



**HAL**  
open science

# Conception et Evaluation de Technique d'Interaction pour Dispositifs Mobiles

Anne Roudaut

► **To cite this version:**

Anne Roudaut. Conception et Evaluation de Technique d'Interaction pour Dispositifs Mobiles. domain\_other. Télécom ParisTech, 2010. Français. NNT : . pastel-00005914

**HAL Id: pastel-00005914**

**<https://pastel.hal.science/pastel-00005914>**

Submitted on 12 Jul 2010

**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.



# Thèse

présentée pour obtenir le grade de Docteur  
de Télécom ParisTech

Spécialité : Informatique et réseaux

**Anne Roudaut**

Conception et Evaluation de Techniques d'Interaction  
pour Dispositifs Mobiles

Soutenue le 5 février 2010 devant le jury composé de

Yves Guiard	Président
Michel Beaudouin-Lafon	Rapporteurs
Joëlle Coutaz	
Monique Noirhomme	Examineurs
Nicolas Roussel	
Eric Lecolinet	Directeur de thèse
Johann Daigremont	Invité

Thèse préparée au département Informatique et Réseaux de Télécom  
ParisTech en collaboration avec Alcatel-Lucent Bell Labs dans le cadre du  
Laboratoire commun UBIMEDIA





## Remerciements

Ces travaux de thèse se sont déroulés grâce à et avec des personnes formidables. C'est donc avec émotion que j'écris cette page de remerciements.

Je remercie en premier lieu Eric Lecolinet, mon directeur de thèse, qui a toujours montré de l'intérêt pour mon travail et qui a su me faire confiance. Son expérience, son écoute, sa patience, sa disponibilité et son humour font de lui un homme que j'apprécie sincèrement. Je serai à jamais reconnaissante envers lui pour m'avoir accueilli dans son équipe et m'avoir formé à ce métier passionnant.

Je remercie vivement mes rapporteurs, Michel Beaudouin-Lafond et Joëlle Coutaz. Je suis très honorée qu'ils aient accepté de lire attentivement et de juger mes travaux car ce sont des personnes que je respecte profondément pour leur parcours ainsi que pour leur qualité humaine. Je remercie particulièrement Joëlle Coutaz qui m'a en premier donné goût à l'interaction homme machine et m'a donné envie de continuer dans la recherche. Je remercie également mes examinateurs Yves Guiard, Monique Noirhomme, Nicolas Roussel et mon invité Johann Daigremont d'avoir accepté d'évaluer mon travail, de s'être déplacés pour m'écouter et de m'avoir fait des remarques constructives. Merci à mon président de jury, Yves Guiard, pour le professionnalisme et l'humour dont il a fait preuve pendant la soutenance, ainsi que toutes les connaissances qu'il m'a transmises depuis son arrivée dans l'équipe.

Merci aux membres de Bell Labs Alcatel-Lucent avec qui j'ai collaboré durant ma thèse : Bruno Aidan, Ronan Daniellou, Johann Draigremont, Bruno Legars, Patrick Legrand, Myriam Ribiere, et tous ceux que j'ai pu oublier. Ils m'ont toujours encouragé à aller au bout de mes idées et leurs conseils avisés m'ont aidé à y parvenir. Leur ouverture d'esprit, leur enthousiasme et leur jovialité aura, selon moi, rendu cette collaboration aussi enrichissante d'un point de vue professionnel que personnel.

Merci aux personnes de Telecom Paris Tech, et surtout à ceux qui peuplent (ou ont peuplé) les couloirs du 2<sup>ème</sup> étage du bâtiment C, en particulier Baptiste Csernel, Billel Gueni, Tahir Muhammad qui m'ont chaleureusement accueilli au début de ma thèse, et Stéphane Huot avec qui j'ai pu me lancer sur mon premier projet.

Je ne saurais pas quels mots choisir pour remercier les personnes du "Budufoduco" (même s'ils n'ont pas tous logé dans ce BUreau DU FOND DU COuloir) : Gilles Bailly, Sylvain Malacria, Mathias Baglioni, Sylvain Ferrandiz, Aurélien Tabard, Halla Olafsdottir et Thomas Pietrzak. Ils sont devenus au fil des années de réels amis et grâce à eux ma thèse restera pour moi un moment inoubliable. Nos discussions (parfois fructueuses, parfois moins), leur aide, leur bonne humeur, leur écoute, leur délire, leur "bouderie", leur personnalité m'ont réconforté et m'ont motivé à plusieurs reprises. Hayette Soussou, notre secrétaire favorite, fait également

partie de ces personnes qui me sont chères. J'ai rarement rencontré une personne si enjouée et qui sait faire partager sa bonne humeur. Elle m'a souvent redonné le moral pendant les étapes difficiles et j'ai trouvé en elle une réelle amie. Je pense avoir eu une chance rare de tomber sur toutes ces personnes, et maintenant que je pars vers de nouveaux horizons je regrette déjà leur présence à mes côtés. J'espère que les nouveaux arrivants Maurice Adjanor, Simon Perrault et Dong-bach Vo sauront profiter et maintenir cette ambiance privilégiée qui existe entre les membres de cette équipe.

Je profite également de cette occasion pour dire à mes amis à quel point ils comptent pour moi. En particulier merci à Aphrodita, Claire, Delphine, Sarah, Thida, Virginie. Notre amitié a plus de 10 ans (20 ans avec Sarah, le cours élémentaire semble si loin !). Elles ont toujours été présentes à mes côtés pour me reconforter dans les moments durs et pour partager mes joies, et je sais que dans 10 ans je pourrai toujours compter sur elles. C'est pourquoi peu de mots peuvent exprimer la joie que j'ai de les avoir dans ma vie. Je dois remercier plus particulièrement Delphine d'avoir joué la correctrice orthographique mais aussi d'être une personne toujours agréable, motivée, courageuse, gaffeuse et pleine d'humour ! J'ai également une pensée pour tous ceux que je vois moins souvent mais qui sont toujours dans mon cœur : Adja, Azzedine, Brunelle, Emilie, Georgette, Hoang, Mamadou, Rachida et Sonia. Enfin un grand merci à Zohra qui est une personne que j'admire et que j'apprécie profondément pour ses valeurs humaines. Même si la distance nous sépare, elle est et restera toujours une amie sincère à mes yeux.

Je vais conclure par les personnes qui me sont le plus cher : ma famille. Les mots ne sont pas assez forts pour dire combien je les aime, en particulier mes parents. Je leur dois tout, ils m'ont permis de me construire, d'être curieuse, enjouée, d'aller au bout de mes ambitions, et de devenir la personne que je souhaitais être. C'est grâce à eux que j'ai pu concrétiser mes rêves et je n'aurais pas pu avoir de meilleurs modèles dans la vie. Merci à mon frère Olivier dont je suis très fière et qui est le meilleur frère du monde !

Merci également à Françoise, Pierre et Sophie d'avoir amené les couleurs de la Bretagne et de m'avoir écouté attentivement. Merci à ceux qui n'ont pas pu venir mais qui je suis sûre étaient avec moi en pensée. Pour finir je voudrais remercier mes grands parents : mon "Grand-Père" François, ma "Grand-Mère" Marie-Pierre, et ma "Mamie" Simone pour le bonheur que j'ai de les avoir encore à mes côtés, et j'ai également une pensée pour mon "Papi" François que je n'ai pas beaucoup connu mais qui a toujours été dans mon cœur.

**Table des matières**

<b>Remerciements</b>	<b>3</b>
<b>Table des matières</b>	<b>5</b>
<b>CHAPITRE 1 : INTRODUCTION</b>	<b>9</b>
<b>1. Constats</b>	<b>11</b>
1.1 Evolution des dispositifs mobiles	11
1.2 Dispositifs mobiles et ordinateurs standard	12
1.3 Interaction à une main sur dispositifs mobiles	13
<b>2. Contributions</b>	<b>14</b>
2.1 Espace de conception	14
2.2 Conception et évaluation de techniques d'interaction	14
<b>3. Organisation du mémoire</b>	<b>15</b>
<b>CHAPITRE 2 : ESPACE DE CONCEPTION</b>	<b>17</b>
<b>1. Introduction</b>	<b>19</b>
<b>2. Espace problème : contraintes des dispositifs mobiles</b>	<b>19</b>
2.1 Limitations matérielles	21
2.2 Prises en compte des capacités humaines	28
2.3 Contraintes dues à l'environnement	33
2.4 Synthèse	33
<b>3. Taxonomie des techniques pour dispositifs mobiles</b>	<b>34</b>
3.1 Techniques d'interaction	35
3.2 Techniques de visualisation	46
<b>4. Conclusion</b>	<b>51</b>
4.1 Conception de techniques pour dispositifs mobiles	51
4.2 Conception de techniques pour d'autres plates-formes	53

<b>CHAPITRE 3 : POINTAGE SUR ECRANS TACTILES</b>	<b>55</b>
<b>1. Introduction</b>	<b>57</b>
<b>2. Etat de l'art</b>	<b>57</b>
2.1 Techniques à curseur décalé	58
2.2 Techniques de franchissement	60
2.3 Techniques de pavage	61
2.4 Techniques à base de gestes sémaphoriques	63
2.5 Pointage et entrée de texte	65
2.6 Synthèse de l'état de l'art	66
<b>3. TapTap et MagStick</b>	<b>69</b>
3.1 Présentation	69
3.2 Propriétés	71
3.3 Implémentation	74
<b>4. Expérimentation</b>	<b>75</b>
4.1 Plan d'expérimentation	75
4.2 Résultats	77
4.3 Discussion	81
<b>5. Intégration</b>	<b>86</b>
<b>6. Conclusion</b>	<b>90</b>
<b>CHAPITRE 4 : MENUS SUR ECRANS TACTILES</b>	<b>93</b>
<b>1. Introduction</b>	<b>95</b>
<b>2. Etat de l'art</b>	<b>96</b>
2.1 Menus à base de gestes déictiques	96
2.2 Menus linéaires gestuels	97
2.3 Menus circulaires gestuels	99
2.4 Synthèse de l'état de l'art	103
<b>3. LeafMenu</b>	<b>105</b>
3.1 Présentation	105
3.2 Propriétés	107
3.3 Implémentation	110

<b>4. Expérimentation</b>	<b>111</b>
4.1 Plan d'expérimentation	111
4.2 Résultats et discussion	113
<b>5. Intégration</b>	<b>116</b>
<b>6. Conclusion</b>	<b>118</b>

## **CHAPITRE 5 : GESTES PHYSIQUES SUR ECRANS TACTILES** **119**

---

<b>1. Introduction</b>	<b>121</b>
<b>2. Etat de l'art</b>	<b>122</b>
2.1 Gestes de lancer	122
2.2 Gestes oscillatoires	123
2.3 Gestes utilisant la forme du doigt	125
2.4 Gestes multi-doigts	126
2.5 Synthèse de l'état de l'art	127
<b>3. MicroRolls</b>	<b>128</b>
3.1 Roulements versus glissements	128
3.2 Algorithme de reconnaissance	130
3.3 Application des MicroRolls	136
<b>4. Expérimentations</b>	<b>137</b>
4.1 Expérience 1	138
4.2 Expérience 2	142
4.3 Préférences utilisateurs	145
<b>5. Intégration</b>	<b>145</b>
<b>6. Conclusion</b>	<b>147</b>

## **CHAPITRE 6 : GESTES INCARNES** **149**

---

<b>1. Introduction</b>	<b>151</b>
<b>2. Etat de l'art</b>	<b>151</b>
2.1 Capteurs de mouvements	152

2.2 Gestes impulsifs	154
2.3 Gestes fluides	155
2.4 Synthèse de l'état de l'art	157
<b>3. TimeTilt</b>	<b>157</b>
3.1 Mode lenticulaire	158
3.2 Mode superposition	159
3.3 Réalisation	160
<b>4. Evaluation</b>	<b>162</b>
4.1 Plan d'expérimentation	162
4.2 Recherche des primitives	163
4.3 Reconnaissance	164
<b>5. Intégration</b>	<b>165</b>
<b>6. Conclusion</b>	<b>168</b>
<b>CHAPITRE 7 : CONCLUSION</b>	<b>171</b>
<b>1. Contributions</b>	<b>173</b>
1.1 Espace de conception	173
1.2 Nouvelles techniques d'interaction	173
<b>2. Perspectives</b>	<b>175</b>
2.1 Perspectives à court terme	175
2.2 Perspectives à moyen terme	176
2.3 Perspectives à long terme	178
<b>Table des figures</b>	<b>180</b>
<b>Publications personnelles</b>	<b>188</b>
<b>Bibliographie</b>	<b>188</b>

# Introduction



"En 2008, 60% de la population possède un téléphone portable. Il y a aujourd'hui dans le monde trois fois plus de dispositifs mobiles que d'ordinateurs conventionnels"

Rapport de l'International Telecommunication Union (2008)



<b>1. Constats</b>	<b>11</b>
1.1 Evolution des dispositifs mobiles	11
1.2 Dispositifs mobiles et ordinateurs standard	11
1.3 Interaction à une main sur dispositifs mobiles	12
<b>2. Contributions</b>	<b>13</b>
2.1 Espace de conception	14
2.2 Conception et évaluation de techniques d'interaction	14
<b>3. Organisation du mémoire</b>	<b>14</b>

## 1. Constats

### 1.1 Evolution des dispositifs mobiles

Plus de 60 ans se sont écoulés depuis l'apparition du premier ordinateur et il devait être difficile à l'époque d'imaginer l'ampleur de cette révolution. Aujourd'hui 60% de la population mondiale possède dans sa poche 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 tout premier ordinateur (l'ENIAC en 1943). Ces dispositifs mobiles ne datent pas de si longtemps : le premier, le Motorola DynaTac, fut proposé en 1983. Mais leur évolution au cours de ces 27 années est impressionnante (FIG. 1) : ces dispositifs sont désormais légers, possèdent des écrans haute résolution ainsi que de nouveaux périphériques d'entrée (écran tactile, capteurs, caméra etc.). Ils offrent de plus une large puissance de calcul, de grandes capacités mémoire et une vaste gamme de moyens de communication possibles (3G, Edge, Wifi, Bluetooth, etc.).

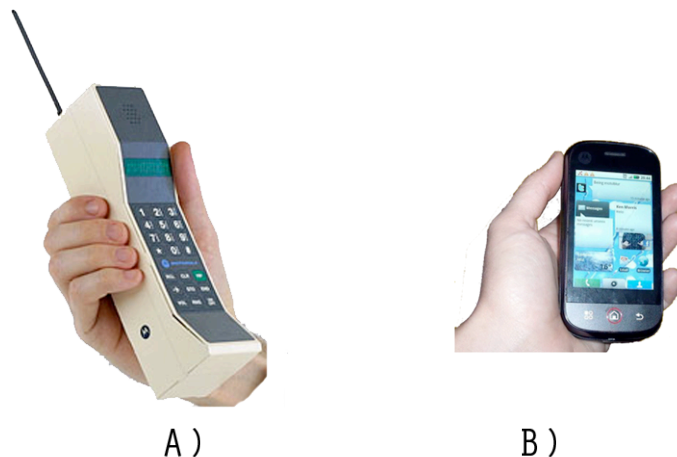


Figure 1. A) Motorola DynaTac (1983) ; B) Motorola Dext (2009).

Si les dispositifs mobiles connaissent un tel engouement, c'est aussi de par l'aspect social et économique : pour l'identité de groupe car ne pas avoir de téléphone portable ou un modèle trop ancien peut être mal vu dans un groupe et, dans certains pays, c'est un moyen peu onéreux d'accéder à internet [ALZOUMA 09]. Il est de plus léger (pour l'avoir toujours à portée de main) et devient modulaire : les modèles récents permettent de renouveler facilement la gamme de fonctionnalités (grâce aux supports de vente d'applications, comme l'AppStore).

Indubitablement le téléphone portable ne sert donc plus seulement à téléphoner. Il joue le rôle d'un outil multi-fonctions qui permet de communiquer, s'informer, jouer, consommer, partager, interagir, etc. L'analogie avec le couteau suisse est renforcée lorsque l'on voit apparaître des utilisations détournées des téléphones [SALOVAARA 09] pouvant faire office de

réveil, montre, lampe, miroir ou même périscope (en utilisant la caméra vidéo pour visualiser des endroits difficiles à atteindre). Cette richesse d'usage n'échappe ni aux fabricants ni aux chercheurs qui proposent régulièrement de nouvelles fonctionnalités, de nouvelles technologies ou de nouveaux modèles pour accroître encore le potentiel de ces dispositifs.

## 1.2 Dispositifs mobiles et ordinateurs standard

Si le marché des dispositifs mobiles est en pleine effervescence, celui des ordinateurs conventionnels n'a pas été délaissé pour autant et ceci pour une simple raison : un dispositif mobile n'est pas un ordinateur de bureau miniaturisé et ne remplace donc pas ce dernier.

"La forme d'un ordinateur reflète son contexte physique d'utilisation. Un ordinateur de bureau n'a pas la même forme qu'un PDA, et ne peut donc permettre les mêmes formes d'usage." [BEAUDOUIN-LAFON 00]

Les formes d'usage sont en effet sensiblement différentes selon les plateformes. L'utilisation d'un ordinateur de bureau se fait souvent sur une assez longue durée, en position assise et l'utilisateur possède des périphériques d'interaction spécifiques (une souris, un clavier, un écran relativement grand) pour réaliser sa tâche. Les dispositifs mobiles sont utilisés dans des contextes variés (chez soi, dans les transports ou dans la rue, etc.) et leur utilisation est plus brève (OULASVIRTA 05), entre autres car l'utilisateur peut être interrompu par une action extérieure.

De plus, les dispositifs mobiles de dernière génération, tels que les Smartphones<sup>1</sup> auxquels nous nous intéressons dans cette étude, n'ont pas les mêmes périphériques d'interaction (il n'y a pas de souris mais un écran tactile, le clavier est absent ou réduit et l'écran est petit). Certaines tâches sont par conséquent plus difficiles à effectuer, comme le traitement de texte qui s'avère délicat avec un clavier physique réduit, et plus encore avec un clavier virtuel sur l'écran tactile. Inversement certaines tâches sont spécifiques aux mobiles, comme dans les applications contextuelles où l'utilisateur reçoit des informations sur le lieu où il se trouve grâce au GPS, ou celles qui exploitent les capteurs de mouvement de l'appareil.

Une autre spécificité de ces mobiles, qui découle de l'usage en conditions de mobilité, est que l'utilisateur n'a pas toujours les deux mains disponibles pour interagir. Des études [PASCOE 00] [KARLSON 07] montrent notamment que les utilisateurs préfèrent souvent utiliser une seule main pour manipuler leur dispositif. Cette situation se produit dans divers cas de la vie courante : en portant un sac, en se tenant à un support pour ne pas perdre l'équilibre dans les

---

<sup>1</sup> Téléphones intelligents, "Pdaphones" ou "ordiphones" qui fournissent les fonctionnalités d'agenda, de calendrier, de navigation web, de consultation de courrier, de messagerie instantanée, de GPS, etc. (WIKIPEDIA)

transports en commun ou en marchant dans la rue. Ceci rend l'utilisation d'un stylet impossible, une seule main étant disponible et l'interaction se faisant alors au pouce. Ce type d'usage engendre diverses contraintes entre autres car le pouce cache alors une partie de l'écran tactile.

Les ordinateurs de bureau et les dispositifs mobiles coexistent donc, mais pour des usages et des types d'interaction différents.

### 1.3 Interaction à une main sur dispositifs mobiles

Les premiers Smartphones proposaient des interfaces dérivant du modèle d'interaction WIMP (fenêtres, icônes, souris, pointeur, ou Windows, Icons, Mouse, Pointer) utilisé dans les interfaces des ordinateurs de bureau. Mais ce style n'est pas adapté à l'interaction à une main sur mobiles. Nous avons par exemple l'habitude d'interagir avec des cibles relativement petites sur nos PCs car la souris nous apporte la précision nécessaire à leur sélection. Sur un dispositif mobile, l'interaction aux doigts, plus larges que le curseur d'une souris, diminue très fortement la précision et rend les cibles difficilement sélectionnables. Pour répondre à ce problème, certaines cibles de l'iPhone sont par exemple agrandies, mais c'est au détriment du nombre de cibles affichées à l'écran.

Cet exemple, parmi bien d'autres, montre que les interfaces de mobiles ne peuvent pas être de simples adaptations de celles des ordinateurs conventionnels. De plus les alternatives existantes tendent à résoudre certains problèmes au détriment d'autres facteurs (comme consommer davantage de place à l'écran) du fait de la faible *bande passante interactionnelle* de ces dispositifs (petit écran, absence de clavier physique, absence de souris, interaction aux doigts etc.). Il n'est donc pas aisé de créer des techniques réellement adaptées aux dispositifs mobiles.

Cette thèse se place dans ce contexte. Elle a pour objectif de mettre en évidence les contraintes des mobiles et de créer de nouvelles formes d'interaction spécifiquement adaptées à ces dispositifs. Dans ce cadre, nous nous limitons aux techniques d'interaction à une main sur l'écran tactile et celles utilisant des capteurs de mouvements. Nous ne traitons pas en détail l'entrée de texte, l'entrée vocale, le retour audio et le retour haptique.

## 2. Contributions

Nous présentons ci-après les contributions de cette thèse qui a été réalisée dans le cadre du laboratoire commun UBIMEDIA entre l'Institut Télécom et Bell Labs Alcatel-Lucent. Nous avons notamment travaillé avec une équipe du département Social Communications qui développe des applications de réseaux sociaux sur dispositifs mobiles. Dans cette collaboration, notre but est d'améliorer les interfaces des dispositifs mobiles existants sur le marché (de type Smartphone), nos travaux étant ensuite utilisés par nos partenaires dans leurs applications.

### 2.1 Espace de conception

Nous proposons tout d'abord un espace de conception de l'interaction à une main sur dispositifs mobiles. Cet espace se décompose en deux parties :

- Un *espace problème* qui dresse la liste des contraintes spécifiques aux dispositifs mobiles. Il expose les limitations de ces dispositifs qu'il faut prendre en compte lors de la création d'une technique. Cet espace a donc pour objectif de faire office de guide lors de la conception et de l'évaluation d'une technique.
- Une *taxonomie* des techniques de visualisation et d'interaction sur dispositifs mobiles qui présente les solutions déjà proposées pour résoudre certaines contraintes identifiées dans l'espace problème. Elle permet ainsi de visualiser les lacunes de l'état de l'art, ce qui peut suggérer des voies intéressantes à explorer.

### 2.2 Conception et évaluation de techniques d'interaction

En s'appuyant sur notre espace problème, nous avons conçu, développé et évalué cinq nouvelles techniques d'interaction.

**TapTap** et **MagStick** sont deux nouvelles techniques qui améliorent la sélection de petites cibles au pouce sur un écran tactile. TapTap est basée sur une stratégie de multiplexage temporel de type "double clic" où le premier tap spécifie un focus d'intérêt de l'interface et le second tap sélectionne une cible dans la zone de focus qui a été agrandie. MagStick est basée sur l'utilisation d'un bâtonnet télescopique qui permet d'atteindre les cibles en s'éloignant d'elles. TapTap et MagStick répondent aux contraintes de l'interaction au doigt et en particulier aux problèmes d'occultation, d'imprécision et aux difficultés d'accès aux bords de l'écran. Une étude comparative a montré que ces deux techniques sont plus performantes que celles proposées dans l'état de l'art.

Le **LeafMenu** est une technique de menu qui enrichit les menus linéaires par des raccourcis gestuels sur écran tactile. Cette technique offre une solution de substitution aux raccourcis claviers et réduit les erreurs des menus linéaires classiques. Les gestes du LeafMenu peuvent être exécutés près des bords de l'écran grâce à l'organisation symétrique du menu. Comme nous le verrons, une expérimentation a été menée pour évaluer l'apprentissage des gestes du LeafMenu. Ses résultats montrent que les gestes sont appris rapidement.

Les **MicroRolls** sont des gestes de roulement du pouce sur un écran tactile. Ils comblent certaines limitations des écrans tactiles en enrichissant le vocabulaire interactionnel de l'utilisateur. Les MicroRolls sont compatibles avec d'autres gestes comme ceux de lancer du doigt. Ils peuvent de plus être combinés avec d'autres techniques : RollTap associe les MicroRolls avec la technique de précision TapTap et RollMark est un Marking menu à base de MicroRolls. Une première expérience a permis de récolter un échantillon significatif de ces gestes afin de construire un algorithme de reconnaissance puis d'évaluer ses performances. Deux expériences contrôlées montrent que les MicroRolls sont une alternative efficace aux techniques habituellement utilisées pour une tâche de copier coller.

Enfin, **TimeTilt** est une technique qui facilite la navigation entre plusieurs vues. Elle utilise l'inclinaison du dispositif (détectée avec un accéléromètre) pour permettre à l'utilisateur de se déplacer dans les différentes vues. TimeTilt propose également d'autres types de gestes comme le tap à l'arrière du dispositif. TimeTilt exploite donc la possibilité d'interagir avec des doigts qui sont habituellement inutilisés. Une expérience a permis de récolter un échantillon significatif de ces nouveaux gestes afin de construire un algorithme de reconnaissance, puis d'évaluer ses performances.

### 3. Organisation du mémoire

Ce manuscrit est organisé en cinq chapitres reflétant les contributions.

- Le chapitre 2 présente l'espace de conception de l'interaction à une main sur dispositifs mobiles. Nous décrivons ses fondements, lesquels sont issus des travaux existants.
- Le chapitre 3 est consacré à l'exploration des techniques de pointage de petites cibles sur écran tactile. Après avoir détaillé l'état de l'art, nous présentons TapTap et MagStick. Nous terminons par l'évaluation comparative de nos techniques avec quatre autres techniques récentes ou proposées précédemment.
- Le chapitre 4 est dédié à l'exploration des gestes sur écran tactile et en particulier pour le cas de la sélection de commandes dans un menu. Nous identifions l'état de l'art et

présentons le LeafMenu. Nous décrivons ensuite une évaluation expérimentale visant à vérifier l'apprentissage et l'utilisation des raccourcis gestuels.

- Le chapitre 5 concerne l'exploration de nouveaux gestes sur écran tactile. Après avoir détaillé l'état de l'art, nous présentons les MicroRolls. Les résultats d'une étude de collecte de gestes, ainsi que deux évaluations expérimentales comparant notre technique aux alternatives existantes, seront ensuite exposées.
- Le chapitre 6 a trait à l'utilisation des gestes de la main dans l'espace grâce à de nouveaux dispositifs tels que les accéléromètres. Nous détaillons l'état de l'art et présentons TimeTilt. Nous décrivons ensuite une étude visant à collecter les gestes utilisés dans TimeTilt pour construire un algorithme de reconnaissance.

Enfin le dernier chapitre conclut ce mémoire et présente les perspectives de ces travaux.

# Espace de conception



"En France, 97% des 16-60 ans ont un téléphone portable. 48% l'ont changé au cours de ces 12 derniers mois et 36% d'entre eux ont choisi un modèle avec écran tactile"

Observatoire de l'Internet mobile (2009)



<b>1. Introduction</b>	<b>19</b>
2.1 Limitations matérielles	21
2.2 Prises en compte des capacités humaines	28
2.3 Contraintes dues à l'environnement	33
2.4 Synthèse	33
<b>3. Taxonomie des techniques pour dispositifs mobiles</b>	<b>34</b>
3.1 Techniques d'interaction	35
3.2 Techniques de visualisation	46
<b>4. Conclusion</b>	<b>51</b>
4.1 Conception de techniques pour dispositifs mobiles	51
4.2 Conception de techniques pour d'autres plates-formes	53

## 1. Introduction

Face aux nombreux travaux existants, il est relativement difficile pour un concepteur de créer ou d'identifier les techniques qui sont les mieux adaptées aux dispositifs mobiles. Les taxonomies peuvent nous aider en organisant les solutions proposées et en mettant en évidence les aspects moins étudiés. Les classifications existantes ne couvrent cependant que certains aspects de l'interaction sur dispositifs mobiles. Par exemple les taxonomies de Poirier (POIRIER 04) et MacKenzie (MACKENZIE 02) font état des méthodes d'entrée de texte, celle de Baffoun (BAFFOUN 06) classe les techniques de visualisation sur petits écrans et celles de Baglioni (BAGLIONI 09) et Ballagas (BALLAGAS 06) présentent les techniques impliquant des gestes en trois dimensions.

Les taxonomies existantes sont donc relativement nombreuses mais éparses. Pour fournir une vue d'ensemble de l'interaction sur dispositifs mobiles, et ce afin d'aider la conception de nouvelles techniques, nous proposons un espace de conception basé sur une étude préliminaire (ROUDAUT 07). Notre espace de conception se décompose en deux parties :

- Un *espace problème* qui organise les contraintes entrant en jeu lors de l'utilisation d'un dispositif mobile. Il décrit les limitations matérielles de ces dispositifs, les éléments à prendre en compte par rapport aux capacités humaines et les contraintes dues à l'environnement. Cet espace permet ainsi de guider les concepteurs en les aidant à mettre en évidence les atouts et les faiblesses d'une technique en cours de conception.
- Une *taxonomie* des techniques existantes sur dispositifs mobiles. Elle présente les solutions répondant aux contraintes identifiées dans l'espace problème et les classe selon la nature des gestes effectués, qui peuvent être déictiques, sémaphoriques ou physiques. Elle permet ainsi de mettre en évidence les lacunes de l'état de l'art, ce qui peut suggérer des voies intéressantes à explorer.

Nous présentons tout d'abord l'espace problème, suivi de la taxonomie des techniques de visualisation et d'interaction sur dispositifs mobiles. Nous terminons avec une synthèse qui montre comment cet espace peut aider lors de la conception de nouvelles techniques.

## 2. Espace problème : contraintes des dispositifs mobiles

Selon Norman (NORMAN 86), la réalisation d'une tâche par l'utilisateur se fait selon un cycle action/perception : l'utilisateur perçoit l'état du système et l'évalue pour décider de l'action nécessaire à l'atteinte de son objectif. Ensuite, l'état du système change, ce qui entraîne un nouveau cycle. L'exécution de ce cycle s'appuie sur l'utilisation d'un ensemble de dispositifs d'entrée et de sortie et ces opérations peuvent être effectuées grâce à diverses *modalités*.

L'approche des ergonomes et des psychologues est d'associer la modalité aux capacités de perception et d'action du sujet humain : le sens visuel, auditif et tactile (haptique) pour la perception (les sens olfactif et gustatif sont encore peu exploités), et le mouvement et la voix pour l'action. Dans notre cas d'étude, la modalité de mouvement correspond aux gestes des doigts sur un clavier physique ou sur un écran tactile (mouvements en deux dimensions) et ceux du dispositif dans l'espace pouvant être détectés par divers capteurs (mouvements en trois dimensions). Nous utilisons cette définition pour dresser la liste des contraintes propres aux dispositifs mobiles, que nous organisons en trois catégories : les limitations matérielles, les éléments à prendre en compte par rapport aux capacités humaines et les contraintes dues à l'environnement. Ce découpage est inspiré des travaux de Calvary (THEVENIN 99) qui définit le contexte d'interaction par le triplet "plate-forme, utilisateur, environnement physique ou social". La FIGURE 2 résume les points que nous abordons en détail dans la suite de la section.

	Sorties	Entrées
Limitations matérielles	Modalité visuelle : manque d'espace écran	Modalité mouvement : clavier réduit ou absent  Modalité mouvement : lacunes écrans tactiles  Modalité mouvement : interaction aux doigts
Prise en compte des capacités humaines	Modalité haptique et auditive : faible bande passante	Modalité mouvement : amplitude réduite des gestes des doigts  Modalité mouvement : amplitude réduite des gestes de la main
Contraintes dues à l'environnement	Modalité auditive : dépendante de l'environnement sonore	Modalité mouvement : gestes dans l'espace et environnement social  Modalité vocale : dépendante de l'environnement sonore

Figure 2. Espace problème : les contraintes des dispositifs mobiles.

## 2.1 Limitations matérielles

Les limitations matérielles sont au nombre de quatre. Nous nous intéressons tout d'abord aux sorties avec le manque d'espace écran. Nous présentons ensuite les contraintes en entrée dues à l'absence ou à la réduction de taille des claviers physiques, à l'interaction aux doigts et aux lacunes des écrans tactiles.

### *Manque d'espace écran*

Les écrans des mobiles sont de petite taille mais leur résolution numérique (en dpi ou "dot per inch") est généralement supérieure à celle des ordinateurs fixes ou portables. Par exemple, une analyse informelle de la taille des pixels de 7 ordinateurs portables ou fixes et de 7 mobiles (produits courants sur le marché<sup>2</sup>) nous indique qu'en moyenne un pixel d'écran d'ordinateur mesure 0.24 mm contre 0.17 mm pour un dispositif mobile. Un même objet graphique semblera donc plus petit sur un mobile. Le fait de tenir le dispositif dans la main permet d'avoir l'écran plus près des yeux, ce qui améliore la lisibilité (excepté pour les personnes atteintes de presbytie) et permet d'afficher des informations de tailles relativement petites. Cependant la taille de l'écran reste bornée par l'impératif de taille réduite que doit respecter un mobile. Cette réduction de l'espace d'affichage a des répercussions sur la visualisation, en particulier en ce qui concerne la lisibilité des informations pour les personnes âgées ou ayant un handicap visuel, mais aussi sur l'interaction. Par exemple, les menus contextuels se retrouvent plus souvent près des bords de l'écran que sur un ordinateur conventionnel et nécessitent donc des mécanismes de débrayage adéquats pour éviter de sortir de l'écran.

Deux solutions matérielles sont possibles pour pallier le manque d'espace écran. La première est le port de lunettes transparentes ou semi-transparentes couplées au dispositif mobile, mais ceci implique l'utilisation d'un outil supplémentaire qui peut avoir certains inconvénients (encombrant, intrusif lorsqu'il est porté dans des lieux publics, pouvant gêner la lisibilité entre les informations réelles et numériques). La seconde solution est d'utiliser un projecteur miniaturisé embarqué dans le dispositif. Cette technologie très récente permet d'afficher de l'information sur n'importe quelle surface pour augmenter l'espace visuel. Des techniques utilisant ce type de dispositif ont notamment déjà été proposées. Dans (BEARDSLEY 05) l'utilisateur peut retrouver des informations qu'il a annotées sur des objets du monde réel. Mistry et al. (MISTRY 09) proposent également de superposer des informations numériques sur les objets de la vie réel. Leur dispositif, que l'on porte autour du cou, est de plus capable de

---

<sup>2</sup> Ordinateurs (2008) : Sony Vaio, MacBook et MacBook pro, HP 8710p et 8510w, iMac 20' et 24'.

Dispositifs mobiles (2008) : HTC p3600 et Touch, iPhone, Nokia n95, Samsung f700, LG viewty, Sony w910.

détecter les mouvements des doigts de l'utilisateur dans l'espace, ce qui lui permet d'interagir avec les données projetées. Cao et al. (CAO 06) présentent une application qui permet de projeter une carte sur un mur et de la zoomer en se rapprochant du mur (FIG. 3.A). Enfin, Miyahara et al. (MIYAHARA 05) proposent une méthode pour faciliter l'échange de fichiers entre plusieurs dispositifs mobiles (FIG. 3.B). Notons cependant que ces techniques nécessitent une surface d'affichage supplémentaire, ce qui peut réduire la mobilité de l'utilisateur.



Figure 3. Utilisation d'un projecteur : A) pour partager des fichiers entre mobiles ; B) pour zoomer sur une carte en s'approchant du mur sur lequel elle est projetée.

### *Clavier physique réduit ou absent*

Certains dispositifs mobiles ne possèdent pas de clavier physique, celui-ci étant remplacé par un clavier virtuel sur l'écran. Malgré certains avantages, comme le fait de pouvoir facilement changer le type de clavier, les claviers virtuels présentent des inconvénients par rapport aux claviers physiques. En particulier, le retour tactile passif (sensation de toucher lors de l'appui sur une touche physique) permet d'entrer du texte plus efficacement (plus rapidement et avec moins d'erreurs). Ce constat a conduit certains constructeurs à envisager des solutions de substitution comme le retour tactile actif (vibrations ou effets sonores lors de l'appui sur l'écran). Des études (BREWSTER JAN 02) (BREWSTER 07) montrent notamment que cet apport diminue le nombre d'erreurs par rapport à un clavier virtuel classique.

Parfois le clavier physique est réduit, ce qui a des répercussions sur les méthodes d'entrée de texte qui sont plus lentes que les techniques classiques car les touches sont petites. Les claviers à 12 touches demandent de plus d'appuyer plusieurs fois sur une touche pour entrer une lettre. Par conséquent ces claviers sont surtout utilisés pour entrer des messages courts ou pour déclencher des fonctionnalités d'appel (entrer un numéro ou joindre un correspondant favori par un appui long sur une touche). Les autres boutons physiques, que l'on trouve sur les cotés du dispositif ou sous l'écran, permettent de contrôler quelques fonctionnalités spécifiques telles que gérer le volume sonore, appeler/raccrocher ou revenir sur le bureau.

Les boutons et claviers physiques peuvent constituer des ressources intéressantes pour améliorer l'interaction, mais en pratique ils sont peu nombreux et difficilement configurables car dédiés à des fonctionnalités standard. Les raccourcis claviers traditionnels sont donc

généralement absents sauf sur quelques modèles (Palm, Nokia) qui possèdent un clavier physique et des modificateurs (Ctrl, Shift). Lors de l'interaction à une main, il est toutefois impossible d'utiliser ces commandes car l'utilisateur ne peut pas maintenir deux touches appuyées simultanément. Les raccourcis claviers, même si les utilisateurs en connaissent peu, sont pourtant fréquemment utilisés sur les PCs, par exemple ceux pour couper, copier ou coller un item. De plus, même lors de l'interaction à deux mains, l'utilisation des raccourcis sur un clavier virtuel est difficile : 1) le clavier est affiché seulement sur demande de l'utilisateur afin d'économiser de l'espace écran ; 2) l'activation du clavier n'est pas toujours possible. C'est le cas dans certaines applications comme les gestionnaires de bibliothèque musicale, appareils photo, réveils ou explorateurs de fichiers où les champs d'entrée de texte sont remplacés par des listes déroulantes (pour éviter à l'utilisateur d'utiliser le clavier).

### *Interaction aux doigts*

Il existe deux types de technologies tactiles actuellement exploités dans les dispositifs mobiles : les écrans résistifs<sup>3</sup> détectent la pression d'un doigt ou d'un stylet, tandis que les écrans capacitifs<sup>4</sup> ne fonctionnent qu'avec le contact d'un objet conducteur (le doigt). Si les écrans résistifs permettent l'utilisation d'un stylet, cette solution n'est pas appropriée pour l'interaction à une seule main [PASCOE 00] [KARLSON 07]. Ainsi dans la majorité des cas, l'utilisateur interagit avec son pouce sur l'écran tactile, ce qui engendre plusieurs problèmes détaillés ci-après.

**Occultation.** Un écran tactile a la particularité d'être à la fois un périphérique de sortie et d'entrée, ce qui entraîne des problèmes d'occultation (FIG 4). Une étude (VOGEL 09) révèle qu'environ 47% d'un écran tactile de 12 pouces est caché par le stylet, la main et l'avant-bras. Un calcul approximatif de la surface d'un pouce montre qu'environ 34% de l'écran est occulté par le pouce sur l'écran d'un iPhone lorsque le pouce couvre toute la diagonale de l'écran. De plus il est difficile de changer la position du pouce pour dégager une zone : la pince, qu'il forme avec les doigts opposés permet de tenir le dispositif d'une façon stable mais contraint les mouvements possibles.

---

<sup>3</sup> Ecrans résistifs : deux couches conductrices sont tenues distantes par des cales d'espacement et placées sous une couche additionnelle en surface pour éviter les égratignures. Un courant électrique est induit dans les 2 couches conductrices. Lorsque l'utilisateur touche avec la pointe d'un stylet (ou d'un doigt), la pression exercée amorce un contact entre les 2 faces électrifiées des couches. La variation dans les champs électriques de ces deux faces permet de déterminer les coordonnées du point de contact. (WIKIPEDIA)

<sup>4</sup> Ecrans capacitifs : une couche qui accumule les charges est placée sur la plaque de verre du moniteur. Lorsque l'utilisateur touche la plaque avec son doigt, certaines de ces charges lui sont transférées. Les charges qui quittent la plaque capacitive créent un déficit quantifiable. Avec un capteur dans chacun des coins de la plaque, il est possible de déterminer les coordonnées du point de contact. (WIKIPEDIA)



Figure 4. Occultation de l'avant-bras et de la main sur une tablette tactile ou du pouce sur un petit écran tactile.

**Imprécision.** Un seul point est détecté lorsque l'on pose un doigt sur un écran tactile (hormis le cas de certaines technologies encore peu présentes sur le marché). Ce point est le barycentre de la surface du doigt en contact avec l'écran. Deux problèmes se posent alors : d'une part le point est caché par le pouce, d'autre part il est difficile de prédire sa position exacte. Benko et al. (BENKO 06) montrent notamment que les utilisateurs perçoivent ce point au bout de leur pouce alors qu'il se trouve en réalité au centre (FIG 5). Cette inexactitude induit des erreurs lors de la sélection des cibles puisque l'utilisateur tend à pointer plus bas qu'il ne le devrait. De plus, l'utilisateur effectue souvent des mouvements infimes du pouce lorsqu'il soulève son pouce de l'écran, ceci ayant pour effet de décaler le barycentre. Comme la sélection se fait au relâchement, l'utilisateur risque de rater la cible souhaitée ou d'en sélectionner une autre.



Figure 5. Erreur de perception de la position du pointeur sous le pouce. A) position supposée par les utilisateurs ; B) position réelle.

La précision est donc dépendante du rapport entre la taille du pointeur ( curseur de la souris ou point récupéré par l'écran tactile) et celle des cibles. Sur un ordinateur ce rapport est inférieur à 1 (la taille du curseur est plus petite que celle des cibles). Sur un mobile, ce rapport peut être respecté en affichant des cibles de taille supérieure ou égale à la taille du pouce. Parhi et al. (PARHI 06) montrent notamment que cette taille doit être égale à 9.2mm pour assurer une bonne précision lors du pointage au pouce, ce qui est contradictoire avec la nécessité d'économiser l'espace écran.

**Instabilité.** L'instabilité de l'interaction au pouce résulte de deux facteurs. D'un côté le pouce a une surface molle, ce qui peut produire des instabilités lorsque cette surface entre en contact avec une surface rigide comme un écran. Le pouce peut rouler sur lui-même ou s'écraser plus ou moins sur la surface, ce qui entraîne des saccades et par conséquent une imprécision lors d'un déplacement. D'autre part, cette imprécision est aussi due à la texture de l'écran. Les écrans capacitifs sont rigides alors que les écrans résistifs sont légèrement déformables, ce qui peut engendrer des variations de pression lorsque l'utilisateur appuie sur l'écran. Dans les deux cas, des soubresauts peuvent survenir lors du déplacement des doigts sur des écrans en verre ou en PVC (particulièrement avec des doigts moites). Ce problème est résolu dans le cas des grandes surfaces interactives (Microsoft Surface) en offrant une texture légèrement sableuse qui réduit l'adhérence des doigts sur la surface.

### *Lacunes des écrans tactiles*

Comme déjà mentionné dans (BENKO 06) (BUXTON 90) (OLWAL 08) (MACKENZIE 98), les interfaces requièrent des états interactionnels et des moyens de spécifier ces états. Le modèle d'interaction varie selon les systèmes interactifs (BUXTON 90), mais tous les ordinateurs classiques permettent à l'utilisateur de déplacer un curseur (par exemple avec la souris), déplacer un objet (via le bouton gauche de la souris), déplacer un contenu (grâce à une barre de défilement ou via la molette de la souris) ou activer un menu contextuel (via le bouton droit de la souris) (FIG. 6.A).

Les écrans tactiles utilisés actuellement dans les dispositifs mobiles ne permettent pas la richesse d'interaction des ordinateurs standard car seule l'absence ou la présence d'un contact est détectée. Cette propriété empêche les interactions à base de survol, comme les bulles d'aide, et force l'utilisateur à pointer directement les objets à l'écran puisqu'il n'y a pas de curseur. Il n'existe pas non plus de clic droit ou gauche, l'écran tactile pouvant être vu comme une souris où le clic gauche serait en permanence enclenché. Cette réduction des moyens de spécification des états interactionnels pousse les concepteurs à trouver des alternatives qui ne sont pas sans contraintes. Pour illustrer les différentes solutions proposées, nous présentons ci-après les modèles d'interaction des plates-formes courantes sur le marché des dispositifs mobiles. Ces modèles ne prennent pas en compte toutes les interactions possibles, le but principal étant de mettre en évidence les problèmes que peuvent poser les solutions existantes.



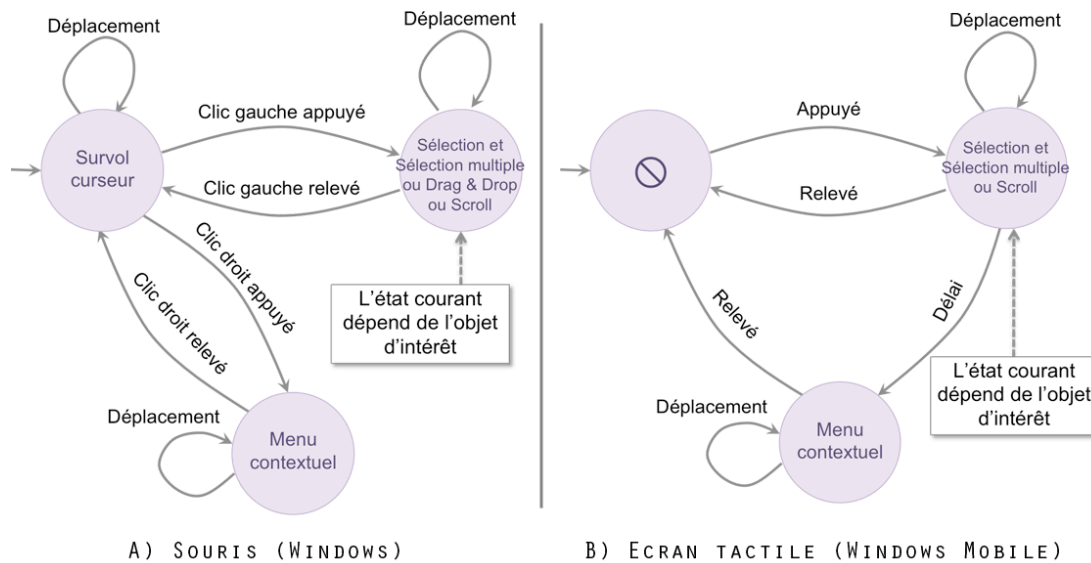


Figure 6. A) Modèle simplifié de l'interaction avec une souris sous le système d'exploitation Windows ; B) Modèle simplifié d'interaction sur l'écran tactile d'un dispositif Windows Mobile.

**Windows Mobile.** Dans le système d'exploitation Windows Mobile (FIG. 6.B), le clic droit (utilisé pour déclencher le menu contextuel) est remplacé par un appui long (environ 300ms) sur l'écran. L'activation du menu consomme donc plus de temps. Les déplacements du doigt sur l'écran servent à spécifier différents états selon l'objet sur lequel est effectuée l'action. Quand le déplacement se produit sur une barre de défilement il permet de faire défiler un contenu, ce qui peut être difficile puisque la barre est relativement petite. Lorsque le déplacement commence sur une cible ou un espace vide, il permet les sélections multiples. Le déplacement ne permet donc pas d'initier un Drag & Drop, car, les espaces vides n'étant pas toujours présents (ce qui est souvent le cas dans une liste), il est impossible de distinguer les opérations de Drag & Drop des sélections multiples.

**Android/Symbian/Palm.** Le modèle d'interaction sur l'écran tactile des dispositifs Android (FIG. 7.A), Symbian ou Palm possède la même forme que celui des systèmes Windows Mobile mais l'attribution des états interactionnels diverge : un appui long déclenche l'ouverture d'un menu dans certaines applications sur Android et Symbian alors qu'il active le Drag & Drop sur plusieurs modèles Palm et sur le bureau des dispositifs Android. Le modèle d'interaction n'est donc identique en fonction des applications, ce qui demande à l'utilisateur de mémoriser plusieurs modèles concurrents sur le même dispositif. De plus, il n'est pas possible d'effectuer les deux actions (ouverture du menu et Drag & Drop) au sein d'une même application.

Une autre différence est la suppression de la barre de défilement, trop imprécise pour la sélection aux doigts. Les concepteurs ont opté pour une solution popularisée par l'iPhone qui consiste à exécuter un geste de déplacement du doigt sur un composant graphique (comme

une liste) pour le faire défiler. Cette approche économise de l'espace écran mais peut créer des incompatibilités avec d'autres techniques d'interaction (typiquement celles utilisant les événements de déplacement comme le Drag & Drop). Par exemple il est difficile d'associer un Marking Menu (KURTENBACH 91) aux items d'une liste car on ne peut pas savoir si le geste vise à faire défiler la liste ou à sélectionner un raccourci gestuel du menu (contrairement aux PCs où le clic gauche ou droit de la souris permet de choisir un des deux modes).

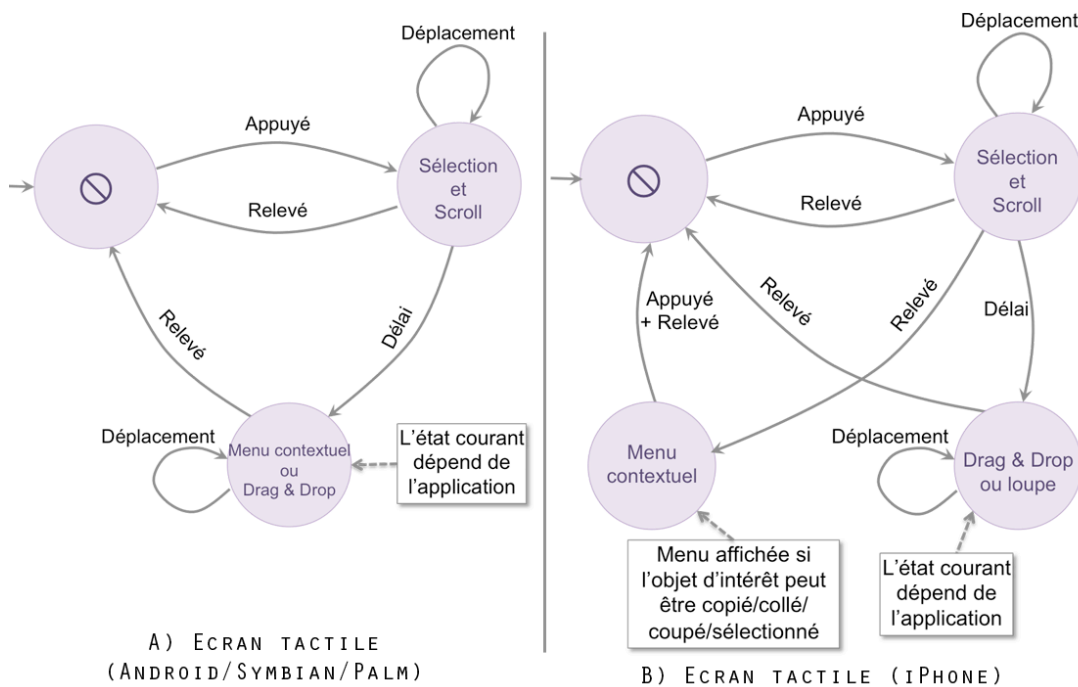


Figure 7. A) Modèle simplifié d'interaction sur l'écran tactile d'un dispositif Android ; B) Modèle simplifié d'interaction sur l'écran tactile d'un iPhone.

**iPhone.** Le modèle d'interaction sur l'écran tactile d'un iPhone (FIG. 7.B) est légèrement différent des précédents modèles. Les concepteurs ont décomposé l'état permettant d'initier un Drag & Drop ou l'ouverture d'un menu contextuel en deux nouveaux états. Tandis que le Drag & Drop est déclenché par un appui long, le menu contextuel s'ouvre au relâchement du doigt (ou après un "double clic"). Cette solution a des inconvénients : d'une part les états interactionnels varient selon les applications car l'appui long active un Drag & Drop sur l'écran d'accueil mais ouvre une loupe qui aide à placer précisément le curseur (barre verticale) dans un mot dans d'autres applications. D'autre part, le menu contextuel ne contient que certaines commandes spécifiques (copier/coller/couper/sélectionner) et ne s'ouvre que si les commandes sont applicables à l'objet sur lequel débute l'action.

De plus le modèle d'interaction de l'iPhone, comme celui des dispositifs Android, Symbian ou Palm, présente des problèmes de compatibilité lorsque l'utilisateur déplace son doigt sur l'écran : il n'est par exemple pas possible de savoir si l'utilisateur veut faire défiler un composant graphique ou déclencher un Drag & Drop (c'est pourquoi un appui long est

nécessaire). Pour résoudre ce problème, certaines applications attribuent des fonctionnalités différentes suivant la direction du geste. Par exemple, dans l'application Mail, un geste vertical fait défiler la liste des messages alors qu'un geste horizontal permet de supprimer un message.

Ces exemples présentent les problèmes que l'on rencontre sur différents types de dispositif, et montrent d'une part qu'il manque des moyens pour spécifier les états interactionnels dans lequel l'utilisateur peut être, et d'autre part qu'il est difficile de trouver des compromis optimaux, tant pour le choix des spécificateurs que dans celui des états auxquels ils mènent.

## 2.2 Prises en compte des capacités humaines

Les éléments à prendre en compte par rapport aux capacités humaines sont au nombre de trois. Nous détaillons tout d'abord les contraintes en sortie qui concernent la faible bande passante des modalités auditive et haptique. Nous présentons ensuite les contraintes en entrée dues aux mouvements limités des doigts et de la main.

### *Faible bande passante des modalités auditive et haptique*

Selon Ware (WARE 96), 70% de l'information perçue par l'homme provient du canal visuel. Les attributs de la vision (contraste, couleur, forme, luminance, distante, etc.) en font un canal de perception très riche. De plus, avec la vision, toutes les dimensions sont perçues quasi *simultanément*, alors qu'avec les modalités auditive ou haptique, la perception est beaucoup plus *séquentielle*. De ce fait la reconnaissance d'objets est plus rapide et nécessite moins d'efforts cognitifs (de mémorisation ou de reconstitution de contenu) que les autres modalités.

**Modalité auditive.** La perception du son est caractérisée par plusieurs attributs : l'intensité, la tonalité, le timbre, la direction et le rythme. La modalité auditive possède plusieurs avantages comme le fait de pouvoir entendre les sons quel que soit l'endroit d'émission (qui peut aussi être une contrainte comme nous le verrons après), contrairement à la modalité visuelle qui nécessite une attention de l'utilisateur à un endroit précis. Le son peut être ainsi utilisé pour décharger la modalité visuelle (éviter de surcharger l'écran). Ceci peut être particulièrement utile en condition de mobilité ou l'attention de l'utilisateur est souvent partagée entre plusieurs tâches et n'est donc pas dirigée en permanence vers l'écran. Par exemple l'utilisateur peut entendre des alertes sonores (sonneries) quelle que soit sa position par rapport au dispositif. EarPod (ZHAO 07) (FIG 8) utilise cette caractéristique pour permettre de sélectionner de la musique sur un lecteur MP3 sans avoir à regarder l'écran (Eyes-free selection). Lorsque l'utilisateur déplace son doigt sur les items d'un menu circulaire, il entend le nom de l'item sur lequel il se trouve. Dans ce système, comme dans (HIIPAKKA 03), le retour auditif est spatialisé, c'est à dire que la sélection d'un item situé à droite du menu entraîne un retour sonore plus fort dans l'écouteur droit.



Figure 8. EarPod aide l'utilisateur à se déplacer dans une bibliothèque musicale grâce à des retours auditifs spatialisés.

Malgré ces avantages, la modalité auditive a une basse résolution comparée à la modalité visuelle (BREWSTER JUNE 02). La localisation spatiale du son est limitée : des différences d'un degré d'angle peuvent être détectées lorsque le son arrive face à l'utilisateur, mais cette précision tombe à 30 degrés d'angle lorsque le son provient des cotés (BREWSTER JUNE 02). En comparaison, des différences d'angle de 2 secondes sont détectées dans le champ visuel central. Le changement d'un attribut du son peut aussi affecter les autres, par exemple changer la fréquence d'une note peut la faire percevoir plus forte. De surcroît, le son est séquentiel : les informations sont énoncées les unes après les autres. L'utilisateur doit donc rejouer le son ou faire un effort de mémorisation si le nombre d'informations énoncées est important.

**Modalité haptique.** La plupart des dispositifs haptiques sont conçus pour la main car les doigts possèdent de nombreux récepteurs sensitifs. Il existe deux sortes de retour haptique (CASIEZ 04) : le retour tactile (perception de surface, de texture, température, détection d'arêtes) et le retour de force (perception de contact, dureté, poids, forces d'inertie) plus rarement utilisé, car les technologies utilisées sont relativement imposantes en taille et coûteuses en énergie, et sont de ce fait difficilement intégrables dans des dispositifs destinés à être portés. Le retour tactile peut être une alternative à la modalité auditive dans des environnements bruyants où il est difficile, voire impossible, de percevoir un son (notamment avec le vibreur du téléphone, qui est aussi utilisé pour éviter de déranger l'entourage). Il peut aussi être utilisé comme substitution à la modalité visuelle, lorsque l'utilisateur ne peut pas regarder l'écran. C'est le cas dans Tactimod (COUTURE 06), où l'utilisateur se dirige grâce à des vibreurs placés sur les cotés du dispositif pour savoir la direction à prendre lors d'un guidage GPS. Enfin, le retour tactile peut aussi servir pour augmenter l'interactivité : dans des jeux (LINJAMA 04) ou dans (LUK 06) où un stimulateur tactile donne une impulsion à l'utilisateur pour l'aider à trouver des informations sur une page Web (liens ou titres par exemple).

Cependant la résolution du sens haptique est basse comparée aux autres modalités. Reconnaître l'information est difficile, même pour des personnes entraînées (personnes à vision réduite) (FRITZ 99). La combinaison de plusieurs dispositifs haptiques peut augmenter cette résolution, comme dans (WILLIAMSON 07) où l'utilisateur peut ressentir des balles rebondissant sur les parois du mobile grâce à plusieurs vibreurs placés à divers endroits sur le

dispositif. Toutefois, comme la modalité auditive, la modalité haptique est exploitée de manière séquentielle (HATWELL 00). En effet, pour appréhender les objets dans leur intégralité, il doit y avoir un mouvement de la part de l'utilisateur. Ceci nécessite un effort de reconstruction mentale des données perçues pour comprendre l'information. Notamment dans (ZIAT 06) (FIG 9.A) des expériences ont été menées pour étudier la perception haptique d'objets (lettres) grâce à des cellules Braille placées sous les doigts. Les auteurs ont alors observé que les utilisateurs effectuaient des mouvements de doigts afin de mieux deviner les formes.

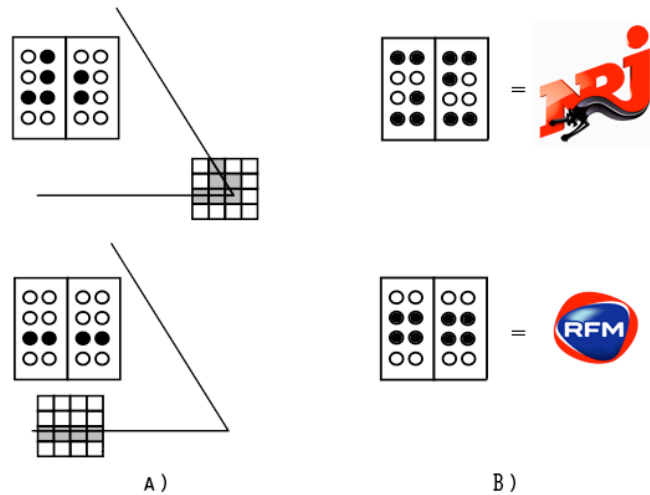


Figure 9. Exemple de retour tactile A) séquentiel : reconnaissance d'un triangle ; B) non séquentiel : reconnaissance d'un motif associé à une station de radio.

Dans certaines conditions, la perception peut être relativement rapide. Par exemple, TactiPod (TAHIR 09) est un dispositif permettant de naviguer dans une liste de radios (et autres données multimédia). Lorsque l'utilisateur atteint une radio favorite, une forme (différente pour chaque favori) est envoyée sur une cellule Braille (FIG 9.B). Avec un peu d'entraînement, l'utilisateur peut reconnaître rapidement la forme et savoir sur quelle radio il se trouve. Cependant, en prenant une cellule Braille à huit picots (d'environ la largeur d'un doigt), le nombre de formes possibles est limité, d'autant que des motifs trop similaires risquent d'être confondus par l'utilisateur (PIETRZAK 06).

### *Amplitude réduite des gestes des doigts*

Les éléments à prendre en compte par rapport aux capacités humaines concernent aussi les modalités en entrée. Notamment la morphologie de la main contraint les mouvements de flexion/extension du pouce (SAVESCU 05) (FIG. 10.A). Comme l'ont montré Karlson et al. (KARLSON 08), il en résulte que certaines zones de l'écran (FIG 10.B) sont plus faciles d'accès que d'autres, ce qui peut être particulièrement gênant pour les personnes ayant une main de petite taille (les enfants notamment). De plus, l'articulation à la base du pouce (articulation carpo-métacarpienne) empêche l'utilisateur d'effectuer aisément des gestes diagonaux dans l'axe du pouce (KARLSON 05) (les gestes Nord-Ouest vers Sud-Est sont plus difficiles pour les

droitiers que les gestes Nord-Est vers Sud-Ouest et inversement pour les gauchers). Enfin, certains écrans sont encadrés d'une bordure qui peut rendre difficile l'accès aux bords, surtout si cette bordure est épaisse car le pouce, ayant une surface bombée, butera dessus contrairement à un stylet qui pourra facilement atteindre ces zones.

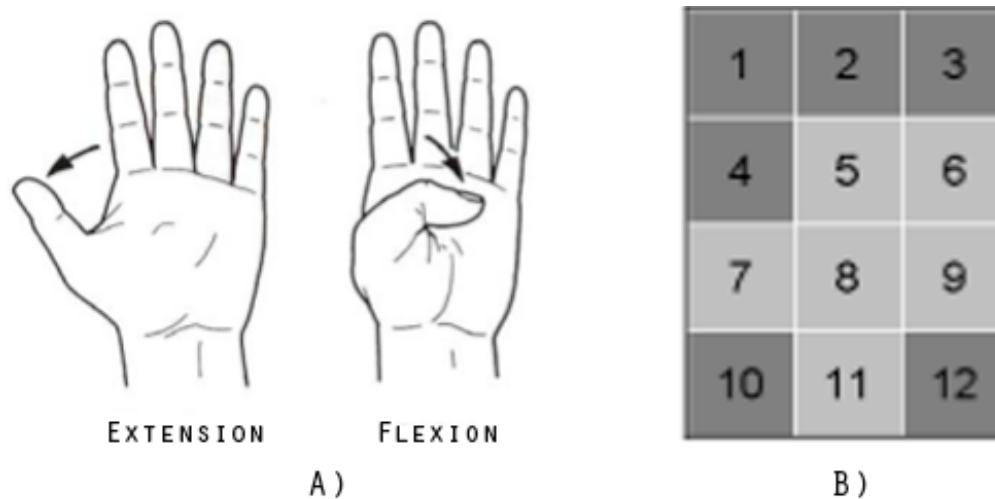


Figure 10. A) Limites de mouvement de flexion/extension du pouce ; B) Impacts sur l'accès à certaines zones de l'écran (les zones sombres sont plus difficiles d'accès).

D'autres contraintes proviennent de la prise en main du dispositif. Tout d'abord, lorsque le dispositif est utilisé à une seule main, le pouce évolue sur la face avant du dispositif tandis que les autres doigts sont sur la face arrière ou sur les bords. Les interactions multi-doigts, comme le "Pinch & Expand" qui consiste à rapprocher ou à écarter deux doigts pour zoomer, sont ainsi impossibles ou périlleuses (FIG 11). Par ailleurs, l'accès aux boutons physiques sur les cotés du dispositif peut nécessiter de changer la prise en main du dispositif comme illustré sur la FIGURE 12. En pratique, les doigts opposés au pouce se situent le plus souvent à l'arrière du dispositif et ne sont pas exploités. Plusieurs études récentes proposent notamment d'ajouter des capteurs supplémentaires à l'arrière du dispositif ("back of device interaction") : comme un joystick dans (CHAU 06) ou une surface tactile dans (BAUDISCH 09).



Figure 11. Les interactions multi-doigts sont difficiles à une seule main.





Figure 12. Changement de prise en main pour atteindre les boutons physiques du dispositif sur les cotés.

### *Amplitude réduite des gestes de la main*

Les mouvements des doigts sont limités par la prise en main du dispositif, mais c'est également le cas pour les mouvements de la main pouvant être exploités par des techniques d'interaction à base de capteurs de mouvements. Une étude (RAHMAN 09) montre que les mouvements de flexion/extension de la main peuvent s'étendre sur une plage de  $105^\circ$  (FIG. 13.A), ceux de pronation/supination peuvent aller jusqu'à  $125^\circ$  (FIG. 13.B) et les mouvements de déviation ulnaire et radiale du poignet se limitent à  $45^\circ$  d'amplitude (FIG. 13.C).

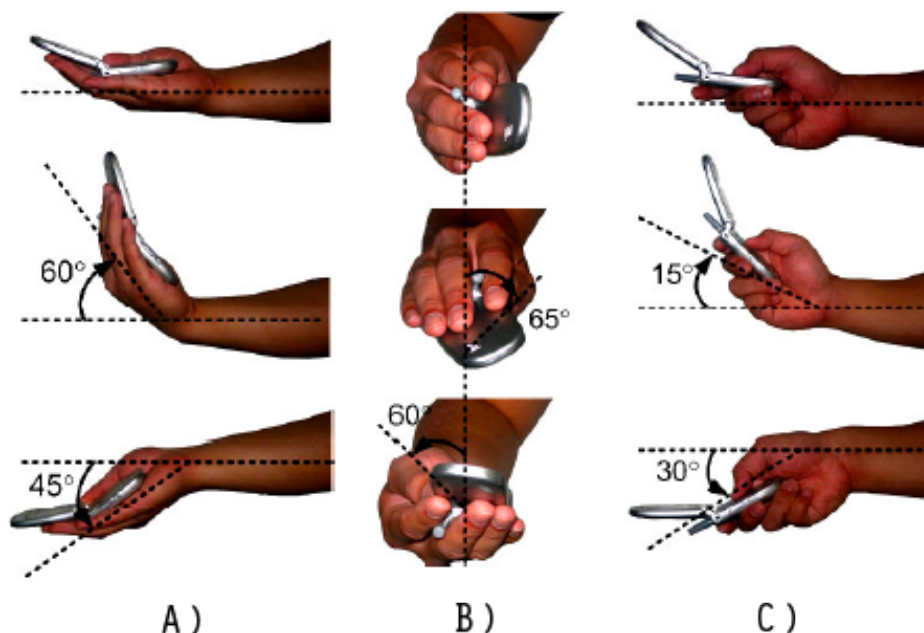


Figure 13. A) Les mouvements de flexion/extension de la main s'étendent sur une plage de  $105^\circ$  : B) ceux de pronation/supination jusqu'à  $125^\circ$  et C) ceux de déviation ulnaire et radiale du poignet jusqu'à  $45^\circ$ .

### 2.3 Contraintes dues à l'environnement

Les contraintes dues à l'environnement sont au nombre de trois. Nous présentons les contraintes causées par le milieu sonore, qui peut gêner l'utilisation des modalités auditives et vocales, puis nous détaillons le cas des gestes larges en trois dimensions qui peuvent être difficiles à exécuter selon l'environnement social de l'utilisateur.

#### *Modalité auditive dépendante de l'environnement sonore*

La modalité auditive est contrainte par l'environnement sonore, qui peut changer rapidement en situation de mobilité. Par exemple l'utilisateur peut être dans une gare bruyante et, de ce fait, ne pas entendre les informations. Quelques instants plus tard il doit mettre son mobile en mode silencieux pour ne pas déranger les autres usagers (ou pour préserver sa confidentialité) car il se trouve dans son train.

#### *Modalité vocale dépendante de l'environnement sonore*

Les entrées vocales sont une alternative aux limitations interactionnelles des mobiles, mais elles sont relativement peu utilisées. Les techniques de reconnaissance actuelles présentent en effet encore certaines limites (ZAYKOVSKIY 06), en particulier la perception du son est relativement longue et difficile en environnement bruyant. De plus, les utilisateurs peuvent être réticents à parler à "une machine" dans un lieu public ou pour préserver leur confidentialité.

#### *Gestes dans l'espace et environnement social*

L'utilisation des gestes détectés par des capteurs de mouvements devient de plus en plus populaire (par exemple avec la console Nintendo Wii). Cependant, l'utilisateur peut être gêné d'avoir à effectuer certains gestes de large amplitude en public, hors d'un contexte familial (RICO 09). Certains gestes peuvent de plus être dangereux pour autrui. A titre d'exemple, Nintendo mentionne à ses utilisateurs de bien accrocher leur dragonne pour éviter des accidents comme les télévisions cassées par une Wiimote non attachée au poignet.

### 2.4 Synthèse

Cet espace problème nous a permis de décrire les contraintes des dispositifs mobiles. En entrée, nous avons vu que la modalité vocale n'était pas toujours utilisable en condition de mobilité, en particulier parce qu'elle dépend de l'environnement sonore de l'utilisateur. Les modalités par mouvement en deux dimensions (exploitant l'écran tactile ou le clavier physique) ou en trois dimensions (utilisant les capteurs de mouvements) sont moins sujettes à ces problèmes mais souffrent d'autres contraintes. En effet, les manques matériels (clavier



physique réduit ou absent, interaction aux doigts et lacunes des écrans tactiles) réduisent sensiblement les possibilités d'interaction avec un dispositif mobile, mais ces limitations sont renforcées dans le cas de l'interaction à une main qui amenuise encore les capacités interactionnelles (les doigts et la main ayant des mouvements d'amplitude limitée).

La modalité auditive souffre d'une faible bande passante informationnelle (comparativement à la modalité visuelle), d'une perception séquentielle et est dépendante de l'environnement. La modalité haptique a une bande passante informationnelle encore plus réduite et peut parfois nécessiter des équipements encombrants et coûteux en énergie, difficiles à déployer sur mobiles. Enfin le manque de place caractérise la modalité visuelle, mais celle-ci reste tout de même prédominante, les autres modalités étant utilisées plutôt comme un supplément de la modalité visuelle (hormis le cas particulier des utilisateurs non voyants). C'est pourquoi les techniques de représentation de l'information mises en œuvre sur dispositifs mobiles sont majoritairement basées sur la modalité visuelle.

Face aux contraintes identifiées dans cette section, nous nous sommes ensuite intéressé aux techniques basées sur les modalités prépondérantes (la modalité par mouvement pour les entrées et la modalité visuelle pour les sorties) bien que les autres modalités ne soient pas à ignorer, en particulier pour concevoir des techniques destinées aux personnes à vision réduite ou souffrant d'un déficit moteur. Des informations complémentaires sur les techniques utilisant d'autres modalités peuvent être trouvées dans (HUERTA 00) pour les entrées vocales, (ZHAO 09) pour le retour audio et (PIETRZAK 08) pour le retour haptique.

### 3. Taxonomie des techniques pour dispositifs mobiles

Dans la section précédente, nous avons utilisé la définition de la modalité utilisée par les psychologues et les ergonomes. Avec une vision plus logicielle, Nigay (NIGAY 96) définit la modalité par un couple  $\langle D, L \rangle$ , où D est un dispositif physique et L un langage d'interaction. 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. Nous utilisons cette définition de la modalité pour construire notre état de l'art des techniques pour dispositifs mobiles. Nous nous intéressons en particulier à la modalité par mouvement (techniques d'interaction) et à la modalité visuelle (techniques de visualisation). Par exemple, lorsque l'utilisateur pointe une cible sur un écran tactile, il utilise la modalité  $\langle \text{écran tactile, geste déictique} \rangle$ . Nous allons voir que pour chaque type de techniques (d'interaction ou de visualisation), il existe différents langages d'interaction (comme les gestes déictiques) pouvant être couplés à divers dispositifs (comme l'écran tactile). La FIGURE 14 présente les axes de notre taxonomie qui sont détaillés dans cette section.

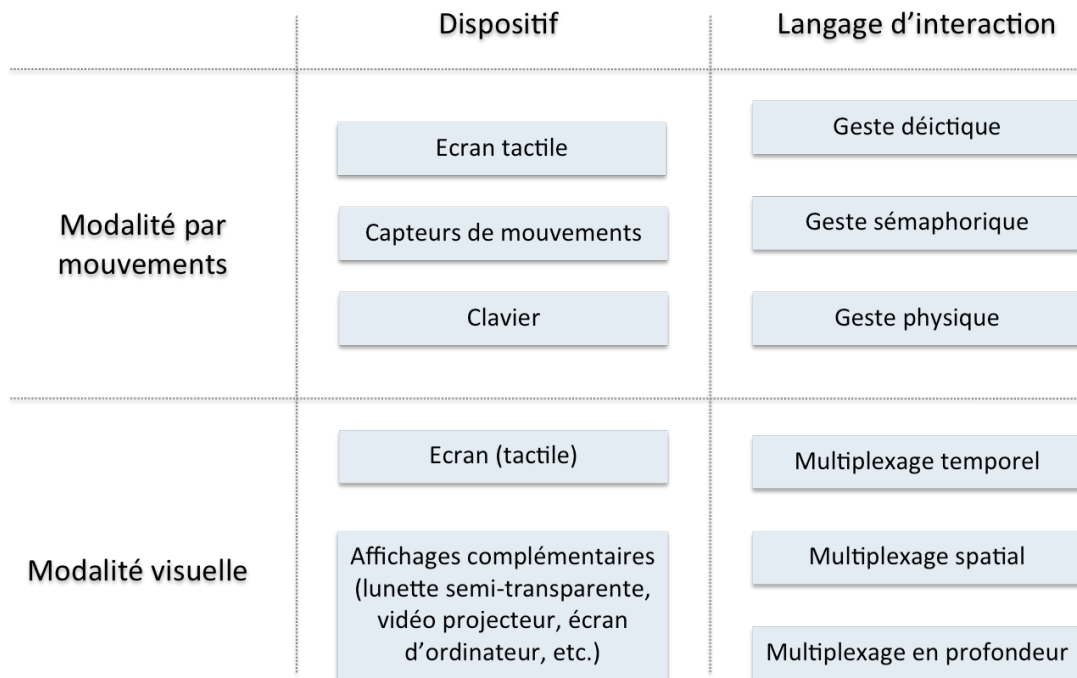


Figure 14. Taxonomie des techniques d'interaction et de visualisation.

### 3.1 Techniques d'interaction

Plusieurs taxonomies ont été proposées pour classifier les différents langages gestuels utilisés dans les techniques d'interaction. Cadoz (CADOZ 94) considère trois types de gestes qui dépendent de leur fonction : pour explorer le monde physique (fonction épistémique), pour agir matériellement sur ce monde (fonction ergotique) et pour communiquer avec les autres (fonction sémiotique). Dans sa taxonomie, Quek (QUEK 02) distingue les gestes manipulatifs, qui correspondent aux fonctions ergotique et épistémique de Cadoz et les gestes communicatifs, équivalant à la fonction sémiotique. Ces derniers sont ensuite composés des gestes symboliques et actifs tels que les gestes de pointage appelés gestes déictiques.

Comme soulevé dans (CHEVALDONNÉ 06), ces taxonomies sont simples mais présentent toutefois des ambiguïtés. Par exemple, lorsque l'utilisateur déplace une souris sur un écran d'ordinateur grâce à son mobile qu'il oriente dans l'espace, il effectue un geste qui peut être à la fois ergotique, épistémique et sémiotique : la manipulation du dispositif correspond à un geste ergotique ; la sensation du dispositif dans la main et la perception de l'amplitude des mouvements correspond à un geste épistémique ; et les informations que l'utilisateur communique en pointant un objet correspond à un geste sémiotique. Pour éviter les confusions, nous utilisons donc un autre point de vue dans notre classification en nous basant sur les travaux de WexelBlat (WEXELBLAT 98) et Karam (KARAM 05). Nous proposons trois

catégories de langages gestuels mis en œuvre dans les techniques d'interaction sur mobiles. Nous illustrons ces catégories et les situons par rapport aux classifications existantes avant de dresser l'état de l'art.

- Gestes **déictiques**. Ils s'utilisent pour désigner un endroit de l'espace comme lorsque l'on pointe une cible sur un écran tactile. Il peut s'agir de pointage ou de déplacement d'objet (lors du déplacement d'un curseur, lors d'un Drag & Drop ou lors du déplacement d'une barre de défilement), le déplacement pouvant être vu comme un double pointage. Nous retrouvons les gestes déictiques pour le pointage simple dans les travaux de Karam, McNeill (MCNEILL 92), Eisenstein (EISENSTEIN 04), Cadoz (CADOZ 94) et Quek (QUEK 02). Le déplacement d'objets (ou double pointage) se retrouve dans les catégories «gestes ergotiques» ou «gestes manipulatifs» de Cadoz.
- Gestes **sémaphoriques**. Ils se basent sur l'utilisation de symboles : l'utilisateur dessine un symbole pour communiquer avec la machine. Il peut s'agir de symboles culturels comme des lettres ou des chiffres ou de symboles plus abstraits comme ceux utilisés dans les Marking Menus (KURTENBACH 91). Cette catégorie apparait dans les travaux de Baglioni (BAGLIONI 09), Karam et Quek. Ceux de Eisenstein, McNeill et Wobbrock (WOBBROCK 09) n'incluent que les gestes basés sur des symboles culturels.
- Gestes **physiques**. Pour expliquer cette catégorie, nous partons de la définition des gestes incarnés ("embodied") définis par (FISHKIN 98) et (DOURISCH 01). Ce sont des gestes qui tirent parti des aptitudes physiques que nous avons acquises dans le monde réel et qui les utilisent pour créer des interactions plus naturelles pour l'utilisateur. Ces gestes sont généralement associés aux gestes tridimensionnels tels que secouer le dispositif (ce geste étant alors détecté par des capteurs de mouvements) et nous avons souhaité étendre cette définition aux gestes en deux dimensions, et notamment ceux effectués sur un écran tactile comme Rubbing (OLWAL 08) basé sur la métaphore de la gomme. Puisque ces gestes tirent également parti des connaissances que nous avons dans le monde réel, nous nommons cette catégorie "physique". On trouve des gestes métaphoriques (le geste effectué dans le monde physique et numérique aboutit à un résultat similaire dans les deux mondes) dans les travaux de Baglioni, McNeil, Eisenstein et Wobbrock. Notons que les gestes sémaphoriques peuvent être considérés comme un sous-ensemble des gestes physiques exploitant la métaphore du dessin.

### **Gestes déictiques**

Les gestes déictiques s'illustrent dans de nombreuses techniques sur écrans tactiles. Nous les présentons en fonction du type de tâche : la sélection de cibles, la sélection d'items dans un menu et le défilement ou déplacement d'objets. Nous parlons ensuite des techniques moins répandues qui combinent les gestes déictiques à un clavier ou à des capteurs de mouvements.

**Sélection de cibles.** La technique la plus courante pour pointer une cible, souvent appelée Direct Touch, consiste à la pointer directement sur l'écran tactile. Elle est cependant imprécise pour la sélection de petites cibles avec le doigt (voir la section "interaction aux doigts"). C'est la raison pour laquelle beaucoup de systèmes actuels intègrent des cibles de taille assez large. Par exemple, les dispositifs Windows Mobile et BlackBerry (FIG. 15) offraient initialement des petites cibles pour l'interaction au stylet ou au joystick, mais leurs concepteurs se dirigent de plus en plus vers une stratégie de larges cibles comme sur l'iPhone ou sur les mobiles Android et Symbian (ces derniers représentés majoritairement par la marque Nokia). Ce choix n'est pas sans contraintes car les cibles prennent alors plus de place à l'écran. L'utilisateur doit donc souvent naviguer dans des listes ou des pages d'items pour trouver la cible désirée. De plus, certaines cibles ne peuvent pas facilement être agrandies. C'est le cas des liens de pages Web qui sont de ce fait difficilement sélectionnables avec le pouce.



Figure 15. Evolution de la taille des cibles entre 2007 et 2009 sur les systèmes d'exploitation Windows Mobile et Rim (BlackBerry).

D'autres stratégies sont envisageables pour faciliter la sélection de cibles. La première a été introduite par le BlackBerry Storm et consiste à intégrer un écran "cliquable" qui permet deux modes de fonctionnement : lorsque l'utilisateur appuie légèrement sur l'écran il met une cible en surbrillance et il la sélectionne en appuyant plus fort sur l'écran. Les sélections accidentelles sont alors évitées car l'utilisateur peut s'assurer que la cible en surbrillance est effectivement celle qu'il souhaite sélectionner avant de cliquer sur l'écran. Cet apport technologique permet d'introduire un nouvel état interactionnel au modèle d'interaction comme le montre la FIGURE 16.A. Certains dispositifs détectant la pression ou la pseudo-pression (par une estimation de la surface du doigt en contact avec l'écran) permettent aussi ce double fonctionnement. Notamment, GraspZoom (MIYAKI 09) est une technique qui permet de faire défiler une image lorsque l'utilisateur déplace son pouce sur l'écran. Si l'utilisateur appuie plus fort sur l'écran pendant le déplacement, il passe alors en mode zoom (FIG. 16.B). Cependant, contrairement au BlackBerry Storm, il n'y a pas de retour passif (sensation de "clic") permettant de distinguer les différents états interactifs.

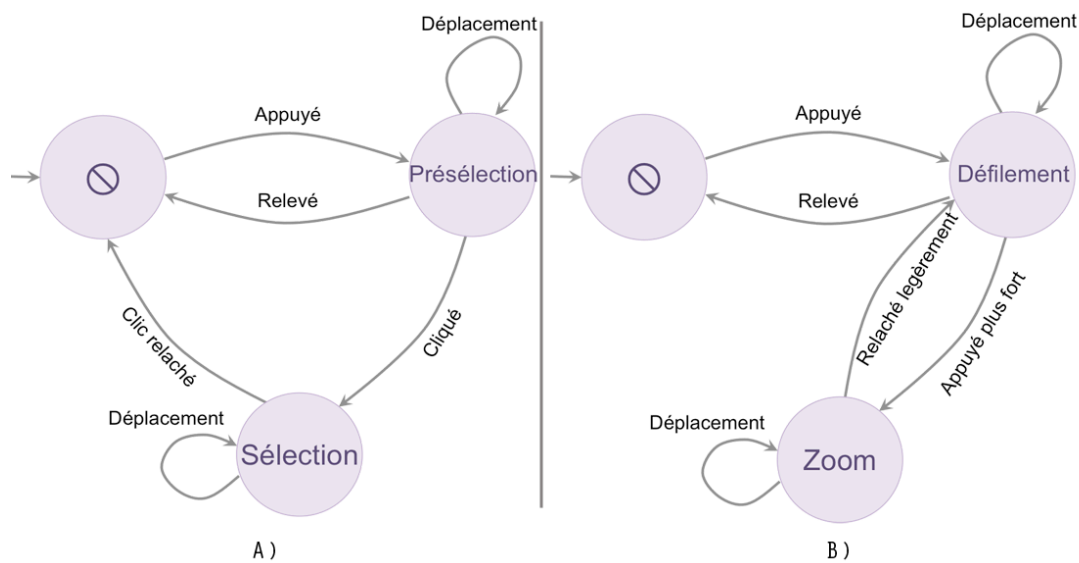


Figure 16. Modèle d'interaction A) de l'écran cliquable du BlackBerry Storm ; B) de GraspZoom, une technique détectant la pression du doigt sur l'écran.

La seconde stratégie est logicielle et consiste à introduire une technique de pointage qui améliore la précision de la sélection sans augmenter la taille des cibles. Il s'agit en particulier des évolutions de Direct Touch comme Starbust (BAUDISCH 08), ou des techniques qui demandent à l'utilisateur de déplacer son pouce sur l'écran comme Offset Cursor (POTTER 98), Adaptive Offset Cursor (HUOT 06), Shift (VOGEL 07), Thumbspace (KARLSON 08) ou Burst (GUNN 09). Néanmoins, ces techniques ne répondent pas à toutes les contraintes des dispositifs mobiles. Nous expliquons pourquoi et détaillons leurs principes dans le CHAPITRE 3 qui est consacré au pointage de petites cibles sur écran tactile. Nous y parlerons aussi des méthodes de pointage pour l'entrée de texte.

**Sélection d'items dans un menu.** Le pointage dans un menu est un autre type de geste déictique. Les barres de menus existent dans tous les systèmes existants, excepté l'iPhone. Lorsque le dispositif le permet, ces menus sont activables via un bouton physique. Dans la majorité des cas ils s'ouvrent à partir de la barre de menu affichée en bas ou en haut de l'écran (FIG. 17.A.B), ce qui peut les rendre difficiles d'accès avec le pouce (voir la section "amplitude réduite des gestes des doigts"). Certains systèmes proposent également des menus contextuels qui sont activés de différentes manières. Par exemple sur les dispositifs Windows Mobile, ceux-ci s'affichent après un appui long sur l'écran, ce qui peut être coûteux en temps (voir la section "lacunes des écrans tactiles"). Comme présenté dans l'espace problème, le menu contextuel de l'iPhone s'affiche après un double tap ou au relâchement du doigt, mais il ne contient que certaines commandes spécifiques (copier, coller, couper et sélectionner).

Les menus sont généralement linéaires, comme ceux des ordinateurs conventionnels. La taille des items est cependant plus large pour faciliter la sélection au pouce, ce qui augmente la longueur du menu. C'est pourquoi on trouve parfois des barres de défilement pour accéder

aux items comme dans les listes (par exemple sur le Nokia n97 illustré sur la FIGURE 17.A). A la différence des menus habituels, l'ArchMenu (HUOT 07) et le ThumbMenu (HUOT 07) sont de forme circulaire. Nous verrons dans le CHAPITRE 4 que ces menus proposent des solutions au problème d'occultation par le pouce mais qu'ils comportent certains inconvénients tel qu'une occupation spatiale supérieure à celle des menus linéaires, une propriété à ne pas négliger étant donné la petite taille des écrans. Un autre problème des menus sur dispositifs mobiles est la difficulté d'utiliser le mode expert, même en présence d'un clavier physique. Les raccourcis claviers sont en effet basés sur l'utilisation de modificateurs (FIG. 17.B) et leur utilisation est difficile lors de l'interaction à une seule main (voir la section "clavier physique réduit ou absent").



Figure 17. A) Le menu du Nokia n97 est placé dans une barre en bas de l'écran et il contient parfois une barre de défilement ; B) Celui du Palm Pre est placé dans une barre en haut de l'écran. Les items peuvent être associés à des raccourcis claviers. C) applications iPhone : une barre de navigation contenant les contrôles principaux en haut de l'écran ; une barre d'outils ou d'onglets en bas de l'écran.

Enfin, l'iPhone est un cas particulier dans la mesure où il ne possède pas de barres de menus. Les concepteurs d'application pour iPhone doivent en effet suivre des recommandations précises pour assurer une cohérence entre les applications vendues sur l'AppStore : les commandes principales doivent être placées dans une barre de navigation en haut de l'écran (FIG. 17.C), les autres commandes étant disposées dans des barres d'outils ou des onglets en bas de l'écran. Les boutons doivent de surcroît être suffisamment larges pour la sélection avec le pouce, ce qui limite le nombre de commandes que l'on peut afficher.

**Défilement ou déplacement d'objets.** Nous considérons le défilement et le déplacement d'objets comme un double pointage puisque l'utilisateur spécifie au système un point de départ et un point d'arrivée. Le défilement peut servir à naviguer dans des contenus en deux dimensions, par exemple des cartes ou des images, ou dans une seule dimension comme avec les listes. Par exemple le défilement est utilisé pour basculer entre les pages ouvertes du navigateur Safari sur iPhone ou pour faciliter l'accès aux fenêtres ouvertes sur le Palm Pre : des

aperçus de fenêtres sont affichées et l'utilisateur déplace son doigt latéralement pour faire défiler la liste des applications (ou pages Web) ouvertes (FIG. 18.A). Il est ainsi possible de naviguer entre les applications et de copier/coller des objets d'une fenêtre à une autre même si celles-ci sont superposées les unes sur les autres à cause du manque de place écran.

Avec la majorité des systèmes modernes, le défilement de contenus s'effectue sans barres de défilement, qui consomment trop d'espace écran et sont imprécises du fait de leur petite taille. L'utilisateur fait défiler un contenu en déplaçant son doigt n'importe où sur le composant graphique qui l'affiche, une fine barre apparaissant parfois près du bord à titre indicatif. Cette méthode de défilement a cependant l'inconvénient d'empêcher les opérations comme le Drag & Drop (déplacement d'objets) comme nous l'avons vu dans la section "lacunes des écrans tactiles". Le Drag & Drop se fait alors généralement via un menu.

Lorsqu'il n'existe pas de menu, comme sur l'iPhone, plusieurs solutions ont été proposées pour le déplacement d'objets : la première consiste à effectuer un appui long sur un item pour passer dans un mode de déplacement. Cependant cette opération ne déclenche pas toujours le Drag & Drop (voir la section "lacunes des écrans tactiles"). La seconde solution est de placer une zone spécifique sur les cibles (FIG. 18.B) : lorsque l'utilisateur déplace son doigt en commençant sur cette marque, il déplace l'objet, sinon il déplace la liste entière.



Figure 18. A) Navigation entre fenêtres ouvertes sur le Palm Pre ; B) Drag & Drop sur iPhone : une marque à droite sur la cible permet de la déplacer.

**Gestes déictiques sur un clavier.** Le clavier peut aussi servir de dispositif de pointage. Lorsqu'elles existent, les touches directionnelles permettent de naviguer dans des tableaux ou des listes. Dans ZoneZoom (ROBBINS 04) (FIG 25.A) et Fathumb (KARLSON 06) (FIG. 26.C), l'utilisateur doit appuyer sur une touche spécifique du clavier physique correspondant à une zone de l'écran. Ceci permet d'effectuer un zoom sur la zone correspondante à l'écran (ZoneZoom) ou de sélectionner une catégorie lors d'une recherche dans une base de donnée (Fathumb).



**Gestes déictiques en trois dimensions.** Le langage déictique peut être combiné à diverses technologies de capteurs. Par exemple dans l'application Google Sky, l'utilisateur pointe son dispositif vers le ciel pour observer les constellations relatives à sa position sur le globe terrestre. Cette opération est possible grâce à une boussole mais d'autres technologies, comme des accéléromètres, peuvent être utilisées, notamment dans (BALLAGAS 05) (FIG. 19.A) où l'utilisateur se sert de son dispositif mobile pour contrôler un curseur se déplaçant sur l'écran d'un ordinateur situé à proximité. Il peut aussi déplacer des objets présents sur l'ordinateur en appuyant sur une touche. Dans Magic Lens (ROHS 08) (FIG. 19.B), un pointeur sur l'écran du dispositif mobile permet de sélectionner des objets sur une carte physique. Cette carte est filmée par la caméra du mobile et affichée sur l'écran pour donner une impression de transparence du dispositif et créer une réalité augmentée. Ce procédé est aussi utilisé dans (GALLO 08), mais c'est la main qui sert à positionner le curseur (FIG. 19.C).

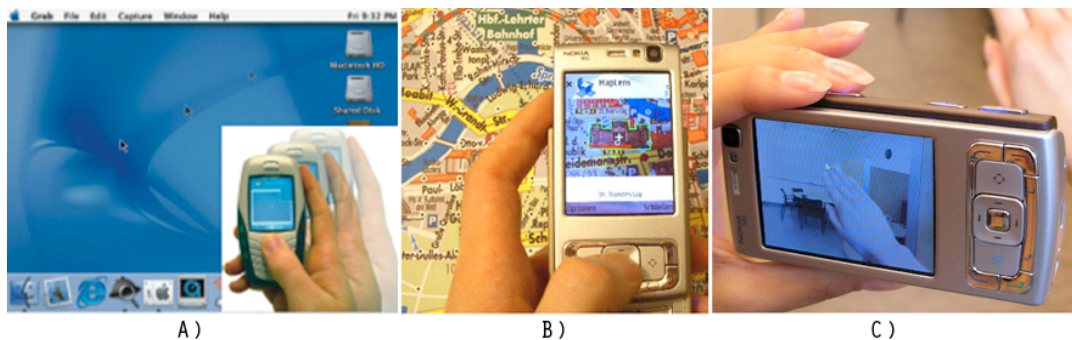


Figure 19. A) Contrôler le curseur d'un ordinateur à proximité ; B) sélectionner des objets sur une carte physique ; C) Reconnaissance de la main pour déterminer la position du curseur.

### *Gestes sémaphoriques*

Diverses techniques proposent des vocabulaires de gestes sémaphoriques à effectuer sur l'écran tactile. Nous les présentons ainsi que celles qui proposent des solutions pour faciliter l'apprentissage de ces gestes. Nous parlons ensuite des techniques qui combinent les gestes sémaphoriques à un clavier ou à des capteurs de mouvements.

**Vocabulaires gestuels.** Les exemples les plus illustratifs des gestes sémaphoriques sur les écrans tactiles sont les vocabulaires pour l'entrée de texte comme Unistroke (GOLDBERD 93) (FIG. 20.A) ou EdgeWrite (WOBBROCK 03) (FIG. 20.B). Ils reposent sur un ensemble de symboles à saisir avec le stylet. Par exemple, pour entrer la lettre A avec la méthode Unistroke, l'utilisateur dessine un trait vertical vers le haut. L'avantage de ces techniques est qu'elles n'associent qu'un seul trait à une lettre, ce qui permet une saisie relativement rapide. Un inconvénient est qu'elles demandent à l'utilisateur d'apprendre et de mémoriser l'alphabet de gestes. Ce problème survient également dans AppLens (KARLSON 05) qui permet à l'utilisateur de lancer des commandes grâce aux gestes illustrés sur la FIGURE 20.C.



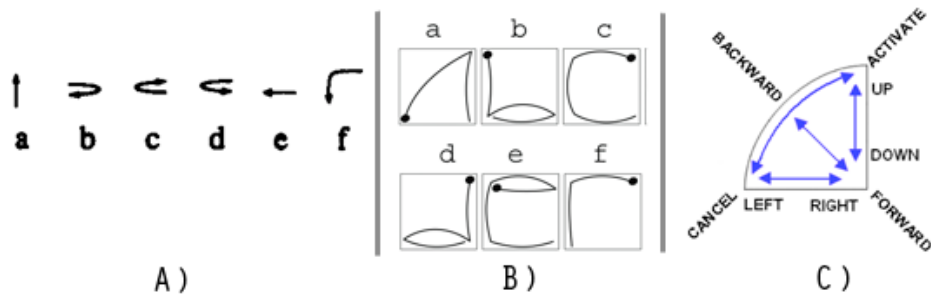


Figure 20. A) Alphabet Unistroke ; B) Alphabet EdgeWrite : C) Gestes utilisés dans LaunchTile pour activer des commandes.

On trouve également les gestes sémaphoriques dans produits commerciaux, notamment sur le HTC Cube et le Palm Pre qui disposent d'une zone tactile spécifique sous l'écran sur laquelle l'utilisateur peut effectuer des gestes. Par exemple sur le Palm Pre, un geste linéaire vers la gauche (FIG. 21.A) permet de lancer la fonctionnalité "précédente" utilisée pour la navigation dans une page Web. Ces gestes débutent toujours dans la zone de lancement, ce qui permet d'éviter les conflits avec les gestes effectués sur l'écran tactile. Notons que les confusions peuvent également être évitées en faisant appel aux gestes multi-touch lorsque cette possibilité est disponible et que l'utilisateur peut interagir avec ses deux mains (une tenant le dispositif).

Un autre type d'utilisation de ces gestes dans les dispositifs sur le marché est la personnalisation : l'utilisateur peut définir ses propres gestes et les associer à certaines fonctionnalités. Par exemple sur les dispositifs Android, l'utilisateur peut déverrouiller l'écran tactile en effectuant un geste de son choix (FIG. 21.B). De même, l'application Gesture Dial de l'iPhone permet à l'utilisateur de définir des raccourcis gestuels pour déclencher des appels vers ses contacts (FIG. 21.C).

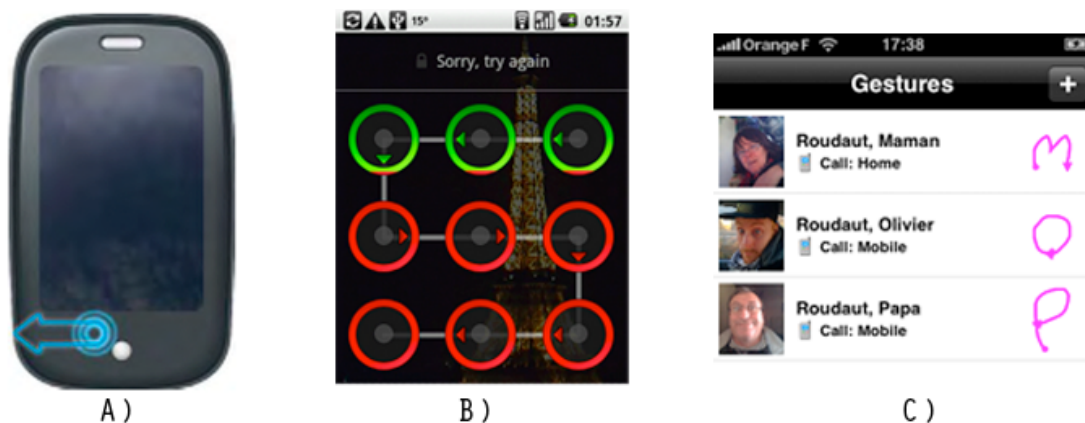


Figure 21. A) Un geste directionnel vers la gauche dans la zone de lancement du Palm Pre lance la commande "précédente" ; B) Sur Android, l'utilisateur définit un mot de passe gestuel pour déverrouiller l'écran tactile ; C) Gesture Dial (iPhone) permet d'appeler ses contacts en dessinant des raccourcis gestuels personnalisés.

**Apprentissage des gestes sémaphoriques.** Pour faciliter l'apprentissage des gestes, plusieurs techniques de menu proposent deux fonctionnements : un mode novice pour les utilisateurs qui découvrent le menu et un mode expert pour ceux qui connaissent les raccourcis gestuels.

Le geste est le même dans les deux modes, ce qui aide l'utilisateur novice à passer progressivement en utilisateur expert. Ce principe a été introduit par les Marking Menus (KURTENBACH 91) puis repris par diverses techniques dérivées comme le MultiStroke menu (ZHAO 04), le Wave menu (BAILLY 07) (et son équivalent sur dispositifs mobiles le Wavelet (FRANCONE 09)), le Flower menu (BAILLY 08), Le PushMenu (HUOT 08) ou encore QuikWriting (ISOKOSKI 04) pour l'entrée de texte sur mobiles. Les menus linéaires de (KURTENBACH 93) et (APPERT 09) sont augmentés d'une interaction gestuelle seulement en mode expert. Le détail de ces techniques sera donné au CHAPITRE 4 et nous verrons que ces techniques, majoritairement destinées aux ordinateurs classiques, ne sont pas toujours facilement transposables au cas des mobiles.

Enfin, les gestes sémaphoriques peuvent offrir une solution à l'imprécision du pointage des petites cibles. Dans Escape (YATANI 08) et les Slidings Widgets (MOSCOVITCH 08), la sélection d'une cible se fait grâce à un geste dans une direction qui est indiquée visuellement sur celle-ci. Ce principe est aussi utilisé sur l'interface de l'iPhone où les boutons sont remplacés par des interrupteurs. Il suffit alors de déplacer l'interrupteur vers la droite ou la gauche pour activer ou désactiver une fonctionnalité.

**Gestes sémaphoriques sur un clavier.** SmartPad (REKIMOTO 03) (FIG. 22.A) est une technique qui propose à l'utilisateur d'effectuer des gestes sur les touches du clavier physique afin lancer des actions sur un écran d'ordinateur se trouvant à proximité. Le clavier joue alors le rôle d'une surface tactile à faible résolution. De même, EdgeWrite (WOBBROCK 06), un vocabulaire gestuel présenté précédemment pour l'entrée de texte au stylet, a été revisité pour l'utilisation sur quatre touches physiques (FIG. 22.B).

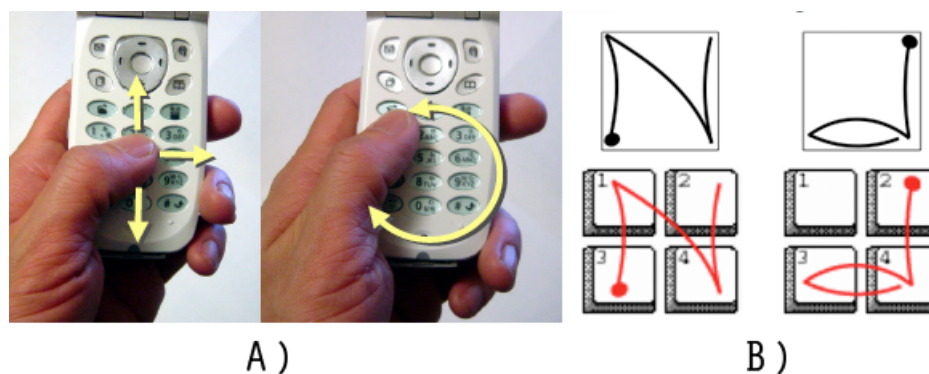


Figure 22. A) SmartPad, l'utilisateur effectue des gestes sur un clavier physique pour déclencher des fonctionnalités ; B) EdgeWrite revisité pour la saisie de texte sur quatre touches.

**Gestes sémaphoriques en trois dimensions.** Les gestes sémaphoriques peuvent être détectés via des capteurs de mouvements, comme dans (REHM 08) ou (KAUPPILA 08) où les utilisateurs doivent dessiner des formes dans l'espace (chiffres, lettres ou formes géométriques) pour lancer

des fonctionnalités. Toutefois ces gestes sont essentiellement exécutés sur un plan 2D car les utilisateurs semblent rencontrer des difficultés à dessiner des formes en 3D, comme montré dans (KELA 06). C'est pour cette raison que nous considérons ces mouvements comme sémaphoriques et non comme physiques (ou incarnés) car l'utilisateur reste dans une tâche de reproduction d'un tracé, comme c'est le cas pour les gestes de cette catégorie.

### Gestes physiques

Les gestes physiques peuvent être couplés aux écrans tactiles et aux capteurs de mouvements (il n'existe pas, à notre connaissance, de techniques de ce type sur un clavier physique).

**Gestes sur écrans tactiles.** Plusieurs gestes physiques ont été proposés pour enrichir l'interaction sur les écrans tactiles en exploitant des mouvements que nous faisons dans la vie courante. Par exemple Boomerang (KOBAYASHI 07) (FIG. 23) utilise deux métaphores : celle du lancer et celle du boomerang. Cette technique consiste à lancer un objet en dehors de l'écran et à le faire revenir sur demande en cliquant sur un cercle affiché sur l'interface graphique. L'utilisateur peut alors facilement déplacer du contenu entre deux fenêtres.

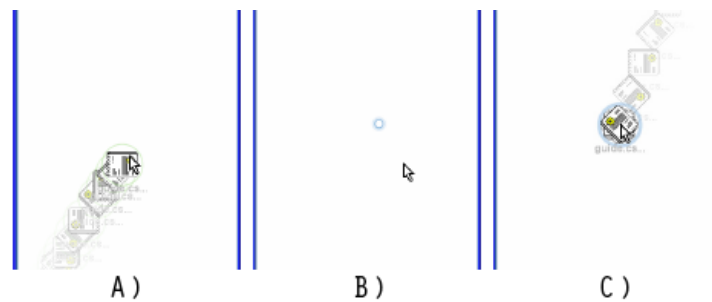


Figure 23. Boomerang : A) l'utilisateur lance un objet hors d'une fenêtre ; B) un cercle apparaît ; C) il ouvre une autre fenêtre et clique sur le cercle resté visible. L'objet lancé apparaît alors sur la nouvelle fenêtre.

De manière générale, les gestes physiques se basent sur la détection de mouvements complexes : les mouvements de lancer avec les gestes de type "Flick" ou "Swipe" (GEIBLER 98) ; les mouvements oscillatoires illustrés dans plusieurs travaux tels que Rubbing (OLWAL 08), Radial Scroll (SMITH 04), Virtual Scroll Ring (MOSCOVICH 04), CurveDial (SCHRAEFEL 05), Motion-Pointing (FEKETE 09) et CycloStar (MALACRIA 10) ; les roulements de doigts dans SimPress (BENKO 06) et RollPad (ONISZCZAK 04) ; ou l'utilisation de plusieurs doigts comme dans les désormais populaires gestes des TouchPad d'Apple. Ces gestes exploitent au mieux les facultés physiques que nous avons acquises dans le monde réel : par exemple les gestes "Flick" sont basés sur la métaphore du lancer, ceux exposés dans Rubbing (OLWAL 08) utilisent les mouvements que l'on effectue habituellement avec une gomme tandis que d'autres exploitent les capacités de l'homme à effectuer des mouvements quasi harmoniques (pendant des oscillations) (GUIARD 93). Nous détaillons l'ensemble de ces techniques dans le CHAPITRE 5.

**Gestes incarnés.** L'utilisation de capteurs de mouvements permet à l'utilisateur d'effectuer des gestes dans l'espace. Dans sa définition de l'interaction incarnée, Fishkin (FISHKIN 98) pose le constat suivant<sup>5</sup> : nous avons l'habitude de secouer, pencher, tourner les objets dans la vie courante, pourquoi ne le ferions nous pas pour contrôler des objets numériques. Plusieurs techniques illustrant ce concept ont été proposées : pour coupler deux dispositifs en les choquant l'un contre l'autre (HINCKLEY 03), ou pour contrôler les mouvements d'une balle évoluant sur l'écran en reconnaissant les chocs des doigts sur le dispositif (LINJAMA 04). Dans (HASSAN 09) un geste directionnel permet d'envoyer des images sur un ordinateur à proximité, et dans (CHONG 09) ces mêmes mouvements permettent d'entrer un mot de passe. Ces techniques utilisent des gestes impulsifs, c'est-à-dire que le mouvement est brusque (ou correspond à un choc), par opposition aux gestes fluides qui sont basés sur des mouvements non brusques comme l'inclinaison du dispositif pour changer l'affichage de l'écran en mode portrait ou paysage (HINCKLEY 00) (FIG. 24). Ils peuvent aussi servir à naviguer dans un menu (OAKLEY 09), une liste de contacts (FISHKIN 00), une image (BARTLETT 00) ou une carte comme dans MotionLens produit par la société Realeyes 3D. Enfin, ils peuvent être utilisés pour entrer du texte comme dans TiltText (WIGDOR 03). Le CHAPITRE 6 présente les avantages et inconvénients des gestes fluides et impulsifs.



Figure 24. L'inclinaison permet de passer du mode portrait au mode paysage.

---

<sup>5</sup> Citation originale : "Like a traditional workstation, interaction with these devices is through a pointing device (a pen rather than a mouse) on a display, plus a few physical buttons. But compare this to a paper artifact, such as a notebook. We not only write on it, but we flip, thumb, bend, and crease its pages. We have highly developed dexterity, skills, and practices with such artifacts, none of which are brought to bear on computational devices. So, why can't users manipulate devices in a variety of ways—squeeze, shake, flick, tilt—as an integral part of using them?"

### 3.2 Techniques de visualisation

Plusieurs taxonomies ont été proposées pour classer les techniques de visualisation (SHNEIDERMAN 96) (CHITTARO 06). Nous utilisons celle de Pook (POOK 01) inspirée de (HERMAN 00) et basée sur plusieurs types de multiplexage<sup>6</sup>. Avant de présenter l'état de l'art, nous illustrons par trois exemples les différents multiplexages d'information (FIG 25).

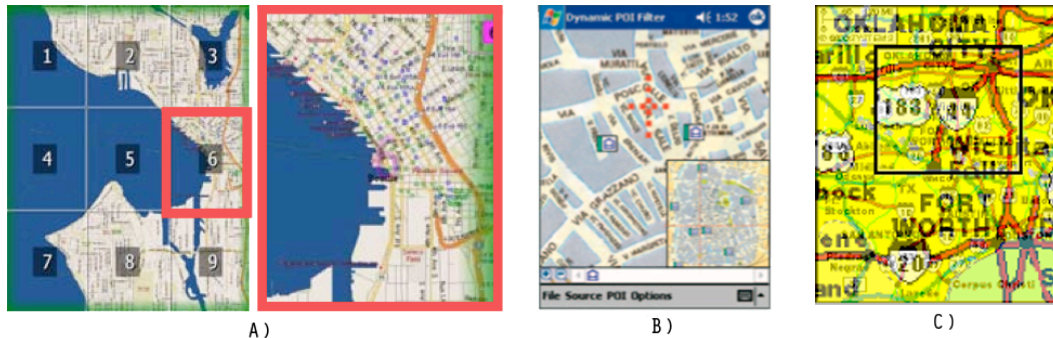


Figure 25. Les différents multiplexages d'information : A) temporel (ZoneZoom), B) spatial (Overview + Detail) et C) en profondeur (Macroscopic).

- Multiplexage temporel : différentes parties des données sont affichées au même endroit, à un moment différent. Ce qui est affiché à un temps  $t$  est contrôlé par le système ou directement par l'utilisateur. Dans la FIGURE 25.A, la zone encadrée en rouge, correspondant à un niveau de zoom, prend la place de la vue initiale.
- Multiplexage spatial : différentes parties des données sont affichées simultanément. Ceci implique l'utilisation de techniques comme les vues Overview + Detail ou encore les vues Focus + Context. Dans la FIGURE 25.B, la vue courante est affichée en même temps que la vue zoomée qui se trouve dans un cadre en bas à droite.
- Multiplexage en profondeur : différentes parties des données sont superposées au même endroit. Dans la FIGURE 25.C, la vue courante est affichée au dessus de la vue zoomée grâce à un effet de transparence.

#### *Multiplexage temporel*

Les techniques de multiplexage temporel sont majoritairement basées sur la navigation multi-échelle. Nous présentons tout d'abord les interfaces zoomables et le zoom sémantique. Nous parlons ensuite des interfaces utilisant des méthodes de filtrage de l'information.

<sup>6</sup>Consiste à faire passer plusieurs flux d'informations à travers un seul support de transmission. (WIKIPEDIA)

**Interfaces zoomables.** Les interfaces zoomables (BEDERSON 95) (POOK 01) utilisent le zoom pour aider l'utilisateur à naviguer dans les données. Dans ZoneZoom (ROBBINS 04) (FIG 25.A) une carte est divisée en neuf zones et chacune de ces zones est associée à une touche du clavier physique. En appuyant sur une touche, la zone de la carte correspondant à la touche est zoomée et prend toute la place à l'écran. La carte est alors à nouveau découpée en neuf zones. Ce mécanisme est aussi mis en pratique dans certains navigateurs Web. Par exemple dans (BAUDISCH 04) (ROTO 06) la page est pavée et l'utilisateur, en cliquant ou en effectuant un geste avec le stylet, peut zoomer sur un de ces pavés et le visualiser en détail. La taille des pavés peut s'adapter à la quantité d'information qu'ils contiennent comme dans les Treemaps zoomables (BLANCH 07). Par exemple dans (ENGDAHL 05) (FIG 26.A) les discussions d'un forum sont affichées dans un Treemap et l'utilisateur navigue grâce au stylet dans les différentes catégories en choisissant une région du Treemap. On retrouve le même mécanisme appliqué à la navigation dans une bibliothèque de photos dans PhotoMesa (KHELLA 04) (BEDERSON 01).

**Zoom sémantique.** Si le zoom est "optique" (les objets graphiques occupant plus ou moins de place à l'écran selon le niveau de zoom), il peut aussi être "sémantique" (la quantité de détails et le type de représentation varient selon le niveau de zoom). C'est le cas de LaunchTile (KARLSON 05) (FIG 26.B). Une matrice de 36 applications est affichée ainsi que des boutons circulaires disposés entre 4 applications. Lorsque l'utilisateur clique sur un bouton, l'interface zoome sur les applications autour du bouton. L'apparence des objets change selon le niveau de zoom : les icônes se transforment en miniatures, puis en applications pouvant être manipulées interactivement.



Figure 26. A) Un Treemap pour représenter les discussions d'un forum ; B) LaunchTile utilise le zoom sémantique pour naviguer dans les applications ; C) Fathumb permet de filtrer parmi 9 catégories grâce à une analogie spatiale avec les touches du clavier.

**Filtrage.** Les techniques de filtrage reposent sur le même principe de multiplexage temporel que les interfaces zoomables : elles permettent de spécifier des catégories pour visualiser un sous ensemble des données. Le niveau de détail varie donc dans le temps. Par exemple, avec Fathumb (KARLSON 06), l'utilisateur navigue dans une base de donnée de type pages jaunes grâce à des filtres. Les données contiennent plusieurs attributs (facettes)



permettant de construire des arbres hiérarchiques et de créer des filtres. Sur la **FIGURE 25.C**, la partie inférieure de l'écran propose neuf catégories qui activent un filtre en fonction de la touche de clavier physique qui est appuyée. Ce même principe est appliqué dans (**DACHSELT 07**) pour naviguer dans une bibliothèque musicale.

### *Multiplexage spatial*

Nous présentons plusieurs catégories de multiplexage spatial : les vues Overview + Detail, les vues Focus + Context et les vues permettant de suggérer un contexte en dehors de l'écran.

**Vues Overview + Detail.** Les vues Overview + Detail (**FIG 25.B**), sont des exemples de multiplexage spatial où l'utilisateur visualise en parallèle deux flux d'information. Généralement la vue zoomée (Detail) prend tout l'écran et une vue générale (Overview) est placée au premier plan dans un coin de l'écran. Cette technique permet à l'utilisateur de connaître la position de la vue zoomée dans la vue générale et lui évite donc d'effectuer de nombreuses opérations de zoom arrière pour se repérer. Bien qu'elles aient été prouvées utiles sur PCs (**HORNBAEK 01**), les vues Overview + Detail sont difficilement utilisables sur petits écrans. En effet, l'espace écran utilisé pour afficher la vue générale est trop petit pour permettre à l'utilisateur de se repérer facilement (**CHITTARO 06**). Une solution possible est de donner des indications à l'utilisateur sans afficher le contenu de la vue générale. Par exemple dans Zen (**BURRIGAT 08**) (**FIG 27**), la vue zoomée prend tout l'espace écran mais la vue générale n'est pas affichée dans un coin de l'écran : seul un cadre noir permet à l'utilisateur de connaître les proportions de la carte. Un cadre rouge permet ensuite à l'utilisateur de savoir la position de la vue zoomée sur la carte.

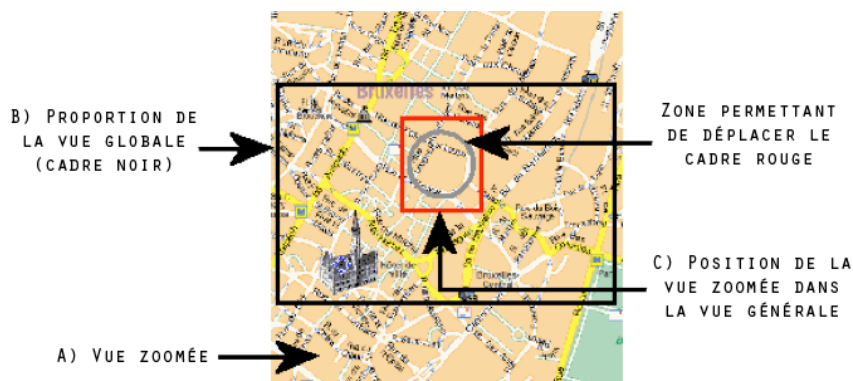


Figure 27. Zen : A) vue zoomée ; B) contour de la vue générale indiquant à l'utilisateur les proportions totales de la carte ; C) cadre rouge permettant à l'utilisateur de savoir la position de la vue zoomée sur la carte.

**Vues Focus + Context.** Les vues Focus + Context (**CARD 99**) évitent la séparation des contenus en affichant la vue zoomée (Focus) et la vue générale (Context) dans une vue unique. C'est le cas des représentations Fish-Eyes (**FURNAS 82**) dont le principe est de distordre les données

pour que les objets sous le focus soient agrandis alors que les objets en périphérie sont rétrécis. Ce type de représentation peut varier selon les objets d'intérêt : les graphiques (SARKAR 92), les arbres hyperboliques (LAMPING 95), les listes (LECOLINET 06), les menus (BEDERSON 00) ou les navigateurs Web (BAUDISCH 04).

Les techniques Focus + Context ne sont pas toujours basées sur des déformations optiques, lesquelles peuvent parfois rendre les informations difficilement visualisables, en particulier le texte (HOLMQUIST 97). Il est parfois préférable de garder les proportions des objets visuels comme dans Flip Zooming (HOLMQUIST 97) (BJÖRK 99) (FIG 28.A) où les pages d'un document sont toutes représentées sous forme de miniatures, hormis la page courante qui ne subit pas de distorsions. DateLens (BEDERSON 03) (FIG 28.B) est basé sur le même principe : un calendrier global est affiché et le jour sélectionné est agrandi au centre. L'espace écran utilisé par les autres jours n'est consacré qu'aux informations indispensables. Lorsque l'utilisateur choisit un autre jour, celui-ci prend toute la place au centre de l'écran.

Ce principe est aussi utilisé dans AppLens (KARLSON 05) pour naviguer dans une matrice d'applications. SpiraList (HUOT 06) (FIG 28.C) propose une technique de représentation Focus + Context d'une liste de contact. Celle-ci est affichée en spirale afin que l'intégralité des contacts soit visible sur une même vue. Cette technique emploie aussi du multiplexage temporel lorsque la liste est trop longue pour être entièrement affichée sur la spirale : plusieurs contacts commençant par les mêmes lettres sont regroupés pour former un groupe, qui peut ensuite être élargi par l'utilisateur lorsqu'il souhaite sélectionner un contact du groupe.

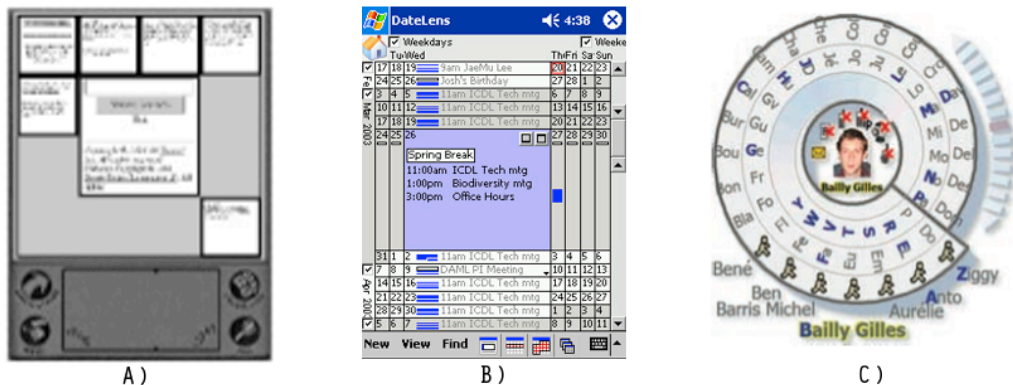


Figure 28. Techniques Focus + Context : A) Flip Zooming ; B) DateLens ; C) SpiraList.

**Suggérer un contexte en dehors de l'écran.** D'autres techniques consistent à suggérer un contexte qui serait à l'extérieur de l'écran. Dans Halo (BAUDISCH 03) (FIG 29.A), des objets sont disposés sur une carte. Chaque objet situé à l'extérieur de l'écran possède un cercle englobant dont une partie apparaît à l'écran. Grâce à la courbure et la position de la courbe, l'utilisateur peut prédire la distance approximative et la direction d'un objet. Wedge (GUSTAFSON 08) (FIG 29.B), une extension de cette technique, affiche des triangles à la place



des cercles. Une extrémité du triangle est formée par un objet en dehors de l'écran et les deux autres extrémités sont calculées, selon l'éloignement de l'objet, pour apparaître à l'écran. De même que précédemment, la distance séparant les deux extrémités visibles du triangle ainsi que sa position permettent de prédire l'écartement et la position de l'objet. Cette amélioration de Halo permet d'éviter une confusion lorsque l'objet est sur une diagonale de l'écran (car la courbure du cercle serait alors la même quelle que soit la distance des objets).

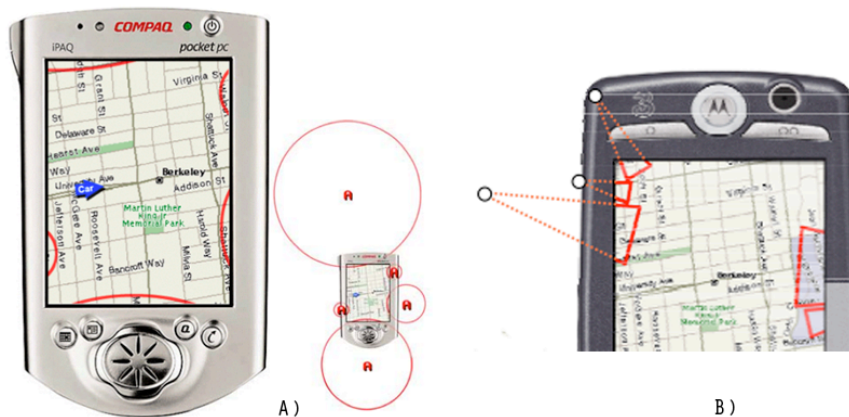


Figure 29. Suggérer l'existence d'informations en dehors de l'écran : A) Halo ; B) Wedge.

### *Multiplexage en profondeur*

Les techniques de multiplexage en profondeur se basent sur le principe de superposition de couches (layers). Plusieurs vues peuvent être affichées en même temps sur un même espace en utilisant des mécanismes de transparence comme dans les SeeThrough Tools et les Toolglasses (BIER 94). Dans Macroscope (LIEBERMAN 97) (FIG 25.C), une carte et la vue zoomée de cette carte sont superposées pour permettre à l'utilisateur de ne pas perdre le contexte dans lequel il se trouve. D'autres études proposent également de superposer des vues locales et des vues globales (HARRISON 95) (COX 98). Cette notion de couches peut être reliée aux techniques de réalité augmentée qui superposent par exemple les images filmées par une caméra du dispositif avec des informations contextuelles. Notamment dans (REILLY 05) l'utilisateur déplace son dispositif mobile au dessus d'une carte pour visualiser des données complémentaires.

Il est toutefois difficile pour l'utilisateur de visualiser séparément les différentes couches de données. Une alternative est d'utiliser le mouvement : lorsque deux couches bougent l'une par rapport à l'autre, elles sont plus faciles à discerner. Ce principe est notamment mis en œuvre dans Zoomit (POOK 00) (FIG. 30) qui est une interface zoomables de navigation dans une bibliothèque. Lorsque l'utilisateur zoome continûment, une couche représentant la vue globale de la bibliothèque est superposée en transparence à la vue principale. L'utilisateur peut facilement distinguer les deux couches de données car la vue zoomée bouge au fur et à mesure de la manipulation interactive.

Position du focus dans  
la couche de contexte

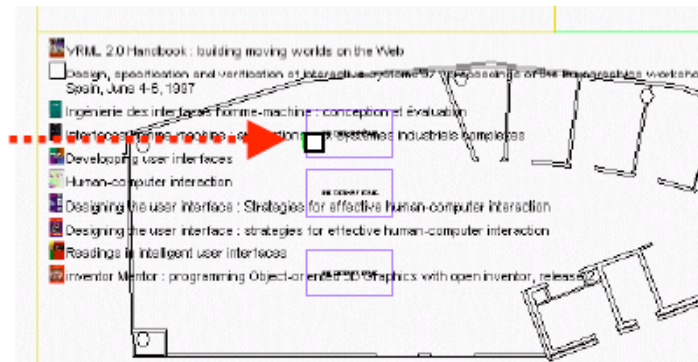


Figure 30. Dans Zoomit, les deux couches sont plus facilement discernables grâce aux mouvements initiés par l'utilisateur lors d'un zoom.

## 4. Conclusion

Nous avons présenté un espace de conception qui se décompose en deux parties. Tout d'abord l'*espace problème* dresse la liste des contraintes spécifiques aux différentes modalités humaines de perception et d'action pouvant être utilisées sur dispositifs mobiles. Nous avons classifié ces contraintes en trois catégories : les limitations matérielles, les éléments à prendre en compte par rapport aux capacités humaines et les contraintes dues à l'environnement. Nous avons ensuite proposé une *taxonomie* des techniques de visualisation et d'interaction sur dispositifs mobiles en les organisant grâce à une définition plus logicielle du terme modalité (une modalité étant vue comme un couple <dispositif, langage d'interaction>). Grâce à trois types de langage d'interaction (déictique, sémaphorique ou physique) pouvant être mis en œuvre sur dispositifs mobiles, nous avons ainsi exposé un panorama des techniques existantes. Nous montrons ci-dessous comment ce travail peut être exploité pour concevoir de nouvelles techniques sur dispositifs mobiles et comment il peut être étendu à d'autres plates-formes.

### 4.1 Conception de techniques pour dispositifs mobiles

Notre espace vise à aider la conception de techniques pour dispositifs mobiles grâce à ses deux facettes complémentaires.

Tout d'abord, l'*espace problème* peut servir lors de la conception pour évaluer les avantages et les faiblesses d'une technique. Les contraintes que nous avons présentées peuvent être vues comme une liste de guides auxquels doivent satisfaire les techniques pour être adaptées aux dispositifs mobiles. Elles peuvent notamment être utilisées en addition d'un espace de critères (tels que ceux de (MCCALL 77) (ABOWD 92) (NIELSEN 93) ou (BAILLY 09)). Le concepteur peut alors vérifier que la technique satisfait à ces critères ainsi qu'aux contraintes que nous avons identifiées.

Nous utilisons plus particulièrement les facteurs d'utilisabilité définis par Shneiderman (SHNEIDERMAN 04), lesquels sont au nombre de cinq :

- *Rapidité* : temps d'accomplissement d'une tâche.
- *Précision* : nombre d'erreurs lors de la réalisation d'une tâche.
- *Apprentissage* : temps nécessaire pour apprendre les actions d'une tâche.
- *Mémorisation* : qualité du savoir retenu sur l'utilisation d'une application après une heure, une journée, un mois.
- *Satisfaction* : plaisir de l'utilisateur à utiliser les différents aspects de l'interface.

Nous considérons aussi le critère de *compatibilité* des heuristiques de Scapin & Bastien : "Il faut qu'il y ait accord entre les caractéristiques des utilisateurs et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part" (BASTIEN 93). Le second aspect doit particulièrement être pris en compte lors de la conception d'une application sur dispositifs mobiles. Il en effet utile de créer des techniques qui soient compatibles pour favoriser leur utilisation conjointe au sein d'une même application.

D'autre part, la *taxonomie* des techniques d'interaction et de visualisation montre que, bien qu'il existe de nombreuses techniques qui tentent de répondre aux contraintes identifiées dans l'espace problème, il reste encore des problèmes non résolus. Nous nous sommes alors penchés sur ces problèmes et avons proposé de nouvelles techniques d'interaction :

- Nous proposons deux techniques à base de gestes déictiques qui améliorent le pointage de petites cibles sur écran tactile. TapTap est basée sur un zoom automatique permettant d'effectuer deux taps précis, et MagStick fournit un bâtonnet télescopique qui contrôle un curseur. Nous les évaluons face à nos contraintes, ainsi que par rapport aux critères de rapidité, précision et satisfaction.
- Le LeafMenu réintroduit un mode expert dans les menus linéaires grâce à des raccourcis gestuels sémaphoriques inspirés du Flower Menu (BAILLY 08). Nous l'évaluons par rapport à nos contraintes et vérifions son adéquation à l'égard du critère mémorisation.
- Les MicroRolls sont des gestes physiques de roulement du pouce qui enrichissent le modèle d'interaction des écrans tactiles, tout en étant compatibles avec les techniques existantes. Nous évaluons les MicroRolls selon nos contraintes et les critères de rapidité, satisfaction et compatibilité.
- TimeTilt, est une technique d'interaction incarnée (gestes physiques et capteurs de mouvement) qui facilite la navigation entre plusieurs vues et qui exploite les avantages du multiplexage temporel. Nous l'évaluons par rapport à nos contraintes et au critère de compatibilité.

Un autre aspect apporté par cette taxonomie est de montrer que peu des techniques avancées que nous avons présentées sont intégrées dans les interfaces des dispositifs commerciaux. Ceci peut s'expliquer de plusieurs façons : 1) les questions de propriété intellectuelle qui empêchent parfois les concepteurs d'utiliser certaines techniques déposées ; 2) Les guides de style établis dans certains systèmes comme l'iPhone peuvent freiner les développeurs à intégrer des techniques différentes ou nouvelles. De plus les utilisateurs risquent de ne pas accepter des techniques non familières ou qui divergent trop des solutions connues. 3) Enfin il n'est pas facile de connaître l'ensemble des techniques existantes et d'évaluer leurs atouts et leurs faiblesses. Et c'est pourquoi nous avons établi cette taxonomie. Nous montrerons dans les chapitres suivants comment ce travail nous a aidé à concevoir de nouvelles techniques et nous proposerons également, pour chacune d'elles, divers exemples d'adaptation à des applications existantes. Nous montrons ainsi les possibilités d'intégration de nos techniques dans des applications et invitons les concepteurs à s'en inspirer pour créer d'autres techniques.

## 4.2 Conception de techniques pour d'autres plates-formes

L'approche que nous proposons n'est pas réservée à l'interaction sur mobiles. La taxonomie prend en compte des techniques provenant d'autres plates-formes (PCs, tablettes tactiles, grands écrans et tables tactiles). De plus, la définition de la modalité que nous exploitons vient de l'interaction multimodale et peut être appliquée à d'autres domaines.

L'espace problème est plus spécifique aux dispositifs mobiles mais il pourrait être adapté à d'autres plates-formes, en particulier à l'interaction sur larges surfaces tactiles qui présente des similarités avec les petites surfaces. Par exemple au niveau des contraintes matérielles, il n'y a pas souvent un clavier physique à disposition, l'écran tactile souffre des mêmes limitations et l'interaction se fait également au doigt. En ce qui concerne les contraintes dues à l'environnement, ces tables tactiles peuvent également être utilisées dans un environnement bruyé où les modalités vocale et auditive ne peuvent être utilisées aisément.

Enfin, la taille des grands écrans interactifs rend difficile l'accès aux bords de l'écran selon la position de l'utilisateur. Ce problème a motivé la création de diverses techniques visant à améliorer le Drag & Drop (COLLOMB 06), en particulier celles techniques qui agissent sur la cible pour la rapprocher du pointeur comme Push-and-pop (COLLOMB 05), Drag-and-pop (BAUDISCH 03) et son extension Drag-and-pick (BAUDISCH 03), et celles qui agissent sur le pointeur pour le rapprocher de la cible telles Drag-and-throw (HASCOËT 03) et Push-and-throw (HASCOËT 03). Ces exemples pourraient être étudiés de manière plus approfondie pour voir dans quelle mesure ces techniques pourraient être adaptées au cas des dispositifs mobiles et, inversement, comment les techniques favorisant l'accès aux bords de l'écran sur dispositif mobile pourraient être exploitées sur grands écrans.

## Espace de conception **2**

# Pointage sur écrans tactiles



"92% des dispositifs mobiles sont vendus avec un navigateur internet. 63% ont une caméra. 1 téléphone sur 5 est équipé d'un capteur de mouvement"

The Size of the Mobile Industry in 2009 (communities-dominate.blogs.com)

<b>1. Introduction</b>	<b>57</b>
<b>2. Etat de l'art</b>	<b>57</b>
2.1 Techniques à curseur décalé	58
2.2 Techniques de franchissement	60
2.3 Techniques de pavage	61
2.4 Techniques à base de gestes sémaphoriques	63
2.5 Pointage et entrée de texte	65
2.6 Synthèse de l'état de l'art	66
<b>3. TapTap et MagStick</b>	<b>69</b>
3.1 Présentation	69
3.2 Propriétés	71
3.3 Implémentation	74
<b>4. Expérimentation</b>	<b>75</b>
4.1 Plan d'expérimentation	75
4.2 Résultats	77
4.3 Discussion	81
<b>5. Intégration</b>	<b>86</b>
<b>6. Conclusion</b>	<b>90</b>

## 1. Introduction

La sélection d'objets est une tâche fréquente dans tous les systèmes interactifs. La méthode la plus répandue sur écrans tactiles (souvent appelée Direct Touch) consiste à appuyer sur l'écran à l'exacte position de la cible puis à relever le doigt. Cette technique est adaptée pour la sélection de cibles relativement larges (9.2 mm (PARHI 06)), mais son efficacité décroît très fortement lorsque les cibles sont de petite taille. Comme mentionné dans le CHAPITRE 2, cette technique de sélection est sujette à l'imprécision, l'occultation et l'instabilité si l'interaction se fait avec les doigts, et l'accès aux cibles près des bords de l'écran est plus difficile dans le cas de l'interaction au pouce.

Il peut cependant être avantageux de garder des petites cibles pour économiser l'espace écran et ainsi afficher plus de cibles. C'est notamment la stratégie mise en place dans certains modèles existants sur le marché (Windows Mobile et BlackBerry) qui proposent des interfaces dotées de cibles d'environ 3mm de diamètre (VOGEL 07) (REN 00) (principalement destinées à une interaction au stylet). Mais cette taille est trop petite par rapport à la largeur du pouce, et c'est pourquoi des techniques spécifiques de pointage doivent être conçues.

Dans ce chapitre nous nous intéressons aux techniques facilitant la sélection de petites cibles sur écran tactile à l'aide des doigts et plus spécifiquement du pouce. Nous présentons tout d'abord l'état de l'art et nous vérifions dans quelle mesure les techniques existantes répondent aux contraintes identifiées dans le CHAPITRE 2. Nous proposons ensuite deux nouvelles techniques, TapTap et MagStick (ROUDAUT, HUOT 08), que nous avons évaluées par une expérimentation contrôlée. Avant de conclure, nous présentons des alternatives et des exemples d'intégration de nos techniques dans des applications.

## 2. Etat de l'art

Malgré les récentes innovations, les contraintes de l'interaction à une main sur écran tactile (occultation, imprécision, instabilité) et les difficultés d'accès aux bords de l'écran ne sont pas complètement résolues par les techniques de pointage proposées récemment. Dans cette section nous présentons les solutions existantes que nous organisons en quatre catégories : les techniques qui utilisent un curseur décalé, les techniques de franchissement, les techniques de pavage, et enfin celles qui sont basées sur des gestes sémaphoriques. Nous parlons aussi des techniques de pointage pour l'entrée de texte puis faisons une synthèse.



## 2.1 Techniques à curseur décalé

Pour éviter que le pouce occulte le point détecté par l'écran tactile, plusieurs techniques proposent d'afficher un curseur de sélection à une certaine distance du pouce. Elles sont basées sur le principe du Take-Off et d'Offset Cursor (POTTER 88) (SEARS 91).

### *Offset Cursor*

Offset Cursor (POTTER 88) (SEARS 91) s'utilise comme suit : lorsque l'utilisateur appuie sur l'écran, un curseur apparaît au dessus du point de contact. Les déplacements du doigt permettent de faire bouger le curseur, lequel se trouve toujours à une distance fixe de la position du doigt. L'utilisateur sélectionne la cible sous le curseur en relâchant le doigt de l'écran. Cette technique permet d'éviter l'occultation puisque le pouce ne cache jamais l'objet que l'on souhaite sélectionner, et elle augmente la précision du pointage. Elle a cependant trois inconvénients. Premièrement, elle est significativement plus lente que la méthode Direct Touch (POTTER 88) (SEARS 91). Dans (BAUDISCH 07), les auteurs montrent que les utilisateurs ont tendance à dépasser et manquer la cible car ils ont des difficultés à estimer la distance exacte entre le curseur et leur doigt. Pour atteindre une cible, ils doivent donc faire des *ajustements de distance* (appelés "net correction distance") qui ralentissent le temps de pointage. Deuxièmement, elle n'apporte pas d'aide pour sélectionner les cibles proches des bords de l'écran, en particulier pour les cibles situées en bas qui sont impossibles à pointer puisque le curseur apparaît au dessus du doigt. Troisièmement, elle est sujette au problème d'instabilité car des saccades du curseur peuvent se produire lorsque le doigt se déplace sur l'écran, ce qui réduit la précision du pointage. Plusieurs alternatives ont été proposées pour répondre à ces inconvénients et sont présentées ci-après.

### *Adaptative Offset Cursor*

Adaptative Offset Cursor, proposé dans SpiraList (HUOT 06), est une évolution d'Offset Cursor qui améliore l'accès aux bords de l'écran. Elle consiste à ajouter, en plus du décalage vertical fixe du curseur, un décalage horizontal qui varie selon la position du doigt sur l'écran. Ce décalage vaut zéro au centre et augmente vers les bords, ce qui permet d'afficher le curseur plus près des bords que ne l'est le doigt. Néanmoins, cette technique peut demander davantage d'entraînement car la distance entre le doigt et le curseur varie. Enfin, comme Offset Cursor, elle ne permet pas de sélectionner des cibles situées en bas de l'écran car le décalage vertical du curseur reste toujours fixe.

**Shift**

Shift (VOGEL 07) diminue le temps de pointage en réduisant la distance d'ajustement du curseur grâce à un mécanisme de réplique : lorsque l'utilisateur appuie sur l'écran, la partie de l'interface cachée sous le doigt est répliquée vers un endroit de l'écran non occulté par le doigt (FIG. 31.A). L'utilisateur affine ensuite la sélection grâce à un curseur se déplaçant dans la zone répliquée, et relâche l'écran lorsque le curseur est correctement positionné sur une cible. L'utilisateur appuie donc directement sur la cible ce qui permet de minimiser la distance à parcourir pour placer le curseur. Ceci contraste avec les deux précédentes techniques où l'utilisateur doit estimer la distance entre le curseur et son pouce pour pointer la cible au plus près, ce qui est une opération non triviale pour un utilisateur non entraîné.

Cette réduction de temps de pointage a été prouvée dans différentes études (VOGEL 07) (YATANI 08) : Shift est plus rapide que Offset Cursor (environ trois fois plus sur des cibles de 3mm). Il est plus intéressant de noter que Shift a un comportement différent selon la taille des cibles. La zone déportée apparaît après un délai qui dépend de la taille de la cible (plus la cible est large plus le délai est long). Cette propriété a l'avantage de ne pas ralentir la sélection de cibles de grande taille car l'utilisateur peut appuyer directement sur la cible comme avec la méthode Direct Touch qui s'avère efficace dans ce cas. Ainsi, contrairement à Offset Cursor, il n'y a pas d'ajustement à effectuer si la cible est large.

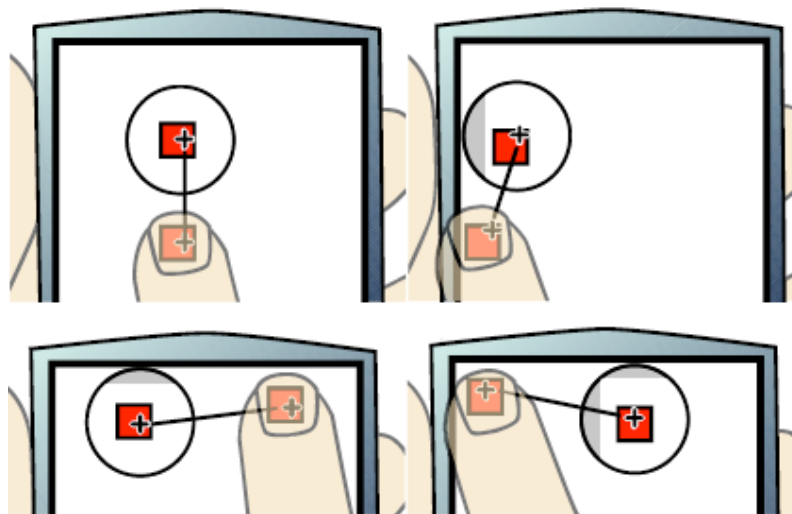


Figure 31. Shift

Une autre propriété de Shift est que le curseur est toujours à une distance fixe du doigt mais la direction varie selon la place disponible à l'écran. Mais si ce mécanisme permet de sélectionner les cibles placées en bas de l'écran, il ne facilite pas pour autant l'accès aux bords de l'écran car l'utilisateur doit toujours placer son doigt près des bords pour sélectionner les cibles qui s'y trouvent (FIG. 31). De plus, le pouce occulte la zone répliquée lorsque l'utilisateur appuie dans la zone supérieure gauche de l'écran (pour un droitier).

Enfin, pour éviter les soubresauts du curseur lors de l'interaction au doigt, Shift utilise un filtre passe-bas pour lisser le tracé du doigt (VOGEL 07) (SEARS 91). Ce filtre augmente la stabilité des points lors des petits mouvements sans introduire de décalages lors de mouvements amples ou rapides.

## 2.2 Techniques de "franchissement"

### Burst

Burst (GUNN 09) (FIG. 32.A) a été proposée comme extension de Shift pour la sélection multiple : lorsque l'utilisateur appuie sur l'écran, seules les cibles situées à proximité du doigt sont déportées vers une zone non occultée par le doigt. Quand l'utilisateur déplace son doigt il laisse une trace qui met en surbrillance les cibles qu'elle croise. Les cibles que l'utilisateur a choisies sont sélectionnées au relâchement du doigt, ceci permettant une sélection multiple. Un inconvénient de cette technique par rapport à Shift est que la sélection peut être imprécise. Il est en effet aussi difficile de tracer précisément un trait sur une petite cible que de la pointer. Néanmoins, avec sa stratégie d'amener les cibles potentielles vers le pouce, Burst évite à l'utilisateur de se positionner trop près des bords. Enfin, comme dans Shift, le pouce occulte les cibles déportées lorsque l'utilisateur appuie dans la zone supérieure gauche de l'écran (pour un droitier).

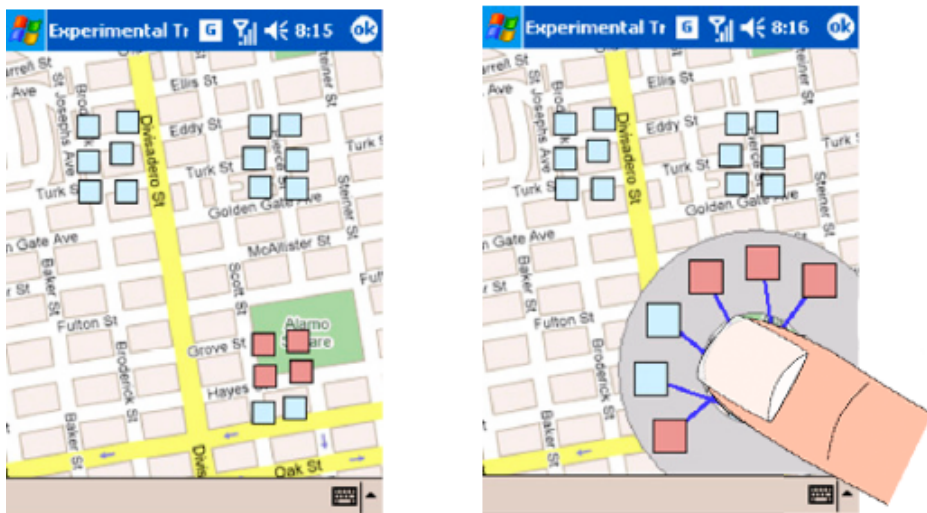


Figure 32. Burst rapproche les cibles potentielles vers le pouce.

### 2.3 Techniques de pavage

La loi de Fitts<sup>7</sup> (MACKENZIE 92) nous indique que pour diminuer le temps de pointage, il faut réduire la distance entre la cible et le curseur ou augmenter la taille de la cible. Par exemple Shift réduit la distance entre la cible et le curseur en permettant à l'utilisateur de pointer directement sur la cible avant d'affiner la sélection. Une autre façon de réduire le temps de pointage est de découpler les mouvements effectués dans l'espace moteur (mouvements du doigt ou de la souris) de ceux effectués par le curseur dans l'espace visuel. Par exemple, le Pointage sémantique (BLANCH 04) consiste à modifier le gain ("control-display ratio") du curseur selon les objets survolés. La vitesse du curseur est alors plus ou moins lente selon l'importance des cibles, et plus rapide sur les espaces vides. Dans le même esprit, le Pointage d'objet (GUIARD, BLANCH 04) consiste à supprimer l'espace vide entre les cibles, le curseur sautant alors d'une cible à une autre. Ce mécanisme de pavage de l'espace est appliqué dans Starbust (BAUDISCH 08) et Thumbspace (KARLSON 08) que nous présentons ci-après.

#### Starbust

Starbust est une amélioration de Direct Touch qui consiste à paver l'espace : l'écran est divisé en zones, chaque zone activant une cible spécifique. Les cibles sont donc plus larges dans l'espace moteur, mais leurs représentations dans l'espace visuel ne changent pas. Le pavage peut être effectué grâce à des diagrammes de Voronoi (FORTUNE 87) qui pavent intelligemment l'espace quand les cibles ne sont pas distribuées de façon homogène (FIG. 33.A). L'utilisateur peut alors appuyer dans la région de la cible pour la sélectionner. Cependant ce pavage dépend de l'homogénéité du placement des cibles. Comme on peut le constater sur la FIGURE 33.A, les régions du centre sont plus petites et difficilement sélectionnables. Starbust (BAUDISCH 08) (FIG. 33.B) propose une modification de l'algorithme pour répondre à ce problème. Une première étape (FIG. 33.C) consiste à créer pour chaque point une ligne qui le connecte à un bord de l'écran, de façon à ce que l'ensemble des lignes soit réparti régulièrement sur l'écran. En calculant ensuite le diagramme de Voronoi de l'ensemble des lignes sur l'écran, on obtient des régions de surfaces pratiquement égales, ce qui augmente la zone active de la cible et évite à l'utilisateur de se positionner trop près des bords.

---

<sup>7</sup>  $T = a + b \log_2(D/W + 1)$

T étant le temps de pointage, a et b des constantes empiriques variant selon le dispositif utilisé, D la distance à la cible, et L la largeur de la cible.

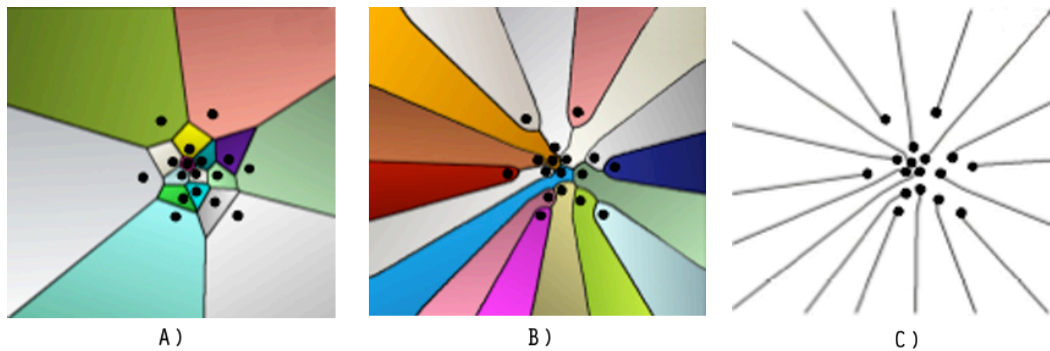


Figure 33. A) Diagramme de Voronoi; B) Starbust ; C) Première étape de l'algorithme de Starbust.

Cette technique possède cependant des inconvénients. Tout d'abord la représentation des cibles ne change pas mais il est nécessaire de paver l'espace visuel afin d'afficher les zones associées à chaque cible, ce qui peut surcharger l'interface graphique. D'autre part, les régions deviennent trop petites pour la sélection lorsqu'il y a beaucoup de cibles à l'écran. Enfin, le pavage supprime les espaces vides qui sont habituellement utilisés pour annuler la sélection. En cas d'erreur, l'utilisateur doit donc attendre que la fonctionnalité associée à la cible se lance et la fermer pour retourner à son activité. Ce mécanisme de suppression des espaces vides implique donc l'existence d'une technique de désélection.

### *Thumbospace*

Thumbospace (KARLSON 08) a été conçue pour faciliter l'accès aux bords de l'écran. En appuyant sur un bouton physique, une vue miniaturisée de l'interface est affichée au centre de l'écran (FIG. 34.A). Cette vue agit comme une vue radar de l'interface. Lorsque l'utilisateur appuie dessus (FIG. 34.B), elle disparaît et un curseur apparaît dans la région de l'interface correspondante (FIG. 34.C). Par exemple, si l'utilisateur appuie dans un coin de la vue miniaturisée, le curseur apparaîtra dans le coin de l'écran, ce qui permet d'atteindre facilement les bords de l'écran. La technique fonctionne ensuite comme les techniques à curseur décalé, sauf que lorsque l'utilisateur déplace le doigt sur l'écran, le curseur saute de cible en cible en ignorant les espaces vides selon le même principe que le Pointage d'objet. Au relâchement du doigt, la cible positionnée au centre du curseur est sélectionnée.

La vue radar de Thumbospace est partitionnée de façon à ce que chaque cible présente dans la vue originale soit associée à une région de la vue radar. Ces régions sont des rectangles de taille variable, contrairement à Starbust où elles sont plutôt de forme patatoïdale. Lorsque le pouce entre dans une région de la vue radar, le curseur apparaît sur la cible correspondante dans la vue originale. Le mouvement du curseur vers une autre cible n'est cependant déclenché que si le pouce se déplace de plus de 10 pixels, ce qui permet d'éviter les mouvements anarchiques du curseur lorsque l'utilisateur fait des gestes involontaires.

Un inconvénient de Thumbspace est que la distance entre le curseur et le doigt n'est pas fixe mais varie selon la position du doigt sur l'écran. Lorsque le doigt est au centre de l'écran, cette distance est alors nulle et le curseur se trouve donc sous le doigt ce qui provoque l'occultation. Pour répondre à ce problème, les auteurs proposent d'utiliser Direct Touch pour les cibles situées au centre, le pavage de l'espace permettant alors d'augmenter la précision comme dans Starbust (BAUDISCH 08). Cependant, comme les solutions basées sur le pavage de l'espace moteur, Thumbspace comporte des inconvénients : l'absence d'espaces vides pour annuler une opération, et, surtout, des performances décroissantes lorsque le nombre de cibles augmente.



Figure 34. Thumbspace : A) l'appui sur un bouton physique provoque l'affichage d'une vue radar de l'interface ; B) l'utilisateur appuie sur le coin de la vue radar ; C) le curseur apparaît près des bord de l'écran alors que le pouce reste dans une zone facilement accessible.

### 2.4 Techniques à base de gestes sémaphoriques

Espace (YATANI 08) et les Sliding Widget (MOSCOVICH 09) sont des techniques basées sur des gestes directionnels qui permettent de désambiguïser et donc à améliorer la précision de Direct Touch. Nous détaillons ces deux techniques ci-après.

#### Escape

Escape (YATANI 08) est une technique modifiant l'aspect visuel des cibles, celles-ci ayant une représentation particulière permettant d'indiquer une direction (FIG. 35). Lorsque l'utilisateur appuie sur la cible et fait le geste dans la direction qu'elle indique, celle-ci est sélectionnée. Le problème d'occultation n'a alors plus d'importance puisque l'utilisateur voit le geste avant de l'effectuer. Un autre avantage d'Escape est que l'utilisateur commence son geste directement sur la cible et fait ensuite un geste pour préciser sa sélection. Ces deux opérations peuvent être enchaînées rapidement. Notamment, les auteurs de la technique montrent que Escape est plus rapide que Shift.

La différence de performance d'Escape peut s'expliquer par la nature du déplacement du doigt. Dans le cas de Shift l'utilisateur doit pointer la cible sur l'écran puis positionner précisément le curseur sur la cible. Avec Espace, d'une part le pointage sur l'écran n'a pas besoin d'être précis (l'utilisateur peut pointer à coté de la cible), d'autre part l'utilisateur doit effectuer un geste linéaire à longueur non fixée (scale-independant), qui demande lui aussi moins de précision. Le pointage peut donc être effectué rapidement et le problème d'instabilité lors de l'interaction au doigt ne joue pratiquement pas. Par contre, nous notons que Escape ne résout pas le problème d'accès aux bords de l'écran.

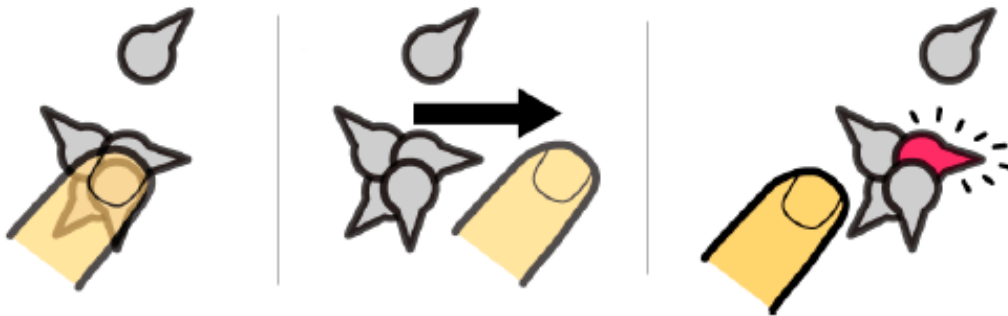


Figure 35. Escape.

### Sliding Widgets

Les Sliding Widgets ([MOSCOVICH 09](#)) reposent sur un mécanisme similaire à Escape. La représentation des cibles indique aussi une direction, mais la différence réside dans le retour visuel utilisé. En effet, les Sliding Widget utilisent de petits interrupteurs pour faciliter la compréhension du geste à effectuer ([FIG. 36](#)). Pour activer une cible, l'utilisateur appuie sur la cible puis effectue un geste directionnel de façon à déplacer l'interrupteur dans le sens qui convient. On retrouve notamment ces Widgets dans l'iPhone pour activer ou désactiver des fonctionnalités.



Figure 36. Sliding Widgets.



## 2.5 Pointage et entrée de texte

L'entrée de texte est souvent traitée à part du pointage de cibles car il existe des différences sensibles. Notamment, l'ensemble des cibles est connu et limité (alphabet), et des modèles prédictifs peuvent être utilisés pour filtrer les mauvaises sélections. Nous présentons ci-après les techniques d'entrée de texte par pointage qui existent.

### Configuration des lettres

Les claviers adoptent généralement des configurations de touches qwerty ou azerty, plus habituelles pour les utilisateurs. Ce n'est cependant pas la stratégie la plus performante et d'autres configurations ont été proposées pour améliorer la saisie de texte. Par exemple, dans le clavier Dvorak, les lettres qui s'utilisent successivement sont côte à côte afin de minimiser les déplacements. En considérant la loi de Fitts, ce clavier est donc plus rapide puisque les distances sont réduites. D'autres configurations ont été proposées comme le clavier Metropolis, Knits, Fitaly ou Opty (la typologie de Poirier (POIRIER 04) synthétise l'ensemble de ces approches). Un autre exemple est ShapeWriter (ZHAI 08) (FIG. 37.A) qui repose aussi sur une organisation différente des lettres et où l'utilisateur peut écrire un mot sans relâcher le doigt de l'écran. Il déplace en effet son doigt sur les lettres et n'a pas d'ajustements à faire au dessus de l'écran, ce qui augmente la vitesse d'écriture.

Ces claviers améliorent la vitesse de saisie mais ont cependant des inconvénients. Ils nécessitent une phase d'apprentissage plus ou moins longue car l'utilisateur doit se familiariser avec la nouvelle configuration. De plus, ils ne répondent pas aux problèmes de l'imprécision du pointage avec les doigts. Une solution possible à l'imprécision est celle que l'on trouve sur le clavier de l'iPhone : la lettre zoomée apparaît au dessus du doigt lorsque l'utilisateur appuie sur une lettre (FIG. 37.B). Mais ce mécanisme est surtout utile aux utilisateurs novices qui ne tapent pas assez vite et ont, de ce fait, le temps de regarder les lettres zoomées.

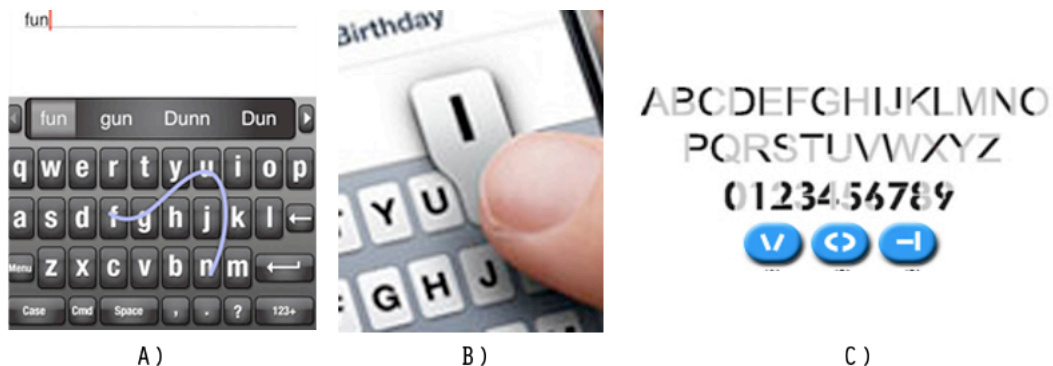


Figure 37. A) ShapeWriter ; B) Clavier virtuel sur l'iPhone ; C) UniGlyph.



### Claviers réduits

Afin d'augmenter la taille des touches et de rendre le pointage plus précis, une solution est d'employer un clavier T12 à neuf touches, la répétition d'une touche permettant d'accéder aux lettres. Dans UniGlyph (BELATAR 07), la saisie d'une lettre consiste à reconstruire sa forme en cliquant sur des boutons virtuels qui représentent des primitives graphiques (FIG. 37.C). Par exemple, comme la lettre B contient une boucle, il faut appuyer sur le bouton "boucle". Un dictionnaire trie ensuite les mots pouvant correspondre. Cette technique pourrait s'avérer efficace pour l'interaction aux doigts puisqu'il n'y a que trois boutons à afficher sur l'écran et qu'ils peuvent donc être de taille suffisante.

## 2.6 Synthèse de l'état de l'art

La FIGURE 38 résume de manière synthétique les propriétés des techniques précédemment exposées. Comme le montre ce tableau et le résumé qui suit, aucune des techniques existantes ne résout à elle seule l'ensemble des problèmes posés par l'interaction au pouce.

Direct Touch est la technique la plus rapide proposée à ce jour. Elle est efficace pour la sélection de larges cibles mais s'avère peu utilisable pour la sélection de petites cibles. Le pouce occulte en effet le point de contact ainsi que la cible, ce qui engendre beaucoup d'erreurs de pointage. C'est aussi le cas de Burst, qui est une technique de franchissement, car il est aussi difficile de tracer précisément un trait sur une petite cible que de la pointer. Pour résoudre ce problème d'échelle entre la taille du doigt et celle des cibles, plusieurs stratégies sont possibles. La première consiste à agrandir les cibles seulement dans l'espace moteur (et non dans l'espace visuel) pour économiser de la place à l'écran, comme dans Starbust.

La seconde solution est "d'amener le pouce à l'échelle des cibles" grâce à un curseur qui peut avoir différents comportements. Dans Offset Cursor, il est à une distance fixe du doigt ce qui évite l'occultation mais ne facilite pas l'accès aux bords. Pour résoudre ce problème, l'Adaptative Offset Cursor introduit une variation de distance horizontale qui dépend de la position du doigt, mais comme Offset Cursor, elle ne permet pas de sélectionner les cibles en bas de l'écran. Avec Shift, la distance "curseur-doigt" est fixe mais la direction du curseur varie en fonction de la position du doigt, ce qui permet l'accès au bas de l'écran. Enfin Thumbspace est la seule technique qui facilite l'accès à tous les bords de l'écran. La distance "curseur-doigt" varie selon la position du doigt (horizontalement et verticalement). Mais ce mécanisme entraîne une occultation par le doigt sur le curseur au centre de l'écran.

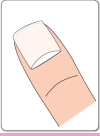
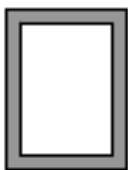


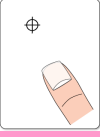
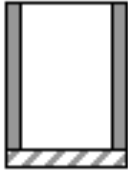
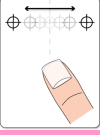
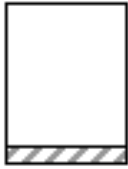

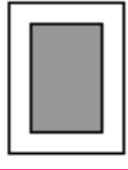
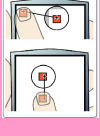
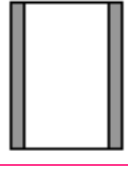
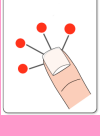

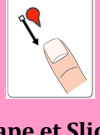
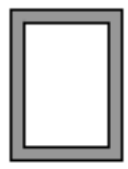
	Occultation	Imprécision	Instabilité	Evènements	Accès bords
 <b>Direct Touch</b>	Partout	Très imprécis	Pas d'instabilité	Press + Release	
 <b>Starburst</b>	Non gênante	Imprécis si beaucoup de cibles	Pas d'instabilité	Press + Release	
 <b>Offset Cursor</b>	Non	Précis mais lent	Peut être instable	Press + Release + Drag	
 <b>Adaptative O C</b>	Non	Précis mais lent	Peut être instable	Press + Release + Drag	
 <b>Thumbspace</b>	Au centre	Curseur aidé par le Pointage d'objet	Seuil de déplacement qui évite les saccades	Press + Release + Drag	
 <b>Shift</b>	En haut à gauche	Précis	Filtre passe-bas pour l'instabilité	Press + Release + Drag	
 <b>Burst</b>	En haut à gauche	Tracé imprécis	Instabilité lors du tracé	Press + Release + Drag	
 <b>Escape et Sliding Widgets</b>	Non gênante lors du geste	Imprécis si trop de cibles	Rapidité du geste empêche l'instabilité	Press + Release + Drag	

Figure 38. Tableau récapitulatif de l'état de l'art vis à vis des problèmes de l'interaction au pouce (Les zones grisées dans la colonne de droite sont plus difficiles à atteindre et les zones rayées sont impossibles à atteindre).

Enfin la troisième stratégie consiste à effectuer des gestes sémaphoriques, comme dans Escape et les Sliding Widgets, lesquels ont l'avantage de pouvoir être effectués rapidement, mais ne résolvent pas le problème d'accès aux bords de l'écran. Nous notons également que certaines techniques ont des inconvénients spécifiques :

- **Contraintes du pavage** : Starbust et ThumbSpace utilisent une méthode de pavage qui augmente le coût des erreurs car il n'existe pas d'espaces vides pour annuler la sélection. La densité des cibles affecte aussi les performances de ces techniques.
- **Consommation des évènements** : Direct Touch et Starbust consomment deux évènements de l'écran tactile (Press, Release), alors que les autres techniques en consomment un de plus (Drag). Ces dernières risquent donc d'être difficilement compatibles avec les techniques utilisant cet événement pour le défilement de liste ou pour lancer des commandes gestuelles comme nous l'avons vu au [CHAPITRE 2](#).
- **Occupation spatiale** : seule Direct Touch n'affecte pas l'interface graphique, contrairement à Starbust, Escape et les Sliding Widgets. En effet le pavage de l'espace ou la modification de l'aspect des cibles est coûteux en place à l'écran puisque ces modifications sont permanentes. Notamment l'espace visuel pavé de Starbust peut gêner la visualisation des informations. Les autres techniques présentées ont un fonctionnement transitoires (ou "transients" ([JACKOBSEN 07](#))) qui ne modifient l'interface que temporairement. Ils apparaissent sur demande et disparaissent après l'opération de pointage ou suite à une annulation.
- **Connaissance des données de l'application** : certaines techniques sont dépendantes des données de l'application. Notamment, la taille et la position des cibles doivent être connues pour interagir avec Shift puisque cette technique se déclenche après un délai dépendant de leur taille. C'est aussi le cas de Burst qui a besoin de connaître la position des cibles pour les rapprocher du doigt de l'utilisateur. Enfin les méthodes de pavage utilisées dans StarBust et ThumbSpace nécessitent de connaître la position des cibles sur l'écran. Cet aspect n'est pas gênant pour l'interaction en elle-même mais peut avoir des implications sur l'implémentation des techniques (une technique indépendante des données des applications peut être implémentée dans une nouvelle version du système d'exploitation sans nécessiter de modifier des applications déjà existantes).

### 3. TapTap et MagStick

Dans l'état de l'art, nous avons mis en évidence diverses stratégies pour améliorer la sélection de petites cibles sur écran tactile, et nous avons vu qu'aucune des techniques présentées ne répondait pleinement à l'ensemble des contraintes de l'interaction au pouce. C'est pourquoi nous proposons deux nouvelles approches : TapTap qui a été conçue comme une amélioration de Direct Touch et MagStick qui est une évolution de Offset Cursor et des autres techniques à curseur décalé. Nous détaillons leur fonctionnement avant de présenter leurs propriétés, puis nous décrivons leurs implémentations respectives.

#### 3.1 Présentation

##### TapTap

TapTap vient d'une idée simple : si un seul tap n'est pas assez précis pour sélectionner de petites cibles, un second tap pourra sans doute désambiguïser cette sélection. Plus précisément : le premier tap définit une *zone d'intérêt* sur l'écran (FIG. 39.A) ; cette zone est agrandie et prend la forme d'une *vue zoomée* au centre de l'écran (FIG. 39.B) ; le second tap sélectionne alors la cible désirée dans la zone zoomée (FIG. 39.C) (ou annule la sélection si le tap se fait sur un espace vide).

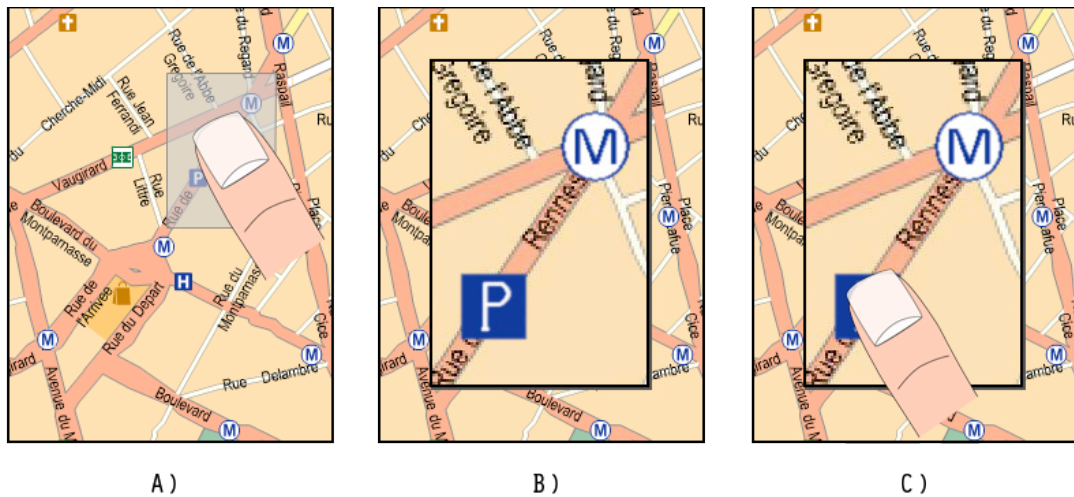


Figure 39. TapTap. A) le premier tap définit une zone d'intérêt ; B) la zone d'intérêt prend la forme d'une vue zoomée au centre de l'écran ; C) le second tap sélectionne la cible, désormais assez large pour être sélectionnée avec le pouce.

TapTap est donc basée sur une stratégie de multiplexage temporel (décrite dans le [CHAPITRE 2](#)) où le premier tap sert à spécifier le focus d'intérêt dans la vue originale, laquelle est zoomée pour afficher les cibles à l'échelle du pouce. Une caractéristique importante de TapTap par rapport aux techniques multi-échelles usuelles est qu'il n'y a pas de contrôle interactif du

niveau de zoom, ni du déplacement en X,Y car ils sont tous deux ajustés automatiquement. L'interaction est de ce fait rapide et fonctionne pratiquement comme un quasimode : le premier tap active le mode de sélection et affiche le zoom, tandis que le second ferme la vue et quitte le mode de sélection. De plus TapTap facilite l'accès aux bords de l'écran car la zone d'intérêt choisie par l'utilisateur est assez large pour lui éviter de s'approcher trop près des bords, et la vue zoomée apparaît toujours au centre.

### MagStick

MagStick est une technique à curseur décalé. Elle empêche l'occultation du doigt sur le curseur, mais contrairement aux techniques proposées auparavant, la distance "curseur-doigt" est dynamiquement ajustée par l'utilisateur : 1) quand l'utilisateur appuie sur l'écran, il définit un point de référence (FIG. 40.A) ; 2) en déplaçant son doigt, un bâtonnet ("stick" d'où le nom de la technique) en deux parties apparaît (FIG. 40.B) : les parties émanent du point de référence et finissent pour l'une au curseur et pour l'autre à la position du doigt ; 3) comme chaque partie du bâtonnet est de même longueur et dans la même direction, l'utilisateur contrôle le curseur en déplaçant son pouce sur l'écran (changer la taille d'une partie change automatiquement la taille de l'autre partie) ; 4) l'utilisateur relâche son doigt pour sélectionner la cible sous le curseur (ou un espace vide pour annuler la sélection).

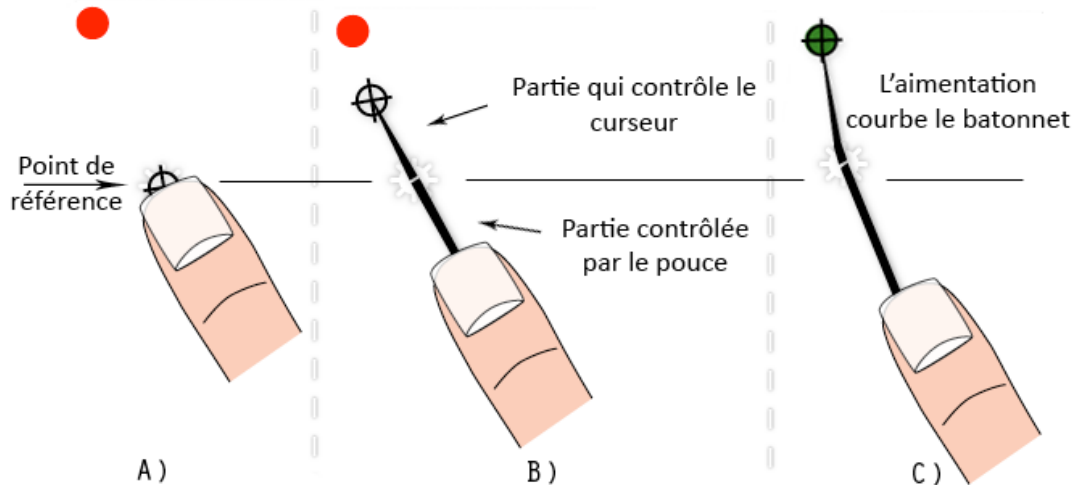


Figure 40. MagStick : A) l'utilisateur appuie sur l'écran pour définir un point de référence ; B) en déplaçant le pouce un bâtonnet ("stick") télescopique apparaît. Le stick a un mouvement inversé ; C) le curseur est magnétisé par les cibles pour éviter les erreurs lors du relâchement du pouce.

Le mécanisme de bâtonnet télescopique a deux avantages par rapport aux techniques existantes : d'une part l'utilisateur contrôle la *distance* curseur-doigt ce qui évite l'occultation au centre comme avec Thumbspace où la distance dépend de la position absolue du doigt sur l'écran. D'autre part, l'utilisateur contrôle la *direction* du curseur ce qui permet d'atteindre

facilement les bords de l'écran contrairement à Offset Cursor ou Shift. Il peut par exemple se positionner au centre de l'écran pour atteindre une cible située en bas de l'écran. De plus contrairement aux autres techniques, le doigt s'éloigne des cibles au lieu de s'en rapprocher, ce qui permet d'avoir une meilleure visibilité sur l'interface.

MagStick exploite de plus le Pointage sémantique (BLANCH 04) car les cibles attirent le curseur comme s'il était aimanté. Ceci a pour effet de courber le bâtonnet (FIG. 40.C) et de le bloquer dans la cible. Lorsque l'utilisateur déplace son doigt suffisamment, le curseur sort de la cible et les deux parties du bâtonnet se réalignent. Contrairement aux autres techniques, le curseur est donc plus stable lorsqu'il se trouve sur une cible, ce qui permet à l'utilisateur de faire un geste "grossier" qui peut dépasser un peu la cible, et qui empêche le doigt de la quitter lors de mouvements involontaires (particulièrement au relâchement du doigt).

### 3.2 Propriétés

Cette section compare les propriétés et les avantages respectifs de nos techniques. En particulier, nous montrons qu'elles offrent des solutions efficaces aux problèmes de l'interaction au pouce.

#### *Imprécision et rapidité*

Les deux techniques tentent de contourner la loi de Fitts de différentes manières. TapTap se base sur un espace multi-échelle (qui peut être vu comme une technique de zoom proche de celles proposées dans (MANKOFF 00) et (GROSSMAN 05)). Comme montré par Guiard et al. (GUIARD, BEAUDOUIN-LAFON 04), la loi de Fitts s'applique toujours mais l'indice de difficulté est réduit par le zoom : la taille des "cibles" est augmentée dans les deux étapes de l'interaction (la "cible" étant la zone d'intérêt dans le premier cas, et une cible zoomée dans le second cas). Les deux taps peuvent ainsi être exécutés plus rapidement que deux taps standards successifs (cette supposition sera confirmée plus tard par l'expérimentation contrôlée). De plus la vue zoomée apparaît toujours au centre ce qui permet à l'utilisateur d'anticiper la position du second tap. Ces deux propriétés, *augmentation de la taille des cibles* et *anticipation du second tap*, permettent à l'utilisateur d'interagir avec TapTap comme s'il s'agissait d'un double clic avec un déplacement rapide du doigt entre les deux taps. Cet effet a été remarqué pendant les évaluations : les utilisateurs n'ont pas eu l'impression d'effectuer deux taps successifs mais plutôt un geste composé.

Pour améliorer la précision et l'efficacité, MagStick possède deux propriétés. Elle est basée sur le Pointage sémantique qui modifie l'espace moteur et *réduit artificiellement les distances de pointage*, et par la même occasion le temps de pointage. L'autre propriété est due à son design

symétrique qui permet à l'utilisateur de *prédire facilement les mouvements à effectuer*. Un inconvénient notable d'Offset Cursor est que la plupart des utilisateurs ont du mal à estimer la distance entre leur pouce et le curseur tant qu'ils n'ont pas touché l'écran, ce pourquoi ils doivent attendre que le curseur apparaisse avant d'ajuster sa position. Avec MagStick, ce problème est évité car l'utilisateur peut lui-même choisir la distance entre le curseur et le doigt. De plus, comme les deux parties du bâtonnet sont de la même taille, l'utilisateur peut prédire facilement le mouvement qu'il devra faire avant même de toucher l'écran. Lors d'un premier design de MagStick nous avons opté pour une taille variable des deux parties du bâtonnet, comme dans (ALBINSSON 03) : lorsque l'utilisateur étire la partie du bâtonnet accrochée au pouce, l'allongement de l'autre partie est plus ou moins important selon la distance restante pour aller jusqu'aux bords de l'écran. Cependant nous avons abandonné cette idée car elle rendait la technique plus difficilement contrôlable et semblait faire baisser ses performances. Le design final permet ainsi à l'utilisateur de prédire plus facilement l'endroit où il va appuyer.

Une autre propriété de MagStick est qu'elle fonctionne comme Direct Touch pour les larges cibles : lorsque l'utilisateur appuie sur l'écran, le curseur se trouve initialement sous son doigt, ce qui revient à effectuer un tap normal. En présence de petites et larges cibles, les performances ne sont donc pas dégradées. TapTap pourrait également être modifiée pour nécessiter un seul tap lorsque la cible est assez large. La représentation de la cible peut être légèrement changée pour indiquer ce fonctionnement comme dans Escape, mais choisir un seuil de taille de cible pour décider l'apparition de TapTap devrait suffire. Pouvoir sélectionner des cibles de deux manières différentes ne devrait pas poser de problème si une convention uniforme est suivie : les utilisateurs de PCs le font déjà couramment (les icônes de bureau sont double cliquées tandis que les autres nécessitent généralement un seul clic).

### **Occultation**

La stratégie de zoom de TapTap évite l'occultation : comme les cibles sont agrandies, elles sont assez larges pour ne pas être cachées complètement par le pouce (FIG. 41.A). Une propriété originale de MagStick, inspirée par des jeux électroniques de billard ou de golf ou par la métaphore du tir à l'arc comme dans (HASCOËT 03), est que le curseur bouge dans la direction opposée du pouce. Cette stratégie évite l'occultation puisque le doigt s'éloigne de la cible : non seulement le pouce ne cache pas la cible, mais en plus une large partie du contexte visuel de la cible devient visible. Ce mécanisme atténue aussi l'occultation de la main lors de la sélection de cibles proches de la base du pouce, comme montré sur la FIGURE 41.B : le pouce est naturellement placé au centre de l'écran et peut être facilement décalé vers le haut ou vers le bas quand cela est nécessaire.



Figure 41. A) Avec TapTap les cibles sont assez larges pour éviter l'occultation ; B) MagStick évite l'occultation car le pouce s'éloigne de la cible au lieu de s'en rapprocher.

### *Instabilité*

Les problèmes d'instabilité ne se posent pas avec TapTap puisque cette technique ne demande pas à l'utilisateur de déplacer son doigt sur l'écran. Ce n'est pas le cas de MagStick qui est alors sujette au problème d'instabilité. Pour palier cette contrainte, un filtre passe-bas permet de stabiliser le tracé de l'utilisateur. L'effet magnétique permet de plus de bloquer le curseur dans la cible et empêche qu'il ne la quitte lors de mouvements involontaires du doigt. Cependant, cet effet a pour inconvénient de dégrader les performances de MagStick lorsque le nombre de cibles augmente, contrairement à TapTap. En effet si la densité des cibles est importante, les espaces vides seront plus rares, et le curseur aura tendance à sauter d'une cible à l'autre comme pour Thumbspace, ce qui peut rendre la technique plus difficile à contrôler.

### *Zones difficilement accessibles de l'écran*

TapTap et MagStick permettent de sélectionner des cibles quelles que soient leurs positions sur l'écran. Avec TapTap, non seulement le premier tap n'a pas besoin d'être effectué exactement sur la cible (il doit être fait dans son voisinage), mais la vue zoomée apparaît toujours au centre de l'écran. Les cibles proches des bords dans la vue originale apparaissent donc dans une région de l'écran facile à atteindre avec le pouce.

Le bâtonnet télescopique de MagStick permet d'atteindre les cibles près des bords car la distance entre le curseur et le pouce n'est pas fixe mais ajustée par l'utilisateur. Pour ce faire, l'utilisateur est libre de choisir sa propre stratégie. La première consiste à toucher l'écran très près de la cible pour minimiser le déplacement du pouce. Une autre est de commencer systématiquement le geste au centre de l'écran. Toutes les cibles peuvent alors être



atteintes par construction du double bâtonnet : en déplaçant le pouce en bas la cible si celle-ci est dans le haut de l'écran ou inversement. Cette stratégie a été utilisée par la majorité de nos participants durant l'évaluation. Elle a également pour avantage de permettre à l'utilisateur de tenir le dispositif fermement dans la main. La base du pouce est alors dans la partie droite de l'écran (pour un droitier) et le centre de gravité du dispositif est environ au centre de la main.

### *Consommation des évènements*

TapTap n'utilise que deux évènements de l'écran tactile (Press et Release). Elle est donc compatible avec les techniques de déplacements utilisées pour faire défiler une liste ou déplacer un objet (utilisant l'évènement Drag). MagStick utilise l'évènement Drag de l'écran tactile et peut donc poser certains problèmes de compatibilité. Nous verrons des alternatives qui contournent ce problème dans la section **INTÉGRATION**.

### 3.3 Implémentation

#### *TapTap*

La zone d'intérêt sélectionnée par le premier tap doit être assez large pour contenir la cible désirée même si celui ci est relativement loin de la cible. Nous avons choisi une taille de 80 x 120 pixels autour du point de contact (pour un écran QVGA 240 x 320 pixels). Ceci laisse 40 pixels en largeur (1/6 de l'écran) et 60 pixels en hauteur (environ 1/5 de l'écran) de marge pour prévenir tout risque d'erreurs lors du premier tap. Cependant, la taille de la zone d'intérêt est contrainte par le facteur de zoom qui peut être appliqué dans la vue zoomée sur un petit écran, et l'écran ne peut pas être utilisé en entier pour afficher la vue zoomée à cause des problèmes d'accès aux bords. Par conséquent, cette vue est zoomée par un facteur de deux (sa taille est de 160 x 240 pixels) et elle est placée au centre de l'écran pour être facilement accessible au pouce.

En suivant le schéma décrit précédemment, le facteur de zoom des cibles est le même que celui de la vue zoomée (soit 2). Mais ce facteur peut ne pas être suffisant pour rendre les cibles assez larges. Ceci nous a amené à considérer une autre version de TapTap, utilisée dans nos évaluations, qui est dépendante aux données de l'application : la technique connaît l'emplacement des cibles afin de les agrandir par un facteur de trois. Nos observations montrent que ce choix est efficace et que les utilisateurs ne sont pas désorientés par la différence de facteur de zoom entre les cibles et la vue zoomée (aucun des utilisateurs ayant participé à l'évaluation n'a remarqué cette propriété).

### *MagStick*

MagStick nécessite une connaissance des données de l'application pour permettre l'effet magnétique. Chaque cible possède une zone de proximité invisible (plus précisément un cercle englobant de 10.8 mm pour des cibles de 3mm) qui attire le curseur et courbe le bâtonnet. Quand le curseur entre dans une zone de proximité, il est attiré au centre de la cible correspondante. Cette propriété permet d'éviter à l'utilisateur de faire des ajustements ou de dépasser/manquer la cible en sélectionnant un espace vide. Ensuite, lorsque l'utilisateur déplace le curseur hors d'une zone de proximité (de façon à ce que le point, non visible, symétriquement opposé au pouce ne soit plus dans une zone d'attraction), l'effet magnétique s'estompe et les deux parties du bâtonnet se réalignent jusqu'à ce que le curseur soit attiré par une autre cible. Une possible amélioration de la technique serait d'assigner différentes puissances d'attraction aux cibles, comme pour le Pointage sémantique, afin de faciliter la sélection de commandes fréquentes ou d'éviter l'activation de commandes dangereuses. Enfin, pour augmenter la stabilité de la technique nous avons mis en œuvre un filtre passe-bas similaire à celui proposé dans Shift (VOGEL 07).

## **4. Expérimentation**

Nous avons mené une expérience contrôlée pour comparer TapTap et MagStick avec les principales techniques de l'état de l'art : Direct Touch, Offset Cursor, Thumbspace, et Shift (les autres techniques présentées dans l'état de l'art n'étaient pas publiées au moment de cette étude).

### **4.1 Plan d'expérimentation**

#### *Tâche*

La tâche consistait à effectuer une série de pointage de cibles avec les six techniques. Les participants tenaient le dispositif avec leur main dominante et interagissaient avec le pouce. Plusieurs cibles étaient affichées à l'écran, et l'une d'entre elles devait être sélectionnée. Il était demandé aux participants de sélectionner la cible aussi vite et précisément que possible. Avant chaque essai, l'utilisateur appuyait sur un bouton "essai suivant" et une carte apparaissait avec plusieurs cibles bleues, excepté celle à sélectionner qui était rouge (FIG. 42.A). Les cibles bleues servaient de distracteurs pour accroître le réalisme de la tâche de pointage. La cible rouge devenait verte lors que le curseur la survolait (excepté pour Direct Touch et TapTap, cette fonctionnalité n'étant alors pas applicable). L'essai finissait quand l'utilisateur relâchait le pouce de l'écran, qu'il ait réussi ou non la sélection. Il n'avait donc qu'une seule chance de sélectionner la cible. Un retour sonore indiquait si la sélection était correcte.

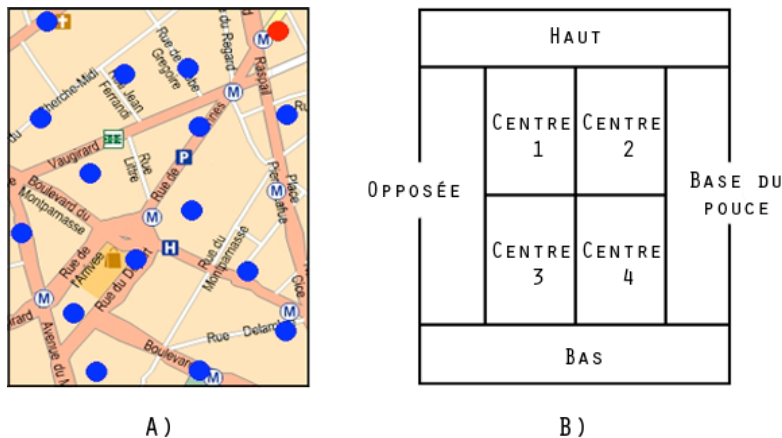


Figure 42. A) la cible rouge doit être sélectionnée parmi les 15 autres distracteurs ; B) L'écran est divisé en 8 zones de même surface.

### Matériel et participants

Les techniques ont été implémentées en C# (.Net Compact Framework) et fonctionnaient sur Windows Mobile 5.0. L'expérience a été réalisée sur un HTC P3600 QVGA (320x240). 12 volontaires (dont 1 femme), âgés de 23 à 47 ans, ont été recrutés dans notre institution et ont reçu une poignée de bonbons pour leur participation. Tous utilisaient un dispositif mobile à écran tactile pour la première fois. Deux participants étaient gauchers et leurs résultats ont été inversés afin que chaque utilisateur puisse utiliser sa main dominante.

### Conditions expérimentales

Nous avons souhaité vérifier si l'efficacité des techniques comparées dépendait de la position de la cible à l'écran. Dans (KARLSON 08) les auteurs ont divisé l'écran en une matrice régulière de 12 cases. Nous avons utilisé un découpage différent, avec 8 zones de surfaces égales (FIG. 42.B). Cette division permet une séparation claire entre les zones du centre de l'écran et les zones proches des bords.

L'efficacité des techniques peut aussi dépendre de la densité des cibles, laquelle varie selon les applications. C'est pourquoi nous avons considéré deux configurations différentes : 16 cibles étaient représentées à l'écran pour la condition de faible densité contre 32 pour celle de forte densité. Ces facteurs peuvent influencer les performances de certaines techniques. Notamment ThumbSpace qui est basée sur un pavage de l'espace, et MagStick qui utilise un effet magnétique. Dans le cas de MagStick, nous avons exploité les avantages du Pointage sémantique pour modifier les forces d'attraction des cibles selon la densité : la taille des zones de proximité des cibles était de 10,8 mm pour la faible densité et de 3,6 mm pour la forte densité.

Afin de réduire le temps d'expérience pour nos participants, nous avons considéré des cibles de 3mm car cette taille correspond à la taille minimale des cibles que l'on peut rencontrer dans les dispositifs mobiles sur le marché. Une amélioration mineure a été faite pour Offset Cursor pour permettre d'atteindre les cibles en bas de l'écran, ceci afin de comparer les différentes techniques de manière équitable. Dans ce cas, l'utilisateur pouvait faire apparaître le curseur sous le pouce en appuyant sur un bouton avant de toucher l'écran. L'analyse des données de l'expérience confirme que cette modification n'affecte pas les résultats (le temps de sélection n'est pas significativement différent entre la zone du bas et les autres zones).

Pendant la tâche, le temps de sélection, les erreurs et les mouvements du doigt ont été collectés. A la fin les participants devaient répondre à un questionnaire pour donner leur opinion et indiquer leur préférence sur une échelle de Likert pour chaque technique testée.

### *Design expérimental*

Nous avons conduit une analyse de variance à mesures répétées chez les mêmes sujets (within-subject). Les variables indépendantes sont les *Techniques* (Direct Touch, Offset Cursor, Thumbspace, Shift, TapTap and MagStick), les *Zones de l'écran* (8 zones illustrées sur la **FIGURE 42.B**), et les *densités* (faible ou forte). La présentation des Techniques a été contrebalancée circulairement entre les participants. Tous ont effectué 32 sélections : 2 fois dans chacune des 8 zones de l'écran pour chacune des 2 densités. Les zones de l'écran étaient ordonnées par une séquence contrebalancée circulairement pour chaque technique. Cette séquence avait pour but d'alterner les régions faciles et difficiles à sélectionner. L'ordre des densités était aléatoire de façon à ce que les utilisateurs ne puissent pas prédire la condition de densité suivante. Enfin, les participants se sont entraînés sur 10 essais avant de commencer chaque technique. Pour résumer le design expérimental était le suivant :

6 *Techniques* x 8 *Zones de l'écran* x 2 *Densités* x 2 *Blocs* = 96 sélections (20 minutes) par participant

## 4.2 Résultats

Une analyse de variance montre que l'ordre de présentation des techniques n'a pas d'effet significatif sur le temps de sélection et le taux d'erreur. Le temps de sélection a été calculé à partir du moment où l'utilisateur relâchait l'écran sur le bouton "essai suivant" jusqu'au moment où il relâchait l'écran (les essais avec erreurs ont été exclus de ce calcul). Les erreurs comprennent les sélections vides aussi bien que les sélections de mauvaises cibles.

*Temps de sélection*

L'ANOVA montre des effets significatifs pour les *Techniques* ( $F_{5,55} = 18.37, p < .001$ ), l'interaction *Techniques x Zones de l'écran* ( $F_{35,268} = 1.63, p < .02$ ), les *Densités* ( $F_{1,11} = 5.45, p < .05$ ) et l'interaction *Techniques x Densités* ( $F_{5,49} = 3.09, p < .02$ ). Les tests de comparaison multiple (tests post hoc de Tukey) toutes conditions de densité confondues permettent de classer les techniques dans l'ordre suivant : Direct Touch (1361.8ms) et TapTap (1616.8ms), MagStick (2730.2ms), Shift (3473.3ms) et Offset Cursor (3379.9ms) (pas de résultats significatifs entre ces deux dernières techniques), et Thumbspace (4498,1ms). Les données montrent que : TapTap est environ 2.1 fois plus rapide que Offset Cursor et Shift et presque 2.8 fois plus que Thumbspace ; MagStick est environ 1.2 fois plus rapide que Offset Cursor, 1.3 fois plus que Shift et 1.6 fois plus que Thumbspace. Les résultats de Direct Touch sont donnés à titre informatif car son taux d'erreurs élevé implique une quantité de données collectées plus petite comparativement aux autres techniques. Toutefois, on peut considérer Direct Touch hors classement puisqu'une technique causant trop d'erreurs est frustrante pour l'utilisateur.

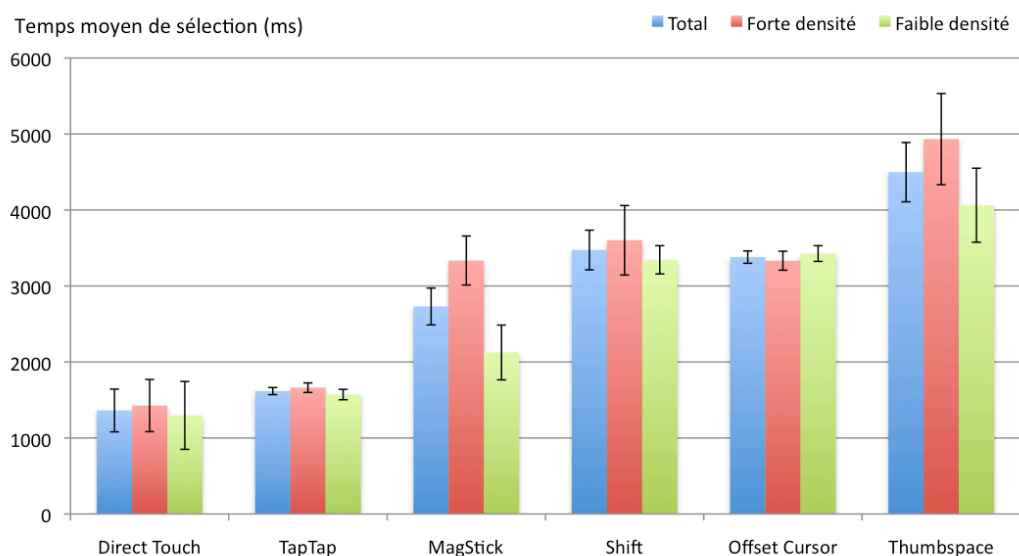


Figure 43. Temps moyen (ms) pour chaque technique (les barres représentent un intervalle de confiance de 95%).

L'analyse de l'interaction *Techniques x Zones de l'écran* montre que les zones de l'écran n'ont pas d'effets significatifs sur les performances de TapTap et MagStick. Par contre l'effet est significatif pour Offset Cursor, moins efficace dans la zone de la "base du pouce" (FIG. 42.B) (2246.2ms de différence en moyenne) et dans la zone "opposée" (1250ms de différence en moyenne) que dans les autres zones. Un effet similaire a été trouvé pour Shift dans la zone "haut" (2445.6ms de différence en moyenne) et la zone "opposée" (936.8ms de différence en moyenne). Ces résultats confirment nos observation durant les évaluations où nous avons vu

les utilisateurs gênés dans ces zones du fait de l'occultation. D'autres effets significatifs ont été trouvés sur Thumbspace, qui est plus efficace sur les bords qu'au centre (1381.9ms de différence en moyenne). Ceci corrobore les explications données précédemment et par les auteurs de la technique (KARLSON 08).

Mis à part Direct Touch, TapTap est la technique la plus rapide et MagStick la seconde. Les bords de l'écran sont atteints plus rapidement avec MagStick que les autres techniques à curseur décalé. Non seulement MagStick est performante pour atteindre les bords mais elle ne gêne pas l'interaction au centre (contrairement à Thumbspace). TapTap est particulièrement rapide quelle que soit la zone de l'écran.

L'analyse de l'interaction *Techniques x Densité* montre que deux techniques sont dépendantes à la densité des cibles : Thumbspace (4062.9ms en faible densité et 4933.2ms en forte densité) et MagStick (2125.3ms en faible densité et 3335ms en forte densité) qui sont moins performantes en condition de forte densité, ce qui n'est pas surprenant étant donné qu'elles sont basées respectivement sur du Pointage d'objet et du Pointage sémantique. Notons que MagStick (2125.3ms) devient significativement plus performante que Offset Cursor (3427.1ms) et Shift (3344.7ms) en condition de faible densité (1571.6ms pour TapTap et 4062.9ms pour Thumbspace), alors qu'aucun résultat significatif ne distingue ces trois techniques en condition de forte densité.

### Taux d'erreurs

Comme précédemment, l'ANOVA montre des effets significatifs pour les *Techniques* ( $F_{5,55} = 59.38, p < .001$ ), l'interaction *Techniques x Zones de l'écran* ( $F_{35,268} = 1.81, p < .001$ ), les *Densités* ( $F_{1,47} = 11.75, p < .001$ ) et l'interaction *Techniques x Densités* ( $F_{5,49} = 3.75, p < .001$ ). Des tests de comparaison multiple (tests post hoc de Tukey) toutes conditions de densité confondues montrent que TapTap a le taux d'erreurs le plus bas (8.3%) et Direct Touch le plus haut (62%) (FIG. 44). Aucun effet significatif n'a été trouvé pour les autres techniques (MagStick (18.7%), Offset Cursor (17.5%), Shift (16.1%) et Thumbspace (22.6%)). De plus, nous avons jusqu'à maintenant combiné les sélections d'espaces vides ou de mauvaises cibles. En distinguant les deux, Thumbspace ne produit que des erreurs de mauvaises sélections (par définition) alors que les erreurs des autres techniques sont plutôt des sélections d'espaces vides.

L'analyse de l'interaction *Techniques x Zones de l'écran* montre que le taux d'erreurs de Thumbspace est plus élevé au centre de l'écran, zone où le doigt occulte le curseur. Aucun effet significatif n'a été trouvé pour les autres techniques. Il y a aussi un effet significatif pour les résultats de Direct Touch qui a un taux d'erreurs plus élevé sur les bords.

L'analyse de l'interaction *Techniques x Densités* montre qu'en condition de forte densité, TapTap (9,9% d'erreurs) a le taux d'erreurs le plus bas alors que MagStick (26% d'erreurs) et Thumbspace (26.5 % d'erreurs) ont les plus hauts taux d'erreurs (sans considérer Direct Touch). En condition de faible densité, MagStick arrive en seconde (10.9%) après TapTap (7.3%), les autres techniques ayant un taux d'erreurs similaire compris entre 17% et 19%.

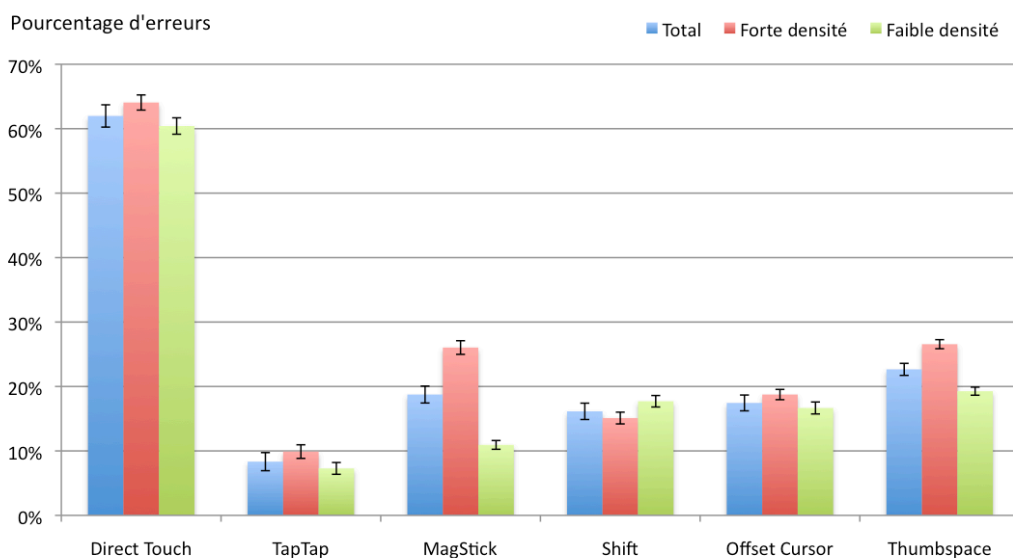


Figure 44. Taux d'erreurs (%) pour chaque technique (Les barres représentent un intervalle de confiance de 95%).

En résumé, TapTap a le taux d'erreurs le plus bas et Direct Touch le plus haut. Les techniques à base de curseurs ont approximativement le même taux d'erreur, excepté Thumbspace dont les performances sont assez faibles. En condition de faible densité, MagStick surpasse les techniques à curseur, mais sa précision baisse en condition de forte densité.

### *Préférences utilisateurs*

Grâce au questionnaire qui a suivi l'évaluation, les participants ont classé les techniques par ordre de préférence : TapTap, MagStick, Shift, Offset Cursor, Thumbspace et enfin Direct Touch. Leurs opinions à propos de la rapidité, la précision, l'attrait, la simplicité, la facilité d'apprentissage et l'aspect "fun" ont été récoltées sur une échelle de Likert (sur 5) (FIG. 45). TapTap est préférée pour tous les critères, excepté pour l'aspect "fun" où elle est placée en second après MagStick. TapTap et Direct Touch sont classées premières pour la rapidité et les utilisateurs estimaient TapTap plus rapide que Direct Touch (ce qui est inexact dans l'expérience réalisée mais serait vrai si l'on devait sélectionner à nouveau les cibles manquées). Direct Touch récolte le moins de points pour la précision, l'attrait et l'aspect fun. Les approches à curseur décalé ont des résultats similaires, avec MagStick et Offset Cursor en dessus de Shift et Thumbspace. MagStick est jugée un peu moins facile à apprendre (arrivant cependant avant Thumbspace) mais est classée première pour l'aspect "fun".

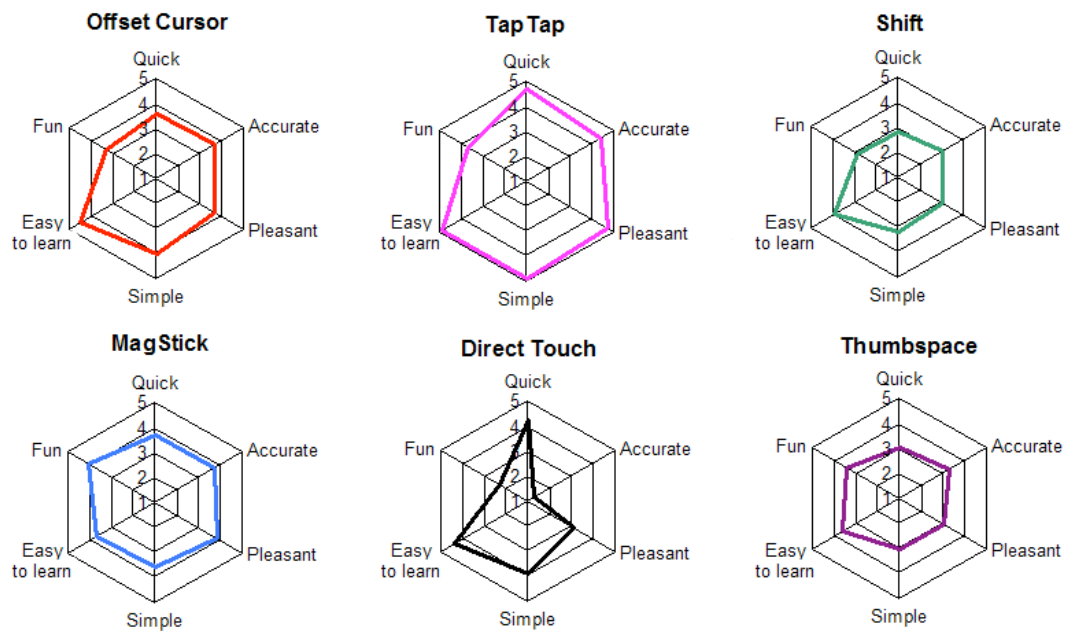


Figure 45. Résultats des questionnaires utilisateurs.

### 4.3 Discussion

Pour mener cette discussion, nous avons établi un modèle analytique des performances des techniques testées, que nous comparons aux résultats obtenus dans notre expérimentation. Ce modèle (FIG. 46) décompose le fonctionnement des techniques en quatre étapes :

- 1) déplacement du doigt au dessus de l'écran
- 2) appui sur l'écran
- 3) déplacement du doigt sur l'écran
- 4) relâchement du doigt.

Ces étapes sont soumises à la loi de Fitts puisque l'on peut distinguer deux tâches de pointage, la première se produisant au dessus de l'écran (étapes 1 et 2), la seconde d'effectuant en déplaçant le doigt sur l'écran (étapes 3 et 4, excepté pour les techniques sans étape 3 comme TapTap et Direct Touch). Le temps d'exécution des ces pointages peut ainsi être réduit en jouant sur la distance à parcourir et la taille des cibles. Les étapes 3 et 4 sont aussi soumises à l'instabilité de l'interaction au doigt qui peut entrainer des soubresauts lors du tracé et du relâchement, et donc affecter les performances de la technique. Dans le tableau qui suit, nous résumons les stratégies utilisées pour réduire le temps de pointage de chaque étape (les cases colorées correspondent à la mise en place d'une stratégie).



	Temps	Déplacement au dessus de l'écran	Appui sur l'écran	Déplacement sur l'écran	Relâchement du doigt
<b>Direct Touch</b>	1 <sup>ère</sup>	A la cible	Pouce trop large pour la cible		Pouce trop large pour la cible
<b>TapTap</b>	2 <sup>nd</sup>	1) A proximité de la cible 2) au centre	Cibles agrandies pour le pouce		Cibles agrandies pour le pouce
<b>MagStick</b>	3 <sup>ème</sup>	Au centre	Introduction d'un curseur à la taille des cibles	Distance réduite Pointage sémantique	Curseur à la taille des cibles. Pointage sémantique
<b>Shift</b>	4 <sup>ème</sup>	A la cible	Introduction d'un curseur à la taille des cibles	Distance de correction réduite	Curseur à la taille des cibles
<b>Offset Cursor</b>	5 <sup>ème</sup>	En dessous de la cible	Introduction d'un curseur à la taille des cibles	Distance de correction	Curseur à la taille des cibles
<b>Thumbspace</b>	6 <sup>ème</sup>	Sur vue miniaturisée au centre	Introduction d'un curseur à la taille des cibles, et cibles agrandies	Distance réduite Pointage d'objet	Seuil de mouvement

Figure 46. Tableau récapitulatif des six techniques de pointage évaluées. Les cases colorées correspondent à la mise en place d'une stratégie pour réduire le temps d'exécution de l'étape.

### *Direct Touch et TapTap*

Le premier constat que nous pouvons faire est que l'utilisation du déplacement du doigt sur l'écran (étape 3) a des répercussions non négligeable sur la technique de pointage. D'une part les techniques utilisant cette phase consomment l'événement Drag de l'écran tactile et peuvent donc être incompatibles avec d'autres fonctionnalités. D'autre part elles sont moins rapides que les techniques n'utilisant pas cette étape. Direct Touch et TapTap, qui ne nécessitent pas cette phase, sont en effet plus rapides que les techniques à curseur décalé (même si TapTap demande le double des opérations de Direct Touch).

TapTap est moins rapide que Direct Touch mais elle a le taux d'erreurs le plus bas. Direct Touch entraîne en effet beaucoup trop d'erreurs, ce qui force l'utilisateur à tenter plusieurs fois de sélectionner la cible. Notre expérimentation ne laissait qu'une seule chance à l'utilisateur de sélectionner la cible. Dans des conditions réalistes, le temps moyen nécessaire pour sélectionner une cible avec Direct Touch serait significativement plus élevé que le temps récolté dans cette évaluation car l'utilisateur serait souvent amené à faire plusieurs tentatives. Ce temps moyen peut être estimé en considérant que les erreurs de sélection de l'expérience nécessiteraient un second essai pour réussir. Dans ce cas, Direct Touch serait à environ 3661,8ms (au maximum puisque le second tap devrait être plus court, le doigt se trouvant déjà au dessus de la cible) alors que TapTap ne nécessiterait que 1919,6ms étant donné qu'elle produit beaucoup moins d'erreurs.

Un autre point intéressant est qu'un tap de Direct Touch (étape 1 à 4) prend nettement plus de temps (1361.8ms) qu'un des deux tap de TapTap (759.1ms pour le premier tap en moyenne et 857.7ms pour le second). Ces résultats vont dans le sens de la loi de Fitts : les cibles étant plus proches et plus larges, il est cohérent que le temps de pointage diminue.

### *Techniques à curseur décalé*

Parmi les techniques à curseur décalé, MagStick est la plus rapide et produit le moins d'erreurs toutes conditions confondues. Ceci peut être expliqué par plusieurs mécanismes. Premièrement, la stratégie la plus utilisée par nos participants consistait à placer le pouce directement au centre de l'écran, ce qui permet ainsi de réduire le temps de réaction de l'étape 1 de notre tableau analytique car l'utilisateur ne réfléchit à l'endroit où il pose son doigt. Nous avons en effet observé que le temps nécessaire à effectuer l'étape 1 (temps de déplacement au dessus de l'écran) est légèrement plus court pour MagStick (891.5ms) que pour les autres techniques à curseur décalé (1129,1ms pour Offset Cursor, 973,8ms pour Shift et 934ms pour Thumbspace). Deuxièmement, le temps nécessaire aux étapes suivantes (2, 3 et 4) est également plus court pour MagStick que pour les autres techniques. Nous pensons que l'utilisateur fait une estimation de la distance à accomplir avec le doigt avant de toucher l'écran grâce à l'effet de symétrie (seulement 1 participant sur 12 a fait des erreurs dues à un mauvais positionnement du pouce à l'appui sur l'écran). Cette rapidité d'exécution est aussi due au Pointage sémantique qui réduit la distance à parcourir dans l'étape 3, et qui évite aussi les erreurs de relâchement à l'étape 4.

Une étude intéressante à mener serait de tester l'effet du Pointage sémantique indépendamment des techniques. Ce mécanisme pourrait en effet être appliqué à Shift et Offset Cursor (même s'il ne permettrait pas de faciliter l'accès aux bords de l'écran pour ces deux techniques). Le Pointage sémantique dépend cependant de la densité des cibles. Lorsque de nombreuses cibles sont présentes à l'écran, nous avons vu que les performances de MagStick sont équivalentes à celles de Shift et Offset Cursor (alors que celles de TapTap restent stables). De ce fait la combinaison du Pointage sémantique à Shift et Offset Cursor pourrait éventuellement dégrader leurs performances.

### *Thumbspace : un cas particulier*

En observant le tableau analytique et les performances de Thumbspace, nous pouvons constater un résultat étonnant : cette technique est la moins rapide et cause le plus d'erreurs (excepté Direct Touch) alors qu'elle présente à chaque étape une stratégie pour réduire le temps de sélection : à l'étape 1, la distance à parcourir est réduite car l'utilisateur doit appuyer sur la vue miniaturisée qui se situe au centre de l'écran ; à l'étape 2, la taille du curseur est réduite et celle des cibles sont agrandie dans l'espace moteur (grâce au pavage de l'espace) ; à

l'étape 3 le Pointage d'objet permet de réduire les distances à parcourir puisqu'il supprime les espaces vides ; et à l'étape 4, un seuil de mouvement permet de stabiliser le mouvement lors du relâchement du doigt pour éviter au curseur de sauter vers une autre cible.

Plusieurs raisons peuvent être à l'origine des performances relativement faibles de Thumbspace. Tout d'abord, l'utilisateur peut mettre plus de temps à l'étape 1 pour choisir l'endroit où il va appuyer. Pourtant le temps nécessaire à l'étape 1 de Thumbspace est assez similaire à celui de Offset Cursor ou Shift. Nous ne pensons pas non plus que les étapes 2 et 4 ralentissent l'interaction car aucun de nos participants n'a été gêné par la taille des cibles, ou par des problèmes de relâchement. Le problème semble donc provenir de l'étape 3 (déplacement du doigt sur l'écran). Nous avons pu en effet constater que les utilisateurs mettaient beaucoup de temps pour effectuer cette étape (3564ms pour Thumbspace alors que les autres techniques ne dépassent pas 2500ms). Le Pointage d'objet n'est peut-être pas une technique optimale pour l'interaction sur un écran tactile, lequel est à la fois un outil de visualisation et un dispositif de pointage absolu. Dans le cas de Thumbspace, la distance "curseur-doigt" n'est jamais la même (horizontalement et verticalement) et varie selon la position du doigt sur l'écran, alors que dans les autres techniques à curseur décalé, soit la distance est fixe (Offset Cursor, Shift), soit elle est sous le contrôle de l'utilisateur (MagStick). Plusieurs participants nous ont d'ailleurs confié qu'ils trouvaient les mouvements du curseur de Thumbspace un peu déroutants. C'est aussi une raison pour laquelle les performances de MagStick décroissent en condition de forte densité. En effet, dans ce cas le curseur peut sauter d'une cible à l'autre, ce qui revient pratiquement à faire du Pointage d'objet.

### *Accès aux bords de l'écran*

Notre expérience montre aussi que le temps de sélection et le nombre d'erreurs ne dépendent pas des positions des cibles sur l'écran avec TapTap et MagStick, ce qui contraste avec les autres techniques. Ces différences peuvent être observées lorsque l'on regarde l'ensemble des gestes effectués sur l'écran avec les six techniques (FIG. 47). Les gestes de MagStick et de Thumbspace sont concentrés au centre de l'écran. Ceux de TapTap sont au centre pour le second tap et décalés des bords par rapport à Direct Touch pour le premier tap. Nous pouvons également observer la présence d'agglomérats de points avec Shift et Offset Cursor, qui correspondent aux ajustements du curseur pour aller sur la cible. Enfin nous voyons les gestes de déplacement du curseur, qui sont plus présents avec Offset Cursor que Shift (où l'utilisateur appuie directement sur la cible) excepté pour la partie supérieure gauche de l'écran où le pouce cache la zone déportée par Shift.

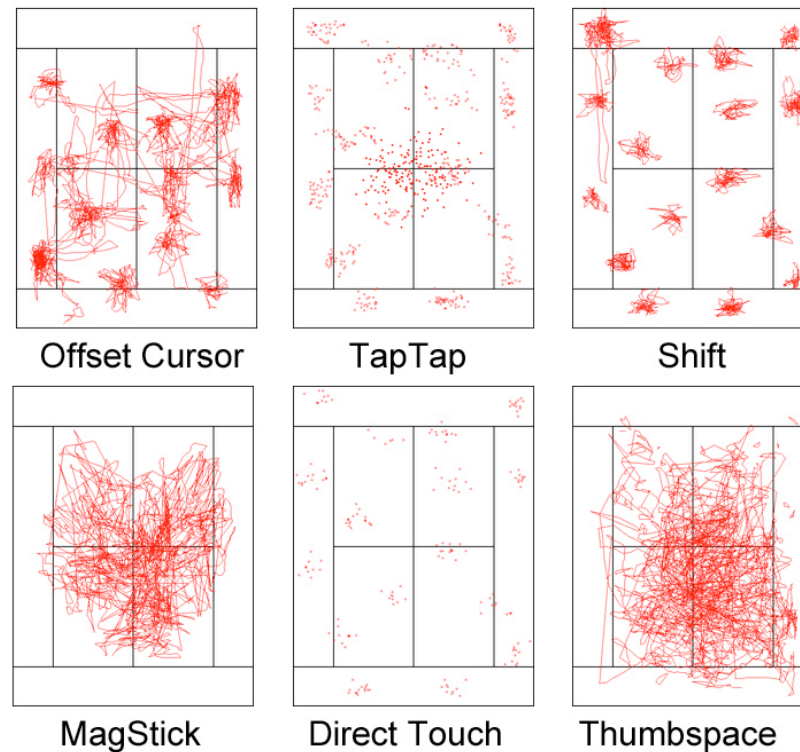


Figure 47. Traces du pouce laissées sur l'écran pour chaque technique.

### Préférences

Les questionnaires indiquent que TapTap est plus simple et plus facile à apprendre que les autres techniques. Nous avons remarqué que plusieurs participants se "concentraient" ou ajustaient leur position pour avoir un "meilleur angle d'attaque" avant de taper l'écran lorsqu'ils utilisaient Direct Touch. Cette préparation n'existe pas avec TapTap puisque le premier pointage peut être "grossier". Offset Cursor est très proche de ce que l'on trouve sur ordinateur (nous avons l'habitude de manipuler un curseur avec la souris) et ceci peut expliquer sa facilité d'utilisation. MagStick et Shift offrent plus de contrôle aux utilisateurs, lesquels les trouvent de ce fait plus difficiles à apprendre (pour comprendre où s'affiche la zone décalée de Shift ou pour manier le mouvement inversé de MagStick). Malgré les scores un peu plus faibles de MagStick en simplicité et apprentissage, c'est la deuxième technique préférée des participants. L'aspect "fun", où elle est classée en première, joue sûrement un rôle important. La métaphore des jeux utilisée dans MagStick rend l'interaction plus ludique. Enfin, Thumbspace est classée dernière, ce qui peut être dû au nombre important d'opérations demandées pour une simple tâche de pointage.

## 5. Intégration

Dans cette section, nous présentons des exemples d'intégration de nos techniques dans des applications, le but étant de montrer que les propriétés de nos techniques peuvent être exploitées pour divers cas d'utilisation.

### TapMap

TapMap est un prototype que nous avons développé qui permet à l'utilisateur de géolocaliser ses contacts sur une carte. Lorsque l'utilisateur sélectionne un contact, grâce à TapTap, l'application déclenche un mode de communication adéquat (un appel si la personne est disponible ou l'envoi d'un email sinon (FIG. 48.D)). Cette application conserve les propriétés de TapTap : les cibles sont assez petites pour éviter de gêner la visualisation de la carte mais la sélection reste rapide, précise et facile sur tout l'écran.

TapMap dispose également d'une autre spécificité : les contacts sont regroupés sous une même icône lorsque le niveau de zoom est trop petit (FIG. 48.A). Ce mécanisme a deux avantages. D'une part, il évite de surcharger graphiquement la carte si le groupe comporte de nombreux contacts. D'autre part, il permet à l'utilisateur de zoomer sur la carte sans technique d'interaction particulière. Par exemple, l'utilisateur visualise la carte de la France et voit deux groupes. Il sélectionne le groupe de Paris grâce à TapTap (FIG. 48.B), ce qui a pour effet de zoomer sur la région parisienne (FIG. 48.C). Il continue de la sorte jusqu'à arriver à un niveau de zoom lui permettant de visualiser tous les contacts séparément (FIG. 48.D).

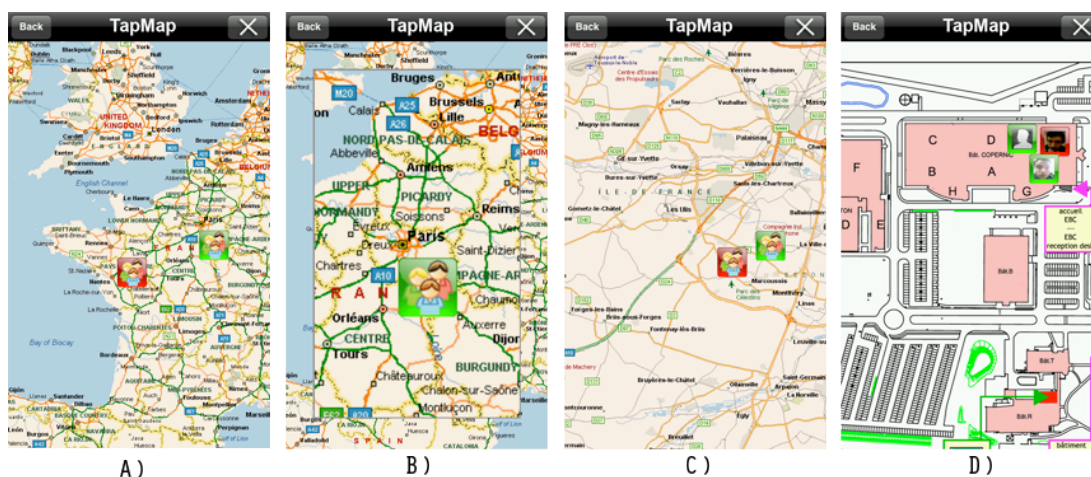


Figure 48. TapMap : A) Les contacts sont regroupés sous une même icône ; B) l'utilisateur utilise TapTap pour sélectionner un groupe ce qui a pour effet de C) zoomer sur la région parisienne. D) Lorsque l'échelle est assez large, les contacts n'apparaissent plus sous forme de groupes.

*TapWeb et ZoomTap*

Nous avons vu que TapTap ne consommait que deux évènements de l'écran tactile, ce qui permet de la combiner à des techniques impliquant un déplacement du doigt sur l'écran. Par exemple l'évènement Drag peut être détecté avant le premier tap pour permettre à l'utilisateur de faire défiler une liste ou naviguer dans une image. Il peut aussi être détecté après le premier tap (une fois la vue zoomée visible) comme dans TapWeb et ZoomTap.

TapWeb (FIG. 49) est une idée d'adaptation de TapTap pour la navigation sur le Web. L'utilisateur ouvre la vue zoomée de TapTap sur un endroit qui l'intéresse en effectuant un premier tap. Il peut ensuite déplacer le contenu de la vue zoomée de TapTap en déplaçant son doigt sur l'écran et effectuer un second tap sur un lien qui l'intéresse pour le sélectionner (ce lien étant alors de taille suffisante pour une sélection facile avec le doigt). Il peut aussi taper en dehors de la vue zoomée pour revenir à la vue globale du navigateur. Comme TapMap, TapWeb conserve les propriétés de TapTap, mais met également à profit la phase de zoom pour l'aide à la navigation sur le Web.

TapTap a aussi inspiré d'autres chercheurs. Notamment, ZoomTap (GUNN 09) est une technique dérivant de TapTap qui permet la sélection multiple : l'utilisateur effectue un premier tap pour afficher la vue zoomée. Il peut ensuite choisir des cibles sur la vue, ou faire défiler celle-ci en déplaçant son doigt sur l'écran pour atteindre des cibles non encore visibles. L'utilisateur ferme la vue en tapant en dehors ce qui a pour effet de sélectionner les cibles qu'il a choisies. Un inconvénient est que l'on perd la possibilité d'annuler qui est normalement possible lorsque l'utilisateur tape en dehors de la vue zoomée.

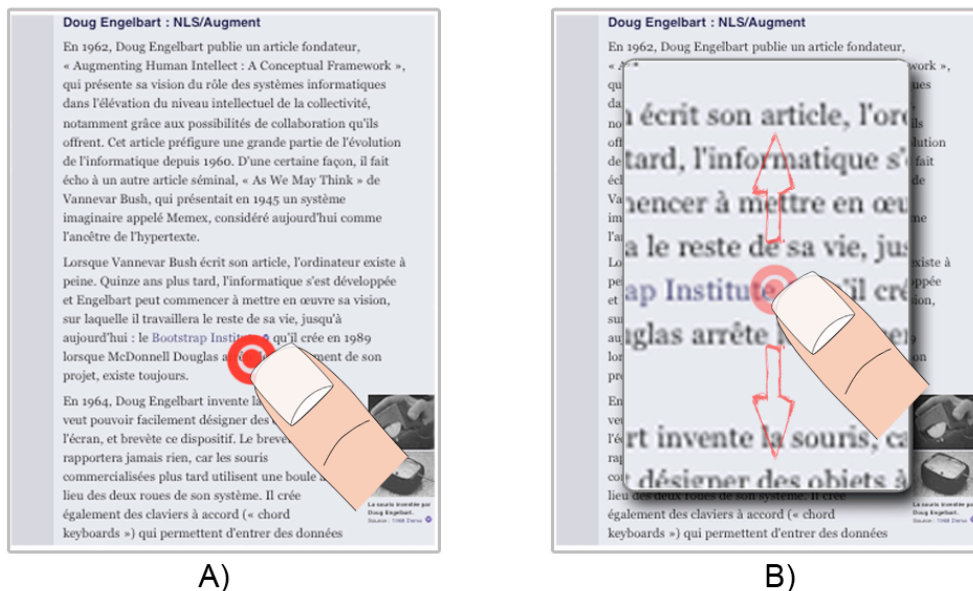


Figure 49. TapWeb permet de zoomer et de naviguer dans une page Web et elle facilite la sélection des liens (relativement petits) qui s'y trouvent.



## TapMenu

Les cibles sont souvent alignées ou groupées dans les applications réelles. C'est le cas des menus, listes, boîtes à outils ou onglets. Nous proposons TapMenu une idée d'adaptation de TapTap pour la sélection d'items dans un menu ou une liste. Cette technique est basée sur le même principe que TapTap : le menu contient des groupes d'items sur lesquels l'utilisateur peut appuyer pour les agrandir (en appuyant sur le cadre correspondant (FIG. 50.A)). Le groupe est alors élargi, les items pouvant alors être facilement sélectionnés au moyen d'un second tap (FIG. 50.B). Un second tap en dehors du menu permet de revenir à la vue non zoomée du menu. Ce menu peut accueillir environ 49 (7x7) items en considérant que la taille d'un item est de 9.2mm (pour faciliter la sélection au pouce) et que les écrans courant peuvent accueillir environ 7 items en hauteur (6.5 cm de hauteur).

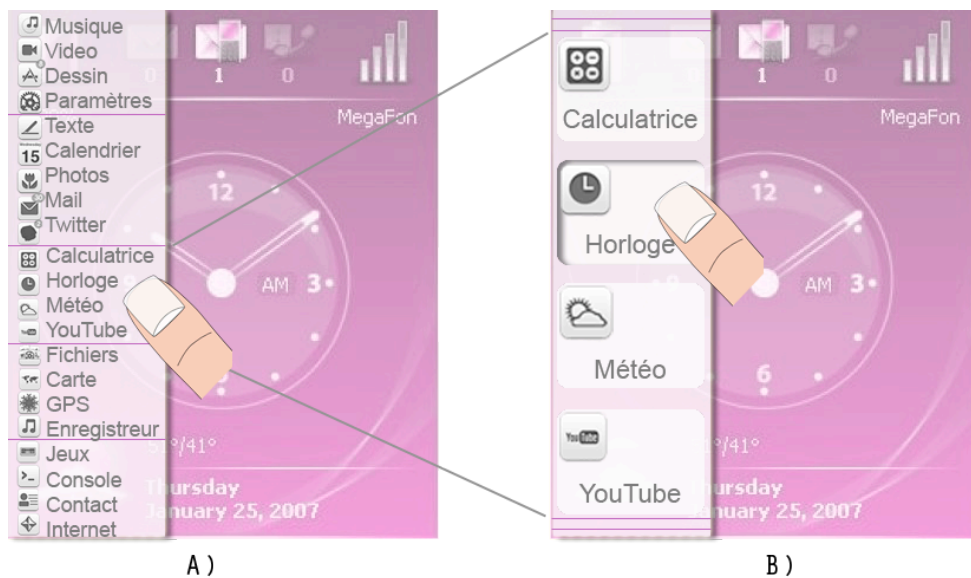


Figure 50. Le TapMenu repose sur la sélection en deux taps et permet d'afficher jusqu'à 49 items dans un menu linéaire.

## MagPaint

MagStick étant une technique à curseur décalé, elle consomme l'événement Drag de l'écran tactile, ce qui peut gêner l'activation d'autres fonctionnalités. Une solution pourrait être d'attendre un délai (ou d'appuyer plus fort sur l'écran s'il détecte la pression) une fois le curseur sur la cible pour activer un mode spécifique. Par exemple dans MagPaint, ce mode permettrait à l'utilisateur de retailler et d'incliner une forme géométrique en une seule opération (FIG. 51) : l'utilisateur place le curseur sur un coin actif de la forme et déclenche le mode. Il peut ensuite grandir la forme en éloignant son pouce du point de référence de MagStick (et inversement pour rétrécir l'objet) et l'incliner simultanément, avec un peu d'entraînement, en effectuant une rotation par rapport au même point de référence. Nous souhaitons développer cette technique et vérifier si les utilisateurs sont capables de contrôler ces deux opérations en même temps.

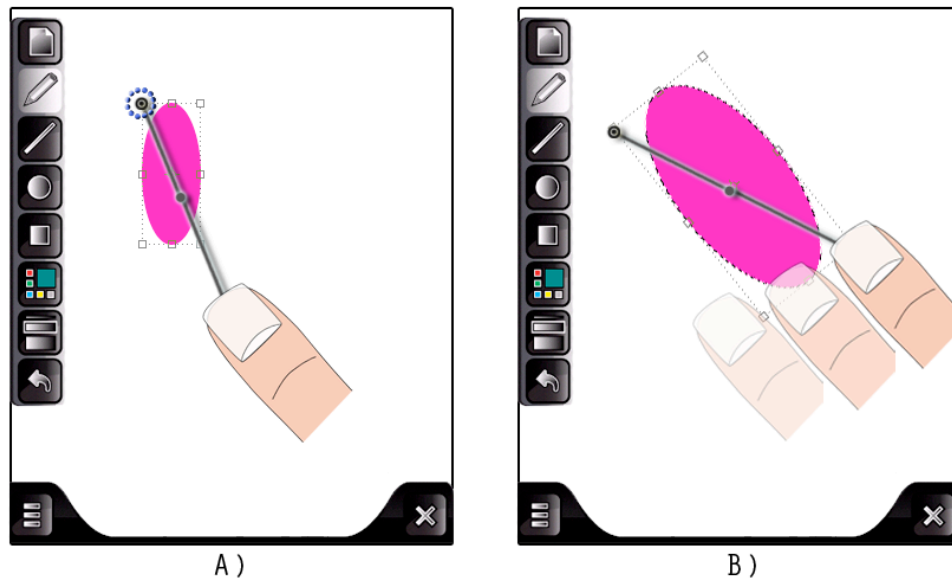


Figure 51. MagPaint : A) L'utilisateur place le curseur sur une zone active d'un objet et attend un délai. B) Il peut ensuite retailler et incliner des formes géométriques en une seule opération.

### MagMenu

Nous proposons aussi MagMenu (FIG. 52) qui est une idée de menu circulaire contenant des items relativement petits que l'utilisateur peut sélectionner au pouce en utilisant MagStick. Le design de MagStick permet en effet d'atteindre l'item d'une liste tout en gardant le pouce approximativement à la même place sur l'écran. De plus, comme le pouce est placé loin des cibles, il risque moins de cacher une partie du menu ou de la liste.



Figure 52. MagMenu est un menu circulaire basé sur l'utilisation de MagStick. Les items ne sont pas occultés par le pouce puisqu'il s'éloigne de ceux-ci au lieu de s'en rapprocher.



## MagCane

La technique Drag-and-throw (HASCOËT 03) a été proposée sur grands écrans pour faciliter le Drag & Drop : lorsque l'utilisateur veut déplacer une cible vers un dossier, il arme son arc en se déplaçant dans la direction opposée au dossier, puis relâche pour envoyer la cible dans le dossier. MagStick utilise un mécanisme assez proche de cette technique mais nous proposons une idée d'amélioration pour la sélection de cibles sur les grands écrans : MagCane vise à aider les utilisateurs (en particulier les enfants) à accéder à la partie haute du tableau (FIG. 53). MagCane ne conserve pas un bâtonnet dont les deux parties sont de tailles égales mais introduit un gain : la partie supérieure du bâtonnet valant le double (ou le triple selon la taille de l'écran) de la partie inférieure. Ce mécanisme devrait permettre de conserver la propriété de prédiction de MagStick, laquelle aide l'utilisateur à estimer la distance qu'il doit parcourir avec son doigt (ce qui reste à vérifier de façon plus approfondie).

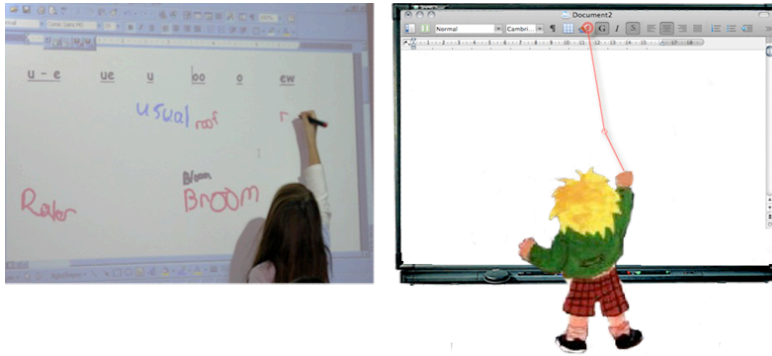


Figure 53. MagCane est une évolution de MagStick pour grands écrans qui permet d'atteindre des zones éloignées comme le haut de l'écran.

## 6. Conclusion

Nous avons présenté TapTap et MagStick, deux techniques qui améliorent le pointage de petites cibles au pouce sur écran tactile. TapTap est basée sur une stratégie de multiplexage temporel où le premier tap sert à spécifier le focus d'intérêt de l'interface et le second tap à sélectionner une cible dans la zone de focus qui a été agrandie. MagStick est basée sur l'utilisation d'un bâtonnet télescopique qui permet d'atteindre les cibles en s'éloignant d'elles et qui contrôle un curseur aimanté (ceci constituant une variante du Pointage sémantique). Les expériences ont montré que ces deux techniques sont plus rapides et produisent moins d'erreurs que les techniques de l'état de l'art<sup>8</sup>. Elles prennent également en compte les contraintes de la main (FIG. 54) en évitant l'occultation du pouce et en permettant d'accéder facilement à toutes les zones de l'écran. De plus, le design de TapTap favorise sa compatibilité avec d'autres techniques.

<sup>8</sup> Escape, Burst et Starbust n'étaient pas publiées au moment de ces travaux, mais nous avons montré dans l'état de l'art de ce chapitre que ces techniques présentent plusieurs contraintes.



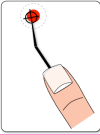

	Occultation	Imprécision	Instabilité	Evènements	Zones difficiles d'accès
 <p><b>TapTap</b></p>	Pas d'occultation	La cible est à l'échelle du pouce	Pas d'instabilité	Press + Release	
 <p><b>MagStick</b></p>	Pas d'occultation car le pouce s'éloigne de la cible	Curseur aidé par le Pointage sémantique	Filtre passe-bas et effet magnétique qui permet un geste rapide	Press + Drag + Release	

Figure 54. Tableau récapitulatif des propriétés de TapTap et MagStick face aux problèmes de l'interaction à une main.

Nous avons ensuite proposé plusieurs exemples d'intégration de nos techniques dans des applications. Ces alternatives montrent comment exploiter leurs propriétés pour divers cas d'utilisation pratique.

Pour finir, nous avons également observé, grâce à l'évaluation contrôlée et aux évènements auxquels nous avons participé (conférences et démonstrations) que TapTap a suscité un certain engouement, lequel peut être dû à sa simplicité d'utilisation. Nous avons notamment présenté TapMap aux journées portes ouvertes de Bell Labs Alcatel-lucent. Concernant MagStick, nous avons également eu des retours positifs, entre autres de personnes travaillant sur grands écrans qui nous ont suggéré l'adaptation de MagStick pour ce type de plate-forme.

## Pointage sur écrans tactiles **3**

# Menus sur écrans tactiles



"Plus de 2 milliards d'applications ont été vendues sur l'AppStore depuis son lancement"

Steve Jobs dans une interview au Apple's headquarters (2009)

<b>1. Introduction</b>	<b>95</b>
<b>2. Etat de l'art</b>	<b>96</b>
2.1 Menus à base de gestes déictiques	96
2.2 Menus linéaires gestuels	97
2.3 Menus circulaires gestuels	99
2.4 Synthèse de l'état de l'art	103
<b>3. LeafMenu</b>	<b>105</b>
3.1 Présentation	105
3.2 Propriétés	107
3.3 Implémentation	110
<b>4. Expérimentation</b>	<b>111</b>
4.1 Plan d'expérimentation	111
4.2 Résultats et discussion	113
<b>5. Intégration</b>	<b>116</b>
<b>6. Conclusion</b>	<b>118</b>

## 1. Introduction

Dans le chapitre précédent, nous avons présenté des techniques de pointage sur écrans tactiles. Nous nous intéressons ici à un cas particulier du pointage qui est la sélection d'items dans un menu. Au même titre que les techniques de sélection de cibles, les menus sont en effet des éléments incontournables dans les systèmes interactifs car ils permettent de manipuler les nombreuses fonctionnalités offertes par ces systèmes.

Comme nous avons vu au [CHAPITRE 2](#), la technique de menu la plus répandue est le menu linéaire qui s'ouvre généralement à partir d'une barre de menus placée en bas ou en haut de l'écran. Les items sont souvent plus larges que sur les ordinateurs conventionnels pour faciliter la sélection aux doigts, ce qui augmente la hauteur du menu et diminue le nombre d'items possible. L'interface de l'iPhone ne propose pas de barres de menus mais plutôt des barres de navigation (d'outils ou d'onglets) contenant des boutons pour accéder aux commandes. Le problème reste cependant le même puisque les boutons doivent être assez larges pour la sélection au pouce, ce qui diminue le nombre de commandes que l'on peut afficher.

Certains systèmes proposent également des menus contextuels qui se déclenchent après un appui long sur l'écran (Windows Mobile, Android ou Symbian). Comme les menus sont décalés pour éviter de sortir de l'écran, ils sont souvent cachés par le pouce. De plus, les items subissent un décalage pour éviter de sortir de l'écran lorsque le menu est activé en bas de l'écran. Initialement absents de l'iPhone, les menus contextuels ont été introduits récemment pour permettre le copier/coller. Ces menus très spécifiques ne contiennent donc qu'un nombre limité d'items (copier, coller, couper et sélectionner). Ils sont déclenchés au relâchement du doigt ou après un double tap et s'affichent toujours au dessus du doigt pour éviter l'occultation. Enfin, un autre problème commun à tous les menus sur mobiles est l'absence d'un mode expert facilement utilisable. Les raccourcis claviers sont en effet basés sur l'usage de modificateurs, lesquels sont soit absents (pas de clavier physique), soit difficilement utilisables à une main.

Une question se pose alors : faut-il renoncer aux menus ou réfléchir à des adaptations plus adéquates aux dispositifs mobiles ? Dans ce chapitre, nous nous intéressons à la deuxième solution. Nous analysons tout d'abord l'état de l'art et nous étudions de quelle façon les techniques de menu existantes répondent aux contraintes des dispositifs mobiles établies dans le [CHAPITRE 2](#). Puis nous proposons une nouvelle technique de menu, le LeafMenu ([BAILLY, ROUDAUT 08](#)) ([ROUDAUT, BAILLY 09](#)), dont nous testons les propriétés grâce une expérimentation contrôlée. Avant de conclure, nous présentons des alternatives ou des exemples d'intégration du LeafMenu dans des applications. Ce travail a été réalisé en collaboration avec Gilles Bailly qui présente également le LeafMenu dans ses travaux de thèse ([BAILLY 09](#)).

## 2. Etat de l'art

Plusieurs espaces de conception facilitent la création de nouvelles formes de menus, comme MenUA (BAILLY 09) ou l'espace morphologique de Nancel et al. (NANCEL 09). Nous nous basons sur ces travaux pour présenter les solutions existantes que nous organisons en trois catégories : les techniques à base de gestes déictiques, les techniques de menus linéaires à base de gestes sémaphoriques et les techniques de menus circulaires à base de sémaphoriques.

### 2.1 Menus à base de gestes déictiques

Le ThumbMenu et l'ArchMenu (HUOT 07) sont des menus semi-circulaires qui évitent l'occultation par le pouce. Ils sont basés sur l'utilisation d'une technique de pointage vue au CHAPITRE 3 : Adaptative Offset Cursor (HUOT 06) qui consiste à appuyer sur l'écran, puis à déplacer le curseur sur un item et relâcher le doigt pour sélectionner.

#### ThumbMenu

Le ThumbMenu (HUOT 07) (FIG. 55) est un menu circulaire dont un quart de cercle est supprimé pour éviter l'occultation par le pouce. Cette idée est également utilisée sur les larges tables tactiles (BRANDL 09) pour éviter l'occultation par la main. Comme avec le menu linéaire classique, l'utilisateur peut sélectionner un item pour activer des fonctionnalités ou visualiser un sous menu. Le ThumbMenu repose sur un mécanisme de basculement afin que le sous-menu soit toujours dans l'axe du pouce : si l'utilisateur ouvre un sous-menu sur la partie haute du menu (FIG. 55.A) puis replace son doigt au centre du menu, l'ensemble du menu pivote afin de placer le sous-menu dans le prolongement du pouce (FIG. 55.B). Ce mécanisme évite l'occultation des items puisque ceux-ci sont toujours en face du pouce. Ce menu n'offre pas de mode expert et ne permet pas les sélections près des bords de l'écran. Le premier niveau de la hiérarchie risque en effet d'apparaître en dehors de l'écran si le menu est déclenché trop près d'un bord, ce qui empêche l'accès à certains items.

#### ArchMenu

L'ArchMenu (HUOT 07) (FIG. 56.A) est un menu en quart de cercle placé dans un coin inférieur de l'écran (droit ou gauche selon les préférences de l'utilisateur). Il s'active en cliquant dans cette zone spécifique. L'ArchMenu est hiérarchique, un quart de cercle ne permettant pas d'afficher un grand nombre de commandes. Les sous-menus se déploient en éventail lorsque le curseur est au dessus d'un item (FIG. 56.B). Cette organisation spatiale évite de consommer trop d'espace en largeur. Enfin il n'existe pas de version contextuelle de l'ArchMenu et, comme le ThumbMenu, il n'y a pas de mode expert.

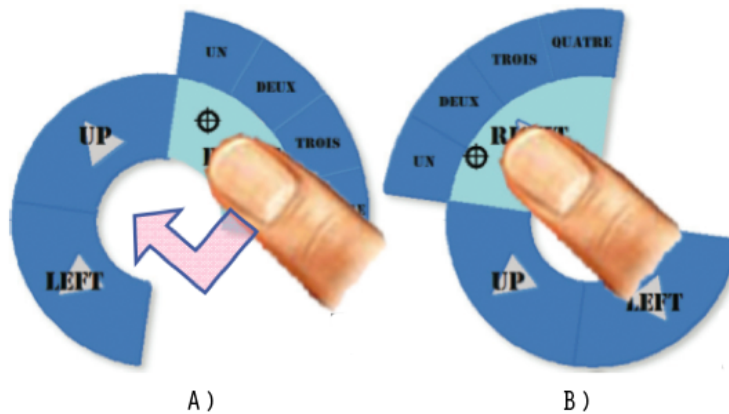


Figure 55. ThumbMenu : A) l'utilisateur ouvre un sous-menu dans la partie supérieure du menu et peut déplacer son doigt au centre pour faire basculer le menu ; B) le sous-menu est alors affiché dans le prolongement du pouce.

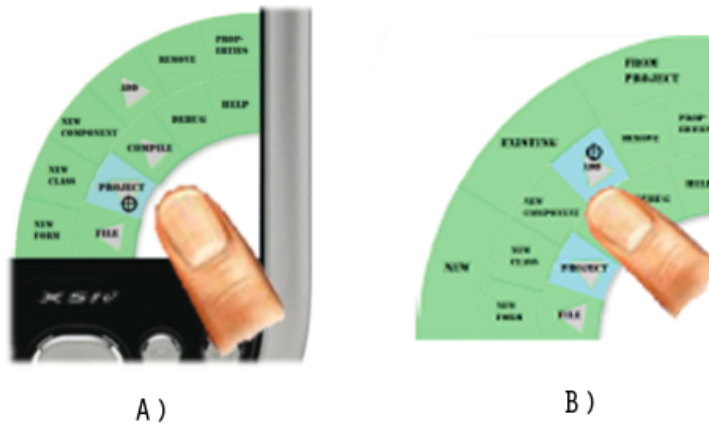


Figure 56. ArchMenu : A) le menu s'affiche dans un coin ; B) les sous-menus se déploient en éventail lorsque le curseur est au dessus d'un item.

## 2.2 Menus linéaires gestuels

Kurtenbach et Appert (KURTENBACH 93) (APPERT 09) (FIG. 57) ont proposé une amélioration du menu linéaire qui consiste à associer un geste à une commande. Les gestes sont indépendants à l'échelle, c'est à dire que les utilisateurs peuvent indifféremment effectuer de petits comme de larges gestes. Ces deux techniques offrent ainsi la possibilité d'activer le menu en mode expert même en l'absence de clavier. Dans (APPERT 09), les auteurs montrent que les raccourcis gestuels sont aussi efficaces que les raccourcis claviers, mais sont plus facilement appris et mémorisés par les utilisateurs. Pour un même niveau d'entraînement, les utilisateurs pouvaient en effet apprendre trois fois plus de raccourcis gestuels que de raccourcis claviers. Même si cette étude a été effectuée sur des ordinateurs conventionnels, les auteurs encouragent l'utilisation de gestes sur les dispositifs ne possédant pas de clavier.



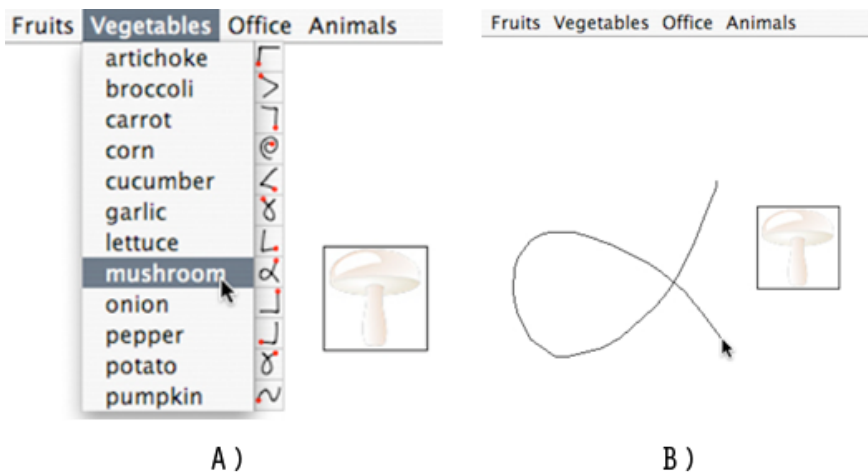


Figure 57. Menu linéaire avec raccourcis gestuels : A) mode novice ; B) mode expert.

Un avantage de ces techniques est de conserver la forme habituelle du menu linéaire. Celle-ci est en effet plus compacte que celle d'un menu circulaire, comme ceux que nous avons décrit précédemment (un menu linéaire consomme environ deux fois moins de place en largeur). De plus, lorsque le menu est contextuel, la cible n'est que partiellement cachée par le menu contrairement aux menus circulaires qui occultent toute la cible (FIG. 58.A ET 58.B).

Toutefois, le fait de conserver la forme du menu linéaire entraîne certains problèmes. D'une part la sélection d'items en mode novice repose sur des gestes déictiques (l'utilisateur doit déplacer son pouce jusqu'à l'item, une opération qui manque de précision si les items sont petits). D'autre part, l'affichage du menu par défaut en bas à droite du curseur entraîne son occultation par le pouce et par une partie de la main, ce qui n'est pas le cas quand le menu s'affiche sur la gauche du curseur (pour un droitier). Lorsque l'utilisateur appuie en bas de l'écran, le menu est décalé vers le haut pour ne pas sortir de l'écran, ce qui peut nécessiter un temps supplémentaire à l'utilisateur pour se repérer dans le menu. Enfin, le mode expert est difficilement utilisable dans certaines conditions : lorsque l'utilisateur active un menu contextuel dans un coin de l'écran, il ne peut pas faire un geste allant plus loin que les bords.

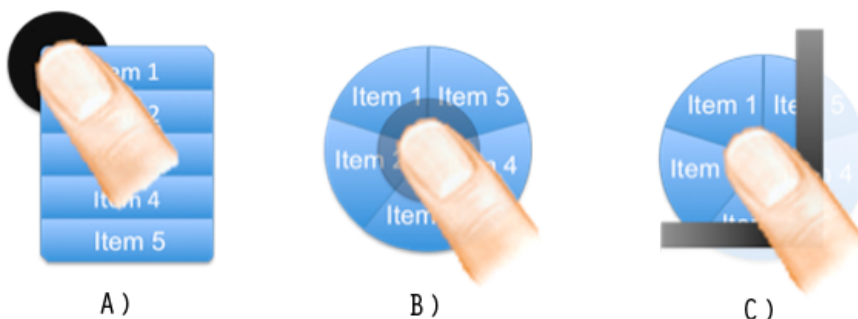


Figure 58. A) le menu linéaire n'occulte qu'une partie de la cible ; B) le menu circulaire cache toute la cible ; C) le menu circulaire peut sortir des bords de l'écran.

## 2.3 Menus circulaires gestuels

Les menus circulaires gestuels ont la particularité de permettre une interaction gestuelle dans les deux modes de fonctionnement du menu (novice et expert). Ils évitent ainsi l'utilisation de gestes de pointage qui sont souvent imprécis à cause de l'interaction aux doigts. Nous présentons les principales techniques existantes ci-après.

### Marking Menu

Dans le mode novice des Marking Menus (KURTENBACH 93), l'utilisateur appuie sur l'écran et attend 300ms que le menu apparaisse. Il fait ensuite une marque dans la direction de l'item désiré (FIG. 59.A). Dans le mode expert, il n'a pas besoin d'attendre de délai et effectue directement un geste pour sélectionner une commande (FIG. 59.B). L'avantage principal du Marking Menu est qu'il favorise le passage entre le mode novice et expert puisque l'utilisateur exécute le même geste dans les deux modes. En fait, l'utilisateur utilise le mode expert sans s'en rendre compte car le passage d'un mode à l'autre est implicite et basé sur la répétition. De plus, contrairement aux menus linéaires, tous les items sont à la même distance du point d'activation, ce qui permet de tous les sélectionner à la même vitesse.

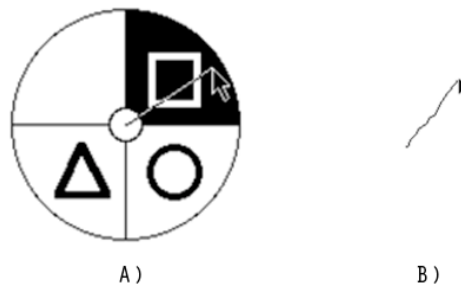


Figure 59. Marking menu : A) mode novice ; B) mode expert.

Les inconvénients du Marking Menu proviennent principalement de sa forme circulaire. Le pouce occulte en effet un quart du menu (la partie inférieure droite pour un droitier et inversement pour un gaucher). Comme expliqué précédemment, lorsqu'il est contextuel, le menu risque de cacher la cible entièrement (FIG. 58.B). De plus, quand le menu est activé près des bords, certains gestes ne peuvent être exécutés, par exemple un geste vers le haut sur une cible se trouvant près du bord supérieur. Dans ce même cas, une partie du menu peut sortir de l'écran ce qui gêne le mode novice (FIG. 58.C). Une possibilité est d'utiliser un mécanisme de débrayage qui décale le menu s'il sort de l'écran. Dans ce cas, le point d'activation n'est cependant plus au centre du menu.

Enfin, la sélection des items peut poser problème dans plusieurs cas. Certains gestes diagonaux sont par exemple difficiles à effectuer dans l'axe du pouce (Sud-Est vers Nord-Ouest pour un droitier et Nord-Est vers Sud-Ouest pour un gaucher) (KARLSON 05). Le nombre d'items est

aussi limité par la taille des secteurs angulaires qui doivent être assez larges pour éviter les erreurs. Zhao et Al. (ZHAO 06) montrent qu'il est préférable de ne pas utiliser plus de 8 items dans un menu circulaire sur PCs. On peut craindre que ce nombre diminue lorsque l'on interagit avec le pouce. Plusieurs solutions ont été proposées pour augmenter le nombre d'items des Marking Menus, nous les détaillons ci-après.

### Compound Marking Menus

Le Compound Marking Menu (KURTENBACH, BUXTON 93), est un Marking Menu hiérarchique qui permet d'augmenter le nombre d'items : lorsque l'utilisateur fait un geste pour choisir un item, un sous-menu apparaît à l'extrémité de l'item (FIG. 60.A). Les sous-menus se déploient donc autour des menus, ce qui permet à l'utilisateur de prévisualiser la hiérarchie des menus. En mode novice et expert, l'utilisateur peut combiner les différents gestes directionnels en un seul geste continu pour accéder à l'item d'un sous-menu (FIG. 60.B).

D'après plusieurs travaux (ZHAO 04) (BAILLY 09), cette approche a cependant des inconvénients : les Compound Marking Menus sont relativement imprécis en mode expert (KURTENBACH 93) (ZHAO 04) lorsque le nombre de commandes est important, car les utilisateurs ont des difficultés à composer spatialement plusieurs marques (en particulier sur les diagonales). De plus, certains gestes sont ambigus en mode expert (ZHAO 04) car, les marques étant indépendantes à l'échelle, une trace peut parfois correspondre à plusieurs commandes (par exemple deux commandes verticales à la suite). Ceci a pour effet de diminuer le nombre de commandes disponibles. Enfin, ces menus nécessitent beaucoup de place à l'écran en mode novice (ZHAO 04) car les sous-menus sont affichés dans la direction de l'item parent. Un Compound Marking Menu de profondeur 3 nécessite ainsi près de 4 fois plus d'espace en largeur qu'un menu linéaire.

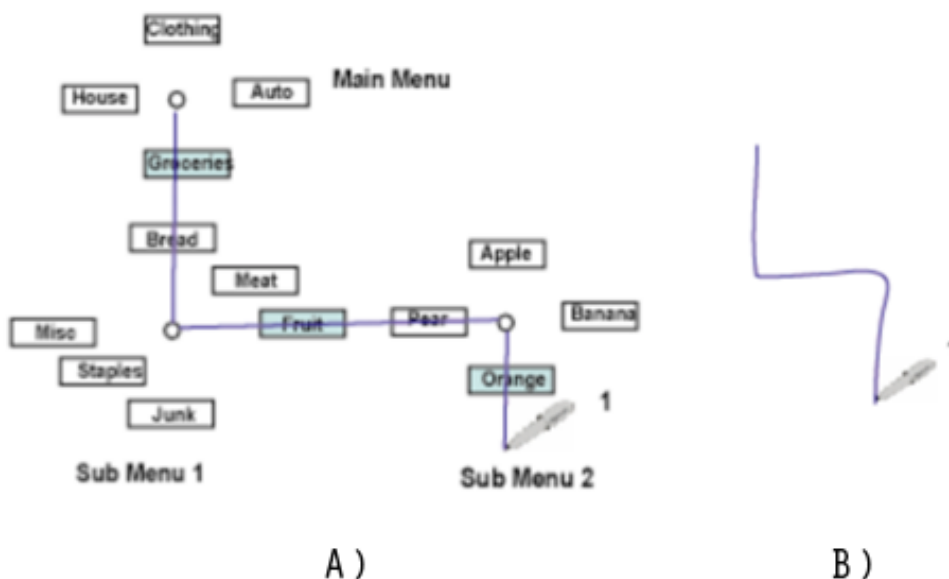


Figure 60. Compound Marking Menu : A) mode novice ; B) mode expert.

### Multi-Stroke Menus

Le Multi-Stroke Menu (ZHAO 04), illustré dans T-Cube (VENOLIA 94) pour entrer du texte (FIG. 61), est aussi un Marking Menu hiérarchique. Néanmoins, son fonctionnement est différent du Compound Marking Menu : lorsque l'utilisateur finit son premier geste directionnel et relâche le stylet de l'écran, le sous-menu apparaît au dessus du premier menu et l'utilisateur fait alors un second geste pour sélectionner un item. Une caractéristique du Multi-Stroke Menu est que les gestes sont organisés temporellement au lieu de spatialement comme avec le Compound Marking Menu. Les sous-menus étant superposés, ils consomment ainsi moins de place à l'écran. Cependant, comme le montrent les travaux de Bailly (BAILLY 08), cette disposition spatiale dégrade le mode novice car elle ne permet pas à l'utilisateur de prévisualiser les sous-menus ni d'observer le chemin parcouru dans la hiérarchie.

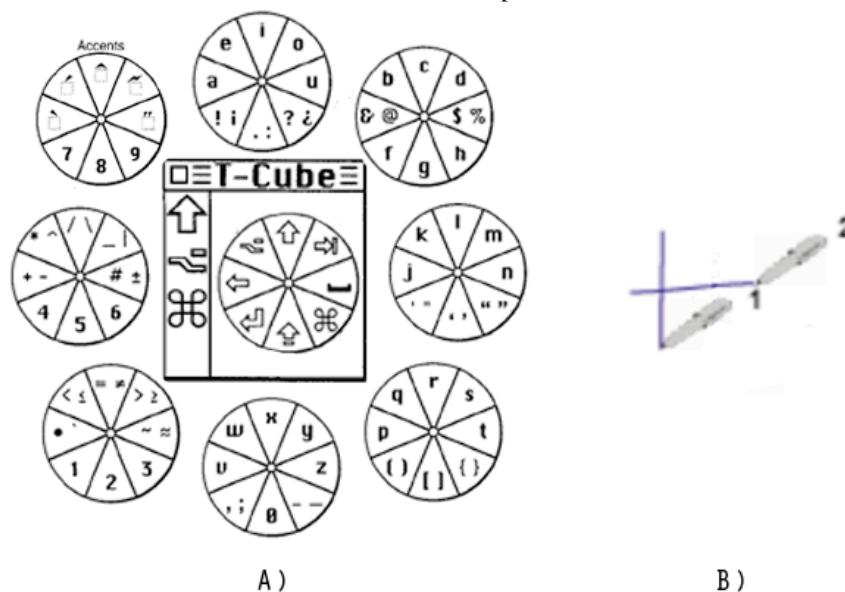


Figure 61. Multi-Stroke Menu illustré par la technique d'entrée de texte T-Cube : A) les différents menus se superposent ; B) en mode expert l'utilisateur effectue des marques temporelles.

Le WaveMenu (BAILLY 07), et son équivalent sur mobile le Wavelet (FRANCONI 09), est un Multi-Stroke Menu qui optimise l'occupation spatiale pour permettre de mieux prévisualiser les items et faciliter la navigation dans le menu. Lorsque l'utilisateur fait un geste dans la direction d'un item, le menu s'élargit comme si le pouce poussait le premier niveau de la hiérarchie vers l'extérieur, ceci faisant apparaître le sous-menu à la place du menu principal (FIG. 62). Toutes les hiérarchies du menu sont visibles, les sous-menus prenant la forme d'anneaux concentriques (le premier niveau du menu se trouvant dans l'anneau le plus éloigné). Sur un petit écran, les informations sortant de l'écran sont les anneaux extérieurs du menu, qui correspondent aux premiers niveaux de la hiérarchie du menu. Ce design permet d'interagir même si l'écran est petit : l'information manquante n'est pas indispensable, ce menu hiérarchique pouvant toujours être manipulé à partir de sa zone centrale.



Figure 62. Wavelet Menu.

### Marking Menus enrichis

Une solution pour augmenter le nombre d'items des Marking Menus est d'enrichir le vocabulaire gestuel utilisé. Les Marking Menus classiques nécessitent l'utilisation de deux coordonnées pour reconnaître le geste : le point de départ et le point d'arrivée. Dans QuikWriting (ISOKOSKI 04), la longueur du geste compte : pour sélectionner la lettre W, l'utilisateur fait un geste du centre au coin inférieur gauche, puis repart verticalement pour aller jusqu'à la case du centre (FIG. 63.A). S'il dépasse la case, le V sera sélectionné. Cependant, en utilisant la longueur du geste on perd la propriété d'indépendance à l'échelle. L'utilisateur doit en effet effectuer un geste de longueur donnée, ce qui peut être problématique si les cases sont petites.

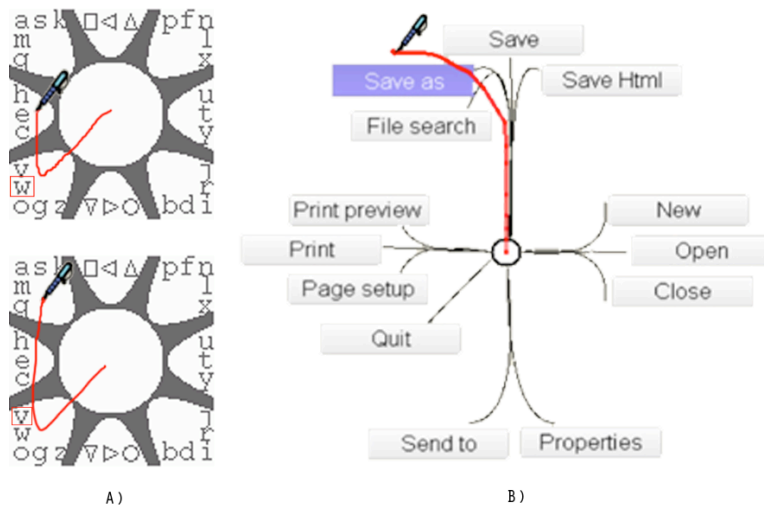


Figure 63. A) QuikWriting ; B) Flower menu.

Une autre solution consiste à utiliser la courbure comme dans le Flower Menu (BAILLY 08) (FIG. 63.B), qui permet de sélectionner jusqu'à 56 commandes et de faire des groupements d'items comme on en trouve dans les menus linéaires standard. Le PushMenu (HUOT 09) augmente le nombre d'items en combinant les gestes d'un Marking Menu avec différents niveaux de pression du stylet sur une tablette tactile. Enfin l'inclinaison d'un stylet est utilisée d'une manière similaire dans le TiltMenu (TIAN, 08). D'autres menus utilisent ou combinent davantage de dimension comme dans les travaux de (NANCEL 09).

Cependant, si ces techniques augmentent le nombre d'items possible, elles héritent des mêmes contraintes que les menus circulaires : une occupation spatiale importante, l'occultation du pouce sur le menu et les cibles et des gestes parfois difficiles à effectuer (dans l'axe du pouce ou près des bords de l'écran).

## 2.4 Synthèse de l'état de l'art

La **FIGURE 64** résume les propriétés des techniques de menu que nous avons exposées. Comme le montre ce tableau et le résumé qui suit, ces techniques ne résolvent pas tous les problèmes de l'interaction à une main sur un petit écran.

	Occupation spatiale	Occultation	Imprécision	Zones difficiles	Manque de clavier
<b>ThumbMenu</b>	Largeur écran, sort de l'écran si contextuel	Pas d'occultation du menu / menu occulte la cible	Offset Cursor (précis et lent)	Pas de zone plus difficiles	Pas de mode expert
<b>ArchMenu</b>	½ écran dans un coin			Activation près de la base du pouce (gênant pour la prise en main)	
<b>Linéaires gestuels</b>	½ écran mais peut sortir en bas écran	Partie du menu occulté par pouce / menu occulte ¼ cible	Du mode novice si items petits	Gestes impossibles près des bords (contextuel)	Mode expert gestuel
<b>Compound Marking Menu</b>	Très large, sort écran si contextuel	Partie du menu occulté par pouce / menu occulte la cible	Si plus de 8 items	Gestes diagonaux dans axe pouce et près des bords (contextuel)	Mode expert gestuel similaire au mode novice
<b>Multi-Stroke Menus</b>	Largeur écran, sort écran si contextuel				
<b>Wavelet</b>	Circulaire, items cachés non vitaux				
<b>QuikWriting</b>	1/2 écran environ (car items = lettres)				
<b>FlowerMenu</b>	Largeur écran, sort écran si contextuel		Geste dépendant de l'échelle		
			Précis car courbure et sens		

Figure 64. Tableau récapitulatif des techniques de menus.

Le menu linéaire, une technique très répandue basée sur des gestes déictiques, souffre d'une précision insuffisante sur dispositifs mobiles. Comme nous l'avons montré dans le **CHAPITRE 3**, le pointage de petites cibles est imprécis lors de l'interaction au doigt et nécessite l'utilisation d'une technique de pointage adaptée. Nous avons notamment présenté le ThumbMenu et l'ArchMenu (**HUOT 07**) qui reposent sur le mécanisme d'Adaptative Offset

Cursor (HUOT 07), détaillé au CHAPITRE 3, qui améliore la précision mais est relativement lent. De plus, ces techniques ne possèdent pas de mode expert et le ThumbMenu peut sortir de l'écran, ce qui empêche l'accès à certains items.

Les gestes sémaphoriques constituent une alternative au pointage. Ils sont réalisés plus rapidement que les gestes de pointage et permettent le fonctionnement des menus en mode expert. Plusieurs techniques ont exploité cette propriété : les menus linéaires gestuels qui n'utilisent les gestes qu'en mode expert et les menus circulaires gestuels qui utilisent les gestes dans les deux modes. Ces deux types de menu ont un inconvénient commun : certains gestes du mode expert (ceux en direction des bords) ne peuvent être effectués si le menu est activé près des bords. Ensuite, chaque menu a des avantages et inconvénients spécifiques :

- Les menus linéaires ont une occupation spatiale moins grande que les menus circulaires. En effet, la largeur d'un menu circulaire correspond à peu près à la largeur de deux menus linéaires.
- Les menus linéaires ne sont pas centrés (le point d'activation peut indifféremment être l'un des quatre coins du menu, voir même un autre point), ce qui permet de les décaler lorsqu'ils apparaissent près des bords de l'écran. Par exemple, si l'utilisateur active le menu en bas de l'écran, il s'affichera en haut du point d'activation. Ce mécanisme est également possible avec les menus circulaires, mais dans ce cas, le doigt n'est plus au centre du menu et la correspondance entre la direction des gestes et la position des items est perdue.
- Les menus circulaires cachent la cible sur laquelle ils sont activés alors que les menus linéaires n'occulent environ qu'un quart de la cible.
- Les menus circulaires ne peuvent avoir qu'un nombre d'items limité (environ 8). En comparaison un menu linéaire peut contenir au moins 2 fois plus d'éléments sur un même niveau de hiérarchie. Pour augmenter le nombre d'items, plusieurs stratégies sont possibles : les menus hiérarchiques comme le Compound Marking Menu (KURTENBACH 93) et le Multi-Stroke Menu (ZHAO 04). Une autre solution est de combiner la direction de gestes à d'autres dimensions comme la courbure dans le Flower Menu (BAILLY 08), la distance dans QuickWriting (ISOKOSKI 04), la pression dans le PushMenu (HUOT 09), ou l'inclinaison d'un stylet dans le TiltMenu (TIAN, 08).
- Dans les menus linéaires proposés par (KURTENBACH 93) (APPERT 09), le mode novice et le mode expert sont différents, alors que les Marking Menus proposent un mode novice et expert qui reposent sur les mêmes gestes, ce qui facilite leur apprentissage et leur mémorisation.

### 3. LeafMenu

Partant des problèmes posés par l'utilisation des principales techniques de menu existantes, nous avons conçu une nouvelle technique de menu adaptée aux contraintes de l'interaction aux doigts sur petits écrans : le LeafMenu. Nous détaillons son fonctionnement avant de présenter ses propriétés et son implémentation.

#### 3.1 Présentation

Le LeafMenu dispose de deux modes de fonctionnement : l'interaction en mode novice est similaire à celles des menus linéaires usuels et l'interaction en mode expert repose sur des raccourcis gestuels. Chacun des deux modes présente des avantages pour l'interaction au pouce que nous détaillons ci-après.

##### *Mode novice*

Le mode novice se déclenche après un appui long sur l'écran (300ms). L'utilisateur déplace ensuite son doigt sur les items du menu et relâche l'écran pour sélectionner un item (FIG. 65.A.B.C). Si le menu est activé près d'un bord de l'écran, il se décale verticalement et/ou horizontalement pour ne pas sortir de l'écran (FIG. 66). L'originalité du LeafMenu réside donc dans son *effet miroir*. L'aspect visuel du LeafMenu subit en effet une symétrie horizontale selon sa position (à droite ou à gauche du pouce). L'effet miroir s'applique également lorsque l'espace sous le point d'activation n'est pas suffisant pour afficher le menu. Dans ce cas, le menu s'affiche au dessus du point d'activation et l'ordre des items est inversé par symétrie verticale. Ainsi l'item 1 est toujours l'item le plus proche du point d'activation et l'item 7 le plus éloigné.

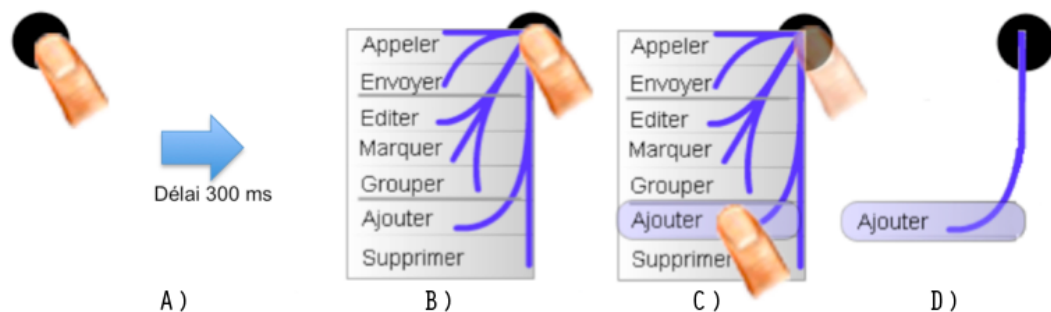


Figure 65. Mode novice : A) l'utilisateur presse une cible avec son pouce ; B) après un délai de 300ms, le menu apparaît ; C) l'utilisateur relâche son doigt sur l'item désiré pour l'activer ; D) un retour visuel confirme l'item sélectionné.



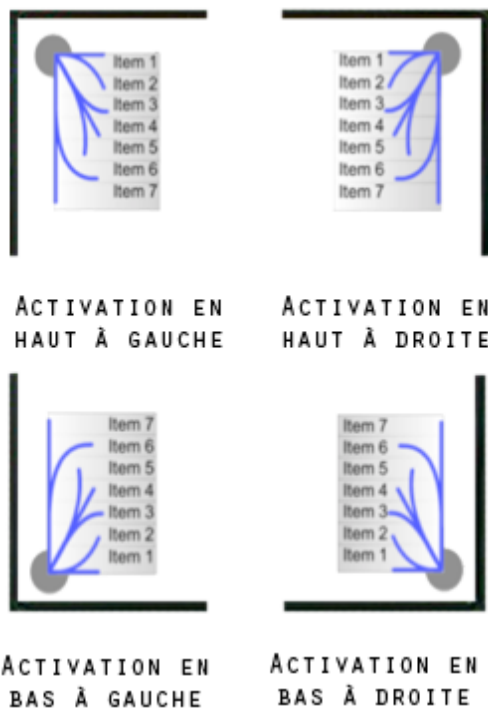


Figure 66. Grâce à l'effet miroir du LeafMenu, la distance d'un item au point d'activation est invariante comme le montrent ces quatre cas d'activation du menu.

### Mode expert gestuel

Le LeafMenu possède un mode expert basé sur des raccourcis gestuels qui sont dessinés sur le menu (FIG. 66). Dans ce mode, l'utilisateur n'attend pas le délai d'affichage du menu et effectue directement un geste pour activer un item. Les gestes à effectuer sont inspirés de ceux du Flower Menu (BAILLY 08). Ils sont au nombre de sept, dont trois gestes droits et quatre gestes courbes (FIG. 67.A). Ces sept gestes peuvent ensuite être effectués dans une des quatre directions cardinales (FIG. 67.B). On obtient ainsi 24 gestes possibles (FIG. 67.C).

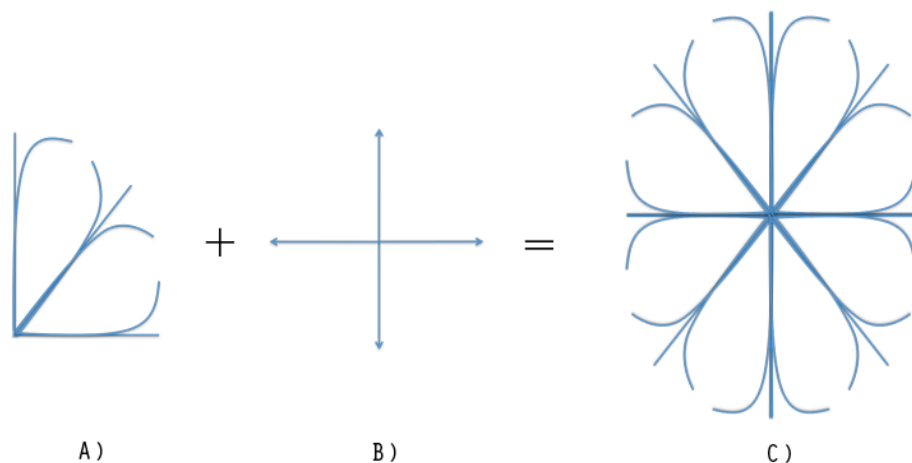


Figure 67. A) 3 gestes droits et 4 gestes courbés combinés à B) 4 directions cardinales créent un ensemble de C) 24 gestes possibles.

L'effet miroir s'applique également aux gestes : les sept gestes effectués dans une direction cardinale sont *redondants* avec ceux correspondant aux trois autres directions. Chaque item est donc associé à quatre raccourcis gestuels (sauf pour les items 1 et 7 pour lesquels deux des quatre gestes sont confondus). De ce fait, l'interaction en mode expert devient possible près des bords de l'écran, contrairement aux techniques de menus présentées dans l'état de l'art. Par exemple, si l'utilisateur souhaite activer l'item 4 près d'un bord, il y a toujours une direction possible (FIG 68) (dans ce cas la direction sud-est).

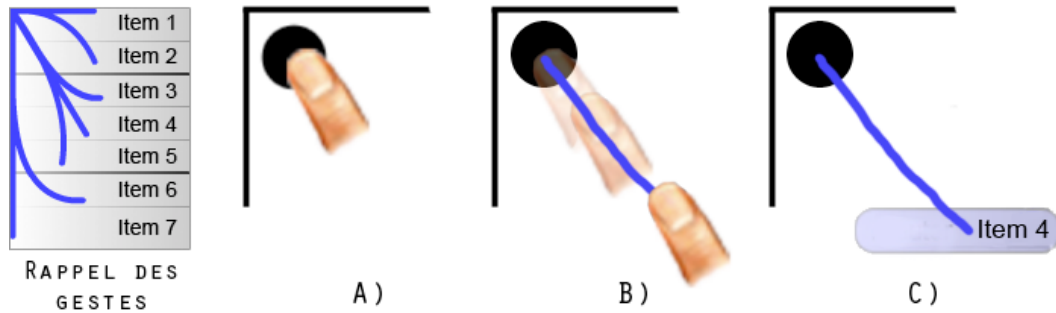


Figure 68. Mode expert : A) l'utilisateur presse une cible puis B) dessine directement la marque associée à l'item désirée. C) A la fin de son geste, un retour visuel apparaît pour confirmer l'item sélectionné.

### 3.2 Propriétés

Dans cette section nous montrons pourquoi les propriétés du LeafMenu offrent des solutions intéressantes aux problèmes de l'interaction au pouce. Nous détaillons aussi les avantages de cette technique en ce qui concerne l'apprentissage et la mémorisation des gestes. Enfin, nous aborderons la question de la compatibilité de son mode expert avec d'autres techniques.

#### *Précision/rapidité*

Les items fréquents (par exemple ceux pour couper, copier ou coller) des menus linéaires sont généralement situés en haut du menu, près du point d'activation, afin de pouvoir les sélectionner plus rapidement (FIG. 69.A). Mais lorsqu'un menu linéaire est affiché trop près du bas de l'écran, il est translaté vers le haut pour que les items restent visibles (FIG. 69.B). Les items fréquents sont alors éloignés du point d'activation et l'utilisateur doit parcourir une plus grande distance pour les sélectionner. Au contraire, l'effet miroir du LeafMenu permet de placer les items fréquents toujours près du point d'activation (FIG. 69.C), ce qui réduit le temps nécessaire pour les sélectionner. Par exemple si l'item 1 correspond à une commande fréquente, l'utilisateur n'aura pas à le chercher dans le menu puisque c'est toujours l'item le plus proche de son doigt.

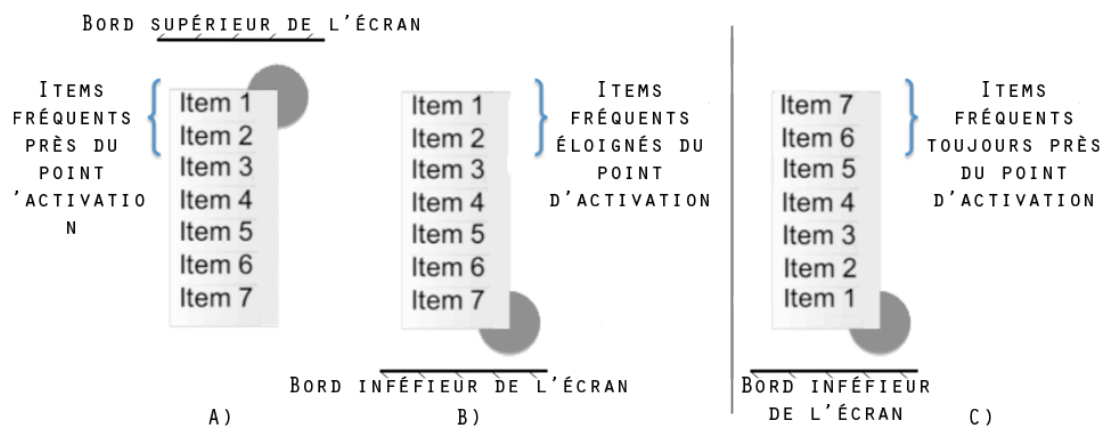


Figure 69. Menu linéaire standard : A) si le point d'activation est près du bord supérieur de l'écran, le menu s'affiche en dessous; B) si le point d'activation est près du bord inférieur de l'écran, le menu est traduit vers le haut mais l'ordre des items reste inchangé ; C) avec le LeafMenu, l'ordre des items est inversé de façon à ce que l'item 1 soit toujours le plus près du point d'activation.

Une autre propriété du LeafMenu qui améliore la précision est qu'un filtre passe-bas est utilisé pour stabiliser le mouvement du pouce et éviter les relâchements accidentels. Enfin, rappelons que les gestes du mode expert ne nécessitent pas d'opérations de pointage, ce qui permet de les effectuer rapidement.

### Occultation

Le LeafMenu peut s'afficher aussi bien à droite ou à gauche, en bas ou en haut du point d'activation pour éviter d'apparaître en dehors de l'écran. Ceci permet aussi de minimiser l'occultation par le doigt dans la partie droite de l'écran. Lorsque le menu est affiché dans la partie gauche de l'écran, le pouce peut cacher une partie du menu (FIG. 70.A), mais lorsque l'utilisateur déplace son pouce, les items du menu redeviennent visibles (FIG. 70.B).

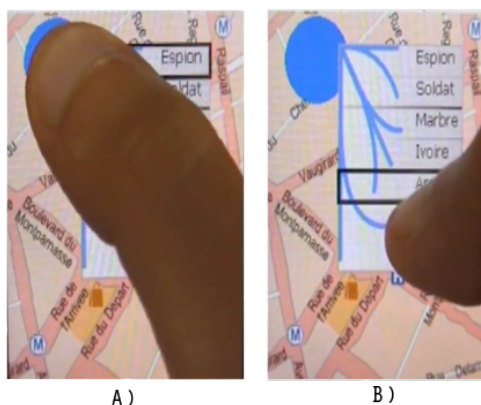


Figure 70. A) le pouce peut occulter le menu dans la partie gauche de l'écran; B) dès que l'utilisateur déplace son pouce, il peut voir de nouveau le menu.

*Zones difficiles d'accès*

Contrairement aux techniques existantes, les gestes du mode expert peuvent être effectués près des bords de l'écran grâce à la propriété de redondance du LeafMenu. La **FIGURE 71** montre qu'il est possible de sélectionner l'item 6 dans les quatre coins de l'écran. Comme expliqué précédemment, dans le cas des menus circulaires, une partie des items ne peut pas être sélectionnée en mode expert près des bords de l'écran. Enfin la redondance des gestes permet d'éviter à l'utilisateur d'effectuer des gestes inconfortables. Notamment nous avons vu dans le **CHAPITRE 2** que les gestes diagonaux dans l'axe du pouce étaient plus difficiles à exécuter. Avec le LeafMenu, l'utilisateur a le choix : par exemple l'item 4 peut être sélectionné aussi bien par des gestes diagonaux dans l'axe du pouce que par des gestes diagonaux perpendiculaires à l'axe du pouce (**FIG. 72**).

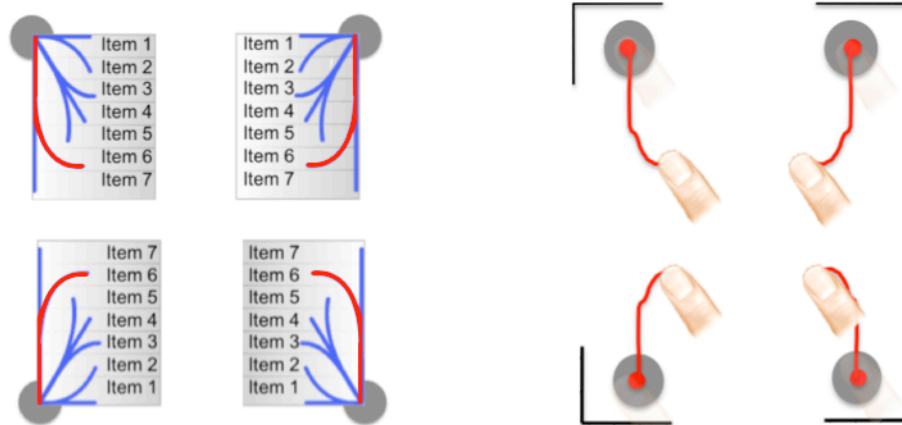


Figure 71. Les 4 raccourcis gestuels associés à l'item 6 permettent la sélection de cet item en mode expert quel que soit l'endroit où se situe le point d'activation.

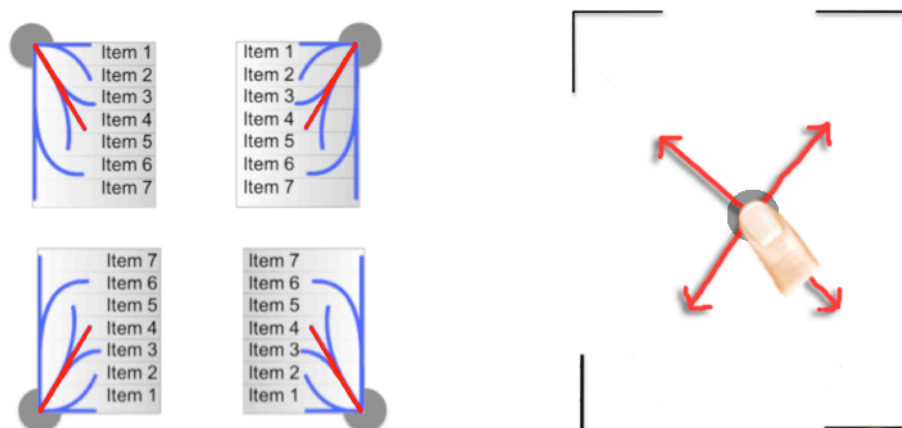


Figure 72. L'utilisateur peut choisir un des quatre gestes possibles pour sélectionner un item.

### *Occupation spatiale*

Le LeafMenu conserve plusieurs propriétés des menus linéaires. Tout d'abord l'occupation spatiale qui est moindre que celle des menus circulaires. Il permet aussi les groupements internes. Dans ce cas, les séparateurs peuvent renforcer l'appartenance des items à un style de geste : les marques associées aux items d'un même groupe partagent la même orientation (FIG. 65). Enfin il n'est pas limité à sept raccourcis gestuels. Etant un menu linéaire, il peut contenir plus de sept commandes. Dans cette situation, les raccourcis gestuels sont alors réservés aux commandes les plus fréquentes.

### *Apprentissage et mémorisation*

Le LeafMenu est une technique hybride qui fonctionne comme les menus linéaires en mode novice, mais qui permet comme les Marking Menus, une interaction gestuelle en mode expert. Ainsi, un utilisateur néophyte peut facilement prendre en main le LeafMenu car il ne sera pas surpris ni par sa représentation ni par son interaction en mode novice. De plus, une fois la commande activée, le retour visuel suggère à l'utilisateur d'effectuer le geste adéquat (TAPIA 95). Un autre avantage du LeafMenu est la correspondance spatiale entre les gestes et les items. Il existe en effet une corrélation entre la position d'un item dans le menu et la forme de son raccourci gestuel, ce qui devrait favoriser la mémorisation du mode expert. Dans les menus linéaires gestuels existants (KURTENBACH 93) (APPERT 09), la correspondance est parfois sémantique (par exemple, une geste en forme de croix correspond à la commande "fermer"), mais elle est plus souvent arbitraire. Enfin, nous pensons que l'effet miroir et la redondance des gestes aident les utilisateurs à se souvenir des gestes. En effet au lieu de mémoriser 24 gestes différents, nous suggérons qu'il est seulement nécessaire d'apprendre les 7 primitives et de les associer aux directions. Dans la section suivante, nous détaillons une expérience préliminaire que nous avons menée pour vérifier cet effet.

### *Consommation des événements*

Le LeafMenu utilise les trois événements de l'écran tactile (Press, Drag et Release). Un délai permet de passer dans le mode novice mais le mode expert peut être incompatible avec les techniques de déplacements utilisées pour faire défiler une liste ou déplacer un objet. Nous verrons comment contourner ce problème dans la section "intégration".

## 3.3 Implémentation

Le mode expert du LeafMenu est basé sur un algorithme de reconnaissance qui fonctionne comme suit : l'ensemble des points récupérés par l'écran tactile lorsque l'utilisateur effectue un geste forme une trace. Nous calculons la longueur de cette trace et considérons deux points : le

point situé au tiers de la longueur totale du tracé et celui situé aux deux tiers de cette longueur. Nous calculons ensuite deux angles : celui formé entre le point de départ du geste de le premier point calculé à l'étape précédente et celui formé entre le second point et le point final de la trace. En cherchant la position de ces angles sur un cercle trigonométrique découpé en 8 secteurs réguliers, nous pouvons reconnaître le geste effectué.

## 4. Expérimentation

Nous avons mené une expérience contrôlée pour vérifier la capacité des utilisateurs à mémoriser correctement les gestes du LeafMenu. Nous cherchions à savoir s'ils peuvent facilement apprendre les gestes du mode expert et si l'effet miroir a un impact sur cet apprentissage.

### 4.1 Plan d'expérimentation

#### *Tâche*

La tâche consistait à effectuer une suite de sélections de commandes avec le LeafMenu. Les participants tenaient le dispositif avec leur main dominante et interagissaient avec le pouce, sans s'aider d'un support. Le menu devait être activé sur une cible affichée à l'écran et l'item à sélectionner était affiché en haut de l'écran (FIG. 73). Après avoir expliqué le fonctionnement du menu, les participants se sont entraînés pendant 5 minutes. Il leur était ensuite demandé de sélectionner la cible aussi vite et précisément que possible. Aucune consigne n'indiquait le mode de fonctionnement (novice, expert) à utiliser. Cependant le mode expert étant plus rapide que le mode novice, les utilisateurs devaient donc préférentiellement utiliser le mode expert (et donc l'apprendre) pour suivre la consigne. Les utilisateurs pouvaient faire une pause à chaque fin de bloc (voir le design expérimental). L'essai finissait quand les utilisateurs relâchaient le pouce de l'écran, qu'ils aient réussi ou non la sélection, et un son indiquait si celle-ci était correcte.

#### *Matériel et participants*

Les techniques ont été implémentées en C# (.Net Compact Framework) et fonctionnaient sur Windows Mobile 5.0. L'expérience a été réalisée sur un HTC P3600 QVGA (320x240). 8 volontaires ont été recrutés dans notre institution et ont reçu une poignée de bonbons pour leur participation. Tous utilisaient un dispositif mobile à écran tactile pour la première fois et n'étaient pas familiarisés avec les interfaces gestuelles.



Figure 73. Sélection de l'item "Soldat" parmi 7 autres.

### Conditions expérimentales

Le LeafMenu contenait 7 items (FIG. 73) organisés en trois groupements internes de taille 2-3-2. Les noms des items dans un même groupe étaient sémantiquement proches (professions, matériaux, enfance). Les cibles étaient placées dans les quatre coins de l'écran afin de pouvoir tester les quatre représentations du LeafMenu. Ainsi nous plaçons les utilisateurs dans le cas le plus défavorable car ils sont confrontés successivement aux différentes symétries horizontales et verticales. Les résultats obtenus au cours de cette expérimentation doivent ainsi être considérés comme une borne inférieure et de meilleurs résultats seraient vraisemblablement obtenus dans un contexte réel.

### Design expérimental

Nous avons conduit une analyse de variance à mesures répétées chez les mêmes sujets (within-subject). Les variables indépendantes étaient les *Items* (de 1 à 7) et les *Zones de l'écran* (les cibles étaient dans les quatre coins de l'écran). La présentation des items était contrebalancée par un carré latin et la position de la cible choisie aléatoirement sans remise. Tous les participants ont sélectionné 10 fois les 7 items dans les 4 zones de l'écran. Un essai correspond à l'activation d'un des 7 items du menu sur l'une des 4 cibles. Un bloc comprend  $7 \times 4 = 28$  essais afin que chaque item puisse être sélectionné sur chaque cible aux quatre coins de l'écran. Les participants se sont entraînés 5 minutes avant de réaliser les 10 blocs. Pour résumer le design expérimental était le suivant :

*1 Technique x 4 Zones x 7 Items x 10 Blocs = 280 sélections (20 minutes) par participant*

## 4.2 Résultats et discussion

L'analyse des résultats a principalement porté sur l'utilisation du mode expert au cours de l'expérience. Nous étudions ci-après les résultats de manière générale, puis nous détaillons les résultats relatifs aux sept gestes du LeafMenu.

### *Utilisation du mode expert*

L'analyse de la variance (ANOVA) montre un effet significatif du nombre de sélections réussies en mode expert en fonction de l'indice du bloc ( $F_{9,63} = 25.9, p < 0.0001$ ). La **FIGURE 74** représente les taux de sélection en mode expert selon les blocs (sélections que les utilisateurs ont tenté en mode expert, sur l'ensemble des sélections). Nous pouvons constater que 68,8% des sélections du bloc 2 sont effectuées en mode expert et 88,9% en moyenne des sélections le sont pour les blocs suivants. Les utilisateurs ont donc mémorisé les gestes du mode expert dès le bloc 2. En nous intéressant plus particulièrement aux taux de réussite en mode expert (sélections que les utilisateurs ont tentées et réussies en mode expert) (**FIG. 75**), nous pouvons voir que non seulement les utilisateurs tentent le mode expert, mais ils le réussissent avec un taux d'erreurs acceptables (75,4% de réussite sur le bloc 2 et 89,2% en moyenne sur les blocs suivants). Ceci montre que les utilisateurs apprennent les raccourcis gestuels et les exécutent correctement.

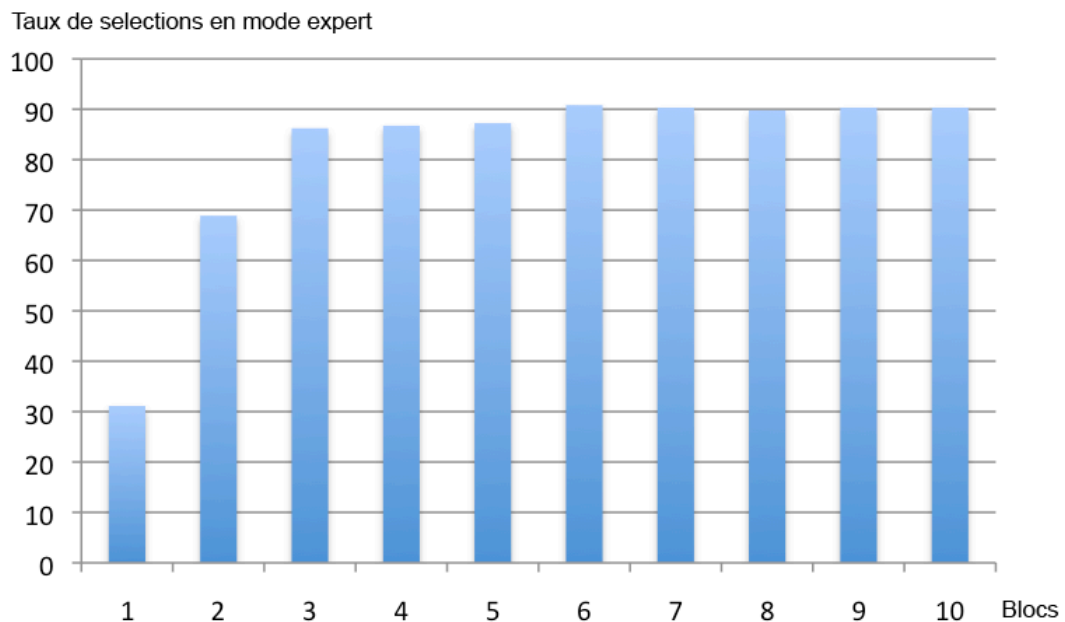


Figure 74. Pourcentage de sélections testées en mode expert.



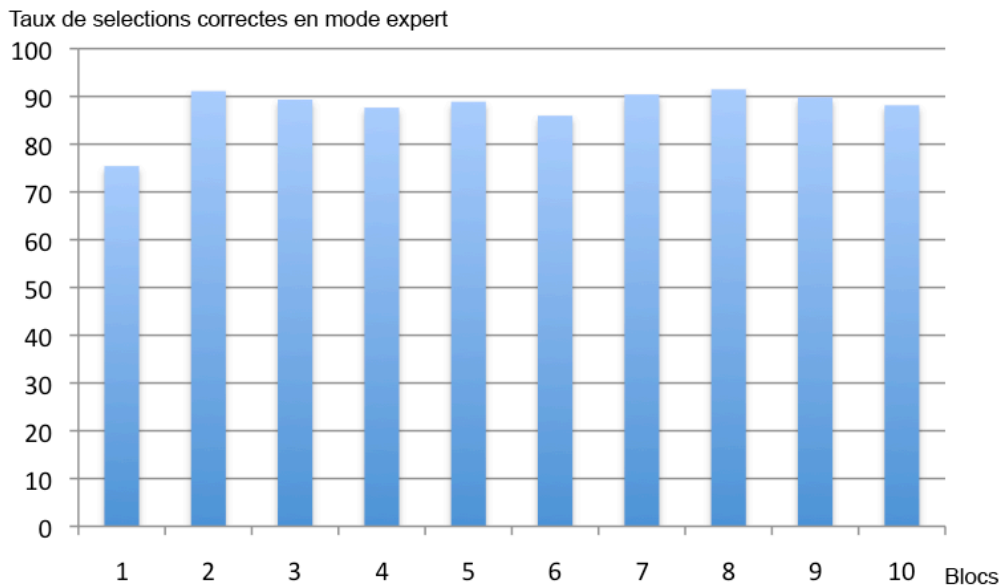


Figure 75. Pourcentage de sélections réussies en mode expert (sur le nombre total de sélections tentées en mode expert).

Nous avons aussi constaté que le taux de réussite du mode expert (89,1%) dépassait celui du mode novice (84,3%) sur les cinq derniers blocs. Le mode novice est en effet moins précis à cause de la petite taille des items du menu. On peut ainsi supposer que les participants utilisaient le mode expert non seulement pour aller plus vite (0,7s en mode expert contre 3,3s en mode novice), mais aussi pour réduire les erreurs.

Enfin, un autre résultat intéressant émerge des données : le taux de sélection en mode expert n'est pas nul au premier bloc. Ceci est surprenant car les 28 sélections demandées au bloc 1 étaient toutes différentes. Les utilisateurs les rencontraient donc pour la première fois et on peut émettre l'hypothèse qu'ils n'étaient pas censés connaître les raccourcis gestuels adéquats (ne les ayant jamais réalisés auparavant). Or 31,1 % des sélections ont été réalisées en mode expert au bloc 1. De plus, le taux de réussite de ces tentatives en mode expert est de 75,4%. Ceci montre que les participants ont déduit certains raccourcis gestuels à partir de ceux qu'ils avaient déjà fait pour d'autres orientations. Il n'est donc pas nécessaire d'apprendre explicitement les 4 variantes de gestes associées à chaque commande.

#### *Détail des gestes du mode expert*

La **FIGURE 76** détaille l'évolution des sélections effectuées correctement en mode expert en fonction des différents gestes et montre que ceux-ci ne sont pas appris de la même manière ( $F_{6,42} = 3.76$ ,  $p < 0.005$ ). Les gestes diagonaux courbés (items 3 et 5) sont moins bien appris que les autres. Au premier bloc ils sont moins bien réussis (6,2% de réussite) que les autres gestes courbés (items 2 et 4) (17,2% de réussite) et que les gestes droits (items 1, 6 et 7) (36,5% de réussite). Les autres gestes, c'est à dire ceux correspondant aux items 1, 2, 4, 6 et 7, sont réussis

à plus de 73,8% dès le bloc 2, alors que les gestes diagonaux courbés (items 3 et 5) atteignent plus difficilement ce seuil, excepté pour l’item 5 qui rejoint les performances des autres à partir du bloc 7 (75% de réussite). Ceci a été confirmé par les entretiens avec les utilisateurs qui ont perçus les gestes diagonaux courbés (items 3 et 5) comme plus difficiles à assimiler. Ceci suggère que la répétition n’est pas le seul facteur favorisant l’apprentissage du mode expert. La simplicité des gestes et leur représentation dans le menu sont sans doute des éléments essentiels à leur apprentissage (BAILLY 09).

Il faut enfin noter que le taux de réussite en mode expert pourrait être amélioré en révisant l’algorithme de reconnaissance de gestes pour mieux prendre en compte les caractéristiques de l’interaction au pouce. Par exemple, 28% des sélections de l’item 3 en mode expert ont été reconnues comme erronées (28%). Mais une analyse des tracés montre que 12% de ces erreurs auraient pu être évitées en prenant en compte la morphologie du pouce qui introduit une variation angulaire qu’il convient d’étudier plus en détail. Par exemple, nous nous sommes rendu compte que les utilisateurs ne dessinaient pas une diagonale exacte pour le geste associé à l’item 3, ce qui n’était pas prévu par notre algorithme.

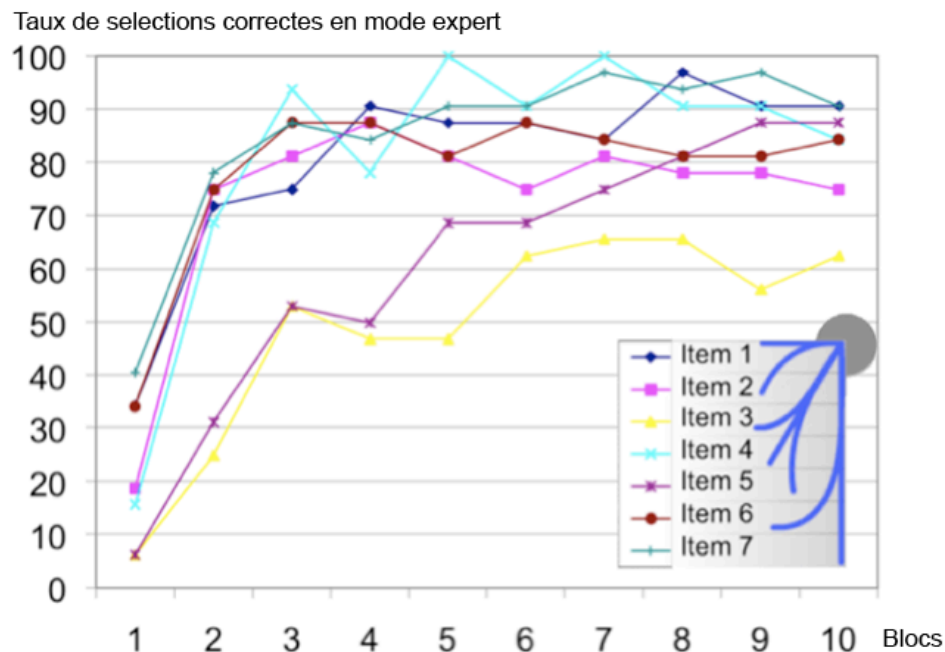


Figure 76. Pourcentage de selections correctes en mode expert en fonction des différents items et de l’indice du bloc.

### Avis subjectifs

Après l’expérience, nous avons collecté les remarques spontanées de nos participants. Ces derniers ont tous affirmé ne pas avoir été gênés par l’effet miroir du LeafMenu. Certains ont même ajouté que cette représentation semblait plus "logique". Ils ont trouvé "amusant" le fait

de pouvoir dessiner des marques, même s'ils étaient parfois frustrés que le système ne reconnaisse pas le geste exécuté en particulier sur les diagonales courbées. Ceci confirme qu'il convient d'améliorer notre algorithme de reconnaissance. Les utilisateurs ont également apprécié le fait de pouvoir se passer du stylet pour interagir avec le menu et d'utiliser des commandes gestuelles.

## 5. Intégration

Dans cette section, nous présentons des alternatives et des exemples d'intégration du LeafMenu dans diverses applications.

### *LeafMenu Hiérarchique*

Bien qu'il n'y ait que sept raccourcis gestuels dans le LeafMenu, il peut en contenir davantage en n'attribuant les gestes qu'aux items les plus fréquemment utilisés. Le LeafMenu peut de plus être hiérarchique. Dans ce cas, nous utilisons le même mécanisme que les Multi-Stroke Menus (ZHAO 04), où l'utilisateur dessine des marques temporelles. Grâce à la forme linéaire du LeafMenu, il est possible d'afficher au moins deux sous-menu côte à côte sur l'écran. Lorsque le pouce est au-dessus d'un item hiérarchique, le sous-menu correspondant est affiché, à gauche ou à droite en fonction de la place disponible. Ainsi, les utilisateurs peuvent facilement prévisualiser les commandes situées dans les sous-menus. Lorsqu'il y a plus d'item, le LeafMenu hiérarchique superpose les sous-menus comme dans le Multi-Stroke Menu.

Nous avons souhaité vérifier si ce LeafMenu à deux niveaux (permettant la prévisualisation) pouvait accueillir les commandes d'applications courantes. Nous avons donc étudié 24 applications disponibles sur un HTC Touch HD tournant sous le système Windows Mobile 6.0. Sur ces 24 applications, une seulement a un menu de profondeur supérieure à trois, environ un tiers possède 2 niveaux et les autres n'ont pas de sous-menus. Ainsi la grande majorité des applications peuvent utiliser le LeafMenu à deux niveaux. Il y a de plus en moyenne 6,3 items au premier niveau des menus, ce qui permettrait d'associer un geste du LeafMenu à chacun des items. Certaines applications ont davantage d'items (quatre ont 10 items, deux ont 9 items et quatre ont 8 items). Cependant c'est dans ces cas extrêmes que l'on trouve les commandes les plus "redondantes". Par exemple, la commande "quitter" peut aussi être activée en cliquant sur un bouton en haut de la fenêtre. C'est aussi le cas des commandes "zoomer/dézoomer" qui pourraient être supprimées des menus et remplacées par des gestes spécifiques comme ceux de pincements des deux doigts (Pinch & Expand). De plus, certaines commandes très courantes (par exemple couper/copier/coller) pourraient être activables via des gestes spécifiques en dehors des menus. Ceci permettrait de diminuer le nombre de commandes à placer dans les menus.

### LeafMail

Le LeafMenu consomme l'événement Drag de l'écran tactile, ce qui peut gêner l'activation d'autres fonctionnalités. C'est en particulier le cas pour le mode expert car l'utilisateur peut directement effectuer un geste, alors que le mode novice est activé après un appui long sur l'écran. Ceci peut poser problème dans une application affichant des listes, par exemple Mail sur l'iPhone : un déplacement vertical du doigt permet de faire défiler la liste des mails et un déplacement horizontal permet de supprimer un item (FIG. 77.A). Afin de permettre à la fois à l'utilisateur de faire défiler la liste et d'effectuer les gestes du mode expert d'un LeafMenu, une solution est d'utiliser le principe du Control Menu (POOK 00) ou Flow Menu (GUIMBRETIERE 00) : le menu contient les commandes de suppression et de défilement (FIG. 77.B). Lorsque l'utilisateur est novice, il appuie sur l'écran, attend un délai et sélectionne la commande "Scroll". Il peut alors faire défiler la liste des mails en continuant à déplacer son doigt sur l'écran (le menu disparaissant). En mode expert, l'utilisateur peut faire défiler la liste en déplaçant son doigt verticalement puisque ce geste correspond également à l'activation de la commande de défilement.



Figure 77. Leaf Mail.

### CornerMenu

Baglioni (BAGLIONI 08) propose une adaptation du LeafMenu : le CornerMenu (FIG. 78) qui conserve les gestes courbes du LeafMenu et du Flower Menu. Cette technique nécessite une zone d'activation dans un coin de l'écran. Elle peut être compatible avec les autres interactions pouvant exister dans une application puisque l'utilisateur doit commencer les gestes dans une zone spécifique. L'utilisateur peut donc à la fois faire défiler le contenu d'une application et effectuer les gestes du CornerMenu.



Figure 78. Corner Menu.

## 6. Conclusion

Nous avons présenté le LeafMenu, une nouvelle technique de menu utilisable au pouce sur écrans tactiles qui comble les inconvénients des techniques de menus existantes (FIG. 79). Le LeafMenu augmente les menus linéaires par des raccourcis gestuels (inspirés du Flower Menu). Contrairement aux menus circulaires, la représentation graphique du LeafMenu, identique à celle des menus linéaires, ne masque que partiellement les petites cibles et évite les dépassements des bords de l'écran. Le LeafMenu permet aussi de minimiser l'occultation du menu par le pouce et offre à l'utilisateur la possibilité de faire les gestes du mode expert quelle que soit la position du doigt sur l'écran (grâce à la redondance des gestes et l'effet miroir). L'expérience contrôlée a montré que les utilisateurs étaient capables d'effectuer et de mémoriser les gestes du mode expert. La simplicité des gestes et leur corrélation avec la position d'un item dans le menu permettent de les apprendre rapidement. Les utilisateurs ont apprécié le fait de pouvoir se passer du stylet pour interagir avec le menu et n'ont pas été gênés par l'organisation symétrique du menu contrairement à ce que l'on aurait pu éventuellement penser. Enfin nous avons mis en évidence le fait que le mode expert (geste) était plus précis que le mode novice (pointage). Ce dernier constat suggère une évolution du LeafMenu où la reconnaissance gestuelle pourrait être activée à la fois dans le mode expert et le mode novice pour laisser le choix à l'utilisateur novice de pointer un item ou de faire un geste.

	Occultation	Imprécision	Zones difficiles d'accès	Manque de	Occupation spatiale
<b>LeafMenus</b>	Près du bord opposé au pouce	Mode expert plus précis que mode novice	Redondance des gestes + Effet symétrique	Mode expert gestuel	½ écran, symétrie évite dépassement près des bords

Figure 79. Tableau récapitulatif des propriétés du LeafMenu.

# Gestes physiques sur écrans tactiles



"Le téléphone portable est l'invention que les utilisateurs détestent le plus (obligation d'être joignable à toute heure, gêne dans les lieux publics etc.) mais ils ne peuvent vivre sans. 95% de sondés (Etats-Unis) prétendent qu'ils améliorent la qualité de vie"

The eighth annual Lemelson-MIT Invention Index study  
(2004)

<b>1. Introduction</b>	<b>121</b>
<b>2. Etat de l'art</b>	<b>122</b>
2.1 Gestes de lancer	122
2.2 Gestes oscillatoires	123
2.3 Gestes utilisant la forme du doigt	125
2.4 Gestes multi-doigts	126
2.5 Synthèse de l'état de l'art	127
<b>3. MicroRolls</b>	<b>128</b>
3.1 Roulements versus glissements	128
3.2 Algorithme de reconnaissance	130
3.3 Application des MicroRolls	136
<b>4. Expérimentations</b>	<b>137</b>
4.1 Expérience 1	138
4.2 Expérience 2	142
4.3 Préférences utilisateurs	145
<b>5. Intégration</b>	<b>145</b>
<b>6. Conclusion</b>	<b>147</b>

## 1. Introduction

Dans le **CHAPITRE 4**, nous avons vu que les gestes sémaphoriques sur écrans tactiles sont des alternatives aux gestes déictiques (de pointage). Ils peuvent en effet être effectués rapidement et précisément et permettent de sélectionner des commandes sans regarder l'écran. Les techniques d'interaction utilisant ces gestes sont cependant incompatibles entre elles puisqu'elles exploitent toutes l'événement Drag de l'écran tactile. Comme nous l'avons vu dans l'espace problème, les écrans tactiles souffrent en effet d'importantes limitations : il n'existe que trois événements (Press, Drag et Release), le mode survol n'existe pas et il n'existe pas non plus de clic droit ou gauche comme avec une souris. Des alternatives sont alors nécessaires pour tenter d'augmenter la bande passante interactionnelle de ces dispositifs.

Une solution consiste à introduire des modes ou quasimodes (**RASKIN 00**). Un mode est un état dans lequel l'ordinateur produit un résultat différent de celui qu'aurait eu la même action dans un autre mode. Un quasimode nécessite une action de l'utilisateur pour maintenir le système dans un certain mode. Par exemple, la touche "Caps Lock" permet de verrouiller le clavier en mode majuscules, alors que la touche "Shift" active cet effet seulement lorsque l'utilisateur maintient la touche appuyée, ce qui en fait un quasimode. De la même manière, l'utilisateur pourrait choisir, en cliquant sur des boutons, d'entrer dans des modes où les gestes auraient des significations différentes, par exemple pour pointer une cible avec MagStick ou pour sélectionner un item en mode expert avec le LeafMenu. Cette approche comporte l'inconvénient de consommer de l'espace écran si le nombre de modes est important. De plus, l'utilisateur tend à oublier le mode dans lequel il se trouve (**SELLEN 92**) (**RASKIN 00**), raison pour laquelle les quasimodes sont généralement préférables aux modes.

Une alternative est de discriminer différents gestes pour activer un mode de façon automatique. Par exemple sur l'iPhone, le déplacement d'une liste se fait par un geste de lancer du doigt (appelé "Flick" ou "Swipe") et la rotation d'une image se fait en effectuant une rotation simultanée de deux doigts (appelé "Pitch & Expand"). Ces gestes ont la particularité de pouvoir être différenciés, soit par le biais d'enrichissements technologiques comme les écrans tactiles multi-doigts, soit par une analyse de tracé. Par exemple le geste "Swipe" se caractérise par une forte accélération en fin de geste ce qui permet de le différencier d'un geste de déplacement normal. Ces gestes sont de plus basés sur des métaphores ou pseudo-métaphores qui exploitent les compétences que nous avons acquises dans le monde réel, comme la métaphore du lancer ou l'utilisation simultanée de plusieurs doigts.

Dans ce chapitre nous nous intéressons aux gestes physiques permettant de combler le manque d'états interactionnels des écrans tactiles. Nous présentons l'état de l'art, puis proposons de nouveaux gestes appelés MicroRolls (**ROUDAUT, LECOLINET 09**), qui enrichissent le



vocabulaire interactionnel et sont compatibles avec les techniques existantes. Ces gestes sont discriminables grâce à une analyse de tracé que nous explicitons. Puis nous évaluons un cas d'application des MicroRolls dans deux expérimentations contrôlées. Avant de conclure nous présentons des exemples d'intégration des MicroRolls dans plusieurs applications.

## 2. Etat de l'art

Plusieurs gestes physiques ont été proposés pour étendre le vocabulaire d'interaction des écrans tactiles : les gestes de lancer, les gestes oscillatoires, les gestes qui utilisent la forme du doigt et les gestes multi-doigts (ou "multi-touch"). Pour faciliter la compréhension des techniques nous appellerons gestes "Drag" les gestes de déplacement du doigt habituellement utilisés sur la majorité des mobiles pour déplacer une carte ou faire défiler une liste.

### 2.1 Gestes de lancer

Les gestes de type "Flick" (ou "Swipe") (GEISSLER 98) sont caractérisés par une accélération à la fin du geste, ce qui permet de les différencier des gestes "Drag". Les points récupérés par l'écran tactile (collectés suivant la rapidité du dispositif), permettent d'observer cette accélération : l'écartement entre les points du tracé augmente jusqu'au relâchement du doigt. Ces gestes ont notamment été utilisés pour la navigation sur le Web (MOYLE 03) où des gestes "Swipe" directionnels permettent d'accéder à la page suivante ou précédente. Ils sont aussi utilisés dans BezelSwipe (ROTH 09) où l'utilisateur effectue ces gestes en partant de différents bords de l'écran pour sélectionner des images, des mots ou des paragraphes (FIG. 80).

D'autres techniques basées sur ces gestes ont été proposées, comme SuperFlick (REETZ 06) qui consiste à lancer un objet avec un geste "Swipe" et, pendant le mouvement de l'objet, à déplacer le stylet sur l'écran pour ajuster la direction du mouvement. Enfin, dans MultiFlick (ALIAKSEYEU 08) l'utilisateur effectue des gestes "Swipe" successifs (de gauche à droite et inversement) en relâchant le stylet à la fin de chaque geste pour faire défiler un document.

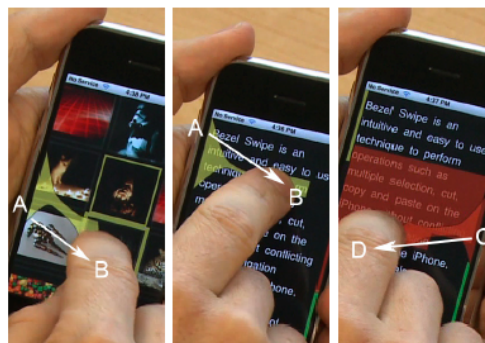


Figure 80. BezelSwipe : l'utilisateur effectue des gestes "Swipe" en partant des bords de l'écran. Plusieurs endroits sur les bords possèdent une fonction spécifique : la sélection d'images, de mots ou de paragraphes.

## 2.2 Gestes oscillatoires

Dans (OLWAL 08), les auteurs proposent des gestes basés sur la métaphore du gommage ou "Rubbing" (FIG. 81). Ce sont des gestes diagonaux de va-et-vient sur l'écran tactile (gestes oscillatoires d'excentricité nulle). Un zoom avant ou arrière est déclenché selon la direction du geste. A chaque composante diagonale supplémentaire du geste, le facteur de zoom est multiplié ou divisé d'une valeur donnée. Les rebroussements (changements de sens du geste) permettent d'identifier ce type de geste. Un calcul d'angle permet ensuite de déterminer son orientation (Sud-Ouest/Nord-est ou Sud-Est/Nord-Ouest). Les composantes diagonales du geste doivent avoir une longueur comprise entre 3 et 50 pixels et une durée inférieure à 500ms, ce qui permet de combiner les gestes "Rubbing" avec les gestes "Drag", ce que les auteurs appelle le Rub-Pointing : si l'utilisateur déplace son doigt sur l'écran sans faire de rebroussement, la vue est déplacée et le zoom n'est pas activé. Les auteurs proposent aussi des variantes comme Rub-Tapping qui permet de zoomer avec un doigt et de sélectionner un objet d'intérêt avec l'autre main.

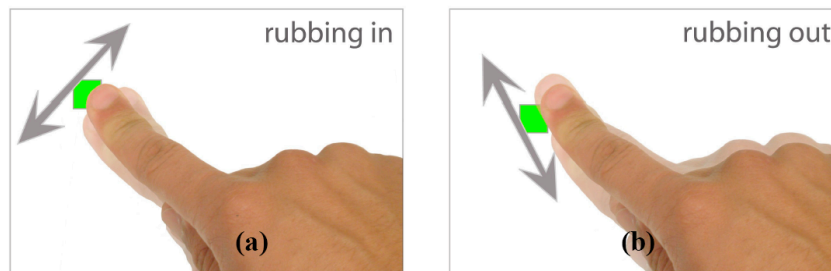


Figure 81. Les gestes "Rubbing" sont basés sur la métaphore du gommage.

D'autres études ont aussi exploité les mouvements oscillatoires de différentes manières. Virtual Scroll Ring (MOSCOVICH 04) permet de faire défiler des documents en effectuant des gestes circulaires plutôt que linéaires. Radial Scroll (SMITH 04) utilise un cercle divisé en segments placés sur un document (FIG. 82.A). A chaque fois que le tracé de l'utilisateur croise un des segments, le document est déplacé d'une valeur donnée. Cette technique force cependant l'utilisateur à effectuer le geste autour d'une position fixe sur l'écran (le centre du cercle), ce qui demande à l'utilisateur d'être attentif pendant l'exécution du geste pour éviter de sortir du cercle. Les molettes circulaires des iPods utilisent aussi ce principe mais ont l'avantage d'être physique, l'utilisateur pouvant alors effectuer le geste sans regarder la molette. CurveDial (SCHRAEFEL 05) (FIG. 82.B) répond aux problèmes de Radial Scroll. Au lieu de s'intéresser à la position de la souris autour d'un centre, cette technique détecte la courbure des arcs de cercle tracés par l'utilisateur. Le centre n'est donc plus fixe et le geste est indépendant à la position sur l'écran. De plus une nouvelle dimension est exploitée : plus la courbure du geste est grande, plus le défilement est lent et inversement. Enfin, comme dans Radial Scroll, le sens du mouvement détermine le sens de défilement du document.

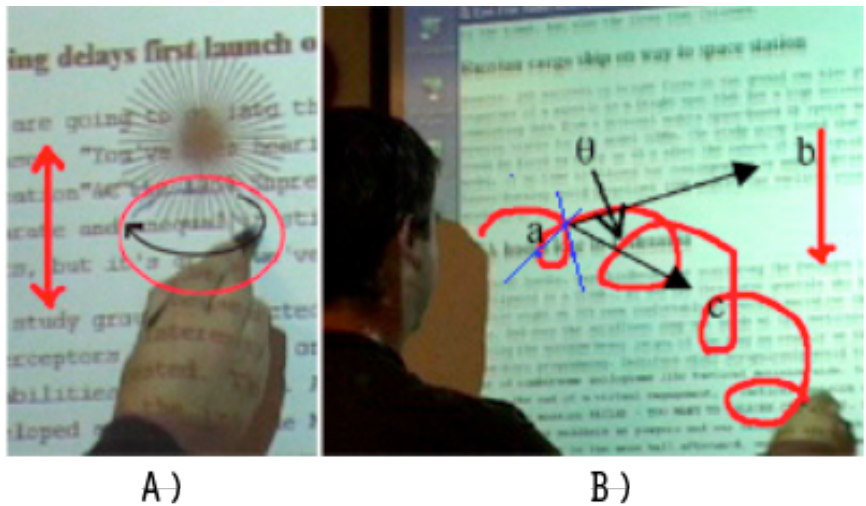


Figure 82. A) Radial Scroll, B) CurveDial.

Récemment, deux études (FEKETE 09) (MALACRIA 10) ont proposé d'exploiter de façon plus systématique les atouts des mouvements oscillatoires elliptiques, ceux-ci offrant cinq dimensions qui peuvent être contrôlées plus ou moins indépendamment : fréquence, échelle, excentricité, orientation et sens de rotation. Par exemple, dans CycloStar (MALACRIA 10) (FIG. 83), ces propriétés sont utilisées pour contrôler le zoom et le déplacement d'une carte. Lorsque l'oscillation est un cercle (excentricité proche de 1) l'utilisateur contrôle le zoom, lorsque l'oscillation est plate (excentricité proche de 0) l'utilisateur contrôle le déplacement. La fréquence de l'oscillation sert à contrôler le gain pour accélérer le zoom ou le déplacement. Comme montré par les auteurs, ces gestes sont compatibles avec d'autres styles d'interaction (comme les gestes "Drag" et "Swipe") car le mouvement oscillatoire n'est déclenché qu'après le premier rebroussement de la trace.

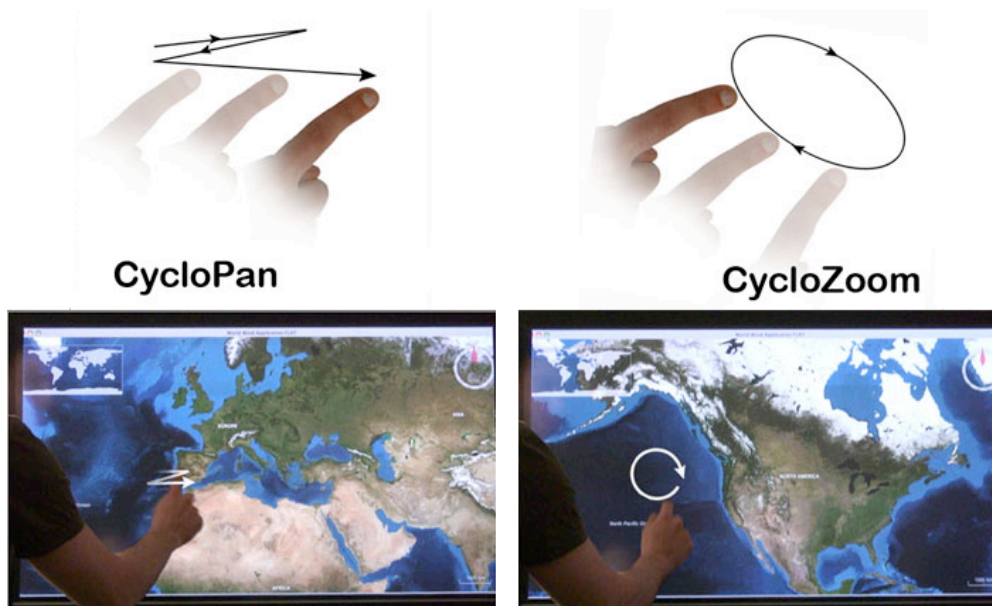


Figure 83. CycloStar.

### 2.3 Gestes utilisant la forme du doigt

RollPad (ONISZCZAK 04) (FIG. 84) est un clavier virtuel basé sur le principe du clavier T12. Dans le clavier T12 original, trois lettres partagent la même touche et la répétition d'une touche permet d'atteindre une de ces lettres. Avec RollPad, l'appui sur une touche sélectionne la lettre centrale. Mais pour accéder aux deux autres lettres l'utilisateur doit faire un mouvement de roulement du doigt sur la gauche ou la droite de la touche. D'un point de vue technique, le système calcule l'orientation et la distance entre le premier et le dernier point de contact du doigt. Si la distance est inférieure à 5 pixels, la lettre centrale est sélectionnée, sinon l'orientation du tracé détermine la lettre sélectionnée. Ces gestes sont alors de petits gestes "Drag" effectués d'une façon différente : si l'utilisateur déplace son doigt de plus de 5 pixels sans faire de roulement, le résultat est le même. Ces deux types de gestes ne peuvent donc pas être différenciés.

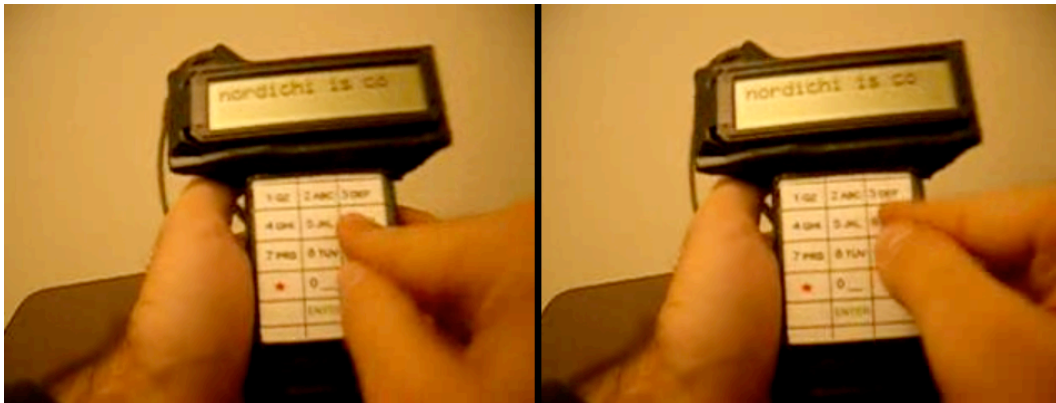


Figure 84. RollPad (l'utilisateur appuie son doigt normalement puis le fait rouler sur la droite).

SimPress (BENKO 06) simule les boutons d'une souris d'ordinateur. Cette technique détecte si l'utilisateur pose le bout ou la surface entière du doigt sur l'écran tactile (FIG. 85). Dans le premier cas, le système considère que l'utilisateur survole l'interface (comme on peut le faire avec une souris). Dans le second cas le système considère que le "bouton gauche" de la souris est enfoncé, ce qui permet, par exemple, à l'utilisateur de déplacer une cible. Cependant cette technique nécessite une technologie permettant de reconnaître la forme de la surface du doigt sur l'écran, comme le fait par exemple la table Microsoft Surface grâce à une technologie de reconnaissance vidéo placée sous la table. Cependant, les technologies utilisées actuellement sur les dispositifs mobiles détectent au mieux 10 points de contacts (soit 10 doigts comme sur le Google G1) ou une pseudo-pression (estimation de la surface du doigt pour simuler la pression).

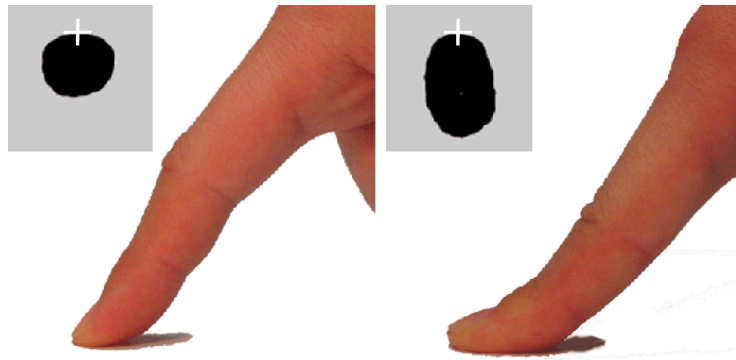


Figure 85. SimPress.

## 2.4 Gestes multi-doigts

L'arrivée de l'iPhone et des nouveaux pavés tactile pour ordinateurs portables d'Apple ont popularisé divers gestes basés sur l'utilisation de plusieurs doigts (FIG. 86). Le geste le plus connu est sans doute le "Pitch & Expand" qui sert à changer l'échelle d'un élément (FIG. 86.A). Le facteur de zoom varie en fonction de la différence de distance entre les doigts. Un autre type de geste est la rotation de deux doigts qui permet les rotations d'objets (FIG. 86.B). Par ailleurs, une translation de deux doigts permet de faire défiler verticalement ou horizontalement un document (FIG. 86.C), ce qui évite de manipuler une barre de défilement. Les mêmes gestes effectués avec trois doigts permettent de faire défiler rapidement des photos ou des pages Web (FIG. 86.D). Enfin, l'appui simultané de deux doigts entraîne l'ouverture d'un menu contextuel (FIG. 86.E).

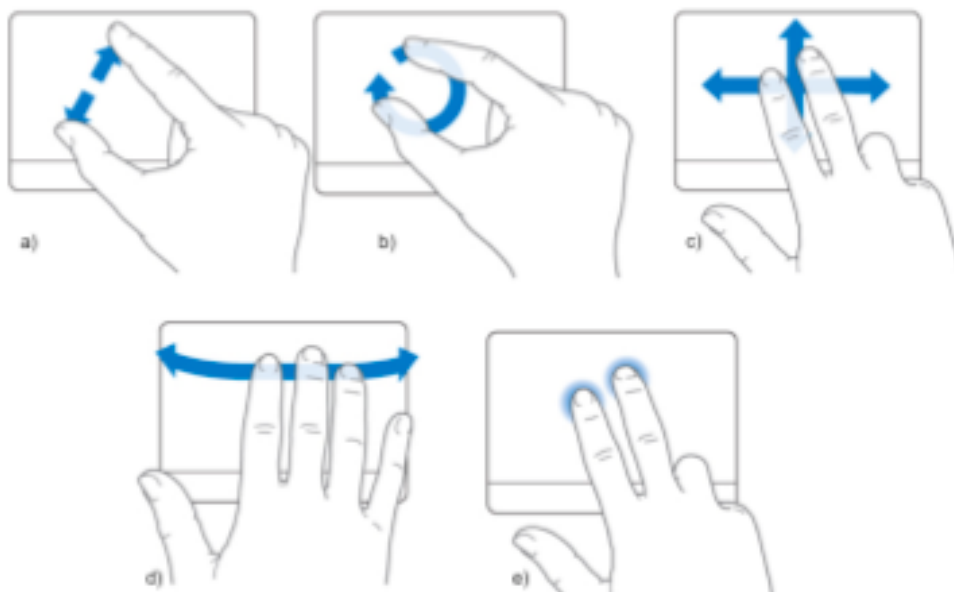


Figure 86. Gestes à deux ou trois doigts sur le pavé tactile du Mac Book Pro.

Wobbrock et al. (WOB BROCK 09) se sont également intéressés à la création d'un vocabulaire de gestes multi-doigts. Leur étude consistait à demander à 20 personnes d'effectuer, sur une table tactile, les gestes qu'ils souhaiteraient exécuter pour accomplir 27 opérations différentes (telles que déplacer, supprimer, coller ou agrandir un objet, etc.). Parmi les 1080 gestes collectés, des similitudes entre utilisateurs ont émergé, ceci permettant aux auteurs d'analyser les comportements des utilisateurs et de proposer un ensemble de gestes, dont une partie est illustrée sur la FIGURE 87.

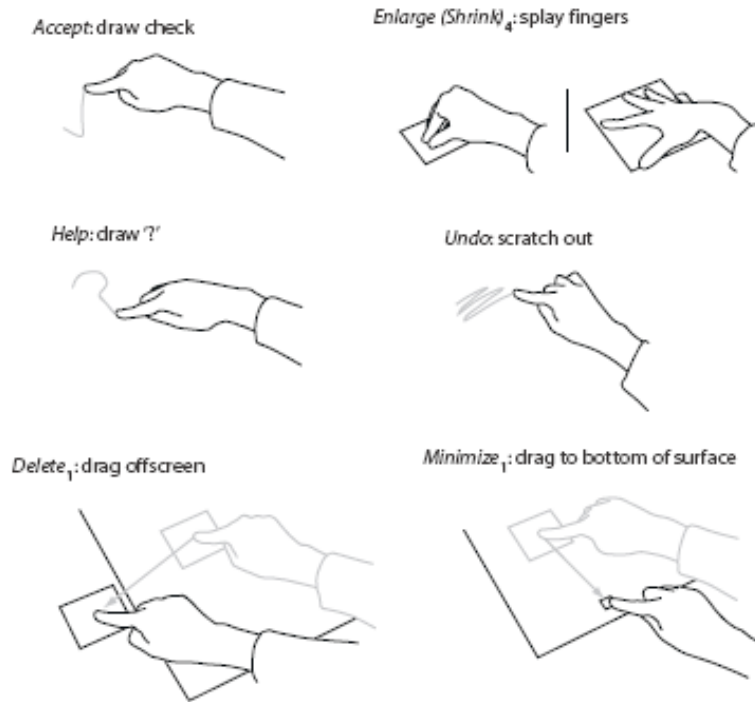


Figure 87. Vocabulaire de gestes multi-doigts.

## 2.5 Synthèse de l'état de l'art

L'état de l'art est riche de nouveaux vocabulaires de gestes pour les surfaces tactiles. Ces gestes peuvent être discriminés soit en utilisant de nouvelles technologies comme les technologies multi-touch ou vidéo (SimPress), soit par une approche logicielle en analysant la trace des gestes pour les différencier. Tous les gestes exposés présentent des caractéristiques spécifiques permettant de les distinguer d'autres gestes (notamment les gestes "Drag"), à l'exception de RollPad. Cette technique semble en effet tirer parti d'une nouvelle dimension (les roulements de doigts), mais ne propose en réalité qu'une autre manière d'effectuer un geste "Drag".



### 3. MicroRolls

Nous avons voulu participer à l'enrichissement des capacités des écrans tactiles en proposant de nouveaux gestes, les MicroRolls qui s'effectuent en "roulant" la surface du pouce sur l'écran tactile. Contrairement à RollPad, nous exploitons de manière systématique la distinction entre les gestes de roulement et ceux de glissement du pouce (tels que les gestes "Swipe", "Rubbing" ou "Drag"). Nous proposons ainsi six nouveaux gestes (FIG. 88) : quatre MicroRolls cardinaux qui sont exécutés dans les directions cardinales de l'axe du pouce (qui est légèrement incliné sur la gauche pour un droitier) et deux MicroRolls circulaires qui sont effectués dans le sens horaire ou anti-horaire.

Notons que notre hypothèse est que ces six gestes, basés sur des roulements de doigts, sont différenciables des gestes de glissement, ceci assurant la comptabilité avec ces derniers. Pour valider cette hypothèse, nous avons examiné dans un premier temps ce que la littérature nous apprend sur la discrimination de ces deux types de geste. Nous avons ensuite construit un algorithme de reconnaissance permettant cette discrimination. Nous présentons notre démarche ci-après et proposons ensuite un cas d'application de ces nouveaux gestes.



Figure 88. MicroRolls.

#### 3.1 Roulements versus glissements

Afin de vérifier s'il est possible de discriminer les gestes de roulements et de glissement, nous sommes tout d'abord penchés sur les principes de la physique élémentaire. La surface d'un doigt ayant une forme sphérique, son mouvement sur une surface tactile devrait être soumis aux mêmes lois que le mouvement d'une balle sur une surface plane. Resnik (RESNIK 66) explique que le mouvement de cette balle peut être de deux natures : les roulements ou les glissements. Nous décrivons, ci-après, une partie des travaux de Resnik qui explique cette différence.

Quand une balle touche une surface solide, la force de contact est composée de deux forces perpendiculaires N et T dont les orientations sont respectivement normale et tangentielle à cette surface (FIG. 89).

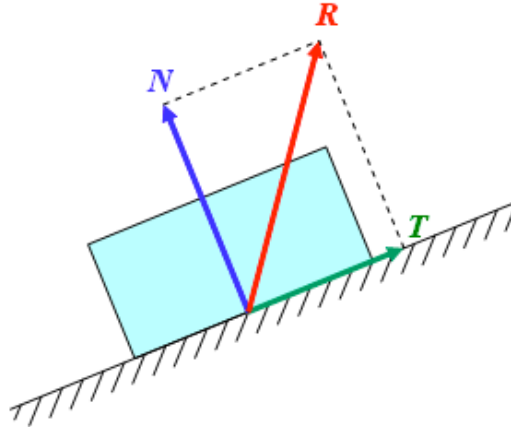


Figure 89. Décomposition d'une force (R) en ses composantes normale (N) et tangentielle (T).

- Lorsque le système est immobile, il existe une force minimum à appliquer au solide pour qu'il se déplace. C'est la force de friction statique, qui s'oppose au déplacement même lorsque le solide est à l'arrêt :  $T_s \leq \mu_s * N$  où  $\mu_s$  est le coefficient de friction statique. Lorsque les deux surfaces se déplacent l'une par rapport à l'autre, la force tangentielle est opposée à la force de friction cinétique :  $T_k = \mu_k * N$ , où  $\mu_k$  dépend des coefficients de friction cinétiques des deux matériaux en contact. La force de friction cinétique, qui s'oppose au mouvement lorsque l'objet bouge, est donc plus faible que la force de friction statique. La force nécessaire pour initier un mouvement est en effet plus importante que celle nécessaire à le poursuivre (ce que l'on peut facilement ressentir lorsque l'on pousse un corps lourd).
- Lorsqu'un solide roule sur une surface, le mécanisme est différent mais peut être décrit de la même façon à l'aide de la friction cinétique et statique :  $T_r = \mu_r * N$  où  $\mu_r$  est le coefficient de friction avec roulement. La force de roulement, qui s'oppose au mouvement de roulement, est plus faible que la force de friction cinétique (et par déduction à la force de friction statique). Il est en effet plus facile de faire rouler une roue que de la faire glisser (ce que l'on peut facilement ressentir en prenant le cas d'un corps lourd).

Le pouce, par ses propriétés sphériques, permet en théorie ces deux types de mouvement. Lorsque la peau frotte sur la surface de l'écran, c'est un glissement avec frottement (FIG. 90.A), sinon c'est un roulement (sans frottement) (FIG. 90.B). La présence ou l'absence d'un frottement permet donc à l'utilisateur de facilement différencier ces deux types de mouvements.



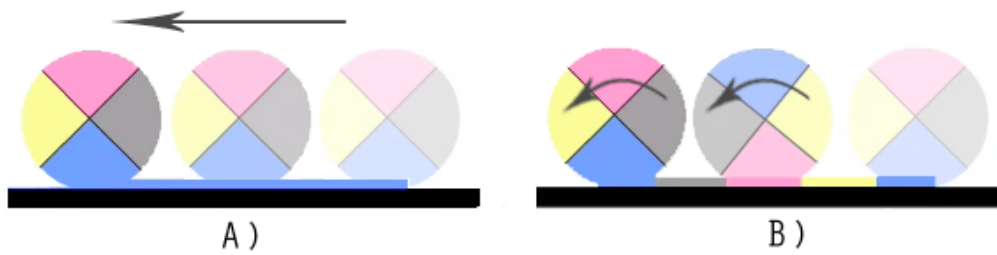


Figure 90. A) Glissement d'une balle sur une surface, le point de contact reste le même tout au long du geste ; B) Roulement d'une balle sur une surface plane, le point de contact change pendant le mouvement, il n'y a pas de frottements.

### 3.2 Algorithme de reconnaissance

Comme nous l'avons exposé dans l'état de l'art, certaines techniques ont déjà utilisé des mouvements de roulement (ONISZCZAK 04) (BENKO 06), mais à notre connaissance, aucune étude n'a exploité la possibilité de les différencier des gestes de glissement. Nous avons donc souhaité vérifier la faisabilité d'une telle discrimination. Des technologies permettant une reconnaissance de la forme du doigt ou un capteur permettant de détecter les frottements des doigts pourraient probablement distinguer ces deux types de mouvements. Nous proposons une méthode de discrimination plus simple basée seulement sur l'analyse des traces de l'utilisateur. Contrairement aux techniques existantes qui tentent de lisser le tracé du pouce sur l'écran, nous exploitons tous les points du tracé afin de discriminer les roulements des glissements du pouce. Notre approche consiste en trois phases : nous avons tout d'abord mené une expérience afin de collecter un ensemble de gestes ; nous avons ensuite analysé le tracé de ces gestes pour extraire les caractéristiques utiles à notre algorithme de reconnaissance ; enfin nous avons calculé le taux de reconnaissance de cet algorithme. Les types de geste récoltés sont au nombre de 16 (FIG. 91) :

- 6 gestes de roulement : 4 MicroRolls cardinaux et 2 MicroRolls circulaires
- 10 gestes de glissement : 4 Drags cardinaux, 4 Swipes cardinaux et les 2 gestes de Rubbing orientés à 45° et 135°.

#### *Phase 1 : plan d'expérimentation/récolte d'échantillons de gestes*

**TÂCHE.** La tâche consistait à effectuer les 16 gestes élémentaires. Les participants tenaient le dispositif avec leur main dominante et interagissaient avec le pouce. L'essai commençait par un message indiquant le geste à effectuer. Une cible de 60 pixels de diamètre indiquait à l'utilisateur l'endroit où exécuter le mouvement. Chacun des 16 gestes était effectué 4 fois sur 9 cibles présentées sur l'écran. La position de ces cibles correspond à la division de l'écran en 9 zones de 80x106 pixels (les cibles se trouvant au centre des zones). L'essai finissait quand l'utilisateur relevait son pouce de l'écran.

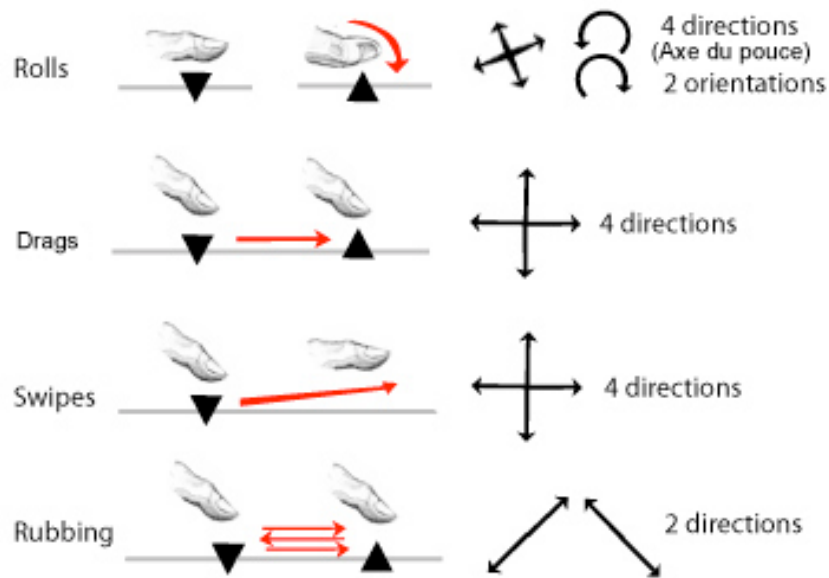


Figure 91. Vocabulaire de 16 gestes élémentaires.

**MATÉRIEL ET PARTICIPANTS.** Les techniques ont été implémentées en C# (.Net Compact Framework) et fonctionnaient sur Windows Mobile 5.0. L'expérience a été réalisée sur un HTC P3600 QVGA (320x240). 10 volontaires droitiers (dont 3 femmes), âgés de 23 à 28 ans, ont été recrutés dans notre institution et ont reçu une poignée de bonbons pour leur participation. Aucun d'entre eux n'utilisaient quotidiennement un dispositif tactile.

**DESIGN EXPÉRIMENTAL.** Un carré latin a été utilisé pour contrebalancer l'ordre des gestes entre les participants. Ceux-ci se sont entraînés 10 minutes avant de commencer l'expérience. Pour résumer le design expérimental était le suivant :

16 Gestes x 9 Zones x 4 Répétitions = 576 gestes (40 minutes) par participant

### *Phase 2 : recherche des primitives caractéristiques des gestes*

Nous pouvons observer sur la **FIGURE 92** que chaque type de geste présente des caractéristiques particulières. Par exemple, les gestes "Swipe" sont en moyenne les plus rapides (106ms). Il faut environ 451ms pour effectuer les MicroRolls cardinaux, 706ms pour les circulaires, 1139ms pour les gestes "Rubbing" et 557ms pour les gestes "Drag".

Après une analyse plus approfondie des données, nous avons développé un module de reconnaissance basé sur un algorithme d'apprentissage. Son fonctionnement est le suivant : 1) les gestes récoltés sont étiquetés et enregistrés ; 2) à chaque geste sont associées 9 primitives inspirées de l'algorithme de Rubine (**RUBINE 91**) ayant des poids différents ; 3) lorsqu'un

nouveau geste doit être reconnu, l'algorithme calcule une distance<sup>9</sup> entre le vecteur de primitive du nouveau geste et ceux des gestes de la base d'apprentissage ; 4) l'algorithme détermine alors les K plus proches voisins (dans notre cas K = 5), c'est à dire les K gestes ayant une distance euclidienne minimale avec le geste à reconnaître ; 5) Il retourne ensuite le type du geste dominant parmi les K. Les primitives utilisées pour caractériser les gestes ainsi que les poids de ces primitives ont été choisies empiriquement et sont les suivantes :

- La durée du geste (poids : 0.1)
- La distance moyenne entre les points du geste (poids : 0.1)
- La distance totale du geste (poids : 0.4)
- Le rapport entre la longueur totale et la distance entre le premier et le dernier point (poids : 0.4)
- La différence d'angle entre le premier et le dernier point, et le premier point et le point au milieu du tracé (poids : 0.1)
- La distance entre le premier point et le dernier point (poids : 0.6)
- L'angle entre le premier point et le dernier point (poids : 0.1)
- La somme des angles traversés (poids : 0.2)
- La somme absolue des angles traversés (poids : 0.1)

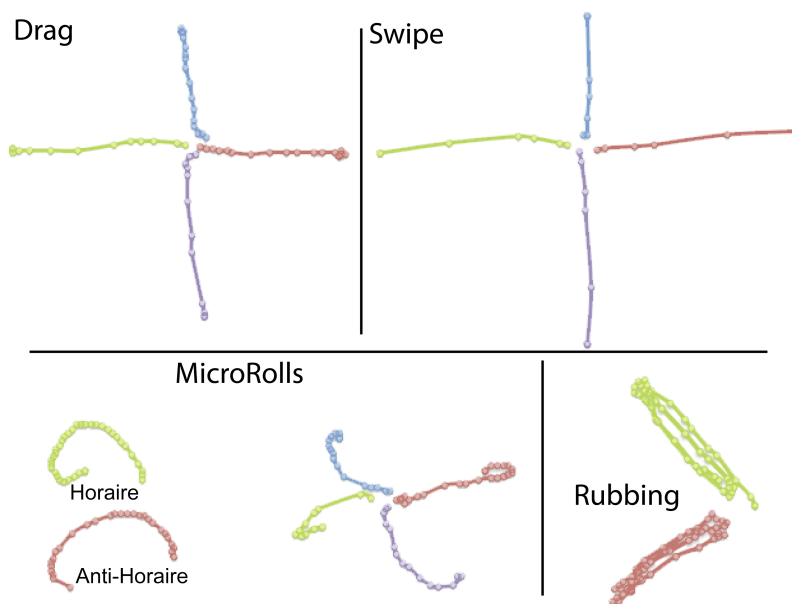


Figure 92. Tracés des 16 gestes élémentaires. Les points représentent les positions successives du barycentre de la surface du pouce sur l'écran.

<sup>9</sup> Soit deux vecteurs V1 et V2 contenant les primitives pi et leurs poids respectifs p'i.

$$Distance = \sqrt{\sum \sqrt{(V1.p_i * V1.p_i - V2.p_i * V2.p_i)}}$$

### Phrase 3 : reconnaissance des gestes

Pour tester l'efficacité de l'algorithme de reconnaissance, nous avons utilisé la moitié des gestes effectués par les utilisateurs comme base d'apprentissage (le premier et le troisième essai de chaque geste) tandis que l'autre moitié (le second et le quatrième essai de chaque geste) a servi à calculer le taux de reconnaissance. Notons que 1,7% des gestes, qui correspondaient à des tracés sans rapport avec la consigne, ont été supprimés des jeux de données pour éviter les fautes d'apprentissage de notre algorithme.

Nous nous sommes d'abord intéressé aux taux de reconnaissance des différentes catégories de gestes (toutes directions confondues). Les taux de reconnaissance que nous présentons ne sont donc pas les taux de reconnaissance de chaque geste (que nous exposons plus loin), mais ils nous donnent une indication sur la capacité de notre algorithme à reconnaître les différentes catégories de gestes.

La première version de notre algorithme permettait de reconnaître les gestes "Drag" à hauteur de 95.1%, les MicroRolls à 96.1%, les gestes "Swipe" à 97.4% et les gestes "Rubbing" à 100%. Nous avons cependant amélioré l'algorithme pour favoriser la reconnaissance des MicroRolls en pondérant différemment les primitives utilisées pour la reconnaissance. Comme le montrent les matrices de confusion de la [FIGURE 93.A](#), le taux de reconnaissance le plus élevé (99,45%) est celui des MicroRolls, suivi de celui des gestes "Rubbing" (97,92%), des gestes "Swipe" (97,68%) et enfin des gestes "Drag" (87,53%).

Pour comprendre le taux de reconnaissance relativement faible des gestes "Drag", nous avons observé les traces des gestes non reconnus. Il s'avère que la majorité d'entre eux sont ceux effectués en direction des bords de l'écran. Les utilisateurs ont en effet été gênés par l'épaisseur des bords de l'écran, particulièrement haute sur les dispositifs utilisés dans cette étude. Certaines traces sont ainsi "coupées" car les utilisateurs terminaient ces gestes sur le bord (n'étant pas sensible au contact). Cependant, on peut supposer que les utilisateurs n'exécutent pas ce type de geste naturellement, et lorsqu'ils veulent déplacer une carte par exemple, ils commencent le geste dans une partie favorable de l'écran. De ce fait nous avons établi une matrice de confusion qui correspond à un calcul qui reste fidèle à la réalité, où les gestes effectués près d'un bord de l'écran et en direction de ce même bord sont ignorés. Comme le montre la [FIGURE 93.B](#), les taux de reconnaissance augmentent pour chaque catégorie et plus spécifiquement pour les gestes "Drag" qui sont alors reconnus à 95,3%.

En gardant la configuration précédemment décrite (les gestes près des bords et vers les mêmes bords exclus), nous avons ensuite calculé la matrice de confusion pour chaque geste, directions comprises ([FIG. 94](#)), qui donne une réelle estimation des performances de l'algorithme.

	Classes reconnues					Classes reconnues			
	MicroRolls	Drag	Swipe	Rubbing		MicroRolls	Drag	Swipe	Rubbing
MicroRolls	99,45	0,53	0	0	MicroRolls	99,99	0	0	0
Drag	9,72	87,53	2,72	0	Drag	3	95,29	1,7	0
Swipe	1,85	0,46	97,68	0	Swipe	1,38	0	98,6	0
Rubbing	1,56	0,52	0	97,92	Rubbing	1,56	0,52	0	97,92

A) Avec bords

B) Sans bords

Figure 93. Matrices de confusion obtenues pour chaque catégorie de gestes dans différentes conditions : A) tous gestes compris B) tous les gestes exceptés les gestes effectués près d'un bord de l'écran et en direction de ce même bord.

De manière générale, les taux de reconnaissance sont aux alentours de 95% avec certains gestes reconnus à 100% : les MicroRolls vers le bas, les gestes "Swipe" vers le bas et les gestes "Rubbing" dans la direction Sud-Ouest/Nords-Est. Une légère baisse d'efficacité est obtenue pour les gestes "Swipe" vers le haut (93,75%). En observant les tracés ces gestes, nous avons observé que certains d'entre eux étaient formés de très peu de points, c'est à dire que les utilisateurs relâchaient très vite le doigt de l'écran, ce qui peut expliquer la baisse du taux de reconnaissance. Les gestes "Drag" vers le bas (91,52%) sont parfois confondus avec les gestes "Swipe" vers le bas (5,08%). Il semble que les utilisateurs aient effectué certains gestes "Drag" vers le bas avec une accélération en fin de tracé, ce qui peut expliquer cette confusion.

Enfin les MicroRolls circulaires ont les taux de reconnaissance les plus bas (88,42 % pour les horaires et 87,36% pour les anti-horaires). Ceux-ci sont parfois reconnus comme des MicroRolls cardinaux. Ces résultats peuvent s'expliquer par une exécution trop rapide des gestes : l'échantillonnage de points effectué par l'écran tactile n'est pas assez précis pour pouvoir détecter l'arrondi de ces gestes (FIG.92) lorsque les gestes sont effectués très rapidement, ceci ayant pour conséquence que leur forme se rapproche de celle des MicroRolls cardinaux. Notons enfin que ces résultats ont été obtenus sans utilisation d'un retour indiquant si le geste était correctement reconnu (la phase d'échantillonnage des gestes étant nécessaire et donc antérieure à la mise au point de notre algorithme). Il était donc impossible à nos participants de corriger leurs gestes et de s'adapter aux particularités de l'algorithme pour maximiser les performances. De plus l'algorithme de reconnaissance pourrait facilement être adapté pour chaque utilisateur (la base d'apprentissage contenant principalement leurs gestes), ce qui aurait certainement pour effet d'améliorer les performances. Ceci fait d'autant plus sens qu'un dispositif mobile est habituellement un appareil personnel. Il conviendrait alors de trouver des méthodes pour réduire le temps d'apprentissage du système de façon à ce que la durée d'entraînement de l'utilisateur soit tolérable.

	MicroRolls				Drag				Swipe				Rubbing			
	Horaire	Anti-horaire	Gauche	Haut	Droite	Bas	Gauche	Haut	Droite	Bas	Gauche	Haut	Droite	Bas	Sud-Est / Nord-Ouest	Sud-Ouest / Nord-Est
Horaires	88,42			1,05	9,47	1,05										
Anti-Horaire	1,05	87,36	5,26	6,31												
Gauche			96,55	3,44												
Haut	1,81		1,81	94,54	1,81											
Droite	5,17			1,72	93,1											
Bas						100										
Gauche			3,44				94,82			1,72						
Haut				3,44	1,72		94,82									
Droite								97,77	2,22							
Bas						3,38			91,52					5,08		
Gauche			2,77	1,38						94,44	1,38					
Haut										1,56	93,75	1,56		3,12		
Droite					1,38						1,38	97,22				
Bas													100			
Sud-Est / Nord-Ouest	2,08	1,04						1,04							95,83	
Sud-Ouest / Nord-Est																100

Figure 94. Matrice de confusion totale (les gestes effectués près d'un bord de l'écran et en direction de ce même bord étant ignorés).

### 3.3 Application des MicroRolls

Nous avons donc vérifié par une expérimentation contrôlée que les MicroRolls peuvent effectivement être discriminés et différenciés des gestes de glissement (10 gestes). Les 16 gestes que nous avons étudiés offrent de nombreux avantages pour l'interaction. Tandis que les gestes "Drag" et "Swipe" sont déjà couramment utilisés pour effectuer des déplacements, les gestes "Rubbing" peuvent servir d'alternative efficace aux interactions multi-doigts pour contrôler le niveau de zoom. Les interactions multi-doigts sont en effet difficilement réalisables lors de l'interaction avec une seule main. Les MicroRolls peuvent être utilisés de diverses manières dont celles que nous proposons sur la **FIGURE 95.B** où ils servent à activer les fonctionnalités les plus fréquemment utilisées dans les systèmes interactifs.

- Les MicroRolls vers le bas, la gauche et la droite remplacent les commandes couper, copier, coller
- Le MicroRoll vers le haut active un mode de précision comme TapTap pour faciliter la sélection de petites cibles
- Le MicroRoll horaire remplace le clic droit d'une souris pour ouvrir un menu contextuel (sans avoir à attendre un délai)
- Le MicroRoll anti-horaire déclenche un quasimode pour déplacer des cibles (Drag & Drop)

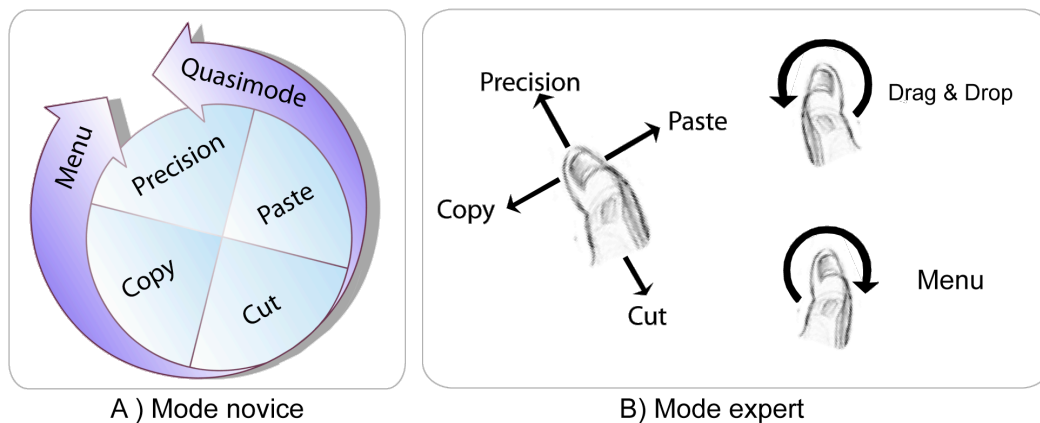


Figure 95. Association des MicroRolls aux commandes fréquemment utilisées dans les systèmes interactifs : A) mode novice et B) mode expert d'un RollMark menu.

Un problème qui peut affecter l'apprentissage des interfaces gestuelles est que les commandes sont "invisibles" : il n'y a pas de retour visuel pour aider l'utilisateur à deviner le geste adéquat. Les Marking Menus (**KURTENBACH 93**) résolvent ce problème en affichant les items si l'utilisateur appuie sur la souris pendant un bref délai, ce qui lui permet de connaître la direction du geste associé à un item. Le même mécanisme peut être appliqué aux MicroRolls.

Comme illustré sur la **FIGURE 95.A**, un *RollMark* menu permet de montrer l'association entre les MicroRolls et les commandes pour faciliter leur apprentissage et leur mémorisation. Ce menu apparaît si l'utilisateur maintient immobile son pouce à l'écran pendant 300ms. Comme pour les Marking Menus traditionnels, les gestes que l'utilisateur doit effectuer sont les mêmes en mode novice et en mode expert, la différence portant sur l'affichage ou non du menu. De plus, le RollMark menu n'interdit pas d'ouvrir un menu linéaire contextuel si cela est nécessaire, ceux-ci étant par exemple déclenchés par un MicroRoll horaire (**FIG. 95.B**).

#### 4. Expérimentations

Nous avons mené deux expériences afin d'évaluer l'efficacité des MicroRolls (MR) sur une tâche simple de copier/coller et de les comparer à deux techniques utilisées fréquemment pour cette même tâche : les barres d'outils (BO) et les menus contextuels (MC). Ces deux techniques présentent toutes deux des avantages et des inconvénients : la barre d'outils est composée de plusieurs boutons directement sélectionnables, ce qui permet donc une sélection rapide. Elle a cependant pour inconvénient d'être affichée en permanence à l'écran et consomme ainsi une quantité non négligeable d'espace écran. Un menu contextuel possède l'avantage d'être affiché seulement à la demande de l'utilisateur, mais celui-ci doit attendre un délai avant de pouvoir pointer les items du menu. Nous avons choisi ces deux techniques car elles sont les plus présentes sur les dispositifs grand public, avec une prépondérance des barres d'outils sur l'iPhone et des menus contextuels sur les terminaux Windows Mobile, Android ou Symbian.

Deux expériences ont été conduites : l'expérience 1 s'intéresse aux cas des cibles relativement larges et l'expérience 2 prend le cas de cibles plus petites, difficilement sélectionnables au pouce et nécessitant une opération supplémentaire de zoom. Les deux expériences mettent en jeu un certain nombre d'entités graphiques. Pour clarifier le vocabulaire, nous utiliserons le terme "cible" pour désigner les objets à copier/coller et le terme "bouton" pour désigner les interacteurs contenus dans les différentes techniques (boutons de la barre d'outils ou items du menu).



## 4.1 Expérience 1

### Plan d'expérimentation

**TÂCHE.** La tâche consiste à effectuer une série de copiés/collés avec les trois techniques (MicroRolls (MR), barre à outils (BO) et menu contextuel (MC)). Les participants tiennent le dispositif avec leur main dominante et interagissent avec le pouce. Pour rendre la tâche plus réaliste, la zone où l'utilisateur doit coller la cible se situe en dehors de l'écran. L'utilisateur doit donc déplacer la vue entre les deux opérations (FIG. 96.A). Les participants effectuent ainsi cinq opérations consécutives à chaque essai :

1. Sélectionner l'objet à copier (un cercle rouge de 60 pixels placé au centre de l'écran)
2. Activer la commande copier
3. Déplacer la vue pour faire apparaître la zone où l'objet doit être collé
4. Sélectionner cette zone (un cercle bleu de 60 pixels situé à 240 pixels à droite de la cible rouge)
5. Activer la commande coller

Chacune de ces opérations est réitérée en cas d'erreur (par exemple si l'utilisateur sélectionne un espace vide), ce qui fait que la tâche peut impliquer plus de cinq opérations. Les temps récoltés dans l'expérience peuvent ainsi être considérés comme une bonne mesure de la performance réelle des trois techniques. Il n'y a donc pas de taux d'erreurs, celles-ci étant prises en compte dans le temps de complétion de la tâche.

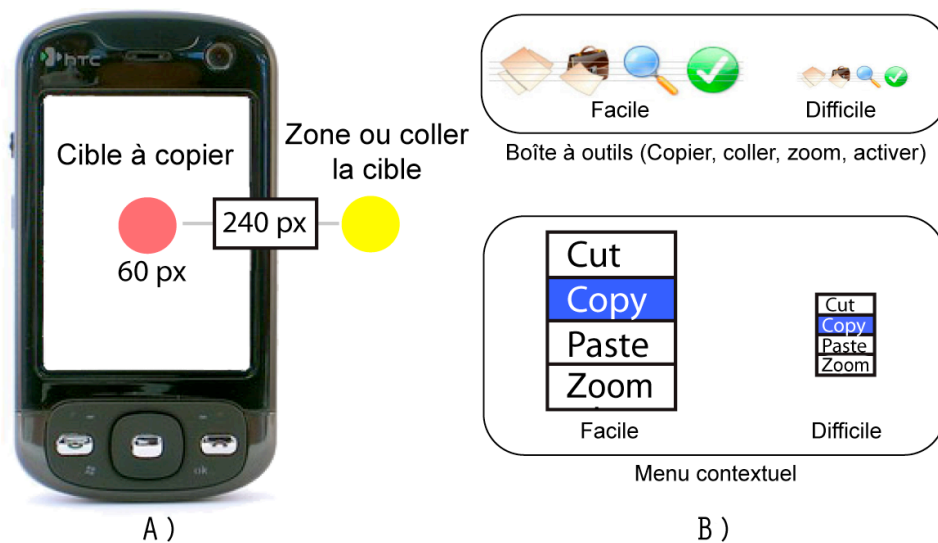


Figure 96. A) la tâche de copier/coller ; B) la barre d'outils et le menu contextuel pour les deux conditions de difficulté.

Le déplacement de la vue est effectué au moyen de geste "Drag" quelle que soit la technique utilisée. Pour conserver une égalité entre les techniques, quatre commandes sont disponibles pour toutes les techniques utilisées, mais seulement deux d'entre elles (copier et coller) sont utilisées pour cette expérience. L'activation de ces commandes varie selon les techniques comme suit :

- La barre d'outil, affichée en haut de l'écran, permet de sélectionner les 4 commandes : copier, coller, zoomer (non utilisée dans cette première expérience) ou activer. Le tap étant utilisé pour mettre en surbrillance les cibles, il n'existe pas de moyen d'activer la cible (comme le fait un double clic sur un PCs par exemple). Pour accentuer le réalisme de la tâche, la commande "activer" servait à cet effet même si elle n'était pas utilisée dans l'expérience.
- Pour faire apparaître le menu contextuel les participants appuient sur la cible et attendent un délai de 300ms. Le menu possède 4 boutons: couper, copier, coller et zoomer.
- Le MicroRolls sont au nombre de 4 : un MicroRolls vers la gauche active la commande copier et vers la droite la commande coller. Les MicroRolls vers le bas et vers le haut ne déclenchent aucune action. Il n'y a pas d'indication visuelle (textes ou icônes) et les utilisateurs doivent mémoriser les gestes.

L'efficacité des techniques étudiées dépend de plusieurs paramètres. En particulier, un point critique de cette étude est le choix de la taille des boutons. De larges boutons sont évidemment plus facilement à sélectionner, mais consomment beaucoup d'espace écran. C'est pourquoi nous avons considéré deux tailles standard de boutons qui correspondent à des cas typiques : d'un côté le cas des dispositifs proposant des petites cibles comme certains modèles Windows Mobile ou BlackBerry (20 pixels soit environ 3mm) et de l'autre, le cas des dispositifs offrant de plus larges cibles comme l'iPhone (60 pixels soit environ 9 mm). Cette taille de 9 mm correspond aussi aux recommandations établies par Parhi et al (PARHI 06) pour la sélection au pouce. Les boutons de la barre d'outils sont carrés et ceux des menus rectangulaires du fait de leur contenu textuel (un rapport de 3 entre la largeur et la hauteur a été utilisé) (FIG. 96.B).

**MATÉRIEL ET PARTICIPANTS.** Les techniques ont été implémentées en C# (.Net Compact Framework) et fonctionnaient sur Windows Mobile 5.0. L'expérience a été réalisée sur un HTC P3600 QVGA (320x240). 12 volontaires droitiers (dont 2 femmes), âgés de 23 à 31 ans, ont été recrutés dans notre institution et ont reçu une poignée de bonbons pour leur participation.

**DESIGN EXPÉRIMENTAL.** Nous avons conduit une analyse de variance à mesures répétées chez les mêmes sujets (within-subject). Les variables indépendantes sont les *Techniques* (MR, BO, MC) et les *Niveaux de difficulté* (Facile avec des boutons de 60 pixels, Difficile pour ceux de 20 pixels). La présentation des conditions a été contrebalancée entre les participants grâce à un carré latin. Pour chaque technique, les participants ont effectué la tâche de copier-coller huit fois pour les deux niveaux de difficulté, en commençant par le niveau Facile afin de se familiariser avec la tâche. La taille des boutons n'entrant pas en compte avec les MicroRolls (puisque'il n'y a pas de retour visuel), ceux-ci ont été testés dans les deux mêmes conditions pour les deux blocs consécutifs. Ceci se justifie par le fait qu'il aurait été gênant que les utilisateurs effectuent moins d'essais avec l'une des trois techniques, d'autant que l'une est nouvelle alors que les deux autres sont bien connues. Ceci a permis d'obtenir le même nombre d'essais pour chaque technique, et par conséquent une analyse de variance. Pour résumer le design expérimental était le suivant :

3 *Techniques* x 2 *Niveaux de difficulté* x 8 *Essais* = 576 copier/coller (40 minutes) par participant

### Résultats et discussion

Etant donné qu'un essai fini était un essai réussi (les erreurs étant corrigées au fur et à mesure), seul le temps de complétion de la tâche a été étudié. Celui-ci (FIG. 97) a été mesuré à partir du moment où l'utilisateur appuyait sur l'écran, jusqu'au moment où il le relâchait et que la cible avait été collée. Nous rappelons que la tâche impliquait au moins cinq opérations et que le déplacement de la vue pouvait demander plusieurs gestes "Drag". L'ANOVA montre des effets significatifs pour les *Techniques* ( $F_{2,22}=12.54$ ,  $p<.001$ ), les *Niveaux de difficulté* ( $F_{1,11}=24.74$ ,  $p<.001$ ) et l'interaction *Techniques x Niveaux de difficulté* ( $F_{2,22}=4.44$ ,  $p<.016$ ). Les tests de comparaison multiple (test post hoc de Tukey) montrent qu'en condition Facile (gros boutons), la BO (4.0s en moyenne) était significativement plus rapide que le MC (7.0s). Les MR s'intercalent entre les deux (5.4s) de façon non significative. Cependant pour la condition plus Difficile (petits boutons), les MR (5.8s) sont significativement plus rapides que la BO (8.1s) et que le MC (10.1s). On peut également noter que la différence du temps de complétion entre les deux blocs de MR est très petite et non significative, ce qui suppose l'absence d'un effet d'apprentissage puisque les utilisateurs étaient capables d'effectuer rapidement les MR dès le premier bloc.

L'ANOVA montre aussi un effet significatif des techniques sur le temps de déplacement de la vue (gestes "Drag") ( $F_{2,22}=6.27$ ,  $p<.004$ ). Ce temps est plus court avec la BO (0.68s) qu'avec les MR (1.07s) et le MC (1.17s). Une explication possible est que la position de placement de la cible, au moment de l'opération de coller, dépend des techniques. Pour comprendre cet effet nous avons calculé une ANOVA sur les abscisses des cibles lors de l'opération de coller. Nous avons effectivement observé un effet significatif ( $F_{2,22}=17,27$ ,  $p<.0001$ ). Les cibles sont collées

plus près du bord pour la BO (à 20 pixels du bord en moyenne) qu’avec le MC (43 pixels du bord en moyenne) et les MR (53 pixels du bord en moyenne). Nous pouvons supposer que les utilisateurs ont pris l’habitude de placer la cible moins près des bords avec les MR et le MC, afin de ne pas être gêné par ceux ci. Nous reviendrons sur ce point dans la seconde expérience.

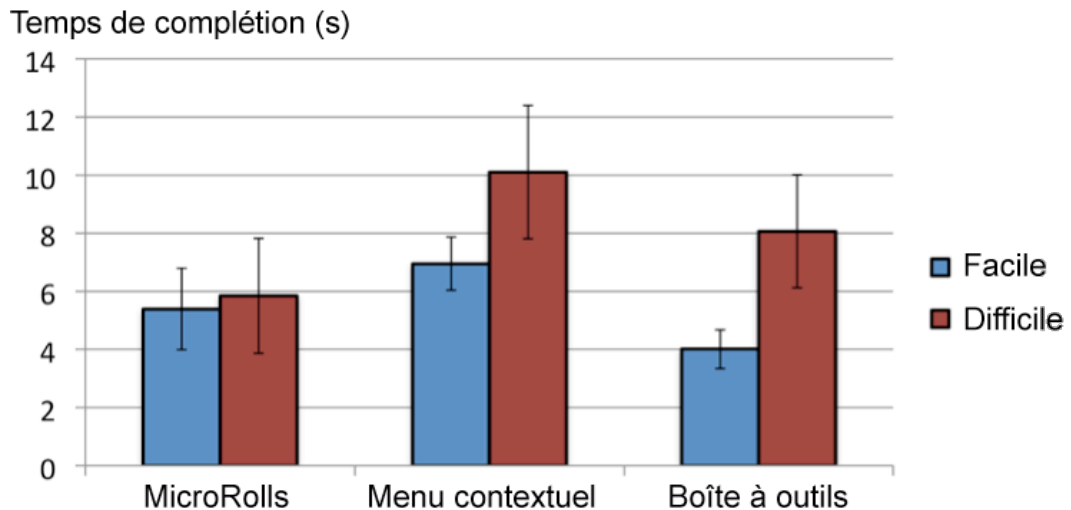


Figure 97. Temps total de complétion de la tâche (les barres représentent un intervalle de confiance de 95%).

Finalemnt, une ANOVA a été calculée sur le temps total de complétion sans le temps de déplacement de la vue. De même que précédemment, un effet significatif a été trouvé pour les *Techniques* ( $F_{2,22}=13.56, p<.001$ ), les *Niveaux de difficulté* ( $F_{1,11}=28.42, p<.001$ ) et l’interaction *Techniques x Niveaux de difficulté* ( $F_{2,22}=5.5, p<.01$ ). Les test de comparaison multiple donnent des résultats similaires au temps de complétion total : pour le niveau facile la BO (3.4s en moyenne) est plus rapide que le MC (5.9s). Les MR se situent entre les deux (4.4s) sans effet significatif pour les différencier des deux autres. Pour le niveau difficile MR (4.7s) est plus rapide que la BO (7.4s) et que le MC (8.8s).

Conformément à ce que l’on pouvait attendre, les performances de la barre à outils (BO) en condition Facile sont relativement bonnes, même si l’analyse ne nous permet pas de dire quelle technique, entre la barre à outils et les MicroRolls (MR), est la meilleure. Il faut cependant noter qu’une barre à outils aussi large n’est pas une solution optimale pour économiser l’espace écran (dans notre expérience, sa largeur correspondait à la largeur de l’écran, et sa hauteur correspondait à 1/5<sup>eme</sup> de la hauteur l’écran). Finalement, le résultat intéressant à noter est que les MicroRolls sont non seulement plus performants que le menu contextuel (MC), toutes conditions confondues (23% et 43% de temps gagné respectivement pour la condition Facile et Difficile), mais ils sont plus efficaces que la barre à outils (28% de temps gagné) en condition Difficile (petits boutons).

## 4.2 Expérience 2

Dans cette expérience, nous testons deux conditions supplémentaires. Les participants devaient également copier/coller une cible, mais celle-ci pouvait se trouver à différentes positions sur l'écran (FIG. 98.A) et mesurait 20x20 pixels au lieu des 60x60 pixels de l'expérience précédente. Cette condition correspond à un cas d'usage plus réaliste, par exemple assez proche des opérations à effectuer pour copier/coller un texte. Afin d'alléger l'expérience pour nos participants, nous n'avons considéré que la BO et les MR, le MC ayant montré des performances plus faibles que les deux autres techniques dans la première expérience.

### *Plan d'expérimentation*

**TÂCHE.** La tâche est la même que dans l'expérience précédente à part le nombre d'opérations demandées qui est le double du cas précédent (sans considérer les gestes "Drag" nécessaires à déplacer la vue). Que ce soit pour copier ou coller, les participants doivent :

1. Appuyer sur l'écran pour sélectionner une zone
2. Déclencher le mode de précision permettant de zoomer sur la zone sélectionnée (FIG. 98.B)
3. Appuyer sur l'écran pour sélectionner la cible dans la vue zoomée
4. Activer la commande de copier (ou coller selon le cas)

Les participants doivent effectuer toutes les étapes et ne peuvent pas sélectionner la cible sans zoomer, ceci afin d'obtenir des données homogènes. Trois commandes au lieu de deux doivent être utilisées. Le bouton "zoomer" de la BO ou un MR vers le haut permettent d'activer la commande de sélection précise. Cette commande, qui est inspirée par TapTap (voir le CHAPITRE 3), fonctionne comme suit :

- Pour la BO, l'utilisateur pointe un endroit de la vue qu'il souhaite zoomer, puis pointe la commande "zoomer". La vue zoomée de TapTap apparaît au centre de l'écran et l'utilisateur sélectionne alors la cible ainsi devenue assez large pour une interaction au pouce. La vue disparaît et l'utilisateur choisit une autre commande de la BO, telle que copier (ou coller selon les cas).
- TapTap a été combiné aux MR en remplaçant le premier tap par un MR vers le haut. La vue zoomée de TapTap apparaît au centre de l'écran, l'utilisateur effectue un second MR pour copier (ou coller selon les cas) et la vue disparaît. Cette technique que nous nommons *RollTap* a un avantage intéressant par rapport à la technique originale TapTap : le zoom est uniquement activé à la demande de l'utilisateur, lorsqu'il le juge nécessaire. Il peut par exemple sélectionner les larges cibles en tapant directement sur celles-ci ou activer RollTap lorsque les cibles sont trop petites.

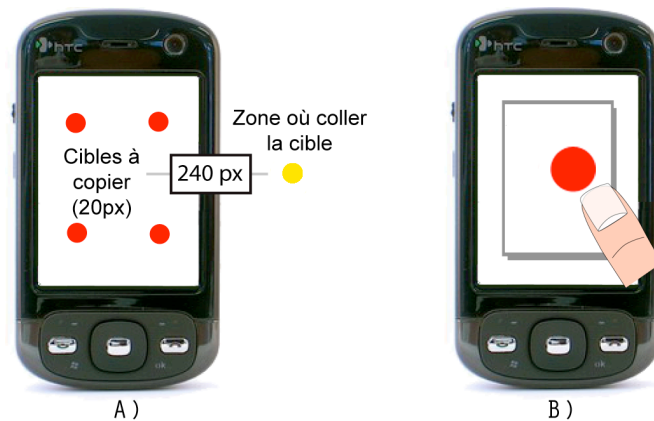


Figure 98. A) la tâche de copier/coller sur des petites cibles ; B) Mode de sélection précise (inspiré de TapTap).

**MATÉRIEL ET PARTICIPANTS.** Les techniques ont été implémentées en C# (.Net Compact Framework) et fonctionnaient sur Windows Mobile 5.0. L'expérience a été réalisée sur un HTC P3600 QVGA (320x240). 12 volontaires droitiers (dont 2 femmes), âgés de 23 à 31 ans, ont été recrutés dans notre institution et ont reçu une poignée de bonbons pour leur participation.

**DESIGN EXPÉRIMENTAL.** Nous avons conduit une analyse de variance à mesures répétées chez les mêmes sujets (within-subject). Les variables indépendantes sont les *Techniques* (MR, BO), les *Niveaux de difficulté* (Facile avec des boutons de 60 pixels ou 9mm, Difficile pour ceux de 20 pixels ou 3mm) et les *Zones des cibles* à copier (l'espace de l'écran sous la barre d'outils a été divisé en 4 zones d'égale surface et les cibles se trouvaient au centre de ces zones). La présentation des conditions a été contrebalancée entre les participants grâce à un carré latin. Le reste de l'expérience est similaire à l'expérience 1. En résumé, le design expérimental était le suivant :

2 *Techniques* x 2 *Niveaux de difficulté* x 4 *Zones de cibles* x 4 *Essais* = 768 copier/coller (40 minutes) par participant

### Résultats et discussion

Le temps de complétion a été mesuré de la même façon que dans l'expérimentation précédente. Cette expérience montre deux résultats intéressants. Le premier est que les MR sont plus performants que la BO ( $F_{1,11} = 10.60, p < .02$ ). Alors qu'il faut 12.06s en moyenne pour exécuter la tâche avec la BO, seulement 8.04s (soit 2/3 du temps) le sont pour les MR. L'interaction *Techniques x Niveaux de difficulté* ( $F_{1,11} = 21.15, p < .002$ ) montre un effet significatif qui est dû à la dépendance de la BO à la taille des boutons. Comme le montre la **FIGURE 20**, les MR sont environ deux fois plus rapides que la BO en condition difficile (15.71s contre 8.04s pour les MR). Ils ont des performances équivalentes à la version facile de la BO (8.41s), mais avec le notable avantage de ne pas consommer de place à l'écran.

Les mêmes tendances de performance ont été trouvées pour le temps de complétion sans le temps de déplacement de la carte. Ceci n'est pas surprenant puisque le temps de déplacement n'occupe qu'une petite proportion du temps total (1.16s et 0.97s en moyenne pour MR et BO, soit respectivement 14,4% et 8.0% du temps total de complétion).

Le second résultat intéressant est que les performances des MR sont plus dépendantes à la position de la cible sur l'écran que celles de BO, comme le révèle l'interaction *Technique x Zone des cibles* ( $F_{3,33} = 7.29, p < .002$ ). La **FIGURE 99** montre que les MR sont plus rapides sur la partie gauche que sur la partie droite de l'écran pour un droitier (6,5s en moyenne pour la partie droite et 9,6s pour la partie gauche). Ceci peut être expliqué par les contraintes morphologiques de la main qui impliquent des mouvements plus faciles du pouce dans cette partie de l'écran. Cependant les résultats de cette expérience montrent aussi que les MR peuvent être effectués correctement et rapidement quelle que soit la position de la cible.

Nos observations durant l'expérience montrent que les participants ont appris facilement les MR. Cette impression a aussi été observée par les données de l'expérience. Nous aurions en effet pu observer un effet d'apprentissage entre les deux blocs. Comme dans l'expérience 1, aucun effet n'a été observé (8.16s pour le premier bloc contre 7.93s pour le second, une différence non significative d'un point de vue statistique).

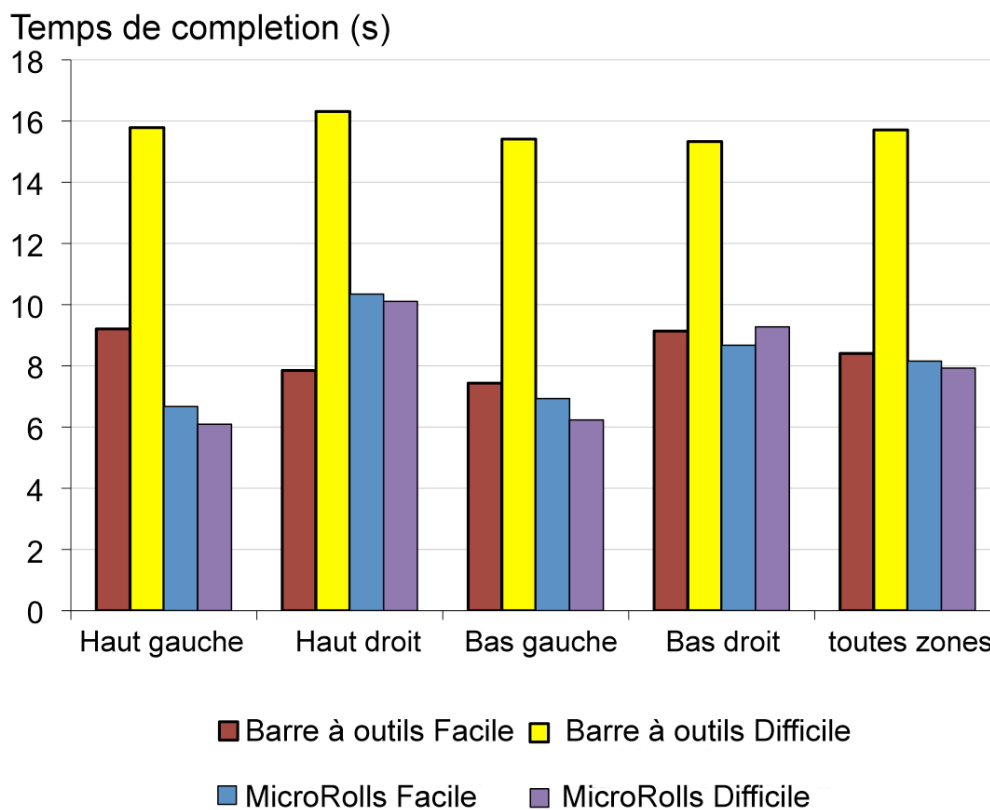


Figure 99. Temps total de complétion de la tâche.

### 4.3 Préférences utilisateurs

Après les deux expériences, plusieurs participants ont spontanément déclaré être fatigués lors de l'utilisation du MC et de la BO en condition difficile (petits boutons). Grâce au questionnaire qui a suivi les deux expérimentations, les participants ont pu donner leur ordre de préférence des techniques. Les MR ont obtenu le plus haut score. Nous avons aussi demandé à nos participants leurs opinions à propos de la rapidité, la précision, l'attrait, la simplicité, la facilité d'apprentissage sur une échelle de Likert. Bien que ces résultats de soient pas significatifs d'un point de vue statistique, nous les résumons sur la **FIGURE 100**.

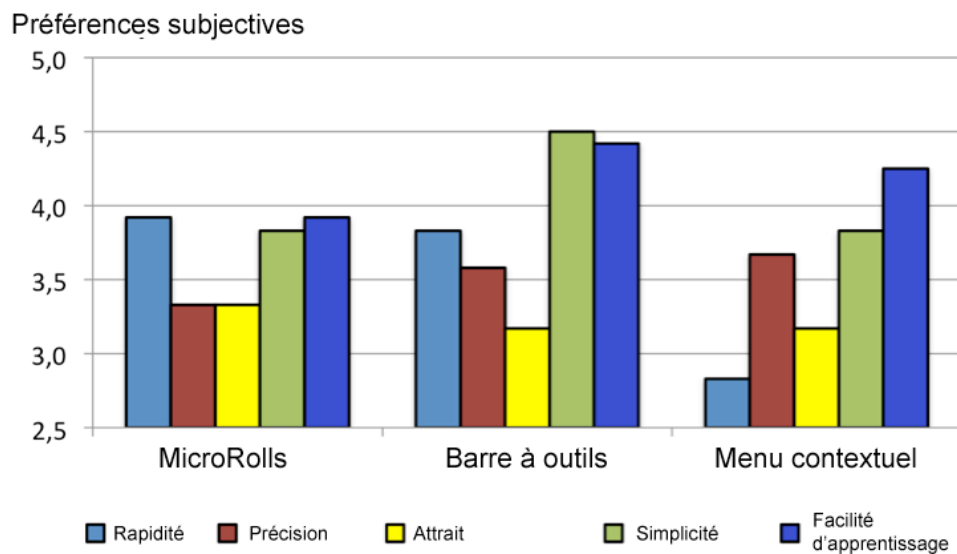


Figure 100. Résultats des questionnaires utilisateurs (les deux expériences confondues).

## 5. Intégration

Dans cette section, nous présentons des exemples d'intégration de nos techniques dans divers cas d'application. Nous avons déjà présenté RollTap et le RollMark menu et proposons maintenant deux autres possibilités.

### RollPod

RollPod est une idée d'application pour la gestion d'une bibliothèque musicale telle que iTunes. En utilisant les MicroRolls, RollPod permet d'activer certaines fonctionnalités utiles lorsque l'écran du dispositif n'est pas visible (eyes-free sélection), ce qui peut être le cas quand l'utilisateur manipule son dispositif alors qu'il est dans sa poche ou son sac. RollPod propose un ensemble de gestes pour activer les diverses commandes (**FIG. 101**) :



- Un MicroRoll vers le bas permet de lire un morceau de musique ou de mettre en pause
- Les MicroRolls vers la gauche ou la droite permettent d'avancer ou de reculer dans le morceau courant
- Un MicroRoll vers le haut permet de remonter à un niveau supérieur de l'arborescence de la bibliothèque musicale
- Les MicroRolls horaire et anti-horaire permettent d'augmenter ou de diminuer le volume. Ils seraient donc utilisés pour effectuer du contrôle continu comme les Control Menu (POOK 00) ou les Flow Menus (GUIMBRETIERE 00)
- Comme c'est le cas sur la mollette des iPods, les gestes "Drag" circulaires (autour de la molette) permettent de se déplacer dans la liste de musiques

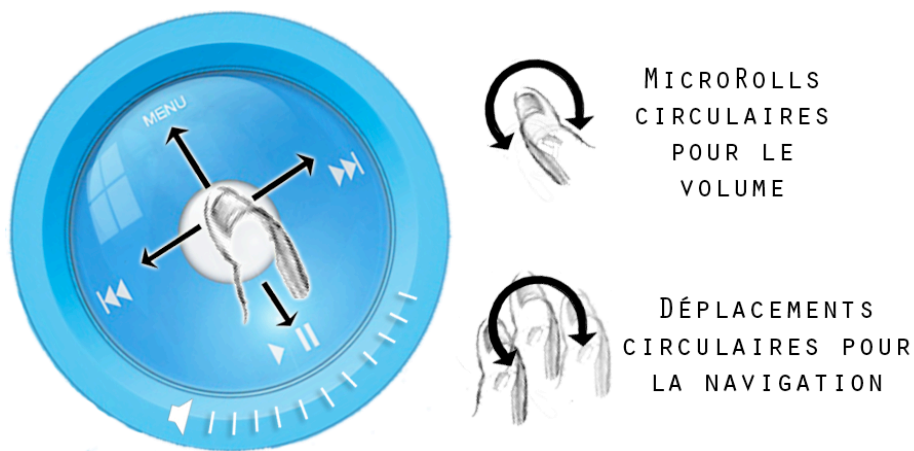


Figure 101. RollPod utilise les MicroRolls pour contrôler une bibliothèque musicale.

### RollColor

RollColor (FIG. 102) exploite les MicroRolls cardinaux pour faciliter la sélection de cases dans un tableau. A chaque MicroRoll cardinal correspond une couleur (rouge, vert, bleu, jaune) (FIG. 102.B) et les cases du tableau sont colorées par cinq couleurs (rouge, vert, bleu, jaune et gris) (FIG. 102.A). Lorsque l'utilisateur tape sur l'écran tactile et relève son doigt sans effectuer de MicroRolls, la case grise la plus proche de son doigt est sélectionnée. Mais si l'utilisateur effectue un MicroRoll vers la droite (MicroRoll jaune) par exemple, la case jaune la plus proche de son doigt dans le tableau est sélectionnée. Ce mécanisme peut aussi être utilisé pour la sélection dans de longues liste ou dans les menus. Dans les deux cas, il permet d'améliorer la précision du pointage de petites cibles.

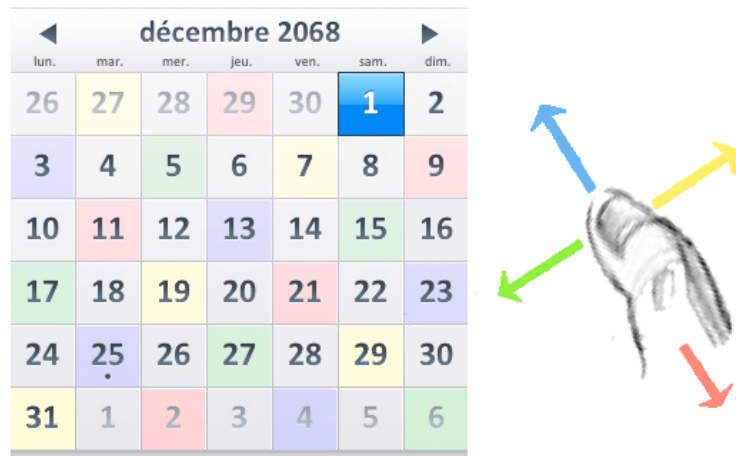


Figure 102. RollPod utilise les MicroRolls pour permettre la sélection de cases dans un tableau.

## 6. Conclusion

Dans cette étude nous avons proposé un nouveau type de gestes, les MicroRolls, qui exploitent la capacité de distinguer les mouvements de roulement de ceux de glissement du doigt sur l'écran tactile. Nous avons construit un algorithme de reconnaissance avancé, capable de différencier un vocabulaire de 16 gestes élémentaires (MicroRolls, gestes "Drag", "Swipes" et "Rubbing"). Nous avons ensuite montré comment les MicroRolls peuvent enrichir l'interaction sur les dispositifs mobiles en permettant d'accéder rapidement à des fonctionnalités fréquemment utilisées (comme le copier/coller, l'activation d'un menu contextuel, d'un mode de précision ou d'un quasimode comme le Drag & Drop). Puis nous avons conduit deux expériences ayant pour but de tester l'efficacité des MicroRolls dans ce cadre d'application. Ces expériences ont comparé les MicroRolls avec les deux techniques les plus souvent utilisées sur mobiles pour une tâche de copier/coller : la barre d'outils et le menu contextuel. Nous avons montré que les MicroRolls sont une alternative avantageuse à ces deux techniques :

- Les barres à outils permettent de compenser le manque d'états interactionnels des écrans tactiles. Comparées aux menus contextuels, elles permettent une sélection plus rapide mais consomment une place non négligeable à l'écran. C'est en particulier le cas lorsque les boutons sont larges, c'est à dire lorsqu'ils ont une taille adéquate pour la sélection au pouce. Nous avons vu que la diminution de la taille des boutons entraîne un sérieux problème de précision lors de la sélection. Les MicroRolls offrent alors deux avantages importants par rapport aux barres d'outils : non seulement ils sont sensiblement plus rapides mais ils ne consomment aucune place à l'écran.

- Les menus contextuels fournissent aussi une solution au manque de place car ils s'ouvrent sur demande de l'utilisateur. Leur performance est cependant limitée du fait du délai nécessaire à leur déclenchement et de la nécessité de pointer sur un item qui n'apparaît qu'après un délai : l'expérience 1 montre que les MicroRolls sont pratiquement deux fois plus rapides que les menus contextuels dans des conditions réalistes (items de 20 pixels). Les MicroRolls offrent donc une alternative efficace lorsque le nombre de commandes est limité. De plus, l'usage des MicroRolls n'interdit pas l'utilisation de menus linéaires contextuels si nécessaire (typiquement lorsque le nombre de commandes dépasse la capacité offerte par les MicroRolls), car ceux-ci peuvent être déclenchés par un MicroRoll, par exemple un MicroRoll horaire.

En résumé, les MicroRolls sont une alternative intéressante à ces deux techniques puisqu'ils ne consomment pas de place à l'écran (contrairement aux barres à outils) et n'imposent pas le recours à un délai temporel (contrairement aux menus contextuels).

Nous avons aussi montré l'intérêt de combiner les MicroRolls à d'autres techniques. RollTap combine TapTap et les MicroRolls pour permettre à l'utilisateur de déclencher TapTap sur demande. Le RollMark menu combine les Marking Menus et les MicroRolls. Enfin nous avons aussi proposé deux autres techniques (RollPod et RollColor) qui illustrent également le fait que les MicroRolls peuvent être utilisés de nombreuses manières.

# Gestes incarnés



"66% de sondés (Etats-Unis) ont déjà utilisé leur téléphone dans un supermarché, 45% dans les transports, 63% en voiture et 38 % dans leur salle de bain"

Letstalk annual cell phone etiquette survey (2008)

<b>1. Introduction</b>	<b>151</b>
<b>2. Etat de l'art</b>	<b>151</b>
2.1 Capteurs de mouvements	152
2.2 Gestes fluides	155
2.3 Gestes impulsifs	154
2.4 Synthèse de l'état de l'art	157
<b>3. TimeTilt</b>	<b>157</b>
3.1 Mode lenticulaire	158
3.2 Mode superposition	159
3.3 Réalisation	160
<b>4. Evaluation</b>	<b>162</b>
4.1 Plan d'expérimentation	162
4.2 Recherche des primitives	163
4.3 Reconnaissance	164
<b>5. Intégration</b>	<b>165</b>
<b>6. Conclusion</b>	<b>168</b>

## 1. Introduction

Jusqu'à présent nous avons étudié les techniques d'interaction basées sur l'exécution de gestes en deux dimensions sur un écran tactile. Dans ce chapitre nous élargissons notre travail de recherche en nous intéressant aux gestes effectués en trois dimensions. Ceux-ci peuvent être détectés par des capteurs, lesquels sont de plus en plus présents et populaires sur les dispositifs mobiles. Par exemple l'orientation d'un iPhone contrôle le passage entre deux modes d'affichage (portrait ou paysage) et le secouement de l'appareil annule un copier coller. Ces gestes offrent l'opportunité de créer des techniques d'interaction adaptées à la petite taille des mobiles car l'utilisateur peut facilement, avec une seule main, effectuer des mouvements dans l'espace. Ces techniques sont de plus compatibles avec les techniques d'interaction impliquant des gestes en deux dimensions puisqu'elles n'exploitent pas les mêmes ressources matérielles. Ces nouveaux gestes offrent donc une possibilité supplémentaire d'enrichir la bande passante interactionnelle des dispositifs mobiles.

Ce chapitre présente une étude préliminaire de ces nouveaux gestes qui nous permet d'ouvrir nos travaux de recherche à d'autres possibilités d'interaction. Bien qu'il existe de nombreuses technologies de capteurs, nous nous sommes principalement intéressé aux techniques utilisant les capteurs de mouvements les plus couramment rencontrés sur les dispositifs mobiles (accéléromètres, gyromètres et caméras vidéo). Ces techniques sont à la base des interactions "incarnées" (FISHKIN 00) (voir le CHAPITRE 2), c'est-à-dire qu'elles tirent parti des compétences que nous utilisons dans le monde réel (secouer, déplacer, lancer, tourner etc.).

Nous détaillons tout d'abord brièvement les différences entre les technologies de capteurs de mouvements puis exposons l'état de l'art. Nous présentons ensuite TimeTilt (ROUDAUT, BAGLIONI 09), une technique exploitant les gestes en trois dimensions pour faciliter la navigation entre plusieurs vues. Nous exposons ensuite les résultats d'une expérience de collecte de gestes ainsi que des exemples d'application et d'évolution de TimeTilt avant de conclure. TimeTilt a été réalisée en collaboration avec Mathias Baglioni, doctorant effectuant également ses travaux de thèse en partenariat avec Bell Labs Alcatel-Lucent.

## 2. État de l'art

En nous basant sur la classification de (BAGLIONI 09) nous présentons tout d'abord les principaux capteurs de mouvements existants puis diverses techniques d'interaction qui reposent sur l'utilisation de ces capteurs. Nous verrons que les gestes mis en œuvre dans ces techniques d'interaction peuvent être de deux types : impulsifs ou fluides.

## 2.1 Capteurs de mouvements

### Accéléromètre et gyromètre

Un accéléromètre est un capteur qui mesure l'accélération d'un dispositif sur deux ou trois de ses axes. Par exemple, si le dispositif est à plat sur une surface, la composante en z du vecteur accélération est d'environ  $-9.8 \text{ ms}^{-2}$  (accélération moyenne de la gravité sur terre) alors que celles en x et y sont nulles (FIG. 103). Dans cette même configuration horizontale, si le dispositif est jeté vers le sol, la composante en z augmente.

Les accéléromètres peuvent servir d'inclinomètres pour déterminer l'orientation de l'écran par rapport au sol. Contrairement aux gyromètres qui mesurent la vitesse de rotation d'un dispositif autour d'un ou plusieurs de ses axes, les accéléromètres ne peuvent détecter les rotations de l'appareil que dans certaines positions (si deux axes sont parallèles au sol durant la rotation, les composantes du vecteur d'accélération ne varient pas). Ils ne mesurent ni la position (comme le fait un GPS), ni la direction (comme le permet une boussole) et ils ne peuvent pas détecter une translation de l'appareil à vitesse constante (l'accélération étant nulle). Enfin, la dérivée de l'accélération (variation d'accélération, secousse ou "jerk") permet de distinguer les accélérations fortes et brèves (comme un secouement) d'une accélération lente et douce du dispositif.



Figure 103. Les composantes du vecteur accélération d'un accéléromètre à trois axes dans le référentiel du dispositif mobile.

### Caméra

La caméra, présente sur la majorité des dispositifs mobiles, peut être détournée de son utilisation initiale pour détecter les mouvements. L'analyse des différences entre deux images successives permet de reconnaître des mouvements de *translation* et de *rotation* de l'appareil. Le procédé est basé sur l'observation des déplacements entre les pixels identiques aux deux images, ce qui nécessite une caméra dont la fréquence d'échantillonnage est suffisante. Dans le cas contraire, les mouvements trop rapides ne sont pas détectés. De plus cette méthode n'est pas utilisable sur fonds unis et montre des faiblesses en condition de faible luminosité. Cette méthode de détection a aussi l'inconvénient d'être assez coûteuse en énergie.

La caméra peut également être associée à des sources lumineuses infrarouges comme dans la console de jeux Nintendo Wii. Grâce à la caméra placée sur la manette Wiimote, ainsi que deux diodes infrarouges positionnées près de la télévision, le système est capable de connaître la position de la manette par rapport aux diodes (dans la mesure où la caméra n'est pas trop éloignée des diodes). Dans (LEE 08), quatre diodes infrarouges sont placées sur les coins d'une feuille. L'analyse de la position des diodes permet de mesurer l'inclinaison et la taille de la feuille afin de projeter le contenu d'une page de journal dans la bonne orientation (FIG. 104).

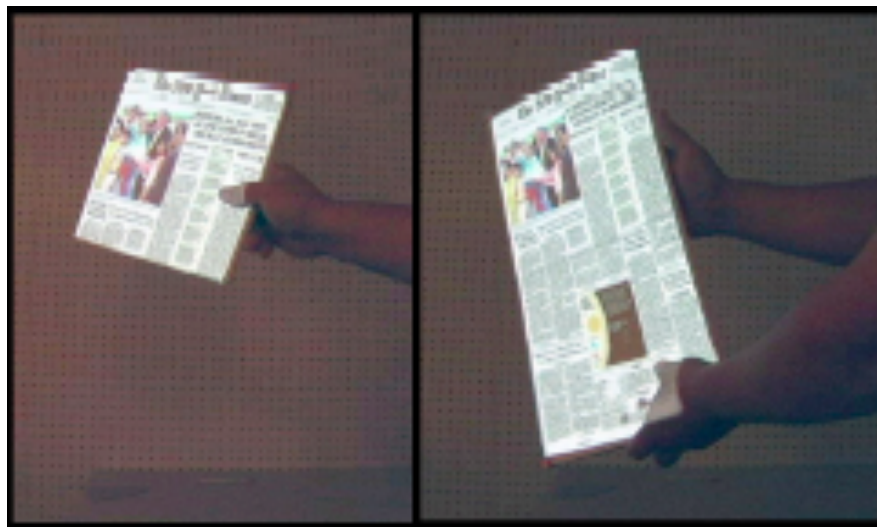


Figure 104. L'inclinaison et la taille d'une feuille de papier sont reconnues grâce à des diodes infrarouges placées aux quatre coins de la feuille. La projection de la page de journal suit donc les mouvements de la feuille.

Ayant explicité le fonctionnement des principaux capteurs de mouvements, nous présentons ci-après les deux types de gestes mis en œuvre dans les techniques d'interaction existantes.



## 2.2 Gestes impulsifs

Les gestes impulsifs mettent en jeu de fortes accélérations du dispositif dans un court laps de temps, ce qui est par exemple le cas lorsque l'utilisateur effectue un mouvement vif vers le bas avec le dispositif. La détection de ce type de gestes est possible grâce au calcul de variation d'accélération (dérivée de l'accélération ou secousse) et nécessite donc un accéléromètre.

Plusieurs gestes sont ensuite discriminables grâce à des algorithmes de reconnaissance comme les secousses. Par exemple dans (HARRISON 98) elles servent à déclencher des commandes telles qu'effacer un contenu. Cette idée a notamment été reprise dans DrawR (application iPhone) où l'utilisateur secoue son dispositif pour effacer un dessin comme le faisaient les tables de dessin (Etch-A-Sketch) à deux molettes pour enfants. Chucking (HASSAN 09) (FIG. 105) est une technique permettant de déplacer des documents d'un dispositif mobile vers un ordinateur conventionnel. L'utilisateur sélectionne un document sur l'écran du dispositif mobile, puis effectue un geste vif de rotation du poignet vers la gauche ou la droite pour "jeter" le document vers la deuxième surface (la métaphore utilisée est celle de la distribution de cartes à jouer). Le lecteur de musique Sony Ericsson W508 change de morceaux lorsque l'utilisateur fait un mouvement vif sur la droite ou la gauche avec l'appareil. Une autre étude propose de créer des séquences de gestes impulsifs directionnels pour coder des mots de passe (CHONG 09).



Figure 105. Chucking permet de déplacer une image d'un dispositif mobile vers un ordinateur grâce à un geste impulsif.

De fortes variations de l'accélération peuvent aussi être produites lors de chocs sur le dispositif. Par exemple, Hinckley et al. (HINCKLEY 03) (FIG. 106.A) proposent de coupler deux écrans d'ordinateurs en choquant les dispositifs l'un contre l'autre. Dans (LINJAMA 04) (FIG. 106.B) l'utilisateur tape avec ses doigts sur les cotés du dispositif pour déplacer une bille représentée à l'écran.



Figure 106. A) Deux écrans sont couplés en les choquant l'un contre l'autre ; B) l'utilisateur contrôle une bille sur l'écran en tapant sur les cotés du dispositif.

### 2.3 Gestes fluides

Contrairement aux mouvements impulsifs, les mouvements fluides mettent en jeu des variations d'accélération ou de vitesse de faible amplitude et étalées sur de plus longues durées, ce qui est par exemple le cas lorsque le dispositif est incliné progressivement. Dans ce cas, la variation d'accélération est constante au cours du temps, ce qui permet de discriminer ces gestes des mouvements impulsifs. Les translations et rotations peuvent être détectées grâce à une caméra ou un accéléromètre (dans certaines conditions comme nous l'avons vu à la section "capteurs de mouvements"), alors que seules les rotations peuvent être reconnues avec un gyromètre.

Plusieurs travaux ont proposé d'utiliser l'inclinaison du dispositif pour contrôler des fonctionnalités comme le changement du mode d'affichage (portrait ou paysage) (HINCKLEY 00). L'inclinaison peut aussi servir à naviguer dans un menu (OAKLEY 09) ou une liste de contacts (FISHKIN 00). Elle est aussi utilisée pour contrôler le volume de l'appareil dans le lecteur de musique Sony Ericsson W508. Ces exemples n'utilisent qu'un axe d'inclinaison (haut/bas) alors que dans Rock'nScroll (BARTLETT 00) l'utilisateur peut aussi incliner le dispositif de gauche à droite pour naviguer dans une image. Enfin MotionLens (SOCIÉTÉ REALEYES) (FIG. 107.A) est une technique permettant de naviguer dans une carte grâce à des translations du dispositif (dans les directions cardinales pour déplacer la carte ou d'avant en arrière pour changer l'échelle). Dans le même esprit, Peephole display (YEE 03) (FIG. 107.B) permet à l'utilisateur de dessiner sur une surface plus large que l'écran de son dispositif mobile en le translatant.

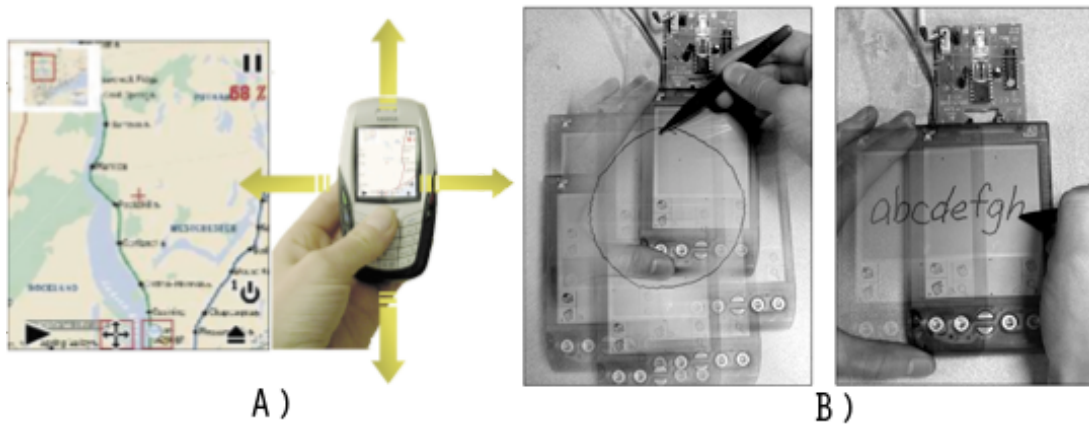


Figure 107. A) Dans MotionLens, les translations du dispositif permettent de naviguer dans une carte ou de changer d'échelle ; B) Dans Peephole Display, les translations du dispositif permettent d'écrire ou de dessiner sur une plus grande surface que l'écran.

Dans 3D Eyes Tracking (proposé la société TAT <http://www.tat.se>) (FIG. 108.A), l'orientation du dispositif permet de simuler une profondeur dans l'écran : comme dans le monde réel, lorsque nous changeons notre point de vue en regardant un objet la perspective change. Enfin TiltText (WIGDOR 03) (FIG. 108.B) est une variante du clavier T12 : à chaque touche correspond 4 lettres et un chiffre. Si l'utilisateur appuie sur la touche il sélectionne le chiffre, mais s'il incline en même temps le dispositif dans une direction, il accède aux lettres (en majuscule ou minuscule selon le degré d'inclinaison).

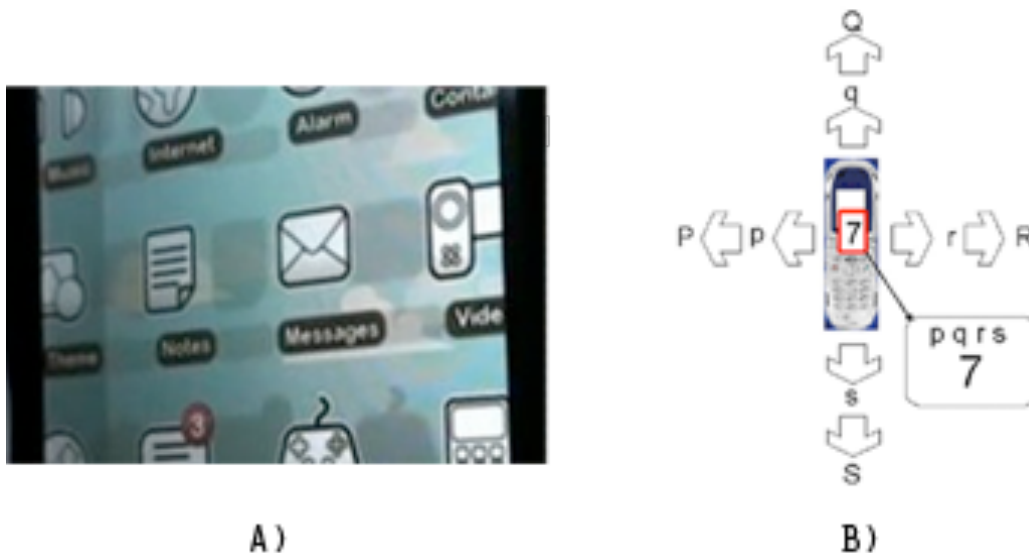


Figure 108. A) L'inclinaison du dispositif permet de simuler de la profondeur dans l'écran ; B) TiltText permet de sélectionner des lettres en appuyant sur une touche puis en inclinant le dispositif dans diverses directions.

## 2.4 Synthèse de l'état de l'art

Nous avons présenté deux types de gestes mis en œuvre dans les techniques d'interaction utilisant des capteurs de mouvements. Les gestes impulsifs sont caractérisés par une forte accélération du dispositif dans un laps de temps court. Une propriété importante de ces gestes est qu'ils sont facilement différenciables de ceux que l'utilisateur peut effectuer dans la vie de tous les jours. Par exemple secouer son dispositif n'est pas un geste habituel. Cependant, leur utilisation nécessite des techniques de reconnaissance et des capteurs relativement précis, ce qui peut poser des problèmes d'implémentation avec le matériel grand public usuel.

Les gestes fluides sont caractérisés par des variations d'accélération de faible amplitude et étalées sur d'assez longues durées. Ils ont l'avantage de permettre à l'utilisateur de contrôler de manière continue (ou semi-continue) une valeur comme lors d'un zoom sur une image ou de la sélection d'items d'un menu. Un inconvénient majeur de ces gestes est qu'ils sont difficiles à différencier de ceux que l'utilisateur peut réaliser dans son environnement quotidien (comme sortir le téléphone de sa poche). Par exemple, avec la technique Rock'nScroll, l'interaction par inclinaison est le mode par défaut : rien n'est prévu pour permettre à l'utilisateur d'indiquer un passage dans ce mode, ce qui peut entraîner des exécutions involontaires de commandes. Les fonctionnalités associées à ces gestes nécessitent donc une activation sur demande (via l'appui sur un bouton par exemple) pour éviter les confusions avec les gestes naturels de l'utilisateur.

## 3. TimeTilt

Nous avons mis en œuvre les deux types de gestes présentés précédemment pour concevoir une nouvelle technique d'interaction, TimeTilt, qui facilite le basculement entre plusieurs vues. TimeTilt peut notamment servir à naviguer dans un ensemble d'images, à passer d'une fenêtre à une autre ou encore à visualiser plusieurs niveaux d'échelle d'une carte. Les vues sont représentées selon une stratégie de multiplexage temporel plutôt que spatial (où les représentations des vues sont toutes représentées au même endroit mais dans un temps différent). L'utilisateur peut atteindre les vues grâce à deux modes d'interaction :

- Le **mode lenticulaire** (FIG. 109) : il est basé sur l'inclinaison (mouvement fluide) autour de l'axe x du référentiel du dispositif. Il permet, via une métaphore de cartes lenticulaires, de naviguer dans une liste de vues puis d'en choisir une. Ce mode est activable sur demande par un tap avec l'index à l'arrière du dispositif (mouvement impulsif) pour éviter les confusions avec les gestes naturels de l'utilisateur.

- Le **mode superposition** (FIG. 109) : il est basé sur des gestes vifs (mouvements impulsifs) vers l'avant ou l'arrière avec le dispositif et permet de basculer rapidement entre plusieurs vues superposées les unes sur les autres. Ce mode est activé par défaut, les gestes étant distincts de ceux que l'utilisateur peut effectuer dans le monde réel.

Nous explicitons ci-après les deux modes d'interactions de TimeTilt et leurs avantages, puis nous abordons les points techniques impliqués dans leur réalisation.



Figure 109. TimeTilt propose deux modes d'interaction et se base sur l'exécution de quatre gestes (fluides ou impulsifs).

### 3.1 Mode lenticulaire

#### Principe

Pour interagir avec TimeTilt en mode lenticulaire : 1) l'utilisateur tape avec l'index à l'arrière du dispositif (FIG. 110.A) ; 2) un aperçu de la vue courante apparaît au centre de l'écran ; 3) en inclinant le dispositif vers l'avant ou vers l'arrière, l'utilisateur navigue dans la liste des vues proposées grâce à leurs aperçus qui défilent un à un (FIG. 110.B) ; 4) lorsque l'utilisateur trouve une vue qui l'intéresse, il appuie sur l'écran tactile pour la faire apparaître au premier plan (ou appuie en dehors de l'aperçu pour annuler l'opération).

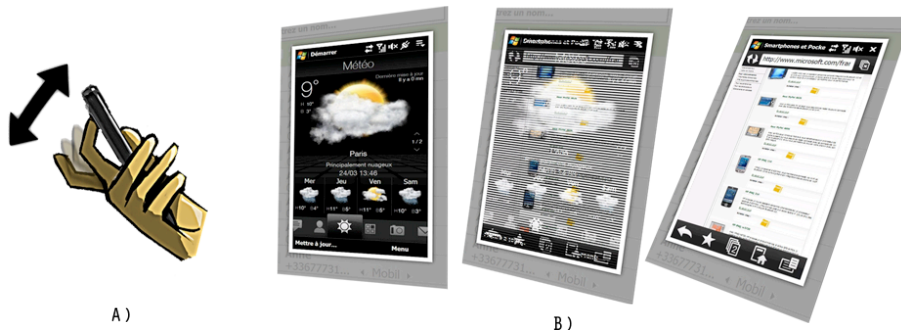


Figure 110. A) TimeTilt s'active par un tap à l'arrière du dispositif. B) Un aperçu de la vue courante apparaît au centre de l'écran. Une animation permet de passer progressivement d'un aperçu à un autre.

### Avantages

Le défilement des vues est basé sur la métaphore des cartes lenticulaires. Utilisées dans les objets publicitaires ou les jouets, celles-ci permettent à l'utilisateur, en changeant son angle de vue (en inclinant la carte), de visualiser successivement plusieurs images imprimées sur une seule et même surface. Cette métaphore est retranscrite dans TimeTilt par une animation (FIG. 110.B) qui transforme progressivement une vue en une autre au fur et à mesure de l'inclinaison grâce à un effet de transparence. Cette métaphore a d'une part l'avantage d'amener l'utilisateur à effectuer le geste d'inclinaison approprié (dans l'axe z du référentiel), mais rend aussi l'interaction ludique (l'utilisateur pouvant avoir l'impression de "retourner dans le passé" pour retrouver son mobile tel qu'il était quelques minutes auparavant).

Nous avons vu dans l'état de l'art que les gestes fluides risquent d'être confondus avec les gestes de la vie courante. Le mode lenticulaire de TimeTilt est donc activé grâce à un geste impulsif : l'utilisateur tape avec l'index à l'arrière du dispositif. Ce mécanisme a deux avantages : d'une part il permet de déclencher le mode lenticulaire de TimeTilt sur demande sans nécessiter de bouton dédié, d'autre part, il utilise les doigts placés à l'arrière du dispositif lesquels sont généralement inexploités. Plusieurs études ont d'ailleurs ajouté des outils à l'arrière du dispositif : un joystick dans (CHAU 06) et une surface tactile dans (BAUDISCH 09).

Le mode 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. Cet aspect peut être par exemple utile pour comparer des vues représentant différents niveaux de zoom sur une carte.

## 3.2 Mode superposition

### Principe

Pour interagir en mode "superposition", l'utilisateur fait un geste vif avec le dispositif (FIG. 111) pour basculer à la vue précédente ou suivante de la liste ou de l'ensemble manipulé (respectivement en effectuant le geste vers l'arrière ou vers l'avant). Il n'y a pas d'aperçus, la vue étant directement affichée en plein écran, ce qui permet à l'utilisateur de basculer rapidement entre deux vues. Les vues sont "superposées" les unes sur les autres et sont organisées selon une liste circulaire.



Figure 111. Les mouvements impulsifs vers l'avant ou vers l'arrière du dispositif permettent de basculer entre deux vues.

### Avantages

Le mode superposition de TimeTilt permet d'atteindre rapidement une vue précédemment ouverte et de retourner à la vue courante. Ce mode peut en particulier être utile pour naviguer dans les fenêtres ouvertes sur le dispositif : lorsque l'utilisateur effectue un appel téléphonique, il peut rapidement atteindre son répertoire et dicter à son interlocuteur un numéro de téléphone, puis revenir à la fenêtre d'appel. Ce mode permet également, comme dans le mode lenticulaire, 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.

### 3.3 Réalisation

TimeTilt est basée sur l'utilisation d'un accéléromètre à trois axes qui transmet les données à intervalles réguliers de 50 ms. Elle mesure les variations d'accélération au cours du temps pour distinguer les gestes impulsifs des gestes fluides comme nous allons le voir. Une fois différencié, les gestes sont analysés séparément : le calcul inclinaison de l'appareil sert lors de gestes fluides et un algorithme de reconnaissance est utilisé pour reconnaître les différents types de gestes impulsifs.

#### Gestes fluides

Pour détecter l'inclinaison du dispositif autour de l'axe x (FIG. 102), nous utilisons les composantes en z et en y fournies par le capteur (FIG. 112.A), lesquelles permettent de calculer de manière précise l'angle formé entre l'axe y du dispositif et la verticale.

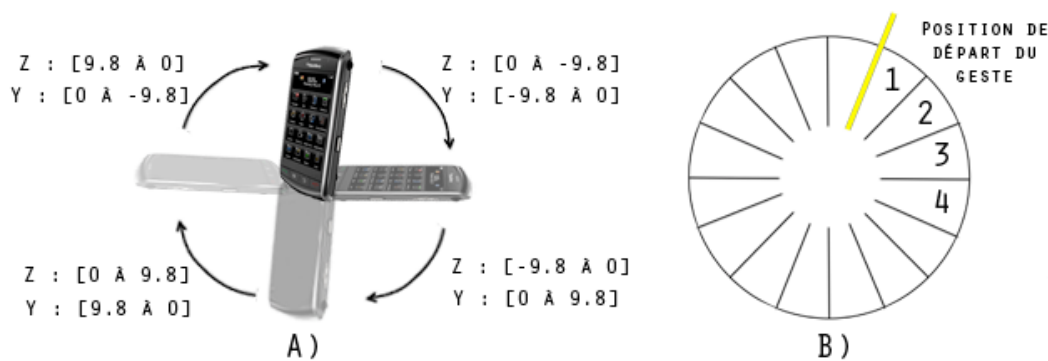


Figure 112. A) Intervalles de valeur des composantes en z et y du vecteur d'accélération lorsque le dispositif tourne autour de l'axe x ; B) Découpage en secteurs réguliers pour 4 vues.



Lorsque l'utilisateur tape à l'arrière du dispositif pour activer le mode lenticulaire de TimeTilt, l'orientation courante du dispositif est calculée et mémorisée. Cette calibration permet à l'utilisateur de déclencher le mode lenticulaire dans une position favorable lui permettant de bien voir l'écran. Cette position sert de point de départ pour définir des secteurs réguliers de rotation autour de l'axe x (FIG. 112.B). La taille des secteurs correspond à la division d'un angle de  $90^\circ$  par le nombre de vues présentes dans la liste. Selon le secteur sélectionné, l'aperçu suivant ou précédent (suivant le sens de l'inclinaison) est affiché. Le contrôle se fait donc en position et non en vitesse (la vitesse de défilement serait liée au degré d'inclinaison du mobile), ce qui améliore la précision du contrôle (OAKLEY 09). Enfin, l'animation lenticulaire est réalisée comme suit : quatre images sont mémorisées entre deux vues successives. Ces images, représentant un mélange progressif du contenu des deux vues afin de créer un effet de transparence, sont ensuite affichées un peu avant le changement de vue.

Le choix d'un angle de  $90^\circ$  pour calculer la taille des secteurs permet à l'utilisateur d'atteindre les vues de manière confortable : il n'est pas nécessaire d'incliner davantage la main et, dans le cas contraire, l'utilisateur revient à la première vue de la liste. Cependant, ce découpage en secteurs limite le nombre de vues aisément contrôlables. Nous avons estimé grâce à des tests informels qu'il est difficile pour l'utilisateur de contrôler le défilement et de s'arrêter précisément sur l'une d'entre elles avec plus de six vues dans la liste. Une récente étude (RAHMAN 09) a cependant présenté des résultats qui suggèrent que l'on pourrait augmenter le nombre de vues. Les auteurs ont montré que les mouvements de flexion/extension de la main peuvent s'étendre sur une plage de  $105^\circ$  (FIG. 113.A), ce qui permettrait d'augmenter la taille des secteurs. D'autre part, un découpage en secteurs astucieux peut améliorer le contrôle. Les auteurs présentent notamment deux méthodes que nous pourrions adapter à TimeTilt :

- Le découpage sigmoïde (FIG. 113.B) : une fonction sigmoïde ( $y=1/(1+e^{-x})$ ) crée des secteurs plus larges au centre d'un demi cercle et plus petits aux extrémités
- Le découpage quadratique (FIG. 113.C) : une fonction quadratique ( $y=x^2$ ) crée des secteurs plus petits au centre et plus larges aux extrémités

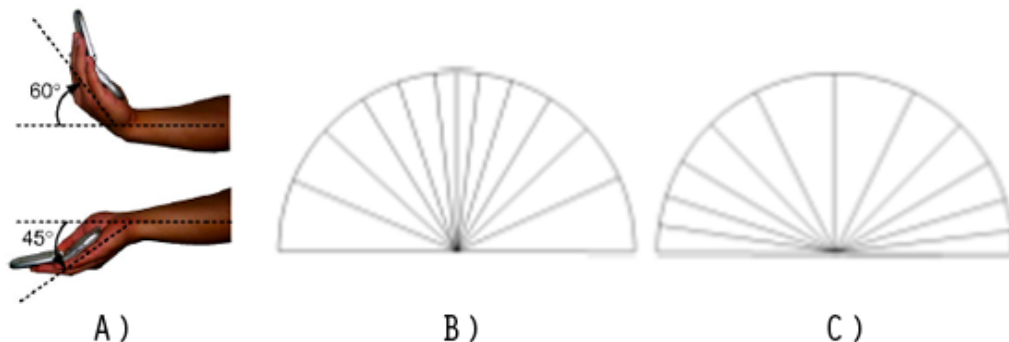


Figure 113. A) Les mouvements de flexion/extension de la main peuvent s'étendre sur une plage de  $105^\circ$  ; B) Découpage sigmoïde ; C) Découpage quadratique.



### Gestes impulsifs

Pour détecter les gestes impulsifs nous utilisons la variation d'accélération au cours du temps (dérivée de l'accélération) selon l'axe z. Cette méthode comporte plusieurs avantages : tout d'abord la variation d'accélération en z 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 gestes comme suit : si la variation d'accélération dépasse un certain seuil (plus ou moins  $2 \text{ ms}^2$ ) 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 détaillé dans la section suivante.

## 4. Evaluation

Dans cette partie nous présentons les résultats d'une expérience qui montre qu'il est possible de discriminer les trois différents gestes impulsifs utilisés dans TimeTilt (geste vif vers l'avant, geste vif vers l'arrière et tap à l'arrière du dispositif). L'expérience vise à collecter un ensemble de gestes afin de construire un algorithme d'apprentissage efficace.

### 4.1 Plan d'expérimentation

Les trois gestes impulsifs de TimeTilt peuvent être réalisés de différentes manières selon les utilisateurs, ce qui peut induire des variations et gêner la reconnaissance. C'est pourquoi nous avons demandé aux participants de les exécuter de deux manières différentes :

- Taps à l'arrière du dispositif : soit en relâchant le doigt à la fin du geste, soit en gardant le doigt appuyé contre le dispositif à la fin du geste
- Mouvements vifs vers l'arrière : soit en appliquant une faible force, soit en appliquant une force plus vive
- Mouvements vifs vers l'avant : avec deux intensités de force comme précédemment.

**TÂCHE.** La tâche consistait à effectuer les 3 gestes précédents de 2 manières différentes. Les participants tenaient le dispositif avec leur main dominante. Chacun des gestes a été répété 6 fois au total. Chaque utilisateur a effectué l'expérience avec un observateur lui indiquant le geste à exécuter. L'utilisateur devait attendre un certain laps de temps en maintenant le dispositif dans une position stable avant de commencer le geste suivant, ceci afin de faciliter l'analyse des données.

**MATÉRIEL ET PARTICIPANTS.** Les techniques ont été implémentées en C# (.Net Compact Framework) et fonctionnaient sur Windows Mobile 6.0. L'expérience a été réalisée sur un HTC HD (800x480). 8 volontaires (dont 2 femmes) ont été recrutés dans notre institution et ont reçu une poignée de bonbons pour leur participation.

**DESIGN EXPÉRIMENTAL.** Un carré latin a été utilisé pour contrebalancer l'ordre des gestes entre les participants. Ceux-ci se sont entraînés 5 minutes avant de commencer l'expérience. En résumé le design expérimental était le suivant :

3 Gestes x 2 Façons de les effectuer x 6 Répétitions = 36 gestes (10 minutes) par participant

## 4.2 Recherche des primitives

La **FIGURE 114** représente la variation d'accélération sur l'axe z au cours du temps lorsque l'utilisateur effectue plusieurs gestes. Nous pouvons observer d'une part une différence importante entre un geste fluide d'inclinaison (**FIG. 114.A**) et un geste impulsif de secousse du dispositif (**FIG. 114.B**). D'autre part, les trois types de gestes ciblés dans cette étude présentent aussi des différences de tracés (sens, durée amplitude et nombre de pics visibles) sur les graphiques comme on peut l'observer sur les **FIGURES 114.C.D.E**.

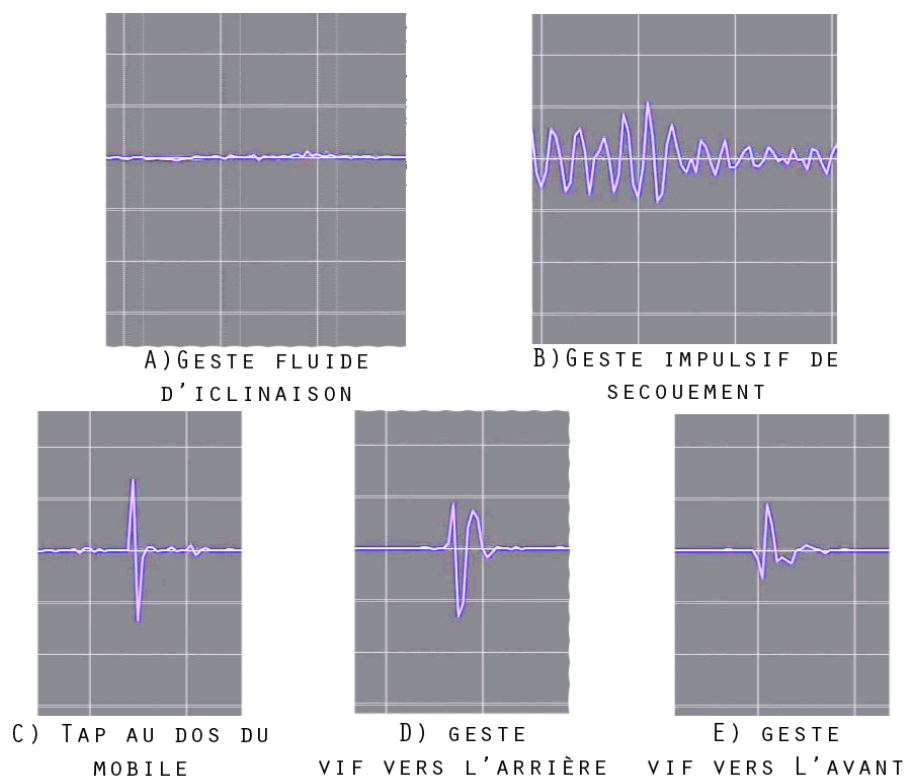


Figure 114. Variation de l'accélération sur l'axe z en fonction du temps pour différents gestes.

Après une analyse plus approfondie des données, nous avons développé un module de reconnaissance basé sur un algorithme d'apprentissage. Son fonctionnement est le suivant : 1) une base de gestes étiquetés est mémorisée ; 2) à chaque geste sont associées 5 primitives inspirées de l'algorithme de Rubine (RUBINE 91) ayant des poids différents ; 3) lorsqu'un nouveau geste doit être reconnu, l'algorithme calcule une distance<sup>10</sup> entre le vecteur de primitive du nouveau geste et ceux des gestes de la base d'apprentissage ; 4) l'algorithme détermine alors les K plus proches voisins (dans notre cas K = 3), c'est à dire les K gestes ayant une distance euclidienne minimale avec le geste à reconnaître ; 5) Il retourne ensuite le type du geste dominant parmi les K. Les primitives utilisées pour caractériser les gestes ainsi que les poids de ces primitives ont été choisies empiriquement et sont les suivantes :

- L'amplitude maximale entre deux points consécutifs (poids : 0.1)
- La durée 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 (poids : 0.1)
- Le signe du premier pic de variation d'accélération (poids : 0.8)

### 4.3 Reconnaissance

Pour tester l'efficacité de notre algorithme, nous avons utilisé la moitié des gestes récoltés comme base d'apprentissage (les essais pairs), alors que l'autre moitié a permis de calculer le taux de reconnaissance moyen, dont la valeur est de 92,3%. La FIGURE 115 montre la matrice de confusion que nous avons obtenue pour ces trois gestes. Les taps à l'arrière du dispositif sont les plus reconnus avec 93.8% de réussite et les gestes impulsifs vers l'avant et l'arrière sont reconnus à hauteur de 91.7%. Les résultats sont donc très encourageants.

	Reconnu en taps au dos	Reconnu en impulsif vers l'avant	Reconnu en impulsif vers l'arrière
Taps au dos	<b>93,8%</b>	4,1%	2,1%
Impulsif vers l'avant	2,1%	<b>91,7%</b>	6,2%
Impulsif vers l'arrière	4,15%	4,15%	<b>91,7%</b>

Figure 115. Matrice de confusion.

<sup>10</sup> Soit deux vecteurs V1 et V2 contenant les primitives pi et leurs poids respectifs p'i.

$$Distance = \sqrt{\sum \sqrt{(V1.pi * V1.p'i - V2.pi * V2.p'i)}}$$

## 5. Intégration

Dans cette section, nous présentons des exemples d'application et d'évolution de TimeTilt.

### *TimeTilt pour la gestion de fenêtres*

Comme nous l'avons plusieurs fois mentionné, TimeTilt peut être utile pour faciliter la navigation entre les fenêtres ouvertes sur le dispositif. En effet, la petite taille des écrans ne permet pas l'affichage de plusieurs fenêtres, bien que la majorité des mobiles le permettent d'un point de vue technique (la plupart des dispositifs actuels sont multi-tâches). Nous avons donc développé un prototype tournant sur le système Windows Mobile. Ce logiciel détecte l'ouverture des applications et mémorise les six dernières applications ouvertes dans une liste, ce qui permet à l'utilisateur de naviguer dans les applications qu'il utilise plus fréquemment à un moment donné.

Lors de la réalisation de notre application, nous avons supposé que les utilisateurs n'utilisent pas plus de six applications durant les périodes d'utilisation du mobile. 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 réparties en trois parties. La première partie visait à collecter des informations relatives aux répondants et aux types de dispositifs 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 ont un accès Internet et s'en servent. Ils sont 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%) utilise leur dispositif environ 4 fois par jour, 33% toutes les heures et 17% toutes les 30 minutes. La durée varie entre 1 ou 2 minutes (40%) et 10 minutes (17%), en passant par 5 minutes (38%). Seulement 4% des sondés y consacrent 30 minutes ou plus. Les répondants ouvrent 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 116** montrent les réponses aux questions suivantes : A) vous est-

t'il arrivé de vouloir utiliser ou d'utiliser plusieurs fenêtres en même temps ? ; B) vous est-t'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.

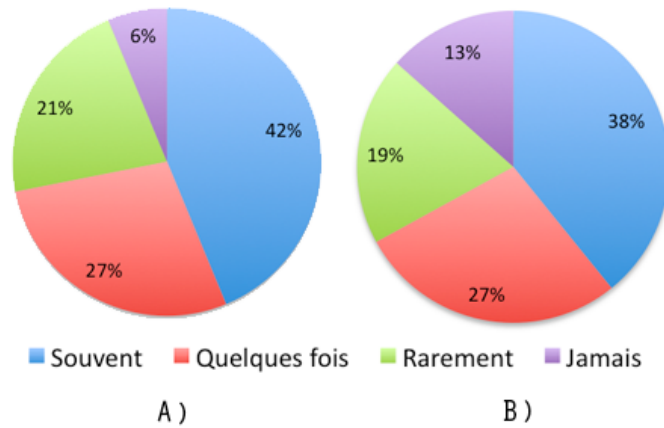


Figure 116. Taux de réponses aux questions : A) vous est-t'il arrivé de vouloir utiliser ou d'utiliser plusieurs fenêtres en même temps ? ; B) vous est-t'il arrivé de vouloir déplacer du texte ou des fichiers entre plusieurs fenêtres ?

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 facilement 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.

### *TimeTilt ++*

Nous envisageons de développer d'autres prototypes de TimeTilt. Le premier permettrait de naviguer entre différentes vues d'une application, comme dans les navigateurs Web où l'utilisateur peut ouvrir plusieurs pages. Le second servirait à activer les opérations de "undo/redo". La métaphore lenticulaire est susceptible de bien convenir à cette tâche puisqu'elle peut suggérer à l'utilisateur qu'il "remonte" le temps pour revenir à un stade antérieur de l'application (FIG. 117).



Figure 117. TimeTilt appliqué au cas des opérations de "undo/redo".

Afin de combiner ces différents cas d'application au sein d'un seul et unique logiciel, nous souhaitons également étendre le vocabulaire gestuel de TimeTilt. Le mode lenticulaire de TimeTilt peut en effet être déclenché par différents gestes selon les objets à contrôler. Par exemple une double tap à l'arrière déclencherait la navigation entre plusieurs pages Web alors qu'un tap sur le coté du dispositif activerait les opérations de "undo/redo". Un retour tactile ou visuel permettrait ensuite à l'utilisateur de savoir dans quel mode il se trouve.

### **FanTilt**

Comme TimeTilt n'utilise que les données de rotation autour de l'axe x du dispositif, elle laisse d'autres possibilités d'interaction sur les autres axes. Notamment, TimeTilt peut être combinée à FanTilt une technique que nous souhaitons développer et qui est basée sur la métaphore de l'éventail (FIG. 118) : en inclinant le dispositif de gauche à droite, l'utilisateur peut naviguer entre les vues ou les onglets de la fenêtre, alors qu'en inclinant de haut en bas, il navigue dans les fenêtres. Un récent brevet d'Apple propose d'ailleurs d'appliquer ces gestes à la navigation dans un menu.



Figure 118. FanTilt : évolution de TimeTilt basée sur la métaphore de l'éventail. L'inclinaison selon l'axe z permet de naviguer entre plusieurs onglets.

## 6. Conclusion

Nous avons présenté TimeTilt, une technique facilitant la navigation entre plusieurs vues. TimeTilt est basée sur un vocabulaire de quatre gestes tridimensionnels et 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.

Grâce à son interaction basée sur des gestes en trois dimensions, TimeTilt est compatible avec les techniques d'interactions utilisant des gestes en deux dimensions sur la surface de l'écran tactile. Elle peut notamment être combinée à des techniques de copier coller comme les MicroRolls, qui ont été présentés au [CHAPITRE 5](#). Dans ce cas l'utilisateur copie un objet avec un MicroRoll, active TimeTilt pour trouver la vue où il souhaite coller son objet, puis colle l'objet avec un autre MicroRoll. TimeTilt peut de plus s'utiliser facilement à une seule main.

Nous avons mené une expérience pour construire un algorithme de reconnaissance et montrer qu'il était possible de discriminer les trois types de gestes impulsifs utilisés dans TimeTilt. Puis nous avons proposé divers exemples d'application de TimeTilt.

Cette étude étant préliminaire nous n'avons pas encore mené d'évaluation comparative. Nous souhaitons maintenant comparer 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 (l'utilisateur peut ensuite naviguer dans la liste en déplaçant son doigt sur l'écran ([FIG. 119.A](#))) et l'interface Cube que l'on trouve sur certains dispositifs HTC où les applications favorites sont disposées sur les faces d'un cube ([FIG. 119.B](#)) que l'on contrôle en déplaçant son doigt sur l'écran.



A)



B)

Figure 119. A) Gestion de fenêtres multiples dans le Palm Pre ; B) interface Cube des dispositifs HTC pour gérer les applications favorites.

Enfin, bien que cette étude n'ait pas fait l'objet d'étude 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. Cet engouement peut être dû à la métaphore des cartes lenticulaires qui semble donner un aspect ludique à la technique.





# Conclusion



"2/3 des utilisateurs se servent de leur téléphone portable pour s'éclairer dans le noir. 7% des sondés (Etats Unis) l'utilisent pour remplacer le briquet dans les concerts et 4% illuminent leur visage avec lorsqu'ils racontent une histoire de fantômes"

Sprint Survey (2006)

<b>1. Contributions</b>	<b>173</b>
1.1 Espace de conception	173
1.2 Nouvelles techniques d'interaction	173
<b>2. Perspectives</b>	<b>175</b>
2.1 Perspectives à court terme	175
2.2 Perspectives à moyen terme	176
2.3 Perspectives à long terme	178

Aujourd'hui un même dispositif sert pour communiquer, s'informer, interagir, dans des contextes variés comme au bureau, au domicile ou dans les transports. Cependant, le développement des fonctionnalités des dispositifs mobiles est fortement contraint par les limitations de l'interface homme-machine (petit écran, clavier physique réduit ou absent, interaction à une main, etc.). L'amélioration de l'interface de ces dispositifs constitue donc un enjeu scientifique et industriel majeur. Ces travaux de thèse ont été menés dans ce contexte dans le cadre du laboratoire commun UBIMEDIA entre l'Institut Télécom et BellLabs Alcatel-Lucent. Nous résumons ci-après nos contributions et identifions plusieurs perspectives de développement.

## 1. Contributions

Les deux facettes complémentaires de ces travaux de thèse sont l'espace de conception de l'interaction sur dispositifs mobiles ainsi que la conception, la réalisation et l'évaluation de cinq techniques d'interaction.

### 1.1 Espace de conception

Pour faciliter la conception de nouvelles techniques d'interaction nous avons proposé un espace de conception composés de deux parties : un *espace problème* qui décrit les contraintes spécifiques des dispositifs mobiles, en particulier lorsqu'ils sont utilisés à une main, et une *taxonomie* des techniques de visualisation et d'interaction qui répondent aux contraintes identifiées dans l'espace problème.

Cet espace de conception peut faire office de guide de deux manières lors de la conception et de l'évaluation d'une nouvelle technique. D'une part, il vise à mettre en évidence les atouts et les faiblesses d'une technique. La liste des contraintes que nous avons dressée doit en effet être prise en compte pour créer des techniques adaptées aux dispositifs mobiles. D'autre part, la taxonomie permet de mettre en évidence les lacunes des solutions existantes, ce qui peuvent suggérer des voies intéressantes à explorer.

### 1.2 Nouvelles techniques d'interaction

En explorant les pistes identifiées dans la taxonomie et en utilisant les contraintes de l'espace problème comme guides, nous avons conçu et évalué cinq techniques d'interaction qui répondent à des problèmes spécifiques de l'interaction sur dispositifs mobiles.

### *TapTap et MagStick*

TapTap et MagStick sont des techniques de pointage sur écran tactile qui permettent de sélectionner de petites cibles en interagissant avec le pouce. TapTap permet de sélectionner une cible grâce à un double tap et MagStick offre un bâtonnet télescopique pour atteindre les cibles. Ces techniques permettent d'économiser l'espace écran et elles répondent aux contraintes de l'interaction au doigt (occultation, imprécision et accès aux bords de l'écran). Nous avons montré grâce à une expérience contrôlée qu'elles sont plus rapides et entraînent moins d'erreurs que quatre autres techniques récemment proposées dans l'état de l'art.

### *LeafMenu*

Le LeafMenu est un menu linéaire enrichi par des gestes directionnels droits ou courbes. Grâce à sa forme linéaire, il conserve une occupation spatiale minimale sur l'écran et permet aux utilisateurs de facilement se familiariser avec la technique. Les raccourcis gestuels remplacent les raccourcis clavier inutilisables lors de l'interaction à une main et peuvent être effectués près des bords de l'écran grâce à l'effet miroir du LeafMenu. Les résultats de l'expérimentation montrent que les gestes mis en œuvre dans le LeafMenu peuvent être appris rapidement.

### *MicroRolls*

Nous avons proposé d'enrichir le vocabulaire interactionnel des écrans tactiles par des gestes qui exploitent la malléabilité du pouce : les MicroRolls. Ces gestes sont compatibles avec les gestes usuels grâce à un algorithme de reconnaissance permettant de distinguer les différents types de tracés sur l'écran. Une expérience a permis de récolter un échantillon de gestes et de construire cet algorithme de reconnaissance. Des études comparatives ont ensuite montré que les MicroRolls constituent une alternative efficace à deux autres techniques couramment utilisées pour effectuer une tâche de copier/coller.

### *TimeTilt*

La majorité de nos contributions portent sur les gestes en deux dimensions sur écran tactile, mais nous avons également élargi notre cadre d'étude à l'interaction gestuelle en trois dimensions. Nous avons ainsi proposé TimeTilt, une technique facilitant le basculement entre plusieurs vues grâce à un vocabulaire de gestes exécutés dans l'espace et détectés par un accéléromètre. Une base d'échantillons de gestes a été récoltée et a permis de construire un module de reconnaissance des gestes.

## 2. Perspectives

Les perspectives de travaux futurs sont nombreuses et divisées en trois parties : à court, moyen et long terme.

### 2.1 Perspectives à court terme

#### *Evaluations approfondies de nos techniques*

Nos travaux nous ont amené à évaluer nos techniques par diverses expérimentations contrôlées mais nous souhaiterions approfondir ces études. Nous résumons ci-après les évaluations que nous aimerions conduire pour enrichir le travail déjà établi.

- Le bâtonnet de **MagStick** est composé de deux parties de taille égale. Une étude complémentaire consisterait à comparer différents comportements du bâtonnet de MagStick. Ce comportement pourrait être : variable (un gain permettant de définir la taille d'une partie du bâtonnet par rapport à l'autre selon la place disponible à l'écran) ou homothétique (les tailles des deux parties du bâtonnet étant proportionnelles). Nous pourrions également tester ces variantes de MagStick sur différentes tailles d'écran (petite, moyenne, large) et vérifier si certains comportements du bâtonnet sont plus adaptés à ces diverses tailles d'écran.
- Nous avons mené une expérience d'apprentissage des gestes du **LeafMenu**. L'étape suivante serait de le comparer aux Marking Menus ([KURTENBACH 91](#)).
- Les **MicroRolls** sont reconnus via un algorithme construit en collectant des échantillons de gestes effectués par les utilisateurs. Lors de cette expérience, ces derniers n'avaient pas de retour visuel et ne pouvaient donc pas savoir si le geste était correct. Or, un retour adapté peut aider l'utilisateur à effectuer et à se mémoriser les gestes ([BAU 08](#)). Nous aimerions donc effectuer une expérience complémentaire pour étudier si la présence d'un retour visuel permet aux utilisateurs de s'adapter aux spécificités de l'algorithme et, par la même, d'améliorer le taux de reconnaissance.
- Nous souhaitons poursuivre les travaux sur **TimeTilt** en menant 2 expériences : l'une pour déterminer le nombre maximum de degrés pouvant être contrôlés en inclinant le dispositif et la seconde pour comparer TimeTilt avec d'autres alternatives (citées au [CHAPITRE 6](#)).

## 2.2 Perspectives à moyen terme

### *Compléter l'espace de conception par une typologie des "facultés"*

En plus d'un espace problème et d'une taxonomie, notre espace de conception pourrait être approfondi pour mettre en évidence des "facultés" intéressantes à exploiter dans le cadre de l'interaction sur mobiles.

Par exemple, le fait d'avoir les doigts placés à l'arrière du dispositif est une caractéristique résultant de la prise en main du mobile qui n'est exploitée que depuis peu, pour quelques techniques. Les MicroRolls sont un autre exemple de technique exploitant une aptitude intéressante : la capacité de rouler son doigt sur une surface grâce à la forme bombée et molle du pouce. La "proprioception" est une autre faculté qui pourrait être explorée : le dispositif mobile étant petit et tenant dans une main, l'utilisateur est capable de sentir les proportions de l'appareil et de toucher un endroit spécifique de son écran sans avoir à le regarder (par exemple le coin supérieur gauche, le centre, etc.). Cette propriété pourrait par exemple être utilisée dans des techniques d'interaction où l'utilisateur devrait pointer ou effectuer des gestes à des endroits spécifiques de l'écran pour déclencher des commandes.

Ces exemples montrent qu'il serait intéressant de lister ces "facultés" afin de mieux les mettre en relation avec les problèmes qu'elles peuvent résoudre. Pour démarrer ce travail, nous envisageons de prendre l'organisation de notre espace problème et d'identifier, pour chaque contrainte, une propriété pouvant être mise à profit pour transformer la contrainte en un avantage. Par exemple la surface bombée et molle du pouce entraîne des instabilités lors des déplacements sur l'écran mais rend possibles les MicroRolls.

### *Généraliser l'espace de conception à d'autres plates-formes*

Une autre perspective a trait à la généralisation de notre espace de conception, initialement centré sur l'interaction sur dispositifs mobiles. Nous pensons qu'il existe un lien fort entre les petites et les grandes surfaces tactiles car les contraintes sur ces deux types de plates-formes sont assez similaires. Il existe par exemple dans les deux cas des manques matériels comme l'absence de clavier physique et le manque d'états interactionnels de l'écran tactile. Les techniques proposées doivent aussi prendre en compte les limitations liées aux capacités humaines, par exemple l'accès aux bords d'un tableau blanc interactif peut être difficile du fait de la taille de ce dispositif.

Pour commencer ce travail, nous envisageons de reprendre les contraintes relatives aux dispositifs mobiles et de vérifier dans quelle mesure elles s'appliquent aussi au cas des grandes surfaces tactiles. Nous chercherons ensuite les contraintes plus spécifiques à ces systèmes

interactifs en nous basant sur la littérature existante dans ce domaine. Nous souhaitons également étudier l'adaptation de nos techniques aux grands écrans, en particulier MagCane (citées au [CHAPITRE 3](#)), une adaptation de MagStick à ces systèmes.

### *Développement de nouvelles techniques d'interaction*

Nous aimerions approfondir les quatorze alternatives qui tirent parti des propriétés de nos techniques (présentées dans les sections "intégrations" des [CHAPITRES 3, 4, 5 ET 6](#)). En particulier :

- Nous souhaitons comparer les différentes techniques de menus proposées : le CornerMenu (adaptation fixe du LeafMenu présentée au [CHAPITRE 4](#)) ainsi que le TapMenu et le MagMenu qui sont des menus utilisant les propriétés de TapTap et MagStick (voir le [CHAPITRE 3](#)).
- Nous pensons étudier plus en détail RollPod (voir le [CHAPITRE 5](#)) qui utilise les MicroRolls pour permettre à l'utilisateur de contrôler une bibliothèque musicale même lorsque le dispositif est dans sa poche. Nous souhaiterions vérifier que les MicroRolls sont effectivement réalisables lorsqu'ils sont effectués dans un espace réduit comme une poche. Nous envisageons également d'enrichir le vocabulaire gestuel de l'utilisateur en proposant d'autres gestes qui pourraient être facilement effectués lorsque la main et le dispositif mobile sont dans une poche, par exemple des gestes utilisant la paume de la main.
- Nous avons proposé au [CHAPITRE 6](#) une adaptation de TimeTilt permettant d'activer les commandes "undo/redo". Nous pensons que cette tâche est intéressante car il est souvent difficile pour l'utilisateur d'annuler ses actions fautes de raccourcis clavier facilement utilisables. Cette version de TimeTilt faciliterait également l'entrée de texte puisque l'utilisateur pourrait supprimer des mots sans avoir à appuyer sur une touche de suppression : il lui suffirait d'incliner le dispositif vers l'arrière pour revenir à un moment où le mot n'est pas écrit.

Par ailleurs nous souhaitons également continuer à concevoir, implémenter et évaluer de nouvelles techniques pour dispositifs mobiles. Nous pensons en particulier étendre les travaux du [CHAPITRE 6](#) pour créer des techniques d'interaction à base de capteurs. Ce travail nécessitera une analyse plus poussée des technologies de capteurs existantes (les capteurs de mouvements mais également ceux de pression, de luminosité ou encore de présence) et des techniques d'interaction les exploitant.



## 2.3 Perspectives à long terme

### *Intégration matérielle des techniques sur les dispositifs grand public*

Nous avons vu que les dispositifs mobiles actuellement disponibles sur le marché intègrent relativement peu de techniques avancées comparativement à l'ensemble des solutions présentées dans notre espace de conception (CHAPITRE 2). Il est en effet assez difficile de transférer les techniques proposées dans le milieu académique aux dispositifs grand public à cause des aspects liés à l'implémentation. Il existe en effet peu de boîtes à outils permettant de développer facilement ces techniques avancées, en particulier à cause de la diversité des systèmes d'exploitation et de l'évolution rapide des dispositifs. Une boîte à outils peut en effet vite devenir obsolète si elle ne peut pas intégrer de nouveaux modules tels que la prise en compte de nouveaux capteurs (via des mises à jour ou des plugins).

Un autre point est que les boîtes à outils existantes sur mobiles sont simplifiées et ne prennent en compte que les événements basiques tels que le Press, Drag, Release (ainsi qu'un identifiant lorsque l'écran est multi-point). Une extension intéressante consisterait à intégrer un vocabulaire événementiel plus riche et de plus haut niveau où les gestes seraient des événements à proprement parler. Ceci faciliterait la conception de nouvelles techniques puisque le concepteur n'aurait pas à développer entièrement les modules de reconnaissance de gestes dans son application. Enfin, une boîte à outils évoluée permettrait d'aider le concepteur à choisir les techniques compatibles pouvant être intégrées dans une même application. Le concepteur pourrait notamment voir que deux techniques utilisent les mêmes événements matériels, ce qui les rend incompatibles.

Nous avons déjà entrepris le développement d'une boîte à outils permettant d'enrichir le vocabulaire événementiel des écrans tactiles. Cette boîte à outils a été réalisée dans le cadre de notre collaboration avec Bell Labs Alcatel-Lucent. Nous avons en particulier travaillé avec une équipe du département Social Communications qui s'intéresse aux réseaux sociaux. Notre boîte à outils vise à faciliter l'intégration des techniques que nous avons développées dans les applications de Bell Labs (par exemple pour qu'elles puissent utiliser les MicroRolls ou les gestes du LeafMenu).

Cependant ce travail reste préliminaire en comparaison des nombreux travaux que l'on peut envisager pour construire une boîte à outils complète qui serait capable de prendre en compte les évolutions rapides des dispositifs et qui aiderait les concepteurs à intégrer plus facilement les techniques d'interaction dans les dispositifs grand public.

### *Etudier les aspects de design*

L'interaction sur dispositifs mobiles fournit des opportunités pour proposer de nouveaux paradigmes et de nouvelles techniques d'interaction. Ceci contraste avec le cas des ordinateurs conventionnels où les habitudes sont fortement enracinées et les utilisateurs assez peu enclins aux changements (BEAUDOUIN-LAFON 04) (ZHAI 09). Les dispositifs mobiles offrent donc un cadre d'étude intéressant pour comprendre pourquoi certaines techniques nouvelles sont plus appréciées et utilisées que d'autres. Nous voyons en particulier deux pistes à étudier par rapport à nos travaux.

D'une part, nous pensons qu'il serait intéressant de pondérer les contraintes identifiées dans le CHAPITRE 2 pour aider le concepteur à choisir entre plusieurs techniques. Par exemple, entre une technique répondant au problème d'occultation et une autre facilitant l'accès aux bords de l'écran, laquelle est-il préférable de choisir pour garantir une meilleure appropriation de la technique par l'utilisateur ?

D'autre part, selon Shneiderman (SHNEIDERMAN 04), les critères d'utilisabilité (présentés au CHAPITRE 2) sont tous au même niveau. La satisfaction est donc un critère aussi important que les critères de rapidité, précision, apprentissage et mémorisation. Il existe pourtant relativement peu d'études concernant ce critère, et nous pensons que certains facteurs peuvent rendre une technique plus attrayante. Par exemple TapTap a été fortement appréciée par les utilisateurs qui l'ont jugée plus "simple" que les autres techniques. De même MagStick a été appréciée grâce à son design inspiré de certains jeux de Golf, et TimeTilt pour la métaphore des cartes lenticulaires. Certaines questions se posent alors et mériteraient un travail de recherche plus approfondi, par exemple :

- Comment définir qu'une technique est plus "simple" qu'une autre ? Peut-on dire qu'une technique "simple" sera plus appréciée par les utilisateurs ?
- Quel rôle a le jeu ou l'aspect ludique dans l'appréciation d'une technique ? Cet aspect favorise-t-il la découverte, l'apprentissage ou la mémorisation de la technique ?
- La présence d'animations affecte-t-elle l'appréciation d'une technique ? Peut-on construire une liste de recommandations pour aider à concevoir une animation adéquate (par exemple combien de temps doit-elle durer) ?
- Quel est l'impact relatif des critères d'efficacité (rapidité et précision) et de satisfaction par rapport à l'adoption d'une technique par les utilisateurs ?

## Table des figures

### Chapitre 1 : Introduction

Figure 1. A) Motorola DynaTac (1983) ; B) Motorola Dext (2009). ..... 11

### Chapitre 2 : Espace de conception

Figure 2. Espace problème : les contraintes des dispositifs mobiles. .... 20

Figure 3. Utilisation d'un projecteur : A) pour partager des fichiers entre mobiles ; B) pour zoomer sur une carte en s'approchant du mur sur lequel elle est projetée. .... 22

Figure 4. Occultation de l'avant-bras et de la main sur une tablette tactile ou du pouce sur un petit écran tactile. .... 24

Figure 5. Erreur de perception de la position du pointeur sous le pouce. B) position supposée par les utilisateurs ; B) position réelle. .... 24

Figure 6. A) Modèle simplifié de l'interaction avec une souris sous le système d'exploitation Windows ; B) Modèle simplifié d'interaction sur l'écran tactile d'un dispositif Windows Mobile. .... 26

Figure 7. A) Modèle simplifié d'interaction sur l'écran tactile d'un dispositif Android ; B) Modèle simplifié d'interaction sur l'écran tactile d'un iPhone. . 27

Figure 8. EarPod aide l'utilisateur à se déplacer dans une bibliothèque musicale grâce à des retours auditifs spatialisés. .... 29

Figure 9. Exemple de retour tactile A) séquentiel : reconnaissance d'un triangle) B) non séquentiel : reconnaissance d'un motif associé à une station de radio. ... 30

Figure 10. A) Limites de mouvement de flexion/extension du pouce ; B) Impacts sur l'accès à certaines zones de l'écran (les zones sombres sont plus difficiles d'accès). .... 31

Figure 11. Les interactions multi-doigts sont difficiles à une seule main. .... 31

Figure 12. Changement de prise en main pour atteindre les boutons physiques du dispositif sur les cotés. .... 32

Figure 13. A) Les mouvements de flexion/extension de la main s'étendent sur une plage de 105° : B) ceux de pronation/supination jusqu'à 125° et C) ceux de déviation ulnaire et radiale du poignet jusqu'à 45°. .... 32

Figure 14. Taxonomie des techniques d'interaction et de visualisation. .... 35

Figure 15. Evolution de la taille des cibles entre 2007 et 2009 sur les systèmes d'exploitation Windows Mobile et Rim (BlackBerry). .... 37

Figure 16. Modèle d'interaction A) de l'écran cliquable du BlackBerry Storm ; B) de GraspZoom, une technique détectant la pression du doigt sur l'écran. .... 38

Figure 17. A) Le menu du Nokia n97 est placé dans une barre en bas de l'écran et il contient parfois une barre de défilement ; B) Celui du Palm Pre est placé dans une barre en haut de l'écran. Les items peuvent être associés à des raccourcis claviers. C) applications iPhone : une barre de navigation contenant les contrôles principaux en haut de l'écran ; une barre d'outils ou d'onglets en bas de l'écran. ....	39
Figure 18. A) Navigation entre fenêtres ouvertes sur le Palm Pre ; B) Drag & Drop sur iPhone : une marque à droite sur la cible permet de la déplacer. ....	40
Figure 19. A) Contrôler le curseur d'un ordinateur à proximité ; B) sélectionner des objets sur une carte physique ; C) Reconnaissance de la main pour déterminer la position du curseur. ....	41
Figure 20. A) Alphabet Unistroke ; B) Alphabet EdgeWrite ; C) Gestes utilisés dans LaunchTile pour activer des commandes. ....	42
Figure 21. A) Un geste directionnel vers la gauche dans la zone de lancement du Palm Pre lance la commande "précédente" ; B) Sur Android, l'utilisateur définit un mot de passe gestuel pour déverrouiller l'écran tactile ; C) Gesture Dial (iPhone) permet d'appeler ses contacts en dessinant des raccourcis gestuels personnalisés. ....	42
Figure 22. A) SmartPad, l'utilisateur effectue des gestes sur un clavier physique pour déclencher des fonctionnalités ; B) EdgeWrite revisité pour la saisie de texte sur quatre touches. ....	43
Figure 23. Boomerang : A) l'utilisateur lance un objet hors d'une fenêtre ; B) un cercle apparaît ; C) il ouvre une autre fenêtre et clique sur le cercle resté visible. L'objet lancé apparaît alors sur la nouvelle fenêtre. ....	44
Figure 24. L'inclinaison permet de passer du mode portrait au mode paysage. ....	45
Figure 25. Les différents multiplexages d'information : A) spatial (ZoneZoom), B) temporel (Overview + Detail) et C) en profondeur (Macroscope). ....	46
Figure 26. A) Un Treemap pour représenter les discussions d'un forum ; B) LaunchTile utilise le zoom sémantique pour naviguer dans les applications ; C) Fathumb permet de filtrer parmi 9 catégories grâce à une analogie spatiale avec les touches du clavier. ....	47
Figure 27. Zen : A) vue zoomée ; B) contour de la vue générale indiquant à l'utilisateur les proportions totales de la carte ; C) cadre rouge permettant à l'utilisateur de savoir la position de la vue zoomée sur la carte. ....	48
Figure 28. Techniques Focus + Context : A) Flip Zooming ; B) DateLens ; C) Spiralist. ....	49
Figure 29. Suggérer l'existence d'informations en dehors de l'écran : A) Halo ; B) Wedge. ....	50

Figure 30. Dans Zoomit, les deux couches sont plus facilement discernables grâce aux mouvements initiés par l'utilisateur lors d'un zoom. .... 51

### Chapitre 3 : Pointage sur écrans tactiles

Figure 31. Shift .....	59
Figure 32. Burst rapproche les cibles potentielles vers le pouce. ....	60
Figure 33. A) Diagramme de Voronoi; B) Starbust ; C) Première étape de l'algorithme de Starbust. ....	62
Figure 34. Thumbspace : A) l'appui sur un bouton physique provoque l'affichage d'une vue radar de l'interface ; B) l'utilisateur appuie sur le coin de la vue radar ; C) le curseur apparaît près des bord de l'écran alors que le pouce reste dans une zone facilement accessible. ....	63
Figure 35. Escape. ....	64
Figure 36. Sliding Widgets. ....	64
Figure 37. A) ShapeWriter ; B) Clavier virtuel sur l'iPhone ; C) UniGlyph. ....	65
Figure 38. Tableau récapitulatif de l'état de l'art vis à vis des problèmes de l'interaction au pouce (Les zones grisées dans la colonne de droite sont plus difficiles à atteindre et les zones rayées sont impossibles à atteindre). ....	67
Figure 39. TapTap. A) le premier tap définit une zone d'intérêt ; B) la zone d'intérêt prend la forme d'une vue zoomée au centre de l'écran ; C) le second tap sélectionne la cible, désormais assez large pour être sélectionnée avec le pouce. ....	69
Figure 40. MagStick : A) l'utilisateur appuie sur l'écran pour définir un point de référence ; B) en déplaçant le pouce un bâtonnet ("stick") télescopique apparaît. Le stick a un mouvement inversé ; C) le curseur est magnétisé par les cibles pour éviter les erreurs lors du relâchement du pouce. ....	70
Figure 41. A) Avec TapTap les cibles sont assez larges pour éviter l'occultation ; B) MagStick évite l'occultation car le pouce s'éloigne de la cible au lieu de s'en rapprocher. ....	73
Figure 42. A) la cible rouge doit être sélectionnée parmi les 15 autres distracteurs ; B) L'écran est divisé en 8 zones de même surface. ....	76
Figure 43. Temps moyen (ms) pour chaque technique (les barres représentent un intervalle de confiance de 95%). ....	78
Figure 44. Taux d'erreurs (%) pour chaque technique (les barres représentent un intervalle de confiance de 95%). ....	80
Figure 45. Résultats des questionnaires utilisateurs. ....	81

Figure 46. Tableau récapitulatif des six techniques de pointage évaluées. Les cases colorées correspondent à la mise en place d'une stratégie pour réduire le temps d'exécution de l'étape. ....	82
Figure 47. Traces du pouce laissées sur l'écran pour chaque technique. ....	85
Figure 48. TapMap : A) Les contacts sont regroupés sous une même icône ; B) l'utilisateur utilise TapTap pour sélectionner un groupe ce qui a pour effet de C) zoomer sur la région parisienne. D) Lorsque l'échelle est assez large, les contacts n'apparaissent plus sous forme de groupes. ....	86
Figure 49. TapWeb permet de zoomer et de naviguer dans une page Web et elle facilite la sélection des liens (relativement petits) qui s'y trouvent. ....	87
Figure 50. Le TapMenu repose sur la sélection en deux taps et permet d'afficher jusqu'à 49 items dans un menu linéaire. ....	88
Figure 51. MagPaint : A) l'utilisateur place le curseur sur une zone active d'un objet et attend un délai. B) Il peut ensuite retailler et incliner des formes géométriques en une seule opération. ....	89
Figure 52. MagMenu est un menu circulaire basé sur l'utilisation de MagStick. Les items ne sont pas occultés par le pouce puisqu'il s'éloigne de ceux-ci au lieu de s'en rapprocher. ....	89
Figure 53. MagCane est une évolution de MagStick pour grands écrans qui permet d'atteindre des zones éloignées comme le haut de l'écran. ....	90
Figure 54. Tableau récapitulatif des propriétés de TapTap et MagStick face aux problèmes de l'interaction à une main. ....	91

#### **Chapitre 4 : Menus sur écrans tactiles**

Figure 55. ThumbMenu : A) l'utilisateur ouvre un sous-menu dans la partie supérieure du menu et peut déplacer son doigt au centre pour faire basculer le menu ; B) le sous-menu est alors affiché dans le prolongement du pouce. ....	97
Figure 56. ArchMenu : A) le menu s'affiche dans un coin ; B) les sous-menus se déploient en éventail lorsque le curseur est au dessus d'un item. ....	97
Figure 57. Menu linéaire avec raccourcis gestuels : A) mode novice ; B) mode expert. ....	98
Figure 58. A) le menu linéaire n'occulte qu'une partie de la cible ; B) le menu circulaire cache toute la cible ; C) le menu circulaire peut sortir des bords de l'écran. ....	98
Figure 59. Marking menu : A) mode novice ; B) mode expert. ....	99
Figure 60. Compound Marking Menu : A) mode novice ; B) mode expert. ....	100

Figure 61. Multi-Stroke Menu illustré par la technique d'entrée de texte T-Cube : A) les différents menus se superposent ; B) en mode expert l'utilisateur effectue des marques temporelles. .... 101

Figure 62. Wavelet Menu. .... 102

Figure 63. A) QuikWriting ; B) Flower menu. .... 102

Figure 64. Tableau récapitulatif des techniques de menus. .... 103

Figure 65. Mode novice : A) l'utilisateur presse une cible avec son pouce ; B) après un délai de 300ms, le menu apparaît ; C) l'utilisateur relâche son doigt sur l'item désiré pour l'activer ; D) un retour visuel confirme l'item sélectionné. .... 105

Figure 66. Grâce à l'effet miroir du LeafMenu, la distance d'un item au point d'activation est invariante comme le montrent ces quatre cas d'activation du menu. .... 106

Figure 67. A) 3 gestes droits et 4 gestes courbés combinés à B) 4 directions cardinales créent un ensemble de C) 24 gestes possibles. .... 106

Figure 68. Mode expert : A) l'utilisateur presse une cible puis B) dessine directement la marque associée à l'item désirée. C) A la fin de son geste, un retour visuel apparaît pour confirmer l'item sélectionné. .... 107

Figure 69. Menu linéaire standard : A) si le point d'activation est près du bord supérieur de l'écran, le menu s'affiche en dessous; B) si le point d'activation est près du bord inférieur de l'écran, le menu est translaté vers le haut mais l'ordre des items reste inchangé ; C) avec le LeafMenu, l'ordre des items est inversé de façon à ce que l'item 1 soit toujours le plus près du point d'activation. .... 108

Figure 70. A) le pouce peut occulter le menu dans la partie gauche de l'écran; B) dès que l'utilisateur déplace son pouce, il peut voir de nouveau le menu. .... 108

Figure 71. Les 4 raccourcis gestuels associés à l'item 6 permettent la sélection de cet item en mode expert quel que soit l'endroit où se situe le point d'activation. .... 109

Figure 72. L'utilisateur peut choisir un des quatre gestes possibles pour sélectionner un item. .... 109

Figure 73. Sélection de l'item "Soldat" parmi 7 autres. .... 112

Figure 74. Pourcentage de sélections testées en mode expert. .... 113

Figure 75. Pourcentage de sélections réussies en mode expert (sur le nombre total de sélections tentées en mode expert). .... 114

Figure 76. Pourcentage de sélections correctes en mode expert en fonction des différents items et de l'indice du bloc. .... 115

Figure 77. Leaf Mail. .... 117

Figure 78. Corner Menu. ....	118
Figure 79. Tableau récapitulatif des propriétés du LeafMenu. ....	118
 <i>Chapitre 5 : Gestes physiques sur écrans tactiles</i>	
Figure 80. BezelSwipe : l'utilisateur effectue des gestes "Swipe" en partant des bords de l'écran. Plusieurs endroits sur les bords possèdent une fonction spécifique : la sélection d'images, de mots ou de paragraphes. ....	122
Figure 81. Les gestes "Rubbing" sont basés sur la métaphore du gommage. ....	123
Figure 82. A) Radial Scroll, B) CurveDial. ....	124
Figure 83. CycloStar. ....	124
Figure 84. RollPad (l'utilisateur appuie son doigt normalement puis le fait rouler sur la droite). ....	125
Figure 85. SimPress. ....	126
Figure 86. Gestes à deux ou trois doigts sur le pavé tactile du Mac Book Pro. ....	126
Figure 87. Vocabulaire de gestes multi-doigts. ....	127
Figure 88. MicroRolls. ....	128
Figure 89. Décomposition d'une force (R) en ses composantes normale (N) et tangentielle (T). ....	129
Figure 90. A) Glissement d'une balle sur une surface, le point de contact reste le même tout au long du geste ; B) Roulement d'une balle sur une surface plane, le point de contact change pendant le mouvement, il n'y a pas de frottements. ..	130
Figure 91. Vocabulaire de 16 gestes élémentaires. ....	131
Figure 92. Tracés des 16 gestes élémentaires. Les points représentent les positions successives du barycentre de la surface du pouce sur l'écran. ....	132
Figure 93. Matrices de confusion obtenues pour chaque catégorie de gestes dans différentes conditions : A) tous gestes compris B) tous les gestes exceptés les gestes effectués près d'un bord de l'écran et en direction de ce même bord. .	134
Figure 94. Matrice de confusion totale (les gestes effectués près d'un bord de l'écran et en direction de ce même bord étant ignorés). ....	135
Figure 95. Association des MicroRolls aux commandes fréquemment utilisées dans les systèmes interactifs : A) mode novice et B) mode expert d'un RollMark menu. .	136
Figure 96. A) la tâche de copier/coller ; B) la barre d'outils et le menu contextuel pour les deux conditions de difficulté. ....	138
Figure 97. Temps total de complétion de la tâche (les barres représentent un intervalle de confiance de 95%). ....	141
Figure 98. A) la tâche de copier/coller sur des petites cibles ; B) Mode de sélection précise (inspiré de TapTap). ....	143
Figure 99. Temps total de complétion de la tâche. ....	144



Figure 100. Résultats des questionnaires utilisateurs (les deux expériences confondues). ..... 145

Figure 101. RollPod utilise les MicroRolls pour contrôler une bibliothèque musicale. .... 146

Figure 102. RollPod utilise les MicroRolls pour permettre la sélection de cases dans un tableau. .... 147

**Chapitre 6 : Gestes incarnés**

Figure 103. Les composantes du vecteur accélération d'un accéléromètre à trois axes dans le référentiel du dispositif mobile. .... 152

Figure 104. L'inclinaison et la taille d'une feuille de papier sont reconnues grâce à des diodes infrarouges placées aux quatre coins de la feuille. La projection de la page de journal suit donc les mouvements de la feuille. .... 153

Figure 105. Chucking permet de déplacer une image d'un dispositif mobile vers un ordinateur grâce à un geste impulsif. .... 154

Figure 106. A) Deux écrans sont couplés en les choquant l'un contre l'autre ; B) l'utilisateur contrôle une bille sur l'écran en tapant sur les cotés du dispositif. .... 155

Figure 107. A) Dans MotionLens, les translations du dispositif permettent de naviguer dans une carte ou de changer d'échelle ; B) Dans Peephole Display, les translations du dispositif permettent d'écrire ou de dessiner sur une plus grande surface que l'écran. .... 156

Figure 108. A) L'inclinaison du dispositif permet de simuler de la profondeur dans l'écran ; B) TiltText permet de sélectionner des lettres en appuyant sur une touche puis en inclinant le dispositif dans diverses directions. .... 156

Figure 109. TimeTilt propose deux modes d'interaction et se base sur l'exécution de quatre gestes (fluides ou impulsifs). .... 158

Figure 110. A) TimeTilt s'active par un tap à l'arrière du dispositif. B) Un aperçu de la vue courante apparaît au centre de l'écran. Une animation permet de passer progressivement d'un aperçu à un autre. .... 158

Figure 111. Les mouvements impulsifs vers l'avant ou vers l'arrière du dispositif permettent de basculer entre deux vues. .... 159

Figure 112. A) Intervalles de valeur des composantes en z et y du vecteur d'accélération lorsque le dispositif tourne autour de l'axe x ; B) Découpage en secteurs réguliers pour 4 vues. .... 160

Figure 113. A) Les mouvements de flexion/extension de la main peuvent s'étendre sur une plage de 105° ; B) Découpage sigmoïde ; C) Découpage quadratique. .... 161

Figure 114. Variation de l'accélération sur l'axe z en fonction du temps pour différents gestes. ....	163
Figure 115. Matrice de confusion. ....	164
Figure 116. Taux de réponses aux questions : A) vous est-t'il arrivé de vouloir utiliser ou d'utiliser plusieurs fenêtres en même temps ? ; B) vous est-t'il arrivé de vouloir déplacer du texte ou des fichiers entre plusieurs fenêtres ? .....	166
Figure 117. TimeTilt appliqué au cas des opérations de "undo/redo". ....	167
Figure 118. FanTilt : évolution de TimeTilt basée sur la métaphore de l'éventail. L'inclinaison selon l'axe z permet de naviguer entre plusieurs onglets. ....	167
Figure 119. A) Gestion de fenêtres multiples dans le Palm Pre ; B) interface Cube des dispositifs HTC pour gérer les applications favorites. ....	169

## Publications personnelles

Roudaut, A., Bailly, G., Lecolinet, E., and Nigay, L. (2009). Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. In Proc. of Interact'08, August 24 - 28, 2009, Uppsala. Springer-Verlag, Berlin, Heidelberg, 616-619.

Roudaut, A., Baglioni, M., and Lecolinet, E. (2009). TimeTilt: Using Sensor-Based Gestures to Travel through Multiple Applications on a Mobile Device. In Proc. of Interact'08, August 24 - 28, 2009, Uppsala. Springer-Verlag, Berlin, Heidelberg, 830-834.

Roudaut, A., Lecolinet, E., and Guiard, Y. (2009). MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In Proc. of CHI'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 927-936.

Roudaut, A. (2009). Visualization and Interaction Techniques for mobile devices. In Proc. of CHIEA'09 doctoral consortium, April 04 - 09, 2009, Boston. ACM, New York, NY, 3153-3156.

Bailly, G., Roudaut, A., Lecolinet, E., and Nigay, L. (2008). Menus leaf: enrichir les menus linéaires par des gestes. In Proc. of IHM'08, September 02 - 05, 2008, Metz. ACM, New York, NY, 169-172.

Roudaut, A. (2008). Visualisation et interaction sur dispositifs mobiles. In Proc. of IHM'08 doctoral consortium, September 02 - 05, 2008, Metz. ACM, New York, NY, 169-172.

Roudaut, A., Huot, S., and Lecolinet, E. (2008). TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In Proc. of AVI'08, May 28 - 30, 2008, Napoli. ACM, New York, NY, 146-153.

Roudaut, A. and Lecolinet, E. (2007). Un espace de classification pour l'interaction sur dispositifs mobiles. In Proc. of IHM'07, November 12 - 15, 2007, Paris. ACM, New York, NY, 99-106.

Roudaut A., Coutaz J. (2006). Méta-IHM, ou comment contrôler son espace interactif ambiant. Ubimob'06, Sep 2006, Paris.

## Bibliographie

Abowd, G. D., Coutaz, J., and Nigay, L. (1992). Structuring the Space of Interactive System Properties. In Proceedings of the IFIP Tc2/Wg2.7 Working Conference on Engineering For Human-Computer interaction (August 10 - 12, 1992). J. A. Larson and C. Unger, Eds. IFIP Transactions, vol. A-18. North-Holland Publishing Co., Amsterdam, 113-129.

Albinsson, P. and Zhai, S. (2003). High precision touch screen interaction. In Proc. of Chi'03, April 05-10, 2003, Ft. Lauderdale. ACM, New York, NY, 105-112.

Aliakseyeu, D., Irani, P., Lucero, A., and Subramanian, S. (2008). Multi-flick: an evaluation of flick-based scrolling techniques for pen interfaces. In Proc. of CHI'08, April 05 - 10, 2008, Florence. ACM, New York, NY, 1689-1698.

- Alzouma, G. (2009) «Téléphone mobile, Internet et développement : l'Afrique dans la société de l'information Revue tic&société, 2009.
- Appert, C. and Zhai, S. (2009). Using strokes as command shortcuts: cognitive benefits and toolkit support. In Proc. of CHI'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 2289-2298.
- Baffoun, S., Robert, J.M. (2006). Etat de l'art des techniques de présentation d'information sur écran d'assistant numérique personnel. In Proc. of IHM'06, April 18 - 21, 2006, Montreal. ACM, New York, NY, 27-34.
- Baglioni, M., Lecolinet, E., and Guiard, Y. (2009). Espace de caractérisation des interactions gestuelles physiques sur dispositifs mobiles. In Proc. of IHM'09, October 13 - 16, 2009, Grenoble. ACM, New York, NY, 203-212.
- Bailly, G. (2009). Techniques de menus : Caracterisation, Conception et Evaluation. PhD Thesis University of Grenoble.
- Bailly, G., Lecolinet, E., and Nigay, L. (2007). Wave menus: Improving the Novice Mode of Hierarchical Marking menus. In Proc. of Interact'07. Springer, 475-488.
- Bailly, G., Lecolinet, E., and Nigay, L. (2008). Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In Proc. of AVI'08, May 28 - 30, 2008, Napoli. New York, NY, 15-22.
- Ballagas, R., Rohs, M., and Sheridan, J. G. (2005). Sweep and point and shoot: phonecam-based interactions for large public displays. In CHIEA'05, April 02 - 07, 2005, Portland. ACM, New York, NY, 1200-1203.
- Ballagas, R., Rohs, M., Sheridan, J., Borchers, J. (2006). The Smart Phone: A Ubiquitous Input Device, Pervasive Computing 2006.
- Bartlett, J. F. (2000). Rock'n' Scroll Is Here to Stay. IEEE Comput. Graph. Appl. 20, 3 (May. 2000), 40-45.
- Bastien, J.M.C., Scapin, D. (1993). Ergonomic Criteria for the évaluation of Human-Computer interfaces. Institut National de recherche en informatique et en automatique.
- Bau, O. and Mackay, W. E. (2008). OctoPocus: a dynamic guide for learning gesture-based command sets. In Proc. of UIST'08, USA, October 19. ACM, New York, NY, 37-46.
- Baudisch, P, X Xie, C Wang, et WY Ma. (2004). Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. In Proc. of UIST'04, USA, 2004, ACM New York, NY, 91-94.
- Baudisch, P. and Chu, G. (2009). Back-of-device interaction allows creating very small touch devices. In Proc. of CHI'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 1923-1932.
- Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A. (2003). Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch and Penoperated Systems. In Proc Interact'03, pp. 57-64.

- Baudisch, P., Rosenholtz, R. (2003). Halo: a technique for visualizing off-screen objects. In Proc. of CHI'03, April 05 - 10, 2003, Ft. Lauderdale. ACM, New York, NY, 481-488.
- Baudisch, P., Zotov, A., Cutrell, E., and Hinckley, K. (2008). Starburst: a target expansion algorithm for non-uniform target distributions. In Proc. of AVI'08, May 28 - 30, 2008, Napoli. ACM, New York, NY, 129-137.
- Beardsley, P., Baar, J. V., Raskar, R., and Forlines, C. (2005). Interaction Using a Handheld Projector. IEEE Comput. Graph. Appl. 25, 1 (Jan. 2005), 39-43.
- Beaudouin-Lafon, M. (2004). Designing interaction, not interfaces. In Proc. AVI '04, 15-22.
- Beaudouin-Lafon, M. (2000). «Ceci n'est pas un ordinateur: Perspectives sur l'interaction homme-machine.» TSI. Technique et science informatiques, Jan 2000.
- Bederson, B. B. (2000) Fisheye menus. In Proc. of UIST'00, November 06 - 08, 2000, San Diego. ACM, New York, NY, 217-225.
- Bederson, B. B. (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. In proc. of UIST'01, CHI Letters, 3(2), 71-80.
- Bederson, B. B., Clamage, A., Czerwinski, M. P., and Robertson, G. G. (2004). DateLens: A fisheye calendar interface for PDAs. In Proc. Of CHI'04, Mar. 2004, 90-119.
- Bederson, B. B., Hollan, J. D. (1995). Pad++: a zoomable graphical interface system. In Proc. of CHI'95, May 07 - 11, 1995, Denver. ACM, New York, NY, 23-24.
- Belatar, M. and Poirier, F. (2007). UniGlyph: une méthode universelle pour la saisie de texte sur dispositifs mobiles. In Proc. of IHM'07, November 12 - 15, 2007, Paris. ACM, New York, NY, vol. 397 : 111-118.
- Benko, H., Wilson, A. D., and Baudisch, P. (2006). Precise selection techniques for multi-touch screens. In Proc. of CHI'06, April 22 - 27, 2006, Montréal. ACM, New York, NY, 1263-1272.
- Bier, E. A., Stone, M. C., Fishkin, K., Buxton, W., and Baudel, T. (1994). A taxonomy of see-through tools. In Proc. CHI'94, April 24 - 28, 1994, Boston. ACM, New York, NY, 358-364.
- Björk, S., Holmquist, L. E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., and Franzén, K. (1999). WEST: a Web browser for small terminals. In Proc of UIST'99, November 07 - 10, 1999, Asheville. ACM, New York, NY, 187-196.
- Blanch, R. and Lecolinet, E. (2007). Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques. IEEE Transactions on Visualization and Computer Graphics 13, 6 (Nov. 2007), 1248-1253.
- Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. In Proc. of CHI'04, April 24 - 29, 2004, Vienna. ACM, New York, NY, 519-526.
- Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., and To, P. (2009). Occlusion-aware menu design for digital tabletops. In Proc. of CHIEA'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 3223-3228.

- Brewster, S. (Jan 2002). Overcoming the Lack of Screen Space on Mobile Computers. *Personal Ubiquitous Comput.* 6, 3, 188-205.
- Brewster, S., Chohan, F., and Brown, L. (2007). Tactile feedback for mobile interactions. In *Proc. of CHI'07*, April 28 - May 03, 2007, San Jose. ACM, New York, NY, 159-162.
- Brewster, S.A. (June 2002). Non-speech auditory output. In Jacko, J.A. and Sears, A.(Eds) *HumanComputer Interaction Handbook*, Chap 12, pages pp. 220-239. Mahwah, NJ.: Lawrence Erlbaum Associates.
- Burigat, S., Chittaro, L., and Gabrielli, S. (2008). Navigation techniques for small-screen devices: An evaluation on maps and web pages. *Int. J. Hum.-Comput. Stud.* 66, 2 (Feb. 2008), 78-97
- Buxton, W. (1990). A three-state model of graphical input. In *Proc. of CHI'90*, August 27 - 31, 1990, Amsterdam. ACM, New York, 449-456.
- Cadoz, C. (1994), *Le geste, canal de communication homme/machine: la communication instrumentale*. *Technique et Science de l'Information*, 13(1):31-61.
- Cao, X. and Balakrishnan, R. (2006). Interacting with dynamically defined information spaces using a handheld projector and a pen. In *Proc. of UIST'06*, October 15 - 18, 2006, Montreux. ACM, New York, NY, 225-234.
- Card, S. K., Mackinlay, J. D., Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc. Jan 1999: 686.
- Casiez, G. (2004). *Contribution à l'étude des interfaces haptiques. Le DigiHaptic: un périphérique haptique de bureau à degrés de liberté séparés*. Ph.D. Thesis, Université des Sciences et Technologies de Lille, October 2004.
- Chau, D. H., Wobbrock, J. O., Myers, B. A., and Rothrock, B. (2006). Integrating isometric joysticks into mobile phones for text entry. In *Proc. of CHIEA'06*, April 22 - 27, 2006, Montréal. ACM, New York, NY, 640-645.
- Chevaldonné, M., Merienne, F., Neveu, M. (2006). *Etude de l'interaction dans une interface de type cockpit*. In *Virtual Images*. Paris, France.
- Chittaro L. (2006). *Visualizing Information on Mobile Devices*, *IEEE Computer*, March 2006, Vol. 39, No. 3, 34-39.
- Chong, M. K. and Marsden, G. (2009). Exploring the Use of Discrete Gestures for Authentication. In *Proc. of Interact'09*, August 24 - 28, 2009, Uppsala. Springer-Verlag, vol. 5727 : 205-213.
- Collomb, M. (2006). *Vers des systèmes de fenêtrage distribués : l'évolution du drag-and-drop*. PhD Thesis university of Montpellier.
- Collomb, M., Hascoet, M., Baudisch, P., Lee, B. (2005). Improving drag-and-drop on wall-size displays. In *ProcGraphics Interface'05*, pp. 25-32.
- Couture, N. (2006). TactiMod dirige et oriente un piéton. In *Proc. of Ubimob'06*, Jun 2006, Paris 1-8.

- Cox, D. A., Chugh, J. S., Gutwin, C., and Greenberg, S. (1998). The usability of transparent overview layers. In Proc. of CHI'98, April 18 - 23, 1998, Los Angeles. ACM, New York, NY, 301-302.
- Dachselt, R. and Frisch, M. (2007). Mambo: a facet-based zoomable music browser. In Proc. of MUM'07, December 12 - 14, 2007, Oulu. ACM, New York, NY, vol. 284 110-117.
- Dourish, P. (2001). *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge: MIT Press.
- Eisenstein, J. and Davis, R. (2004). Visual and linguistic information in gesture classification. In Proc. of ICMI '04 the 6th international Conference on Multimodal interfaces, October 13 - 15, 2004, State College. ACM, New York, NY, 113-120.
- Engdahl, B., Köksal, M., and Marsden, G. (2005). Using treemaps to visualize threaded discussion forums on PDAs. In Proc. of CHIEA '05, April 02 - 07, 2005, Portland. ACM, New York, NY, 1355-1358.
- Fekete, J.-D., Elmqvist, N. and Guiard, Y. (2009). Motion-Pointing: Target Selection using Elliptical Motions. In Proc. of CHI'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 289-298.
- Fishkin, K. P., Gujar, A., Harrison, B. L., Moran, T. P., and Want, R. (2000). Embodied user interfaces for really direct manipulation. *Commun. ACM* 43, 9 (Sep. 2000), 74-80.
- Fishkin, K., Moran, T. and Harrison, B. (1998). Embodied User Interfaces: Towards Invisible User Interfaces. In Proc. of EHCI'98 the IFIP Conference on Engineering the Human-Computer Interface, Heraklion, Crete.
- Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381-391, 1954.
- Fortune, S. J. (1987). A sweepline algorithm for Voronoi diagrams, *Algorithmica* 2, 153-174.
- Francone, J., Bailly, G., Nigay, L., and Lecolinet, E. (2009). Wavelet menus: a stacking metaphor for adapting marking menus to mobile devices. In Proc. of MobileHCI'09, September 15 - 18, 2009, Bonn. ACM, New York, NY, 1-4.
- Fritz, J, K Barner, I Inc, et M Cataumet. (1999). Design of a haptic data visualization system for people with visual impairments.» *IEEE Transactions on rehabilitation engineering*, Jan 1999.
- Furnas G.W. (1986). Generalized Fisheye Views. Proc. CHI, 1986 ACM Press, pp. 16-23.
- Gallo, O. Arteaga, S.M. Davis, J.E.. (2008). Camera-based pointing interface for mobile devices, Image In Proc. of the 15th IEEE International Conference on Publication, 12-15 Oct. 2008, 1420-1423.
- Geißler, J. Shuffle, throw or take it! working efficiently with an interactive wall. In Proc. of CHIEA'98, April 18 - 23, 1998, Los Angeles. ACM, New York, NY, 265-266.
- Guimbretière, F., Winograd, T. (2000). FlowMenu: combining command, text, and data entry. In Proc. UIST'00, 213-216.

- Goldberg, D. and Richardson, C. (1993). Touch-typing with a stylus. In Proc. of INTERCHI '93 Conference on Human Factors in Computing Systems. Eds. IOS Press, Amsterdam, 80-87.
- Grossman, T. and Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In Proc. of CHI'05, April 02 - 07, 2005, Portland. ACM, New York, NY, 281-290.
- Guiard, Y. (1993). On Fitts' and Hooke's laws : simple harmonic movement in upper-limb cyclical aiming. *Acta Psychologica*, 82, 139-159.
- Guiard, Y., Beaudouin-Lafon, M. (2004). Target Acquisition in Multi-Scale Electronic Worlds. *International Journal of Human-Computer Studies*, 61, 875-905.
- Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. (2004). Object pointing: a complement to bitmap pointing in GUIs. In Proc. of GI'04, May 17 - 19, 2004, London. ACM Press. 9-16.
- Gunn, T. J., Zhang, H., Mak, E., and Irani, P. (2009). An evaluation of one-handed techniques for multiple-target selection. In Proc. of CHIEA'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 4189-4194.
- Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. (2008). Wedge: clutter-free visualization of off-screen locations. In Proc. of CHI'08, April 05 - 10, 2008, Florence. ACM, New York, NY, 787-796.
- Häger-Ross, C., Schieber, M.H. (2000) Quantifying the independence of human finger movements: comparisons of digits, hands, and movement frequencies. *Journal Neuroscience*. 15;20(22):8542-50.
- Harrison B.L., Kurtenbach G. et Vicente K.J. (1995). An Experimental Evaluation of Transparent User Interface Tools and Information Content. Proc. UIST, 1995, ACM Press, pp. 81-90.
- Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C., and Want, R. (1998). Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In Proc. of CHI'98, April 18 - 23, 1998, Los Angeles. ACM New York, NY, 17-24.
- Hascoët, M. (2003) Throwing models for large displays, Proceedings of HCI'2003, Designing for Society, Volume 2, pp. 73-77, British HCI Group.
- Hassan, N., Rahman, M. M., Irani, P., and Graham, P. (2009). Chucking: A One-Handed Document Sharing Technique. In Proc. of Interact'09, August 24 - 28, 2009, Uppsala. Springer-Verlag, vol. 5727 : 264-278.
- Hatwell, Y., Streri, A. & Gentaz, E. (Eds.) (2000). Toucher pour connaître. Psychologie cognitive de la perception tactile manuelle. Paris: Presses Universitaires de France.
- Herman I., Melancon G, Marshall M.S. (2000). Graph visualization and navigation in information visualization: a survey. *IEEE trans. on Visualization and Computer Graphics* vol. 6, No. 1, 2000, pp. 24-43.
- Hiipakka, J, et G Lorho. (2003). A Spatial Audio User Interface for Generating Music Playlists. In Proc. of the 2003 International Conference on Auditory Display, 2003: 267-270.



- Hinckley, K. (2003). Synchronous gestures for multiple persons and computers. In proc. of UIST'03, November 02 - 05, 2003, Vancouver. ACM, New York, NY, 149-158.
- Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. (2000). Sensing techniques for mobile interaction. In Proc. of UIST'00, November 06 - 08, 2000, San Diego. ACM, New York, NY, 91-100.
- Holleis, P., Otto, F., Hussmann, H., and Schmidt, A. (2007). Keystroke-level model for advanced mobile phone interaction. In Proc. of CHI'07, April 28 - May 03, 2007, San Jose. ACM, New York, NY, 1505-1514.
- Holmquist, L. E. (1997). Focus+context visualization with flip zooming and the zoom browser. In Proc. of CHIEA'97, March 22 - 27, 1997, Atlanta. ACM, New York, NY, 263-264.
- Hornbaek, K. and Frokjaer, E. (2001), Reading of Electronic Documents: The Usability of Linear, Fisheye, and Overview+Detail Interfaces, in Proc. of CHI'01. ACM Press, 293-300.
- Huerta, Juan M. (2000). Speech Recognition in Mobile Environments. PhD thesis.
- Huot, S. and Lecolinet, E. (2007). ArchMenu et ThumbMenu: contrôler son dispositif mobile "sur le pouce". In Proc. of IHM'07, November 12 - 15, 2007, Paris. ACM, New York, NY, vol. 397 107-110.
- Huot, S., Lecolinet, E. (2006). SpiraList: a compact visualization technique for one-handed interaction with large lists on mobile devices. In Proc. of NordiCHI '06, Oct 2006.
- Huot, S., Nancel M., Beaudouin-Lafon, M. (2008) PushMenu: Extending Marking Menus for Pressure-Enabled Input Devices. Technical report 1502, LRI.
- Isokoski, P. and Raisamo, R. (2004). Quikwriting as a multi-device text entry method. In Proc. of NordiCHI'04, October 23 - 27, 2004, Tampere. ACM, New York, NY, 105-108.
- Jakobsen, M. R. and Hornæk, K. (2007). Transient visualizations. In Proc. of OZCHI '07 the Conference of the Computer-Human interaction Special interest Group (Chisig) of Australia on Computer-Human interaction: design: Activities, Artifacts and Environments, November 28 - 30, 2007, Adelaide. ACM, New York, NY, vol. 251. 69-76.
- Karam, M. and schraefel, m. c. (2005) A Taxonomy of Gestures in Human Computer Interactions. Technical Report ECSTR-IAM05-009, Electronics and Computer Science, University of Southampton.
- Karlson, A, Bederson, B., Contreras-Vidal, J. (2007). Understanding One Handed Use of Mobile Devices, in Handbook of Research on User Interface Design and Evaluation for Mobile Technology, 86-101, IGI Global, 2007.
- Karlson, A. K. and Bederson, B. B. (2008). One-handed touchscreen input for legacy applications. In Proc. of CHI'08, April 05 - 10, 2008, Florence. ACM, New York, NY, 1399-1408.
- Karlson, A. K., Bederson, B. B., and SanGiovanni, J. (2005). AppLens and launchTile: two designs for one-handed thumb use on small devices. In Proc. of CHI'05, April 02 - 07, 2005, Portland. ACM, New York, NY, 201-210.

- Karlsen, A. K., Robertson, G. G., Robbins, D. C., Czerwinski, M. P., and Smith, G. R. (2006). FaThumb: a facet-based interface for mobile search. In Proc. of CHI'06, April 22 - 27, 2006, Montréal. ACM, New York, NY, 711-720.
- Kauppila M, Inkeroinen T, Pirttikangas S, Riekkilä J (2008) Mobile Phone Controller Based on Accelerative Gesturing. Adjunct Proceedings Pervasive 2008, Sydney, Australia, 130-133.
- Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. (2006). Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.* 10, 5 (Jul. 2006), 285-299.
- Khella, A. and Bederson, B. B. (2004). Pocket PhotoMesa: a Zoomable image browser for PDAs. In Proc of MUM'04 the 3rd international Conference on Mobile and Ubiquitous Multimedia, October 27 - 29, 2004, College Park. ACM, New York, NY, vol. 83,19-24.
- Kobayashi, M. and Igarashi, T. (2007). Boomerang: suspendable drag-and-drop interactions based on a throw-and-catch metaphor. In Proc. of UIST'07, October 07 - 10, 2007, Newport. ACM, New York, NY, 187-190.
- Kurtenbach, G, Buxton, W. (1991). Issues in combining marking and direct manipulation techniques. In Proc. of UIST'91, November 11 - 13, 1991, Hilton Head. ACM, New York, NY, 137-144.
- Kurtenbach, G. (1993). "The Design and Evaluation of Marking Menus. Ph. D. Thesis. University of Toronto.
- Lamping J, Rao R. (1996). The hyperbolic browser: a focus+context technique for visualizing large hierarchies». *Journal of. Visual. Languages and Computing*, vol. 7, 1996, 3-32.
- Lecolinet, E. and Nguyen, D. (2006). Représentation focus+contexte de listes hiérarchiques zoomables. In Proc. of IHM'06, April 18 - 21, 2006, Montreal. ACM, New York, NY, vol. 133, 195-198.
- Lee, J. C., Hudson, S. E., and Tse, E. (2008). Foldable interactive displays. In Proc. of UIST'08, October 19 - 22, 2008, Monterey. ACM, New York, NY, 287-290.
- Lieberman, H. (1997). A Multi-Scale, Multi-Layer, Translucent Virtual Space. In Proc. of the IEEE Conference on information Visualisation, August 27 - 28, 1997. IV. IEEE Computer Society, Washington, DC, 126.
- Lin, J., Wu, Y., Huang, T.S. (2000) Modeling the Constraints of Human Hand Motion. IEEE Workshop, HUMO'00, 121-126.
- Linjama, J. and Kaaresoja, T. (2004). Novel, minimalist haptic gesture interaction for mobile devices. In Proc. of NorfiCHI'04, October 23 - 27, 2004, Tampere. ACM, New York, NY, 457-458.
- Luk, J., Pasquero, J., Little, S., MacLean, K., Levesque, V., and Hayward, V. (2006). A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. In Proc. of CHI'06, April 22 - 27, 2006, Montréal. ACM, New York, NY, 171-180.

- MacKenzie, I. S. and Oniszczak, A. (1998). A comparison of three selection techniques for touchpads. In Proc. of CHI'98, April 18 - 23, 1998, Los Angeles. ACM Press New York, NY, 336-343.
- MacKenzie, I. S., Soukoreff, R. W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17, 147-198.
- Malacria, S., Lecolinet, E., Guiard, Y. (2010). Clutch-Free Panning and Integrated Pan-Zoom Control on Touch-Sensitive Surfaces : the Cyclostar approach . In Proc. of CHI'10. To appear.
- Mankoff, J., Hudson, S. E., and Abowd, G. D. (2006). Interaction techniques for ambiguity resolution in recognition-based interfaces. In Proc. of UIST'00, 11-20.
- McCall, J. (1977). *Factors in Software Quality*. General Electric Eds.
- McNeill, D. (1992). *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago, Press, Chicago.
- Mistry, P., Maes, P. (2009). SixthSense - A Wearable Gestural Interface. In Proc. of SIGGRAPH'09, Yokohama, Japan.
- Miyahara, K., Inoue, H., Tsunesada, Y., and Sugimoto, M. (2005). Intuitive manipulation techniques for projected displays of mobile devices. In Proc. of CHIEA '05, April 02 - 07, 2005, Portland. ACM, New York, NY, 1657-1660.
- Miyaki, T. and Rekimoto, J. (2009). GraspZoom: zooming and scrolling control model for single-handed mobile interaction. In Proc. of MobileHCI '09, September 15 - 18, 2009, Bonn. ACM, New York, NY, 1-4.
- Moscovich, T. (2009). Contact area interaction with sliding widgets. In Proc. of UIST '09, October 04 - 07, 2009, Victoria. ACM, New York, NY, 13-22.
- Moscovich, T. and Hughes, J. F. (2004). Navigating documents with the virtual scroll ring. In Proc. of UIST'04, October 24 - 27, 2004, Santa Fe. ACM, New York, NY, 57-60.
- Moyle, M. and Cockburn, A. (2003). The design and evaluation of a flick gesture for 'back' and 'forward' in web browsers. In Proc. of the Fourth Australasian User interface Conference on User interfaces 2003, ACM, vol. 36. Australian Computer Society, Darlinghurst, Australia, 39-46.
- Nancel, M., Beaudouin-Lafon M., Huot, S. (2009). Un espace de conception fondé sur une analyse morphologique des techniques de menus. In Proc. of IHM 2009, Grenoble. ACM Press, pages 13-22.
- Nielsen, J. (1993). *Usability Engineering*, Academic Press Professional, 362.
- Nigay, L., Coutaz, J. (1996). *Espaces conceptuels pour l'interaction multimédia et multimodale*, TSI, spéciale Multimédia et collectif, AFCET & HERMES, 1996.
- Norman, D.A., Draper, S.W. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Erlbaum Associates, Hillsdale, NJ, 1986.

- Oakley, I. and Park, J. (2009). Motion Marking Menus: an eyes-free approach to motion input for handheld devices. *International Journal of Human-Computer Studies*, 67 (June 2009), pp 515-532.
- Olwal, A., Feiner, S., and Heyman, S. (2008). Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proc. of CHI'08*, April 05 - 10, 2008, Florence. ACM, New York, NY, 295-304.
- Oniszczyk, A. and MacKenzie, I. S. (2004). A comparison of two input methods for keypads on mobile devices. In *Proc. of NordiCHI '04*, October 23 - 27, 2004, Tampere. ACM, New York, NY, 101-104.
- Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In *Proc. of CHI'05*, April 02 - 07, 2005, Portland. ACM, New York, NY, 919-928.
- Parhi, P., Karlson, A. K., and Bederson, B. B. (2006). Target size study for one-handed thumb use on small touchscreen devices. In *Proc. of MobileHCI '06*, September 12 - 15, 2006, Helsinki. ACM, New York, NY, 203-210.
- Pascoe, J., Ryan, N., and Morse, D. (2000). Using while moving: HCI issues in fieldwork environments. *ACM Trans. Comput.-Hum. Interact.* 7, 3 (Sep. 2000), 417-437.
- Pietrzak, T. (2008). Contributions à la dissémination d'informations haptiques dans un environnement multimodal. PhD Thesis. Université Paul Verlaine - Metz.
- Pietrzak, T., Pecci, I., Martin, B. (2006) Static and dynamic tactile directional cues experiments with VTPlayer mouse. In *Proc. of Eurohaptics'06*, Paris, France, 2006. p. 63-68.
- Poirier, F. and Schadle, I. (2004). État de l'art des méthodes de saisie de données sur dispositifs nomades: typologie des approches. In *Proc. of IHM'04*, August 30 - September 03, 2004, Namur. ACM, New York, NY, 133-140.
- Pook, S. (2001). Interaction et contexte dans les interfaces zoomables. PhD Thesis, Paris, 2001.
- Pook, S., Lecolinet, E., Vaysseix, G., Barillot, E. (2000). Control Menus: Execution and Control in a Single Interactor. In *Proc. of CHI'00*, April 2000, The Hague. ACM Press, 263-264.
- Potter, R. L., Weldon, L. J., and Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proc. of CHI'88*, May 15 - 19, 1988, Washington. ACM, New York, NY, 27-32.
- 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, 3, 171-193.
- Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. (2009). Tilt techniques: investigating the dexterity of wrist-based input. In *Proc. of CHI'09*, April 04 - 09, 2009, Boston. ACM, New York, NY, 1943-1952.
- Raskin, J. (2000). *The Humane Interface*. Reading, Massachusetts: Addison Wesley.

- 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 Proc. of GI'06, June 07 - 09, 2006, Quebec. ACM International Conference Proceeding Series, 163-170.
- Rehm, M., Bee, N., André, E. (2008). Wave Like anEgyptian - Accelerometer Based Gesture Recognition for Culture Specific Interactions. In Proc. of HCI 2008 Culture, Creativity, Interaction.
- Reilly, D., Welsman-Dinelle, M., Bate, C., and Inkpen, K. (2005). Just point and click?: using handhelds to interact with paper maps. In Proc. of MobileHCI'05, September 19 - 22, 2005, Salzburg. ACM, New York, NY, 239-242.
- Rekimoto, J., Oba, H., and Ishizawa, T. (2003). SmartPad: a finger-sensing keypad for mobile interaction. In Proc. CHIEA'03, April 05 - 10, 2003, Ft. Lauderdale. ACM, New York, NY, 850-851.
- Ren, X., Moriya, S. (2000). Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. ACM TOCHI. 7(3):384-416.
- Resnick, R. & Halliday, D. (1966). Physics, Part I. New York: John Wiley & Sons, Inc.
- Rico, J. and Brewster, S. (2009). Gestures all around us: user differences in social acceptability perceptions of gesture based interfaces. In Proc. of MobileHCI '09, September 15 - 18, 2009, Bonn. ACM, New York, NY, 1-2.
- Robbins, D. C., Cutrell, E., Sarin, R., and Horvitz, E. (2004). ZoneZoom: map navigation for smartphones with recursive view segmentation. In Proc. of AVI'04, May 25 - 28, 2004, Gallipoli. ACM, New York, NY, 231-234.
- Rohs, M. and Oulasvirta, A. (2008). Target acquisition with camera phones when used as magic lenses. In Proc. of CHI'08, April 05 - 10, 2008, Florence. ACM, New York, NY, 1409-1418.
- Roth, V. and Turner, T. (2009). Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In Proc. of CHI'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 1523-1526.
- Roto, V., Popescu, A., Koivisto, A., and Vartiainen, E. (2006). Minimap: a web page visualization method for mobile phones. In Proc. of CHI'06, April 22 - 27, 2006, Montréal. ACM, New York, NY, 35-44.
- Rubine, D. (1991). Specifying gestures by example. SIGGRAPH Comput. Graph. 25, 4 (Jul. 1991), 329-337.
- Salovaara, A. (2009). Studying appropriation of everyday technologies - a cognitive approach. In proc. of CHIEA'09, April 4, 2009, Boston, MA. 3141-3144.
- Savescu, A. «Numerical Model of the Thumb.» csa.com, Jan 2005.
- Schraefel, M., Smith, G. M., Baudisch, P. (2005). Curve Dial: Eyes-free Parameter Entry for GUIs. In Proc. CHIEA'05, ACM, New York, NY, 1146-1147.

- Sears, A., Shneiderman, B. (1991). High precision touchscreens: design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.* 34(4):593-613.
- Sellen, A. J., Kurtenbach, G. P., and Buxton, W. A. (1992). The prevention of mode errors through sensory feedback. *Hum.-Comput. Interact*, 141-164.
- Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. of the 1996 IEEE Symposium on Visual Languages*, September 03 - 06, 1996). VL. IEEE Computer Society, Washington, DC, 336.
- Shneiderman, B. and Plaisant, C. (2004). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison Wesley; 4th edition.
- Smith, G. M., Schraefel, M. C. (2004). The radial scroll tool: scrolling support for stylus- or touch-based document navigation. In *Proc. of UIST'04*, 53-56.
- Tahir, M., Bailly, G., Lecolinet, E., and Mouret, G. (2009). Tactile Assistance for Selecting List Favorites with a Bifocal Absolute and Relative Presentation. In *Proc. of ChiNL'09 the 13th annual conference on Human Computer Interaction*, Leiden, The Netherlands, June 11th.
- Tapia, M. A. and Kurtenbach, G. (1995). Some design refinements and principles on the appearance and behavior of marking menus. In *Proc. of UIST'95*, November 15 - 17, 1995, Pittsburgh. ACM, New York, NY, 189- 195.
- Thevenin, D., Coutaz, J. (1999). Plasticity of User Interfaces: Framework and Research Agenda. In Johnson, C., editor, *Interact'99*, pages 110-117, Edinburgh, 1999. IFIP IOS Press.
- Tian, F. L. Xu, H. Wang, X. Zhang, Y. Liu, V. Setlur, and G. Dai. (2008). Tilt menu : using the 3d orientation information of pen devices to extend the selection capability of pen-based user interfaces. In *Proc. of CHI'08*, ACM, New York, NY, 1371-1380.
- Venolia, D. and Neiberg, F. (1994). T-Cube: a fast, self-disclosing pen-based alphabet. In *Proc. of CHI'94*, April 24 - 28, 1994, Boston. ACM, New York, NY, 265-270.
- Vogel, D. and Baudisch, P. (2007). Shift: a technique for operating pen-based interfaces using touch. In *Proc. of CHI'07*, April 28 - May 03, 2007, San Jose. ACM, New York, NY, 657-666.
- Vogel, D., Cudmore, M., Casiez, G., Balakrishnan, R., and Keliher, L. (2009). Hand occlusion with tablet-sized direct pen input. In *Proc. of CHI'09*, April 04 - 09, 2009, Boston. ACM, New York, NY, 557-566.
- Ware, C. (1996). Multimedia output devices and techniques. *ACM Computing Surveys (CSUR)*, Jan 1996.
- Wexelblat, A. (1998). Research challenges in gesture: Open issues and unsolved problems. In *Proc. of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*. Springer-Verlag, 1-11.
- Wigdor, D. and Balakrishnan, R. (2003). TiltText: using tilt for text input to mobile phones. In *Proc. of UIST'03*, November 02 - 05, 2003, Vancouver. ACM, New York, NY, 81-90.

- Williamson, J., Murray-Smith, R., and Hughes, S. (2007). Shoogle: excitatory multimodal interaction on mobile devices. In Proc. of the CHI'07, April 28 - May 03, 2007, San Jose. ACM, New York, NY, 121-124.
- Wobbrock, J. O., Morris, M. R., and Wilson, A. D. (2009). User-defined gestures for surface computing. In Proc. of CHI'09, April 04 - 09, 2009, Boston. ACM, New York, NY, 1083-1092.
- Wobbrock, J., Myers, B., and Rothrock, B. (2006). Few-key text entry revisited: mnemonic gestures on four keys. In Proc. Of CHI'06, April 22 - 27, 2006, Montréal. ACM, New York, NY, 489-492.
- Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003) EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In Proc. of UIST'03, November 2-5, 2003, Vancouver. ACM Press, New York, 61-70.
- Yatani, K., Partridge, K., Bern, M., and Newman, M. W. (2008). Escape: a target selection technique using visually-cued gestures. In Proc. of CHI'08, April 05 - 10, 2008, Florence. ACM, New York, NY, 285-294.
- Yee, K. (2003). Peephole displays: pen interaction on spatially aware handheld computers. In Proc. of CHI'03, April 05 - 10, 2003, Ft. Lauderdale. ACM, New York, NY, 1-8.
- Zaykovskiy, D. (2006). Survey of the Speech Recognition Techniques for Mobile Devices. In Proc. of SPECOM the 11th International Conference on Speech and Computer, St. Petersburg (Russia), June 2006.
- Zhai, S., Kristensson, P., Gong, P., Greiner, M., Peng, S. A., Liu, L. M., and Dunnigan, A. (2009). Shapewriter on the iPhone: from the laboratory to the real world. In Proc. CHIEA'09, 2667-2670.
- Zhai, S. and Kristensson, P. (2008). Interlaced QWERTY: accommodating ease of visual search and input flexibility in shape writing. In Proc. of CHI'08, April 05 - 10, 2008, Florence. ACM, New York, NY, 593-596.
- Zhao, S. (2009). earPod: Efficient Hierarchical Eyes-free Menu Selection. PhD thesis.
- Zhao, S. and Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. In Proc. of UIST'04, October 24 - 27, 2004, Santa Fe. ACM, New York, NY, 33-42.
- Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. (2007). Earpod: eyes-free menu selection using touch input and reactive audio feedback. In Proc. of CHI'07, April 28 - May 03, 2007, San Jose. ACM, New York, NY, 1395-1404.
- Ziat, M. (2006). Conception et implémentation d'une fonction zoom haptique sur PDAs. PhD Thesis.