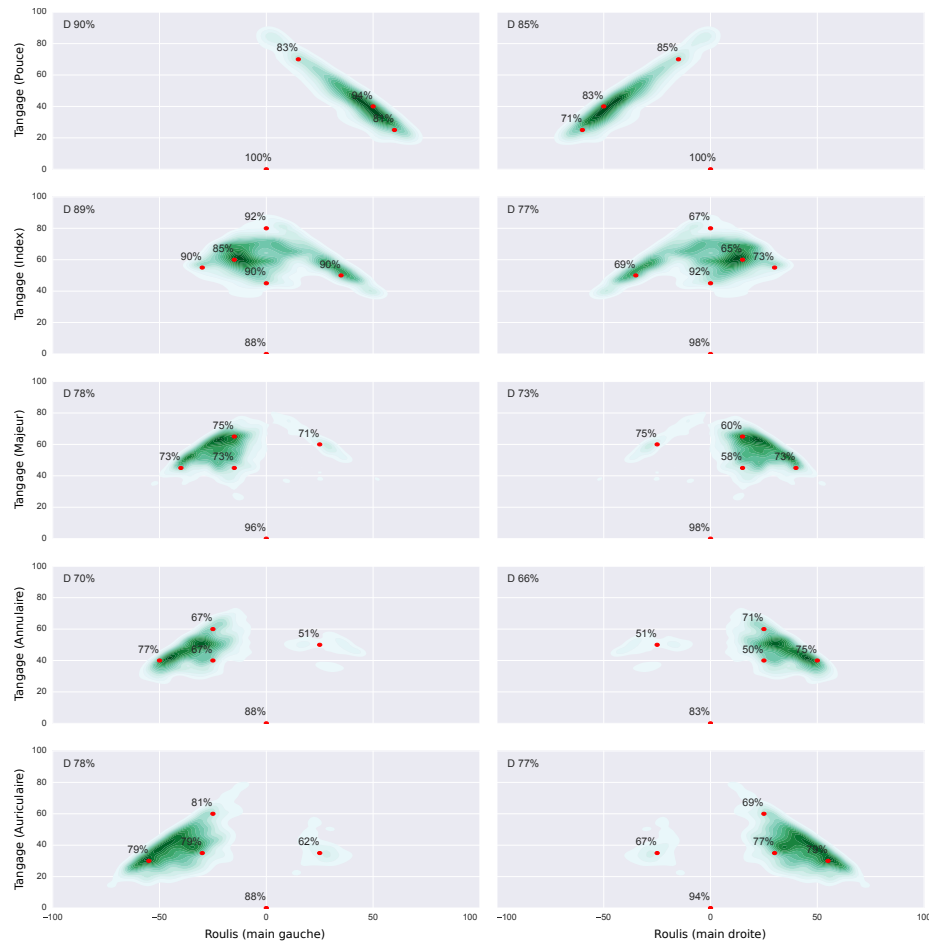


FIGURE C.1: Taux de succès moyen de reconnaissance du doigt pour $n = 1$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt pour l'orientation courante. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

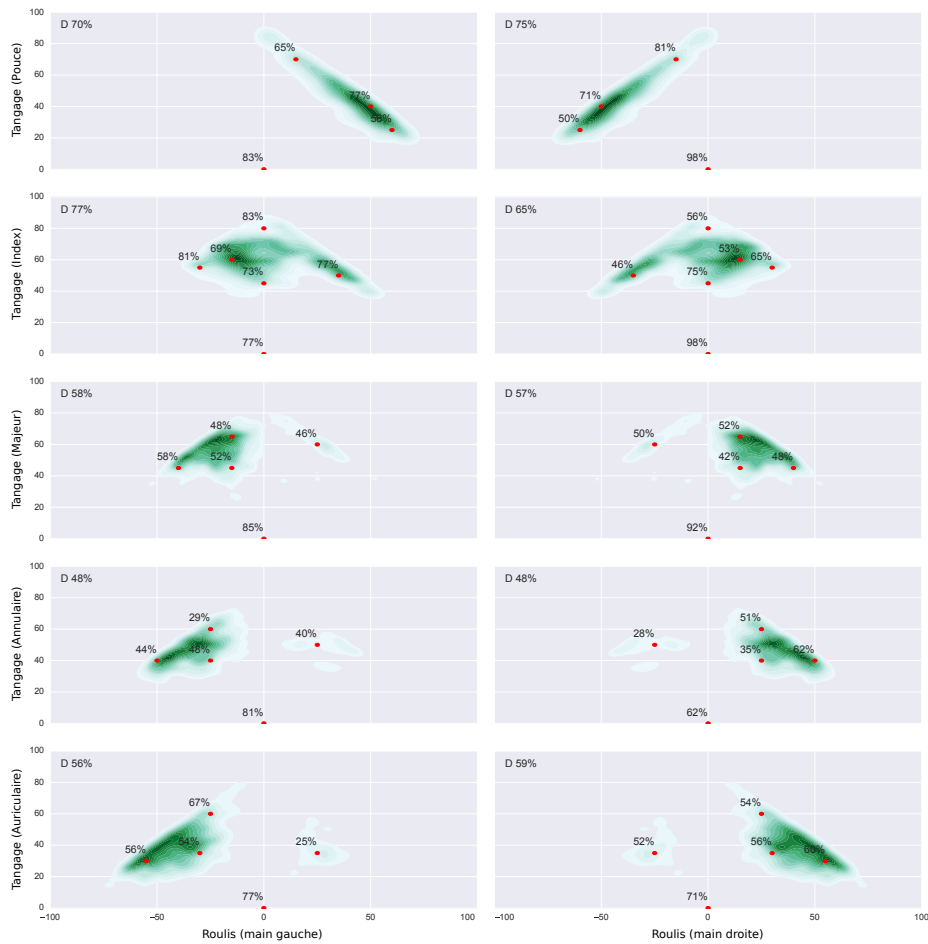


FIGURE C.2: Taux de succès moyen de reconnaissance du doigt pour $n = 2$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt pour l'orientation courante. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

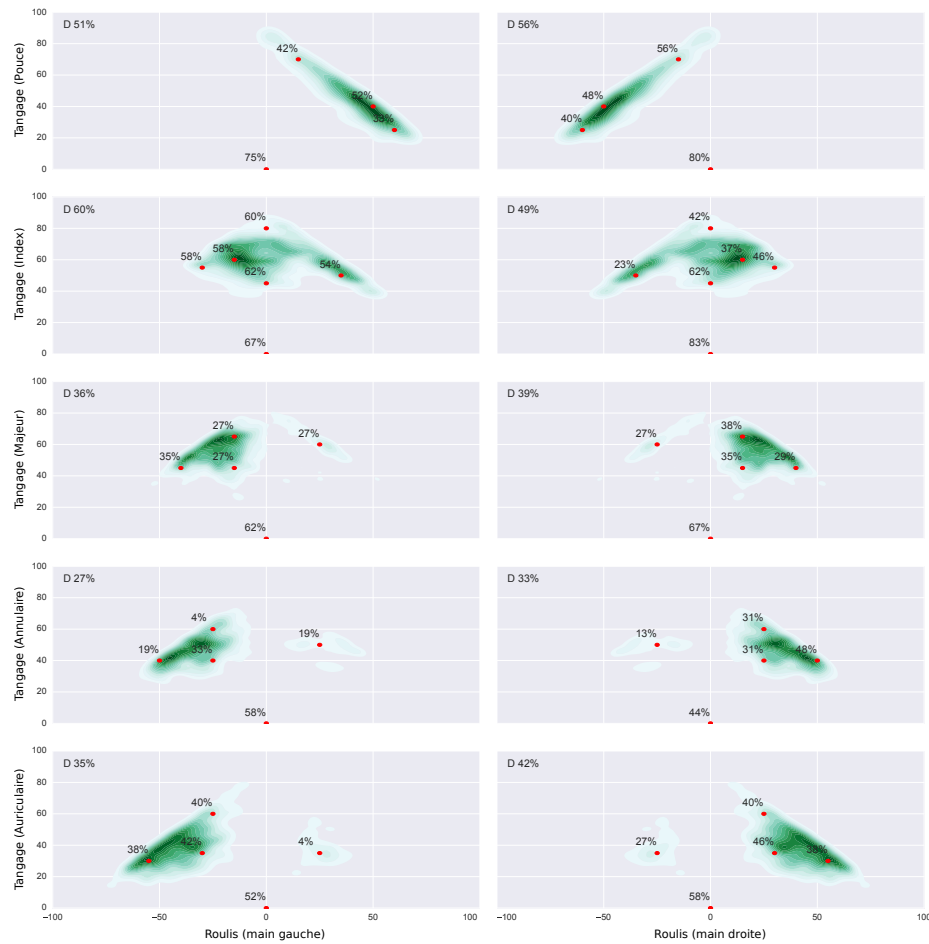


FIGURE C.3: Taux de succès moyen de reconnaissance du doigt pour $n = 3$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt pour l'orientation courante. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

doigts de la main gauche et celle de droite à ceux de la main droite. Les lignes correspondent aux différents doigts des pouces (en haut) aux auriculaires (en bas). Les points rouges correspondent aux taux de succès pour reconnaître un doigt et son orientation à une orientation particulière (*ex.* reconnaître le majeur gauche à plat lorsque celui-ci est à plat). En haut de chaque graphique est affiché $D + O$ le taux de succès pour reconnaître le doigt associé et ses orientations (*ex.* reconnaître l'annulaire droit et les différentes orientations testées). Nous affichons aussi D précédemment introduit : le taux de succès pour reconnaître le doigt associé (*ex.* reconnaître l'annulaire droit).

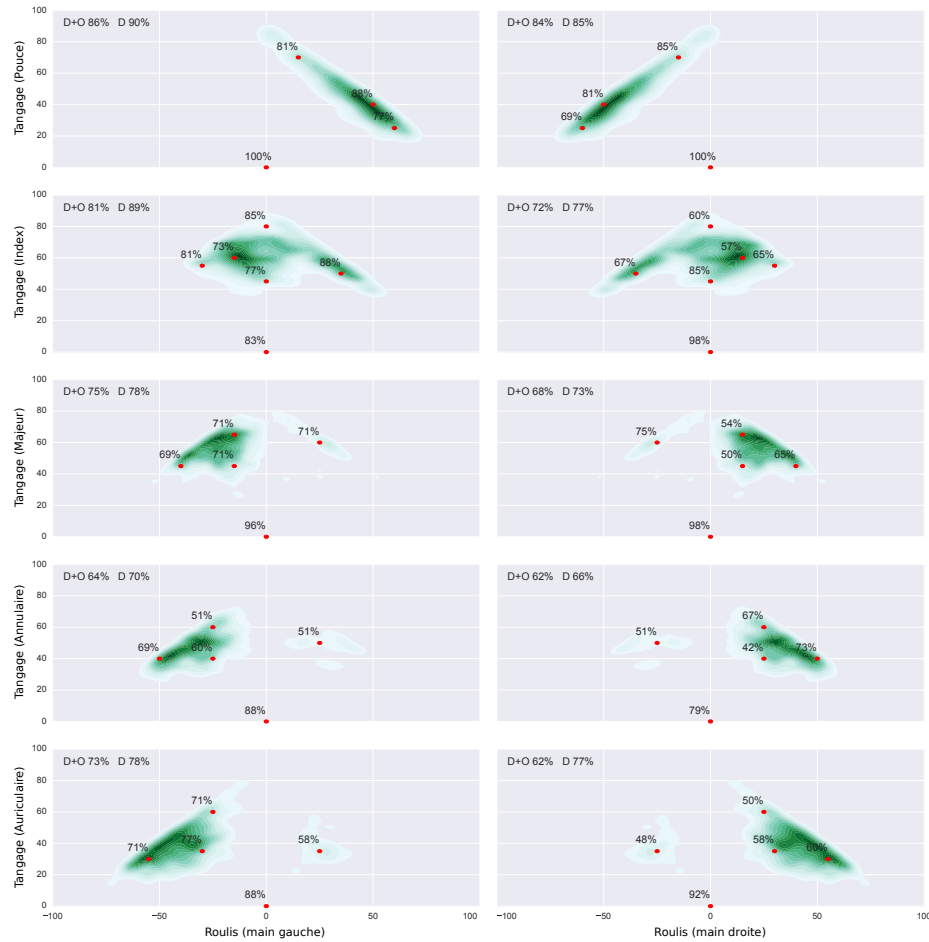


FIGURE C.4: Taux de succès moyen de reconnaissance du doigt et de l'orientation pour $n = 1$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt et de l'orientation courante. Le taux $D+O$ correspond au taux de succès moyen de reconnaissance du doigt et de l'orientation. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

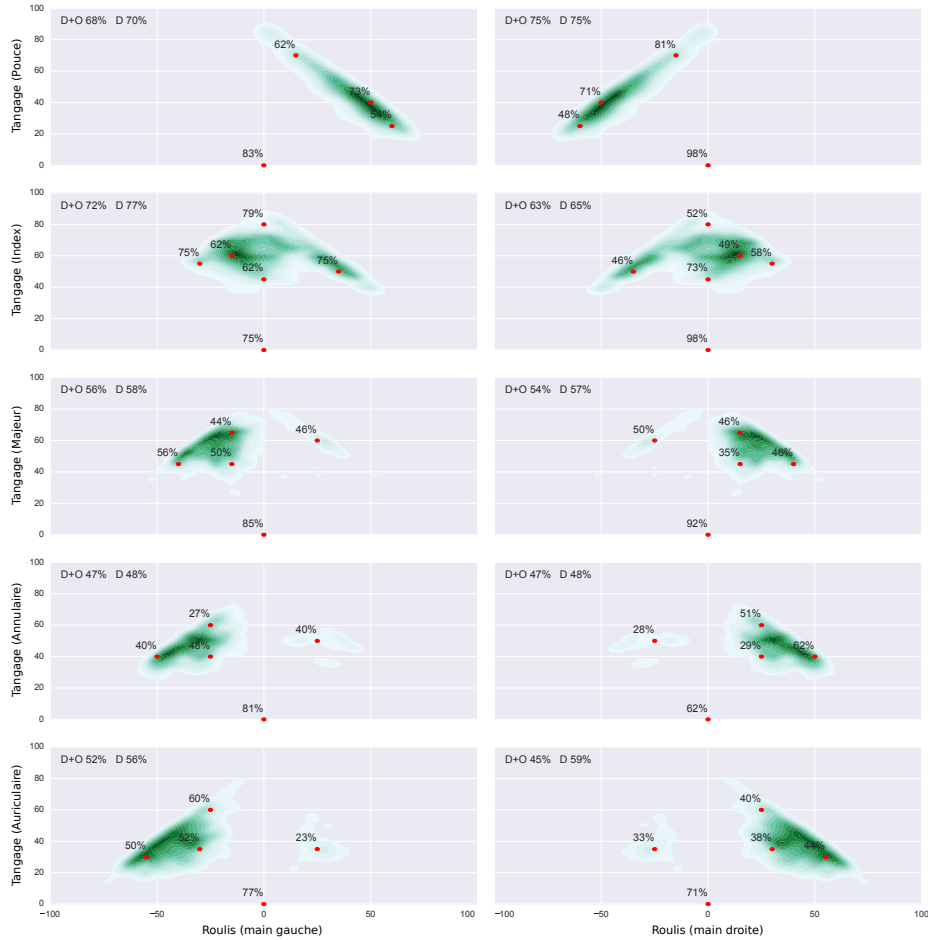


FIGURE C.5: Taux de succès moyen de reconnaissance du doigt et de l'orientation pour $n = 2$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt et de l'orientation courante. Le taux $D+O$ correspond au taux de succès moyen de reconnaissance du doigt et de l'orientation. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

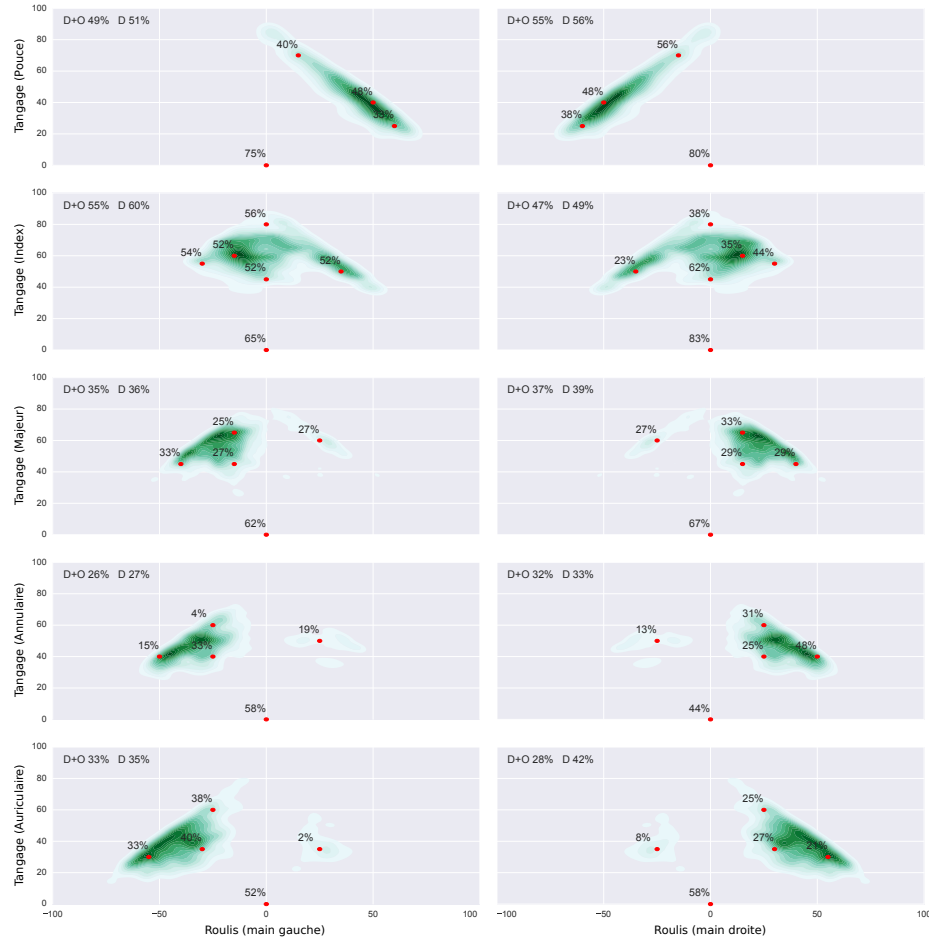


FIGURE C.6: Taux de succès moyen de reconnaissance du doigt et de l'orientation pour $n = 3$. La colonne de gauche (respectivement droite) correspond aux doigts de la main gauche (respectivement droite). Pour chaque point rouge, le taux indiqué correspond au taux de succès moyen de reconnaissance du doigt et de l'orientation courante. Le taux $D+O$ correspond au taux de succès moyen de reconnaissance du doigt et de l'orientation. Le taux D correspond au taux de succès moyen de reconnaissance du doigt.

Annexe D

Bibliographie

- [ABLM05] Caroline Appert, Michel Beaudouin-Lafon, and WendyE Mackay. Context matters : Evaluating interaction techniques with the cis model. In *Proc. HCI*, pages 279–295. Springer, 2005. URL : http://dx.doi.org/10.1007/1-84628-062-1_18.
- [ACVL14] Jeff Avery, Mark Choi, Daniel Vogel, and Edward Lank. Pinch-to-zoom-plus : An enhanced pinch-to-zoom that reduces clutching and panning. In *Proc. UIST*, pages 595–604. ACM, 2014. URL : <http://doi.acm.org/10.1145/2642918.2647352>.
- [AHJ13] Aitor Apaolaza, Simon Harper, and Caroline Jay. Understanding users in the wild. In *Proc. W4A*, pages 13 :1–13 :4. ACM, 2013. URL : <http://doi.acm.org/10.1145/2461121.2461133>, doi:10.1145/2461121.2461133.
- [App] Apple Inc. Pages for iOS (iPad) : Touchscreen basics. <https://support.apple.com/kb/PH3571>. Accessed : 2015-08-22.
- [AT10] Oscar Kin-Chung Au and Chiew-Lan Tai. Multitouch finger registration and its applications. In *Proc. OZCHI*, pages 41–48. ACM, 2010. URL : <http://doi.acm.org/10.1145/1952222.1952233>.
- [AZ99] Johnny Accot and Shumin Zhai. Performance evaluation of input devices in trajectory-based tasks : An application of the steering law. In *Proc. CHI*, pages 466–472. ACM, 1999. URL : <http://doi.acm.org/10.1145/302979.303133>, doi:10.1145/302979.303133.
- [AZ12] Shiri Azenkot and Shumin Zhai. Touch behavior with different postures on soft smartphone keyboards. In *Proc. MobileHCI*, pages 251–260. ACM, 2012. URL : <http://doi.acm.org/10.1145/2371574.2371612>.
- [BC96] R. D. Bahuguna and Tom Corboline. Prism fingerprint sensor that uses a holographic optical element. In *AO journal*, number 26, pages 5242–5245. OSA, 1996. URL : <http://ao.osa.org/abstract.cfm?URI=ao-35-26-5242>, doi:10.1364/AO.35.005242.

- [BJ98] Suresh K. Bhavnani and Bonnie E. John. Delegation and circumvention : Two faces of efficiency. In *Proc. CHI*, pages 273–280. ACM, 1998. URL : <http://dx.doi.org/10.1145/274644.274683>.
- [BJ00] Suresh K. Bhavnani and Bonnie E. John. The strategic use of complex computer systems. In *HCI journal*, pages 107–137. L. Erlbaum Associates Inc., 2000. URL : http://dx.doi.org/10.1207/S15327051HCI1523_3.
- [BL00] Michel Beaudouin-Lafon. Instrumental interaction : an interaction model for designing post-WIMP user interfaces. In *Proc. CHI*, pages 446–453. ACM, 2000. URL : <http://doi.acm.org/10.1145/332040.332473>.
- [BLD⁺11] Nikola Banovic, Frank Chun Yat Li, David Dearman, Koji Yatani, and Khai N. Truong. Design of unimanual multi-finger pie menu interaction. In *Proc. ITS*, pages 120–129. ACM, 2011. URL : <http://doi.acm.org/10.1145/2076354.2076378>.
- [BLG10] Gilles Bailly, Eric Lecolinet, and Yves Guiard. Finger-count & radial-stroke shortcuts : 2 techniques for augmenting linear menus on multi-touch surfaces. In *Proc. CHI*, pages 591–594. ACM, 2010. URL : <http://doi.acm.org/10.1145/1753326.1753414>, doi:10.1145/1753326.1753414.
- [BLZ13] Xiaojun Bi, Yang Li, and Shumin Zhai. Fitts law : Modeling finger touch with fitts’ law. In *Proc. CHI*, pages 1363–1372. ACM, 2013. URL : <http://doi.acm.org/10.1145/2470654.2466180>.
- [BML12] Gilles Bailly, JöRg Müller, and Eric Lecolinet. Design and evaluation of finger-count interaction : Combining multitouch gestures and menus. In *IJHCS journal*, pages 673–689. Elsevier, 2012. URL : <http://dx.doi.org/10.1016/j.ijhcs.2012.05.006>, doi:10.1016/j.ijhcs.2012.05.006.
- [BPO⁺15] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauff. Performance and ergonomics of touch surfaces : A comparative study using biomechanical simulation. In *Proc. CHI*, pages 1817–1826. ACM, 2015. URL : <http://doi.acm.org/10.1145/2702123.2702607>.
- [BSMT09] Hrvoje Benko, T. Scott Saponas, Dan Morris, and Desney Tan. Enhancing input on and above the interactive surface with muscle sensing. In *Proc. ITS*, pages 93–100. ACM, 2009. URL : <http://doi.acm.org/10.1145/1731903.1731924>.
- [BSP⁺93] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses : the see-through interface. In *Proc. SIGGRAPH*, pages 73–80. ACM, 1993. URL : <http://doi.acm.org/10.1145/166117.166126>.
- [BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf : Speeded up robust features. In *Proc. ECCV*, pages 404–417. Springer Berlin

- Heidelberg, 2006. URL : http://dx.doi.org/10.1007/11744023_32, doi:10.1007/11744023_32.
- [Bux86] William Buxton. Chunking and phrasing and the design of human-computer dialogues. In *Proc. IFIP WCC*, pages 475–480. North Holland Publishers, 1986. URL : <http://www.dgp.toronto.edu/OTP/papers/bill.buxton/chunking.html>.
- [CAG12] A. Cockburn, D. Ahlström, and C. Gutwin. Understanding performance in touch selections : Tap, drag and radial pointing drag with finger, stylus and mouse. In *IJHCS journal*, pages 218–233. Academic Press, Inc., 2012. URL : <http://dx.doi.org/10.1016/j.ijhcs.2011.11.002>.
- [CBBL07] Olivier Chapuis, Renaud Blanch, and Michel Beaudouin-Lafon. Fitts’ law in the wild : A field study of aimed movements. In *LRI technical report*, pages 1–11, 2007. URL : <https://hal.archives-ouvertes.fr/hal-00612026>.
- [CGSM14] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. Supporting novice to expert transitions in user interfaces. In *Computing Surveys journal*, pages 31.1–31.36. ACM, 2014. URL : <http://doi.acm.org/10.1145/2659796>, doi:10.1145/2659796.
- [CH04] Herbert A Colle and Keith J Hiszem. Standing at a kiosk : Effects of key size and spacing on touch screen numeric keypad performance and user preference. In *Ergonomics journal*, pages 1406–1423. Taylor & Francis, 2004. URL : <http://www.tandfonline.com/doi/abs/10.1080/00140130410001724228>.
- [CH14] Ashley Colley and Jonna Häkkinä. Exploring finger specific touch screen interaction for mobile phone user interfaces. In *Proc. OZCHI*, pages 539–548. ACM, 2014. URL : <http://doi.acm.org/10.1145/2686612.2686699>.
- [CMN80] Stuart K Card, Thomas P Moran, and Allen Newell. The keystroke-level model for user performance time with interactive systems. In *Communications of the ACM*, pages 396–410. ACM, 1980. URL : <http://doi.acm.org/10.1145/358886.358895>.
- [CWB⁺08] Xiang Cao, A. D. Wilson, R. Balakrishnan, K. Hinckley, and S. E. Hudson. Shapetouch : Leveraging contact shape on interactive surfaces. In *Proc. TABLETOP*, pages 129–136. IEEE, 2008. URL : <http://dx.doi.org/10.1109/TABLETOP.2008.4660195>, doi:10.1109/TABLETOP.2008.4660195.
- [DAJ⁺16] Jonathan Deber, Bruno Araujo, Ricardo Jota, Clifton Forlines, Darren Leigh, Steven Sanders, and Daniel Wigdor. Hammer time! : A low-cost, high precision, high accuracy tool to measure the latency of touchscreen devices. In *Proc. CHI*, pages 2857–2868. ACM, 2016. URL : <http://doi.acm.org/10.1145/2858036.2858394>, doi:10.1145/2858036.2858394.

- [EKF12] Philipp Ewerling, Alexander Kulik, and Bernd Froehlich. Finger and hand detection for multi-touch interfaces based on maximally stable extremal regions. In *Proc. ITS*, pages 173–182. ACM, 2012. URL : <http://doi.acm.org/10.1145/2396636.2396663>.
- [FWSB07] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proc. CHI*, pages 647–656. ACM, 2007. URL : <http://doi.acm.org/10.1145/1240624.1240726>.
- [GB16] Aakar Gupta and Ravin Balakrishnan. Dualkey : Miniature screen text entry via finger identification. In *Proc. CHI*, pages 59–70. ACM, 2016. URL : <http://doi.acm.org/10.1145/2858036.2858052>, doi:10.1145/2858036.2858052.
- [GCP⁺14a] Alix Goguey, Géry Casiez, Thomas Pietrzak, Daniel Vogel, and Nicolas Roussel. Adoiraccourcix : Multi-touch command selection using finger identification. In *Proc. IHM*, pages 28–37. ACM, 2014. URL : <http://doi.acm.org/10.1145/2670444.2670446>, doi:10.1145/2670444.2670446.
- [GCP⁺14b] Alix Goguey, Géry Casiez, Thomas Pietrzak, Daniel Vogel, and Nicolas Roussel. Sélection de commandes sur tables interactives multi-points par identification des doigts, 2014. URL : <https://hal.archives-ouvertes.fr/hal-01089614>.
- [GCS⁺14a] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. Faster command selection on tablets with fasttap. In *Proc. CHI*, pages 2617–2626. ACM, 2014. URL : <http://doi.acm.org/10.1145/2556288.2557136>.
- [GCS⁺14b] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. Faster command selection on tablets with fast-tap. In *Proc. CHI*, pages 2617–2626. ACM, 2014. URL : <http://doi.acm.org/10.1145/2556288.2557136>, doi:10.1145/2556288.2557136.
- [GCV⁺14] Alix Goguey, Géry Casiez, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak, and Nicolas Roussel. A three-step interaction pattern for improving discoverability in finger identification techniques. In *Proc. Adjunct UIST*, pages 33–34. ACM, 2014. URL : <http://doi.acm.org/10.1145/2658779.2659100>, doi:10.1145/2658779.2659100.
- [GDB07] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. Strategies for accelerating on-line learning of hotkeys. In *Proc. CHI*, pages 1591–1600. ACM, 2007. URL : <http://doi.acm.org/10.1145/1240624.1240865>.
- [GFH⁺12] Emilien Ghomi, Guillaume Faure, Stéphane Huot, Olivier Chapuis, and Michel Beaudouin-Lafon. Using rhythmic patterns as an input method. In *Proc. CHI*, pages 1253–1262. ACM, 2012. URL : <http://doi.acm.org/10.1145/2207676.2208579>.

- [GHB⁺13] Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E. Mackay. Arpège : Learning multitouch chord gestures vocabularies. In *Proc. ITS*, pages 209–218. ACM, 2013. URL : <http://doi.acm.org/10.1145/2512349.2512795>.
- [GMW05] François Guimbretière, Andrew Martin, and Terry Winograd. Benefits of merging command selection and direct manipulation. In *ToCHI journal*, pages 460–476. ACM, 2005. URL : <http://doi.acm.org/10.1145/1096737.1096742>.
- [GNCV16] Alix Goguey, Mathieu Nancel, Géry Casiez, and Daniel Vogel. The performance and preference of different fingers and chords for pointing, dragging, and object transformation. In *Proc. CHI*, pages 4250–4261. ACM, 2016. URL : <http://doi.acm.org/10.1145/2858036.2858194>, doi:10.1145/2858036.2858194.
- [GWC15] Alix Goguey, Julie Wagner, and Géry Casiez. Quantifying object- and command-oriented interaction. In *Proc. INTERACT*, pages 231–239. Springer International Publishing, 2015. URL : http://dx.doi.org/10.1007/978-3-319-22723-8_18, doi:10.1007/978-3-319-22723-8_18.
- [Han05] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. UIST*, pages 115–118. ACM, 2005. URL : <http://doi.acm.org/10.1145/1095034.1095054>, doi:10.1145/1095034.1095054.
- [HB11] Christian Holz and Patrick Baudisch. Understanding touch. In *Proc. CHI*, pages 2501–2510. ACM, 2011. URL : <http://doi.acm.org/10.1145/1978942.1979308>.
- [HB13] Christian Holz and Patrick Baudisch. Fiberio : a touchscreen that senses fingerprints. In *Proc. UIST*, pages 41–50. ACM, 2013. URL : <http://dl.acm.org/citation.cfm?id=2501988.2502021>.
- [HBRG05] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proc. CHI*, pages 451–460. ACM, 2005. URL : <http://doi.acm.org/10.1145/1054972.1055035>, doi:10.1145/1054972.1055035.
- [HCRC88] 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. In *Journal of applied psychology*, pages 711–720. American Psychological Association, 1988. URL : <http://psycnet.apa.org/journals/apl/73/4/711.pdf>.
- [HH10] Morten Hertzum and Kasper Hornbæk. How age affects pointing with mouse and touchpad : A comparison of young, adult, and elderly users. In *IJHCI journal*, pages 703–734. Taylor & Francis, 2010. URL : <http://www.tandfonline.com/doi/abs/10.1080/10447318.2010.487198>, doi:10.1080/10447318.2010.487198.

- [HH12] Chris Harrison and Scott Hudson. Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In *Proc. CHI*, pages 3149–3152. ACM, 2012. URL : <http://doi.acm.org/10.1145/2207676.2208730>, doi:10.1145/2207676.2208730.
- [HHN85] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. In *HCI journal*, pages 311–338. L. Erlbaum Associates Inc., 1985. URL : http://dx.doi.org/10.1207/s15327051hci0104_2.
- [HL11] Seongkook Heo and Geehyuk Lee. Force gestures : Augmenting touch screen gestures with normal and tangential forces. In *Proc. UIST*, pages 621–626. ACM, 2011. URL : <http://doi.acm.org/10.1145/2047196.2047278>, doi:10.1145/2047196.2047278.
- [HS84] M Hase and A Shimisu. Entry method of fingerprint image using a prism. In *TIECEJ journal*, pages 627–628, 1984.
- [HSH11] Chris Harrison, Julia Schwarz, and Scott E. Hudson. TapSense : enhancing finger interaction on touch surfaces. In *Proc. UIST*, pages 627–636. ACM, 2011. URL : <http://doi.acm.org/10.1145/2047196.2047279>, doi:10.1145/2047196.2047279.
- [HSP12] Chris Harrison, Munehiko Sato, and Ivan Poupyrev. Capacitive fingerprinting : Exploring user differentiation by sensing electrical properties of the human body. In *Proc. UIST*, pages 537–544. ACM, 2012. URL : <http://doi.acm.org/10.1145/2380116.2380183>, doi:10.1145/2380116.2380183.
- [HWO⁺13] Eve Hoggan, John Williamson, Antti Oulasvirta, Miguel Nacenta, Per Ola Kristensson, and Anu Lehtiö. Multi-touch rotation gestures : Performance and ergonomics. In *Proc. CHI*, pages 3047–3050. ACM, 2013. URL : <http://doi.acm.org/10.1145/2470654.2481423>.
- [HXSH14] Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. Touchtools : Leveraging familiarity and skill with physical tools to augment touch interaction. In *Proc. CHI*, pages 2913–2916. ACM, 2014. URL : <http://doi.acm.org/10.1145/2556288.2557012>, doi:10.1145/2556288.2557012.
- [Ibr16] Noor A Ibraheem. Finger identification and gesture recognition using gaussian classifier model. In *IJAER journal*, pages 6924–6931. Research India Publications, 2016. URL : http://www.ripublication.com/ijaer16/ijaerv11n10_08.pdf.
- [ISO] ISO ISO. 9241-9 ergonomic requirements for office work with visual display terminals (vdts)-part 9 : Requirements for non-keyboard input devices (fdis-final draft international standard), 2000. In *International Organization for Standardization*. URL : http://www.iso.org/iso/catalogue_detail.htm?csnumber=30030.
- [KB91] Gordon Kurtenbach and William Buxton. Issues in combining marking and direct manipulation techniques. In *Proc. CHI*, pages 137–

144. ACM, 1991. URL : <http://doi.acm.org/10.1145/120782.120797>.
- [KB94] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proc. CHI*, pages 258–264. ACM, 1994. URL : <http://doi.acm.org/10.1145/191666.191759>.
 - [KBBC05] Martin Kaltenbrunner, Till Bovermann, Ross Bencina, and Enrico Costanza. Tuio - a protocol for table based tangible user interfaces. In *Proc. GW*, 2005. URL : <http://www.tuio.org>.
 - [KHA11] Kenrick Kin, Björn Hartmann, and Maneesh Agrawala. Two-handed marking menus for multitouch devices. In *ToCHI journal*, pages 16 :1–16 :23. ACM, 2011. URL : <http://doi.acm.org/10.1145/1993060.1993066>.
 - [KKS⁺12] Peter Kung, Dominik Küser, Craig Schroeder, Tony DeRose, Donald Greenberg, and Kenrick Kin. An augmented multi-touch system using hand and finger identification. In *EA CHI*, pages 1431–1432. ACM, 2012. URL : <http://doi.acm.org/10.1145/2212776.2212471>, doi:10.1145/2212776.2212471.
 - [KMB⁺11] Kenrick Kin, Tom Miller, Björn Bollensdorff, Tony DeRose, Björn Hartmann, and Maneesh Agrawala. Eden : A professional multitouch tool for constructing virtual organic environments. In *Proc. CHI*, pages 1343–1352. ACM, 2011. URL : <http://doi.acm.org/10.1145/1978942.1979141>, doi:10.1145/1978942.1979141.
 - [Kot14] Stephan Kotin. A comparison of user preference for mouse and touchpad with windows 8 interaction. In *Proc. HFES*, pages 1114–1118. SAGE, 2014. URL : <http://pro.sagepub.com/content/58/1/1114.full.pdf>, doi:10.1177/1541931214581233.
 - [KSR06] Urmila Kukreja, William Stevenson, and Frank Ritter. Rui : Recording user input from interfaces under windows and mac os x. In *BRM journal*, pages 656–659. Springer, 2006. URL : <http://link.springer.com/article/10.3758/BF03193898>, doi:10.3758/BF03193898.
 - [LGF10] G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. The design and evaluation of multitouch marking menus. In *Proc. CHI*, pages 2233–2242. ACM, 2010. URL : <http://doi.acm.org/10.1145/1753326.1753663>.
 - [LGK15] Sandi Ljubic, Vlado Glavinic, and Mihael Kuvec. Finger-based pointing performance on mobile touchscreen devices : Fitts’ law fits. In *Proc. UAHCI*, pages 318–329. Springer International Publishing, 2015. URL : http://dx.doi.org/10.1007/978-3-319-20678-3_31.
 - [LSR⁺15] Ahreum Lee, Kiburum Song, Hokyoung Blake Ryu, Jieun Kim, and Gyuhyun Kwon. Fingerstroke time estimates for touchscreen-based mobile gaming interaction. In *HMS journal*, pages 211–224. Elsevier,

2015. URL : <http://www.sciencedirect.com/science/article/pii/S0167945715300373>.
- [LV14] Yuexing Luo and Daniel Vogel. Crossing-based Selection with Direct Touch Input. In *Proc. CHI*, pages 2627–2636. ACM, 2014. URL : <http://doi.acm.org/10.1145/2556288.2557397>.
- [Mac89] I Scott MacKenzie. A note on the information-theoretic basis for fitts' law. In *Journal of motor behavior*, pages 323–330. Taylor & Francis, 1989. URL : <http://www.tandfonline.com/doi/abs/10.1080/00222895.1989.10735486>.
- [Mac92] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. In *HCI journal*, pages 91–139. L. Erlbaum Associates Inc., 1992. URL : http://dx.doi.org/10.1207/s15327051hci0701_3.
- [Mac02] Wendy Mackay. Which interaction technique works when? Floating palettes, marking menus and toolglasses support different task strategies. In *Proc. AVI*, pages 203–208. ACM, 2002. URL : <http://doi.acm.org/10.1145/1556262.1556294>.
- [Mac13] I Scott MacKenzie. A note on the validity of the shannon formulation for fitts' index of difficulty. In *OJAPPS journal*, pages 360–368. Scientific Research Publishing, 2013. URL : <http://file.scirp.org/Html/38031.html>.
- [MBH⁺13] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting hotkey use through rehearsal with ExposeHK. In *Proc. CHI*, pages 573–582. ACM, 2013. URL : <http://doi.acm.org/10.1145/2470654.2470735>.
- [MCPR13] Jonathan Morgan, Chen-Yang Cheng, Christopher Pike, and Frank Ritter. A design, tests and considerations for improving keystroke and mouse loggers. In *IC journal*, pages 242–258. Oxford University Press, 2013. URL : <http://iwc.oxfordjournals.org/content/25/3/242.abstract>, doi:10.1093/iwc/iws014.
- [MDLB13] Max Möllers, Norbert Dumont, Stefan Ladwig, and Jan Borchers. Improving touch accuracy on large tabletops using predecessor and successor. In *Proc. CHI*, pages 755–758. ACM, 2013. URL : <http://doi.acm.org/10.1145/2470654.2470760>.
- [MGBC16] Sylvain Malacria, Alix Goguey, Gilles Bailly, and G ry Casiez.  tude de terrain sur l'utilisation des pav s tactiles. In *To appear in Proc. IHM*. ACM, 2016.
- [MH06] Tomer Moscovich and John F. Hughes. Multi-finger cursor techniques. In *Proc. GI*, pages 1–7. Canadian Information Processing Society, 2006. URL : <http://dl.acm.org/citation.cfm?id=1143079.1143081>.
- [MKG86] IC McManus, R.I Kemp, and J Grant. Differences between fingers and hands in tapping ability : dissociation between speed

- and regularity. In *Cortex journal*, pages 461–473. Elsevier, 1986. URL : <http://www.sciencedirect.com/science/article/pii/S0010945286800089>.
- [MKL⁺11] Nicolai Marquardt, Johannes Kiemer, David Ledo, Sebastian Boring, and Saul Greenberg. Designing user-, hand-, and handpart-aware tabletop interactions with the TouchID toolkit. In *Proc. ITS*, pages 21–30. ACM, 2011. URL : <http://doi.acm.org/10.1145/2076354.2076358>.
- [ML09] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. VISAPP*, pages 331–340. INSTICC Press, 2009. URL : <http://www.cs.ubc.ca/~lowe/papers/09muja.pdf>.
- [MLH⁺16] K. Ma, J. Li, J. Han, Y. Hu, and W. Xu. Enhancing touchscreen input via finger identification. In *Proc. CSTIC*, pages 1–3. IEEE, 2016. URL : <http://dx.doi.org/10.1109/CSTIC.2016.7463927>, doi:10.1109/CSTIC.2016.7463927.
- [MMJP09] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. Handbook of fingerprint recognition. Springer Science & Business Media, 2009.
- [MRB05] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proc. UIST*, pages 43–52. ACM, 2005. URL : <http://doi.acm.org/10.1145/1095034.1095042>.
- [MSC⁺13] Sylvain Malacria, Joey Scarr, Andy Cockburn, Carl Gutwin, and Tovi Grossman. Skillometers : Reflective widgets that motivate and help users to improve performance. In *Proc. UIST*, pages 321–330. ACM, 2013. URL : <http://doi.acm.org/10.1145/2501988.2501996>.
- [MSY07] Mark Micire, Martin Schedlbauer, and Holly Yanco. Horizontal selection : An evaluation of a digital tabletop input device. In *Proc. AMCIS*, pages 164–170. AIS, 2007. URL : <http://robotics.cs.uml.edu/fileadmin/content/publications/2007/MJM.AMCIS.2007.MTU.final.pdf>.
- [MVER12] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. Extended multitouch : recovering touch posture and differentiating users using a depth camera. In *Proc. UIST*, pages 487–496. ACM, 2012. URL : <http://doi.acm.org/10.1145/2380116.2380177>.
- [NBW14] Alexander Ng, Stephen A. Brewster, and John H. Williamson. Investigating the effects of encumbrance on one- and two- handed interactions with mobile devices. In *Proc. CHI*, pages 1981–1990. ACM, 2014. URL : <http://doi.acm.org/10.1145/2556288.2557312>.
- [NLW⁺12] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. Designing for low-latency direct-touch input. In *Proc.*

- UIST*, pages 453–464. ACM, 2012. URL : <http://doi.acm.org/10.1145/2380116.2380174>.
- [PFB04] Barry A. Po, Brian D. Fisher, and Kellogg S. Booth. Mouse and touchscreen selection in the upper and lower visual fields. In *Proc. CHI*, pages 359–366. ACM, 2004. URL : <http://doi.acm.org/10.1145/985692.985738>.
- [PH08] Keith B. Perry and Juan Pablo Hourcade. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proc. GI*, pages 57–64. Canadian Information Processing Society, 2008. URL : <http://dl.acm.org/citation.cfm?id=1375714.1375725>.
- [PH10] Yong S Park and Sung H Han. One-handed thumb interaction of mobile devices from the input accuracy perspective. In *International Journal of Industrial Ergonomics*, pages 746–756. Elsevier, 2010. URL : <http://www.sciencedirect.com/science/article/pii/S0169814110000806>.
- [PKB06] Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. Target size study for one-handed thumb use on small touchscreen devices. In *Proc. MobileHCI*, pages 203–210. ACM, 2006. URL : <http://doi.acm.org/10.1145/1152215.1152260>.
- [PWS88] R. L. Potter, L. J. Weldon, and B. Shneiderman. Improving the accuracy of touch screens : an experimental evaluation of three strategies. In *Proc. CHI*, pages 27–32. ACM, 1988. URL : <http://portal.acm.org/citation.cfm?id=57167.57171>.
- [Ras00] Jef Raskin. 3-2-3 Modes and Quasimodes. In *The humane interface : new directions for designing interactive systems*, pages 55–59. Addison-Wesley, 2000. URL : http://en.wikipedia.org/wiki/The_Humane_Interface.
- [RGB⁺15] Quentin Roy, Yves Guiard, Gilles Bailly, Eric Lecolinet, and Olivier Rioul. Glass+skin : An empirical evaluation of the added value of finger identification to basic single-touch interaction on touch screens. In *Proc INTERACT*, pages 55–71. Springer, 2015. URL : http://link.springer.com/chapter/10.1007/978-3-319-22723-8_5.
- [RLG09] Anne Roudaut, Eric Lecolinet, and Yves Guiard. Microrolls : Expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proc. CHI*, pages 927–936. ACM, 2009. URL : <http://doi.acm.org/10.1145/1518701.1518843>, doi:10.1145/1518701.1518843.
- [Sch77] Michael Schiller. Finger identification, 1977. US Patent 4,053,228. URL : <https://www.google.com/patents/US4053228>.
- [Sch82] M. Schiller. Finger identification, 1982. US Patent 4,322,163. URL : <https://www.google.com/patents/US4322163>.
- [Sch85] M. Schiller. Finger identification, 1985. US Patent 4,544,267. URL : <https://www.google.com/patents/US4544267>.

- [SK98] Atsushi Sugiura and Yoshiyuki Koseki. A user interface using fingerprint recognition : holding commands and data objects on fingers. In *Proc. UIST*, pages 71–79. ACM, 1998. URL : <http://doi.acm.org/10.1145/288392.288575>.
- [SLG13] Marcos Serrano, Eric Lecolinet, and Yves Guiard. Bezel-tap gestures : quick activation of commands from sleep mode on tablets. In *Proc. CHI*, pages 3027–3036. ACM, 2013. URL : <http://doi.acm.org/10.1145/2470654.2481421>.
- [SM15] I. Scott MacKenzie. Fitts’ throughput and the remarkable case of touch-based target selection. In *Proc. HCI*, pages 238–249. Springer International Publishing, 2015. URL : http://dx.doi.org/10.1007/978-3-319-20916-6_23.
- [SMS09] Farzan Sasangohar, I Scott MacKenzie, and Stacey D Scott. Evaluation of mouse and touch input for a tabletop display using fitts’ reciprocal tapping task. In *Proc. HFES*, pages 839–843. Sage Publications, 2009. URL : <http://www.yorku.ca/mack/hfes2009.html>.
- [SS91] Andrew Sears and Ben Shneiderman. High precision touchscreens : Design strategies and comparisons with a mouse. In *IJHCS journal*, pages 593–613. Academic Press Ltd., 1991. URL : [http://dx.doi.org/10.1016/0020-7373\(91\)90037-8](http://dx.doi.org/10.1016/0020-7373(91)90037-8).
- [VC12] Daniel Vogel and Géry Casiez. Hand occlusion on a multi-touch tabletop. In *Proc. CHI*, pages 2307–2316. ACM, 2012. URL : <http://doi.acm.org/10.1145/2207676.2208390>.
- [VF13] Katia Vega and Hugo Fuks. Beauty tech nails : Interactive technology at your fingertips. In *Proc. TEI*, pages 61–64. ACM, 2013. URL : <http://doi.acm.org/10.1145/2540930.2540961>.
- [Wal12] Geoff Walker. A review of technologies for sensing contact location on the surface of a display. In *SID journal*, pages 413–440, 2012. URL : <http://dx.doi.org/10.1002/jsid.100>, doi:10.1002/jsid.100.
- [WBP⁺11] Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. Rock & rails : Extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *Proc. CHI*, pages 1581–1590. ACM, 2011. URL : <http://doi.acm.org/10.1145/1978942.1979173>, doi:10.1145/1978942.1979173.
- [WC04] Jingtao Wang and John Canny. FingerSense : augmenting expressiveness to physical pushing button by fingertip identification. In *Proc. CHI EA*, pages 1267–1270. ACM, 2004. URL : <http://doi.acm.org/10.1145/985921.986040>.
- [Wes99] Wayne Westerman. *Hand tracking, finger identification, and chordic manipulation on a multi-touch surface*. PhD thesis, University of Delaware, 1999. URL : <http://www.eecis.udel.edu/~westerma/main.pdf>.

- [WFGH11] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proc. CHI*, pages 143–146. ACM, 2011. URL : <http://doi.acm.org/10.1145/1978942.1978963>.
- [WHM12] Julie Wagner, Stéphane Huot, and Wendy Mackay. Bitouch and bipad : designing bimanual interaction for hand-held tablets. In *Proc. CHI*, pages 2317–2326. ACM, 2012. URL : <http://doi.acm.org/10.1145/2207676.2208391>.
- [WLS14] Julie Wagner, Eric Lecolinet, and Ted Selker. Multi-finger chords for hand-held tablets : Recognizable and memorable. In *Proc. CHI*, pages 2883–2892. ACM, 2014. URL : <http://doi.acm.org/10.1145/2556288.2556958>.
- [WR09a] Feng Wang and Xiangshi Ren. Empirical evaluation for finger input properties in multi-touch interaction. In *Proc. CHI*, pages 1063–1072. ACM, 2009. URL : <http://doi.acm.org/10.1145/1518701.1518864>.
- [WR09b] Feng Wang and Xiangshi Ren. Empirical Evaluation for Finger Input Properties in Multi-touch Interaction. In *Proc. CHI*, pages 1063–1072. ACM, 2009. URL : <http://doi.acm.org/10.1145/1518701.1518864>.
- [XSH15] Robert Xiao, Julia Schwarz, and Chris Harrison. Estimating 3d finger angle on commodity touchscreens. In *Proc ITS*, pages 47–50. ACM, 2015. URL : <http://doi.acm.org/10.1145/2817721.2817737>, doi:10.1145/2817721.2817737.
- [YBL⁺15] M. Yousefpor, J.M. Bussat, B.B. Lyon, G. Gozzini, S.P. Hotelling, and D. Setlak. Fingerprint sensor in an electronic device, 2015. US Patent App. 14/451,076. URL : <http://www.google.com/patents/US20150036065>.
- [ZSB15] Jian Zhao, R. William Soukoreff, and Ravin Balakrishnan. Exploring and modeling unimanual object manipulation on multi-touch displays. In *IJHCS journal*, pages 68–80, 2015. URL : <http://www.sciencedirect.com/science/article/pii/S1071581915000440>.
- [ZV16] Jingjie Zheng and Daniel Vogel. Finger-aware shortcuts. In *Proc. CHI*, pages 4274–4285. ACM, 2016. URL : <http://doi.acm.org/10.1145/2858036.2858355>, doi:10.1145/2858036.2858355.