



HAL
open science

Nouvelles interactions physiques pour dispositifs mobiles

Mathias Baglioni

► **To cite this version:**

Mathias Baglioni. Nouvelles interactions physiques pour dispositifs mobiles. Autre. Télécom Paris-Tech, 2012. Français. NNT : 2012ENST0018 . pastel-00913690

HAL Id: pastel-00913690

<https://pastel.hal.science/pastel-00913690>

Submitted on 4 Dec 2013

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

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



EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique et Réseaux »

présentée et soutenue publiquement par

Mathias BAGLIONI

le 25 Avril 2012

**Nouvelles Interactions Physiques pour
Dispositifs Mobiles**

Directeur de thèse : **Eric LECOLINET**
Co-encadrement de la thèse : **Yves GUIARD**

Jury

Mme Monique NOIRHOMME , Professeur, Institut d'Informatique, FUNDP	Rapporteur
M. Franck POIRIER , Professeur, Lab-STICC (CNRS), Université de Bretagne-Sud	Rapporteur
M. Renaud BLANCH , Maître de conférence, LIG, Université Joseph-Fourier - Grenoble I	Examineur
M. Patrick GIRARD , Professeur, ENSMA	Examineur, Président
M. Stéphane HUOT , Maître de conférence, LRI, Université Paris-Sud 11	Examineur
M. Johann DAIGREMONT , Alcatel-Lucent Bell Labs France	Invité

TELECOM ParisTech

école de l'Institut Télécom - membre de ParisTech

Remerciements

Je tiens d'abord à remercier mes directeurs de thèse Eric Lecolinet et Yves Guiard. Leurs conseils avisés et leur implication constante tout au long de ces trois années m'ont permis de mener à bien ce travail de thèse. Ils m'ont également permis de m'affirmer à l'oral et de progresser dans les différents aspects liés au travail de recherche. Au-delà de leur rôle de directeur de thèse je tiens également à les remercier pour toutes nos discussions. Leur écoute et leur humour en font des personnes que j'apprécie sincèrement.

Je tiens également à remercier les membres du Jury. Tout d'abord Monique Noirhomme et Franck Poirier pour avoir accepté de relire et de juger mon travail. Mais également les examinateurs Renaud Blanch, Patrick Girard et Stéphane Huot pour leur relecture et la pertinence de leurs questions lors de la soutenance. Je remercie également mon invité Johann Daigremont.

Je remercie tous les membres d'Alcatel Lucent Bell Labs avec qui j'ai eu l'occasion de collaborer : Johann Daigremont, Yann Gasté, Bruno Legat, Patrick Legrand, Myriam Ribière et Julien Robinson. Je remercie également ceux que j'aurais oubliés de citer. Leur enthousiasme et leurs conseils ont rendu cette collaboration enrichissante sur le plan professionnel et personnel.

Je remercie toutes les personnes de Télécom Paristech et plus particulièrement les personnes qui sont (ou étaient) dans les bureaux du bâtiment C 2ème étage.

Je tiens à remercier tout particulièrement mes collègues qui sont devenus au cours de ces années des amis. Tout d'abord les plus anciens : Gilles Bailly, Sylvain Malacria et Anne Roudaut qui m'ont donné l'envie de m'engager dans une thèse alors que j'étais en stage et qui m'ont tout au long de ces trois années apporté conseils, soutien et bonne humeur. Je remercie également Simon Perault, Dong-Bach Vo pour leur bonne humeur quotidienne, les déjeuners chez yong et les "debriefs" au spootnik. Je remercie ceux qui ont partagé nos bureaux pendant leur postdoc ou stage, Aurélien Tabard, Thomas Pietrzack, Marcos Ser-

rano, Simon Miccolier, Vincent Bitaud. Je les remercie tous pour tous les moments de travail et de détente que nous avons partagés. Je remercie également Hayette Soussou la secrétaire de notre département pour nos discussions pendant les pauses café et tous nos fous rires. Je remercie tous les joueurs de belote coincée qui se reconnaitront.

Je remercie tous mes amis et notamment Elli Nebout qui en a été la représentante lors de ma soutenance. Je les remercie tous pour tout ce que l'on a partagé depuis tant d'années. Je remercie d'ailleurs mes anciens colocataires qui organisaient des soirées pendant que j'écrivais mes premiers articles.

Je remercie toute ma famille qui m'a toujours soutenu dans mes choix et pour toutes les valeurs qu'ils m'ont apportées.

Enfin je remercie tout particulièrement ma copine Sahar Saeidnia qui partage ma vie depuis bientôt sept ans et qui m'a supporté durant ces trois années de thèse.

Résumés

Résumé

Français

Soixante pour cent de la population mondiale possède aujourd'hui un téléphone portable. Les modèles récents, qui tendent à devenir de petits ordinateurs portables, sont jusqu'à 10 000 fois plus puissants et 300 000 fois moins lourds que le premier ordinateur, apparu voici 60 ans. Malgré cette puissance, et les ressources offertes par les équipements dont ils sont munis (écran tactile multitouch, accéléromètres et autres capteurs...), les mobiles souffrent de certaines limitations du fait de la disparition de nombreux périphériques d'entrée classiques tels que le clavier et la souris. Afin d'augmenter la bande passante interactionnelle de ces dispositifs et de tirer partie de ces capteurs, nous nous sommes intéressés dans cette thèse aux possibilités offertes par l'interaction gestuelle en considérant deux axes de recherche : les mouvements sur le mobile et les mouvements du mobile.

Après avoir proposé un espace de classification nous avons conçu et développé plusieurs techniques d'interaction gestuelle destinées à enrichir et faciliter l'interaction de l'utilisateur avec ces dispositifs.

Concernant les mouvements sur le mobile, nous nous sommes intéressés à l'amélioration du Flick, une technique de défilement largement popularisée ces dernières années. Après avoir étudié son 'anatomie', nous avons proposé d'exploiter plusieurs ressources interactionnelles jusqu'à présent inutilisées. Ce travail a donné naissance à trois nouvelles techniques : Flick-and-brake et LongFlick, qui utilisent respectivement la pression sur l'écran et les caractéristiques du geste de lancer pour mieux contrôler le défilement, et 'Semantic Flicking' qui exploite la sémantique du document pour faciliter la lecture.

Dans un second temps nous avons considéré les possibilités offertes par l'interaction 3D, en déplaçant le dispositif mobile dans l'espace. Cette voie de recherche vise à permettre d'augmenter le vocabulaire d'interaction sans parasiter les interactions tactiles déjà existantes. Tirant parti des capteurs intégrés dans les mobiles, en particulier les accéléromètres, nous avons proposé deux nouvelles techniques d'interaction 3D : TimeTilt qui utilise des gestes fluides et impulsifs pour naviguer aisément entre différentes vues et JerkTilt qui introduit la notion de gestes 'auto-délimités' pour accéder rapidement à des commandes.

Ces gestes auto-délimités ayant également la particularité de pouvoir être facilement combinés avec des interactions sur l'écran, nous avons enfin considéré la combinaison des modalités offertes par les gestes bi- et tri-dimensionnels.

Anglais

Sixty percent of the world's population now owns a mobile phone. Recent models, which tend to become small laptops are up to 10 000 times more powerful and 300 000 times less heavy than the first computer, appeared 60 years ago. Despite this power and all the resources provided by modern equipment (touchscreen, accelerometer and other sensors...), mobile phones suffer from certain limitations due to the lack of conventional input such as keyboard and mouse. To increase the interaction bandwidth of these device and take advantage of these sensors, we investigate the potential offered by gestural interaction by considering two lines of research : movements on the mobile and movements of the mobile.

After defining a classification space we conceived and developed several gestural interaction techniques to enrich and facilitate interaction between the user and these devices.

For the movements on the mobile we have proposed to improve the flick, a scrolling technique widely popularized in recent years. We first studied his 'anatomy' of this technique, then we proposed to exploit multiple interactional resources hitherto unused. This work gave birth to three new techniques : Flick-and-Brake and LongFlick, that respectively use the pressure on the screen and the characteristics of the throwing action to better control scrolling, and Semantic Flicking which leverage on document semantics to facilitate reading.

In a second stage we considered the potential of 3D interaction, by moving the mobile device in the space. This line of research aims at expanding the interaction vocabulary while avoiding interferences with existing touch interactions. Leveraging on integrated sensors such as accelerometers, we proposed two new techniques for 3D interaction : TimeTilt, which uses smooth and impulsive gestures to easily navigate between different views, and JerkTilt, which introduced the notion of self-delimited gestures for accessing quickly to commands.

An interesting property of these self-defined gestures is their ability to be combined with common interactions on the screen. We thus finally considered the combination of modalities offered by the two and three-dimensional gestures.

Table des matières

1	Introduction	11
1.1	Interactions physiques sur dispositifs mobiles	12
1.1.1	Contexte général	12
1.1.2	Des dispositifs physiques	13
1.1.3	Deux types d'interactions	14
1.2	Contributions	14
1.2.1	Espace de classification	15
1.2.2	Mouvements sur le dispositif	15
1.2.3	Mouvements du dispositif	16
1.3	Organisation du mémoire	17
2	Espace de caractérisation	19
2.1	Introduction	20
2.1.1	Taxonomies existantes	21
2.1.2	L'interaction gestuelle physique sur mobile.	25
2.2	Espace de caractérisation proposé	26
2.2.1	Type de capteur	28
2.2.2	Type de mouvement	35
2.2.3	Type de mapping	41
2.2.4	Type de Contrôle	45
2.2.5	Type de retour d'information	47
2.2.6	Contexte	49
2.3	Conclusion	52
3	Mouvements sur le dispositif	57
3.1	Introduction	58
3.1.1	Le Flick	58
3.2	Flick-and-Brake	64
3.2.1	Contexte	64
3.2.2	Concept général	65

3.2.3	Évaluation	72
3.2.4	Conclusion	77
3.3	Semantic Flicking	79
3.3.1	Contexte	79
3.3.2	Description	79
3.3.3	Prototypes	81
3.3.4	Conclusion	84
3.4	Long Flick	85
3.4.1	Contexte	85
3.4.2	Définitions	85
3.4.3	Exploitation	88
3.4.4	Conclusion	91
4	Mouvements du dispositif	93
4.1	Introduction	94
4.1.1	Contexte	94
4.1.2	Types de geste	94
4.1.3	Délimiteurs	95
4.1.4	Deux nouvelles techniques gestuelles	95
4.2	TimeTilt	97
4.2.1	Présentation	97
4.2.2	Vocabulaire Gestuel	97
4.2.3	Evaluation	98
4.2.4	Application	100
4.2.5	Conclusion	106
4.3	JerkTilts	107
4.3.1	Présentation	107
4.3.2	Vocabulaire Gestuel	107
4.3.3	Évaluation	110
4.3.4	Application	121
4.3.5	Extension de JerkTilts : Combinaison 3D/2D	124
4.3.6	Conclusion	130
5	Conclusion	133
5.1	Conclusion	134
5.2	Contributions	134
5.2.1	Espace de classification	134
5.2.2	Mouvements sur le dispositif	135
5.2.3	Mouvements du dispositif	136
5.3	Perspectives	137
5.3.1	Espace de Classification	137

<i>TABLE DES MATIÈRES</i>	9
5.3.2 Mouvements sur le Dispositif	138
5.3.3 Mouvements du Dispositif	138
Bibliographie Personnelle	141
Liste des figures	144
Liste des Tableaux	149
Bibliographie	151
Appendices	165

1

Introduction

Sommaire

1.1 Interactions physiques sur dispositifs mobiles	12
1.1.1 Contexte général	12
1.1.2 Des dispositifs physiques	13
1.1.3 Deux types d'interactions	14
1.2 Contributions	14
1.2.1 Espace de classification	15
1.2.2 Mouvements sur le dispositif	15
1.2.3 Mouvements du dispositif	16
1.3 Organisation du mémoire	17

1.1 Interactions physiques sur dispositifs mobiles

1.1.1 Contexte général

Les dispositifs mobiles ont pris une place importante dans notre quotidien. Que ce soit sur le lieu de travail ou à la maison, de nombreuses personnes utilisent régulièrement des téléphones mobiles, baladeurs musicaux, GPS, etc. Si l'on prend l'exemple des téléphones mobiles, on s'aperçoit que dans certains pays (Italie, Russie, Arabie Saoudite...) les habitants possèdent en moyenne 1.5 téléphone par individu¹. Les différentes formes que peuvent prendre ces terminaux convergent depuis plusieurs années vers des dispositifs composés d'un écran tactile d'une taille d'environ 9 cm de diagonale et qui sont de plus en plus souvent dépourvus de clavier physique. Cependant, ils sont aujourd'hui pour la plupart équipés d'un nombre important de capteurs (accéléromètre, gyroscope, caméra, capteur de lumière, capteur de pression, micro, etc.). Grâce à leur petite taille ces capteurs n'imposent pas de forme particulière au dispositif tout en offrant de nouveaux moyens d'expression et d'interaction. C'est pourquoi la recherche en Interaction Homme-Machine s'intéresse tout particulièrement à ces dispositifs mobiles et à leur nouveaux capteurs ainsi qu'aux enjeux interactionnels qu'ils soulèvent.



FIGURE 1.1: Exemples de Smartphones : iPhone 4 et Google Nexus One. Comme on peut le constater ces deux modèles ont un écran tactile mais ne possèdent plus de clavier physique. Ils comportent seulement un nombre limité de boutons réservés à des tâches spécifiques.

Les dispositifs mobiles tels que les téléphones cellulaires modernes (smartphones **FIGURE 1.1**) ne sont plus seulement des téléphones. Ils sont devenus le moyen le plus courant d'effectuer une communication distante, qu'elle soit synchrone (ex. appel vocal) ou asynchrone (ex. sms)². Aujourd'hui on peut téléphoner, consulter et envoyer des e-mails, naviguer sur internet, envoyer des messages, écrire sur des forums, poster un *tweet*, mettre à jour son statut Facebook, commenter des photos, etc. avec un seul et même dispositif

1. Source : <http://www.somobile.co.uk/news/2012/01/11/infographic-the-world-of-mobile-phones/>

2. Source : <http://www.pewglobal.org/2011/12/20/global-digital-communication-texting-social-networking-popular-worldwide/1/>

[Falaki et al., 2010]. Ils deviennent aussi des objets d'utilisation quotidienne, tant pour rechercher de l'information (internet, carte, etc.) que pour écouter de la musique ou regarder une vidéo. Enfin ils permettent d'accéder à une grande variété d'applications n'ayant aucun lien avec les télécommunications (jeux, notes, dessins, etc.). Ces dispositifs encore coûteux il y a peu sont aujourd'hui accessibles à un large public, du moins dans les pays occidentaux et certains pays émergents. Force est de constater qu'ils y sont devenus omniprésents.

1.1.2 Des dispositifs physiques

Ces travaux de recherche s'articulent autour de l'idée suivante : les nouveaux capteurs apportent une nouvelle *physicalité*³ aux dispositifs mobiles. Par nouveaux capteurs nous considérons aussi bien l'écran tactile que les autres capteurs disponibles (accéléromètre, gyroscope, etc.). La notion de *physicalité* repose sur plusieurs idées :

- Le contexte d'utilisation du mobile.
- Le mobile comme objet physique.
- La facilité naturelle d'interaction du mobile.

Contexte d'utilisation

De par leur facteur de forme et l'utilisation que l'on en fait, les dispositifs mobiles ne sont pas juste des ordinateurs miniatures [Beaudouin-Lafon, 2000]. Les usages en situation de mobilité (dans la rue, dans les transports) définissent une interaction différente, souvent plus courte en temps et à l'aide de moyens redéfinis (écran plus petit mais tactile, pas de souris ou clavier physique mais plus de capteurs, etc.). L'attention est souvent partagée avec une autre activité (discussion, marche, etc.), ce qui implique des enjeux différents pour la présentation d'informations ou tout simplement pour l'interaction elle-même. Enfin, cette situation de mobilité amène généralement l'utilisateur à manipuler son dispositif avec une seule main (la seconde tenant un sac, une barre de métro ou encore étant au repos [Roudaut and Lecolinet, 2007]). Le téléphone mobile devient un objet physique de notre quotidien et les interactions qui en découlent sont fortement corrélées à ce contexte.

Un objet physique

L'arrivée de nouveaux capteurs comme les accéléromètres a aussi contribué à la modification de la perception des dispositifs mobiles, non plus comme de petits ordinateurs mais

3. Néologisme issu de l'adjectif *physique* : *Adj. Relatif au monde matériel, aux phénomènes naturels, aux lois qui les régissent et à la science qui vise à en rendre compte.* Source : Dictionnaire de l'Académie française <http://atilf.atilf.fr>

L'expression "donner de la physicalité" [Moulon, 2010] est aussi utilisé dans l'Art pour décrire le processus qui vise à rendre physique, réel quelque chose de virtuel.

comme de nouveaux objets interactifs à part entière. De plus, notamment grâce à sa petite taille, le dispositif devient lui-même l'objet d'interaction : on secoue le dispositif, il passe à une autre chanson ; on le réoriente, l'écran pivote. Le dispositif d'entrée n'est plus un périphérique, c'est l'objet lui-même qui devient le dispositif d'entrée.

Des interactions naturelles

L'affordance [Gibson, 1977] des objets dans le monde virtuel est généralement traduite par leur aspect visuel à l'écran. Il est également parfois difficile de distinguer l'affordance de la convention. Ainsi lorsqu'un utilisateur voit une croix dans un bouton, il pense tout de suite qu'il faut cliquer pour fermer, supprimer une fenêtre ou un objet : ce n'est pas de l'affordance [Norman, 1999] mais une convention. Avec les dispositifs mobiles équipés d'un écran tactile et de capteurs divers, l'utilisateur agit directement sur l'objet d'interaction. Il en découle une affordance plus forte, par exemple lorsqu'un utilisateur fait glisser son doigt sur l'écran pour faire défiler une liste. Les interactions que l'on peut proposer sur ces dispositifs se rapprochent d'interactions qui nous semblent naturelles, par leur lien étroit avec nos interactions quotidiennes (non informatiques) et par le biais de métaphores gestuelles.

1.1.3 Deux types d'interactions

Nous avons choisi de représenter les interactions gestuelles mobiles selon deux types de mouvements complémentaires :

- Les mouvements *sur* le dispositif. (Interaction sur le dispositif).
- Les mouvements *du* dispositif. (Interaction avec le dispositif).

Ces deux types de mouvements sont indépendants car ils font appel à un canal d'entrée différent. Ils sont chacun représenté dans un repère géométrique différent. Les mouvements *sur* le dispositif sont décrits dans le référentiel du dispositif (FIGURE 1.2 en rouge), alors que les mouvements *du* dispositif s'expriment dans le référentiel terrestre (FIGURE 1.2 en bleu). Ils ont de plus la particularité d'être complémentaires et peuvent même être combinés. On peut donc envisager des interactions qui utilisent simultanément ces deux canaux d'entrée.

1.2 Contributions

Nous présentons ci-après les différentes contributions de cette thèse, qui a été réalisée dans le cadre du laboratoire commun UBIMEDIA entre l'institut Télécom et Alcatel-Lucent Bell Labs. Nous avons notamment travaillé avec une équipe du département Social Communications qui développe des applications pour les réseaux sociaux sur dispositifs mobiles.

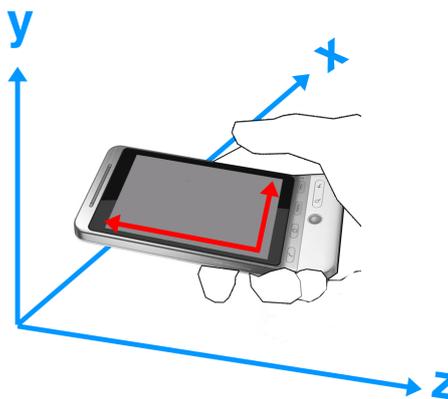


FIGURE 1.2: En bleu le référentiel terrestre et en rouge le référentiel du mobile

Dans cette collaboration, notre but a été de proposer de nouvelles interactions qui permettent d'améliorer l'expérience utilisateur pour les usages et applications proposés par leurs équipes.

1.2.1 Espace de classification

Nous proposons un espace de conception et de classification des interactions gestuelles sur dispositifs mobiles. Comme évoqué précédemment, cet espace se construit sur deux axes : l'interaction sur le dispositif et l'interaction avec le dispositif. Il vise à faciliter la classification et la comparaison des différentes techniques de la littérature. Il apporte aussi un aspect génératif destiné à aider à la conception de nouvelles interactions gestuelles sur dispositif mobile.

Voici les techniques d'interaction que nous avons réalisées en nous appuyant sur cet espace.

1.2.2 Mouvements sur le dispositif



Flick and Brake Cette technique basée sur la technique de *Flick* permet à l'utilisateur de faire défiler une liste et d'en contrôler la vitesse de défilement. L'utilisateur peut en effet, en appliquant une pression sur l'écran, augmenter la friction virtuelle afin de ralentir le défilement automatique. Cette technique présente deux variantes basées sur les métaphores de la roue inertielle entretenue et non entretenue. Elle améliore en particulier le défilement dans les listes à une dimension, comme le montre l'évaluation que nous avons réalisée.



Semantic Flicking Le Semantic Flicking est aussi une amélioration du *Flick* classique. Cette technique se sert du contenu du document parcouru pour moduler le défilement en fonction de l'importance que revêtent pour l'utilisateur les différentes parties du document. Cette technique, qui a fait l'objet d'un dépôt de brevet, a été réalisée en collaboration avec Bell Labs, notamment pour son intégration dans l'application S-Book.



LongFlick La technique LongFlick est basée sur l'étude de l'amorce du mouvement de Flick. Nous présentons les propriétés de l'amorce et proposons de nouvelles idées afin d'exploiter efficacement cette étape du mouvement. Nous présentons plusieurs mappings entre la longueur du geste de l'amorce et la distance parcourue lors du défilement automatique. Cette technique permet d'atteindre facilement une partie de la liste en ne réalisant qu'un seul geste. Elle permet également de parcourir à vitesse constante une distance définie par l'utilisateur.

1.2.3 Mouvements du dispositif



TimeTilt TimeTilt est une technique utilisant les accéléromètres pour interagir gestuellement. Elle propose un ensemble de trois gestes brusques et d'un geste fluide. Elle répond aux problèmes des délimiteurs liés à l'interaction gestuelle 3D en proposant d'utiliser un geste brusque comme activateur de gestes fluides. TimeTilt permet également d'interagir avec des doigts qui sont habituellement inutilisés, en proposant un geste de tap de l'index à l'arrière du dispositif. Nous présentons également une application du système permettant de changer aisément et rapidement de fenêtre sur un dispositif mobile. Cette application propose deux modes d'interaction complémentaires, le mode lenticulaire basé sur la métaphore des cartes lenticulaires et le mode superposition. Une expérience utilisateur a permis de récolter un échantillon significatif de ces nouveaux gestes afin de construire un algorithme de reconnaissance, puis d'évaluer ses performances.



JerkTilt Cette seconde technique utilise également les accéléromètres et propose un vocabulaire de huit gestes. Ces gestes ont la particularité d'être *auto-délimités*, ce qui permet de les réaliser sans avoir recours à une activation préliminaire. Ces Jerktilts sont des rotations aller-retour de faible amplitude permettant une bonne adéquation au contexte de mobilité et l'accès rapide à des commandes. Nous avons réalisé plusieurs expériences visant à évaluer les performances de JerkTilt. Nous présentons également plusieurs applications utilisant ces gestes. Enfin, nous montrons que ces gestes peuvent être combinés avec les interactions *sur* le dispositif.

1.3 Organisation du mémoire

Ce manuscrit est organisé en 5 chapitres illustrant nos différentes contributions :

- Le chapitre 1 introduit ce mémoire et présente le sujet de ces travaux de recherche.
- Le chapitre 2, intitulé “Espace de caractérisation”, expose les travaux de la littérature et présente notre espace de conception et de caractérisation des interactions gestuelles physiques sur dispositifs mobiles.
- Le chapitre 3, intitulé “Mouvement sur le dispositif”, est consacré à l’étude des interactions gestuelles sur l’écran tactile du dispositif. Il présente “Flick and Brake” et son évaluation face à une technique reconnue de la littérature. Il traite également du “Semantic Flicking” et de son contexte d’utilisation dans la collaboration UBIMEDIA. Il présente enfin l’étude de “LongFlick”.
- Le chapitre 4, intitulé “Mouvement du dispositif”, est consacré à l’étude des interactions gestuelles avec le dispositif mobile au moyen de capteurs de mouvements. Il présente la technique “TimeTilt” ainsi qu’une étude permettant la collecte de gestes servant à l’algorithme de reconnaissance de la technique. Il traite aussi de “JerkTilts” et expose les différentes évaluations qui ont permis de valider cette technique : d’abord en récoltant des échantillons de gestes, puis en la comparant avec une technique existante et enfin en testant son applicabilité dans un contexte réel d’utilisation.
- Le chapitre 5 conclut ce mémoire et présente les perspectives relatives à ces travaux.

2

Espace de caractérisation

Sommaire

2.1	Introduction	20
2.1.1	Taxonomies existantes	21
2.1.2	L'interaction gestuelle physique sur mobile.	25
2.2	Espace de caractérisation proposé	26
2.2.1	Type de capteur	28
2.2.2	Type de mouvement	35
2.2.3	Type de mapping	41
2.2.4	Type de Contrôle	45
2.2.5	Type de retour d'information	47
2.2.6	Contexte	49
2.3	Conclusion	52

2.1 Introduction

De nombreuses études ont été consacrées ces dernières années aux contraintes qui affectent les dispositifs mobiles et aux solutions pour les résoudre. Diverses techniques d'interaction (Rock'n'Scroll [Bartlett, 2000], TapTap [Roudaut et al., 2008], Back-of-device interaction [Baudisch and Chu, 2009], CycloStar [Malacria et al., 2010a], etc.) ou de visualisation (Halo [Baudisch and Rosenholtz, 2003], AppLens [Karlson et al., 2005]) ont ainsi été proposées pour palier les limitations de ces dispositifs (petit écran, entrées limitées). Une réponse à l'absence de certains périphériques d'entrée (clavier, boutons de la souris, molette...) consiste à recourir aux interactions sur l'écran ou à l'utilisation des capteurs intégrés au dispositif. Un nombre important de travaux a donc été réalisé utilisant ces technologies en tirant partie de l'interaction dite gestuelle. Selon Cadoz [Cadoz, 1994], l'interaction gestuelle se présente selon trois variantes : les gestes érgotiques, épistémiques et sémiotiques¹. Cependant cette catégorisation de l'interaction gestuelle ne décrit que le geste et ne définit pas le *moyen* (ou le média) avec lequel le geste a été réalisé. C'est ce que nous avons essayé de prendre en compte dans l'espace de classification que nous présentons dans ce chapitre. Cet espace a fait l'objet d'une publication longue à la conférence IHM 2009 traitant de l'interaction gestuelle sur dispositif mobile avec capteur intégré (accéléromètre, etc.). Ce chapitre intègre également les interactions gestuelles sur l'écran tactile.

Interactions sur le mobile. Les écrans présents sur les dispositifs actuels ont favorisé le développement de nouvelles techniques d'interaction. Ils sont plus grands que ceux des dispositifs mobiles antérieurs, tactiles et généralement basés sur une technologie capacitive multi-touch (de deux à dix points de contact simultanés). Ils permettent de détecter non seulement les coordonnées du point de contact sur l'écran mais aussi, dans le cas des écrans capacitifs, l'étendue de la surface de contact. Ils offrent alors la possibilité de créer des interactions riches (interaction gestuelle, multi-touch, etc.) sur les terminaux mobiles.

Interaction avec le mobile. L'intégration de nouveaux capteurs (accéléromètre, gyroscope, caméra, boussole...) permettant de détecter les mouvements et les orientations du mobile apporte de nouvelles possibilités d'interaction fondées sur les gestes dans l'espace que l'utilisateur réalise avec son dispositif. De par leur facteur de forme (petite taille, faible poids), leur connectivité sans fil, leurs spécificités d'usage (l'utilisateur a en permanence le dispositif sur lui et interagit souvent à une seule main), les terminaux mobiles sont particulièrement adaptés à ce nouveau type d'interaction. Cependant ce type d'interaction nécessite généralement le recours à un délimiteur modal indiquant explicitement au système le début

1. Les gestes épistémiques servent à explorer le monde physique, par exemple lorsque l'on touche un mur pour en apprécier le relief. Les gestes érgotiques nous permettent d'agir sur les objets présents dans ce monde, comme lorsque l'on attrape une pomme pour la manger. Enfin les gestes sémiotiques sont ceux qui nous permettent de communiquer avec nos semblables (gestes accompagnant la parole, langage des signes, etc.).

et la fin de chaque phase d'interaction gestuelle.

2.1.1 Taxonomies existantes

L'interaction définit l'action réciproque entre deux objets². Dans le cadre de l'Interaction Homme-Machine, comme son nom l'indique, les deux objets sont l'*homme* (utilisateur) et la *machine* qui interagissent grâce à un *langage* d'interaction. Par exemple, lorsque le langage d'interaction entre ces deux objets est le langage gestuel, nous parlerons d'interaction gestuelle; si le dispositif d'interaction est mobile nous pourrions parler d'interaction mobile. Ainsi les taxonomies que nous présentons couvrent des sous-parties de l'interaction en général en spécifiant le langage et/ou le dispositif utilisés.

Parler d'interaction *sur* le mobile et *avec* le mobile fait appel aux interactions mobiles, gestuelles et tactiles. Cette partie se propose de présenter certaines taxonomies traitant de ces interactions.

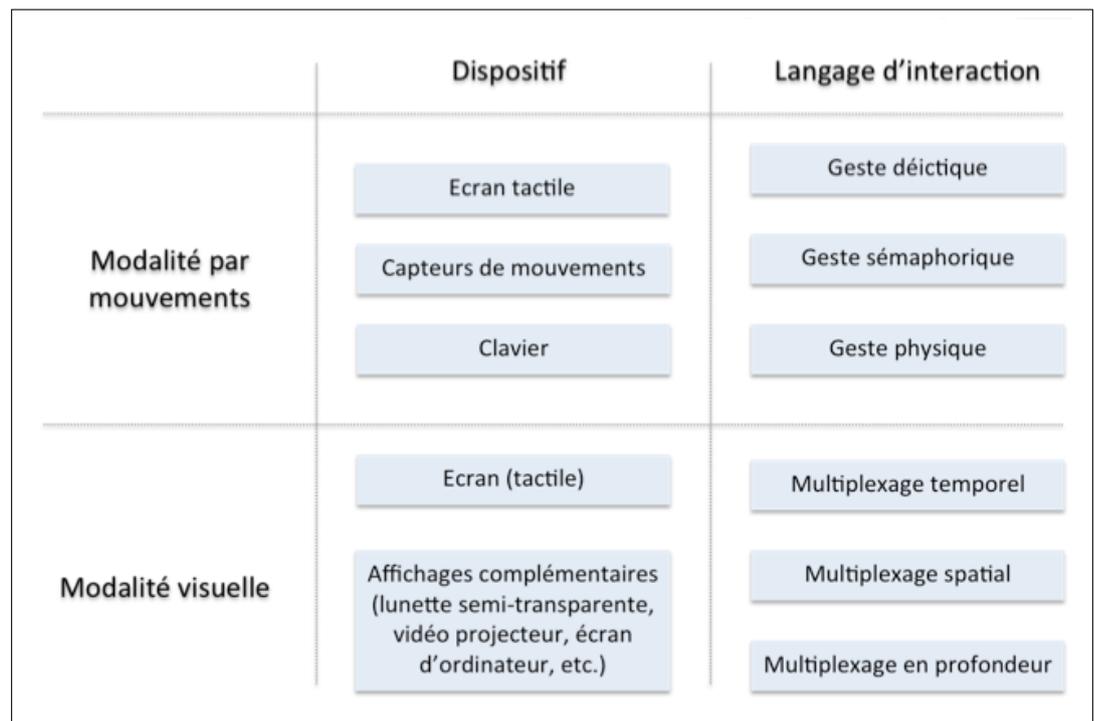
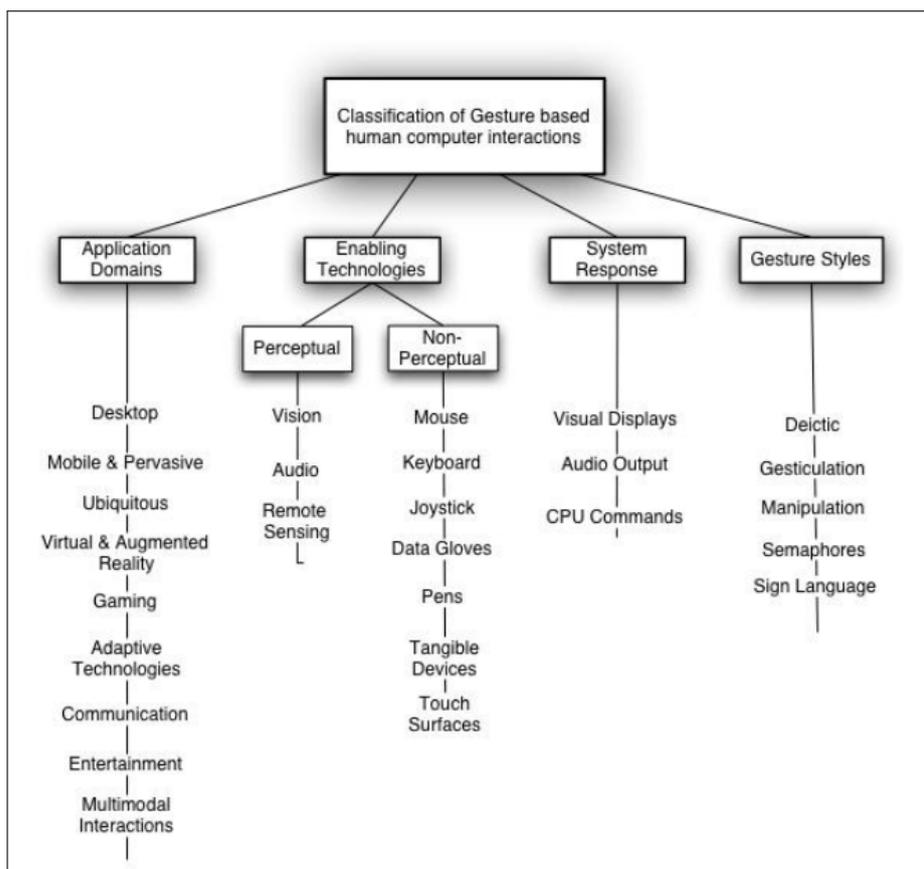


FIGURE 2.1: Taxonomie des techniques d'interaction et de visualisation [Roudaut and Lecolinet, 2007].

2. Définition d'Interaction : "PHYS. Action réciproque de deux ou plusieurs corps. [...] **Par ext. Influence qu'exercent les uns sur les autres des phénomènes, des faits, des objets, des personnes.**" Dictionnaire de l'Académie Française : <http://atilf.atilf.fr/dendien/scripts/generic/cherche.exe?18585;s=3676588980;;>

Roudaut [Roudaut and Lecolinet, 2007] présente une taxonomie de l'interaction mobile intéressante qui essaie de couvrir l'intégralité des techniques d'interaction et de visualisation sur dispositif mobile. Elle utilise la notion de modalité [Nigay and Coutaz, 1996] qui permet de définir l'interaction mobile par un couple <Dispositif, Langage>. Ainsi il est possible de représenter les différentes interactions comme sur la [FIGURE 2.1](#). Un dispositif physique est un élément du système qui acquiert (dispositif d'entrée) ou fournit (dispositif de sortie) des informations à l'utilisateur. Un langage d'interaction est un ensemble d'expressions bien formées et significatives. Par exemple, lorsque l'utilisateur pointe une cible sur un écran tactile, il utilise la modalité <écran tactile, geste déictique>. Cette taxonomie recouvre de manière assez générale les différents aspects de l'interaction mobile. Certaines études s'intéressent à des interactions mobiles plus spécifiques comme [Poirier and Schadle, 2004] qui est consacrée à l'entrée de texte sur dispositifs mobiles. D'autres, comme [Keim, 2002, Shneiderman, 1996], traitent de la question de la visualisation d'information sur les surfaces d'affichage et plus particulièrement sur les écrans de petite taille dans [Baffoun and Robert, 2006].



[FIGURE 2.2](#): Taxonomie des interactions gestuelles [Karam and Schraefel, 2005].

Comme nous avons pu le voir en introduction, Cadoz [Cadoz, 1994] définit l'interaction gestuelle en général. Il présente ainsi trois fonctions des gestes (ergotique, sémiotique, épistémique) qui sont souvent entre-mêlées dans les cas de figure qui nous intéressent. Il décrit ainsi ce qu'il appelle le geste instrumental qui permet d'interagir avec un instrument (machine). Ce geste fait appel à plusieurs fonctions, d'abord la fonction ergotique du geste (parce l'utilisateur agit) puis la fonction sémiotique (lorsque une communication s'installe entre l'utilisateur et la machine). De ce fait un geste servant à interagir avec une machine possède généralement ces trois propriétés. Karam [Karam and Schraefel, 2005] intervient de manière intéressante sur ce point. Elle utilise en effet une définition des gestes (*style* de geste) similaire à celle de Cadoz mais apporte une vision plus large, s'inspirant également des travaux de Quek *et al.* [Quek et al., 2002] sur le geste. Elle l'enrichit avec trois autres catégories permettant de préciser cette interaction en prenant également en compte le contexte et les technologies mises en oeuvre dans cette interaction (FIGURE 2.2).

Scoditti [Scoditti et al., 2011] présente une taxonomie des interactions gestuelles en précisant un peu plus le dispositif. En effet, il se concentre sur l'utilisation des accéléromètres comme périphériques d'entrées. Il reprend notamment les taxonomies présentées dans cette section, ainsi que la nôtre³, en essayant d'en proposer une synthèse. En s'appuyant sur les travaux de Foley *et al.* [Foley et al., 1984] et de Buxton [Buxton, 1983] il définit une nouvelle taxonomie (FIGURE 2.3) basée sur les tâches élémentaires de l'interaction (Select, Position, Orient, Text, Path, Quantify), comme on peut le voir également dans [Ballagas, 2006].

Plusieurs travaux ont déjà été réalisés pour caractériser les interactions gestuelles sur surface tactile. On peut notamment retrouver ce type de classification dans [Wobbrock et al., 2009, Malacria, 2011]. Wobbrock propose une classification des gestes sur table interactive (FIGURE 2.4) centrée utilisateur. Cette taxonomie a été élaborée sur un ensemble de 1080 gestes réalisés par les utilisateurs d'une de ses expérimentations. Les gestes réalisés devaient correspondre à la *cause* du stimulus qui était présenté (*effet*). Les gestes sont décrits selon quatre critères : la *forme*, la *nature*, l'*attache* (*binding*) et le *flot*. Malacria présente d'autres critères permettant de classifier les types de geste réalisés sur les surfaces tactiles (FIGURE 2.5). Il définit trois critères : la *fonction*, la *nature*, et la *morphologie*. La notion de *fonction* qu'il utilise est inspirée des travaux de Cadoz. La *nature* du geste qu'il propose est basée sur différentes taxonomies (notamment celle de Wobbrock et Roudaut) il garde ainsi la définition de Roudaut pour les gestes *physiques*, inspirée de la notion de gestes incarnés (*embodied*) [Fishkin et al., 1999, Dourish, 2001]. Enfin, le critère de *morphologie* s'appuie sur les travaux de Nancel *et al.* [Nancel et al., 2009] et de Card *et al.* [Card et al., 1991] et développe une analyse centrée sur les propriétés des techniques d'interaction.

3. cf. la taxonomie que nous avons présentée dans l'article publié à la conférence IHM 2009.

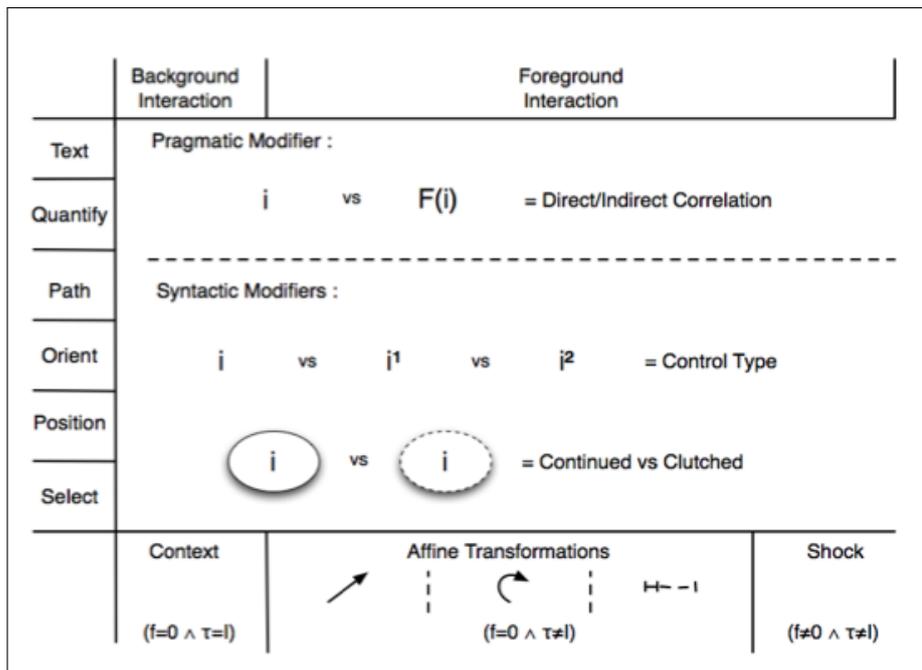


FIGURE 2.3: Taxonomie proposée par Scoditti.

TAXONOMY OF SURFACE GESTURES		
Form	<i>static pose</i>	Hand pose is held in one location.
	<i>dynamic pose</i>	Hand pose changes in one location.
	<i>static pose and path</i>	Hand pose is held as hand moves.
	<i>dynamic pose and path</i>	Hand pose changes as hand moves.
	<i>one-point touch</i>	Static pose with one finger.
	<i>one-point path</i>	Static pose & path with one finger.
Nature	<i>symbolic</i>	Gesture visually depicts a symbol.
	<i>physical</i>	Gesture acts physically on objects.
	<i>metaphorical</i>	Gesture indicates a metaphor.
	<i>abstract</i>	Gesture-referent mapping is arbitrary.
Binding	<i>object-centric</i>	Location defined w.r.t. object features.
	<i>world-dependent</i>	Location defined w.r.t. world features.
	<i>world-independent</i>	Location can ignore world features.
	<i>mixed dependencies</i>	World-independent plus another.
Flow	<i>discrete</i>	Response occurs <i>after</i> the user acts.
	<i>continuous</i>	Response occurs <i>while</i> the user acts.

FIGURE 2.4: Taxonomie proposée par Wobbrock. (“w.r.t” = “with respect to”).

Espace de caractérisation des gestes sur surface 2D			
Fonction		<i>Exploratoire</i>	Le geste sert à s'informer sur la surface
		<i>Manipulative</i>	Le geste sert à de la manipulation direct
		<i>Communicative</i>	Le geste communique une information
Nature		<i>Symbolique</i>	Le geste dessine un symbole
		<i>Physique</i>	Le geste tire parti des aptitudes "physiques"
		<i>Abstraite</i>	Le geste est ni symbolique, ni physique
Morphologie	Topologie	<i>Manuelle</i>	Monopoint, multipoint, accord, forme ou variable
		<i>Matérielle</i>	Appareil identifié par la surface interactive
	Déroulement	<i>Immobile</i>	Tracé statique
		<i>Mono-chemin</i>	Plusieurs tracés simultanés parallèles
		<i>Multi-chemin</i>	Plusieurs tracés simultanés différents
		<i>Oscillatoire</i>	L'un ou plusieurs des tracés est "répété"
	Forme		La forme est droite (polyligne), courbée ou une combinaison des deux
			La forme est ouverte ou fermée
			La forme a un ou plusieurs points d'intersection
	Transformations	<i>Orientation</i>	Rotation appliquée au geste
		<i>Échelle</i>	Homothétie appliquée au geste
		<i>Position</i>	Translation appliquée au geste
	Cinématique	<i>Vitesse</i>	Gestes effectués à différentes vitesses
		<i>Accélération</i>	Gestes fluides vs. gestes impulsifs

FIGURE 2.5: Taxonomie proposée par Malacria.

2.1.2 L'interaction gestuelle physique sur mobile.

Afin de regrouper les interactions *sur* et *avec* le dispositif, nous avons choisi de définir le langage d'interaction comme étant le langage gestuel et la machine comme étant un dispositif mobile capturant les mouvements du ou sur le dispositif. En effet, les taxonomies existantes soit recouvrent un domaine plus large [Roudaut and Lecolinet, 2007, Karam and Schraefel, 2005, Cadoz, 1994], soit n'abordent pas l'aspect mobile du dispositif [Wobbrock et al., 2009, Malacria, 2011] ou encore se définissent par une approche différente et ne prennent pas en compte l'interaction tactile [Scoditti et al., 2011]. A notre connaissance, il n'existe pas de taxonomie sur l'interaction gestuelle physique sur dispositifs mobiles, c'est-à-dire regroupant les interactions gestuelle 3D et 2D avec pour périphérique d'entrée le dispositif mobile lui-même. L'espace que nous proposons est donc dédié aux interactions utilisant des capteurs favorisant une interaction gestuelle avec le mobile comme périphérique d'entrée. Nous traitons également les interactions basées sur l'utilisation d'écrans tactiles que l'on peut considérer comme des capteurs particuliers [Bailly et al., 2008, Roudaut et al., 2009b]. Nous n'étudions pas les techniques utilisant comme moyen d'entrée des claviers [Wobbrock et al., 2006], ni celles utilisant un micro comme dans [Murray-Smith et al., 2008].

2.2 Espace de caractérisation proposé

Nous allons dans cette section expliquer les différentes dimensions que nous avons choisies pour réaliser notre taxonomie. Nous nous sommes appuyés sur les travaux cités dans la section précédente afin de définir les dimensions permettant de classifier l'interaction gestuelle physique mobile. L'interaction gestuelle comme le présente Karam se définit d'une part par l'analyse du geste et, d'autre part, par les technologies qui entrent en jeu dans la capture de ces gestes. Il nous semblait important de garder ces deux axes. Cependant nous pensons que ces deux axes de l'interaction nécessitent une définition plus fine.

Le geste n'est souvent décrit que par sa nature [Wobbrock et al., 2009] ou son style [Karam and Schraefel, 2005] qui exprime un lien entre la forme et la fonction du geste. De ce fait, le type des mouvements qui entrent en jeu dans la réalisation d'un geste n'est pas abordé. Nous avons donc choisi de décrire les gestes selon deux dimensions :

- Le type de mapping. Cette dimension est inspirée de la notion de *nature*, *style* et *fonction* du geste. Elle s'intéresse à la relation entre la forme du geste et la fonction réalisée.
- Le type de mouvement. Ici le geste est défini selon ses caractéristiques cinématiques, notamment en caractérisant les mouvements qui le composent.

Cette séparation permet d'avoir une caractérisation des gestes à différents niveaux d'abstraction. Cette définition a également été reprise par la suite dans les taxonomies de Malacria [Malacria, 2011] et Scoditti [Scoditti et al., 2011].

Du point de vue des technologies il était important de tenir compte du dispositif d'entrée ainsi que de sortie. Nous avons donc deux nouvelles dimensions :

- Le type de capteur. Cette dimension définit le capteur qui va permettre de capturer les mouvements et donc les gestes réalisés par l'utilisateur.
- Le type de retour d'information définit la modalité (langage) qui permet à l'utilisateur d'avoir un retour (réponse) sur le geste qu'il vient de réaliser.

Il nous faut donc considérer une cinquième dimension : le type de contrôle (continu, discret). Enfin, il nous paraissait important de prendre en compte le contexte d'utilisation. En effet le contexte de mobilité induit des contraintes particulières qu'il convient de respecter.

Nous nous trouvons donc avec un espace de classification à six dimensions, les cinq premières étant indépendantes, et la sixième dimension étant complémentaire et se définissant en parallèle des précédentes. La [FIGURE 2.6](#) représente cet espace.

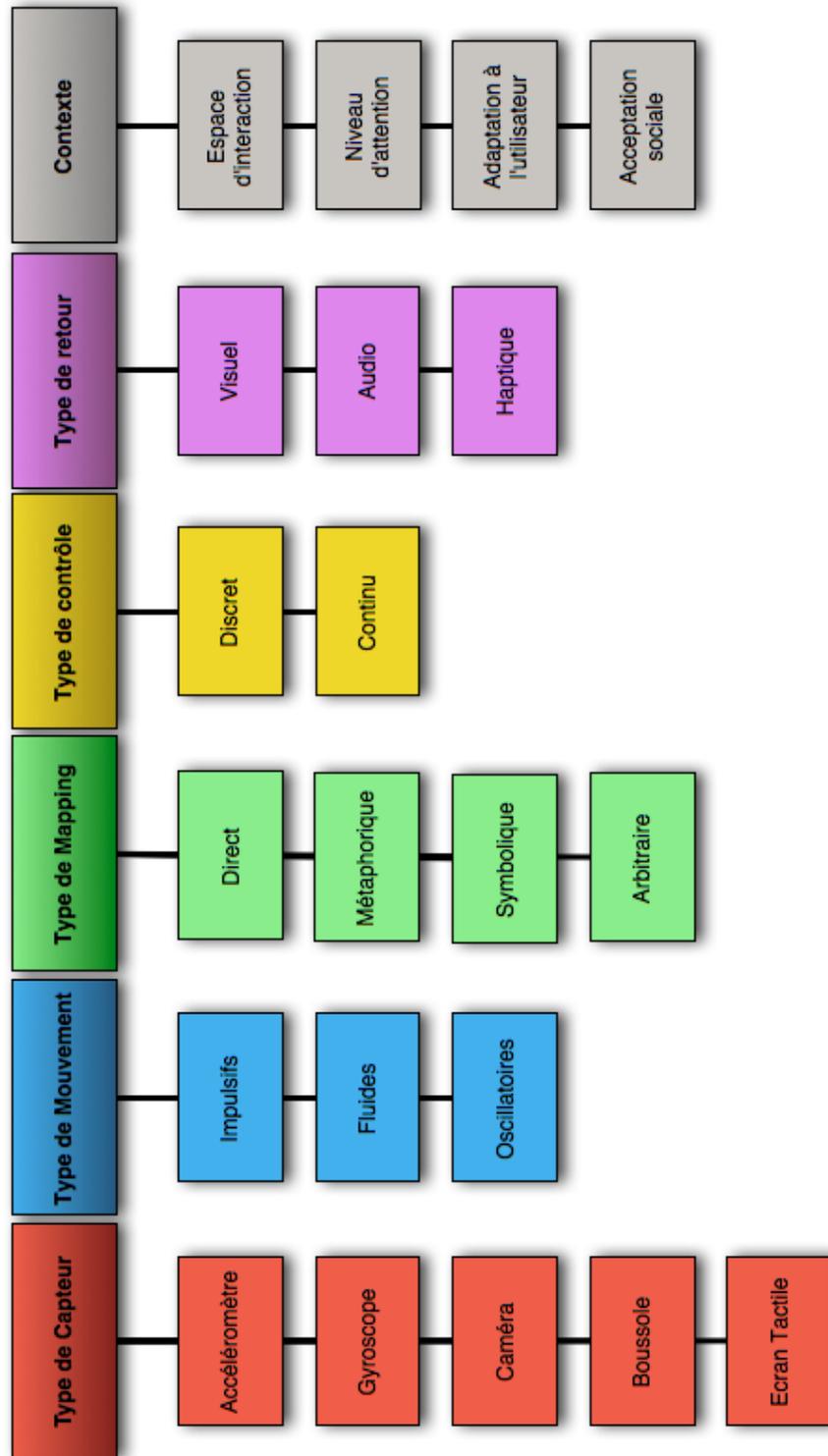


FIGURE 2.6: Espace de caractérisation

2.2.1 Type de capteur

Dans cette section nous présentons les technologies actuelles utilisées dans l'interaction gestuelle et les capteurs associés. Nous exposons d'abord les outils de mesure de mouvement, de rotation et d'accélération du mobile dans l'espace : l'accéléromètre, le gyroscope, la boussole et la caméra qui n'est pas exactement un capteur de mouvement mais qui permet cependant de mesurer les mouvements du dispositif au moyen de l'analyse d'images. Notons que cet ensemble reste ouvert aux nouveaux capteurs qui pourraient apparaître dans les années à venir. Nous présentons ensuite l'écran tactile qui peut-être considéré comme un capteur permettant le recueil des mouvements sur le mobile. Nous ne présentons pas ici les capteurs de proximité ou de luminosité.

Accéléromètres

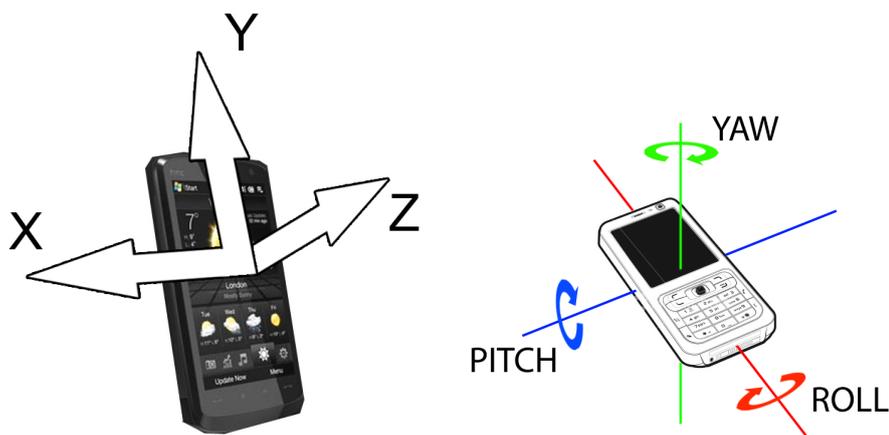


FIGURE 2.7: Référentiel des accéléromètres

Avec l'arrivée de nouveaux dispositifs intégrant des accéléromètres (l'iPhone⁴, certains HTC's ou Sony Ericsson, le NeoFreeRunner⁵), ces capteurs ont pris une grande place dans l'interaction mobile. En effet auparavant ils n'étaient souvent utilisés que pour l'analyse de mouvements en physique ou encore dans les centrales inertiels. Plusieurs types d'accéléromètres existent, dont les plus courants sont les capteurs MEMs (MicroElectroMécanique). Ils permettent, selon les modèles, de mesurer l'accélération en g (accélération gravitationnelle), sur une, deux [Linjama and Kaaresoja, 2004] ou trois¹ dimensions. Ils renvoient au système les valeurs d'accélération dans le référentiel du mobile (FIGURE 2.7). Si leur mesure s'exprime sur deux ou trois axes, ils peuvent servir d'inclinomètre, afin de déterminer l'orientation d'un écran (FIGURE 2.7) (comme par exemple sur l'iPhone¹ FIGURE 2.8), en mesurant l'attraction gravitationnelle terrestre. Ce type de capteur est de loin le plus uti-

4. <http://www.apple.com/iphone>

5. http://wiki.openmoko.org/wiki/Neo_FreeRunner/fr

lisé dans la littérature pour les techniques d'interactions gestuelles. Comme on a pu le voir précédemment Scoditti a même réalisé sa taxonomie exclusivement sur ce type de capteur. Notons aussi leur grande utilisation dans les produits commerciaux puisque de nombreux jeux (par exemple Labyrinth sur iPhone) utilisent ce type de capteur pour interagir. Ils sont également utilisés pour détecter des chocs entre deux dispositifs, par exemple avec Bump ou Paypal pour réaliser un transfert de fichier ou d'argent. On trouve également cette idée dans la littérature, par exemple Hinckley [Hinckley, 2003] propose 'd'entre-choquer' deux tablettes entre elles afin de partager leur affichage.



FIGURE 2.8: Changement d'orientation sur l'iPhone grâce à l'accéléromètre.

Gyroscopes

Les gyroscopes (FIGURE 2.9) permettent de mesurer la position angulaire. Deux exemples connus dans les appareils grand public sont sans doute la PowerBall⁶ (FIGURE 2.10 (a)), qui calcule le nombre de tours par minutes que l'on peut faire en la secouant, et, à une échelle très différente, le Segway TP (véhicule de transport personnel)⁷ (FIGURE 2.10 (b)) dont l'utilisateur incline le guidon vers l'avant pour avancer. Il sont également présents dans les smartphones de dernière génération (iPhone 4 etc.) et dans les contrôleurs de jeu comme la Wiimote, la manette playstation 6-axis, ou même dans les télécommandes nouvelle génération comme celle proposée par Free. On trouve aussi des gyroscopes dans les stabilisateurs de caméra afin d'avoir une image plus nette. Les gyroscopes embarqués dans les dispositifs actuels sont de type MEM. Ce type de capteur tend à être présent de plus en plus dans les dispositifs mobiles modernes (tablette tactile, smartphone). On trouve des techniques utilisant ces gyroscopes pour capter les mouvements de rotation d'un dispositif [Ferscha et al., 2008], l'inclinaison de l'écran par rapport à l'utilisateur [Ballagas, 2006] ou

6. <http://www.france-powerball.com/>

7. <http://www.segway.fr/>

pour l'aide à la navigation pédestre en indiquant à l'utilisateur la direction vers laquelle il se dirige [Ladetto and Merminod, 2002]. Bailly *et al.* [Bailly et al., 2011] proposent également d'utiliser les gyroscopes afin de naviguer dans les applications d'une télévision interactive. L'utilisateur peut en effet sélectionner les éléments d'un menu grâce aux rotations de la main tenant la télécommande (FIGURE 2.11).

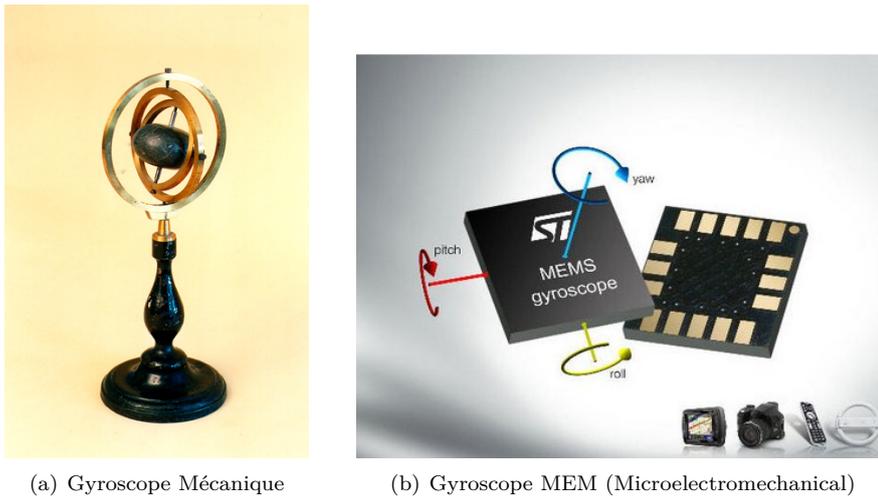


FIGURE 2.9: Deux types de gyroscope, un mécanique et un microélectromécanique. On retrouve ce dernier dans la plupart des dispositifs mobiles actuels.



FIGURE 2.10: Deux types d'utilisation commerciale des gyroscopes

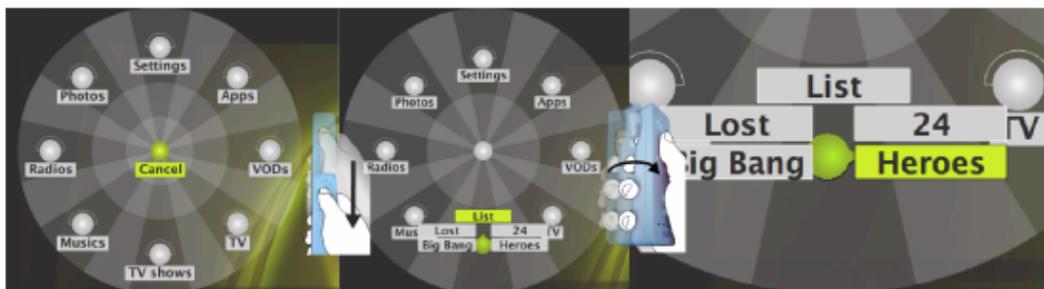


FIGURE 2.11: Sélection d'items dans un menu sur une télévision interactive.

Boussoles

La boussole est bien connue pour son utilisation dans la navigation. Elle permet d'indiquer le nord mais peut aussi être utilisée pour détecter d'autres champs magnétiques. Elle envoie des informations de direction dans un référentiel terrestre. Il existe des boussoles digitales (MEM)⁸ (FIGURE 2.14) dont la petite taille facilite l'intégration dans les dispositifs mobiles (par exemple dans la plupart des téléphones HTC récents⁹ (FIGURE 2.13)). Quelques techniques utilisant ce type de capteur ont été proposées pour l'aide à l'orientation [Ladetto and Merminod, 2002], l'utilisateur pouvant alors visualiser sur son dispositif la direction vers laquelle il se déplace. Abracadabra [Harrison and Hudson, 2009] propose d'utiliser ce capteur sur une montre bracelet. L'utilisateur peut alors contrôler son dispositif, accéder aux fonctionnalités d'un menu ou d'un lecteur de musique, en manipulant un aimant autour de la montre (FIGURE 2.12). L'exploitation de la boussole reste encore limitée dans le cas de l'interaction gestuelle sur mobile. Néanmoins elle est de plus en plus utilisée dans les applications de type google map sur smartphone (FIGURE 2.13), mais également dans la plupart des applications de réalité augmentée (par exemple Layers).



FIGURE 2.12: Abracadabra

8. <http://www.ssec.honeywell.com/magnetic/>

9. <http://www.htc.com/>



FIGURE 2.13: Google Nexus S, appareil construit par Samsung et muni d'une boussole. Ici on peut voir Une vue de Google Map avec l'indication de l'orientation.

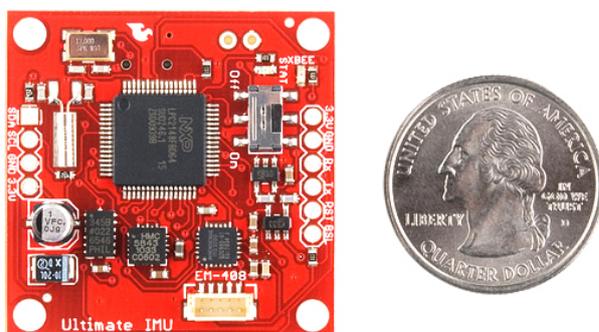


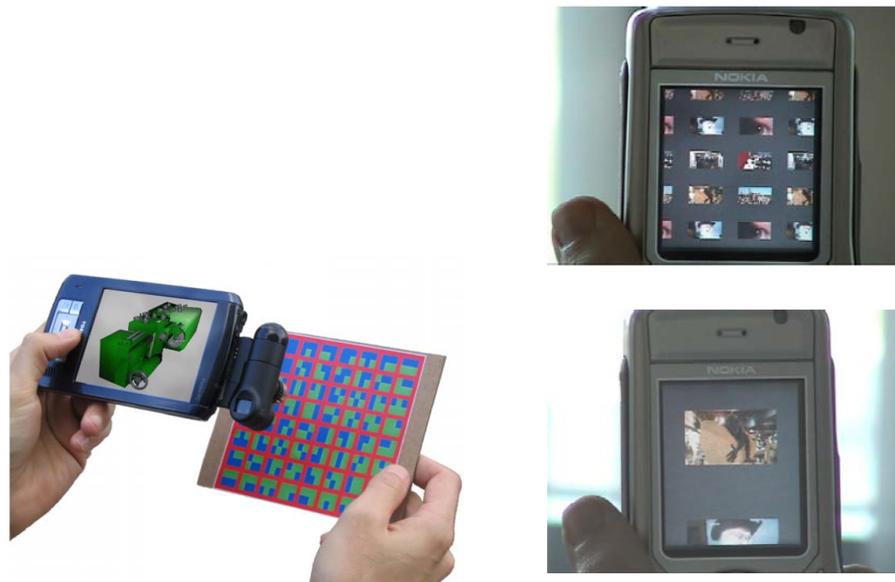
FIGURE 2.14: Exemple de puce servant de magnétomètre.

Caméras

L'utilisation des caméras vidéo dans l'interaction gestuelle est antérieure à celle des capteurs cités précédemment. Bien que la résolution temporelle et spatiale de ces appareils varie beaucoup selon les modèles, elle est relativement standardisée sur les dispositifs mobiles actuels. Initialement destinée à des tâches simples (photo, vidéo), la caméra vidéo a été détournée pour être utilisée comme périphérique d'entrée en interaction gestuelle. On trouve des exemples dans [Hachet et al., 2005] (FIGURE 2.15 (a)) où l'utilisateur tient un dispositif équipé d'une caméra dans une main et une cible dans l'autre. Cette technique permet de récupérer les mouvements du dispositif en analysant les positions de la cible enregistrées par la caméra. L'utilisation d'une caméra embarquée pour détecter les mouvements de translation et de rotation du dispositif permet de faire défiler et de zoomer des images dans [Haro et al., 2005] (FIGURE 2.15 (b)) ou bien une carte dans Realeyes3D¹⁰. Francone *et al.* proposent également d'utiliser la caméra sur un iPad (ou iPhone) afin de détecter les mouvements de l'appareil par rapport à la tête de l'utilisateur ([Francone and Nigay, 2011]). Ceci permet d'estimer les rotations du dispositif, comme on pourrait le faire avec un gy-

10. <http://www.realeyes3d.com/>

roscope, mais surtout d'obtenir les angles de vision de l'utilisateur. Cette information peut être exploitée pour simuler un effet 3D où les objets apparaissent correctement orientés par rapport à l'angle de vision de l'utilisateur (FIGURE 2.16).



(a) Interface d'interaction gestuelle utilisant une carte et une caméra. L'utilisateur déplace une carte devant la caméra afin de faire bouger les objets présents sur son écran.

(b) Interface de navigation dans une liste d'images

FIGURE 2.15: Deux interfaces utilisant la caméra pour interagir gestuellement.



FIGURE 2.16: La métaphore de la boîte proposée par Fracone *et al.*. En changeant l'orientation de l'appareil l'utilisateur peut accéder aux parties cachées de l'interface.

Ecran tactile

Les écrans tactiles équipent les dispositifs commerciaux depuis l'arrivée des PDAs (personnel digital assistant) en 1993 mais se sont surtout répandus de manière exponentielle depuis l'arrivée de l'iPhone en 2007¹¹. D'abord résistifs et nécessitant l'utilisation d'une pression suffisante sur l'écran, ils ont ensuite été produits avec différents moyens technologiques (capacitif, caméra infrarouge¹², acoustique¹³). Cependant, les technologies utilisées sont dépendantes de la taille du dispositif. Ainsi, comme il est encore aujourd'hui difficile de réaliser des écrans capacitifs de grande taille, les technologies infrarouges sont préférées. Par contre, la plupart des écrans destinés aux smartphones sont aujourd'hui des écrans tactiles capacitifs.

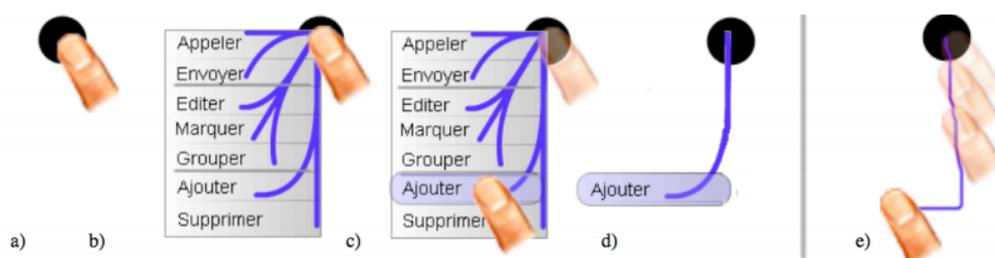


FIGURE 2.17: Le LeafMenu. De a) à d) les différentes étapes de l'interaction novice. e) l'interaction experte.



FIGURE 2.18: Le wavelet menu sur iPhone.

Parmi les techniques exploitant les possibilités des écrans tactiles, on peut par exemple citer Cyclostar et Rubbing [Malacria et al., 2010a, Olwal et al., 2008] qui permettent entre-

11. Le premier écran tactile remonte à 1972 avec le Plato IV d'IBM, un ordinateur pour l'enseignement numérique qui utilisait la technologie infrarouge pour détecter le contact du doigt sur l'écran.

12. La table multitouch Microsoft Surface, la majorité des tables interactives et certains tableaux interactifs. <http://www.microsoft.com/surface/en/us/default.aspx>

13. Les tableaux interactifs ultra-soniques par exemple. <http://www.mimio.fr/produits/mimio-teach/index.html>

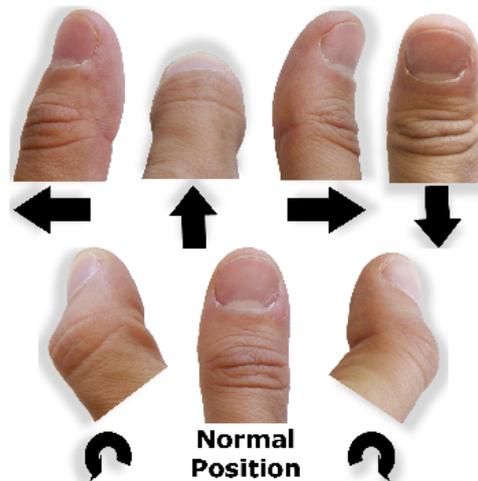


FIGURE 2.19: Les six différents types de roulement du bout du doigt à la base de la technique Microrolls.

autres de zoomer sur une photographie ou une carte grâce à de l'interaction gestuelle. Les LeafMenus [Bailly et al., 2008] (FIGURE 2.17) utilisent également des gestes sur écran tactile, mais pour activer les commandes d'un menu linéaire sur mobile. Les Marking Menus ([Kurtenbach and Buxton, 1991a]) et des techniques dérivées spécialement conçues pour les mobiles comme les Wavelet Menus [Francone et al., 2010](FIGURE 2.18) font de même dans le cas des menus circulaires. Les Microrolls [Roudaut et al., 2009b] (FIGURE 2.19) introduisent un autre type de geste pour activer des commandes. Incidemment cet article présente et compare quatre type de geste (Rubbing, slides, pans et Microrolls). Plus récemment on peut citer l'article de [Lü and Li, 2011] qui propose de faire des gestes représentant des symboles sur l'écran pour faciliter l'accès à des liens sur une page web d'un navigateur mobile.

Tous ces capteurs permettent donc de détecter des mouvements, du dispositif lui-même ou sur le dispositif, favorisant une interaction gestuelle. Il faut préciser qu'il est nécessaire de mettre en place des procédés logiciels algorithmiques pour permettre la reconnaissance des gestes à partir des données renvoyées par ces capteurs. Nous ne détaillerons pas ces différents procédés dont on trouvera des exemples dans [Kauppila, 2008, Schlömer et al., 2008].

2.2.2 Type de mouvement

Les gestes utilisés dans l'interaction mobile sont, comme nous l'avons vu, souvent catégorisés selon leur *style* ou leur *nature* [Karam and Schraefel, 2005, Wobbrock et al., 2009] et non pas selon les mouvements qui entrent en jeu lors de leur réalisation. Nous sommes donc partis d'une classification simple des mouvements comme on en trouve dans le vocabulaire

de la danse (discipline reposant sur le mouvement) notamment en différenciant les mouvements *fluides* des mouvements *impulsifs* (ou brusques). Afin d'apporter une description plus précise de l'ensemble des mouvements utilisés dans le contexte de l'interaction gestuelle nous avons également introduit une troisième catégorie : les mouvements oscillatoires. La [FIGURE 2.20](#) illustre ces trois catégories, mettant en valeur de manière schématique leurs différences relativement aux accélérations produites. Enfin, comme nous le verrons dans la dernière sous-section, tous ces types de mouvement peuvent être combinés. Il est à ce propos utile de remarquer que, comme l'écrit Guiard, les mouvements oscillatoires ne sont pas seulement une concaténation de mouvements [Guiard, 1993b], raison pour laquelle nous leur accordons une catégorie spécifique.

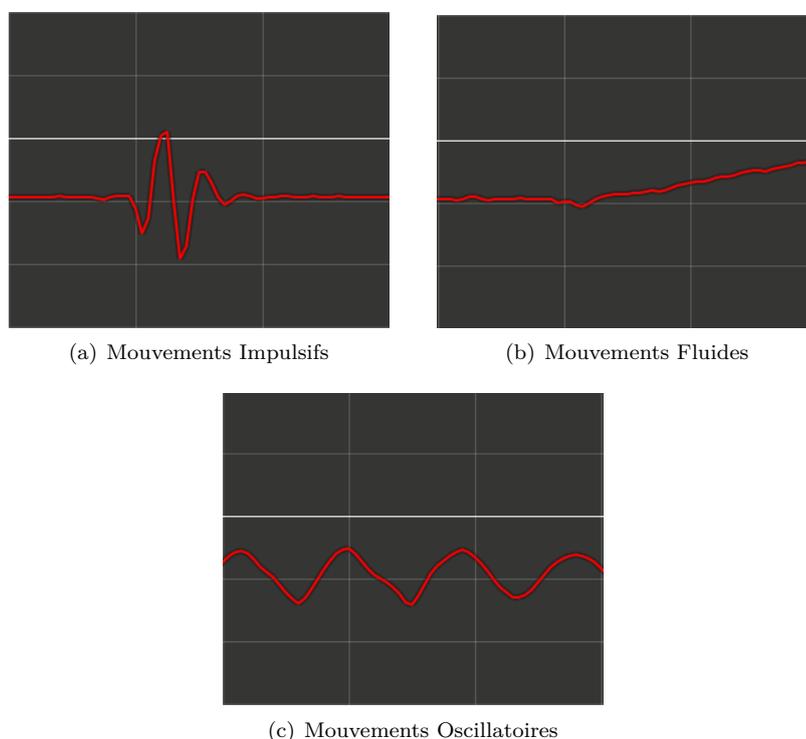


FIGURE 2.20: Les trois différents types de mouvement. Chacun des mouvements est différent des autres au niveau des accélérations en fonction du temps. L'axe des ordonnées représente l'accélération selon l'axe y en m/s^2 et l'axe des abscisses représente le temps.

Mouvements impulsifs

Les mouvements impulsifs ont pour principale caractéristique de mettre en jeu de fortes variations d'accélération pendant un laps de temps très court ¹⁴. Un exemple typique de

¹⁴. En mécanique, on définit souvent par impulsion une force dont la durée d'application est négligeable à l'échelle de temps où l'on considère les phénomènes étudiés.

gestes composés de mouvements impulsifs est le choc entre deux objets, dont la mesure résultante indique un pic d'accélération au moment de l'impact entre les deux objets. Plusieurs exemples de l'utilisation de gestes caractérisés par un mouvement impulsif ont été proposés dans la littérature. Dans [Linjama and Kaaresoja, 2004] (FIGURE 2.21 (a)) l'utilisateur tape sur les cotés d'un téléphone pour déplacer une bille virtuelle. L'application « Sketches » sur iPhone utilise une secousse pour effacer les modifications d'un dessin. Le téléphone et lecteur de musique Sony Ericsson W508 (ou également l'iPhone)¹⁵ permet de passer à une autre chanson lorsque l'utilisateur fait un geste brusque avec le dispositif. Enfin, la détection de chocs a été utilisée pour le couplage de deux dispositifs [Hinckley, 2003] (FIGURE 2.21 (b)), où l'utilisateur couple deux tablette PCs en les choquant l'un contre l'autre. Le même procédé sert dans l'application Bump sur iPhone à échanger des données entre deux dispositifs mobiles.



(a) Exemple de mouvement impulsif permettant de déplacer une bille sur le mobile (preuve de concept sous forme de jeu)



(b) Choc entre deux dispositif pour les coupler.

FIGURE 2.21: Deux exemples de gestes constitués de mouvements impulsifs.

Mouvements fluides

Les mouvements fluides sont plus lents, avec pour conséquence des variations d'accélération de faible amplitude étalées sur des durées plus longues. Nous pouvons donner comme exemple de gestes caractérisés par des mouvements fluides, le geste d'inclinaison qui consiste

15. <http://www.sonyericsson.com/>

à pencher un objet afin d'en changer l'orientation ou la position [Rahman et al., 2009, Rekimoto, 1996]. Par exemple dans « Rock'n'Scroll » [Bartlett, 2000] (FIGURE 2.23) l'utilisateur incline son dispositif mobile pour faire défiler une image dans différentes directions. Il en va de même dans la solution proposée par [Oakley and Park, 2007] (FIGURE 2.24), où l'utilisateur incline son PDA pour sélectionner des items dans un menu. Le navigateur Internet Opéra sur HTC Touch HD change l'orientation de l'écran lorsque l'utilisateur tourne simplement son téléphone. Le même procédé est utilisé sous Safari sur l'iPhone ou dans la technique présentée dans [Hinckley et al., 2000]. Enfin l'étude de [Rahman et al., 2009] présente une analyse des différentes possibilités de réalisation de mouvements d'inclinaison (tilt) autour du poignet. Les auteurs y étudient les limitations des utilisateurs à réaliser des inclinaisons et rotations du poignet (flexion/extension, pronation/supination) (FIGURE 2.22), déterminant le nombre maximum de secteurs angulaires pouvant être différenciés par un utilisateur avec un taux d'erreur raisonnable.

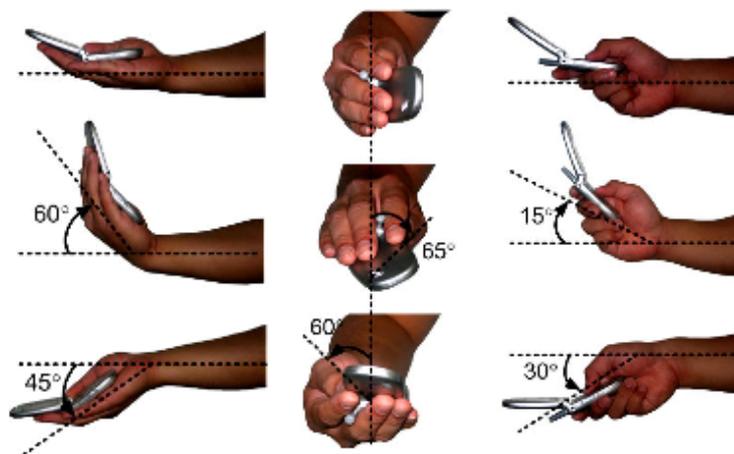


FIGURE 2.22: Rahman *et al.* étudient les possibilités du poignet pour réaliser des gestes de rotation (flexion/extension, pronation/supination)



(a) Position neutre. L'image est centrée.

(b) Inclinaison vers le bas pour atteindre la partie droite de l'image.

(c) Inclinaison vers le haut pour atteindre la partie gauche de l'image.

FIGURE 2.23: Rock'n'Scroll.

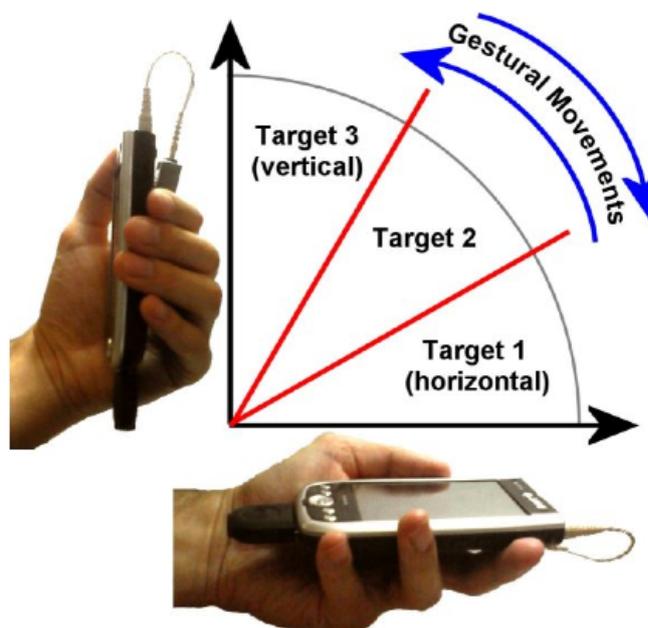


FIGURE 2.24: Le Motion Marking Menu [Oakley and Park, 2007]. L'utilisateur incline le dispositif pour atteindre les différents items du menu.

Mouvements Oscillatoires

Les mouvements oscillatoires sont obtenus par la répétition d'un mouvement impulsif ou fluide. Une propriété particulièrement intéressante des mouvements oscillatoires est qu'ils peuvent théoriquement être effectués 'indéfiniment', l'utilisateur ne rencontrant jamais les limites physiques de la surface ou de l'espace dans lequel ces gestes sont réalisés. Motion Pointing [Fekete et al., 2009] est un exemple de technique exploitant des gestes oscillatoires. Elle permet à un utilisateur de sélectionner des widgets sans avoir besoin de pointer avec curseur. Chaque widget affiche un pixel animé (appelé "pilote") qui effectue une oscillation elliptique avec une combinaison unique de paramètres géométriques (échelle, excentricité, inclinaison) et cinématiques (fréquence et sens de l'oscillation). L'utilisateur sélectionne une cible en reproduisant son oscillation spécifique (FIGURE 2.25). Cyclostar [Malacria et al., 2010a] (FIGURE 2.26), que nous avons déjà cité, utilise également des mouvements oscillatoires elliptiques afin de contrôler la navigation (zoom, pan, etc.). Nancel dans l'article [Nancel et al., 2011] propose d'utiliser ce type de mouvement pour interagir avec une surface d'affichage de grande taille (WILD¹⁶) et ce en utilisant un dispositif mobile ou la main (FIGURE 2.27). Ces mouvements peuvent aussi bien être réalisés sur une surface tactile qu'avec le dispositif lui-même, par exemple dans l'application *Shake-it* sur iPhone où l'utilisateur doit secouer (répétition de mouvements impulsifs) son dispositif afin de faire monter son score au maximum.

16. <http://insitu.lri.fr/Projects/WILD>

Button#13 	Button#14 	Button#15 
Button#10 	Button#11 	Button#12 
Button#7 	Button#8 	Button#9 
Button#4 	Button#5 	Button#6 
Button#1 	Button#2 	Button#3 

FIGURE 2.25: Motion Pointing. L'utilisateur reproduit l'oscillation affichée sur un item pour le sélectionner.

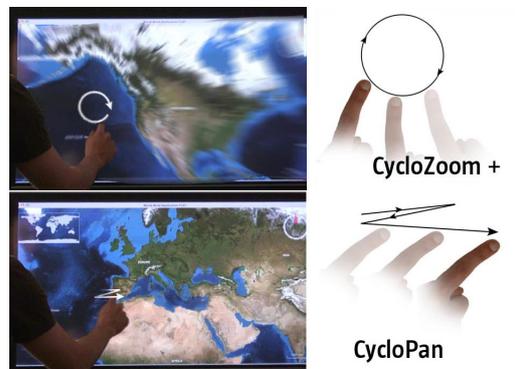


FIGURE 2.26: CycloZoom et CycloStar : deux techniques permettant respectivement de zoomer et panner sur une surface tactile.

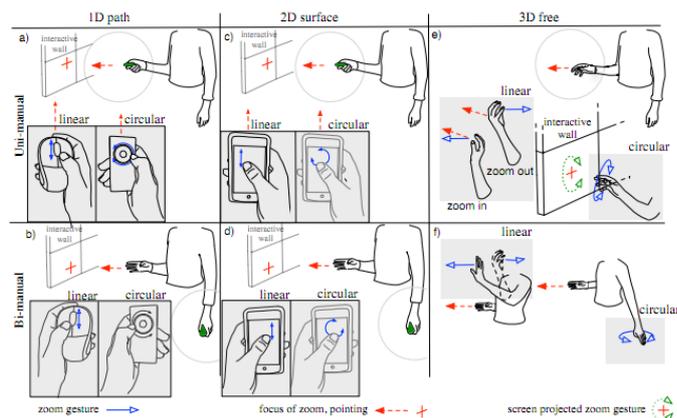


FIGURE 2.27: Représentation des différents gestes proposés par Nancel pour interagir sur très grand écran, (incluant des gestes basés sur des mouvements oscillatoires).

Combinaison de mouvements

Certains gestes requièrent l'exécution d'une combinaison de mouvements impulsifs et/ou fluides, par exemple lorsque l'utilisateur réalise un geste représentant une forme géométrique angulaire (un triangle) pour démarrer la lecture d'une vidéo [Rehm et al., 2008]. De même, dans Gesture Avatar [Lü and Li, 2011] (FIGURE 2.28) l'utilisateur dessine des symboles (des lettres) sur son dispositif mobile, qui sont utilisés comme raccourcis. La réalisation de symboles (dans l'écriture par exemple) alterne généralement des séquences de mouvements fluides et des séquences de mouvements impulsifs.

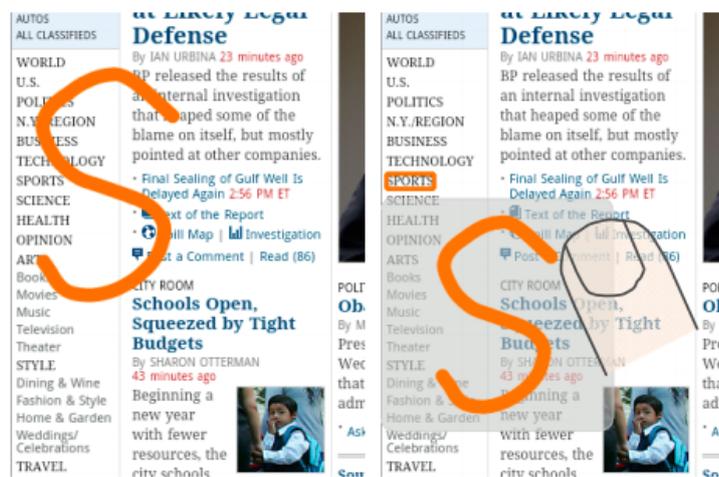


FIGURE 2.28: Gesture Avatar

2.2.3 Type de mapping

Par mapping nous entendons ici la relation entre la forme du geste et la fonction réalisée. Nous distinguons quatre catégories : le mapping direct, symbolique, métaphorique et abstrait. Le choix de ces catégories s'inspire de la taxonomie de Wobbrock [Wobbrock et al., 2009] pour les gestes sur surface tactile. Nous avons préféré cette catégorisation à celles de Karam et Cadoz, qui classifient le geste du point de vue de sa fonction et non de son association entre la forme et la fonction. Cette notion a également été utilisée par Malacria [Malacria, 2011] avec certaines différences (par exemple il ne reprend pas les gestes métaphoriques) sous l'intitulé *nature des gestes* pour sa classification des gestes sur surface 2D.

Mapping direct

Le mapping direct permet de créer un lien direct entre les objets virtuels avec lesquels l'utilisateur interagit et le geste utilisé, de la même manière qu'avec un objet physique du monde réel. Les gestes de pointage font partie de cette catégorie. Par exemple dans [Ballagas, 2006] (FIGURE 2.29) l'utilisateur se sert de son téléphone pour déplacer un curseur sur un écran. Lorsque l'utilisateur déplace son mobile vers la droite il déplace le curseur dans la même direction. Dans [Hinckley et al., 2000] (FIGURE 2.30), lorsque l'utilisateur fait pivoter son dispositif de 90 degrés dans le plan de l'écran, l'orientation du texte change pour s'adapter à la nouvelle orientation, en portrait ou en paysage. Dans Motion-Lens¹⁷ l'utilisateur déplace son dispositif pour naviguer dans une carte : lorsqu'il translate le mobile vers le haut, il accède à la partie supérieure de la carte. Il en va de même avec les gestes utilisés dans Wii Tennis ou iGolf (sur iPhone), où l'utilisateur réalise un geste similaire à celui qu'il devrait effectuer dans la réalité pour frapper la balle de tennis ou de golf.



FIGURE 2.29: Téléphone utilisé comme souris pour un écran distant.

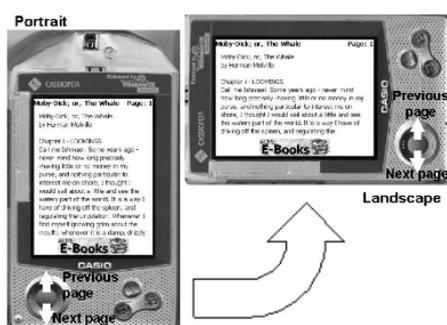


FIGURE 2.30: Réorientation du texte en fonction de l'orientation de l'appareil.

17. <http://www.realeyes3d.com/>

Mapping métaphorique

Les métaphores associent deux termes appartenant à des champs lexicaux différents. La plus fameuse est la métaphore du bureau garni de sa corbeille, de son presse-papiers, de ses dossiers et autres documents. Le mapping métaphorique se fonde sur le même principe de transposition, ce qui le distingue du mapping direct qui est basé sur une analogie directe. Nous rencontrons de nombreux exemples dans la littérature IHM et dans les applications commerciales. 3D Eyes Tracking¹⁸ simule une profondeur de l'écran en 3 dimensions qui permet de voir les éléments disposés en dehors de l'écran lorsque l'utilisateur incline son dispositif. On retrouve le même procédé dans les travaux de Francone sur iPad [Francone and Nigay, 2011] (FIGURE 2.31). Dans Tilt-Text [Wigdor and Balakrishnan, 2003] (FIGURE 2.32) l'utilisateur appuie sur une touche de son téléphone puis l'incline pour faire glisser les lettres correspondant à cette touche du clavier (trois lettres par touche et un chiffre) afin de sélectionner la lettre désirée.

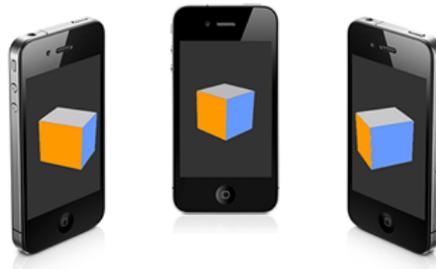


FIGURE 2.31: Head-Coupled Perspective Technique.

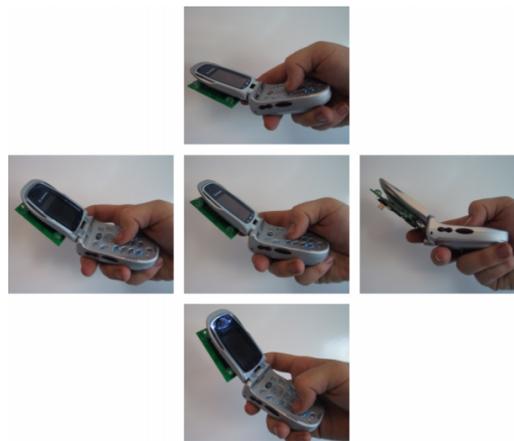


FIGURE 2.32: TiltText

18. <http://www.tat.se/>

Mapping symbolique

Le mapping symbolique se fonde sur l'association d'un geste représentant un symbole qui est lui-même associé à une fonction. Le geste symbolique dessine un symbole dans le plan ou dans l'espace, le plus souvent un caractère alphanumérique ou une forme géométrique élémentaire. On peut également trouver des symboles propres à une application. Ces gestes sont désignés comme "gestes sémaphoriques" dans [Karam and Schraefel, 2005], et comme des *descriptions visuelles* par [Wobbrock et al., 2009]. Le chemin spatial que l'utilisateur fait parcourir au mobile, ou qu'il dessine sur l'écran, évoque une forme ou un symbole du monde réel ou numérique. Par exemple dans [Rehm et al., 2008] (FIGURE 2.34) les gestes réalisés dans l'espace représentent les symboles présents sur les boutons des commandes de lecture utilisés pour les magnétoscopes : ainsi, l'utilisateur dessine un carré dans l'espace avec le dispositif mobile pour arrêter la lecture. Dans [Kauppila et al., 2008] les gestes représentent des chiffres et des lettres, associés à des commandes du dispositif : l'utilisateur utilise les lettres pour accéder aux fonctions du téléphone (lecture/écriture de messages) et, bien sûr, les chiffres pour composer un numéro. Gesture Avatar [Lü and Li, 2011] utilise aussi ce type de geste pour exécuter des commandes dans un navigateur web.

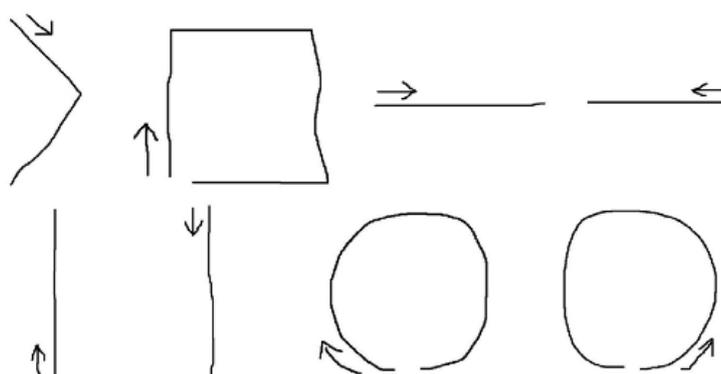


FIGURE 2.33: Symboles associés à des commandes d'un magnéscope.

command	M W P S C Z
number	1 2 3 4 5 6 7 8 9
confirmation	○ →
backlight	○ ↻

FIGURE 2.34: Gestes symboliques utilisés dans les travaux de [Kauppila et al., 2008].

Mapping arbitraire

Nous désignons comme arbitraire un mapping qui n'implique pas de lien direct, ni métaphorique, ni symbolique évident entre le geste et l'objet d'interaction [Wobbrock et al., 2009]. Les Marking Menus [Kurtenbach and Buxton, 1991b] constituent un exemple classique de ce type de mapping : bien que liés à la représentation visuelle du menu, les gestes sont généralement arbitraires par rapport aux commandes qu'ils activent. On peut également citer les raccourcis gestuels utilisés dans les menus linéaires classiques proposés dans [Appert and Zhai, 2009] (FIGURE 2.35). Ils n'ont en effet le plus souvent pas de sens particulier. De même, appuyer une fois pour mettre la musique en pause, deux fois pour passer à la piste suivante et trois fois pour revenir à la précédente, comme avec les écouteurs de l'iPhone, relève de cette catégorie de mapping.

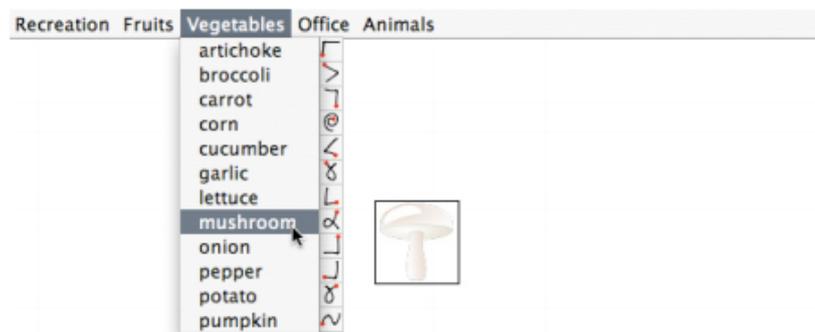


FIGURE 2.35: Gestes associés à des commandes d'un menu linéaire avec un mapping arbitraire.

Ces quatre catégories ne sont bien sûr pas totalement étanches, certains mappings étant à la frontière de deux d'entre-elles. Il est parfois difficile de définir le type de mapping, l'appréciation restant quelques fois subjective, certains voyant du sens là où d'autres ne voient qu'un choix arbitraire. Elles permettent cependant de mieux cerner la nature dominante des gestes et le lien entre leur forme et leur fonction. De plus, le type de mapping que nous réalisons pour interagir avec un dispositif mobile a un lien fort avec le contrôle et le retour d'information. Nous considérons dans la section suivante le type de contrôle.

2.2.4 Type de Contrôle

Le type de contrôle définit le lien entre le geste réalisé et les variables de l'application que le geste permet de manipuler. Le contrôle peut être soit discret, soit continu [Pook et al., 2000b]. Les gestes composés de mouvements impulsifs correspondent

souvent à un contrôle discret et les gestes fluides à un contrôle continu comme nous pourrions le voir plus loin (FIGURE 2.41). Par exemple, dans le cas des interfaces graphiques 2D, un clic souris (geste impulsif) permet de faire basculer un bouton à états (contrôle discret à deux valeurs : actif, inactif) alors qu'il faut faire glisser le curseur d'un potentiomètre (mouvement fluide) pour choisir une valeur continue (par exemple le volume sonore). Il n'existe cependant pas de relation univoque entre type de mouvement et type de contrôle lorsque l'on prend en compte le temps ou les répétitions, donc les mouvements oscillatoires. On peut par exemple contrôler le volume sonore (variable continue) en appuyant un certain temps sur une touche (mouvement impulsif en début et fin de l'interaction). ShakeIt sur iPhone utilise également des mouvements impulsifs pour contrôler une valeur, l'utilisateur devant secouer son mobile pour augmenter son score. Ce logiciel se sert de l'effet de répétition pour contrôler de manière continue ses points.

Contrôle discret

L'interaction gestuelle est souvent associée au contrôle discret. Ce type d'interaction, comme expliqué précédemment, est souvent associé à un mapping symbolique et à des gestes composés de mouvements impulsifs comme dans [Rehm et al., 2008], où les commandes liées à la lecture du magnétoscope ne sont exécutées qu'une fois le geste de l'utilisateur terminé. Ce n'est donc qu'en fin de geste que l'utilisateur peut savoir si le système l'a correctement interprété. On trouve quelques exceptions comme OctoPocus [Bau and Mackay, 2008], une extension des Marking menus (FIGURE 2.36) où l'utilisateur bénéficie d'une suggestion visuelle (feedforward) des gestes qu'il peut encore 'atteindre' en poursuivant son geste.

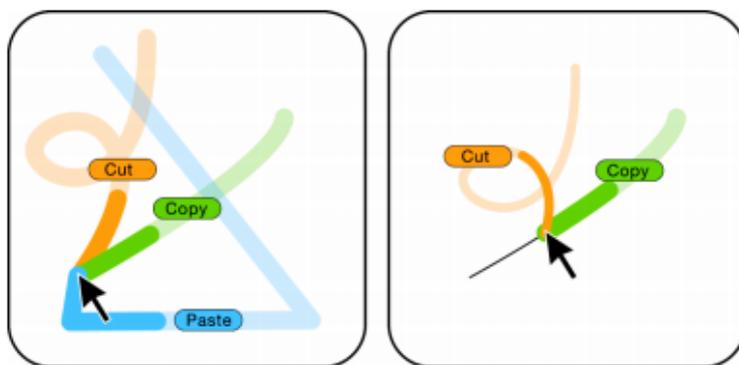


FIGURE 2.36: Octopocus : l'utilisateur clique et voit apparaître les différents chemins possibles. Lorsqu'il se déplace le nombre de chemins se réduit, ceci ne laissant apparaître que les commandes qu'il peut encore atteindre.

Le lecteur de musique du Sony Ericsson W508, de l'iPod Touch et de l'iPhone offrent des illustrations de contrôle discret par un geste impulsif : l'utilisateur secoue le dispositif pour changer de chanson. Dans [Linjama and Kaaresoja, 2004], que nous avons déjà cité dans la

section sur les mouvements impulsifs, l'utilisateur peut taper sur les côtés d'un téléphone mobile, cette impulsion mettant en mouvement une balle. Cependant des gestes fluides peuvent aussi servir à contrôler des valeurs discrètes, comme dans [Oakley and Park, 2007], où l'utilisateur sélectionne un item dans un menu en inclinant son dispositif jusqu'à obtenir l'angle adéquat, chaque item correspondant à un secteur angulaire d'inclinaison prédéfini. Dans [Pering et al., 2007] l'utilisateur pointe vers un certain appareil avec son téléphone, une inclinaison du téléphone activant de manière discrète la connexion à cet appareil (FIGURE 2.37).

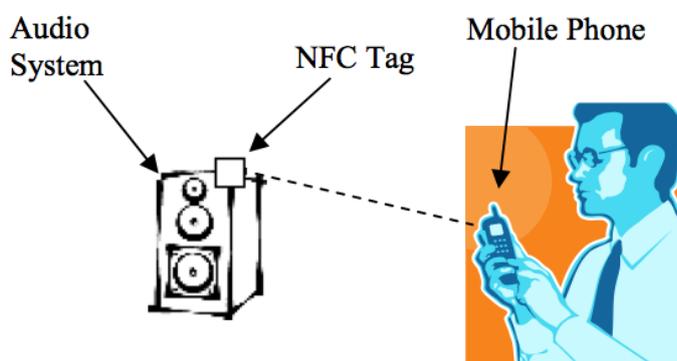


FIGURE 2.37: Geste de pointage pour coupler deux appareils entre eux.

Contrôle continu

Ce type de contrôle est souvent associé à des mouvements fluides car la fonction d'association est directe entre les valeurs renvoyées par le capteur et les valeurs manipulées. Par exemple dans [Ballagas, 2006] l'utilisateur déplace son dispositif mobile devant un écran d'ordinateur pour déplacer le curseur sur l'écran. Dans un autre exemple (RealEyes3D FIGURE 2.38), l'utilisateur incline son dispositif pour faire défiler la carte qu'il affiche sur son dispositif. Enfin dans [Hinckley et al., 2000], l'utilisateur incline plus ou moins son dispositif pour contrôler le niveau de zoom d'une application. Le contrôle continu est parfois associé à des mouvements oscillatoires. CycloZoom [Malacria et al., 2010b] en est un bon exemple puisque l'échelle du zoom est mise à jour de manière continue durant la réalisation d'un geste elliptique. Le contrôle continu nécessite généralement un retour d'information en temps réel afin que l'utilisateur puisse ajuster la valeur qu'il contrôle.

2.2.5 Type de retour d'information

Le retour d'information (feedback) permet à l'utilisateur de percevoir l'effet de l'action qu'il vient de réaliser sur le système qu'il contrôle. Les dispositifs de sortie communément



FIGURE 2.38: RealEyes3D, l'utilisateur incline son dispositif pour naviguer dans la carte.

utilisés sur mobile [Roudaut and Lecolinet, 2007] sont : le retour visuel (écran, lunettes, projecteur), le retour sonore (haut-parleur, écouteurs) et le retour haptique (dispositifs tactiles, vibreurs).

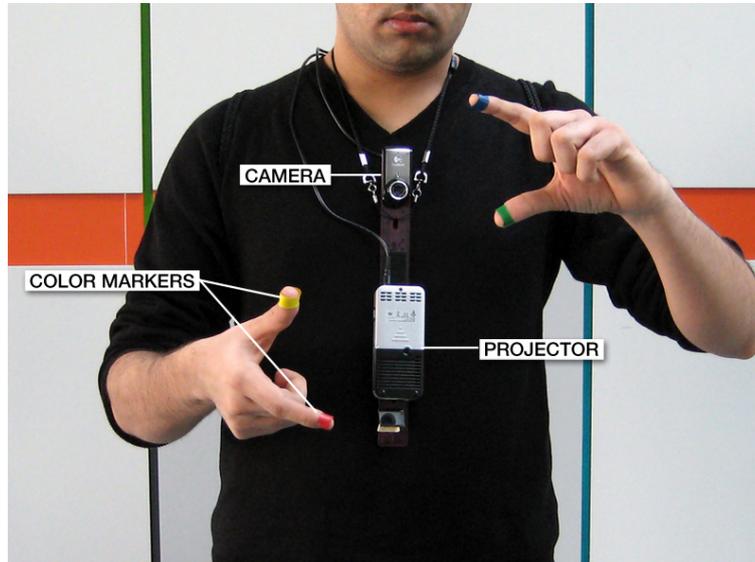


FIGURE 2.39: SixthSense

La plupart des techniques d'interaction gestuelle, et en particulier celles que nous avons citées jusqu'ici, utilisent des retours d'information de type visuel. Par exemple, dans SixthSense [Mistry et al., 2009] (FIGURE 2.39) l'utilisateur interagit gestuellement devant la ca-

méra de son dispositif tandis que le retour d'information est fourni par un projecteur placé autour de son cou : il peut ainsi manipuler directement l'image qui est projetée devant lui sur un support quelconque (mur, main, journal, etc.). Quelques techniques utilisent un retour d'information sonore pour permettre une interaction non visuelle (*eyes-free* interaction) : c'est le cas avec les mobiles Sony Ericsson où l'utilisateur peut secouer son dispositif pour changer de chanson. Enfin quelques techniques proposent un retour haptique ou tactile, comme dans [Linjama and Kaaresoja, 2004] où l'utilisateur sent une vibration lors de la collision entre la balle virtuelle et le bord de l'écran. On trouve également Tilt and Feel [Oakley et al., 2004] (FIGURE 2.40) qui propose un retour d'information vibrotactile.

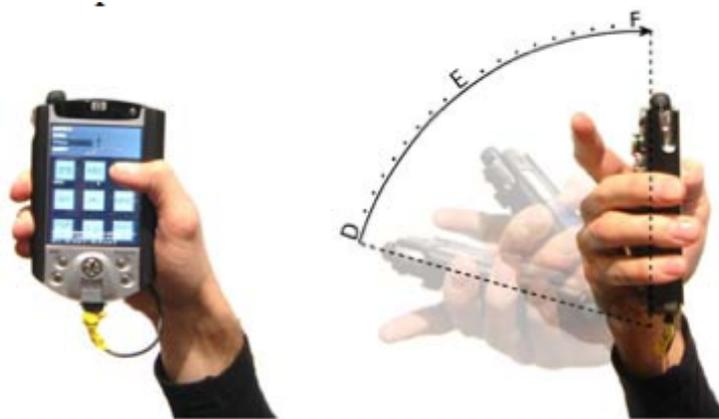


FIGURE 2.40: Tilt and Feel

2.2.6 Contexte

Le contexte d'interaction est une dimension parallèle de cet espace. Elle n'a que très peu été prise en compte dans les taxonomies précédentes. Le concepteur doit en effet penser la technique au travers du contexte. Cette notion de contexte d'interaction est souvent définie selon trois critères [Calvary et al., 2002] : l'utilisateur, la plate-forme et l'environnement. Dans le cas de l'interaction gestuelle sur dispositif mobile, l'utilisateur et la plate-forme sont définis par l'interaction elle-même, la plate-forme étant un dispositif mobile et l'utilisateur une personne pouvant utiliser un dispositif mobile. L'environnement par contre n'est pas fixe puisque l'interaction se passe par définition dans un univers variable de par la mobilité de l'utilisateur. Le contexte influence les facteurs de forme, d'amplitude et de durée des gestes utilisables en situation de mobilité. Lorsque l'utilisateur se trouve dans une situation de mobilité induisant un espace plus étroit, lorsqu'il est en présence d'autres personnes ou d'éléments perturbateurs, lui sera-t-il toujours possible de réaliser le geste imaginé par le concepteur d'interaction ? Un geste exigeant de l'amplitude, coûteux en concentration ou chargé d'un contenu symbolique problématique peut en pratique s'avérer difficilement utilisable en situation de mobilité comme le montre Rico [Rico and Brewster, 2010a].

Espace d'interaction

L'utilisateur est contraint dans ses mouvements par son espace d'interaction. De ce fait, l'amplitude des gestes qu'il peut effectivement réaliser en est tributaire [Rico and Brewster, 2010a]. En situation de mobilité, comme lors de l'utilisation d'un téléphone dans les transports en commun, il ne pourra réaliser que des gestes discrets de faible amplitude, comme dans [Wigdor and Balakrishnan, 2003] en inclinant simplement le poignet vers la gauche ou la droite pour écrire un message. Il en va de même, dans [Linjama and Kaaresoja, 2004], où l'utilisateur tape sur les côtés du mobile pour faire bouger une balle virtuelle.

Niveau d'attention et altérations liées à l'usage

L'attention que peut consacrer l'utilisateur à la réalisation d'une interaction est généralement altérée en situation de mobilité. Son attention est alors partagée entre l'interaction avec l'application et l'environnement immédiat, qu'il s'agisse des personnes ou des divers flux (comme les annonces de stations dans le bus, les publicités sonores ou visuelles, etc.). La précision de l'interaction en sera nécessairement affectée, ce qui nécessite alors l'usage de gestes simples ne demandant pas trop d'attention. Francone présente bien cette idée dans le papier sur le Wavelet Menu [Francone et al., 2010]. Il montre en effet que l'utilisation de son menu en situation de mobilité fait nettement chuter les performances des utilisateurs. De plus, outre le niveau d'attention limité, d'autres éléments peuvent dégrader l'interaction comme par exemple le fait que l'utilisateur ne puisse utiliser qu'une seule main, ou qu'il soit soumis aux vibrations d'un véhicule, une situation fréquente en situation de mobilité.

Adaptation à l'utilisateur

Le retour d'information doit être adapté en conséquence en étant clair (non ambigu) et facilement interprétable par l'utilisateur. L'interaction par secousses du lecteur de musique Sony Ericsson W508 ou de l'iPhone pour passer à une nouvelle chanson constitue un bon exemple. Le geste est simple, ne demande pas une grande précision et l'utilisateur peut le réaliser sans regarder son dispositif mobile (*eyes-free selection*). On observera de surcroît que l'information en retour est obtenue instantanément par l'utilisateur puisqu'il entend la nouvelle chanson. Cependant cet exemple, qui montre bien la nécessité d'un geste et d'un feedback simples, est aussi une bonne illustration des limitations des gestes 'trop' simples. En effet il est fréquent de changer de chanson sans le vouloir alors que le dispositif est rangé dans une poche et que l'on se déplace. Cet exemple est mis en évidence dans le travail de Williamson [Williamson et al., 2011] qui explique que dans un contexte réel la manière même

de marcher d'un utilisateur a un impact significatif sur les seuils de détection des gestes.

Acceptation sociale

Enfin l'environnement social dans lequel l'utilisateur interagit impose des codes et des règles. La signification d'un geste peut être perçue différemment selon les individus. On sait notamment que certains gestes sont compris différemment selon les cultures [Rehm et al., 2008]. Par exemple, secouer sa main devant son visage n'a aucun sens en France alors que ce geste exprime une forme de moquerie pour les Allemands. En raison du facteur de forme ou d'amplitude, des gestes potentiellement intéressants du point de vue de l'interaction en situation de mobilité peuvent être perçus par l'entourage immédiat de l'utilisateur comme inappropriés, voire ridicules. Les gestes susceptibles de servir en IHM doivent être conçus de manière à ne jamais mettre l'utilisateur ou les personnes qui l'entourent dans une situation embarrassante, ce que développe Rico dans son article [Rico and Brewster, 2010a]. En effet les gestes que les utilisateurs sont susceptibles de réaliser dans l'environnement restreint du foyer, par exemple en utilisant un Kinect ou une Wiimote, ne sont plus acceptables lorsqu'ils se trouvent dans une situation incluant une population d'inconnus (par exemple dans la rue).

2.3 Conclusion

Dans ce chapitre nous avons présenté un espace de classification pour l'interaction gestuelle physique sur dispositifs mobiles. Cet espace est constitué de six dimensions :

- Le type de Capteur
- Le type de Mouvement
- Le type de Mapping
- Le type de Contrôle
- Le type de Retour d'information
- Le Contexte.

Ces dimensions permettent de traiter l'interaction gestuelle selon plusieurs aspects : technologie, cinématique, signification et contexte. Nous pouvons ainsi classer les différentes techniques citées dans ce chapitre dans un tableau à quatre dimensions ([FIGURE 2.41](#)). En effet notre tableau ne prend pas en compte le retour d'information puisque la très grande majorité des techniques (>90%) ont un retour visuel, ni le contexte qui est une dimension parallèle. Le fait que la majorité des techniques d'interaction gestuelle utilisent un retour visuel vient du fait que ce type de retour offre généralement de meilleures performances, comme le montrent plusieurs études [Dearman et al., 2010, Ramos et al., 2004].

Comme on peut le voir sur le tableau certaines zones restent inexplorées. Il faut cependant utiliser le mot 'inexploré' avec précaution car il n'est généralement pas possible de savoir si une zone a été explorée sans succès. Le système de publication actuel favorisant la soumission de résultats positifs, il est assez rare de voir publiés des résultats montrant qu'il ne faut pas aller dans une certaine voie. Il est donc parfois difficile de juger des raisons de l'absence de techniques dans certaines parties du tableau. Cependant les parties remplies nous apportent de l'information. On remarquera notamment que l'interaction gestuelle est souvent réalisée avec des mouvements fluides, sûrement en raison de l'analogie directe que l'on trouve avec la manipulation des objets du quotidien. Ensuite, on peut remarquer que le lien entre les gestes réalisés et l'objet que l'on manipule est souvent fort, que ce soit par le biais d'une métaphore ou d'une interaction directe. On pourra également remarquer que de par leur nature les gestes utilisant un mapping symbolique se prêtent difficilement à un contrôle continu de l'interaction.

Type de Capteur

	Accéléromètre		Boussole
	Gyroscope		Caméra
	Ecran tactile		

		Type de Mapping				Type de Contrôle
		Direct	Symbolique	Métaphorique	Arbitraire	
Type de Mouvements	Impulsifs					Continu
	Fluides	 				
	Oscillatoires					
		Direct	Symbolique	Métaphorique	Arbitraire	Discret
Type de Mouvements	Impulsifs					
	Fluides					
	Oscillatoires					

FIGURE 2.41: Tableau de classification des techniques de la littérature.

Liste des travaux représentés dans la [FIGURE 2.41](#).

- 1.[Roudaut et al., 2009a], 2.[Ballagas, 2006], 3.[Bartlett, 2000], 4.[Fekete et al., 2009],
- 5.[Ferscha et al., 2008], 6.[Francone et al., 2010], 7.[Francone and Nigay, 2011],
- 8.[Hachet et al., 2005], 9.[Haro et al., 2005], 10.[Harrison and Hudson, 2009],
- 11.[Hinckley, 2003], 12.[Hinckley et al., 2000],13.[Kauppila et al., 2008],
- 14.[Kurtenbach and Buxton, 1991b], 15.[Ladetto and Merminod, 2002],
- 16.[Linjama and Kaaresoja, 2004], 17.[Lü and Li, 2011], 18.[Malacria et al., 2010a],
- 19.[Mistry et al., 2009], 20.[Nancel et al., 2011], 21.[Oakley and Park, 2007],
- 22.[Olwal et al., 2008], 23.[Pering et al., 2007], 24.[Pook et al., 2000b],
- 25.[Rehm et al., 2008], 26.[Rekimoto, 1996], 27.[Roudaut et al., 2009b],
- 28.[Wigdor and Balakrishnan, 2003]; T :TimeTilt, J :JerkTilt, F :Flick-and-Brake.

3

Mouvements sur le dispositif

Sommaire

3.1	Introduction	58
3.1.1	Le Flick	58
3.2	Flick-and-Brake	64
3.2.1	Contexte	64
3.2.2	Concept général	65
3.2.3	Évaluation	72
3.2.4	Conclusion	77
3.3	Semantic Flicking	79
3.3.1	Contexte	79
3.3.2	Description	79
3.3.3	Prototypes	81
3.3.4	Conclusion	84
3.4	Long Flick	85
3.4.1	Contexte	85
3.4.2	Définitions	85
3.4.3	Exploitation	88
3.4.4	Conclusion	91

3.1 Introduction

Si l'on demande aujourd'hui à quelqu'un de faire défiler une page web sur un smartphone, il est probable que cette personne pose son doigt sur l'écran tactile et effectue un geste de lancer (Flick) pour se déplacer dans la page. Ce qui a fait le succès de cette technique d'interaction, c'est l'analogie évidente avec le monde réel qu'elle propose. Les gestes comme le Flick ont été introduits sur les surfaces interactives pour contourner les limitations de ces dernières. Du fait de la place qu'elle occupe, une barre de défilement n'est pas appropriée pour naviguer dans une liste sur l'écran d'un dispositif de petite taille comme un smartphone. De même, l'habituelle molette de la souris n'est en général pas disponible sur une surface interactive. Il a donc fallu recourir à de nouvelles façons d'interagir. Il est de plus souhaitable que ces techniques aient une *affordance* suffisante pour que l'utilisateur fasse abstraction des interactions indirectes qu'il connaît déjà. Le Flick possède cette propriété : l'utilisateur sait ce qu'il doit faire pour déplacer ou faire défiler un document (FIGURE 3.25) dès la première utilisation.

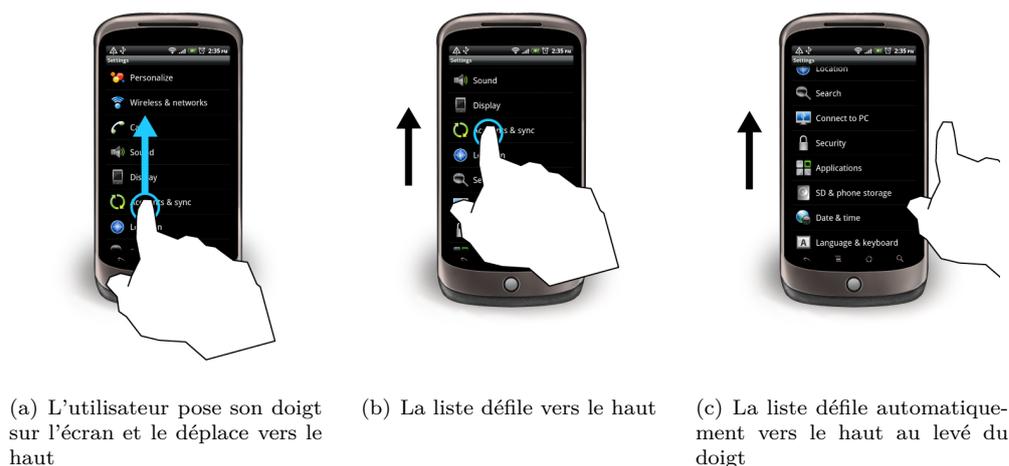


FIGURE 3.1: Le geste de Flick et le défilement (la flèche bleue représente le mouvement du doigt et la flèche noire celui de la liste).

3.1.1 Le Flick

Le geste de Flick consiste en un mouvement dans une direction permettant à l'utilisateur d'activer le défilement d'un document (liste, image, etc) ou le déplacement d'un objet. Initialement développée pour déplacer des objets sur de grandes surfaces comme les tables interactives, cette technique s'est montrée particulièrement efficace pour le défilement de listes sur les petits écrans des smartphones. Elle est aujourd'hui présente dans la quasi intégralité des interfaces pour téléphones mobiles, généralement sous le nom de *Kinetic scrolling*. Cette technique particulièrement appréciée par les utilisateurs apparaît même maintenant

sur les trackpads¹ et les souris² de fabrication récente. Cette popularité nous a conduit à nous intéresser plus précisément à cette technique, avec pour objectif d'améliorer la navigation dans les interfaces mobiles.

Après une revue de l'état de l'art, nous présentons une description détaillée du Flick. Nous décrivons l'anatomie d'un geste de Flick et présentons ensuite les aspects de cette technique que nous avons étudiés plus particulièrement ainsi que trois nouvelles techniques d'interaction que nous avons développées dans ce contexte : le Flick-and-Brake, le Semantic Flicking et le LongFlick.

État de l'art

Shuffle, Throw or Take It! a introduit l'idée de lancer un objet sur une table [Geissler, 1998] pour permettre à plusieurs utilisateurs de s'envoyer des objets sur un mur interactif. Le geste à effectuer est un mouvement d'aller-retour semblable à celui que fait un utilisateur lorsqu'il prend de l'élan pour faire glisser un objet sur une surface. La vitesse du déplacement est fonction du rapport de longueur entre le mouvement aller et le mouvement retour et ne correspond donc pas à la vitesse au moment du relâchement (comme c'est le cas avec la plupart des implementations actuelles du flick). L'article n'explique pas si une action explicite d'un utilisateur est nécessaire pour qu'il interrompe le déplacement de l'objet ou si un coefficient de friction est appliqué. Wu et Balakrishnan [Wu and Balakrishnan, 2003] ont ensuite proposé d'utiliser un geste de lancer pour que plusieurs utilisateurs autour d'une table interactive puissent s'envoyer des objets depuis et vers leurs espaces privés respectifs (FIGURE 3.2). Pour ce faire, l'utilisateur doit déplacer l'objet en suivant une trajectoire vers l'espace privé du destinataire. L'objet lui est envoyé si la vitesse du mouvement au relâchement est supérieure à un certain seuil.

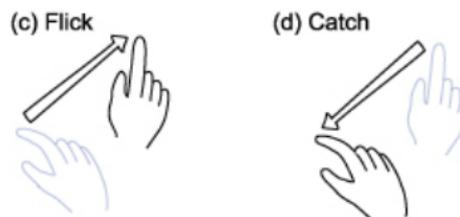


FIGURE 3.2: Gestes de Flick pour lancer et attraper un objet sur une table interactive.

SuperFlick augmente la rapidité et la précision d'un lancer d'objet sur une table interactive [Reetz et al., 2006]. Par défaut, un objet lancé sur la table avec SuperFlick déplace un

1. Apple Magic Trackpad : <http://www.apple.com/fr/magictrackpad/>
 2. Apple Magic Mouse : <http://www.apple.com/fr/magicmouse/>

objet dans la direction du mouvement et applique un coefficient de friction de telle manière que sa vitesse de déplacement se réduise progressivement. Le système est donc capable de pré-calculer la position d'arrêt de l'objet lancé. Un "avatar" de l'objet est affiché en transparence dès que l'animation de déplacement de l'objet démarre. Si l'utilisateur souhaite accélérer et/ou corriger son déplacement, il peut reposer son stylet sur la table interactive pour déplacer cet avatar, ce qui a pour effet de déplacer immédiatement l'objet à la fin de sa trajectoire. De plus, l'utilisateur peut également corriger la position d'arrivée en effectuant un drag traditionnel (FIGURE 3.3).



FIGURE 3.3: Les différentes étapes de SuperFlick. L'utilisateur lance d'abord l'objet puis peut effectuer un drag d'ajustement depuis la position d'arrivée.

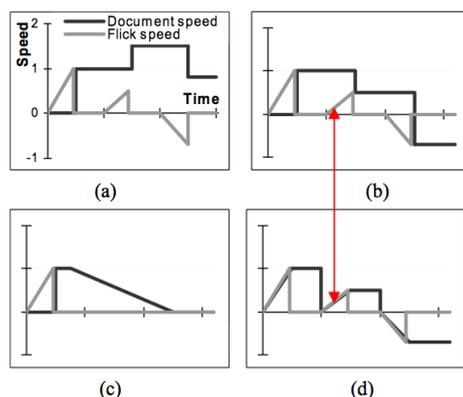


FIGURE 3.4: Les différents mapping de Multi-Flick entre la vitesse du Flick et la vitesse de défilement du document. a) La vitesse du flick s'additionne ou se soustrait à la vitesse courante. b) La vitesse d'un Flick définit la vitesse de défilement quelle que soit la vitesse courante. Le défilement continue jusqu'à ce que le geste soit terminé. c) Une friction est appliquée à la vitesse de défilement. d) Comme pour b) mais le début du geste de Flick arrête le défilement.

Multi-Flick introduit un espace de conception et évalue des techniques de défilement de listes à l'aide de techniques qui s'appuient sur des séries de Flicks consécutifs [Aliakseyeu et al., 2008]. Par exemple, une des techniques proposées additionne la vitesse (FIGURE 3.4 a)) de chaque nouveau geste de Flick effectué par l'utilisateur à la vitesse de défilement courante de la liste. Si cette vitesse est négative (c'est-à-dire inverse au sens de défilement de la liste) la vitesse de défilement du document est d'autant ralentie. Il n'y a pas de défilement pendant le geste de Flick mais seulement une fois le geste fini avec cette

technique. Par ailleurs, à l’instar de Reetz *et al.*, il est intéressant de remarquer que la majorité des travaux existant s’est intéressée au défilement mais pas ou peu aux techniques qui pourraient aider à contrôler la progression et l’arrêt du défilement.

Anatomie du Flick

La description de cette interaction se réduit souvent à considérer la variation de la vitesse de déplacement. Nous en donnons ici une description plus complète en présentant les différentes étapes qui composent le geste du Flick.

Amorce. Le Flick commence toujours par un déplacement : l’utilisateur déplace son doigt sur l’écran avant de le relâcher, avec une certaine vitesse. Cette étape est souvent simplement considérée comme un *pan* (déplacement direct du document) et parfois même ignorée (par exemple dans MultiFlick [Aliakseyeu et al., 2008]). Pourtant, la forme, la longueur, la direction, etc. de ce mouvement peuvent être prises en considération dans l’analyse du Flick. Nous présenterons plusieurs directions possibles concernant l’utilisation de ces paramètres dans la section consacrée à LongFlick.



FIGURE 3.5: Amorce du Flick.

Relâchement. Cette étape est explicitement prise en compte dans les travaux antérieurs. La vitesse de déplacement est mesurée à ce moment afin de permettre le calcul de la vitesse du défilement automatique qui s’ensuivra. Le calcul de cette vitesse diffère selon les techniques : sont utilisés par exemple, la moyenne de la vitesse durant l’intégralité ou sur les derniers points du geste, ou encore le rapport entre deux distances d’un geste d’aller-retour [Geissler, 1998].



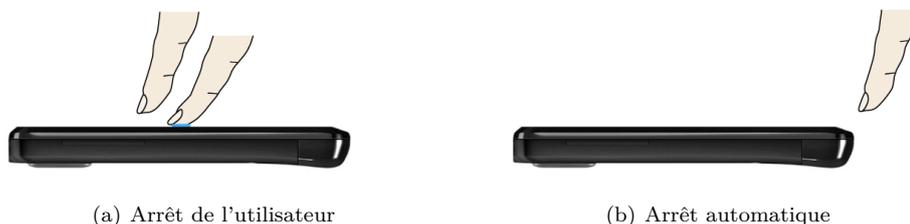
FIGURE 3.6: Relâchement du Flick.

Défilement automatique. A la suite d'un geste de Flick, le défilement se poursuit en évoluant au cours du temps sans intervention de l'utilisateur. Il peut être constant et ne s'arrêter que si l'utilisateur le demande explicitement ; ou encore, comme c'est généralement le cas dans les produits commerciaux actuels (iPhone, HTC, Samsung, etc), varier au cours du temps en ralentissant jusqu'à atteindre une vitesse nulle, ceci simulant l'effet d'une friction au cours du défilement. Le coefficient de friction peut différer selon les systèmes et les applications comme nous le verrons plus bas.



FIGURE 3.7: Défilement automatique du Flick.

Arrêt. L'arrêt du défilement est soit automatique (comme vu précédemment), soit provoqué par l'utilisateur lorsqu'il pose son doigt sur l'écran. La plupart des travaux précédents exploite l'appui sur l'écran comme une fonction binaire : le défilement s'arrête complètement dès que l'utilisateur touche l'écran.



(a) Arrêt de l'utilisateur

(b) Arrêt automatique

FIGURE 3.8: Différents types d'arrêts du Flick

Améliorations

Nous nous sommes plus particulièrement intéressés aux possibles améliorations des étapes les moins étudiées dans la littérature : l'amorce, le défilement automatique et l'arrêt. Nous donnons ici un bref résumé des trois techniques que nous avons proposées avant de les présenter plus précisément dans les sections suivantes. On trouvera en Annexe 5.3.3 le schéma présentant le module Flick Animator que nous avons réalisé pour ces techniques.



Flick & Brake. Cette première technique concerne les deux dernières étapes du Flick. Elle vise en effet à améliorer le défilement automatique et l'arrêt du Flick en utilisant la pseudo-pression de l'écran tactile. Cette technique propose deux modes d'interaction qui permettent à l'utilisateur

de contrôler le défilement automatique en fonction de la pression qu'il exerce sur l'écran. De plus, cette technique complète la métaphore qui a fait le succès du Flick (le lancer avec friction) en prenant en compte la "friction" exercée par le doigt en contact avec le dispositif. Pour permettre un défilement suffisant le Flick nécessite que l'utilisateur entretienne le mouvement en répétant le geste mais ne requiert pas d'action explicite pour l'arrêter. Inversement, Flick-and-Brake ne nécessite pas d'entretien du mouvement par l'utilisateur mais requiert une action pour stopper le défilement. Cette technique a fait l'objet d'un article à CHI EA 2011.



Semantic Flicking. Cette seconde technique porte sur le contrôle de la vitesse et trajectoire de défilement automatique d'un document en fonction de la sémantique des parties qui le composent. Le Semantic Flicking, utilise la pertinence du contenu pour moduler la vitesse et la trajectoire lors du défilement. Cette technique a fait l'objet d'une collaboration avec les chercheurs d'Alcatel Lucent Bell Labs pour définir de manière automatique les passages pertinents dans des documents textuels. Ce travail a donné lieu à un dépôt de brevet et a fait l'objet d'une démonstration aux Open Days 2011 d'Alcatel Lucent.



Long Flick. Cette dernière technique a pour objet l'amorce du mouvement de Flick. Nous présenterons les propriétés de cette étape et expliquerons comment on peut les exploiter pour moduler le défilement et la distance parcourue. Cette technique permet d'atteindre facilement une partie donnée d'une liste en ne réalisant qu'un seul geste. Elle permet également de parcourir à vitesse constante une distance définie par l'utilisateur.

3.2 Flick-and-Brake

3.2.1 Contexte

Comme indiqué dans l'introduction, le succès du Flick vient sans doute du fait qu'il repose sur une métaphore simple. Les implémentations actuelles simulent de l'inertie et de la dissipation d'énergie via un coefficient de friction qui permet à la liste de s'arrêter d'elle-même, sans action de l'utilisateur. Ce dernier peut néanmoins interrompre le défilement de son propre chef en touchant la liste pendant le défilement.

L'utilisateur dispose cependant dans le monde réel de davantage de contrôle sur le défilement que dans la métaphore usuellement employée. Par exemple, lorsqu'un utilisateur manipule un globe terrestre (FIGURE 3.9) il peut non seulement amorcer un mouvement de rotation mais également ralentir la rotation en appliquant une pression avec son doigt sur la surface du globe. La métaphore actuellement implémentée dans les techniques de Flick est donc incomplète. Dans ce chapitre, nous proposons de la compléter en permettant à l'utilisateur de contrôler le coefficient de friction appliqué au défilement en exploitant le niveau de pression sur la surface interactive.

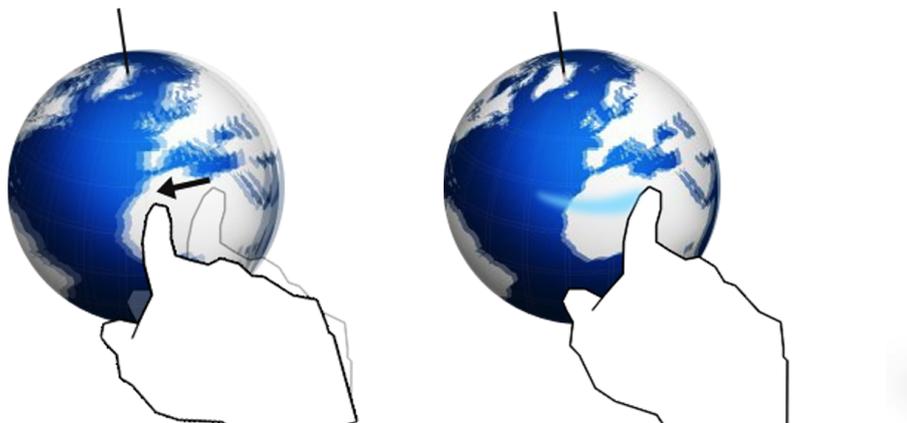


FIGURE 3.9: Globe terrestre

Après un aperçu de l'état de l'art, nous présentons le concept de Flick-and-Brake et deux implémentations alternatives de ce principe. Nous présentons ensuite un automate fini qui introduit la pression du doigt comme un paramètre continu permettant de contrôler le coefficient de friction appliqué au défilement. Cet automate permet de décrire nos techniques ainsi que les techniques de Flick habituelles. Ensuite nous présentons différents feedbacks étudiés. Enfin, nous décrivons les résultats d'une étude qui a été conduite pour comparer nos techniques au Flick standard.

Bien que les études de ce chapitre aient été menées sur dispositif mobile, il est intéressant

de signaler que le concept de Flick-and-Brake peut en fait être étendu à n'importe quelle surface interactive capable d'identifier une valeur de pression. Initialement développée pour un HTC Hero (smartphone ayant servi pour l'expérimentation), cette technique fonctionne également sur tous les téléphones sous Android détectant la pseudo-pression (notamment sur le HTC Desire, Nexus One). Ce travail a été réalisé en collaboration avec Sylvain Malacria alors doctorant à Télécom ParisTech.

État de l'art. Quelques techniques d'interaction exploitant la pression ont déjà été proposées dans des études antérieures. Par exemple, différents niveaux de pression sont utilisés dans PressureText [McCallum et al., 2009] comme alternative au multi-tap pour saisir du texte sur un clavier physique de dispositif mobile à 12 touches. Chaque touche est augmentée d'un capteur de pression et associe un caractère à un intervalle de pression donné. Brewster *et al.* [Brewster and Hughes, 2009] proposent également d'utiliser la pression pour simplifier la saisie de majuscules sur un clavier logiciel de dispositif mobile : un léger contact (respectivement fort) avec une touche insère une lettre minuscule (majuscule) dans le texte. Les PressureWidgets [Ramos and Balakrishnan, 2007] sont différents types de composant graphique qui exploitent la pression de la mine d'un stylet. Notamment, le Pressure Marking Menu augmente le nombre d'éléments disponibles en largeur dans un Marking menu. Il associe pour cela trois éléments différents à un même tracé en fonction de la différence de pression entre le premier point et le dernier point du tracé. GraspZoom [Miyaki and Rekimoto, 2009] propose d'utiliser la pression pour permettre à l'utilisateur de contrôler la vitesse de défilement sur une liste : la cible commence à défiler vers le haut (respectivement vers le bas) dès qu'un seuil de pression est dépassé sur la partie haute (respectivement basse) de l'écran. Enfin, Zoofing! [Quinn and Cockburn, 2009] utilise la pression d'un stylet pour zoomer sur une liste jusqu'à atteindre un niveau de zoom satisfaisant. L'utilisateur peut ensuite naviguer dans la liste avec du défilement conventionnel (et notamment du Flick). L'affichage revient au niveau de zoom minimal après 3 secondes d'inactivité de l'utilisateur.

3.2.2 Concept général

Nous avons vu en introduction que les techniques de Flick habituelles appliquent un coefficient de friction qui ralentit la vitesse de défilement jusqu'à l'arrêter. La décélération est dans ce cas uniquement contrôlée par le système et l'utilisateur ne peut pas la contrôler autrement qu'en touchant l'écran pour arrêter le défilement. De plus, cette décélération contrôlée par le système arrête automatiquement le défilement au bout d'un certain temps. Par conséquent, l'utilisateur doit effectuer plusieurs gestes de lancer et donc interrompre puis reprendre à plusieurs reprises son contact avec l'écran (*clutching*) s'il veut parcourir une distance importante. Le concept Flick-and-Brake offre à l'utilisateur la possibilité de contrôler cette décélération en lui permettant de freiner (*brake*) en appuyant plus ou moins

fort sur l'écran (FIGURE 3.10). Comme dans le monde physique, la décélération est proportionnelle à la pression appliquée. Le défilement automatique est quant à lui entretenu par le système : sans action explicite de l'utilisateur, le défilement ne s'arrête que lorsqu'une des extrémités de la liste est atteinte. Nous présentons dans cette section deux variantes de Flick-and-Brake qui reposent sur deux métaphores différentes.

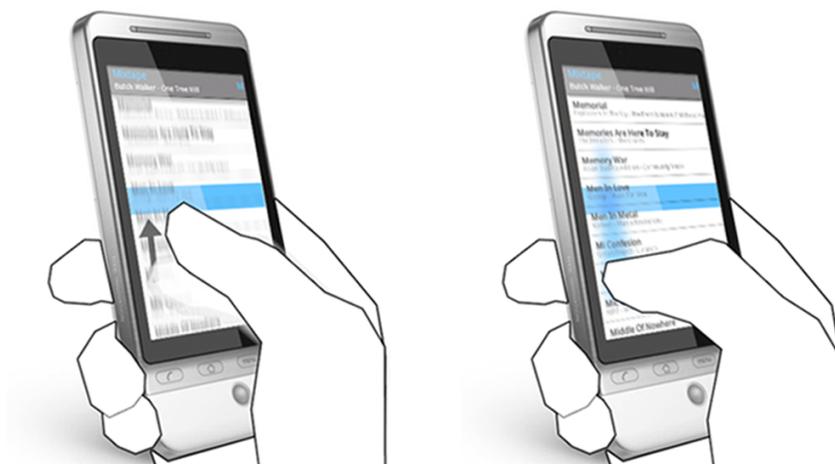


FIGURE 3.10: Le concept général de Flick-and-Brake : l'utilisateur lance le défilement avec un Flick puis en contrôle le freinage en modulant la pression de son doigt sur l'écran.

Métaphore du mouvement inertiel

Cette variante repose sur la métaphore d'une roue sans frottement dont le mouvement une fois activé continuerait indéfiniment sans une action explicite de l'utilisateur pour l'interrompre. L'utilisateur doit alors freiner pour contrôler la vitesse en pressant son doigt plus ou moins fort sur l'écran. Si l'utilisateur pose légèrement son doigt sur la surface et maintient un niveau de pression constant, la vitesse de défilement décroît progressivement selon un coefficient de friction constant jusqu'à devenir nulle et que le défilement s'arrête. Plus l'utilisateur appuie fort plus le coefficient de friction est élevé. Il s'agit donc d'un contrôle d'ordre 2 puisque la pression du doigt contrôle le coefficient de décélération appliqué au système. De fait, une faible pression appliquée suffisamment longtemps aura pour effet d'arrêter totalement le défilement (FIGURE 3.11). Si l'utilisateur réalise un drag ou appuie suffisamment fort (i.e dépasse un certain seuil) le défilement est interrompu. S'il le souhaite l'utilisateur peut aussi réduire la vitesse jusqu'à la valeur voulue et laisser ensuite défiler sans toucher l'écran. Cette technique permet donc à l'utilisateur de réduire facilement la vitesse. Cependant elle ne permet pas de l'augmenter (à moins de recourir à un nouveau geste de Flick), ce qui constitue sa principale limitation.

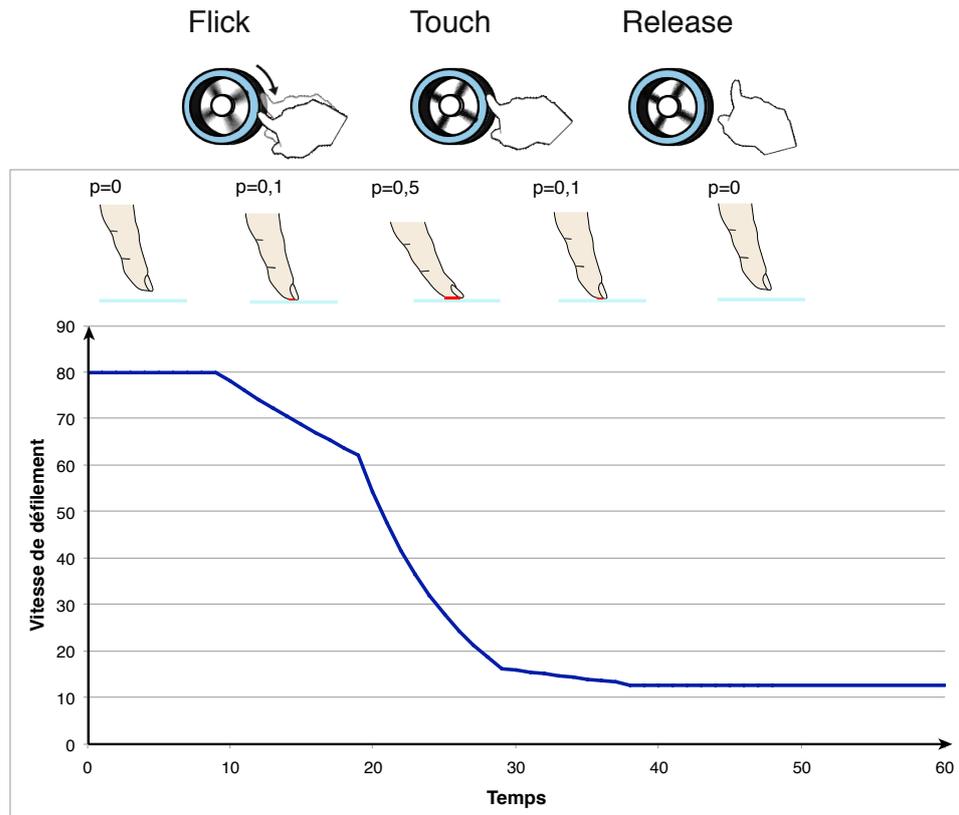


FIGURE 3.11: Comportement de la technique Flick-and-Brake avec la métaphore du mouvement inertiel.

Métaphore du mouvement entretenu

La seconde variante repose également sur la métaphore d'une roue sans frottement mais cette fois entretenue par un moteur qui s'oppose à la friction du doigt (FIGURE 3.12). Comme précédemment la pression du doigt permet de faire diminuer la vitesse de défilement, mais il s'agit ici d'un contrôle d'ordre 1 : tant que le doigt est en contact avec la roue, la vitesse de défilement est inversement proportionnelle à la pression. Relâcher le doigt ramène la vitesse de défilement à la valeur initiale amorcée par le geste de Flick. Cependant, afin que l'utilisateur puisse arrêter le défilement simplement, la vitesse de défilement devient nulle (et ne reprend donc pas sa valeur initiale) si l'utilisateur appuie suffisamment fort ou réalise un drag. Cette métaphore, proche de celle d'une voiture avec frein et accélérateur, est utilisée très naturellement par les utilisateurs, comme nous le verrons par la suite dans l'expérience. Elle a en effet le précieux avantage de permettre à l'utilisateur de ralentir mais aussi d'accélérer le défilement. En contrepartie, contrairement à la méthode précédente, elle oblige l'utilisateur à garder son doigt sur l'écran s'il souhaite maintenir le défilement à une vitesse inférieure à la vitesse initiale.

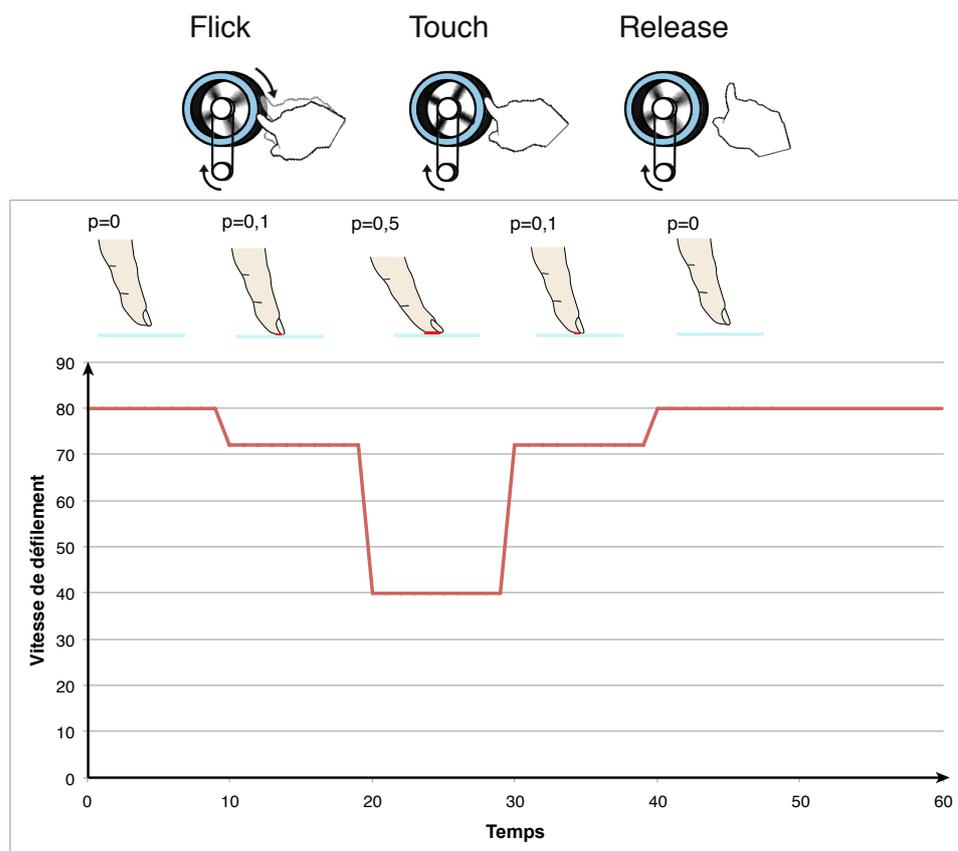


FIGURE 3.12: Métaphore du mouvement entretenu.

Automate fini générique

Flick-and-Brake offre une nouvelle possibilité de contrôle à l'utilisateur en utilisant la pression comme un paramètre continu. De fait, les techniques reposant sur ce concept peuvent être vues comme une extension des techniques de Flick standard. Nous proposons un automate fini (FIGURE 3.13) permettant de décrire l'ensemble de ces techniques.

Cet automate introduit la pression du doigt comme paramètre. Le modèle se compose de quatre états : *Arrêté*, *Touché*, *Défilement automatique* et *Défilement manuel*. Les deux instances décrites ci-dessus ainsi que les techniques de Flick standard peuvent être décrites à l'aide de ces quatre états, avec des conditions d'arrêt différentes. Le Flick standard exploite la pression de manière binaire et interrompt immédiatement le défilement quand le doigt entre en contact avec l'écran. À l'inverse, les techniques Flick-and-Brake considèrent la pression comme un paramètre continu que contrôle l'utilisateur (variable comprise entre 0 et 1).

Le Flick standard (FS) se décrit de la manière suivante : au départ (état *Arrêté*), la

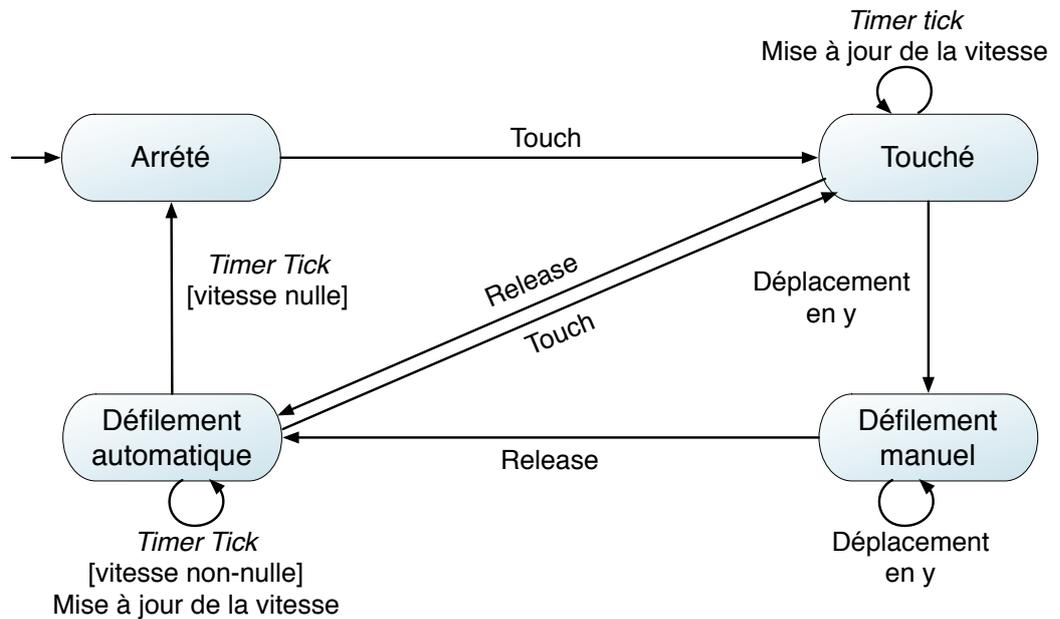


FIGURE 3.13: Automate fini générique permettant de décrire les techniques Flick-and-Brake et les techniques de Flick habituelles

vitesse de défilement est nulle. L'utilisateur touche l'écran (état *Touché*) pour commencer à faire défiler la liste en déplaçant son doigt sur la surface (état *Défilement manuel*) puis soulève son doigt de la surface de l'écran (état *Défilement automatique*). La liste continue à défiler à la vitesse du mouvement correspondant à la vitesse du tracé lorsque le doigt a quitté l'écran. Un coefficient de friction appliqué par le système fait diminuer progressivement la vitesse de défilement jusqu'à zéro (retour à l'état *Arrêté*).

La fonction de mise à jour à l'état *Défilement automatique* modifie la vitesse quand la liste défile et que le doigt n'est pas en contact avec l'écran. Dans la plupart des implémentations actuelles, cette fonction multiplie la vitesse courante par un coefficient k , $k \in [0,1]$ ³. Avec une valeur de $k = 1$, la vitesse est constante et ne diminue pas tandis qu'avec une valeur de $k = 0$ le défilement s'arrête immédiatement (cas du Pan sans inertie). Les valeurs intermédiaires de k permettent de simuler un coefficient de friction plus ou moins important. Le système retourne dans l'état *Arrêté* si la vitesse devient nulle, et dans l'état *Touché* si l'utilisateur touche l'écran.

La fonction de mise à jour à l'état *Touché* modifie la vitesse quand l'utilisateur touche l'écran avec le doigt sans le déplacer sur sa surface. Dans le cas du Flick standard, la pression appliquée par le doigt sur l'écran est considérée de manière binaire et annule immédiatement la vitesse de défilement dès la première itération de la mise à jour. Avec les techniques

3. Selon la norme ISO 31-11 (http://en.wikipedia.org/wiki/ISO_31-11) : $[0,1]$ représente l'intervalle fermé dans \mathbb{R} entre 0 et 1 inclus. $\{0,1\}$ représente l'ensemble constitué des deux éléments 0 et 1.

Flick-and-Brake (F&B), la pression est exploitée de manière continue pour permettre à l'utilisateur de contrôler la décélération : plus l'utilisateur appuie fort, plus la vitesse décroît. La vitesse est mise à jour à chaque itération. Ainsi si V_n est la vitesse à l'itération n et p la pression :

$$\begin{aligned} V_n &= V_{n-1} * (1 - p), p \in \{0,1\} && \text{Flick Standard} \\ V_n &= V_{n-1} * (1 - p), p \in [0,1] && \text{F\&B inertiel} \\ V_n &= V_0 * (1 - p), p \in [0,1] && \text{F\&B entretenu} \end{aligned}$$

Utilisation de la pseudo-pression

Les techniques Flick-and-Brake reposent sur l'utilisation d'un paramètre de pression mais les smartphones actuels ne disposent pas d'un tel capteur. Certains d'entre-eux renvoient cependant une valeur de pseudo-pression (que nous notons pp) comprise entre 0 et 1 et proportionnelle à la surface de contact entre le doigt de l'utilisateur et l'écran du smartphone.

Des études préliminaires [Essl et al., 2010] suggèrent que cette valeur de pseudo-pression est suffisamment fiable pour que l'utilisateur puisse contrôler la valeur en modulant la surface en contact avec l'écran. La surface de contact dépendant de la taille et de la forme du doigt, elle varie donc en fonction de l'utilisateur. Nos techniques d'interactions effectuent une étape de calibration en soustrayant la valeur initiale de pseudo-pression pp_0 (obtenue quand le doigt de l'utilisateur entre en contact avec l'écran) à la valeur de pseudo pression courante pp_i . Une étude préliminaire nous a confirmé que cette valeur varie de manière non linéaire, ce qui n'est pas surprenant puisque la valeur de pp dépend de la *surface* de contact. Nous utilisons donc une fonction quadratique pour estimer p à partir de pp . Au final, nous obtenons donc la valeur de p comme suit : $p = (pp_i - pp_0)^2$

Notons que l'utilisation de Flick-and-Brake serait bien sûr possible (et probablement avec un niveau de contrôle plus fin) avec un véritable capteur de pression.

Feedback visuel

Plusieurs travaux concernant la pression [Shi et al., 2008, McCallum et al., 2009, Cechanowicz et al., 2007], montrent que le contrôle est plus facile (et plus précis) à l'appui qu'au relâchement, ceci étant lié aux capteurs de pression, qui ne sont généralement pas mécaniques. Dans le cas de la pseudo-pression, et donc de notre propre système, nous retrouvons les mêmes limitations. Afin d'améliorer le contrôle de cette pression, comme le suggère [Ramos et al., 2004], nous avons rajouté un retour visuel à notre système.

Nous présentons tout d'abord les différents types de retour visuel envisagés puis celui

que nous avons retenu et implémenté en tenant compte des avis d'utilisateurs questionnés de manière informelle avant de développer le dispositif d'expérimentation. Dans tous les cas le retour visuel n'apparaît qu'au contact du doigt sur l'écran lors du freinage et il est semi-transparent afin de ne pas cacher complètement les items de la liste manipulée.



(a) Feedback en forme de cercle avec une pression moyenne

(b) Feedback elliptique avec une grande vitesse et une pression moyenne

(c) Feedback elliptique avec une faible vitesse et une forte pression



(d) Feedback avec trainée avec une faible vitesse, une forte pression et un défilement vers le bas de la liste

(e) Feedback avec trainée avec une forte vitesse, une forte pression et un défilement vers le bas de la liste

FIGURE 3.14: Différents types de retour visuel pour Flick-and-brake

Nous avons proposé aux utilisateurs trois types de feedback, que nous présentons ici du plus simple (celui véhiculant le moins d'information) au plus complet :

- Le défilement de la liste donnant déjà une indication sur la vitesse et le sens de défilement, le premier feedback affiche seulement un cercle dont le diamètre représente la pression exercée par le doigt. Le cercle est situé sous le doigt de l'utilisateur (FIGURE 3.14 (a)).
- Le deuxième feedback consiste en une ellipse, qui est également située sous le doigt. Cette ellipse possède deux rayons que nous noterons R_x pour le rayon selon l'axe x de l'écran et R_y selon l'axe y . La taille de R_x varie en fonction de la pression : plus la pression augmente plus R_x est grand (FIGURE 3.14 (c)). Similairement R_y varie selon la vitesse de telle sorte que plus elle augmente plus le rayon R_y est grand (FIGURE 3.14 (b)).
- Le dernier feedback tente de retranscrire tous les paramètres de défilement à la fois : vitesse, sens et pression. Il représente une trainée dont la longueur détermine la vitesse, l'épaisseur, la pression et le sens de défilement (FIGURE 3.14 (d), FIGURE 3.14 (e)). De plus ce feedback utilise la métaphore de la 'trace' de freinage que laisserait le doigt sur l'écran, à la manière de l'effet produit par un pneu sur une route.

Le dernier type de feedback (la trainée) a finalement été choisi. Bien qu'ayant l'inconvénient d'être occulté par le doigt lors du défilement vers le haut de la liste, il a été plus apprécié par les utilisateurs lors des tests préliminaires. Il comporte l'avantage d'apporter plusieurs informations à la fois et de ne pas empêcher la lecture de la liste pendant le freinage. De plus, de par son lien avec des éléments du réel bien connus (i.e. l'aspect métaphorique de la trace), ce feedback renforce la métaphore du freinage qui est au centre de Flick-and-Brake. Ceci permet sans doute aux utilisateurs de mieux s'approprier la technique et de comprendre naturellement qu'ils sont en train de freiner le défilement.

3.2.3 Évaluation

Nous avons procédé à une expérience dont le but était d'évaluer l'efficacité d'une décélération contrôlée par l'utilisateur, relativement à une décélération contrôlée par le système, dans une tâche où l'utilisateur devait atteindre une chanson dans une liste ordonnée alphabétiquement. Nous avons délibérément choisi une tâche courante de la vie réelle, à savoir de choisir successivement plusieurs chansons pour composer une liste de lecture sur un dispositif mobile. Nous avons mesuré la vitesse et la précision ainsi que le nombre de gestes de Flicks effectués ainsi que les préférences subjectives des participants.

Méthode

Matériel. Nous avons utilisé un smartphone HTC Hero sous le système d'exploitation Android 2.1. Ce dispositif est équipé d'un écran capacitif de 3,5 pouces de diagonale et d'un affichage de résolution 320 x 480 pixels. Il permet aussi de détecter la pseudo-pression.

Tâche. Chaque élément de la liste correspond à une chanson et occupe un rectangle de 50 pixels de haut, ce qui permet d'afficher 8,5 chansons par écran. Le nom de la cible est affiché dans une zone fixe située en haut de l'écran. Une sélection est validée quand le système est dans l'état *Arrêté* et qu'au moins la moitié de l'élément cible est située à l'intérieur d'un rectangle fixe de 50 pixels de haut placé au milieu de l'écran (FIGURE 3.15). Le stimulus en haut de l'écran est alors modifié et l'utilisateur peut parcourir la liste pour atteindre la nouvelle cible, qui se situe à une distance relative de 25, 75, 150 ou 250 éléments, dans une des deux directions (haut ou bas). Le stimulus ne changeant qu'une fois la cible sélectionnée, il s'agit d'une tâche sans erreur possible, le temps de réalisation donnant ainsi une mesure globale de performance tenant compte à la fois de la vitesse et de la précision.

Les participants devaient tenir le dispositif mobile avec une seule main et utiliser leur pouce pour interagir. La distance de la nouvelle cible et la direction ont été tirées au hasard sans remise dans l'urne, chaque combinaison étant présente exactement une fois par série (8 cibles par série). Chaque série utilisait une liste de chansons différente pour éviter les effets d'apprentissage.

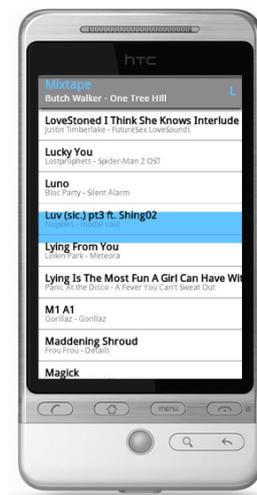


FIGURE 3.15: Dispositif et interface d'expérimentation.

Techniques. Nous avons comparé trois techniques : le Flick standard (SF) et les techniques de Flick-and-Brake (F&B) inertiel et entretenu. La vitesse de défilement dans l'état *Défilement automatique* était limitée pour chaque technique à 2100 pixels par seconde comme suggéré dans [Andersen, 2005]. L'ordre de présentation des techniques était balancé suivant un carré latin.

Procédure. Douze participants adultes (4 femmes et 8 hommes) ont suivi 2 séries d'entraînements avant de procéder à 4 séries d'expérimentations. Chaque participant a donc sélectionné 96 cibles : 3 techniques * 4 distances * 2 directions * 4 series (*within-subject design*).

Après l'expérience, les participants étaient invités à trier les techniques par ordre de préférence et à les évaluer selon quatre critères (rapidité, précision, agrément et simplicité) sur une échelle de Likert à 5 niveaux.

Résultats

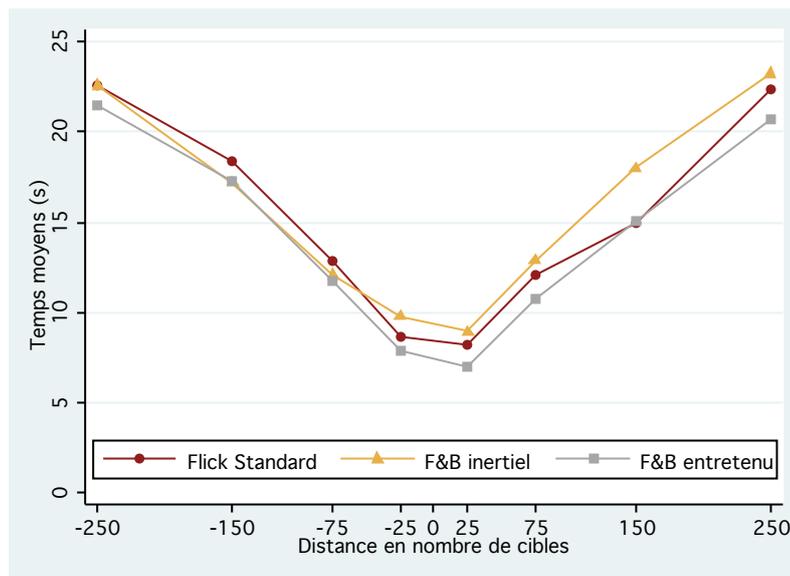


FIGURE 3.16: Temps moyens de sélection par technique et par distance cible.

Temps de sélection de cible. La FIGURE 3.16 montre les temps moyens de sélection par technique et par distance de cible. Une analyse de variance sur les temps de sélection a montré que la technique Flick-and-Brake entretenu était significativement plus rapide que les deux autres techniques : Flick standard (t -test de Student : $t_{11} = 1.84, p = 0.046$) et

Flick-and-Brake inertiel ($t_{11} = 2.99, p = 0.006$). L'effet est cependant relativement petit ($F_{2,22} = 4,70, p < 0,02$, TABLEAU 3.1), ceci probablement en raison de la borne maximale sur la vitesse de défilement automatique (de 2100 pixels par seconde). En revanche Flick-and-Brake inertiel n'est pas significativement plus rapide ou plus lent que le Flick standard ($t_{11} = 1.18, p = 0.131$).

	Temps Moyens
Flick standard	14,99s
Flick-and-Brake inertiel	15,57s
Flick-and-Brake entretenu	13,99s

TABLEAU 3.1: Temps moyens en seconde par technique.

Nombre de gestes de Flick par sélection de cible. Les participants ont effectué beaucoup moins de gestes de Flick avec les deux techniques Flick-and-Brake qu'avec le Flick standard. Le nombre moyen de Flicks par sélection est représenté sur la FIGURE 3.17. Une différence significative est observée sur les techniques ($F_{2,22} > 72.61, p < 0,001$) avec un effet important. Des tests post-hoc ont confirmé que les deux techniques Flick-and-Brake nécessitaient moins de gestes de Flick que le Flick standard pour toutes les distances sauf la moins grande.

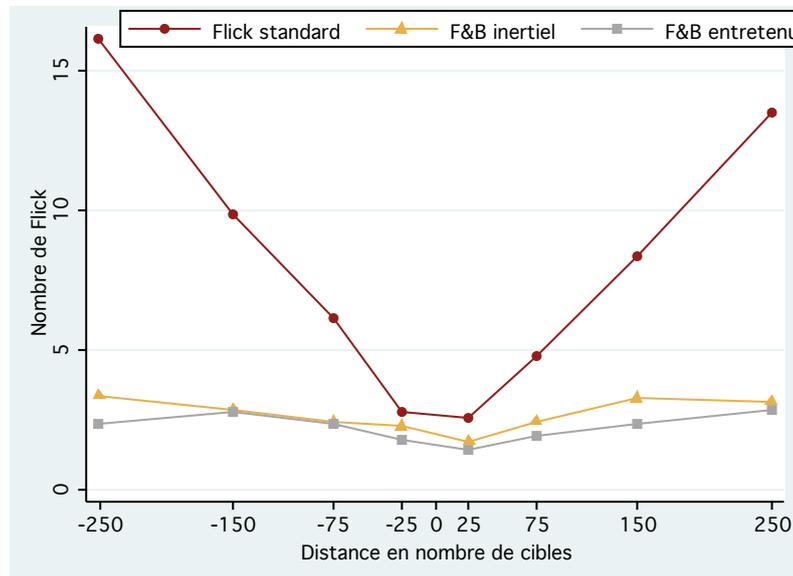


FIGURE 3.17: Nombre de Flicks par sélection de cible.

En effet, comme escompté et comme le montre la FIGURE 3.17 plus la distance augmente plus le nombre de gestes augmente pour le Flick standard alors que pour les deux techniques

Flick-and-Brake le nombre de gestes varie très peu. Il y a donc une forte corrélation entre les techniques et la distance des cibles ($F_{3,33} = 54,73, p < 0,001$). Comme on peut le voir sur la FIGURE 3.17 nous avons obtenu des relations quasi-linéaires qui, après calcul de la régression linéaire séparément pour la courbe positive et négative de chaque technique, donnent les coefficients présentés dans le TABLEAU 3.2. Ces résultats montrent que le nombre de gestes du Flick standard augmente 8 à 29 fois plus vite que le nombre de gestes des techniques de Flick-and-Brake en fonction de l'augmentation de la distance de la cible à atteindre.

	Positive	Negative
Flick standard	0,049	0,058
Flick-and-Brake inertiel	0,006	0,005
Flick-and-Brake entretenu	0,006	0,002

TABLEAU 3.2: Coefficient de la régression linéaire par technique et par direction

Préférences subjectives. Les participants ont eu la nette sensation d'aller plus vite en utilisant les techniques Flick-and-Brake entretenu (Test de Wilcoxon, $p < 0,0003$) et inertiel ($p < 0,0005$) qu'avec le Flick Standard (FIGURE 3.18).

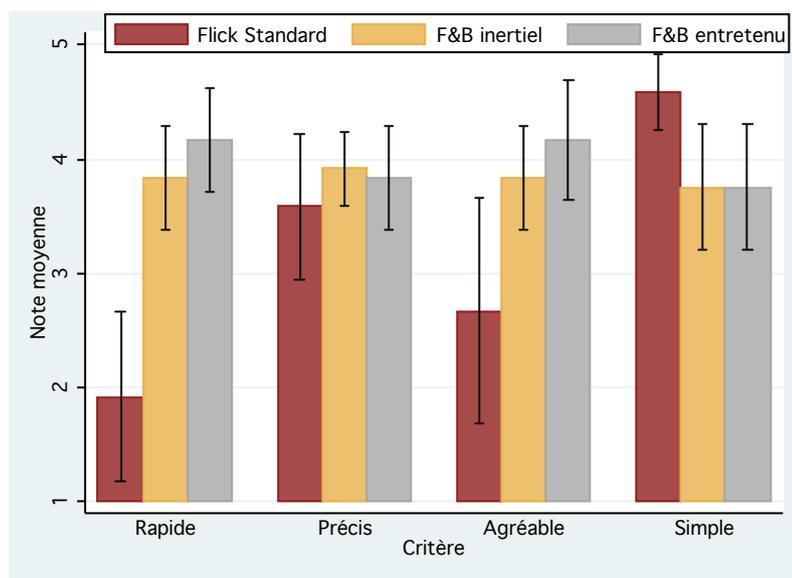


FIGURE 3.18: Préférences subjectives des participants

De plus les techniques Flick-and-Brake ont été jugées plus agréables que le Flick standard ($Kruskal - Wallis \chi_2 = 6,983, p < 0,0305$), avec une préférence particulièrement marquée pour le Flick-and-Brake entretenu face au Flick standard ($p < 0,0176$). Les entretiens post-expérience nous conduisent à penser que cette différence est essentiellement due au nombre

de gestes de Flick effectués, lequel croissait rapidement pour les grandes distances avec le Flick Standard.

Les participants ont particulièrement apprécié de pouvoir contrôler eux-même la décélération du défilement, une possibilité offerte par les techniques Flick-and-Brake. Ils ont également mentionné le côté attractif du retour visuel proposé. Comme le montre le [TABLEAU 3.3](#), les deux variantes de Flick-and-Brake ont été appréciées et sont très majoritairement classées aux deux premières positions.

	1 st	2 nd	3 rd	Moyenne
Flick standard	0	1	11	2,92
Flick-and-Brake inertiel	4	7	1	1,75
Flick-and-Brake entretenu	8	4	0	1,33

[TABLEAU 3.3](#): Classement des techniques préférées par les participants

3.2.4 Conclusion

Nous avons présenté Flick-and-Brake, une approche permettant à l'utilisateur de contrôler à l'aide de la pression le coefficient de friction appliqué à un défilement de liste consécutif à un geste de Flick. Nous avons plus particulièrement décrit deux variantes de cette approche :

- le Flick-and-Brake inertiel qui repose sur la métaphore d'une roue sans frottement dont le mouvement une fois activé continuerait indéfiniment sans action explicite de l'utilisateur pour l'interrompre. L'utilisateur peut alors réduire la vitesse de défilement en appuyant sur l'écran interactif plus ou moins fort.
- le Flick-and-Brake entretenu qui repose également sur la métaphore d'une roue sans frottement mais cette fois entretenue par un moteur qui s'oppose à la friction du doigt. En appuyant sur l'écran, l'utilisateur réduit la vitesse de défilement qui reprend sa vitesse initiale si l'utilisateur relâche le doigt (sauf dans le cas particulier où la vitesse devient nulle).

Nous avons également présenté un automate fini générique capable de décrire ces deux techniques ainsi que les techniques de défilement inertiel habituelles.

Nous avons mené une expérience pour évaluer l'efficacité d'une décélération contrôlée par l'utilisateur, relativement à une décélération contrôlée par le système, dans une tâche où l'utilisateur devait atteindre une chanson dans une liste ordonnée alphabétiquement. Cette étude nous a montré que la technique Flick-and-Brake entretenu était significativement plus rapide que les deux autres techniques et que les participants ont effectué significativement moins de gestes de Flick avec les deux variantes de Flick-and-Brake. De plus, les nouvelles

techniques que nous proposons ont été fortement appréciées par les utilisateurs, probablement car elles leur offrent davantage de contrôle et permettent de parcourir de grandes distances en effectuant moins de gestes.

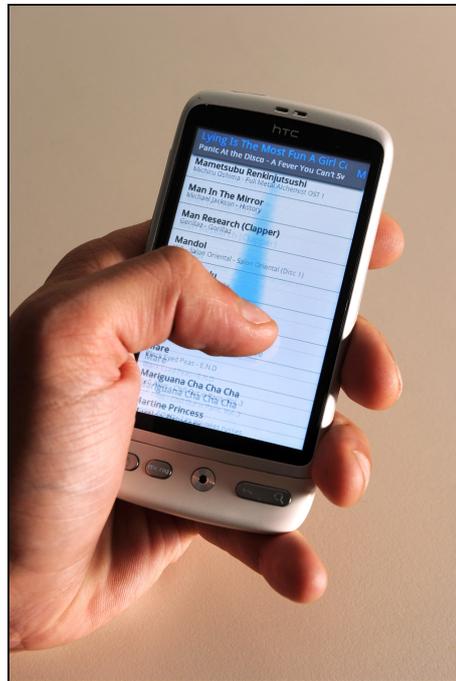


FIGURE 3.19: Flick and Brake

Les techniques Flick-and-Brake proposent ainsi une solution simple et intéressante pour augmenter le Flick et offrir plus de contrôle à l'utilisateur. Des techniques de Flick améliorées comme celles que nous proposons ici sont donc susceptibles de notablement faciliter la navigation dans des documents de longue taille.

Enfin, bien que nous ayons évalué ces techniques dans une tâche de défilement avec une liste à une dimension, leur concept est immédiatement généralisable à la navigation à deux dimensions. De plus en plus courant sur les téléphones mobiles, le Flick se répand aujourd'hui sur tous les types de surface interactive d'où l'intérêt de continuer à améliorer cette technique désormais incontournable.

3.3 Semantic Flicking

3.3.1 Contexte

Comme nous l'avons vu précédemment, le Flick permet de faire défiler un document mais n'offre aucune possibilité de contrôle en cours de défilement, raison pour laquelle nous avons proposé Flick-and-Brake pour donner à l'utilisateur un meilleur contrôle en lui permettant d'ajuster la vitesse de défilement après le relâchement. Dans cette partie nous ne nous intéressons plus aux actions de l'utilisateur mais à l'objet du défilement, c'est-à-dire le document sur lequel il s'opère.

Remarquons tout d'abord que la vitesse du défilement et de la décélération automatiques varie suivant les applications sur les dispositifs commerciaux. Par exemple sur l'iPhone la vitesse de défilement est plus importante dans la liste des contacts que lorsque l'on navigue sur une page web dans Safari. Ces différences sont cependant seulement liées à l'application et non au contenu. L'approche proposée par le Semantic Flicking consiste à prendre en compte le contenu du document que l'on fait défiler. L'idée d'avoir des comportements différents des déplacements en fonction du contexte s'inspire du concept de Semantic Pointing [Blanch et al., 2004] (FIGURE 3.20) où l'espace moteur et l'espace visuel diffèrent, le rapport entre la vitesse du curseur et celle de la souris dépendant de la nature de la cible que l'on veut atteindre. Ceci permettra par exemple de rendre le bouton *Sauvegarder* plus facile à atteindre que le bouton *Quitter* dans une boîte de dialogue. Le Semantic Flicking exploite un concept similaire : l'espace moteur est également différent de l'espace visuel et le même geste aura un effet différent sur le document selon la partie du document que l'on manipule.

3.3.2 Description

La technique se base sur l'idée que toutes les parties d'un document n'ont pas le même poids pour l'utilisateur. Par exemple, dans le cas d'une liste de contacts, les personnes que l'on appelle le plus souvent seront aussi celles le plus fréquemment recherchées par l'utilisateur. Certaines parties ou certains objets du document sont donc plus importants ou correspondent à des favoris. Ces parties et ces favoris peuvent avoir été explicitement définis par l'utilisateur ou automatiquement par le système. Dans les deux cas il est souhaitable qu'ils soient rendus plus facilement accessibles à l'utilisateur.

Une solution proposée dans [Sears and Shneiderman, 1994] (FIGURE 3.21) dans le cas des menus consiste à réorganiser les items en fonction de leur fréquence d'utilisation. Elle n'est cependant pas applicable dans le cas d'un texte car elle en modifierait l'ordre et la structure et sortirait ainsi l'objet d'intérêt de son contexte.



FIGURE 3.20: Semantic pointing. a) espace visuel inchangé, b) espace moteur.

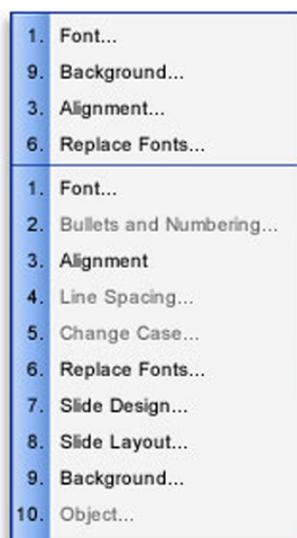


FIGURE 3.21: Splitmenu. Réorganisation des items d'un menu en fonction de leur fréquence d'utilisation.

Une autre approche consiste à surligner les items les plus fréquents. Cette approche est purement visuelle et ne vise pas à faciliter l'interaction. De plus elle modifie l'apparence du texte, ce qui peut être gênant pour la lecture. Elle peut également engendrer des conflits au niveau de la représentation, par exemple si le surlignage est également utilisé pour mettre en évidence le résultat des recherches textuelles. Enfin, elle n'est pas incompatible avec Semantic Flicking avec lequel elle peut être utilisée conjointement si on le désire pour renforcer l'effet de ce dernier.

Contrairement aux deux techniques précédentes, le Semantic Flicking permet de mettre en évidence les éléments importants d'un document sans modifier leur représentation ni leur structure et sans gêner la lecture. Le Semantic Flicking permet d'adapter la vitesse et le parcours de défilement au contenu lors de la phase de défilement automatique : celui-ci est ainsi modifié lorsque des éléments importants sont présentés à l'utilisateur. Le comportement antérieur est ensuite rétabli après le passage important. Les favoris et autres objets d'intérêt se comportent ainsi comme des centres d'attractions.

Dans la section suivante nous présentons deux applications du Semantic Flicking. La première, appelée S-Book, exploite cette technique pour la lecture de livres électroniques. Cette technique a fait l'objet d'un dépôt de brevet et d'une démonstration aux OpenDays d'Alcatel Lucent 2011. La seconde étend le concept de Semantic Flicking à un espace de défilement à deux dimensions.

3.3.3 Prototypes

S-Book

Description. En collaboration avec les chercheurs Myriam Ribière et Julien Robinson d'Alcatel Lucent Bell Labs, nous avons réalisé un prototype fonctionnel du Semantic Flicking adapté à la lecture collaborative de livres. Bell Labs travaille en effet sur une application, le S-Book, dédiée à la lecture et l'annotation des livres numériques. Les annotations sont faites de manière collaborative et tous les utilisateurs inscrits ayant accès au livre peuvent également obtenir les annotations laissées par les autres lecteurs et en ajouter de nouvelles.

L'application permet de générer des *heatmaps* en fonction des passages qui sont le plus annotés par les utilisateurs. L'application tire ainsi avantage du "user-generated content". En d'autres termes, ce sont les utilisateurs qui génèrent le contenu intéressant pour la lecture : les annotations. Le nombre d'annotations concernant une zone du document (e.g. des paragraphes, des mots) pondère alors la chaleur de cette zone. Les *heatmaps* peuvent également être générées en fonction d'une recherche textuelle de l'utilisateur. Dans ce cas, la requête de l'utilisateur couplée aux techniques fournies par le web sémantique permettent de calculer la similarité des annotations vis-à-vis de cette requête. Les zones chaudes sont alors celles dont l'indice de similarité des annotations est le plus élevé. Les zones chaudes étant dans tous les cas celles qui sont les plus intéressantes pour l'utilisateur, elles permettent de définir une *heatmap* (ou carte de chaleur) globale du document servant à configurer automatiquement le Semantic Flicking.

Le Semantic Flicking facilite la navigation de l'utilisateur en utilisant l'information apportée par ces "zones chaudes". A vitesse constante dans l'espace moteur, le défilement est

plus ou moins ralenti dans l'espace visuel suivant l'intensité de la chaleur de la zone affichée à l'écran (plus la zone est chaude plus le défilement est freiné). De plus, la vitesse de défilement diminue également au cours du temps en simulant une friction continue. La [FIGURE 3.22](#) illustre ce mécanisme de manière schématique.

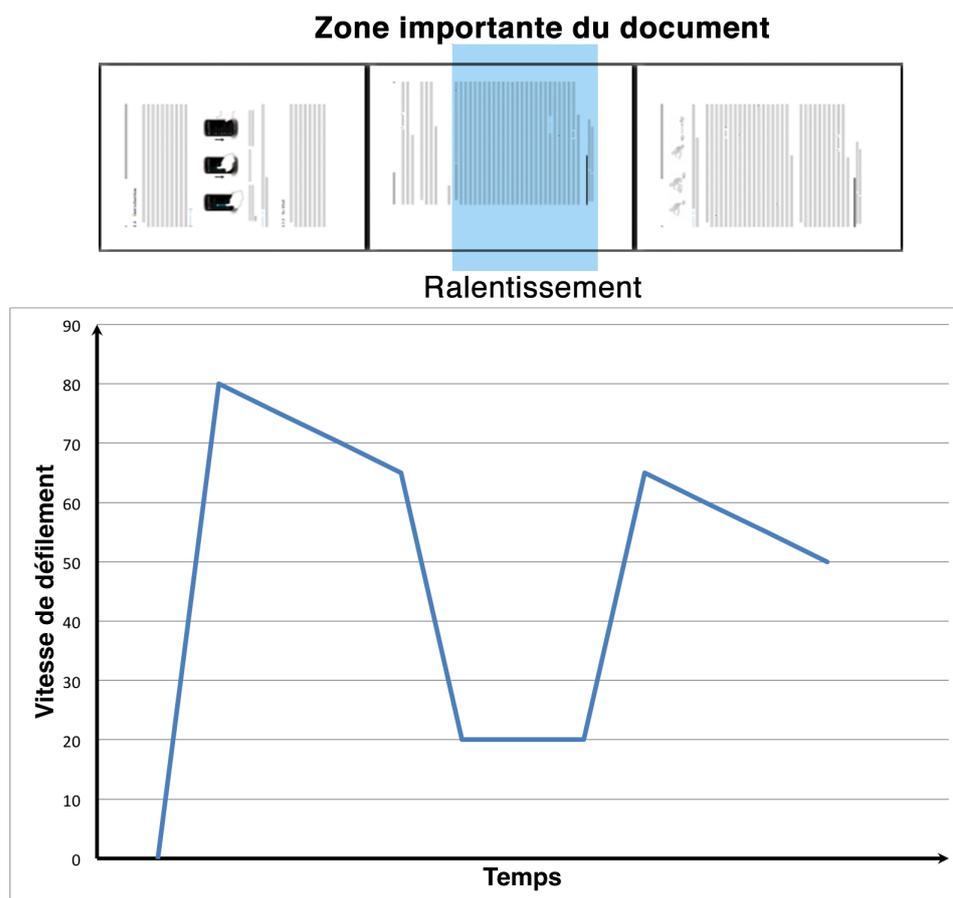


FIGURE 3.22: Ralentissement du défilement lorsqu'un passage plus important apparaît à l'écran.

Implementation. Le système S-Book est composé de deux modules : un côté client et un autre côté serveur. L'interface de lecture implémentant le Semantic Flicking ainsi que l'interface de recherche sont développées côté client. L'architecture de gestion des *heatmaps* est quant à elle développée côté serveur. Les deux modules communiquent par le biais d'une interface REST. La [FIGURE 3.23](#) donne le schéma complet du système réalisé.

La partie client a été développée en Javascript afin d'être facilement adaptable à tout type de support, que se soit un ordinateur de bureau ou une tablette mobile équipée d'un système Android, Windows ou iOS.

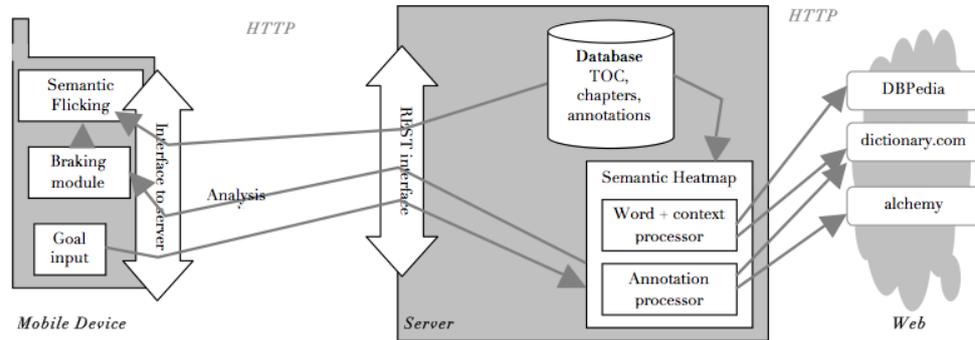


FIGURE 3.23: Schéma complet du système *heatmap* et Flicking sémantique

Semantic Map

Le concept de Semantic Flicking peut facilement être adapté à une interaction en deux dimensions. Un exemple typique est celui du défilement sur une carte de type Google Map, les villes étant les points d'intérêt. Une "chaleur" peut alors être affectée à chaque ville en fonction d'une requête utilisateur prenant typiquement en compte le nombre d'hôtels, de restaurants, de stations essences, etc. Cette technique permet de générer une "trajectoire sémantique" qui modifie la trajectoire rectiligne initiale au gré des attractions engendrées par les points d'intérêts, comme illustré sur la FIGURE 3.24. Cette technique permet ainsi de suggérer un parcours à l'utilisateur en lui montrant les villes les plus intéressantes par rapport à ses critères (ou, éventuellement, ceux rapportés par d'autres internautes, comme dans le cas précédent).



FIGURE 3.24: Représentation des trajectoires de déplacement en fonction de la nature du Flick (sémantique ou non).

Dans le cas du S-Book la vitesse de défilement décélère progressivement au cours du temps. Dans le cas de grands documents ou de cartes sémantiques il est sans doute préférable de maintenir la vitesse, comme pour Flick-and-Brake. L'utilisateur peut ainsi avoir une idée globale du document sans avoir à faire d'efforts pour réamorcer le défilement. Il peut aussi obtenir un niveau de précision élevé sur les éléments clés, soit automatiquement grâce à une heatmap et les ralentissements qu'elle induit, soit manuellement en interagissant comme expliqué plus haut dans la section Flick-and-Brake. La lecture de grands documents où seulement certaines parties sont intéressantes pour l'utilisateur, deviendrait ainsi plus aisée.

3.3.4 Conclusion

Le Semantic Flicking permet de naviguer dans un document en focalisant l'attention de l'utilisateur sur les passages importants. Qu'il soit couplé avec la *heatmap* d'un texte ou les objets d'intérêt d'une carte, le Semantic Flicking permet à l'utilisateur d'avoir à la fois le contexte général et un focus sur les points de priorité. Le Semantic Flicking est facilement adaptable à toutes les interfaces basées sur le Flick mais nécessite toutefois une connaissance du contenu du document parcouru. Celle-ci est nécessaire afin de pouvoir distinguer, soit automatiquement, soit par une requête de l'utilisateur, ce qui est important de ce qui ne l'est pas. Cette application a fait l'objet d'un dépôt de brevet comme précisé précédemment et a suscité un fort intérêt des participants lors de sa présentation aux Open Days d'Alcatel en mai 2011.

3.4 Long Flick

3.4.1 Contexte

Cette dernière partie concerne l'*amorçage* de l'opération de Flick. Comme expliqué plus haut, peu de travaux s'intéressent au démarrage du mouvement avant le relâchement conduisant au défilement automatique. Dans MultiFlick [Aliakseyeu et al., 2008] cette étape est même complètement ignorée : le mouvement n'a aucun effet sur le document dans lequel on navigue, le défilement ne démarrant qu'au relâchement du doigt (FIGURE 3.25 (c)). En général, le mouvement effectué avant le relâchement opère un défilement du document égal à la distance parcourue par le doigt, comme pour une simple opération de défilement (*pan*) (FIGURE 3.25 (a), FIGURE 3.25 (b)). La distance parcourue avant le relâchement du doigt n'a donc aucune incidence sur le défilement automatique, le seul paramètre important étant la vitesse au relâchement. Nous proposons maintenant de nous intéresser à l'impact que peut avoir ce mouvement sur l'ensemble de l'interaction.

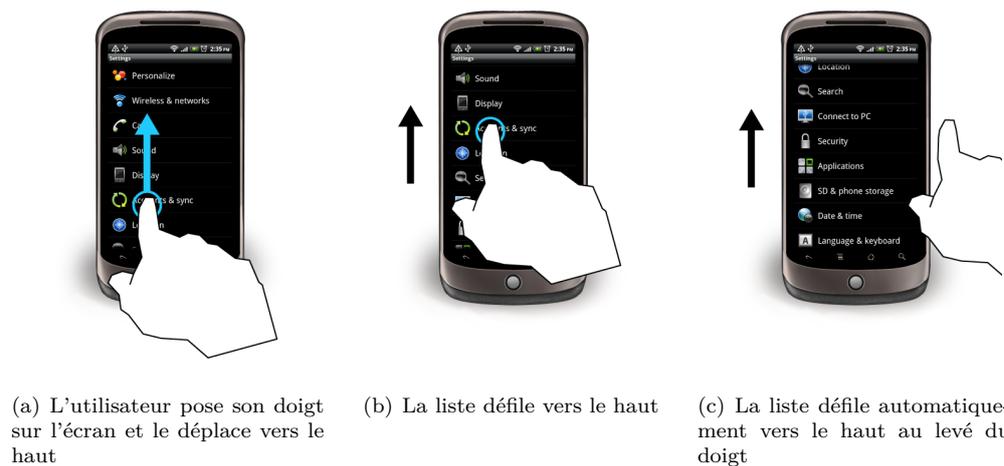


FIGURE 3.25: Le geste de Flick et le défilement (la flèche bleue représente le mouvement du doigt et la flèche noire celui de la liste).

3.4.2 Définitions

Nous avons un mouvement riche qui n'est généralement utilisé que pour du *pan* comme nous l'avons vu précédemment. Avant de proposer trois variantes d'utilisation du mouvement effectué avant le relâchement pour contrôler plus précisément le défilement, nous identifions cinq paramètres susceptibles d'être exploités : la position, la direction, la longueur, la forme et la vitesse.

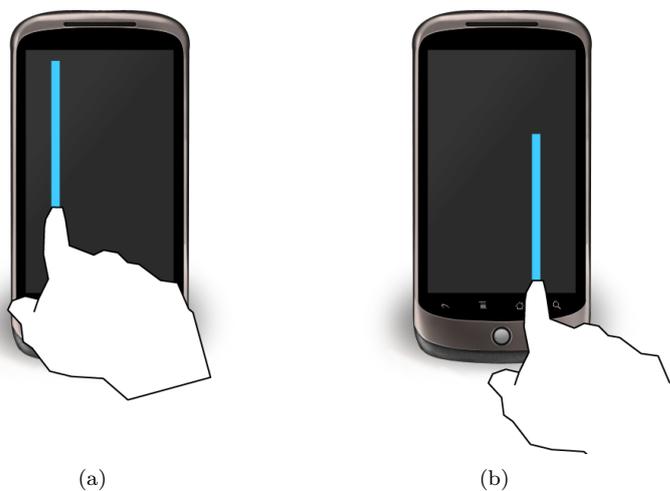


FIGURE 3.26: Différentes positions de démarrage du Flick

Position. Le geste de Flick peut commencer depuis différentes positions sur l'écran (FIGURE 3.26). Cette position initiale s'exprime en deux dimensions selon les axes x et y formés par l'écran. Lorsque l'on se trouve dans un système à une dimension de défilement, comme par exemple dans une liste (verticale) de contacts, la coordonnée en x n'a aucune incidence sur le défilement. Cette coordonnée peut donc être exploitée pour paramétrer le comportement du défilement automatique qui aura lieu au relâchement. Par exemple, x peut alors servir à moduler la vitesse du défilement automatique de telle sorte que plus le point de départ du geste sera proche de la droite de l'écran, plus la vitesse au relâchement sera grande.

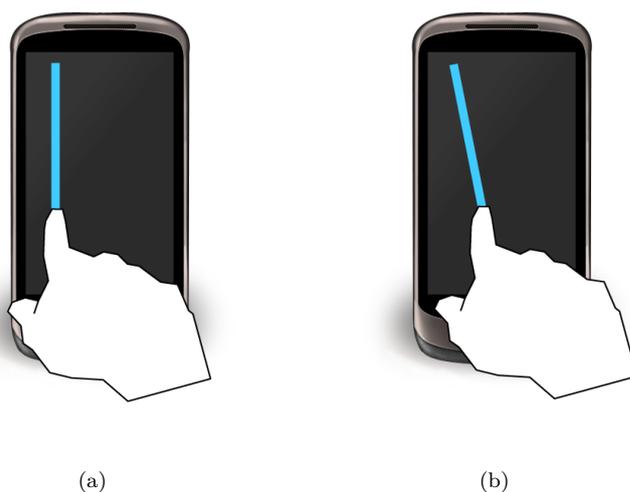


FIGURE 3.27: Différentes directions du Flick

Direction. La direction représente le vecteur formé par le point de contact initial du geste et le point au relâchement (FIGURE 3.27). Elle donne le sens de défilement et s'exprime en deux dimensions selon les axes x et y formés par l'écran. Dans un système à une dimension de défilement l'angle formé par ce vecteur et l'axe de défilement n'est généralement pas utilisé. Cet angle pourrait servir comme dans l'exemple précédent à influencer sur la vitesse au relâchement. Par exemple, plus l'angle serait petit plus la vitesse serait grande.

Longueur. La longueur du tracé entre le point de contact initial et le point de relâchement peut également être mise à profit (FIGURE 3.28). Comme vu précédemment, lors de cette phase la plupart des techniques de Flick effectuent juste un *pan* de telle sorte que l'objet manipulé suive le doigt. Nous proposons d'utiliser la longueur du tracé pour affecter non seulement le déplacement effectué lors de cette phase de Pan mais aussi le défilement automatique qui s'ensuit. Nous présenterons des applications de cette idée dans la section suivante.

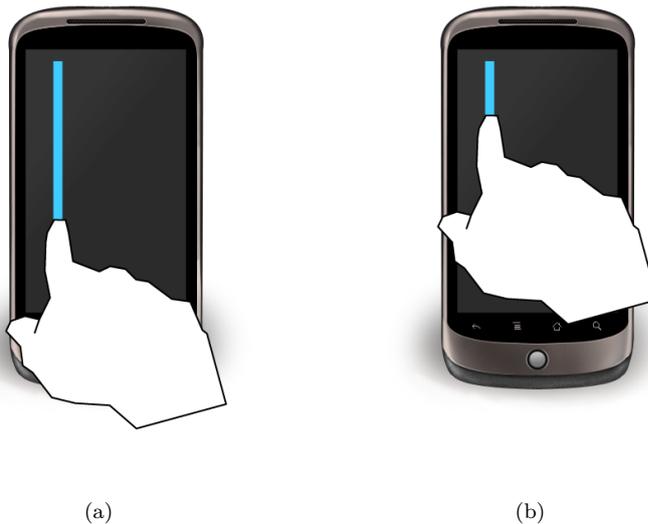


FIGURE 3.28: Différentes longueurs du geste du Flick

Forme. La forme du tracé réalisé avant le relâchement peut être plus complexe que celle d'une simple droite : elle peut par exemple comprendre une courbe, une boucle, etc. Ce paramètre pourrait également être utilisé pour contrôler le défilement automatique. Par exemple, dans le cas d'un défilement en deux dimensions, si la trajectoire de l'amorce représente une boucle comme sur la FIGURE 3.29, cette trajectoire pourrait être répétée durant la phase de défilement automatique.



FIGURE 3.29: Exemple de forme complexe du geste de Flick

Vitesse. La vitesse au cours du tracé est souvent négligée. Seule la vitesse au relâchement est prise en compte et a un impact sur le défilement. Les variations de vitesse durant l'étape d'amorce pourraient également être utilisées afin de moduler l'étape de défilement automatique. Une décélération lors de l'amorce pourrait ainsi servir à indiquer que le défilement automatique doit être soumis à une friction. Inversement, une accélération dans le mouvement d'amorce exprimerait le souhait d'augmenter progressivement la vitesse lors du défilement automatique. De la même manière, une courbe linéaire de vitesse lancerait un défilement à vitesse constante.

Nous nous sommes intéressés plus particulièrement à deux paramètres : la longueur et la position. Nous décrivons ci-après différentes manières de les exploiter pour contrôler le défilement automatique.

3.4.3 Exploitation

Dans la suite nous considérons le cas d'une liste à une dimension pour montrer les possibilités offertes par l'utilisation de ces paramètres. L'idée est de modifier le comportement de la vitesse au relâchement pendant une certaine distance en fonction du geste de l'amorce. Nous proposons ci-après plusieurs fonctions de défilement offrant un mapping différent entre geste d'amorçage et défilement.

Mapping Direct. Lors de l'amorce du Flick, une distance d est parcourue par le doigt sur l'écran. La liste suit le déplacement et défile de cette même distance jusqu'au relâchement du doigt. A ce moment, le défilement automatique est activé, mais d'abord à vitesse constante

pendant une certaine distance D avant de décroître comme lors d'un Flick standard. D est fonction de la distance d'amorçage d et peut s'exprimer ainsi : $D(d) = d * k$ où k est un facteur constant défini par le système. Cette technique permet donc un contrôle de la distance de défilement automatique, qui sera d'autant plus longue que la distance d'amorçage le sera. Alternativement, une variante consisterait à utiliser deux facteurs de friction différents : un pour chaque partie du défilement automatique.

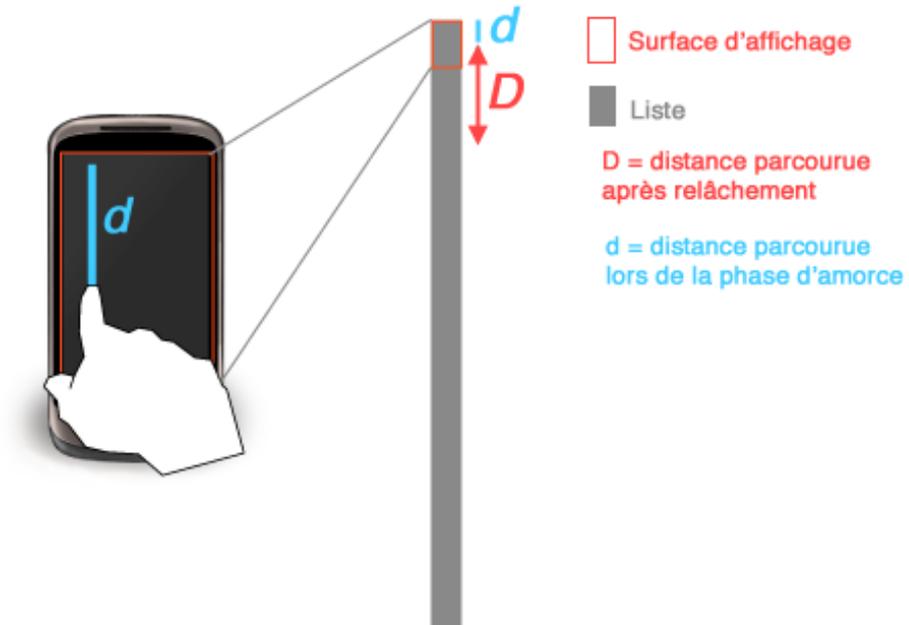


FIGURE 3.30: Mapping direct : $D = d * k$. d est la longueur parcourue par le doigt et D la distance de défilement automatique à vitesse constante.

Mapping Proportionnel Absolu. Le mapping absolu consiste à mettre en correspondance la surface d'affichage de la liste sur l'écran avec le contenu intégral de la liste (y compris les éléments non visibles). La distance d parcourue par le doigt sur l'écran est alors considérée comme un pourcentage de la taille verticale de la surface d'affichage. La distance D parcourue lors du défilement automatique correspondra au même pourcentage, cette fois par rapport à la taille totale de la liste. En d'autres termes, $D = d * L / S$ si L est la longueur de la liste et S celle de la surface d'affichage. Cette technique permet d'atteindre la partie souhaitée avec un seul Flick. Elle impose cependant de bien choisir le point de départ afin de pouvoir tracer la distance voulue avec le doigt et de pouvoir atteindre l'endroit souhaité de la liste. Le rapport entre la distance parcourue et le tracé du doigt reste constant. Comme précédemment la distance D peut être parcourue à vitesse constante ou variable.

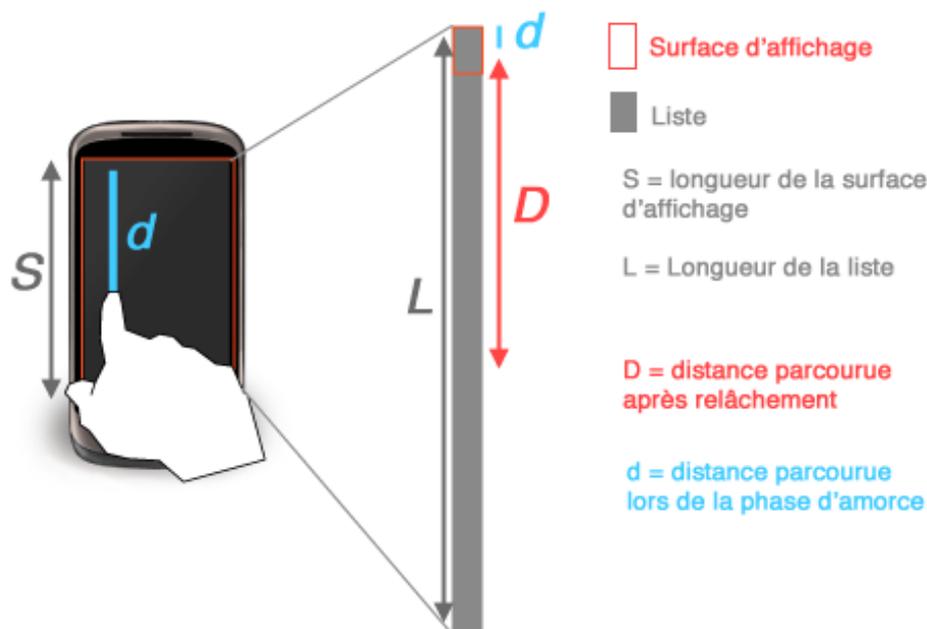


FIGURE 3.31: Mapping Absolu : $D(d) = d * L / S$. d la longueur du tracé, L la longueur totale de la liste, S la longueur de la surface d'affichage et D la distance parcourue.

Mapping Proportionnel Relatif. Ce dernier mapping reprend l'idée de mapping proportionnel mais relativement à la position courante dans la liste. En effet la taille de la surface d'affichage ne correspond plus à la longueur totale de la liste, mais à la longueur restante entre la position actuelle dans la liste et la fin de la liste. Le rapport entre la longueur parcourue et la longueur du tracé varie en fonction de la position dans la liste. Prenons deux positions différentes en exemple : le haut de la liste et le milieu de la liste. Lorsque l'on se trouve en haut de la liste, la longueur de la surface d'affichage correspondra à l'intégralité de la liste. En revanche lorsque l'on se situe au milieu de la liste, la longueur de la surface correspondra à la moitié restante de la liste. Si l'on nomme RL la longueur de liste restante et S la longueur de la surface d'affichage, on obtient la formule suivante $D(d) = d * RL / S$. Notons que la valeur de RL varie en fonction du sens de défilement défini par le Flick. Dans un sens la valeur correspond à la longueur entre la position courante et la fin de la liste et dans l'autre sens à la longueur entre la position courante et le début de la liste. Cette technique permet comme la précédente de choisir la destination du défilement. Contrairement au mapping précédent celui-ci n'a pas un rapport constant entre la distance du tracé et la distance parcourue. Ainsi pour un même geste de tracé, selon la position dans la liste, la distance de défilement ne sera pas la même. Ceci permet d'avoir une meilleure précision que le mapping absolu. Ainsi pour atteindre la moitié de la distance restante, quelle-que-soit la distance, l'utilisateur fera un tracé de la moitié de la surface d'affichage.

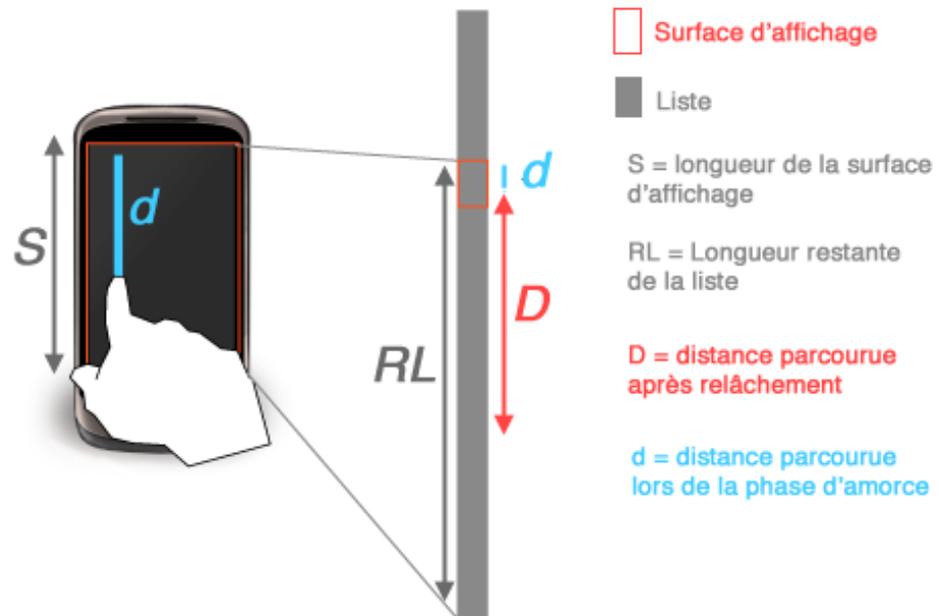


FIGURE 3.32: Mapping Relatif : $D(d) = d * RL/S$. d la longueur du tracé, RL la longueur restante de la liste, S la longueur de la surface d'affichage et D la distance parcourue.

3.4.4 Conclusion

Cette section a mis en évidence l'intérêt de paramètres habituellement non exploités qui permettent de mieux contrôler la navigation et d'économiser le nombre de mouvements réalisés. Dans le cas de documents de taille importante, LongFlick permet de survoler le document à vitesse constante sur une longue distance, ce qui permet de le parcourir à vitesse de lecture sans avoir à réaliser de multiples gestes de Flick. La seconde variante de LongFlick présente l'avantage de pouvoir se rendre directement dans la partie de la liste souhaitée, ce qui est avantageux dans le cas d'un usage expert où l'utilisateur est capable d'évaluer la position de l'élément recherché.

Cette technique (et ses trois variantes) ouvre de nouveaux horizons quand à l'utilisation du Flick. Elle montre qu'il reste encore de nombreuses possibilités d'amélioration du Flick, une technique qui semble aujourd'hui s'imposer sur toutes les plate-formes. De ce fait, même des changements qui pourraient sembler mineurs peuvent avoir un impact notable sur l'interaction étant donné l'omniprésence de cette technique d'interaction dans les interfaces actuelles. Il serait maintenant intéressant de poursuivre ce travail par une évaluation de cette technique, afin de définir la configuration optimale de ces trois mappings. Par ailleurs, les trois autres paramètres évoqués plus haut mériteraient d'être davantage considérés pour concevoir de nouvelles techniques.

4

Mouvements du dispositif

Sommaire

4.1	Introduction	94
4.1.1	Contexte	94
4.1.2	Types de geste	94
4.1.3	Délimiteurs	95
4.1.4	Deux nouvelles techniques gestuelles	95
4.2	TimeTilt	97
4.2.1	Présentation	97
4.2.2	Vocabulaire Gestuel	97
4.2.3	Evaluation	98
4.2.4	Application	100
4.2.5	Conclusion	106
4.3	JerkTilts	107
4.3.1	Présentation	107
4.3.2	Vocabulaire Gestuel	107
4.3.3	Évaluation	110
4.3.4	Application	121
4.3.5	Extension de JerkTilts : Combinaison 3D/2D	124
4.3.6	Conclusion	130

4.1 Introduction

4.1.1 Contexte

Les terminaux mobiles actuels possèdent un nombre très réduit de boutons physiques, notamment pour maximiser l'espace dédié à l'écran. Ces boutons sont réservés à des usages bien spécifiques comme l'augmentation/réduction du volume, la mise en veille de l'appareil ou la fermeture d'une application. L'augmentation des moyens d'interaction (i.e. de la bande passante interactionnelle entre l'utilisateur et son mobile) passe ainsi nécessairement par l'exploitation d'autres dispositifs, et en particulier les capteurs de mouvements. Ceux-ci sont essentiellement de deux sortes : l'écran tactile pour détecter les mouvements 2D du doigt sur l'écran et les capteurs embarqués (accéléromètres, gyroscope) pour les mouvements 3D du mobile. Les interfaces actuelles proposent de nombreuses interactions tactiles avec l'écran, ce qui rend difficile l'apport de nouvelles techniques ne parasitant pas les interactions déjà présentes. L'utilisation d'un langage gestuel basé sur les accélérations du dispositif lui-même apparaît ainsi comme une alternative intéressante, d'autant que ce type de capteur est désormais très répandu. De fait, la plupart des utilisateurs de téléphones mobiles modernes s'en sont déjà servi (peut-être sans s'en rendre compte), par exemple lorsqu'ils changent l'orientation de leur mobile et que l'interface s'adapte en conséquence. Cependant l'utilisation de ce type de geste nécessite une détection non ambiguë du début et de la fin du geste réalisé. Nous abordons cette question dans les sous-sections suivantes après un bref rappel de notre typologie des gestes.

4.1.2 Types de geste

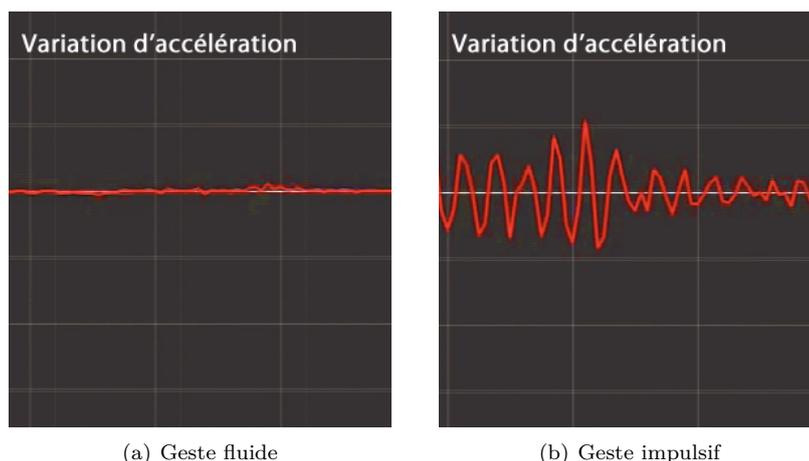


FIGURE 4.1: Profil temporel d'accélération pour les deux types majeurs de geste

Comme nous l'avons vu dans le chapitre consacré à l'espace de classification, on peut

distinguer deux types de mouvement majeurs pour caractériser les interactions gestuelles : les mouvements impulsifs et les mouvements fluides. On trouve de plus des mouvements oscillatoires ou des combinaisons de mouvements, le caractère répétitif ou combinatoire pouvant être associé aux gestes fluides ou impulsifs. Comme vu précédemment, la différence majeure entre ces deux types de geste vient des variations d'accélération qui sont détectées lors de la réalisation du mouvement. Comme illustré sur la [FIGURE 4.1 \(b\)](#) on observe de forts pics et de fortes variations d'accélération pour le geste impulsif et cela dans un laps de temps très court. En revanche les gestes fluides ne laissent apparaître que de faibles variations d'accélération ([FIGURE 4.1 \(a\)](#)).

4.1.3 Délimiteurs

Qui dit interaction gestuelle dit délimiteur. Il n'est en effet pas possible de séparer ce type d'interaction de l'action qui va permettre de définir le début et la fin d'un geste. Lorsqu'il s'agit de gestes sur un écran tactile, la présence ou l'absence de contact sur l'écran est un délimiteur naturellement inclus dans la réalisation du geste. En d'autres termes les gestes sur l'écran n'exploitent que deux dimensions, ce qui permet d'utiliser la dimension restante (appui ou relâchement) pour signifier le début ou la fin du geste. Ceci n'est évidemment pas possible avec des gestes tri-dimensionnels effectués avec le mobile (ni sur le mobile). De plus, comme les dispositifs mobiles sont en déplacement constant dans un contexte de mobilité, il est généralement nécessaire de recourir à des délimiteurs explicites pour distinguer les gestes intentionnels (ceux qui visent à entraîner une action, typiquement activer une commande) de ceux qui résultent de la simple manipulation du mobile (lorsqu'on le prend dans la main ou qu'on le dépose sur une surface, lorsqu'il est dans la poche et pas encore éteint, etc.).

Pour répondre à ce problème les techniques antérieures [Oakley and Park, 2007, Bartlett, 2000, Wigdor and Balakrishnan, 2003] proposent généralement d'utiliser le contact du doigt avec l'écran ou de la pression d'un bouton comme délimiteur. Cependant ceci va à l'encontre de la motivation principale de l'utilisation des gestes qui est de préserver les interactions sur l'écran et les fonctionnalités des boutons. Une solution à ce problème, qui est celle que nous étudions dans ce chapitre, consiste à exploiter les gestes impulsifs pour en faire des délimiteurs naturels. Ceux-ci sont en effet plus facilement identifiables que les gestes fluides vis-à-vis des mouvements non intentionnels dus à la manipulation d'un dispositif mobile.

4.1.4 Deux nouvelles techniques gestuelles

Dans ce chapitre nous présentons deux nouvelles techniques d'interaction gestuelle : TimeTilt et JerkTilts. On trouvera en Annexe 5.3.3 un schéma présentant le module de reconnaissance de geste que nous avons réalisé pour ces techniques.



TimeTilt Cette première technique répond à certains manques mis en évidence précédemment, en particulier l'adéquation au contexte d'utilisation (gestes simples et de faible amplitude) et l'utilisation d'un procédé alternatif servant de délimiteur. TimeTilt, réalisé en collaboration avec Anne Roudaut est une technique utilisant les accéléromètres et le langage gestuel. Cette technique propose deux modes d'interaction complémentaires. Nous présentons également une application du système permettant de changer aisément et rapidement de fenêtre sur un dispositif mobile. Cette technique se sert de la métaphore des cartes à effet lenticulaire qui permettent, selon l'inclinaison de la carte, de présenter plusieurs images distinctes pour en faciliter l'apprentissage. Elle a fait l'objet d'un article court à INTERACT 2009 et d'une démonstration lors des OpenDays 2009 sur le site d'Alcatel-Lucent à Villarceaux.



JerkTilts Cette seconde technique qui utilise également les accéléromètres et le langage gestuel fournit un ensemble de gestes permettant l'accès rapide à des commandes. Elle a de plus la particularité d'utiliser des gestes dits *auto-délimités*, ce qui permet de les réaliser sans avoir recours à une activation préliminaire (bouton physique ou virtuel ou geste spécifique servant de délimiteur). Ces JerkTilts sont des rotations aller-retour de faible amplitude et de type impulsif permettant une bonne adéquation au contexte de mobilité. Ils peuvent aussi être combinés avec les interactions usuelles effectuées *sur* le dispositif sans risque d'interférences. Cette technique d'interaction a donné lieu à la publication d'un article long à ICMI 2011.

4.2 TimeTilt

4.2.1 Présentation

Nous proposons TimeTilt, une technique d'interaction basée sur l'utilisation des accéléromètres. Cette technique propose un vocabulaire gestuel qui préserve les interactions déjà existantes. Elle fournit un ensemble de gestes fluides et de gestes impulsifs (nous appelons "gestes fluides" (respectivement impulsifs) ceux composés de mouvements fluides (respectivement impulsifs)), dont un servant d'activateur pour la détection de gestes basés sur des mouvements fluides. Cette idée a été ensuite reprise dans [Ruiz and Li, 2011]. Nous présentons d'abord le vocabulaire gestuel de TimeTilt. Dans un second temps, afin de vérifier l'exploitabilité des gestes proposés par TimeTilt nous avons conduit une étude visant à discriminer les différents gestes impulsifs. Enfin nous proposons une application de TimeTilt dans un contexte réel d'interaction.

4.2.2 Vocabulaire Gestuel

TimeTilt utilise un vocabulaire composé de deux types de geste : impulsifs et fluides. Les gestes fluides correspondent à des inclinaisons du dispositif tandis que les gestes impulsifs sont composés d'un tap (servant d'activateur pour les gestes fluides) et de deux secousses, une vers l'avant et l'autre vers l'arrière. Ces deux derniers gestes peuvent être qualifiés d'"auto-délimités" dans la mesure où ils permettent d'effectuer une action sans avoir à recourir préalablement à un activateur (nous reviendrons plus avant sur cette notion dans la section dédiée à JerkTilts). On peut voir sur la [FIGURE 4.2](#) les différentes variations d'accélération qui entrent en jeu lors de la réalisation de ces différents gestes.

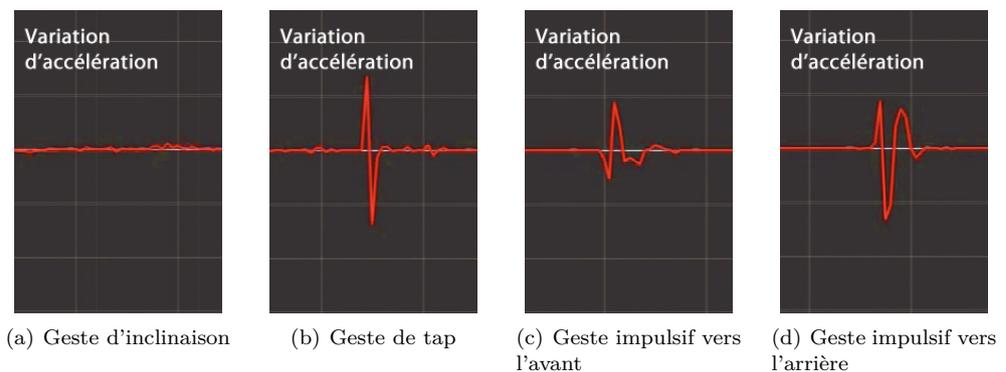


FIGURE 4.2: Variations d'accélération selon l'axe z du mobile pour les gestes de TimeTilt

Les inclinaisons

Les accéléromètres présents sur les dispositifs mobiles mesurent entre autres l'accélération gravitationnelle terrestre. Ceci implique que lorsque le dispositif est en position fixe le vecteur d'accélération selon les trois composantes x , y et z (du référentiel mobile) a toujours une norme égale à $g=9,8m/s^2$. Grâce à cette propriété ils permettent ainsi de servir d'inclinomètres. On peut alors calculer l'angle du vecteur accélération dans le référentiel terrestre et ainsi détecter le *pitch* et le *roll* du mobile (FIGURE 2.7 page 28).

Les gestes impulsifs

Pour détecter les gestes impulsifs nous utilisons la variation d'accélération au cours du temps selon l'axe z . Cette méthode comporte plusieurs avantages : tout d'abord la variation d'accélération en z , dans notre cas, ne dépend pas de l'orientation du dispositif, ce qui permet de discriminer les gestes impulsifs quelle que soit la position de l'appareil dans la main de l'utilisateur. De plus, sa valeur reste relativement constante au cours du temps lors d'un mouvement fluide (puisque l'accélération varie peu). Ceci nous permet de discriminer les deux types de geste suivants : si la variation d'accélération dépasse un certain seuil (plus ou moins $4 m/s^2$) pendant au moins 300 ms (valeurs que nous avons déterminées après plusieurs essais), c'est un geste impulsif, sinon c'est un geste fluide. Nous pouvons ainsi déterminer le début et la fin d'un geste impulsif pour ensuite l'identifier grâce à un algorithme de reconnaissance.

Algorithme de reconnaissance Nous avons alors utilisé un algorithme des K -plus-proches-voisins (K -ppv ou K -nn¹) afin de catégoriser les gestes. En nous inspirant des travaux de Rubine [Rubine, 1991] nous avons pu identifier 5 primitives auxquelles nous avons attribué des poids différents afin d'étiqueter les différents échantillons de geste :

- L'amplitude maximale entre deux valeurs d'accélération consécutives (poids : 0.1).
- La durée totale du geste (poids : 0.3).
- La durée entre la plus haute et la plus basse valeur (poids : 0.4)
- Le nombre de changements de signe durant le geste (poids : 0.1)
- Le signe du premier pic d'accélération (poids : 0.8)

Ces différentes primitives définissent un vecteur qui est la signature du geste, la moitié des gestes récoltés servant de base d'apprentissage et l'autre moitié de base de test. Nous avons fixé K à 3, cette valeur nous donnant les meilleurs résultats.

4.2.3 Evaluation

Nous avons mené une expérimentation afin de démontrer la faisabilité des trois gestes impulsifs (tap, impulsif avant et arrière) et de montrer qu'ils sont détectables et susceptibles

1. http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

d'être discriminés les uns des autres.

Prérequis Nous avons demandé aux utilisateurs de réaliser chacun des trois gestes de deux manières différentes :

- Le tap est réalisé soit en gardant le doigt appuyé sur le mobile à la fin du geste, soit en relevant le doigt.
- Les deux autres gestes (mouvements impulsifs arrière et avant) sont réalisés avec deux intensités différentes (forte et faible).

Tâche et Design Expérimental Les utilisateurs devaient répéter chaque geste six fois. Le stimulus était donné par l'examineur. Entre chaque geste l'utilisateur replaçait son dispositif dans une position stable afin de faciliter le traitement des données. Chaque utilisateur avait 5 minutes d'entraînement. Un carré latin a servi à contrebalancer l'ordre des gestes entre les participants. Nous avons donc 3 gestes x 2 variations x 6 répétitions = 36 échantillons de gestes par participant (pour environ 10 min).

Matériel et Participants Nous avons utilisé un HTC Touch HD avec Windows Mobile 6. L'application était implémentée en C#. Huit volontaires (dont 2 femmes) ont été recrutés dans notre institution. Leur participation était gratifiée d'une poignée de bonbon.

Résultats Nous avons analysé les différentes variations d'accélération pour chaque geste et utilisé l'algorithme présenté précédemment pour les classifier. Le taux de reconnaissance moyen est de 92,3%. Les détails se trouvent dans la matrice de confusion (TABLEAU 4.1). Comme on peut le voir, les gestes de Tap sont les mieux reconnus. La base de geste étant assez réduite, on peut raisonnablement s'attendre à une amélioration des résultats lorsque la base d'apprentissage est plus grande. Un deuxième facteur pouvant améliorer les résultats est l'utilisation d'un système d'apprentissage qui permettrait au système d'apprendre les gestes spécifiques à chaque utilisateur.

	TAr	IAv	IAr
TAr	93,8%	4,1%	2,1%
IAv	2,1%	91,7%	6,2%
IAr	4,15%	4,15%	91,7%

TABLEAU 4.1: Matrice de confusion de reconnaissance des gestes Tap à l'arrière(TAr), Impulsif vers l'avant(IAv) et Impulsif vers l'arrière(IAr)

4.2.4 Application

Afin d'illustrer TimeTilt dans un contexte réel nous avons choisi de l'adapter à la navigation entre applications. En effet plusieurs systèmes ont été proposés sur les dispositifs commerciaux pour résoudre ce problème mais ils restent liés à l'interaction 2D. Apple propose par exemple sur l'iPhone ou l'iPad et plus particulièrement à partir d'iOS 4 la possibilité, en double-cliquant sur le bouton (FIGURE 4.3 (a)) en façade du dispositif, d'accéder à la liste des applications ouvertes et de pouvoir ensuite naviguer en faisant défiler la liste d'icônes (FIGURE 4.3 (b) - FIGURE 4.3 (c)). Il suffit ensuite de cliquer (FIGURE 4.3 (d)) sur l'icône de l'application désirée pour la faire passer au premier plan (FIGURE 4.3 (f)).

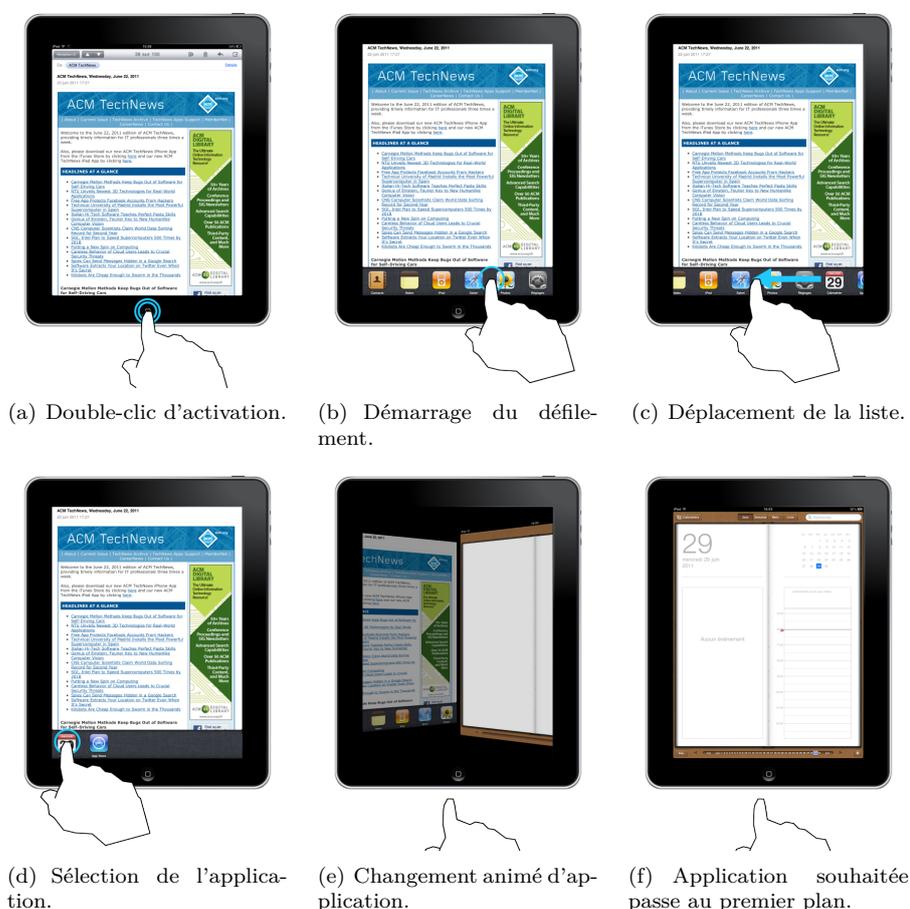


FIGURE 4.3: Changement entre deux applications sur iPad

On peut également trouver sur les dispositifs Android une interface qui permet d'atteindre les applications ouvertes. L'appui prolongé sur la touche *Home* du téléphone (FIGURE 4.4 (a)) permet de faire apparaître la liste des huit dernières applications utilisées (FIGURE 4.4 (b)). Il suffit alors de cliquer sur l'application que l'on souhaite afficher (FIGURE 4.4 (c)).

FIGURE 4.4 (c)). A l'époque où nous avons développé TimeTilt, il n'existait pas d'interface pour changer facilement d'application sur mobile, notamment sur le système Windows Mobile 6 avec lequel nous avons développé notre technique. Il fallait en effet fermer l'application en cours pour pouvoir en ouvrir une autre, et c'était encore le cas sur l'iPhone jusqu'à iOS 4. Encore aujourd'hui on ne trouve pas de technique qui fasse référence pour changer de fenêtre sur mobile, chaque système d'exploitation (iOS, Android, WM7, Meego, etc.) proposant sa propre version au moyen d'interactions plus ou moins commodes.



FIGURE 4.4: Changement entre deux applications sur le Nexus One sous Android

TimeTilt fournit un moyen pour faciliter la navigation entre les fenêtres - ou vues - ouvertes sur le dispositif, ceci étant d'autant plus utile que la petite taille des écrans ne permet pas l'affichage de plusieurs fenêtres. Nous avons développé un prototype tournant sur le système Windows Mobile qui détecte l'ouverture des applications et mémorise les six dernières applications ouvertes dans une liste. Ceci permet à l'utilisateur de naviguer dans les applications qu'il utilise le plus fréquemment à un moment donné. Lors de la réalisation de notre application, nous avons supposé que les utilisateurs n'utilisaient pas plus de six

applications durant les périodes d'utilisation du mobile (cette hypothèse a par ailleurs été confirmée par un questionnaire, comme nous le verrons par la suite).

TimeTilt propose deux techniques de navigation : Lenticulaire et par Superposition. La première technique permet d'afficher les applications ouvertes et de les parcourir pour sélectionner celle à activer tandis que la seconde permet de passer directement d'une application à la précédente ou à la suivante. Nous présentons ces deux techniques ainsi que le questionnaire réalisé pour valider l'utilité d'une telle application.

Navigation Lenticulaire

Cette technique repose sur une métaphore du monde réel : les cartes dites "holographiques" qui permettent en fonction de l'angle de vision de voir différentes images (FIGURE 4.5). Nous avons simulé ce procédé dans le monde virtuel en présentant les captures d'écran des différentes applications ouvertes en fonction de l'inclinaison du dispositif mobile (FIGURE 4.6). L'utilisateur peut ainsi naviguer à travers les applications. Il lui suffit alors de cliquer sur l'écran lorsque l'application souhaitée apparaît à l'écran.

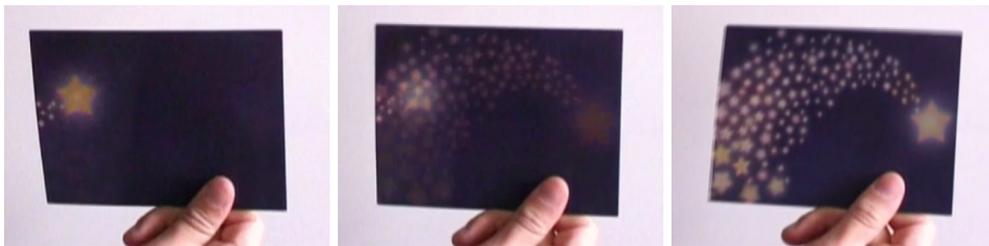


FIGURE 4.5: Carte lenticulaire ou holographique présentant différentes images en fonction de l'angle de vision.



FIGURE 4.6: Effet lenticulaire virtuel en fonction de l'inclinaison du mobile.

Un des points remarquables est l'utilisation d'un geste comme déclencheur de ce mode de navigation. En effet, les gestes d'inclinaison que nous utilisons nécessitent un activateur pour ne pas être confondus avec les variations d'inclinaison inhérentes à l'utilisation normale de ce type de dispositif. Nous avons donc proposé de rajouter un 'bouton gestuel' au téléphone. Plus précisément nous utilisons un choc, ou *tap*, (FIGURE 4.7) avec le doigt au dos du mobile pour déclencher la navigation lenticulaire. Nous préservons ainsi les boutons déjà utilisés pour les tâches spécifiques auxquels ils sont associés (volume, quitter une application, etc.) et tirons partie des doigts de la main qui ne servent habituellement qu'à tenir le dispositif. La nature brusque de ce geste permet de le discriminer du bruit des mouvements liés à l'usage commun du mobile.



FIGURE 4.7: Tap de l'index au dos pour l'activation de la navigation.



FIGURE 4.8: L'angle de 90 degrés de rotation du mode lenticulaire pour différentes positions initiales.

La navigation lenticulaire permet enfin de comparer plusieurs vues ou de vérifier une information sans avoir à afficher une vue en plein écran. Les aperçus sont en effet suffisamment grands pour afficher clairement le contenu des vues. Ceci peut être par exemple utile pour comparer des vues représentant différents niveaux de zoom sur une carte. Nous utilisons un angle de rotation de 90 degrés fixe (FIGURE 4.8) pour assurer une bonne visibilité de l'écran. Cet arc de cercle est alors séparé en fonction du nombre d'applications ouvertes en secteurs

d'égale dimension (relativement à la position initiale du mobile au moment de l'activation). La vue affichée à l'activation est toujours la vue de l'application courante. L'utilisateur a alors un contrôle en position (et non en vitesse) afin d'avoir une interaction d'une meilleure précision ([Oakley and Park, 2007]).

Navigation par Superposition

Cette technique est inspirée du *Alt-Tab* que l'on trouve sur un ordinateur de Bureau. *Alt-Tab* est un raccourci clavier qui permet de revenir aux applications précédemment ouvertes. Notre technique fonctionne sur le même principe : un geste brusque, vers le haut ou vers le bas, permet d'atteindre l'application précédente ou de passer à la suivante (l'organisation de la liste des applications ouvertes étant circulaire). Nous avons utilisé deux gestes opposés le long de l'axe normal à l'écran (FIGURE 4.9).

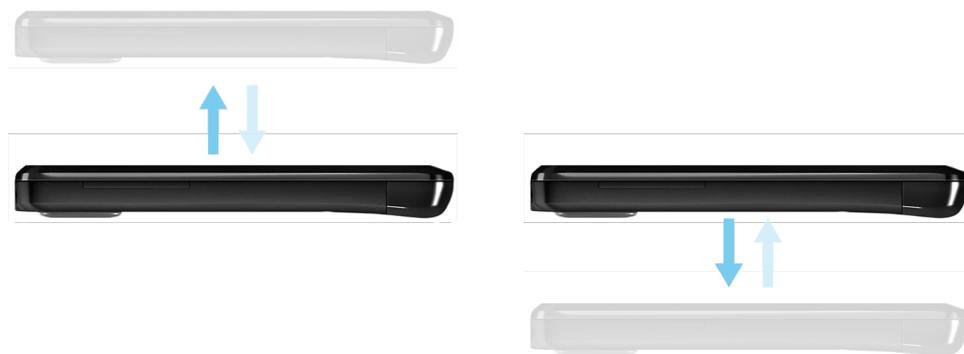


FIGURE 4.9: Les deux gestes du mode Superposition. Un geste impulsif vers le haut et un vers le bas (vue longitudinale du dispositif).

Cette technique de navigation permet également, comme pour la précédente, de faciliter le copier/coller d'objets entre plusieurs vues. La taille des écrans ne permettant pas d'afficher plusieurs fenêtres simultanément, il est donc particulièrement utile de disposer d'une technique permettant de passer rapidement d'une vue à une autre.

Questionnaire

Nous avons supposé précédemment que les utilisateurs n'utilisent pas plus de six applications. Pour vérifier cette hypothèse et l'intérêt d'un tel service, nous avons mené une enquête préliminaire qui se présentait sous la forme d'un questionnaire en ligne (envoyé par mail) incluant 12 questions regroupées en trois parties. La première partie visait à collecter des informations relatives aux répondants et aux types de dispositif qu'ils utilisaient afin de vérifier s'ils appartenaient à la population cible (possesseurs de dispositifs à écran tactile).

La seconde partie présentait des questions relatives à l'utilisation de leurs dispositifs au cours de la journée. La dernière partie posait des questions sur la navigation entre fenêtres. La durée nécessaire pour remplir le formulaire était relativement courte (5 minutes) afin d'inciter le plus de personnes à répondre. 48 possesseurs (âgés de 23 à 35 ans) de dispositifs mobiles à écran tactile ont rempli le formulaire en ligne.

La première partie du questionnaire a révélé que les utilisateurs sondés possédaient en moyenne 30 applications (entre 5 et 100). 77% d'entre eux avaient un accès Internet et s'en servaient. Ils étaient donc susceptibles de télécharger d'autres applications. Dans la seconde partie nous avons demandé aux répondants la fréquence et la durée d'utilisation de leur dispositif mobile au cours de la journée. La majorité (42%) utilisait leur dispositif environ 4 fois par jour, 33% toutes les heures et 17 % toutes les 30 minutes. La durée variait entre 1 ou 2 minutes (40%) et 10 minutes (17%), en passant par 5 minutes (38%). Seulement 4% des sondés y consacraient 30 minutes ou plus. Les répondants ouvraient entre 3 et 4 applications en moyenne lors de ces utilisations, qui sont donc relativement brèves mais fréquentes au cours de la journée. Dans la dernière partie, les questions posées étaient relatives à la navigation entre plusieurs fenêtres. La FIGURE 4.10 montre les réponses aux questions suivantes : FIGURE 4.10 (a) vous est-il arrivé de vouloir utiliser ou d'utiliser plusieurs fenêtres en même temps ? FIGURE 4.10 (b) vous est-il arrivé de vouloir déplacer du texte ou des fichiers entre plusieurs fenêtres ? Enfin 80% des participants ont répondu avoir envie d'une technique pour faciliter le passage entre plusieurs fenêtres.

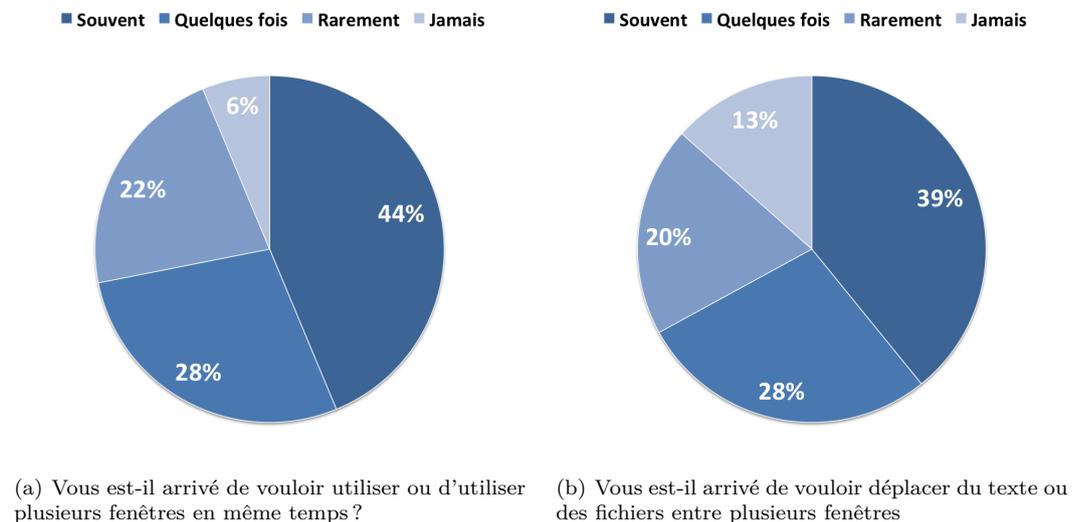


FIGURE 4.10: Taux de réponse à deux questions du questionnaire.

En résumé, cette enquête montre deux résultats intéressants pour notre étude : d'une part les utilisateurs souhaitent avoir une technique leur permettant de naviguer plus facile-

ment entre les fenêtres et de copier coller des éléments d'une fenêtre à une autre. D'autre part, les participants ont indiqué utiliser entre 3 et 4 applications lors de l'utilisation journalière de leur dispositif mobile. Le mode lenticulaire de TimeTilt pouvant aisément accueillir jusqu'à 6 fenêtres, les utilisateurs pourraient ainsi facilement naviguer entre les applications.

4.2.5 Conclusion

Nous avons présenté TimeTilt, une technique basée sur un vocabulaire de quatre gestes tridimensionnels permettant de s'affranchir des activateurs physiques (écran, bouton). En effet, TimeTilt propose des gestes impulsifs pouvant servir de délimiteurs gestuels. De plus nous avons proposé une application de TimeTilt qui semble coïncider avec les attentes des utilisateurs. Cette application propose deux modes d'interaction :

- Le mode lenticulaire où l'utilisateur incline son dispositif (geste fluide) pour naviguer dans une liste d'aperçus grâce à la métaphore des cartes lenticulaires. Cette interaction se déclenche par un tap à l'arrière de l'écran (geste impulsif) afin d'éviter les confusions entre les gestes fluides et les gestes de la vie courante.
- Le mode superposition où des gestes impulsifs (vers l'arrière ou vers l'avant) permettent de basculer entre des vues 'superposées' les unes sur les autres.

Il serait intéressant de poursuivre cette étude par une évaluation comparative et une évaluation en contexte de mobilité. Par exemple en comparant TimeTilt à d'autres techniques pour une tâche de navigation entre fenêtres : la technique du dispositif Palm Pre qui consiste à appuyer sur un bouton pour afficher des aperçus de fenêtres ou encore l'interface Cube que l'on trouve sur certains dispositifs HTC où les applications favorites sont disposées sur les faces d'un cube que l'on contrôle en déplaçant son doigt sur l'écran. La deuxième étude consisterait à étudier TimeTilt dans un contexte d'utilisation réel.

Enfin, bien que cette recherche n'ait pas fait l'objet d'étude systématique de satisfaction, nous avons eu de nombreux retours positifs sur la technique. Nous avons en effet présenté, à l'occasion de journées portes ouvertes de Bell Labs Alcatel-Lucent, une version de TimeTilt permettant de naviguer entre les fenêtres ouvertes d'un dispositif mobile. La technique a été appréciée par de nombreuses personnes qui l'ont jugée pratique et qui ont trouvé convaincante la métaphore des cartes lenticulaires qui donne un aspect ludique à la technique proposée.

4.3 JerkTilts

4.3.1 Présentation

JerkTilts est une technique d'interaction qui se base sur l'utilisation de gestes brusques auto-délimités du dispositif. Elle vise à permettre l'accès rapide à des commandes ou actions en utilisant les accéléromètres pour détecter les mouvements du mobile. Bien que de nombreux travaux de recherche considèrent l'utilisation des accéléromètres et plus spécialement l'inclinaison du dispositif pour interagir, les techniques proposées sont souvent basées sur du contrôle de position angulaire ou nécessitent l'activation d'un mode afin de réaliser les gestes. En revanche très peu de techniques proposent d'utiliser des gestes auto-délimités, c'est-à-dire réalisables sans déclenchement préalable d'un activateur. Sur ce point JerkTilts apporte ainsi une nette innovation.

Dans cette section nous présentons le vocabulaire gestuel de JerkTilts puis nous développons les différentes propriétés et avantages de ces gestes. Nous présentons également la méthode d'évaluation que nous avons mise en place afin de valider notre approche. Enfin, après avoir exposé plusieurs applications tirant profit de JerkTilt, nous proposerons une extension montrant comment ces gestes 3D peuvent être combinés avec des gestes sur l'écran.

4.3.2 Vocabulaire Gestuel

Description

Huit gestes différents, dans huit directions ([FIGURE 4.12](#)), obtenus par plusieurs combinaisons de *roll* (roulis) et de *pitch* (tangage) du dispositif, peuvent être réalisés par les utilisateurs et discriminés par le système. La [FIGURE 4.13](#) suivante représente le signal renvoyé par l'accéléromètre lors de la réalisation des huit gestes. JerkTilts utilise des gestes d'inclinaison rapides et de faible amplitude qui sont composés de rotations selon l'axe de *roll* et de *pitch*. Un geste de JerkTilts a pour caractéristique importante d'être un cycle complet de mouvement aller-retour : le dispositif est incliné dans une certaine direction et immédiatement ramené à sa position initiale de repos ([FIGURE 4.11](#)).

Priopriétés

Le mouvement aller-retour des JerkTilts est rendu possible par l'élasticité naturelle du poignet. Le mouvement de retour de ce geste est quasi-automatique, l'énergie mécanique stockée en tant qu'énergie élastique potentielle dans les muscles antagonistes de l'avant-bras pendant l'inclinaison initiale étant convertie en énergie cinétique pendant le retour en position de repos. Ceci aide à comprendre pourquoi l'exécution spontanée de ce type de mouvement prend très peu de temps et demande très peu d'effort [Guiard, 1993a]. De



FIGURE 4.11: Cycle complet de réalisation d'un geste de JerkTilts. Le geste part de la position initiale puis revient automatiquement en position initiale en fin de geste.

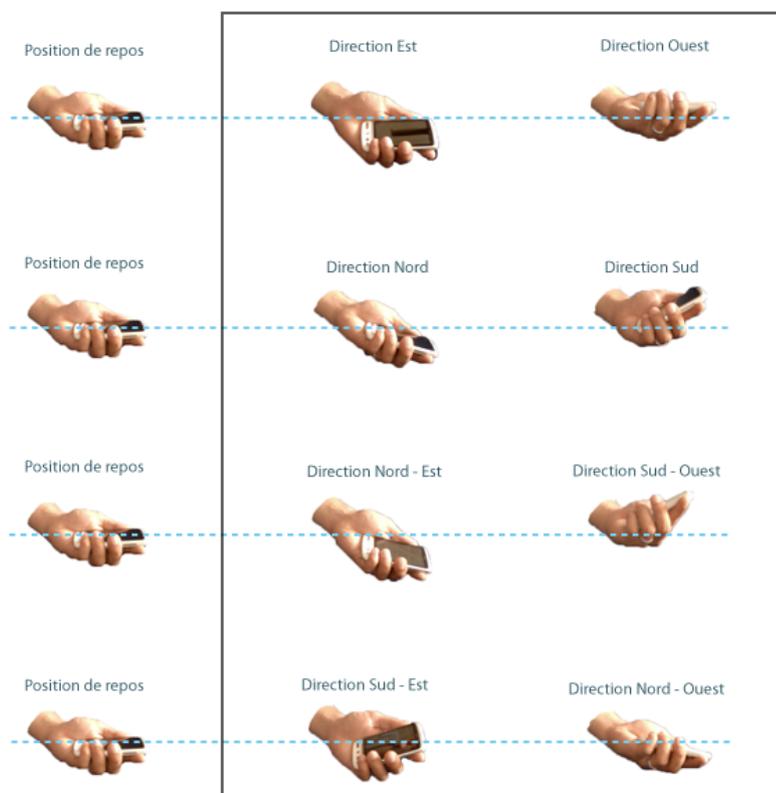


FIGURE 4.12: Les différents gestes composant le vocabulaire de JerkTilts. N'est représentée que la position maximum pour chaque geste aller-retour.

plus, ces gestes sont auto-délimités (ils ne nécessitent pas d'activateur pour les réaliser) et ne parasitent donc pas les interactions sur l'écran ni le rôle des boutons du dispositif. Ils sont respectueux du contexte de mobilité puisqu'ils ne requièrent que peu d'espace (espace autour de l'utilisateur) [Rico and Brewster, 2010b]. Ils sont réalisables à une seule main et

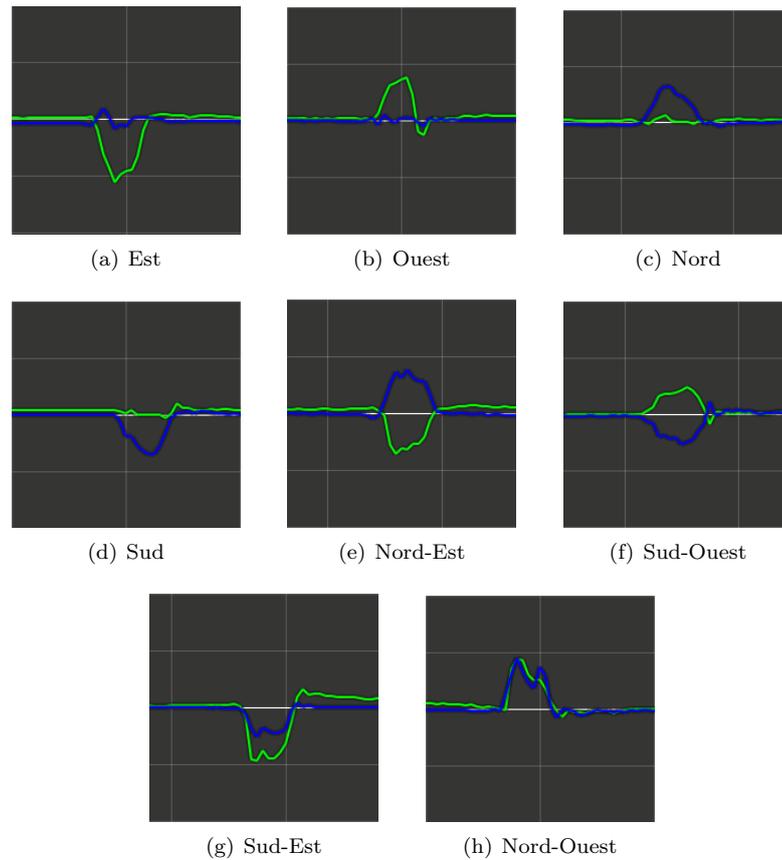


FIGURE 4.13: Courbes d'enregistrement des accélérations : la courbe verte représente les accélérations en x et la courbe bleue en y.

sont peu susceptibles d'être détectés par erreur. Enfin, ils sont réalisables sans regarder le dispositif et peuvent donc être utilisés comme des raccourcis gestuels ne requérant qu'une seule étape pour accéder à une commande.

Algorithme de reconnaissance

Nous avons utilisé un algorithme des K -plus-proches-voisins (K -ppv ou K -nn²) pour une reconnaissance automatique des gestes. Cet algorithme a l'avantage de donner de bons résultats pour un temps de calcul raisonnable sur un dispositif mobile. La valeur de K a été fixée empiriquement à 5 après quelques expérimentations. En réalité la valeur de N n'a pas un impact très important avec cet algorithme tant qu'elle satisfait aux conditions suivantes : rester plus petite que le nombre d'éléments représentant chaque classe de gestes dans la base de données et être supérieure ou égale à 3 et de préférence impaire étant donné que les K -nn sont basés sur un processus de vote.

Pour définir l'étiquette d'un geste nous nous sommes servis d'un vecteur à six dimensions. Nous avons ensuite utilisé la distance Euclidienne afin de calculer la distance qui sépare deux éléments entre eux. Les six dimensions ont été définies d'après les mesures de l'accélération lors d'un cycle complet d'aller-retour gestuel. Au moment où le dispositif quitte sa position neutre avec une accélération d'au moins 4m/s^2 , les accélérations sont enregistrées. Les six dimensions vectorielles sont définies comme dans le [TABLEAU 4.2](#) ci-dessous :

Indice	Valeur
1	Le maximum absolu d'accélération selon l'axe X
2	Le maximum absolu d'accélération selon l'axe Y
3	Le moyenne des accélérations selon l'axe X
4	Le moyenne des accélérations selon l'axe Y
5	Le médiane des accélérations selon l'axe X
6	Le médiane des accélérations selon l'axe Y

[TABLEAU 4.2](#): Les six composantes du vecteur de geste.

Ces six dimensions ont été choisies empiriquement après étude des signaux d'accélération ([FIGURE 4.13](#)) et plusieurs tests préliminaires de reconnaissance. En effet, ces tests ont montré que la simple utilisation du pic d'accélération du mouvement n'est pas suffisante. La moyenne et la médiane des valeurs enregistrées pendant la réalisation d'un geste permettent alors de résumer l'ensemble des accélérations et apportent plus d'efficacité à la reconnaissance. Nous arrivons ainsi à caractériser les gestes de manière concise et efficace.

Bien que des dimensions complémentaires puissent peut-être améliorer la reconnaissance, elles demanderaient aussi plus de temps de calcul, surtout dans le calcul des distances (l'algorithme des K -ppv compare chaque élément avec tous les autres éléments de la base de données). Notre vecteur à six dimensions nous offre un compromis raisonnable entre vitesse de calcul et précision de reconnaissance. Enfin, afin d'éviter les mauvaises détections, les gestes devaient être réalisés dans un intervalle de temps suffisamment court. Les utilisateurs doivent donc réaliser l'aller-retour des gestes de JerkTilts et revenir à leur position initiale en moins de 500 ms, le geste étant ignoré et n'ayant aucun effet dans le cas contraire.

4.3.3 Évaluation

Nous avons évalué notre technique gestuelle en trois étapes :

- 1. *Vérifier la faisabilité des gestes et construire une base de données.* Une première expérience a été réalisée afin de jauger la capacité du système et des utilisateurs à discriminer les gestes de JerkTilts. Nous avons étudié la variabilité entre individus et par individu. Les utilisateurs n'avaient pas de retour sur la reconnaissance des gestes réalisés. Cette première expérience nous a permis d'enregistrer un échantillon

2. http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

de mouvements réalisés par les participants dans les huit directions demandées afin de pouvoir les analyser. Elle nous a de plus permis de constituer une base d'apprentissage pour l'algorithme de reconnaissance utilisé dans l'expérience suivante. Nous avons en effet besoin d'une base de données suffisante pour alimenter le système de classification. Deux buts ont donc été atteints : vérifier que les utilisateurs pouvaient réaliser les JerkTilts avec succès et créer une base de données pour notre algorithme.

- 2. *Comparer notre technique à une technique existante.* Une deuxième expérience, focalisée sur l'interaction sans regarder le dispositif, compare les performances de mouvements du dispositif (JerkTilts) face aux mouvements sur le dispositif (ThumbSlides). Ces derniers sont des gestes de glissé du pouce sur l'écran similaires à ceux des Marking Menus [Kurtenbach and Buxton, 1991b], une technique connue pour son efficacité et qui fait office de référence. Contrairement à l'Expérience 1, chaque geste était ici suivi d'un retour indiquant à l'utilisateur si ce geste avait été bien reconnu. Cette situation se rapproche d'un cas réel d'utilisation où l'utilisateur sait toujours s'il a ou non réalisé l'action souhaitée.
- 3. *Tester les faux positifs dans un contexte d'utilisation réel.* Cette dernière étape visait à vérifier si des activations imprévues et non souhaitées pouvaient survenir pendant une utilisation normale du dispositif mobile. Pour cela nous avons développé un système d'enregistrement afin d'évaluer si les accélérations induites par une utilisation du dispositif pouvant engendrer la reconnaissance des gestes utilisés dans JerkTilts. Nous avons ensuite effectué cette expérience dans un contexte réel d'utilisation.

Première Expérience : Des rotations petites et rapides.

Cette première expérience avait pour but de déterminer si les utilisateurs étaient capables de mémoriser et de reproduire l'ensemble des gestes proposés. Nous voulions de plus vérifier que les utilisateurs étaient capables de réaliser les gestes avec ou sans le pouce posé sur l'écran. Cette propriété est importante puisqu'elle permet de doubler le nombre de commandes activables à l'aide de JerkTilts. La reconnaissance n'était pas faite en temps réel mais à posteriori. Un second but de cette expérience était de construire une base de données pour les expérimentations ultérieures.

Méthode

Equipement Nous avons utilisé un HTC Hero muni du système d'exploitation Android 2.1, possédant un écran tactile capacitif de 3.5 pouces de diagonale et d'une résolution de 320 par 480 pixels. Le logiciel a été développé en Java avec l'API Google Android³. Les participants étaient confortablement installés dans un fauteuil, tenant le dispositif à une

3. <http://developer.android.com>

seule main. Tous les participants étaient droitiers et avaient pour habitude d'utiliser leur propre dispositif mobile (smartphone tactile, iPhone ou Android) avec cette main.

Tâches et conditions Lors de chaque essai les participants devaient réaliser, en réponse à un stimulus visuel (FIGURE 4.14 (a)), un des huit JerkTilts en composant une combinaison de *pitch* et *roll* (par exemple un *roll* dans le sens inverse des aiguilles d'une montre produisant une supination du poignet était appelé "Ouest", et le même *roll* combiné à un *pitch* vers le haut était dénommé "Nord-Ouest"). Le stimulus visuel indiquant la direction dans laquelle réaliser le geste apparaissait sur l'écran du mobile. Un voyant gris (aussi présent sur l'écran FIGURE 4.14 (b)) permettait également aux utilisateurs de savoir si le dispositif était en position neutre, c'est à dire parallèle au sol. Les participants devaient donc garder en permanence le regard fixé sur le dispositif afin de s'assurer qu'ils démarraient bien le geste dans la position neutre.

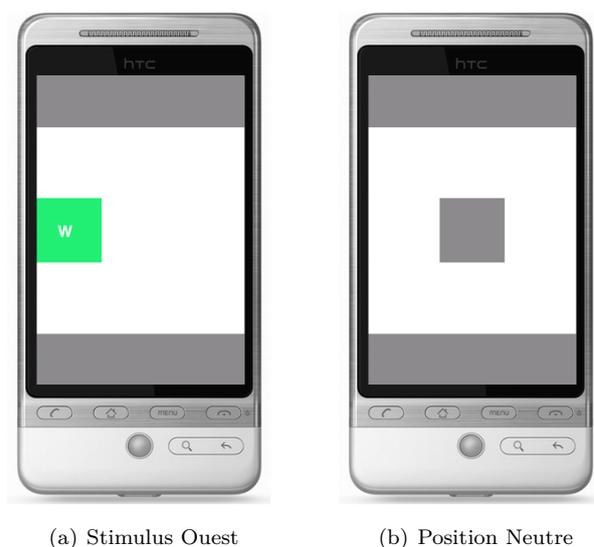


FIGURE 4.14: En vert, à gauche, on peut voir un exemple de stimulus, ici le côté Ouest (correspondant à une rotation selon l'axe de *roll* vers la gauche). En gris, sur la droite, le carré apparaissant lorsque le dispositif est en position neutre.

Les participants devaient réaliser les gestes avec ou sans la contrainte de devoir garder le pouce en contact avec l'écran du mobile. Le dispositif devait être ramené à la position neutre (parallèle au sol) après chaque geste (afin de faciliter l'homogénéité de la base de données produite). L'essai était alors validé par une dernière pression du pouce sur l'écran (ou relâchement du pouce dans la condition "avec pouce" de l'expérience), ce qui déclenchait la présentation du nouveau stimulus.

Procédure Chaque participant réalisait une session d’entraînement et pouvait l’arrêter quand il se sentait suffisamment familier avec les gestes de JerkTilts. Ensuite ils devaient réaliser quatre blocs d’expérimentation. Chaque bloc étant composé d’une séquence de trente-deux essais, chacune elle-même composée de quatre occurrences de chacune des huit directions possibles. Douze adultes volontaires (tous droitiers, dix hommes et deux femmes) ont participé à une session unique de quinze minutes chacun. Nous avons utilisé un *within-subject* design, avec *2 postures différentes * 8 directions * 4 répétitions * 4 blocs* soit 256 mouvements individuels par participant.

Algorithme de reconnaissance L’algorithme de reconnaissance a été utilisé, après l’expérience, afin de réaliser une classification automatique des gestes. Nous avons réalisé une reconnaissance que nous nommons “Reconnaissance de Groupe” et une reconnaissance que nous nommons “Reconnaissance Individuelle” :

- La “Reconnaissance de Groupe” utilise la base complète indépendamment des utilisateurs afin de classifier les gestes. De plus nous avons mis en place un système de cross-validation⁴, c’est à dire que la phase d’apprentissage de l’algorithme se sert des données de tous les individus sauf un, les données de ce dernier servant à la phase de test. Le processus est réitéré pour chaque individu et nous calculons ensuite la moyenne de tous les résultats.
- La “Reconnaissance Individuelle” utilise, pour chaque utilisateur, la base de données qui lui est propre. Ceci signifie que la reconnaissance s’adapte alors aux spécificités de chaque utilisateur. Dans cette configuration les données des blocs d’indices pairs ont été réservées pour la phase d’apprentissage, les blocs restants servant à la phase de test. La moyenne est ensuite calculée sur l’ensemble des résultats de tous les individus.

Résultats et Discussions Les taux de reconnaissance sont assez élevés, spécialement avec l’algorithme utilisant la “Reconnaissance Individuelle” (FIGURE 4.15). La posture du pouce n’a pas d’effet significatif sur les performances de réalisation des gestes ($F_{1,11} < 1$) ni sur le type de reconnaissance de l’algorithme utilisé ($F_{1,11} < 1$).

Comme attendu, les taux de reconnaissance sont meilleurs pour l’algorithme entraîné individuellement ($F_{1,11} = 33.90, p < 0.0001$), cela étant sûrement lié à la manière propre à chaque utilisateur de réaliser les gestes de JerkTilts. Ce résultat confirme la meilleure efficacité des algorithmes de reconnaissance qui s’adaptent aux utilisateurs. Un autre résultat intéressant est que le taux de reconnaissance reste aussi élevé que le pouce soit en contact de l’écran ou pas, ce qui permet d’avoir deux manières de réaliser les gestes et ainsi de doubler

4. [http://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](http://en.wikipedia.org/wiki/Cross-validation_(statistics))

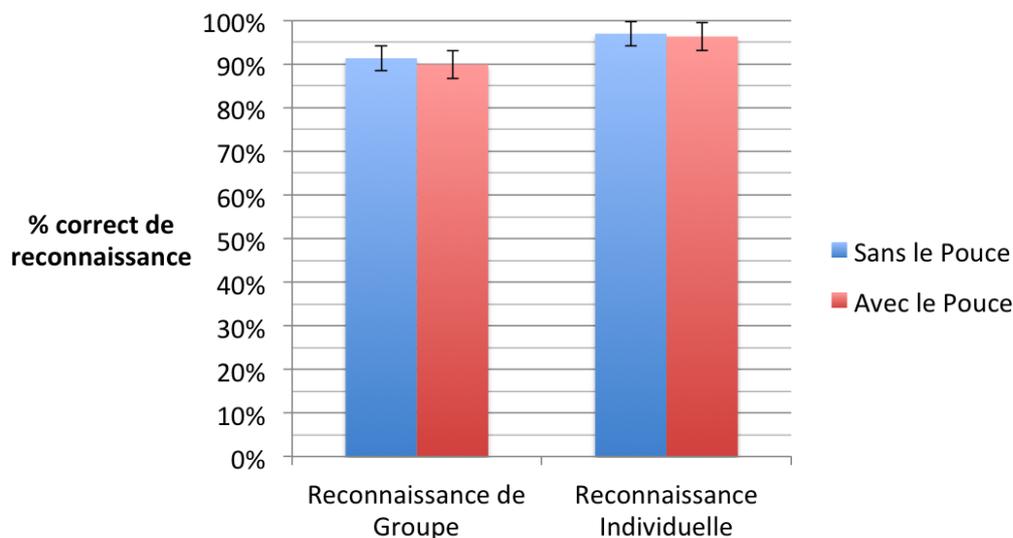


FIGURE 4.15: Les taux de reconnaissance pour les deux types de position du pouce versus les deux types de reconnaissance de l’algorithme.

le vocabulaire proposé par JerkTilts.

Deuxième Experience : JerkTilts face au Marking Menu

Cette seconde expérience a été menée dans des conditions un peu différentes de la précédente. L’idée première était de positionner JerkTilts dans des conditions se rapprochant d’un cas d’utilisation réaliste et de fait plus exigeant. Pour cela les utilisateurs choisissaient la position de départ du dispositif avec laquelle ils se sentaient le plus à l’aise. Ils n’étaient plus obligés de commencer leurs gestes avec le dispositif parallèle au sol. Ensuite nous leur avons demandé de réaliser les gestes sans regarder le dispositif. Enfin, nous voulions comparer notre technique à une technique gestuelle existante, en l’occurrence les Marking Menus sur l’écran du dispositif, afin de comparer les performances des deux types de geste, *sur* le dispositif et *avec* le dispositif, pour réaliser une sélection parmi huit cibles rangées de manière circulaire.

Un mécanisme d’auto-calibration était déclenché à chaque fois que le dispositif restait immobile pendant 200ms, quelle que soit son orientation. Un écran placé à environ 50cm, une distance usuelle de visualisation (FIGURE 4.16), à été utilisé pour permettre aux participants de réaliser les gestes sans regarder le dispositif. Cette contrainte reproduit le cas d’utilisation où un utilisateur interagit avec une application, par exemple un lecteur de musique, sans regarder son dispositif. Le participant avait un retour immédiat (après chaque geste) de la classification du mouvement qu’il venait de réaliser (là encore pour approcher une situation réaliste, l’utilisateur ayant toujours un retour dans la réalité). Par contre il

n’y avait plus de signal indiquant l’orientation du dispositif (le carré gris de la première expérience).

Les performances de sélection des JerkTilts ont été évaluées relativement aux performances des gestes du pouce sur l’écran des Marking Menus. L’utilisateur devait réaliser un geste de tracé depuis n’importe quelle position sur l’écran dans la direction demandée. Reconnus pour leur efficacité, les Marking Menus ont ici servi de technique de référence.

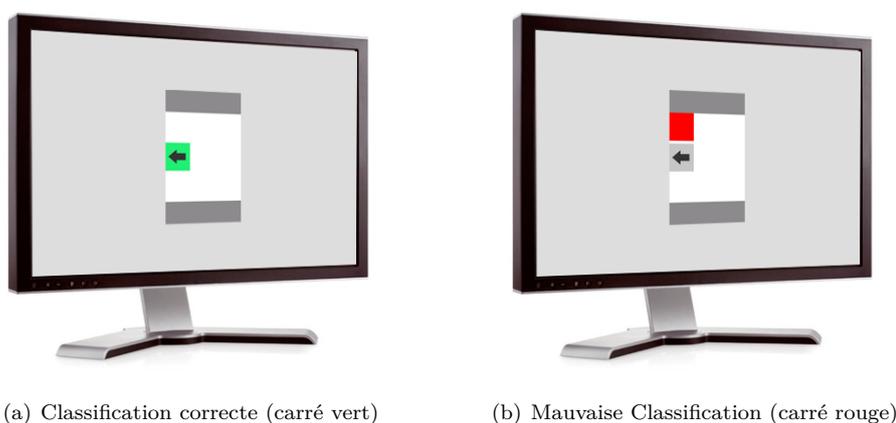


FIGURE 4.16: Exemple de retour d’information renvoyé par le système pour un geste bien réalisé (a) ou raté (b)

Méthode

Equipement Nous avons également utilisé un HTC Hero muni du système d’exploitation Android 2.1, possédant un écran tactile capacitif de 3.5 pouces de diagonale et d’une résolution de 320 par 480 pixels. Le logiciel a été développé en Java avec l’API Google Android⁵. Les participants étaient confortablement installés dans un fauteuil, tenant le dispositif à une seule main. Tous les participants étaient droitiers et avaient pour habitude d’utiliser leur propre dispositif mobile (smartphone tactile, iPhone ou Android) avec cette main. La sortie visuelle se faisait sur un écran (FIGURE 4.16) d’une diagonale de 22 pouces placé à 50cm du participant et dont le logiciel avait été développé en Java. Le stimulus était représenté par une flèche apparaissant dans un carré dont la couleur variait. Le carré était initialement affiché en gris et devenait vert si la sélection était correcte (FIGURE 4.16 (a)). A la suite d’une erreur le carré restait gris et la direction reconnue était montrée à l’utilisateur sous forme d’un carré rouge (FIGURE 4.16 (b)). Le geste devait alors être fait à nouveau. La main des participants tenant le dispositif ainsi que le dispositif n’étaient pas visible.

5. <http://developer.android.com>

Procédure Douze adultes volontaires (tous droitiers, onze hommes et une femme) ont participé à notre expérience lors d’une session de vingt-cinq minutes. Un peu d’entraînement précédait chaque session. Une session se composait de dix blocs de trente-deux essais. Chaque geste était répété quatre fois, de manière aléatoire, au cours d’un bloc. Chaque participant tenait le dispositif dans la main droite.

L’inspection des données ayant révélé que les performances nécessitaient plusieurs blocs avant de se stabiliser, nous avons décidé d’éliminer les trois premiers blocs et de les considérer comme un échauffement. Notre *within-subject* design impliquait $2\text{techniques} * 8\text{directions} * 4\text{répétitions} * 7\text{blocs}$ pour un total de 448 gestes par participant.

Algorithme de reconnaissance L’algorithme de reconnaissance utilisait la base de données construite à partir de l’expérience 1 comme base d’apprentissage. Cependant il a été modifié pour fonctionner en mode adaptatif, une base d’apprentissage personnelle étant créée pour chaque utilisateur. Cette nouvelle base d’apprentissage, initialement construite à partir de tous les gestes réalisés durant la première expérience (moins quelques cas trop distants du reste des données), est donc enrichie de ces nouveaux gestes. Chaque fois qu’un utilisateur réalise un geste validé, ce dernier est ajouté à la base de donnée. Cet algorithme reste simple et permet au système de reconnaissance de s’adapter aux différences gestuelles spécifiques à chaque utilisateur. De plus il fait sens d’utiliser un tel algorithme dans le contexte de mobilité : les smartphones sont des dispositifs essentiellement personnels, cela n’empêche pas d’autres utilisateurs de s’en servir, mais améliore les performances du propriétaire.

Nous avons également ajouté deux autres améliorations à cet algorithme :

- La première consiste à arrêter d’ajouter des nouveaux éléments à la base d’apprentissage lorsque les taux de reconnaissances sont suffisants, ceci afin de ne pas augmenter de manière excessive le nombre d’échantillons présents pour préserver le temps de calcul nécessaire.
- La seconde est basée sur le fait que les gestes dans les direction Sud-Est et Nord-Est sont plus difficiles à réaliser que les autres. L’algorithme précédent ne fonctionne pas très bien dans cette situation : les gestes dits ‘difficiles’ ont tendance à être mal reconnus (notamment lorsque la base d’apprentissage n’a pas encore été personnalisée). Peu d’échantillons sont alors ajoutés dans la base personnalisée, impliquant une mauvaise adaptation pour ces gestes. Ceci a pour conséquence de construire une base de données déséquilibrée, ne contenant qu’un faible pourcentage d’échantillons représentatifs de ces gestes ‘difficiles’. L’algorithme adaptatif réduit alors les performance de ces gestes au lieu d’améliorer leur reconnaissance. Pour répondre à ce problème, nous avons utilisé une approche heuristique consistant à ajouter certains gestes mal identifiés mais restant néanmoins suffisamment proches du geste demandé (la proximité du geste cor-

respondant ici au nombre de votes de l'algorithme K -ppv). En d'autres termes, nous forçons l'algorithme à considérer certains gestes invalides comme valides : cette forme de *bootstrapping* favorise l'apprentissage de la manière personnelle avec laquelle l'utilisateur réalise les gestes 'difficiles'.

Enfin nous avons également développé un mécanisme d'auto-calibration permettant aux utilisateurs de réaliser les gestes à partir de n'importe quelle position initiale de la main choisie arbitrairement. Ce mécanisme consiste à remettre les valeurs d'accélération à la position neutre après être resté immobile pendant 200ms. Nous n'avons donc pas demandé aux utilisateurs de tenir leur dispositif dans une position particulière ou de revenir à une position spéciale dans cette deuxième expérience (ce mécanisme n'a pas été utilisé lors de la première expérience afin de récolter les gestes bruts).

Résultats et Discussion Les variables indépendantes de cette expérience sont : le taux de reconnaissance et le temps total (TT) de sélection. Le temps total de sélection représente la somme du temps de réaction (RT) et du temps de mouvement (MT). Nous avons négligé le temps de calcul pris par l'algorithme de reconnaissance, qui n'a jamais dépassé 50ms.

Concernant les pourcentages de précision, en moyenne le taux de reconnaissance (FIGURE 4.17) est proche de 91.0% pour les JerkTilts et de 92.6% pour les gestes du pouce ($F_{1,11} < 1$). Le seul effet significatif pour cette mesure concerne la direction du mouvement ($F_{7,77} < 2.25, p < 0.04$), un effet commun aux deux techniques (pour l'interaction, $F_{7,77} < 1.41, p < 0.2$). L'effet sur la direction est consistant mais assez faible dans l'absolu, le taux moyen de reconnaissance étant de 92.0% ce qui est tout à fait convenable.

Les performances en rapidité révèlent des valeurs proche de 936ms et 707ms en moyenne pour réaliser un JerkTilts et un glissé du pouce respectivement depuis le stimulus ($f_{1,11} = 8.19, p < 0.02$), comme illustré sur la FIGURE 4.18. La moyenne pour le temps de mouvement était plus courte pour notre technique (177.2ms contre 211.3ms) mais pas de manière significative ($p > 0.05$). Le temps de réaction est à l'opposé plus court pour les glissés du pouce (694.8ms contre 494.9ms) et ce de manière significative ($F_{1,11} = 19.8, p < 0.001$).

Nous avons également trouvé une différence significative sur l'interaction technique par direction pour le temps total ($F_{7,77} = 2.44, p < 0.03$) (FIGURE 4.18) : sans surprise, la direction la plus rapide n'est pas la même pour les deux techniques. Par exemple, la direction Sud est beaucoup plus favorable pour les glissés du pouce qu'elle ne l'est pour les JerkTilts. En effet une supination du poignet est plus difficile à réaliser que le glissement du pouce vers le bas de l'écran.

Les performances générales des deux techniques sont bonnes dans l'ensemble, bien que

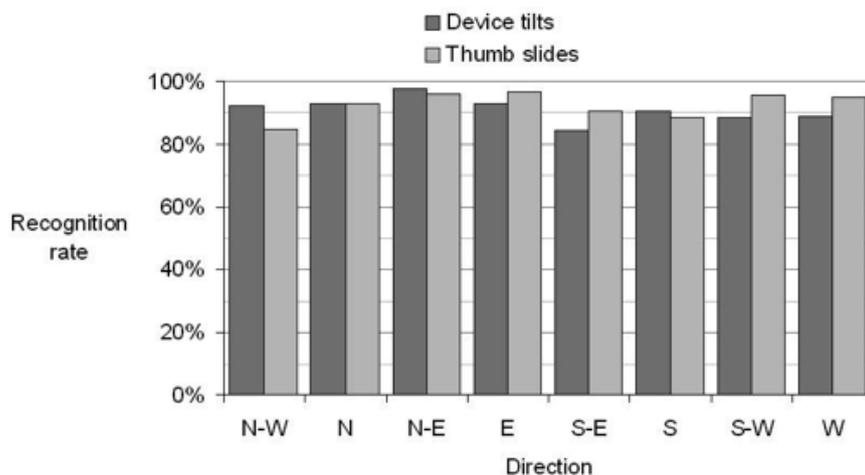


FIGURE 4.17: Les taux de reconnaissance des deux techniques en fonction des huit directions cibles.

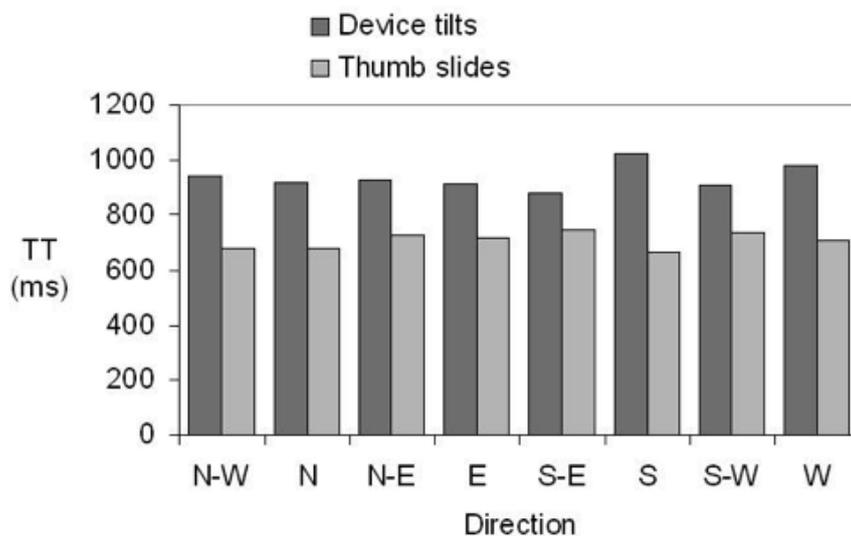


FIGURE 4.18: Les temps totaux (TT) moyens des deux techniques en fonction des huit directions cibles.

relativement éloignées des taux de reconnaissance habituels pour les Marking Menus ou de l'expérience 1 pour les JerkTilts. Ceci peut s'expliquer par les conditions même de l'expérience. En effet la condition *eyes-free* dégrade les performances des deux techniques. Ensuite, comme le décrit Francone dans [Francone et al., 2010], les Marking Menus perdent en efficacité lorsque l'utilisateur interagit à une seule main, ceci étant dû aux limitations biomécaniques des mouvements du pouce dans de telles conditions.

Les performances temporelles des JerkTilts sont inférieures à celles du Marking Menu dans cette expérience. Cependant il faut noter que les conditions du Marking Menu sont particulièrement favorables. Ces gestes pouvaient en effet être démarrés depuis n'importe quelle position sur l'écran et ce sans nécessiter l'activation d'un mode au préalable. Dans un cas d'utilisation réel il peut être nécessaire soit de débiter l'interaction dans une zone spécifique de l'écran, une contrainte à laquelle les JerkTilts échappent, soit de presser préalablement un bouton pour activer un mode adéquat. Les gestes de glissé sont en effet la plupart du temps déjà utilisés dans les interfaces des applications mobiles (typiquement pour le défilement). Un mode est donc nécessaire pour éviter les interférences avec les interactions usuelles. Il est à noter que ce problème peut être résolu sur un PC via l'utilisation des boutons de la souris ou des 'modificateurs' du clavier (touches contrôle, shift, etc.) Les écrans tactiles passifs des mobiles (qui ne permettent pas différents états d'interaction [Bragdon et al., 2011]) et l'absence de clavier physique interdisent de recourir à ce type de solution.

Grâce au fait qu'ils sont auto-délimités, les JerkTilts permettent de s'affranchir des activateurs. Ils ne nécessitent pas non plus de regarder où l'on fait le geste sur l'écran. Ainsi, bien qu'un peu plus lents dans cette expérience, les JerkTilts pourraient finalement être un peu plus rapides que les Marking Menus dans des conditions d'utilisation réalistes, notamment pour l'activation de raccourcis (par exemple pour lancer une application ou rapidement passer d'une chanson à une autre).

Troisième Expérience : enregistrement en conditions réelles d'utilisation.

Cette dernière expérience visait à enregistrer de manière permanente toutes les accélérations d'un dispositif mobile afin d'évaluer l'utilisabilité des JerkTilts dans un contexte de mobilité de la vie réelle. Même si les gestes de JerkTilts sont réalisables nous voulions être sûr qu'ils ne compromettent pas les autres ressources d'entrée du dispositif. Notre intérêt principal dans cette dernière expérience était d'étudier et d'analyser la fréquence des déclenchements inopinés, c'est à dire la mauvaise interprétation d'accélérations comme étant des JerkTilts.

Méthode Nous avons donc conçu et développé un système d'enregistrement. Ce système devait enregistrer toutes les accélérations intervenues pendant une période de vingt-quatre heures d'utilisation standard du dispositif. Le programme enregistrait lorsque le dispositif mobile était en mode "actif", c'est à dire qu'il n'était ni en veille ni éteint. C'est pourquoi, sur la période de vingt-quatre heures, les temps d'enregistrement ont varié en fonction des utilisateurs. Nous avons envoyé un courriel afin de recruter des volontaires, y joignant le programme d'installation de l'application et les instructions à suivre dans un document PDF (Annexe 5.3.3). Quatorze personnes ont accepté de participer à notre expérience (douze utilisant leur propre téléphone et deux utilisant un téléphone que nous leur avons prêté).

L'application enregistrait chaque accélération (selon les axes x , y et z) du mobile pendant une période de vingt-quatre heures et proposait ensuite à l'utilisateur d'envoyer le fichier de données par FTP. Le module d'envoi était directement intégré à l'application afin de faciliter le transfert. Afin de préserver l'anonymat des participants, il leur était proposé de demander la destruction de leurs données une fois analysées. Aucun des participants n'a souhaité recourir à cette possibilité.

	Moyenne	Intervalle de confiance de 95%
Nombre de détections	0.7	0.02 — 1.4
Temps d'activité(en min)	665	340 — 992

TABLEAU 4.3: Nombre de détections d'activations d'un geste et temps moyen de l'enregistrement.

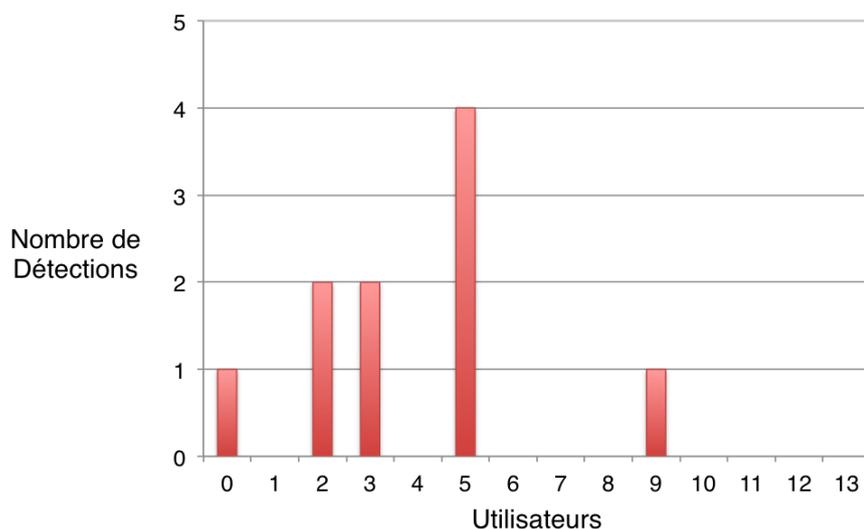


FIGURE 4.19: Nombre d'activations involontaires par utilisateur.

Résultats et Discussion Le système n'a détecté en moyenne que 0.7 (TABLEAU 4.3, FIGURE 4.19) gestes de JerkTilts. Cela pour un temps d'enregistrement moyen de 665 minutes (TABLEAU 4.3). C'est-à-dire que les mouvements liés à l'utilisation normale d'un dispositif mobile n'ont que peu de chance d'être interprétés comme un JerkTilt. Ce résultat confirme que les gestes de JerkTilts sont applicables à un contexte d'utilisation courante. Nous ne sommes pas surpris par ce résultat, puisque, comme nous avons pu le voir précédemment la nature même des mouvements engendrés par ces gestes se traduit par un signal des accélérations qui est tout à fait caractéristique.

Il serait également intéressant de poursuivre cette expérimentation en réalisant une étude sur le long terme afin de voir comment les utilisateurs sont capables de réaliser et d'utili-

ser JerkTilts dans la vie de tous les jours. Ainsi nous aurions une étude plus focalisée sur les conditions de mobilité comme par exemple dans les transport en commun où diverses contraintes altèrent le degré d'attention et l'espace gestuel disponible.

4.3.4 Application

Plusieurs applications peuvent tirer profit de JerkTilts. Il est en effet possible de désigner des interactions gestuelles en trois-dimensions associées à des commandes sans interférer avec les interactions standard sur l'écran ou encore avec les menus ou widgets. De plus les gestes permettent d'activer des commandes facilement sans regarder le dispositif. Le concept de proprioception lui-même encourage les interactions gestuelles à une seule main, les utilisateurs étant conscient de la localisation spatiale et de la posture de leur propre main, et également de la localisation et de l'orientation du dispositif qu'ils tiennent. Ces gestes ne demandent pas non plus une charge cognitive trop élevée.

Nous avons donc conçu et développé trois variantes d'exploitation possibles de JerkTilts. La première, JerkTilts Windows, se situe au niveau *système* et dépend de la combinaison des gestes 3D et du contact du pouce afin de minimiser les risques d'activations involontaires. Les deux autres, JerkTilts Copy&Paste et JerkTilts Music Remote, se situent au niveau des *applications*.

JerkTilts Window

JerkTilts Window est une technique qui permet de naviguer rapidement entre différentes fenêtres sur un mobile. Un peu à la manière de TimeTilt qui a été présenté précédemment, l'utilisateur va pouvoir passer d'une application à une autre. Cependant JerkTilts permet d'atteindre plus rapidement un plus grand nombre de fenêtres que TimeTilt. Nous avons développé deux versions, une permettant de naviguer entre les applications ouvertes et une autre permettant de lancer ou d'atteindre un ensemble d'applications favorites. Avec ces implémentations l'utilisateur peut facilement passer d'une application à une autre avec un seul geste comme le montre la [FIGURE 4.20](#). Nous avons choisi de minimiser le risque d'activations involontaires en demandant l'appui du pouce sur l'écran lors de l'exécution d'un JerkTilt.

Cette application réduit la séquence habituelle d'actions nécessaires à un utilisateur sur les mobiles du commerce pour passer d'une application à une autre ([FIGURE 4.3](#)). Jusqu'à présent dans la littérature, passer d'une application à une autre nécessitait au moins deux actions : cliquer sur un bouton (souvent avec un délai ou un double-clic) puis sélectionner l'application souhaitée. Pour ouvrir une application favorite il faut généralement retourner à l'écran d'accueil et ensuite cliquer sur l'icône correspondante. Similairement, JerkTilts peut également servir à atteindre des favoris préalablement enregistrés dans un navigateur inter-



FIGURE 4.20: JerkTilts Window

net. Ceci permettrait d'éviter aux utilisateurs d'avoir à ouvrir et parcourir un ou plusieurs menus, une opération peu pratique sur un terminal de petite taille.

JerkTilts Copy&Paste

Nous avons également développé une version de JerkTilts qui permet de facilement copier/coller des objets (images, textes, etc.) entre deux fenêtres d'applications différentes en une séquence de seulement trois actions : copier l'objet (FIGURE 4.21 (a)), changer d'application (FIGURE 4.21 (b)) et enfin coller l'objet (FIGURE 4.21 (c)). Par exemple comme dans la FIGURE 4.21, on peut facilement copier une image depuis une page internet vers une autre application. La même logique s'applique pour une séquence de couper/coller avec un geste différent pour l'étape de couper. Cette application de JerkTilts réduit grandement le nombre de manipulations et d'actions nécessaires à ce type d'interaction en comparaison avec les systèmes usuels des smartphones courants, qui nécessitent habituellement des délais et/ou des pointages multiples.

JerkTilts Music Remote

Nous présentons ici une application utilisant les gestes de JerkTilts et notamment leur capacité à être réalisés sans regarder le dispositif. Nous avons développé deux versions de cette technique pour contrôler un lecteur de musique. La première version rend possible le contrôle d'un lecteur sur ordinateur personnel, le dispositif mobile jouant alors le rôle d'une télécommande. Dans la seconde version le lecteur de musique contrôlé est sur le dispositif mobile lui-même, et présente l'avantage de permettre une interaction sans déverrouiller le dispositif. Dans chacune des versions les gestes de JerkTilts ont été associés aux commandes du lecteur comme le montre la FIGURE 4.22 (a). La disposition des commandes reste consistante avec celle des lecteurs musicaux standards. Nous avons également placé les commandes



(a) Copie de l'image par un JerkTilts vers le Sud et le pouce posé sur l'image

(b) Changement d'application par un JerkTilts vers l'Est.



(c) Collage de l'image par un JerkTilts vers le Sud-Ouest.

(d) L'image est collée sur la position désignée par le contact du pouce au moment du geste

FIGURE 4.21: Un exemple de séquence permettant de réaliser un copier/coller d'une image à partir d'une page web vers une autre application.

les plus utilisées sur les quatre directions cardinales : les gestes demandés sont en effet plus faciles à réaliser.

La version de la technique que nous avons développée permet aux utilisateurs d'avoir un contrôle distant sur un lecteur de musique (fonctionnel avec iTunes et Spotify par exemple) grâce à AppleScript (**FIGURE 4.22 (b)**). L'application peut facilement être adaptée pour fonctionner avec un lecteur vidéo (nous avons développé un prototype fonctionnant avec QuickTime).



(a) Association des directions aux commandes du lecteur de musique.

(b) Contrôle de iTunes avec JerkTilts

FIGURE 4.22: JerkTilts Music Remote

4.3.5 Extension de JerkTilts : Combinaison 3D/2D

Nous nous sommes également intéressé à la combinaison de JerkTilts avec les glissés du pouce (ThumbSlides) issus du Marking Menu de l'expérience 2. L'association de ces deux types de geste permet d'utiliser des ressources complémentaires. Elle peut par exemple permettre d'accéder rapidement à des commandes en utilisant JerkTilts, tout en contrôlant un paramètre continu de la commande via un geste du pouce. Les gestes du pouce peuvent être très efficaces dans le contrôle de variables continues [Pook et al., 2000b]. Cette combinaison pourrait éventuellement permettre un parallélisme entre les deux types de geste. Nous avons conçu un menu permettant d'accéder à différentes applications et à leurs fonctionnalités présentes sur les dispositifs mobiles actuels. Nous avons ensuite effectué une expérimentation afin d'évaluer ce menu. Après une étude informelle, nous avons réduit de huit à quatre le nombre de directions présentes au niveau des JerkTilts afin que l'interaction soit plus aisée. Nous avons gardé les directions Nord, Sud, Est, Ouest. L'utilisateur n'a pas besoin de déverrouiller le téléphone mobile, puisque les JerkTilts sont auto-délimités et que le système de détection des accélérations fonctionne même lorsque le dispositif est verrouillé.

Réalisation

Le système de menu comprend deux niveaux d'interaction : le premier pour choisir l'application et le second la fonctionnalité de cette application. Comme indiqué précédemment le premier niveau est contrôlé en effectuant un JerkTilt et le second niveau un glissé du pouce. Ce système a été testé au moyen d'un prototype avec 4 applications courantes (Paramètres, Musique, Téléphone, Mails) et qui illustrent les possibilités offertes par la combinaison de gestes 3D et de gestes 2D sur l'écran.

Les gestes sont réalisables aussi bien en mode novice qu'intermédiaire ou expert. En

effet la plupart des menus proposent plusieurs modes d'interaction [Bailly, 2009]⁶. Le mode novice s'affiche généralement à l'activation du menu et permet de choisir visuellement l'item que l'on souhaite. Le mode expert permet d'accéder directement à une commande sans ouvrir le menu, par exemple lorsque l'on utilise le raccourci clavier *CTRL + C* pour accéder à la commande copier. Enfin le mode intermédiaire permet de passer d'un mode à l'autre. Nous présentons d'abord le mode novice. Le mode novice est activé en maintenant le pouce appuyé sur l'écran. Le mode intermédiaire s'active si l'utilisateur réalise un JerkTilts puis attend un délai : les fonctionnalités de l'application sélectionnée apparaissent alors. Le mode expert consiste en l'enchaînement sans délai d'un JerkTilt suivis d'un ThumbSlide.

Niveau application L'utilisateur appuie sur l'écran pendant un certain délai, permettant au menu d'apparaître. Les 4 directions cardinales (Nord, Sud, Est, Ouest) permettent alors d'accéder à 4 applications (Paramètre, Musique, Téléphone, Mails) comme le montre la [FIGURE 4.23](#). L'utilisateur réalise un JerkTilt pour sélectionner l'application désirée.

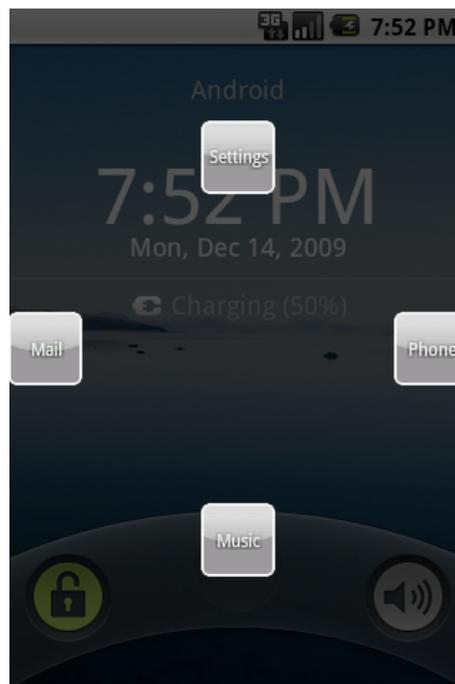


FIGURE 4.23: Niveau Applications du menu

Niveau Fonctionnalité Une fois le JerkTilt effectué l'utilisateur peut accéder aux différentes fonctionnalités de l'application choisie. Il peut alors sélectionner la commande à exécuter grâce à un glissé du doigt ou à un pointage sur le bouton adéquat (cette seconde

6. MenUA. <http://www.gillesbailly.fr/menua/>

manière d'interagir n'étant possible qu'en mode novice). Afin de faciliter le passage du mode novice au mode expert, les commandes ont été placées de manière à permettre à la fois d'appuyer directement ou de réaliser un geste dans la direction choisie.

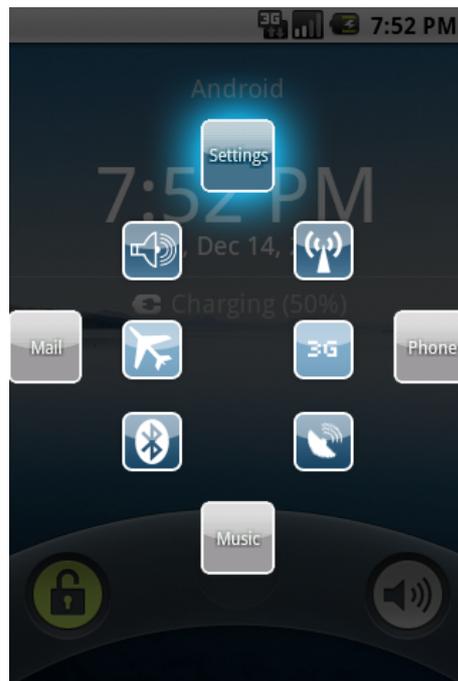
La technique proposée possède plusieurs propriétés intéressantes qui sont illustrées dans les exemples suivants. Dans le premier exemple (lecteur de musique [FIGURE 4.24 \(c\)](#)) le contrôle du volume se fait au moyen d'un geste qui permet un contrôle continu, comme avec les Control Menus [Pook et al., 2000a, Pook et al., 2000b]. Le principe des Control Menus est le suivant : l'utilisateur réalise un geste et lorsqu'il dépasse une certaine distance, la commande est activée et il peut alors la contrôler de manière continue en poursuivant son geste. Le second exemple (application de mail [FIGURE 4.24 \(d\)](#)) montre que les ThumbSlides peuvent être directement associés aux gestes déjà présents dans l'application. Dans ce cas ils sont confondus avec les gestes de pan de la liste de messages

Evaluation

Afin de vérifier l'efficacité de notre technique nous avons mené une expérience visant à tester les performances lors de la combinaison des deux types de geste. L'expérience a été réalisée en mode expert car cela permet d'une part de tester l'efficacité des gestes de la technique et d'autre part de tester une interaction sans regarder le dispositif. Pour cette expérience deux sessions ont été réalisées avec les mêmes utilisateurs. La première session a permis d'adapter l'algorithme d'apprentissage aux particularités des utilisateurs. La seconde nous a permis d'étudier les performances de notre technique.

Méthode Les conditions de cette expérience étaient les mêmes que celles de l'expérience 2 vue précédemment (même terminal mobile utilisé et affichage sur écran distant) sauf que les stimuli ne sont plus les mêmes. Ils présentent désormais les deux directions simultanément comme sur la [FIGURE 4.25](#) via une grosse flèche jaune pour le JerkTilt à réaliser et une petite flèche bleue pour le ThumbSlide. Lorsque l'utilisateur réussit les deux gestes l'écran devient vert et passe au stimulus suivant après un délai de 1 seconde. Si l'un des deux gestes n'est pas réussi, l'écran devient rouge et affiche la combinaison de gestes reconnue par le système pendant 1s puis passe au stimulus suivant.

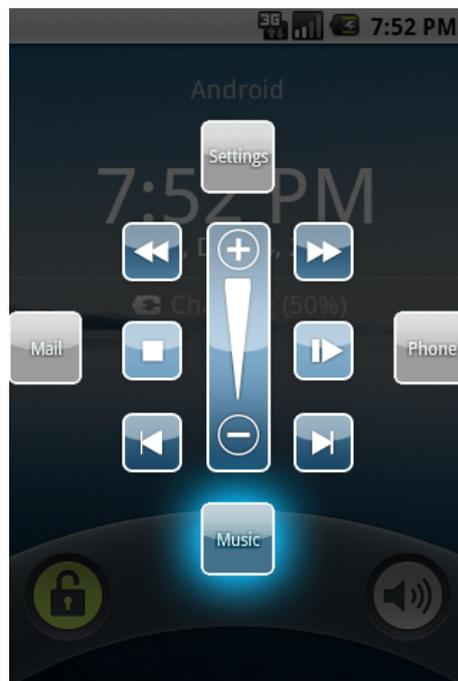
Procédure Six adultes volontaires (tous droitiers, six hommes) ont participé à notre expérience lors de deux sessions. Une première session d'entraînement a permis de calibrer l'algorithme et de familiariser les participants avec la technique. Les deux sessions étaient espacées d'au moins une heure afin de ne pas trop fatiguer les participants. Chaque session se composait de cinq blocs de soixante-quatre essais. Chaque combinaison était répétée deux fois de manière aléatoire au cours d'un bloc. Chaque participant tenait le dispositif dans la main droite. En résumé, notre *within-subject* design impliquait $2 \text{ techniques} * 4 \text{ directions}$



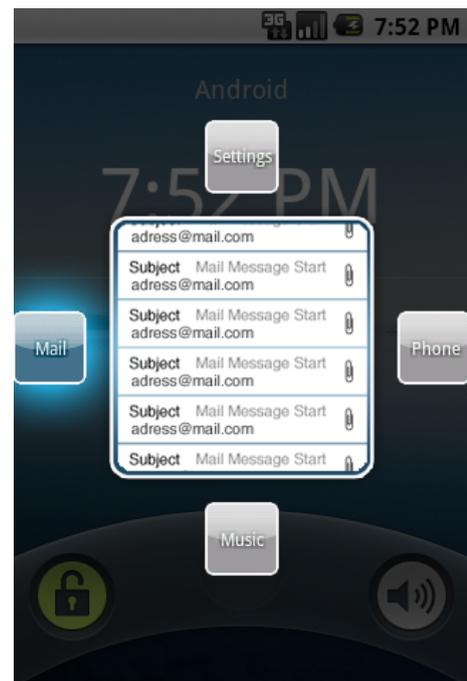
(a) Paramètres du téléphone



(b) Favoris Téléphone



(c) Lecteur de musique



(d) Mails Récents

FIGURE 4.24: Niveau Fonctionnalités du menu.

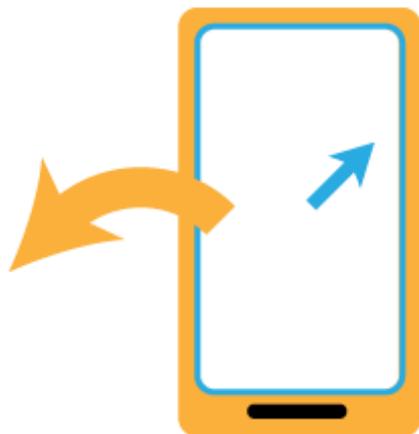


FIGURE 4.25: Exemple de stimulus de combinaison JerkTilt(flèche jaune)/ThumbSlide(flèche bleue).

*3D * 8 directions 2D * 2 répétitions * 4 blocs* pour un total de 256 sélections par participant.

Résultats et discussion Les résultats de cette expérience sont de 98,25% de taux de bonne reconnaissance pour les JerkTilts sur 4 directions et de 95,25% pour les ThumbSlides avec 8 directions. Ces résultats sont meilleurs que pour l'expérience 2 pour plusieurs raisons. Premièrement il n'y a plus que 4 directions pour les JerkTilts ce qui facilite leur réalisation et améliore les performances de la classification. Ensuite l'entraînement était plus long que pour l'expérience 2 ce qui a permis aux utilisateurs de mieux réaliser les gestes et à l'algorithme de mieux s'adapter à l'utilisateur. Le taux de réussite pour la combinaisons des gestes s'élève à 93,75% (FIGURE 4.26) ce qui montre que la technique est bien assimilée par les utilisateurs après une phase de calibrage et d'apprentissage.

Nous avons ensuite étudié les temps totaux pour exécuter une commande. En moyenne il faut 1525 milli-secondes pour exécuter la combinaison des deux gestes (temps de réaction d'en moyenne 741ms). Nous avons cherché à savoir s'il y avait des différences de temps en fonction des directions de la combinaison. Nous avons trouvé qu'il n'y a pas d'effet en fonction de la direction des ThumbSlides ($F_{7,35} = 0.55, p < 0.7926$) mais qu'il y en a un en

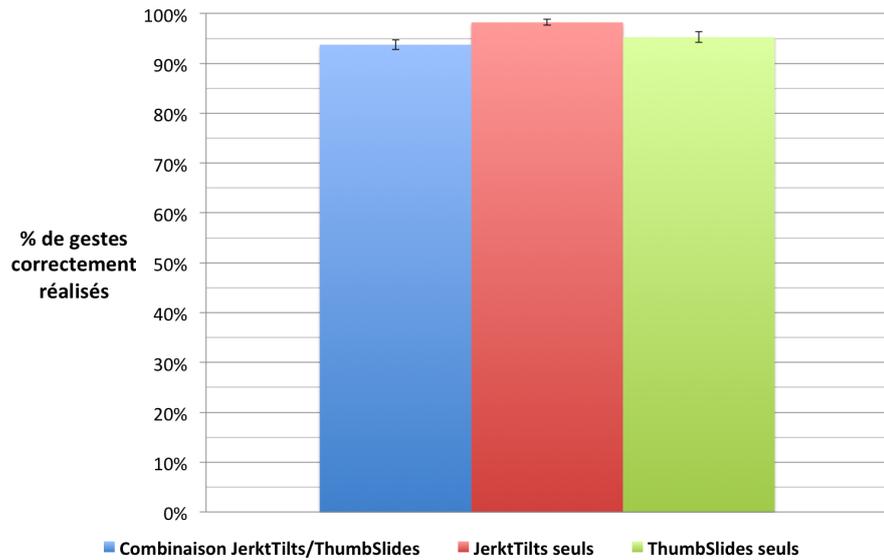


FIGURE 4.26: Taux de reconnaissance de la combinaison JerkTilts/ThumbSlide (les barres correspondent au 95% d'intervalle de confiance).

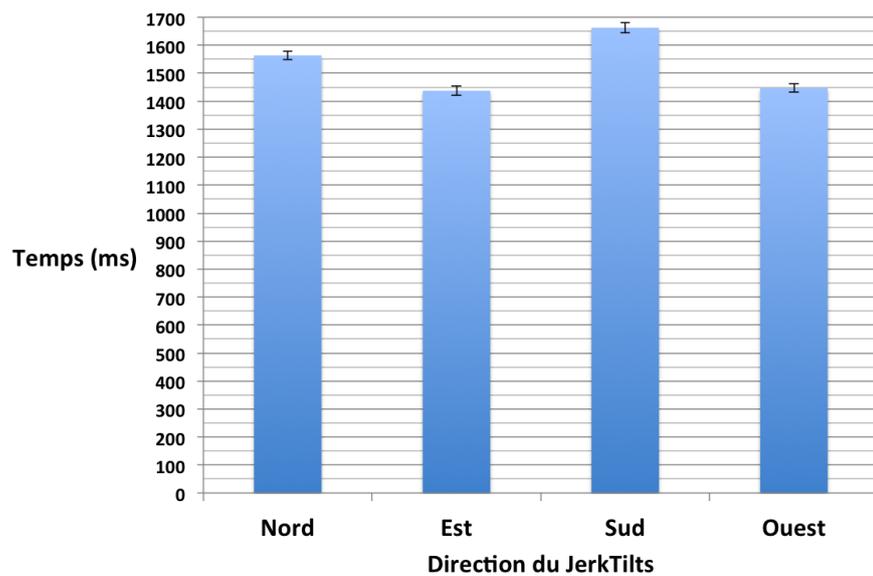


FIGURE 4.27: Temps moyen de la combinaison JerkTilts/ThumbSlide en fonction des 4 directions cardinales (les barres correspondent au 95% d'intervalle de confiance).

fonction de celle des JerkTilts comme le montre la FIGURE 4.27 ($F_{7,15} = 28.69, p < 0.0001$). La direction Sud est significativement plus lente que les trois autres et les directions Sud et Nord sont plus lentes que les directions Ouest et Est. Ces résultats sont encourageants, car ils montrent que la combinaison des deux types de geste peut facilement être réalisée et

utilisée par les utilisateurs. De plus les taux de gestes réussis sont élevés ce qui montre la robustesse de la technique même avec un entraînement limité.

4.3.6 Conclusion

Les gestes de JerkTilts sont des mouvements d’aller-retour rapides combinant des *rolls* et des *pitch*. Ils reposent sur une demi-oscillation rendue possible par l’élasticité naturelle du poignet et peuvent être réalisés sans regarder. Ces gestes ne nécessitent pas le positionnement de l’appareil dans une orientation spécifique avant leur commencement. Étant auto-délimités il évitent d’avoir à recourir à un changement de mode nécessitant un bouton physique ou virtuel. Ils n’interfèrent pas avec les interactions courantes déjà présentes (qui sont basées sur l’écran tactile) et peuvent donc servir de raccourcis gestuels pour la plupart des applications existantes (la principale incompatibilité étant avec les jeux ou rares interfaces ayant déjà recours aux gestes 3D). Enfin les JerkTilts ne nécessitent que peu d’espace du fait que ces mouvements sont essentiellement des rotations stationnaires du dispositif mobile. De ce fait, les gestes utilisés sont discrets (faible amplitude et temps de réalisation) et sont donc appropriés à la plupart des contextes sociaux. Grâce à une signature d’accélération particulière, ces gestes ont peu de chance d’être activés par erreur.

Il serait souhaitable de poursuivre ce travail par une étude de terrain plus approfondie visant à vérifier que les utilisateurs sont capables de réaliser des JerkTilts dans n’importe quelle situation de mobilité, par exemple lorsqu’ils sont dans les transports en commun. Il serait de plus intéressant d’ajouter un capteur de mouvement tel qu’un gyroscope pour essayer d’améliorer la reconnaissance des gestes de JerkTilts.

Enfin, nous avons vu les possibilités offertes par la combinaison des gestes de JerkTilts et des ThumbSlides pour augmenter le vocabulaire gestuel initial. JerkTilts permet un accès rapide aux applications et les gestes du pouce sont très efficaces pour en contrôler les fonctionnalités.

5

Conclusion

Sommaire

5.1 Conclusion	134
5.2 Contributions	134
5.2.1 Espace de classification	134
5.2.2 Mouvements sur le dispositif	135
5.2.3 Mouvements du dispositif	136
5.3 Perspectives	137
5.3.1 Espace de Classification	137
5.3.2 Mouvements sur le Dispositif	138
5.3.3 Mouvements du Dispositif	138

5.1 Conclusion

On compte aujourd'hui plus de 65 millions¹ de téléphones mobiles en France. La plupart des modèles récents sont des smartphones, c'est-à-dire des dispositifs mobiles permettant de communiquer, s'informer, interagir, filmer... et dont les possibilités ne cessent de s'accroître. Ces dispositifs ont des processeurs de plus en plus rapides, leurs écrans permettent de capter plusieurs points de contact et ils possèdent désormais toute une batterie de capteurs (accéléromètres, gyroscopes, etc.). Le développement de ces nouvelles possibilités est cependant contraint par les limitations interactionnelles imposées par la taille réduite de l'écran, l'absence de clavier physique ou les spécificités de l'usage en mobilité (par exemple le fait qu'ils doivent fréquemment être manipulés avec une seule main). C'est pourquoi l'amélioration des interfaces et des interactions sur ce type de dispositif constitue un enjeu scientifique et industriel important. Nous allons maintenant présenter un résumé des contributions de cette thèse qui a été effectuée dans le cadre du laboratoire commun Ubimédia entre l'Institut Télécom et Alcatel Lucent Bell Labs. Nous indiquerons ensuite les perspectives qui semblent s'ouvrir dans la prolongation de ces travaux.

5.2 Contributions

Ce travail de thèse comprend deux parties complémentaires. Nous avons d'abord étudié l'état de l'art afin de réaliser un espace de classification des techniques de la littérature puis nous avons conçu plusieurs techniques d'interactions. Ces techniques sont regroupées suivant deux axes de recherche selon qu'elles exploitent les mouvements *sur* le dispositif ou les mouvements *du* dispositif.

5.2.1 Espace de classification

Pour faciliter la classification des techniques existantes et la conception de nouvelles techniques d'interaction nous avons proposé un espace de classification des interactions gestuelles physiques sur dispositif mobile. Cet espace s'appuie sur différents aspects de l'interaction gestuelle : technologique, cinématique, sémantique et contextuel. Il est composé de six dimensions :

- Le type de capteur définit la technologie d'entrée utilisée pour capter les gestes de l'interaction.
- Le type de mouvement définit la cinématique des mouvements entrant en jeu dans la réalisation des gestes.
- Le type de mapping définit l'association entre la forme des gestes et leur fonction.

1. Source ARCEP : <http://www.arcep.fr/index.php?id=11092&L=>

- Le type de contrôle définit l’association entre les gestes et les valeurs qu’ils contrôlent.
- Le type de retour d’information définit la technologie de sortie utilisée en réponse aux gestes réalisés.
- Le contexte définit le contexte d’interaction dans lequel l’interaction est réalisée.

Cet espace tente de fournir des outils pour améliorer l’étude des techniques existantes et la conception de nouvelles techniques. Il nous a en effet guidé dans l’élaboration des nouvelles techniques d’interaction qui ont été réalisées durant cette thèse.

5.2.2 Mouvements sur le dispositif

Dans cette partie nous avons étudié l’amélioration d’une technique existante, le Flick, par le biais de différents procédés. Nous avons dans un premier temps défini ‘l’anatomie’ du Flick afin de mieux décrire cette technique en mettant en valeur les différentes étapes qui la constituent : amorce, relâchement, défilement et arrêt. Nous avons pu ensuite nous concentrer sur certains aspects de ce type d’interaction qui nous semblaient peu explorés. Cette étude a donné lieu à la conception et à la mise au point de trois techniques d’interaction.



Flick and Brake Flick and Brake est une technique d’interaction qui permet à l’utilisateur de contrôler la vitesse de défilement lors de la réalisation d’un Flick. Cette technique développe la métaphore de l’inertie et de la friction déjà présente avec le Flick standard en y ajoutant l’effet de la friction du doigt au contact de l’écran. Cette technique permet d’atteindre plus rapidement une cible éloignée dans une liste et requiert beaucoup moins de répétitions du geste de Flick pour faire défiler une longue liste. Enfin, cette technique a été préférée à la technique de Flick classique par la majorité des utilisateurs de notre expérimentation.



Semantic Flicking Semantic Flicking est une technique qui permet de faire varier automatiquement la vitesse de défilement d’un document en fonction des contenus locaux de ce document. Le défilement lors du parcours du document est ainsi modifié en fonction de la nature et de l’importance des passages présentés à l’écran. Ainsi l’utilisateur peut grâce à cette technique parcourir un document en ‘sentant’ l’importance ou la pertinence d’une section seulement par la nature du défilement qui lui est associé. Semantic Flicking permet de mettre des items favoris en relief sans changer la structure initiale du document parcouru. Le prototype de cette technique appliqué au S-Book d’Alcatel-Lucent a été particulièrement apprécié par les utilisateurs lors de sa démonstration aux OpenDays 2011.



LongFlick LongFlick permet de faciliter la navigation en se servant du geste d’amorce du Flick pour contrôler la distance du défilement. L’utilisateur peut ainsi atteindre une partie éloignée d’un document avec un seul geste de Flick. Un mapping est créé entre le geste d’amorce réalisé sur la surface d’affichage du document et la taille du document parcouru. Le rapport entre l’amplitude du geste et la taille de la surface d’affichage détermine la longueur parcourue relativement à la taille totale du document. Nous avons proposé trois mappings différents : direct, proportionnel absolu et proportionnel relatif, chacun présentant une association différente entre la longueur de la surface d’affichage et la longueur de la liste. Cette technique dans ses trois variantes permet également de parcourir une distance choisie par l’utilisateur à vitesse constante.

5.2.3 Mouvements du dispositif

Dans cette partie nous nous sommes intéressé aux délimiteurs et nous avons proposé le concept de gestes 3D auto-délimités. L’interaction gestuelle 3D nécessite de pouvoir spécifier sans ambiguïté le début et la fin des gestes. Nous avons tout d’abord proposé une technique exploitant les gestes impulsifs pour délimiter la partie utile des gestes. Nous avons ensuite proposé un ensemble de gestes auto-délimités qui ne nécessitent pas le recours à des activateurs explicites, les délimiteurs étant compris dans la réalisation du geste lui-même.



TimeTilt TimeTilt est une technique basée sur l’utilisation des accéléromètres. Elle propose un vocabulaire gestuel qui préserve les interactions déjà existantes. Le vocabulaire de TimeTilt se compose de trois gestes impulsifs et d’un geste fluide. Un tap à l’arrière du dispositif (geste impulsif) sert d’activateur pour le geste fluide. TimeTilt a été appliquée à la navigation multi-vues sur dispositif mobile. Cette application propose deux modes d’interaction dont un se servant de la métaphore des cartes à effet lenticulaire. Cette technique a rencontré un franc succès lors de sa présentation aux OpenDays 2009.



JerkTilt JerkTilt introduit un ensemble de gestes auto-délimités détectés à partir des données fournies par un accéléromètre. Cet ensemble comprend huit gestes d’aller-retour permettant à l’utilisateur de revenir spontanément à sa position initiale. Ces gestes ont la particularité de requérir peu d’espace et peu d’effort de la part de l’utilisateur. Ils permettent également une interaction sans regarder le dispositif. Nous avons montré que ces gestes rivalisent en efficacité avec ceux des Marking Menus et qu’ils peuvent être combinés avec eux. Nous avons enfin proposé plusieurs applications de JerkTilts tirant partie de la nature des gestes utilisés pour permettre un accès rapide aux diverses commandes de ces applications.

5.3 Perspectives

L'étude de l'interaction gestuelle sur dispositifs mobiles est à notre avis un domaine de recherche primordial. Elle vise à permettre à l'utilisateur d'accéder efficacement à un nombre sans cesse croissant de données. Outre les problèmes de visualisation posés par la petitesse de la surface d'affichage, l'absence de clavier, de modifieurs, de boutons de souris, etc. impose de repenser l'interaction pour la rendre plaisante et efficace pour l'utilisateur. L'interaction gestuelle, en 2D sur la surface du dispositif, ou en 3D en déplaçant le dispositif dans l'espace, constitue une des voies les plus prometteuses. L'interaction 3D permet en outre d'exploiter de nouvelles ressources interactives sans parasiter les interactions déjà présentes.

Le succès impressionnant des smartphones, qui ont supplanté les téléphones mobiles usuels avec une rapidité imprévue, a montré à quel point l'apparition de nouvelles techniques d'interaction pouvait révolutionner un domaine technologique. Ces dispositifs ont largement contribué à populariser l'interaction gestuelle auprès du grand public. Pour autant nous ne sommes pas à la 'fin de l'histoire' et nous sommes convaincus que de nombreuses techniques restent à inventer.

Plus précisément, nous souhaitons étudier certaines pistes induites par les travaux présentés dans ce manuscrit. Nous présentons ci-dessous les perspectives issues des trois sections de ce travail de thèse.

5.3.1 Espace de Classification

Une de nos perspectives de travail est la mise à jour de notre espace de classification. De nouvelles technologies sont susceptibles d'être développées dans le futur pour permettre de nouveaux types d'interaction gestuelle. De ce fait la liste des capteurs que nous avons donnée dans le chapitre 2 devra être tenue à jour. La dimension contextuelle de cet espace nécessitera aussi d'être revisitée, les pratiques et les coutumes liées aux nouvelles technologies évoluant constamment.

À titre d'exemple, il y a encore peu d'années l'utilisation d'oreillettes pour téléphones mobiles donnait l'étrange impression que la personne se parlait à elle-même. Il nous paraît évident aujourd'hui que cette personne est en train de téléphoner. De même, l'apparition des contrôleurs de jeux vidéos gestuels comme la Wiimote, la caméra Kinect ou encore le Playstation Move ont popularisé l'idée d'interagir au moyen de gestes tri-dimensionnels. De tels mouvements, aujourd'hui parfaitement acceptés dans l'environnement familial, auraient été considérés comme inacceptables et parfaitement ridicules il y a peu.

5.3.2 Mouvements sur le Dispositif

Nous avons plusieurs perspectives de travail sur ce domaine. Dans un premier temps nous aimerions évaluer Flick-and-Brake dans un contexte d'interaction réel. L'attention étant souvent partagée entre la tâche que l'on souhaite réaliser et l'environnement alentour dans un contexte de mobilité, nous aimerions évaluer les performances de notre technique dans cette situation. De même, nous aimerions évaluer l'impact du type de retour d'information sur l'efficacité de la technique.

Dans un second temps il serait intéressant de mesurer l'apport du Semantic Flicking sur la consultation de documents textuels à l'aide d'une expérience contrôlée.

Enfin nous aimerions évaluer LongFlick, la dernière technique que nous avons proposée, afin d'étudier quantitativement son apport relativement au Flick classique. Nous aimerions notamment évaluer les différents mapping. Par exemple le fait que la distance soit différente selon la position avec le mapping relatif affecte-t-elle les performances ? Quelles sont les configurations optimales pour ces trois mappings ? Par ailleurs il serait intéressant d'étudier plus en profondeur les différents aspects du Flick, en particulier les divers paramètres de la phase d'amorçage (forme, direction, etc.) afin d'élaborer de nouveaux moyens d'interaction. Une dernière perspective pour cette partie serait d'étudier l'utilisation de ce type de technique sur des surfaces tactiles de grande taille comme les tables et tableaux interactifs.

5.3.3 Mouvements du Dispositif

Comme pour la section précédente une première perspective consisterait à étudier de manière longitudinale l'utilisation de nos techniques dans un contexte d'utilisation réel. Nous savons par exemple que JerkTilt est susceptible de fonctionner dans la vie de tous les jours, mais nous aimerions étudier comment les gens adoptent la technique et s'il est facile de l'utiliser dans un contexte de mobilité réelle. Une deuxième perspective serait d'approfondir l'étude de l'extension de JerkTilt afin d'étudier les limites de cette technique. Nous aimerions notamment savoir dans quelle mesure il est possible d'effectuer simultanément les gestes 3D et 2D. Dans cette hypothèse, il semble important de déterminer les directions qui sont les plus faciles à réaliser et si certaines d'entre-elles demandent un effort cognitif trop important aux utilisateurs.

Dans un second temps nous souhaiterions affiner nos techniques en combinant l'utilisation des accéléromètres et des gyroscopes. Cette combinaison permettrait sans doute de donner plus de précision aux mesures des mouvements et, peut-être, d'augmenter le vocabulaire gestuel.

Enfin, une dernière perspective de travail porterait sur l'étude des gestes auto-délimités sur des dispositifs mobiles de plus grande taille comme les tablettes tactiles. D'abord il serait

intéressant d'évaluer les Jerktilts (à une main) sur ce type de dispositif. Ensuite l'usage montre que les tablettes tactiles sont souvent utilisées avec deux mains. Il pourrait donc être intéressant de déterminer un nouveau vocabulaire gestuel reposant sur des gestes auto-délimités à deux mains.

Publications Personnelles

Baglioni, M., Lecolinet, E., and Guiard, Y. (2011). JerkTilts : Using Accelerometers for Eight-Choice Selection on Mobile Devices. *ICMI '11 : Proceedings of the 13th international conference on Multimodal interfaces*.

Baglioni, M., Malacria, S., Lecolinet, E., and Guiard, Y. (2011). Flick-and-brake : finger control over inertial/sustained scroll motion. *In Proceedings of CHI 2011, ACM Conference on Human Factors in Computing Systems (extended abstracts)*.

Baglioni, M. (2010). Interaction physique sur dispositifs mobiles. *Rencontre Doctorale IHM '10 : In Proceedings of the 22st International Conference on Association Francophone d'Interaction Homme-Machine*.

Baglioni, M., Lecolinet, E., and Guiard, Y. (2009). Espace de caractérisation des interactions gestuelles physiques sur dispositifs mobiles. *IHM '09 : In Proceedings of the 21st International Conference on Association Francophone d'Interaction Homme-Machine*.

Roudaut, A., **Baglioni, M.**, and Lecolinet, E. (2009). TimeTilt : Using Sensor-Based Gestures to Travel through Multiple Applications on a Mobile Device. *INTERACT '09 : In Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction : Part I*.

Liste des Figures

1.1	Exemples de Smartphones : iPhone 4 et Google Nexus One. Comme on peut constater ces deux modèles ont un écran tactile mais ne possèdent plus de clavier physique. Ils comportent seulement un nombre limité de boutons réservés à des tâches spécifiques.	12
1.2	En bleu le référentiel terrestre et en rouge le référentiel du mobile	15
2.1	Taxonomie des techniques d'interaction et de visualisation [Roudaut and Lecolinet, 2007].	21
2.2	Taxonomie des interactions gestuelles [Karam and Schraefel, 2005].	22
2.3	Taxonomie proposée par Scoditti.	24
2.4	Taxonomie proposée par Wobbrock. (“w.r.t” = “with respect to”).	24
2.5	Taxonomie proposée par Malacria.	25
2.6	Espace de caractérisation	27
2.7	Référentiel des accéléromètres	28
2.8	Changement d'orientation sur l'iPhone grâce à l'accéléromètre.	29
2.9	Deux types de gyroscope, un mécanique et un microélectromécanique. On retrouve ce dernier dans la plupart des dispositifs mobiles actuels.	30
2.10	Deux types d'utilisation commerciale des gyroscopes	30
2.11	Sélection d'items dans un menu sur une télévision interactive.	31
2.12	Abracadabra	31
2.13	Google Nexus S, appareil construit par Samsung et muni d'une boussole. Ici on peut voir Une vue de Google Map avec l'indication de l'orientation.	32
2.14	Exemple de puce servant de magnétomètre.	32
2.15	Deux interfaces utilisant la caméra pour interagir gestuellement.	33
2.16	La métaphore de la boîte proposée par Fracone <i>et al.</i> . En changeant l'orientation de l'appareil l'utilisateur peut accéder aux parties cachées de l'interface.	33
2.17	Le LeafMenu. De a) à d) les différentes étapes de l'interaction novice. e) l'interaction experte.	34
2.18	Le wavelet menu sur iphone.	34

2.19	Les six différents types de roulement du bout du doigt à la base de la technique Microrolls.	35
2.20	Les trois différents types de mouvement. Chacun des mouvements est différent des autres au niveau des accélérations en fonction du temps. L'axe des ordonnées représente l'accélération selon l'axe y en m/s^2 et l'axe des abscisses représente le temps.	36
2.21	Deux exemples de gestes constitués de mouvements impulsifs.	37
2.22	Rahman <i>et al.</i> étudient les possibilités du poignet pour réaliser des gestes de rotation (flexion/extension, pronation/supination)	38
2.23	Rock'n'Scroll.	38
2.24	Le Motion Marking Menu [Oakley and Park, 2007]. L'utilisateur incline le dispositif pour atteindre les différents items du menu.	39
2.25	Motion Pointing. L'utilisateur reproduit l'oscillation affichée sur un item pour le sélectionner.	40
2.26	CycloZoom et CycloStar : deux techniques permettant respectivement de zoomer et panner sur une surface tactile.	40
2.27	Représentation des différents gestes proposés par Nancel pour interagir sur très grand écran, (incluant des gestes basés sur des mouvements oscillatoires).	40
2.28	Gesture Avatar	41
2.29	Téléphone utilisé comme souris pour un écran distant.	42
2.30	Réorientation du texte en fonction de l'orientation de l'appareil.	42
2.31	Head-Coupled Perspective Technique.	43
2.32	TiltText	43
2.33	Symboles associés à des commandes d'un magnétoscope.	44
2.34	Gestes symboliques utilisés dans les travaux de [Kauppila et al., 2008].	44
2.35	Gestes associés à des commandes d'un menu linéaire avec un mapping arbitraire.	45
2.36	Octopocus : l'utilisateur clique et voit apparaître les différents chemins possibles. Lorsqu'il se déplace le nombre de chemins se réduit, ceci ne laissant apparaître que les commandes qu'il peut encore atteindre.	46
2.37	Geste de pointage pour coupler deux appareils entre eux.	47
2.38	RealEyes3D, l'utilisateur incline son dispositif pour naviguer dans la carte.	48
2.39	SixthSense	48
2.40	Tilt and Feel	49
2.41	Tableau de classification des techniques de la littérature.	53
3.1	Le geste de Flick et le défilement (la flèche bleue représente le mouvement du doigt et la flèche noire celui de la liste).	58
3.2	Gestes de Flick pour lancer et attraper un objet sur une table interactive.	59
3.3	Les différentes étapes de SuperFlick. L'utilisateur lance d'abord l'objet puis peut effectuer un drag d'ajustement depuis la position d'arrivée.	60

3.4	Les différents mapping de Multi-Flick entre la vitesse du Flick et la vitesse de défilement du document. a) La vitesse du flick s'additionne ou se soustrait à la vitesse courante. b) La vitesse d'un Flick définit la vitesse de défilement quelle que soit la vitesse courante. Le défilement continue jusqu'à ce que le geste soit terminé. c) Une friction est appliquée à la vitesse de défilement. d) Comme pour b) mais le début du geste de Flick arrête le défilement.	60
3.5	Amorce du Flick.	61
3.6	Relâchement du Flick.	61
3.7	Défilement automatique du Flick.	62
3.8	Différents types d'arrêts du Flick	62
3.9	Globe terrestre	64
3.10	Le concept général de Flick-and-Brake : l'utilisateur lance le défilement avec un Flick puis en contrôle le freinage en modulant la pression de son doigt sur l'écran.	66
3.11	Comportement de la technique Flick-and-Brake avec la métaphore du mouvement inertiel.	67
3.12	Métaphore du mouvement entretenu.	68
3.13	Automate fini générique permettant de décrire les techniques Flick-and-Brake et les techniques de Flick habituelles	69
3.14	Différents types de retour visuel pour Flick-and-brake	71
3.15	Dispositif et interface d'expérimentation.	73
3.16	Temps moyens de sélection par technique et par distance cible.	74
3.17	Nombre de Flicks par sélection de cible.	75
3.18	Préférences subjectives des participants	76
3.19	Flick and Brake	78
3.20	Semantic pointing. a) espace visuel inchangé, b) espace moteur.	80
3.21	Splitmenu. Réorganisation des items d'un menu en fonction de leur fréquence d'utilisation.	80
3.22	Ralentissement du défilement lorsqu'un passage plus important apparaît à l'écran.	82
3.23	Schéma complet du système <i>heatmap</i> et Flicking sémantique	83
3.24	Représentation des trajectoires de déplacement en fonction de la nature du Flick (sémantique ou non).	83
3.25	Le geste de Flick et le défilement (la flèche bleue représente le mouvement du doigt et la flèche noire celui de la liste).	85
3.26	Différentes positions de démarrage du Flick	86
3.27	Différentes directions du Flick	86
3.28	Différentes longueurs du geste du Flick	87
3.29	Exemple de forme complexe du geste de Flick	88
3.30	Mapping direct : $D = d * k$. d est la longueur parcourue par le doigt et D la distance de défilement automatique à vitesse constante.	89

3.31 Mapping Absolu : $D(d) = d * L / S$. d la longueur du tracé, L la longueur totale de la liste, S la longueur de la surface d'affichage et D la distance parcourue.	90
3.32 Mapping Relatif : $D(d) = d * RL / S$. d la longueur du tracé, RL la longueur restante de la liste, S la longueur de la surface d'affichage et D la distance parcourue.	91
4.1 Profil temporel d'accélération pour les deux types majeurs de geste	94
4.2 Variations d'accélération selon l'axe z du mobile pour les gestes de TimeTilt	97
4.3 Changement entre deux applications sur iPad	100
4.4 Changement entre deux applications sur le Nexus One sous Android	101
4.5 Carte lenticulaire ou holographique présentant différentes images en fonction de l'angle de vision.	102
4.6 Effet lenticulaire virtuel en fonction de l'inclinaison du mobile.	102
4.7 Tap de l'index au dos pour l'activation de la navigation.	103
4.8 L'angle de 90 degrés de rotation du mode lenticulaire pour différentes positions initiales.	103
4.9 Les deux gestes du mode Superposition. Un geste impulsif vers le haut et un vers le bas (vue longitudinale du dispositif).	104
4.10 Taux de réponse à deux questions du questionnaire.	105
4.11 Cycle complet de réalisation d'un geste de JerkTilts. Le geste part de la position initiale puis revient automatiquement en position initiale en fin de geste.	108
4.12 Les différents gestes composant le vocabulaire de JerkTilts. N'est représentée que la position maximum pour chaque geste aller-retour.	108
4.13 Courbes d'enregistrement des accélérations : la courbe verte représente les accélérations en x et la courbe bleue en y.	109
4.14 En vert, à gauche, on peut voir un exemple de stimulus, ici le côté Ouest (correspondant à une rotation selon l'axe de <i>roll</i> vers la gauche). En gris, sur la droite, le carré apparaissant lorsque le dispositif est en position neutre.	112
4.15 Les taux de reconnaissance pour les deux types de position du pouce versus les deux types de reconnaissance de l'algorithme.	114
4.16 Exemple de retour d'information renvoyé par le système pour un geste bien réalisé (a) ou raté (b)	115
4.17 Les taux de reconnaissance des deux techniques en fonction des huit directions cibles.	118
4.18 Les temps totaux (TT) moyens des deux techniques en fonction des huit directions cibles.	118
4.19 Nombre d'activations involontaires par utilisateur.	120
4.20 JerkTilts Window	122
4.21 Un exemple de séquence permettant de réaliser un copier/coller d'une image à partir d'une page web vers une autre application.	123

4.22 JerkTilts Music Remote	124
4.23 Niveau Applications du menu	125
4.24 Niveau Fonctionnalités du menu.	127
4.25 Exemple de stimulus de combinaison JerkTilt(flèche jaune)/ThumbSlide(flèche bleue).	128
4.26 Taux de reconnaissance de la combinaison JerkTilts/ThumbSlide (les barres correspondent au 95% d'intervalle de confiance).	129
4.27 Temps moyen de la combinaison JerkTilts/ThumbSlide en fonction des 4 directions cardinales (les barres correspondent au 95% d'intervalle de confiance).129	

Liste des tableaux

3.1	Temps moyens en seconde par technique.	75
3.2	Coefficient de la régression linéaire par technique et par direction	76
3.3	Classement des techniques préférées par les participants	77
4.1	Matrice de confusion de reconnaissance des gestes Tap à l'arrière(TAr), Impulsif vers l'avant(IAv) et Impulsif vers l'arrière(IAr)	99
4.2	Les six composantes du vecteur de geste.	110
4.3	Nombre de détections d'activations d'un geste et temps moyen de l'enregistrement.	120

Bibliographie

- [Aliakseyeu et al., 2008] Aliakseyeu, D., Irani, P., Lucero, A., and Subramanian, S. (2008). Multi-flick : an evaluation of flick-based scrolling techniques for pen interfaces. In *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1689–1698, New York, NY, USA. ACM.
- [Andersen, 2005] Andersen, T. H. (2005). A simple movement time model for scrolling. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI '05, pages 1180–1183, New York, NY, USA. ACM.
- [Appert and Zhai, 2009] Appert, C. and Zhai, S. (2009). Using Strokes as Command Shortcuts : Cognitive Benefits and Toolkit Support. In *International conference on Human factors in computing systems*, pages 2289–2298, Boston, MA, United States.
- [Baffoun and Robert, 2006] Baffoun, S. and Robert, J.-M. (2006). État de l’art des techniques de présentation d’information sur écran d’assistant numérique personnel. *IHM '06 : Proceedings of the 18th International Conference of the Association Francophone d’Interaction Homme-Machine*.
- [Bailly, 2009] Bailly, G. (2009). *Technique de menus : Caractérisation, Conception et Evaluation*. PhD thesis, Université de Grenoble.
- [Bailly et al., 2008] Bailly, G., Roudaut, A., Lecolinet, E., and Nigay, L. (2008). Menu leaf : enrichir les menus linéaires par des gestes. *IHM '08 : Proceedings of the 20th International Conference of the Association Francophone d’Interaction Homme-Machine*.
- [Bailly et al., 2011] Bailly, G., Vo, D.-B., Lecolinet, E., and Guiard, Y. (2011). Gesture-aware remote controls : guidelines and interaction technique. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 263–270, New York, NY, USA. ACM.
- [Ballagas, 2006] Ballagas (2006). The smart phone : a ubiquitous input device. *Pervasive Computing, IEEE*, 5(1) :70 – 77.
- [Bartlett, 2000] Bartlett, J. (2000). Rock’n’scroll is here to stay [user interface]. *Computer Graphics and Applications*.
- [Bau and Mackay, 2008] Bau, O. and Mackay, W. E. (2008). Octopocus : a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium*

- sium on User interface software and technology*, UIST '08, pages 37–46, New York, NY, USA. ACM.
- [Baudisch and Chu, 2009] Baudisch, P. and Chu, G. (2009). Back-of-device interaction allows creating very small touch devices. *CHI '09 : Proceedings of the 27th international conference on Human factors in computing systems*.
- [Baudisch and Rosenholtz, 2003] Baudisch, P. and Rosenholtz, R. (2003). Halo : a technique for visualizing off-screen objects. *CHI '03 : Proceedings of the SIGCHI conference on Human factors in computing systems*.
- [Beaudouin-Lafon, 2000] Beaudouin-Lafon, M. (2000). Ceci n'est pas un ordinateur : Perspectives sur l'interaction homme-machine. *TSI. Technique et science informatiques*.
- [Blanch et al., 2004] Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. (2004). Semantic pointing : improving target acquisition with control-display ratio adaptation. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 519–526.
- [Bragdon et al., 2011] Bragdon, A., Nelson, E., Li, Y., and Hinckley, K. (2011). Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 403–412, New York, NY, USA. ACM.
- [Brewster and Hughes, 2009] Brewster, S. A. and Hughes, M. (2009). Pressure-based text entry for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 9 :1–9 :4, New York, NY, USA. ACM.
- [Buxton, 1983] Buxton, W. (1983). Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.*, 17 :31–37.
- [Cadoz, 1994] Cadoz, C. (1994). Le geste canal de communication homme/machine : la communication «instrumentale». *TSI : Technique et Science Informatique*, 13(1) :31–61.
- [Calvary et al., 2002] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., and Vanderdonckt, J. (2002). Plasticity of user interfaces : A revisited reference framework. In *Task Models and Diagrams for User Interface Design*.
- [Card et al., 1991] Card, S. K., Mackinlay, J. D., and Robertson, G. G. (1991). A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.*, 9 :99–122.
- [Cechanowicz et al., 2007] Cechanowicz, J., Irani, P., and Subramanian, S. (2007). Augmenting the mouse with pressure sensitive input. *CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*.
- [Dearman et al., 2010] Dearman, D., Karlson, A., Meyers, B., and Bederson, B. (2010). Multi-modal text entry and selection on a mobile device. *GI '10 : Proceedings of Graphics Interface 2010*.
- [Dourish, 2001] Dourish, P. (2001). *Where the action is : the foundations of embodied interaction*. MIT Press, Cambridge, MA, USA.

- [Essl et al., 2010] Essl, G., Rohs, M., and Kratz, S. (2010). Use the Force (or something) - Pressure and Pressure-Like Input for Mobile Music Performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Sydney, Australia.
- [Falaki et al., 2010] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., and Estrin, D. (2010). Diversity in smartphone usage. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 179–194, New York, NY, USA. ACM.
- [Fekete et al., 2009] Fekete, J.-D., Elmqvist, N., and Guiard, Y. (2009). Motion-pointing : target selection using elliptical motions. In *CHI '09 : Proceedings of the 27th international conference on Human factors in computing systems*, pages 289–298, New York, NY, USA. ACM.
- [Ferscha et al., 2008] Ferscha, A., Vogl, S., Emsenhuber, B., and Wally, B. (2008). Physical shortcuts for media remote controls. *Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, page 9.
- [Fishkin et al., 1999] Fishkin, K. P., Moran, T. P., and Harrison, B. L. (1999). Embodied user interfaces : Towards invisible user interfaces. In *Proceedings of the IFIP TC2/TC13 WG2.7/WG13.4 Seventh Working Conference on Engineering for Human-Computer Interaction*, pages 1–18, Deventer, The Netherlands, The Netherlands. Kluwer, B.V.
- [Foley et al., 1984] Foley, J. D., Wallace, V. L., and Chan, P. (1984). The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl.*, 4 :13–48.
- [Francone et al., 2010] Francone, J., Bailly, G., Lecolinet, E., Mandran, N., and Nigay, L. (2010). Wavelet menus on handheld devices : stacking metaphor for novice mode and eyes-free selection for expert mode. *AVI '10 : Proceedings of the International Conference on Advanced Visual Interfaces*.
- [Francone and Nigay, 2011] Francone, J. and Nigay, L. (2011). Using the user's point of view for interaction on mobile devices. In *Actes de la 23ème Conférence francophone sur l'Interaction Homme-Machine (IHM'2011, Nice, Octobre2011)*. ACM press, pages 25–31.
- [Geissler, 1998] Geissler, J. (1998). Shuffle, throw or take it ! working efficiently with an interactive wall. In *CHI '98 : CHI 98 conference summary on Human factors in computing systems*, pages 265–266, New York, NY, USA. ACM.
- [Gibson, 1977] Gibson, J. J. (1977). The Theory of Affordances. *Perceiving, Acting and Knowing : Toward an Ecological Psychology*, pages 67–82.
- [Guiard, 1993a] Guiard, Y. (1993a). On fitts's and hooke's laws : Simple harmonic movement in upper-limb cyclical aiming. *Acta Psychologica*, 82 :139–159.
- [Guiard, 1993b] Guiard, Y. (1993b). On fitts's and hooke's laws : Simple harmonic movement in upper-limb cyclical aiming. *Acta Psychologica*, 82(1-3) :139–159.

- [Hachet et al., 2005] Hachet, M., Pouderoux, J., and Guitton, P. (2005). A camera-based interface for interaction with mobile handheld computers. *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 65–72.
- [Haro et al., 2005] Haro, A., Mori, K., Setlur, V., and Capin, T. (2005). Mobile camera-based adaptive viewing. *Proceedings of the 4th international Conference on Mobile and Ubiquitous Multimedia*, pages 78–83.
- [Harrison and Hudson, 2009] Harrison, C. and Hudson, S. E. (2009). Abracadabra : wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pages 121–124, New York, NY, USA. ACM.
- [Hinckley, 2003] Hinckley, K. (2003). Synchronous gestures for multiple persons and computers. *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 149–158.
- [Hinckley et al., 2000] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. (2000). Sensing techniques for mobile interaction. *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 91–100.
- [Karam and Schraefel, 2005] Karam, M. and Schraefel, M. C. (2005). A taxonomy of gestures in human computer interactions. Technical Report ECSTR-IAM05-009, University of Southampton.
- [Karlson et al., 2005] Karlson, A., Bederson, B., and SanGiovanni, J. (2005). Apppens and launchtile : two designs for one-handed thumb use on small devices. *CHI '05 : Proceedings of the SIGCHI conference on Human factors in computing systems*.
- [Kauppila, 2008] Kauppila, M. (2008). *Filler Models for Accelerometer Based Continuous Gesture Recognition*. PhD thesis, University of Oulu.
- [Kauppila et al., 2008] Kauppila, M., Inkeroinen, T., Pirttikangas, S., and Riekkilä, J. (2008). Mobile phone controller based on accelerative gesturing. *Adjunct Proceedings Pervasive*, pages 130–133.
- [Keim, 2002] Keim, D. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*.
- [Kurtenbach and Buxton, 1991a] Kurtenbach, G. and Buxton, W. (1991a). Issues in combining marking and direct manipulation techniques. In *UIST '91 : Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 137–144, New York, NY, USA. ACM.
- [Kurtenbach and Buxton, 1991b] Kurtenbach, G. and Buxton, W. (1991b). Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, UIST '91, pages 137–144, New York, NY, USA. ACM.
- [Ladetto and Merminod, 2002] Ladetto, Q. and Merminod, B. (2002). Digital magnetic compass and gyroscope integration for pedestrian navigation. *9th Saint Petersburg Inter-*

- national Conference on Integrated Navigation Systems, Saint Petersburg, Russia*, pages 111–120.
- [Linjama and Kaaresoja, 2004] Linjama, J. and Kaaresoja, T. (2004). Novel, minimalist haptic gesture interaction for mobile devices. *Proceedings of the third Nordic conference on Human-computer interaction*, pages 457–458.
- [Lü and Li, 2011] Lü, H. and Li, Y. (2011). Gesture avatar : a technique for operating mobile user interfaces using gestures. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 207–216, New York, NY, USA. ACM.
- [Lü and Li, 2011] Lü, H. and Li, Y. (2011). Gesture avatar : A technique for operating mobile user interfaces using gestures. In *CHI'11 : Proceedings of the 29th international conference on Human factors in computing systems*, New York, NY, USA. ACM.
- [Malacria, 2011] Malacria, S. (2011). *Conception et Evaluation de Techniques d'Interaction pour Surfaces Tactiles et Papier Augmenté*. PhD thesis, Télécom ParisTech.
- [Malacria et al., 2010a] Malacria, S., Lecolinet, E., and Guiard, Y. (2010a). Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces : the cyclostar approach. *CHI '10 : Proceedings of the 28th international conference on Human factors in computing systems*.
- [Malacria et al., 2010b] Malacria, S., Lecolinet, E., and Guiard, Y. (2010b). Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces : the cyclostar approach. In *CHI '10 : Proceedings of the 28th international conference on Human factors in computing systems*, pages 2615–2624, New York, NY, USA. ACM.
- [McCallum et al., 2009] McCallum, D., Mak, E., Irani, P., and Subramanian, S. (2009). Pressuretext : pressure input for mobile phone text entry. *CHI EA '09 : Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*.
- [Mistry et al., 2009] Mistry, P., Maes, P., and Chang, L. (2009). Wuw-wear ur world : a wearable gestural interface. *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4111–4116.
- [Miyaki and Rekimoto, 2009] Miyaki, T. and Rekimoto, J. (2009). Graspzoom : zooming and scrolling control model for single-handed mobile interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 11 :1–11 :4, New York, NY, USA. ACM.
- [Moulon, 2010] Moulon, D. (2010). Tendances d'un art d'aujourd'hui. *Arts Média*, 41 :101.
- [Murray-Smith et al., 2008] Murray-Smith, R., Williamson, J., Hughes, S., Quaade, T., and Strachan, S. (2008). Rub the stone. *CHI'08 extended abstracts on Human factors in computing systems*, pages 2355–2360.
- [Nancel et al., 2009] Nancel, M., Huot, S., and Beaudouin-Lafon, M. (2009). Un espace de conception fondé sur une analyse morphologique des techniques de menus. In *IHM '09 :*

- Proceedings of the 21st International Conference on Association Francophone d'Interaction Homme-Machine*, pages 13–22, New York, NY, USA. ACM.
- [Nancel et al., 2011] Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W. (2011). Mid-air pan-and-zoom on wall-sized displays. In *CHI '11 : Proceedings of the 29th international conference on Human factors in computing systems*, pages 177–186. ACM.
- [Nigay and Coutaz, 1996] Nigay, L. and Coutaz, J. (1996). Espaces conceptuels pour l'interaction multimédia et multimodale. *Technique et science informatiques*, 15(9) :1195–1225.
- [Norman, 1999] Norman, D. A. (1999). Affordance, conventions, and design. *interactions*, 6 :38–43.
- [Oakley et al., 2004] Oakley, I., Ängeslevä, J., and Hughes... , S. (2004). Tilt and feel : Scrolling with vibrotactile display. *EuroHaptics 2004*.
- [Oakley and Park, 2007] Oakley, I. and Park, J. (2007). A motion-based marking menu system. *CHI'07 extended abstracts on Human factors in computing systems*, pages 2597–2602.
- [Olwal et al., 2008] Olwal, A., Feiner, S., and Heyman, S. (2008). Rubbing and tapping for precise and rapid selection on touch-screen displays. In *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 295–304, New York, NY, USA. ACM.
- [Pering et al., 2007] Pering, T., Anokwa, Y., and Want, R. (2007). Gesture connect : facilitating tangible interaction with a flick of the wrist. *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 259–262.
- [Poirier and Schadle, 2004] Poirier, F. and Schadle, I. (2004). État de l'art des méthodes de saisie de données sur dispositifs nomades : typologie des approches. *Proceedings of the 16th conference on Association Francophone d'Interaction Homme-Machine*, pages 133–140.
- [Pook et al., 2000a] Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. (2000a). Context and interaction in zoomable user interfaces. In *Proceedings of the working conference on Advanced visual interfaces, AVI '00*, pages 227–231, New York, NY, USA. ACM.
- [Pook et al., 2000b] Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. (2000b). Control menus : execution and control in a single interactor. *CHI'00 extended abstracts on Human factors in computing systems*, pages 263–264.
- [Quek et al., 2002] Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-F., Kirbas, C., McCullough, K. E., and Ansari, R. (2002). Multimodal human discourse : gesture and speech. *ACM Trans. Comput.-Hum. Interact.*, 9 :171–193.
- [Quinn and Cockburn, 2009] Quinn, P. and Cockburn, A. (2009). Zoofing! : faster list selections with pressure-zoom-flick-scrolling. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group : Design : Open 24/7, OZCHI '09*, pages 185–192, New York, NY, USA. ACM.

- [Rahman et al., 2009] Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. (2009). Tilt techniques : investigating the dexterity of wrist-based input. *CHI '09 : Proceedings of the 27th international conference on Human factors in computing systems*.
- [Ramos and Balakrishnan, 2007] Ramos, G. and Balakrishnan, R. (2007). Pressure marks. *CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*.
- [Ramos et al., 2004] Ramos, G., Boulos, M., and Balakrishnan, R. (2004). Pressure widgets. *CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*.
- [Reetz et al., 2006] Reetz, A., Gutwin, C., Stach, T., Nacenta, M., and Subramanian, S. (2006). Superflick : a natural and efficient technique for long-distance object placement on digital tables. In *GI '06 : Proceedings of Graphics Interface 2006*, pages 163–170, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.
- [Rehm et al., 2008] Rehm, M., Bee, N., and André, E. (2008). Wave like an egyptian : accelerometer based gesture recognition for culture specific interactions. *BCS-HCI '08 : Proceedings of the 22nd British HCI Group Annual Conference on People and Computers : Culture, Creativity, Interaction*, 1.
- [Rekimoto, 1996] Rekimoto, J. (1996). Tilting operations for small screen interfaces. *UIST '96 : Proceedings of the 9th annual ACM symposium on User interface software and technology*.
- [Rico and Brewster, 2010a] Rico, J. and Brewster, S. (2010a). Usable gestures for mobile interfaces : evaluating social acceptability. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 887–896, New York, NY, USA. ACM.
- [Rico and Brewster, 2010b] Rico, J. and Brewster, S. (2010b). Usable gestures for mobile interfaces : evaluating social acceptability. *CHI '10 : Proceedings of the 28th international conference on Human factors in computing systems*.
- [Roudaut et al., 2009a] Roudaut, A., Bailly, G., Lecolinet, E., and Nigay, L. (2009a). Leaf menus : Linear menus with stroke shortcuts for small handheld devices. In *INTERACT '09 : Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction*, pages 616–619, Berlin, Heidelberg. Springer-Verlag.
- [Roudaut et al., 2008] Roudaut, A., Huot, S., and Lecolinet, E. (2008). Taptap and magstick : improving one-handed target acquisition on small touch-screens. *AVI '08 : Proceedings of the working conference on Advanced visual interfaces*.
- [Roudaut and Lecolinet, 2007] Roudaut, A. and Lecolinet, E. (2007). Un espace de classification pour l'interaction sur dispositifs mobiles. *Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine*, pages 99–106.
- [Roudaut et al., 2009b] Roudaut, A., Lecolinet, E., and Guiard, Y. (2009b). Microrolls : expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. *CHI '09 : Proceedings of the 27th international conference on Human factors in computing systems*.

- [Rubine, 1991] Rubine, D. (1991). Specifying gestures by example. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 329–337, New York, NY, USA. ACM.
- [Ruiz and Li, 2011] Ruiz, J. and Li, Y. (2011). Doubleflip : a motion gesture delimiter for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2717–2720, New York, NY, USA. ACM.
- [Schlömer et al., 2008] Schlömer, T., Poppinga, B., Henze, N., and Boll, S. (2008). Gesture recognition with a wii controller. *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14.
- [Scoditti et al., 2011] Scoditti, A., Blanch, R., and Coutaz, J. (2011). A novel taxonomy for gestural interaction techniques based on accelerometers. In *Proceedings of the 15th international conference on Intelligent User Interfaces (IUI 2011)*, pages 63–72. ACM.
- [Sears and Shneiderman, 1994] Sears, A. and Shneiderman, B. (1994). Split menus : effectively using selection frequency to organize menus. *ACM Trans. Comput.-Hum. Interact.*, 1 :27–51.
- [Shi et al., 2008] Shi, K., Irani, P., Gustafson, S., and Subramanian, S. (2008). Pressurefish : a method to improve control of discrete pressure-based input. *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*.
- [Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it : a task by data type taxonomy for information visualizations. *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336 – 343.
- [Wigdor and Balakrishnan, 2003] Wigdor, D. and Balakrishnan, R. (2003). Tilttext : using tilt for text input to mobile phones. *UIST '03 : Proceedings of the 16th annual ACM symposium on User interface software and technology*.
- [Wiliamson et al., 2011] Wiliamson, J. R., Crossan, A., and Brewster, S. (2011). Multimodal mobile interactions : usability studies in real world settings. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 361–368, New York, NY, USA. ACM.
- [Wobbrock et al., 2009] Wobbrock, J., Morris, M., and Wilson, A. (2009). User-defined gestures for surface computing. *CHI '09 : Proceedings of the 27th international conference on Human factors in computing systems*.
- [Wobbrock et al., 2006] Wobbrock, J., Myers, B., and Rothrock, B. (2006). Few-key text entry revisited : mnemonic gestures on four keys. *CHI '06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*.
- [Wu and Balakrishnan, 2003] Wu, M. and Balakrishnan, R. (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03 : Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA. ACM.

Appendices

Questionnaire sur les usages mobiles pour TimeTilt

Etude sur l'utilisation des dispositifs mobiles

Ce questionnaire a pour but de mieux comprendre comment vous utilisez vos téléphones mobiles et nous permettront de proposer des services innovants et adaptés.

Nous souhaitons avoir les avis d'utilisateur possédant des téléphones nouvelles générations (de type iPhone ou bien des téléphones à écrans tactiles)

Merci de participer à cette étude qui ne vous prendra que quelques minutes :-)

***Obligatoire**

Prenom Nom

(non obligatoire)

Email

(non obligatoire)

Age *

Quel type de téléphone possédez-vous ? *

(Si vous ne connaissez pas le type, entrez la marque du téléphone dans le champ "autre")

- iPhone
- SmartPhone sous Windows Mobile
- SmartPhone sous Android
- Autre :

Avez-vous accès au web (par forfait ou par wifi) *

- Oui et je vais sur le web
- Oui mais je ne vais pas sur le web
- Non

Combien d'applications environ avez-vous sur votre téléphone ? *

(Telephone, Mail, SMS, Internet, Bourse, Meteo, Carte, Casse-Brique, Tetris ...)

Combien de fois par jour en moyenne, sortez vous votre téléphone pour l'utiliser ? *

- 1 fois par jour
- 2 fois par jour
- 4 fois par jour
- Toutes les heures
- Toutes les 30 minutes

- Toutes les 10 minutes
- Toutes les 5 minutes

Lorsque vous sortez votre téléphone pour l'utiliser, combien de temps environ y passez vous ? *

- Moins de 1 minute
- 1 à 2 minutes
- 5 minutes
- 10 minutes
- 20 minutes
- 30 minutes
- Plus de 30 minutes

Toujours pendant ces "sorties", combien de services en moyenne utilisez vous ? *

Pendant ces "sorties", cochez dans la liste ci-dessous, les services que vous utilisez le plus souvent : *

- Telephone
- SMS
- Mail
- Internet
- Google map (ou carte)
- GPS
- MP3
- Video
- Jeux
- Appareil photo
- Consultation de photos ou videos personnelles
- Recherche d'application (AppStore ou autres services équivalents)
- Edition de texte
- Lecture de livres électroniques
- Utilitaires (calculatrice, convertisseur ...)
- Autre :

Pendant ces sorties, vous arrive t'il d'utiliser (ou de vouloir utiliser) plusieurs applications en même temps ? *

(Par exemple : consulter un sms, puis basculer sur Google Map pour voir où est l'adresse indiquée dans le sms, puis revenir sur le sms pour répondre à la personne)

- Très souvent
- Souvent
- Quelques fois

Rarement

Jamais

Vous est t'il arrivé de vouloir déplacer du texte ou des fichiers entre ces applications ? *

(Par exemple, copier le texte d'un sms dans un mail, ou bien copier une adresse d'un site pour la coller sur Google Map)

Très souvent

Souvent

Quelques fois

Rarement

Jamais

Etes vous gêné par le fait qu'il n'y a de place que pour une seule fenêtre sur l'écran du téléphone ? *

Oui

Non

Aimeriez-vous pouvoir passer plus facilement d'une application à une autre ? *

(Par exemple sur un ordinateur on peut avoir plusieurs fenêtres ouvertes en même temps)

Oui

Non

Si vous avez répondu oui à la question précédente : Comment imaginez (ou aimeriez) vous que le passage d'une application à une autre se fasse ?

(si vous n'avez pas de suggestion vous pouvez passer la question)

Merci d'avoir répondu à ce questionnaire !!

Fourni par [Google Documents](#)

[Signaler un cas d'utilisation abusive](#) - [Conditions d'utilisation](#) - [Clauses additionnelles](#)

Document d'instructions pour l'expérience 3 de JerkTilts

Baglioni Mathias, Doctorant à Télécom ParisTech
Bureau C201-8, 46 rue Barrault 75013 Paris
baglioni@enst.fr

Expérience utilisateur

Prérequis

L'utilisateur doit posséder et utiliser quotidiennement un téléphone avec une version d'Android supérieure ou égale à la version 2.0.

Description de l'expérience

L'expérience consiste à enregistrer les valeurs mesurées par les accéléromètres d'un téléphone mobile pendant un usage courant de l'appareil.

L'application une fois lancée enregistre les mouvements de l'appareil ainsi que le nom de l'application en cours d'exécution et la model de téléphone. Toutes les interactions, dans l'application ne sont pas enregistrées (Annexe).

Les données seront récoltées et traitées de manière anonyme, et ne seront en aucun utilisées à des fins commerciales. Si l'utilisateur a le souhait qu'elle soit détruite après leur traitement il peut m'envoyer un mail (baglioni@telecom-paristech.fr) en me donnant le nom (prénom ou pseudo) utilisé pour l'expérience afin que je les efface.

Si l'utilisateur veut les résultats et analyses tirés de cette expérience il peut aussi les demander par mail.

Si l'utilisateur à une question particulière vis à vis de l'expérience il peut me contacter par mail.

Étape à suivre

L'utilisateur doit installer le package joint au mail. Pour cela il peut utiliser deux méthodes :

- Ouvrir le mail directement avec son client mail mobile. Cliquer sur la pièce jointe (TwistService.apk) et cliquer sur le bouton installer.
- Ouvrir le mail sur son ordinateur. Enregistrer la pièce jointe (TwistService.apk) dans un lieu facile d'accès (ex. sur le bureau). Ensuite brancher son téléphone et le mettre en mode clef USB. Copier le fichier sur le téléphone. Un fois copier l'utilisateur peut alors débrancher son téléphone (après avoir désactivé le mode clef USB). Enfin il doit accéder au fichier par un navigateur (ex. Astro est gratuit sur l'android market) cliquer dessus pour l'installer.

Une fois l'application installée il suffit de la lancer depuis le navigateur d'application.

L'utilisateur doit avoir une application qui ressemble à l'image Fig. 1 a). L'utilisateur doit ensuite entrer son nom, prénom ou pseudo et appuyer sur le bouton « Start Service ». L'application se ferme et une notification apparaît dans la barre des événements en haut de l'écran comme encadré en rouge sur l'image Fig1. b) indiquant que le service fonctionne.



Fig. 1 : Captures d'écran au cours de l'expérience.

Au bout de 24h le service se termine et une nouvelle notification (sonnerie + vibration) apparaît (Fig. 1 c)). L'utilisateur doit alors ouvrir le panneau des notifications et cliquer sur la notification « Sensor Log ». L'application TwiService s'ouvre à nouveau et le bouton « Send to FTP » n'est plus grisé.

L'utilisateur doit s'assurer d'avoir activé le wifi et d'être connecté à un réseau wifi permettant l'accès à internet. Il doit alors cliquer sur le bouton « Send to FTP » afin d'envoyer les fichiers de données. Une boîte de dialogue de chargement apparaît signalant que la transmission du fichier est en cours Fig. 1 d). Une fois la transmission terminée le dialogue disparaît et l'application se ferme automatiquement.

L'utilisateur peut alors désinstaller l'application.

Recommandations

Il est important de suivre les recommandations suivantes afin que l'expérience se déroule correctement :

- L'utilisateur ne doit pas éteindre son dispositif pendant la durée de l'expérience. (il peut naturellement mettre le téléphone en veille)
- L'utilisateur doit prendre soin de garder la batterie de son téléphone suffisamment chargée afin qu'il ne s'éteigne pas tout seul.
- L'utilisateur doit utiliser son téléphone de la manière la plus habituelle possible. C'est à dire essayer de ne pas changer ses habitudes d'utilisation.
- Enfin l'utilisateur ne doit pas utiliser son téléphone en mode clef USB pendant toute la durée de l'expérience (sauf lors de l'installation).
- Eviter de faire l'expérience pendant les jours où vous n'utilisez pas ou très peu votre téléphone.
- L'expérience doit être terminée au plus tard le lundi 18 avril à minuit.

Annexe

Exemple de fichier de données :

```
ben92;com.sec.android.app.twlauncher/.Launcher;0.2681506;7.0485296;6.9719152;4540
ben92;com.sec.android.app.twlauncher/.Launcher;0.2681506;7.0485296;6.9719152;4599
ben92;com.sec.android.app.twlauncher/.Launcher;0.2681506;7.0485296;6.9719152;4659
```

Schéma du module d'animation du Flick

Le schéma suivant représente le module Flick Animator utilisé pour le calcul de la vitesse de défilement pour les techniques présentées dans le chapitre 3.

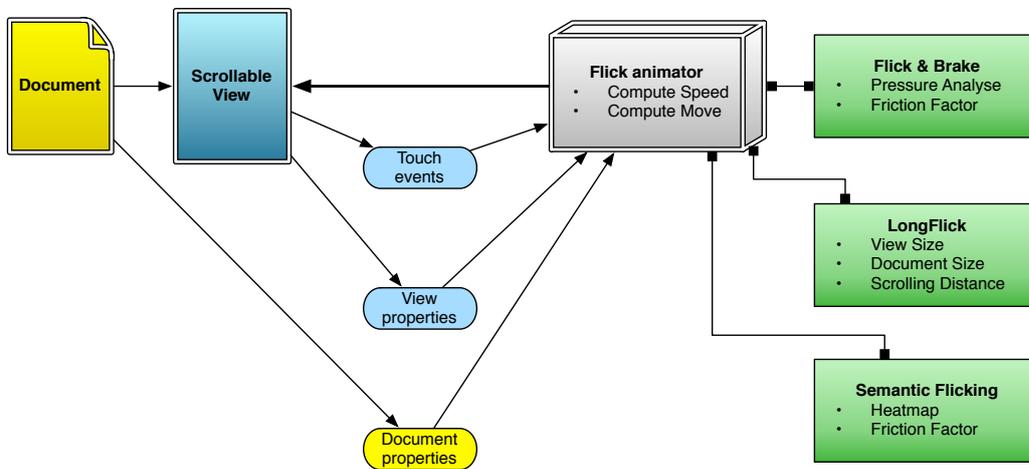


Schéma du module de reconnaissance de geste

Le schéma suivant représente le module de reconnaissance de geste utilisé pour les techniques présentées dans le chapitre 4.

