



Expansion de cibles pour le pointage et sélection : application à l'interaction à distance en chirurgie augmentée

Maxime Guillon

► To cite this version:

Maxime Guillon. Expansion de cibles pour le pointage et sélection : application à l'interaction à distance en chirurgie augmentée. Interface homme-machine [cs.HC]. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAM066 . tel-01869267

HAL Id: tel-01869267

<https://theses.hal.science/tel-01869267>

Submitted on 6 Sep 2018

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

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Maxime GUILLON

Thèse dirigée par **Laurence NIGAY**, Professeur, UGA

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans l'**École Doctorale Mathématiques, Sciences et
technologies de l'information, Informatique**

Expansion de cibles pour le pointage et sélection. Application à l'interaction à distance en chirurgie augmentée

Target expansion for facilitating the pointing and selection tasks - Application to distant interaction in augmented surgery

Thèse soutenue publiquement le **7 novembre 2017**,
devant le jury composé de :

Madame LAURENCE NIGAY

PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Directeur de thèse

Monsieur MICHEL BEAUDOIN-LAFON

PROFESSEUR, UNIVERSITE PARIS-SUD, Rapporteur

Monsieur GERY CASIEZ

PROFESSEUR, UNIVERSITE LILLE 1, Rapporteur

Monsieur FRANÇOIS LEITNER

DIRECTEUR R&D, AESCULAP SAS - ECHIROLLES, Examineur

Monsieur ERIC LECOLINET

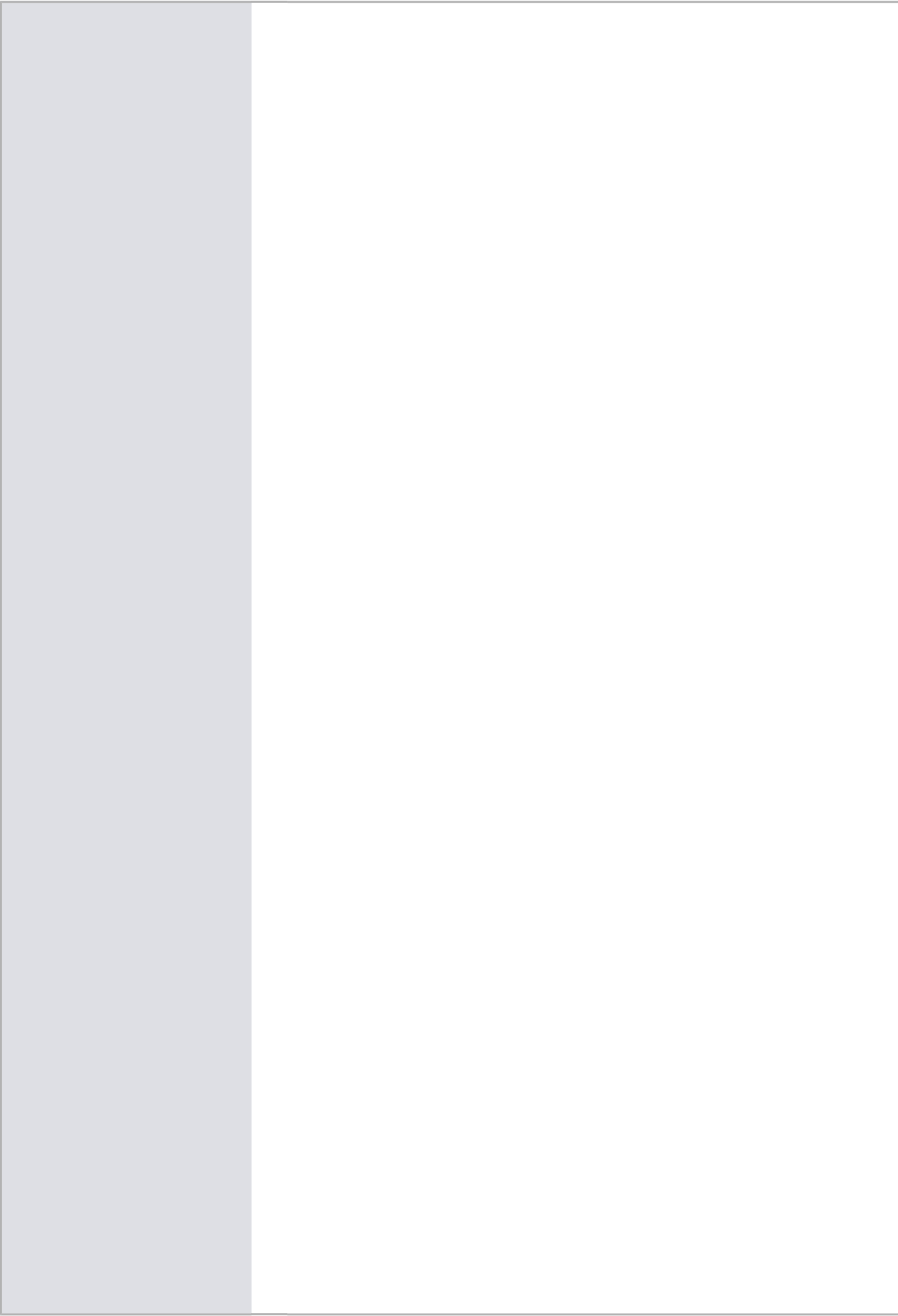
MAITRE DE CONFERENCES, TELECOM PARISTECH, Examineur

Monsieur EMMANUEL DUBOIS

PROFESSEUR, UNIVERSITE TOULOUSE-III-PAUL-SABATIER,
Président

Madame CAROLINE APPERT

CHARGE DE RECHERCHE, CNRS DELEGATION ILE-DE-FRANCE
SUD, Examineur



THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité Informatique, Arrêté ministériel du 25 mai 2016

Présentée par

Maxime GUILLON

sous la direction de Laurence Nigay
co-encadrant François Leitner
préparée au sein du Laboratoire d'Informatique de Grenoble
dans l' École Doctorale Mathématiques, Sciences et
Technologies de l'Information, Informatique

**EXPANSION DE CIBLES
POUR LE POINTAGE ET LA SÉLECTION :**
Application à l'interaction à distance
en chirurgie augmentée

*Thèse soutenue publiquement le 7 novembre 2017
devant le jury composé de*

Président : **Pr. Emmanuel Dubois**
Professeur à l'Université Toulouse III

Rapporteurs : **Pr. Michel Beaudouin-Lafon**
Professeur à l'Université Paris Sud
Pr. Géry Casiez
Professeur à l'Université Lille 1

Examineurs : **Dr. Caroline Appert**
Chargée de Recherche au CNRS
Dr. Éric Lecolinet
Maître de Conférences à Télécom ParisTech

Directrice de thèse : **Pr. Laurence Nigay**
Professeur à l'Université Grenoble Alpes

Co-encadrant de thèse : **Dr. François Leitner**
Directeur R&D à Aesculap SAS



Abstract

Pointing to/ Selecting targets is an elementary task universally present in graphical user interfaces (GUI). This task can be difficult to perform if the control of the cursor is not easy or if the target is far away or small. Thus facilitating target selection is a fundamental and active research topic in Human-Computer Interaction (HCI) and researchers have proposed numerous targeting assistance techniques.

Our research focuses on targeting assistance techniques that allocate larger activation areas to targets. Such target expansion techniques rely on two basic elements : the expansion algorithm and the visual aid. The expansion algorithm distributes partly or wholly the free space among the targets. The visual aid presents the resulting target expansion to the users. Our work is dedicated to the visual aid that enables the users to take full advantage of the target expansion technique during the pointing/selection tasks.

We first propose a three-axes design space for visual aid mechanisms. We further define a matrix-based notation for concisely describing a target expansion technique along the three design axes. We provide an analytical exploration of the design space by classifying existing target expansion techniques and by designing eight novel target expansion techniques, thus demonstrating the generative power of the design space. We also provide an experimental exploration of the design space by conducting two in-lab experiments. Based on the experimental results, we build a set of design recommendations.

We then put forward a conceptual predictive model of performance. The model relies on a systematic analysis of the relevance of the visual aid provided by a target expansion technique based on the three goal-oriented phases of a selection task : the starting phase to initiate the movement towards the target, the transfer phase to bring the cursor into the goal target and the selection validation phase. To test the model we consider experimental results of the literature and of a third conducted in-lab experiment.

Our contributions are applied to the field of Augmented Surgery and in particular interaction with a distant screen during a surgery in the operating theatre. As part of a Aesculap-CIFRE thesis, the next version of the product Aesculap's OrthoPilot® Navigation System for orthopaedic surgery will include a target expansion technique for facilitating target selection by the surgeon in the operating theatre. We also provide a new foot gesture-based technique, namely Medical TapTap, for the validation of selection in the operating theatre.

Résumé

Pointer et sélectionner une cible avec un curseur est une interaction omniprésente dans l'utilisation des ordinateurs. La sélection peut néanmoins être difficile à réaliser si le contrôle du curseur n'est pas aisé, ou encore si la cible est petite ou distante. Aussi, faciliter la sélection de cibles est un sujet d'étude essentiel et actif en Interaction Homme-Machine (IHM) et les techniques conçues sont nombreuses.

Notre travail de recherche se focalise sur l'étude des techniques d'expansion de cibles. Celles-ci facilitent le pointage et la sélection en allouant à une cible une plus grande zone active, sans toutefois modifier la cible elle-même. L'algorithme d'expansion d'une technique détermine la zone active étendue de chaque cible, tandis que l'aide visuelle de la technique fournit à l'utilisateur les informations nécessaires pour exploiter l'expansion lors du pointage et sélection. Notre travail est dédié à l'aide visuelle des techniques d'expansion de cibles en abordant leur conception et leur impact sur les performances.

Nous proposons dans un premier temps un espace de conception des aides visuelles. Cet espace est accompagné d'une notation matricielle de description des aides visuelles et il permet une classification des techniques d'expansion existantes. Son pouvoir génératif s'illustre par la création de huit techniques d'expansion, que nous évaluons au travers de deux expériences en laboratoire. Les résultats de ces deux expériences nous permettent de formuler un ensemble de recommandations pour les concepteurs de techniques d'expansion.

Nous avançons ensuite un modèle conceptuel de performance des techniques d'expansion. Ce modèle s'appuie sur la notion de pertinence de l'information contenue dans l'aide visuelle des techniques. Il considère une tâche de sélection d'un point de vue cognitif en la décomposant en trois phases déterminées par les buts de l'utilisateur : la phase préliminaire au geste, la phase de transfert du curseur vers la cible et la phase de validation de la sélection. Nous confrontons ce modèle de performance avec des résultats expérimentaux de la littérature et d'une troisième expérience en laboratoire.

Nos contributions sont appliquées à l'interaction chirurgien-ordinateur en contexte per-opératoire, c'est-à-dire en bloc opératoire durant une intervention chirurgicale. En particulier, l'entreprise partenaire de cette thèse CIFRE, Aesculap, intègre dans la prochaine version du produit OrthoPilot®, un système informatique pour la chirurgie orthopédique, une des techniques d'expansion étudiées. Nous présentons également Medical Tap-Tap, une nouvelle technique d'interaction gestuelle au pied conçue pour la validation de la sélection en contexte per-opératoire.

Remerciements

Je remercie Laurence Nigay, qui m'a fait confiance, qui m'a appris le métier et qui m'a toujours soutenu dans mon travail. Tout thésard devrait pouvoir apprendre avec un mentor aussi compétent et investi ! Je remercie François Leitner, qui a proposé et initié cette thèse CIFRE et qui m'a également toujours soutenu dans mon travail. Je remercie l'équipe Aesculap-Grenoble pour leur convivialité et leur disponibilité et je remercie l'équipe IIHM du Laboratoire d'Informatique de Grenoble pour leurs compétences et leur soutien.

Je remercie mes rapporteurs, Michel Beaudouin-Lafon et Géry Casiez, pour leur travail de relecture et leurs commentaires constructifs. Je remercie mes examinateurs, Emmanuel Dubois, Caroline Appert et Éric Lecolinet.

Je remercie Florence Velut, qui a su, dès sa première année à l'ENSC (École Nationale Supérieure de Cognitique), dont j'ai arpenté avant elle le(s) couloir(s), programmer la tessellation de Voronoï en distance de Manhattan. Je remercie Sylvain Guy pour son travail sur Medical TapTap. Je remercie les participants de mes expériences, qui ont subi avec politesse l'interminable enchaînement des pointages et des questionnaires.

Je remercie l'équipe du Centre François Magendie, au CHU de Pessac, qui m'a sauvé la vie en 2016 et je remercie l'équipe de Médecine Interne du CHU de Grenoble et l'équipe de la clinique du Grésivaudan, qui m'ont sauvé la vie une deuxième fois en 2017. Je remercie la promo 2012 "Norman" pour son soutien.

Puis, il y a les personnes qui me sont chères, celles qui tiennent l'homme debout par leur affection et leur présence. Ma gratitude et mon affection leur sont acquises.

Je remercie mes amis du beau Sud-Ouest de mon enfance : Hervé, Matxel, Pru, Simon, Sarah, JC, Doomy, Marco, Cyrille et sa douce, Arash et Anielka, Flavien, Alex Bigwood, Max Reynal, Lolo, Willy, Mousse, Marielle, Dub, Audren et Justine.

Je remercie mes amis de Grenoble : Winnax, Clém, Maître-brasseur, Aurel, Audrey, Aurélie, Polo, Flavien, Anne-Cé, Læt, Loona, Agnès, Alexandra, Manutman, Renaud, Luc, Amélie et les SevenSi.

Enfin, mes sentiments les plus profonds vont à mes parents, qui soignent, écoutent, éduquent, accueillent et me soutiennent depuis ma naissance, à mes sœurs, qui m'émerveillent et me soutiennent depuis leur naissance, à Papy, Mamie, Annie et Hachemi ; et à Élisabeth, qui a traversé avec moi tant d'épreuves, m'a toujours soutenu et m'a fait boire le philtre d'amour. Mon bonheur dépend du leur.

Grenoble, le 12 juillet 2017.

Table des matières

Abstract	II
Résumé	III
Remerciements	IV
Table des Figures	IX
Introduction	1
1 Contexte applicatif et application-cible	5
<i>Introduction</i>	5
1.1 Contexte chirurgical	6
1.2 Chirurgie augmentée	7
1.3 Application-cible : Total Knee Arthroplasty (TKA) version 5	11
<i>Conclusion</i>	13
I État de l’art	15
2 Pointage	17
<i>Introduction</i>	18
2.1 Modèles de performance	19
2.2 Espace visuel et espace moteur	27
2.3 Pointage à distance	28
2.4 Pointage à distance en contexte chirurgical	34
<i>Conclusion</i>	39
3 Aide au pointage	41
<i>Introduction</i>	41
3.1 Manipuler la trajectoire du curseur	42
3.2 Agrandir les cibles	46
3.3 Sélectionner en plusieurs temps	58
3.4 Aide au pointage à distance	60
3.5 Quelle approche pour le contexte chirurgical ?	64
<i>Conclusion</i>	66

II	Conception d'aides visuelles pour l'expansion de cibles	67
4	Espace de conception	69
	<i>Introduction</i>	69
4.1	Espace conceptuel	70
4.2	Exploration analytique de l'espace conceptuel	74
	<i>Conclusion</i>	76
5	Nouvelles techniques d'expansion	77
	<i>Introduction</i>	78
5.1	Nouvelles techniques <i>sur-espace</i>	79
5.2	Nouvelles techniques <i>sur-curseur</i>	87
5.3	Combinaisons d'aides visuelles : trois nouvelles techniques	92
5.4	Extensions	94
	<i>Conclusion</i>	98
6	Exploration expérimentale de l'espace de conception	101
	<i>Introduction</i>	102
6.1	Expérience 1 : VTE	103
6.2	Expérience 2 : Axes de conception	113
	<i>Conclusion</i>	125
III	Modèle de performance des aides visuelles	127
7	Modèle conceptuel de performance	129
	<i>Introduction</i>	129
7.1	Information fournie par les techniques	131
7.2	Pertinence de l'information selon les phases orientées-buts	135
7.3	Complémentarités et limites	138
7.4	Mise en œuvre du modèle : quantifier l'information	140
7.5	Validation analytique : aides visuelles de Fan Cursor	140
	<i>Conclusion</i>	143
8	Validation expérimentale du modèle conceptuel de performance	145
	<i>Introduction</i>	146
8.1	Analyse de l'expérience 2	146
8.2	Expérience 3	152
	<i>Conclusion</i>	163
IV	Domaine applicatif : la chirurgie augmentée	165
9	Techniques d'interaction avec OrthoPilot® d'Aesculap	167
	<i>Introduction</i>	168
9.1	Interviews de deux chirurgiens	169
9.2	Total Knee Arthroplasty (TKA) version 6	178
9.3	Medical TapTap	180
	<i>Conclusion</i>	189

Conclusion	191
Conclusion générale	191
Perspectives à court-terme	193
Perspectives à long-terme	199
A System Usability Scale	201
B Questionnaire comparatif de l'expérience 1	203
C Rappel visuel des techniques de l'expérience 2	207
D Version simplifiée du NASA-TLX (Task Load Index)	209
Bibliographie	211

Table des figures

1	Pointage à distance avec un système de CAO.	2
2	Structure du mémoire.	3
1.1	Robot da Vinci.	8
1.2	OrthoPilot® d'Aesculap.	9
1.3	Palpeur, corps-rigide et pédale médicale.	10
1.4	Écran de planification du logiciel TKA 5.	12
2.1	Loi de Fitts.	18
2.2	Distracteurs.	20
2.3	Tunnels à deux dimensions.	22
2.4	Sous-mouvements.	22
2.5	Détecter les sous-mouvements.	23
2.6	Profil de vitesse bruité.	24
2.7	Phases du mouvement de pointage.	25
2.8	Espace visuel et espace moteur.	27
2.9	Deux exemples de CD-gain.	28
2.10	Pointeur laser.	29
2.11	Principe du ray-casting.	29
2.12	Exemple de ray-casting	30
2.13	Modèle de performance du ray-casting.	31
2.14	Pointage main libre à 3 DDL.	32
2.15	Pointage main libre à 2 DDL.	33
2.16	Kinect et image de profondeur.	34
2.17	Technique de joystick en bloc opératoire.	36
2.18	Virtual Pointer.	37
2.19	Asus Xtion Pro Live et suivi de la main.	38
3.1	Drag-and-Pick.	43
3.2	Satellite Cursor.	45
3.3	Tessellation de Voronoï	46
3.4	Recouvrements et ambiguïtés.	47
3.5	Incohérence de l'expansion	48
3.6	Sigma Lenses.	49
3.7	Expansions visuelles de cibles.	50
3.8	Cell Painting et TARGET.	51
3.9	Ghost-hunting.	52
3.10	Starburst.	52

3.11	Area Cursors.	53
3.12	Bubble Cursor.	54
3.13	Lazy Bubble et Cone Cursor.	54
3.14	Dynaspot.	55
3.15	Implicit Fan Cursor.	56
3.16	Aides visuelles de Fan Cursor.	56
3.17	Bubble Lens.	57
3.18	Cell Cursors.	58
3.19	Enhanced Area Cursors.	59
3.20	TractorBeam.	60
3.21	Aides visuelles pour TractorBeam.	61
3.22	Jeu de tir.	62
3.23	Speech-Filtered Bubble Ray.	63
3.24	Speech-Filtered Bubble Ray et Bubble Ray.	63
3.25	Écran de planification et tessellation de Voronoï.	65
4.1	Trois axes de conception pour les aides visuelles atomiques.	70
4.2	Bubble Cursor et Starburst.	71
4.3	Classification des techniques d'expansion de la littérature.	73
4.4	Notation matricielle des techniques d'expansion.	74
4.5	Notation matricielle des techniques d'expansion de la littérature.	74
5.1	Classification des techniques conçues.	78
5.2	Notation matricielle des techniques conçues.	79
5.3	VTE.	80
5.4	Variantes de VTE.	82
5.5	Résultats de l'expérience pilote : Variantes de VTE.	84
5.6	MTE.	86
5.7	Expansion Lens.	87
5.8	Fonctionnement d'Expansion Lens.	88
5.9	Rope Cursor.	89
5.10	Fonctionnement de Rope Cursor.	90
5.11	Transition au changement de cible désignée avec Rope Cursor.	91
5.12	VTEtoROPE.	92
5.13	VTEtoCELL.	93
5.14	VTEtoTARGET.	93
5.15	Changement de mode implicite.	95
5.16	Tripler la taille des cibles.	97
6.1	Classification des techniques de l'expérience 1.	103
6.2	Implémentation de Bubble Ray et VTE.	104
6.3	Intégration de l'agencement de cibles de Blanch et al. [26].	105
6.4	Calcul de la taille étendue d'une cible.	107
6.5	Résultats de l'expérience 1 - Temps de sélection et taux d'erreurs.	108
6.6	Résultats de l'expérience 1 - Ray-casting : coefficients de régression.	109
6.7	Résultats de l'expérience 1 - Pointage main libre : indice de performance.	110
6.8	Classification des techniques de l'expérience 2.	112
6.9	Nouvel agencement de cibles.	116

6.10 Aides visuelles atomiques de l'expérience 2.	116
6.11 Résultats quantitatifs de l'expérience 2 - Tableau.	120
6.12 Résultats quantitatifs de l'expérience 2 - Graphiques.	120
6.13 Résultats qualitatifs de l'expérience 2.	121
7.1 Théorie de l'action.	130
7.2 Trois phases orientées-but.	131
7.3 Cell Painting et TARGET.	132
7.4 Bubble Cursor et tessellation de Voronoï.	132
7.5 Déplacement du bord de la bulle de Bubble Cursor.	133
7.6 Analyse de l'information fournie par neuf techniques d'expansion.	134
7.7 Aides visuelles de Fan Cursor.	141
7.8 Comparaison des aides visuelles de Fan Cursor.	142
8.1 Analyse de l'information fournie par 5 techniques de l'expérience 2.	147
8.2 Résultats de l'expérience 2 - Phase de transfert.	149
8.3 Résultats de l'expérience 2 - Phase de validation.	150
8.4 Analyse de l'information fournie par 5 techniques de l'expérience 3.	152
8.5 Techniques implémentées pour l'expérience 3.	154
8.6 Agencement des cibles de l'expérience 3.	155
8.7 Résultats de l'expérience 3 - Phase de transfert.	158
8.8 Résultats de l'expérience 3 - Phase de validation.	159
8.9 Résultats globaux de l'expérience 3.	160
8.10 Résultats de l'expérience 3 - Régression linéaire et loi de Fitts.	161
8.11 Résultats de l'expérience 3 - Coefficients de régression.	161
9.1 Écran de planification modifié du logiciel TKA 5.	170
9.2 Techniques implémentées pour les interviews.	171
9.3 Matériel papier pour la co-conception de techniques.	173
9.4 Résultats quantitatifs des interviews.	175
9.5 Écran de planification du logiciel TKA 6.	179
9.6 Pédale médicale de l'OrthoPilot®.	180
9.7 Bracelet Myo.	182
9.8 Tap-Kick-Click.	184
9.9 Kickables.	184
9.10 4 mouvements de base du pied et espace d'interaction.	185
9.11 Langage gestuel de Fukahori et al. [48].	185
A.1 Questionnaire SUS (System Usability Scale)	201
C.1 Rappel visuel des techniques de l'expérience 2.	208
D.1 Version simplifiée du NASA-TLX (Task Load Index).	209

Introduction

Vous souvenez-vous de cette publicité humoristique des années 2000 pour un fournisseur d'accès à l'internet ? Elle met en scène un couple de personnes âgées découvrant difficilement l'informatique. Les échecs se multipliant, le ton monte et l'homme dit à sa femme cette phrase illustre : "Mais tu ne sais pas cliquer ! Avec un doigt, comme ça !" Celle-ci ne nie pas.

Cet exemple télévisuel nous rappelle que cliquer sur de petites cibles – une icône, un bouton trop petit ou même pire : une interlettre – n'est pas si facile, que l'utilisateur soit novice ou expérimenté. À mesure que l'informatique pénètre les milieux professionnels et privés, que la variété des objets interactifs explose, que les pixels rétrécissent, le besoin d'aider les utilisateurs à désigner l'objet virtuel de leur choix se fait plus grand.

La recherche scientifique s'intéressait à ce problème avant même que la souris soit inventée, en 1964, par Douglas Engelbart [20]. Pour donner quelques chiffres, considérons par exemple le Projet Marmotte (<http://marmot.gruk.org/>), qui sur la base du volontariat, observe comment ses participants utilisent une souris. Chaque heure, un français utilisant son ordinateur parcourt en moyenne 140 mètres avec sa souris et clique en moyenne 905 fois. Tous ces clics ne servent pas qu'à sélectionner une cible à l'écran, comme une icône ou le bouton de fermeture d'une fenêtre, mais une grande part y est dédiée. Le geste de pointage, grâce auquel nous amenons le curseur sur l'objet virtuel de notre choix, est donc répété des millions de fois. Pour la recherche en Interaction Homme-Machine (IHM), faciliter ce geste est donc un enjeu majeur. Les chercheurs en IHM produisent une littérature abondante sur ce sujet avec des contributions conceptuelles et pratiques et créent chaque année de nouvelles techniques pour faciliter la sélection des cibles petites ou lointaines.

Dans les blocs opératoires aussi, l'informatique est présente, au profit des patients et des chirurgiens, qui ont vu les temps d'opération se raccourcir et les résultats cliniques s'améliorer grâce aux systèmes de Chirurgie Assistée par Ordinateur (CAO). Pour sélectionner une cible sur un tel système, les chirurgiens contrôlent un curseur sans toucher l'ordinateur (figure 1). En effet, interagir au toucher ou avec une souris corromprait l'asepsie de leurs mains gantées, indispensable pour ne pas introduire d'agent infectieux dans les plaies du patient. Les écrans tactiles et les souris ne sont pas stérilisables comme un bistouri. Sans les techniques de pointage à distance sans contact, ils seraient contraints de déléguer le pointage à un assistant ou de quitter la table d'opération à chaque fois



FIGURE 1 – Chirurgiens pointant une cible à distance avec un système de CAO. Image fournie par Aesculap.

que ce geste est nécessaire. Dans notre travail de recherche, nous poursuivons l'objectif de faciliter le pointage à distance dans ce contexte particulier qu'est le bloc opératoire pendant une intervention.

Pour faciliter le pointage, de nombreuses études proposent d'agrandir les cibles, de leur allouer temporairement, quand c'est possible, plus de pixels pour cliquer. En IHM, les solutions de ce type s'appellent des *techniques d'expansion de cibles*. Comme il y a plusieurs façons, que nous nommons *algorithmes d'expansion*, d'agrandir la zone interactive d'une cible, il y a plusieurs façons, que nous nommons *aides visuelles*, de représenter ce gain d'espace à l'écran. Or, l'un et l'autre ont démontré leur influence majeure sur l'efficacité de la sélection d'une cible et sur la satisfaction des utilisateurs. Nos travaux s'intéressent particulièrement aux aides visuelles des techniques d'expansion de cibles et tentent de répondre à la question de recherche suivante :

Comment l'aide visuelle d'une technique d'expansion de cibles impacte-t-elle les performances de sélection ?

Pour mieux cerner le contexte applicatif de nos travaux, nous présentons l'environnement chirurgical et l'application cible dans un premier chapitre. Nous présentons ensuite nos travaux et contributions en 4 parties principales (figure 2).

La première partie dresse un état de l'art du pointage, y compris à distance (chapitre 2), et des techniques d'aide au pointage (chapitre 3). Celles-ci utilisent principalement trois approches : manipuler la trajectoire du curseur (par exemple faire "coller" le curseur à la cible), effectuer la sélection en plusieurs temps ou agrandir les cibles. C'est cette

dernière approche que nous avons étudiée et adoptée pour faciliter le pointage pour les chirurgiens.

La deuxième partie se déroule en trois temps. Tout d'abord, nous présentons un espace de conception des aides visuelles que nous avons conçu pour décrire les techniques existantes et pour créer de nouvelles techniques (chapitre 4). Cet espace est défini par 3 axes de conception : la *dynamacité* de l'aide visuelle, l'*observabilité de l'expansion* et l'*élément augmenté* par l'aide visuelle pour représenter l'expansion des cibles. Nous illustrons ensuite son pouvoir génératif par la création de 9 techniques d'expansion (chapitre 5) : trois techniques augmentent l'espace entre le curseur et la cible, trois techniques augmentent le curseur et trois techniques combinent plusieurs aides visuelles au cours d'un même geste de pointage. Enfin, nous évaluons expérimentalement ces techniques et leur rationnel de conception au travers de deux expériences, aboutissant à formuler des recommandations pour les concepteurs (chapitre 6).

Dans la troisième partie, nous enrichissons la caractérisation d'une aide visuelle selon les trois axes de conception afin d'étudier les performances. Pour cela, le point de vue adopté sur une aide visuelle est celui du transfert d'information vers l'utilisateur : l'information fournie par l'aide visuelle au cours de la tâche de pointage et de sélection est alors étudiée pour définir un modèle conceptuel de performance. Le modèle de performance défini est basé sur la pertinence de l'information contenue dans les aides visuelles (chapitre 7) au regard des trois phases de la tâche : phase préliminaire, phase de transfert et phase de validation. S'en suit une confrontation de ce modèle à des résultats expérimentaux de la littérature et de deux expériences que nous avons conduites (chapitre 8).

Nous présentons dans la quatrième partie nos contributions au domaine de la chirurgie augmentée (chapitre 9), notamment aux logiciels d'Aesculap, l'entreprise qui a initié et accueilli cette thèse CIFRE.

Enfin, nous concluons ce mémoire en résumant nos contributions et en présentant les pistes de recherche à court et à long terme qui font suite à nos travaux, ainsi que notre bibliographie.

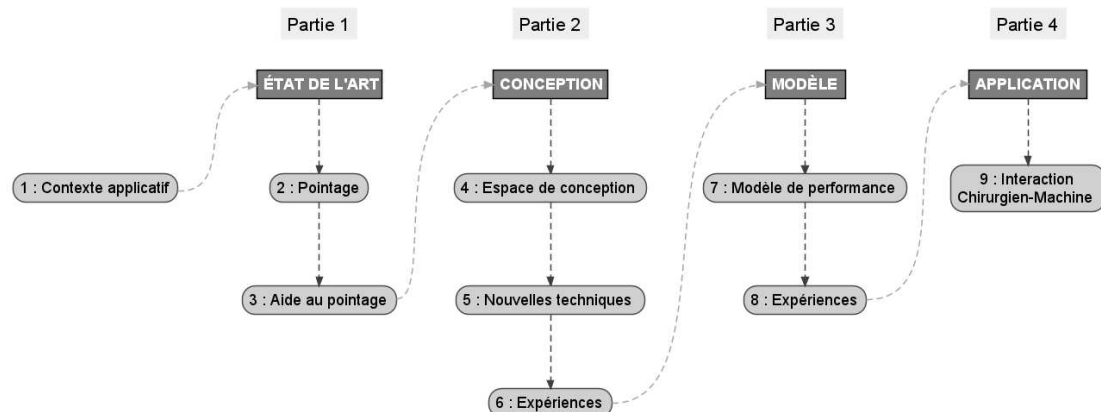


FIGURE 2 – Structure du mémoire.

Chapitre 1

Contexte applicatif et application-cible

Sommaire

Introduction	5
1.1 Contexte chirurgical	6
1.1.1 Tâche chirurgicale	6
1.1.2 Contexte d'utilisation : bloc opératoire	6
1.1.3 Utilisateur : chirurgien	7
1.2 Chirurgie augmentée	7
1.2.1 Chirurgie Assistée par Ordinateur	7
1.2.2 OrthoPilot® : un système de navigation	8
1.3 Application-cible : Total Knee Arthroplasty (TKA) version 5	11
Conclusion	13

Introduction

Le contexte applicatif de notre travail de recherche est d'améliorer l'interaction entre les chirurgiens et les systèmes de chirurgie augmentée, en contexte per-opératoire c'est-à-dire pendant les opérations chirurgicales. Nous présentons donc, dans un premier temps, la tâche chirurgicale, le contexte d'utilisation qu'est le bloc opératoire et les utilisateurs que sont les chirurgiens. Dans un second temps, nous présentons la chirurgie augmentée, le système-cible de nos travaux chez Aeculap, OrthoPilot®, et particulièrement l'écran de planification du logiciel-phare d'OrthoPilot®, TKA 5.

1.1 Contexte chirurgical

1.1.1 Tâche chirurgicale

La tâche chirurgicale est une tâche critique, au même titre que le pilotage d'un avion ou d'une centrale nucléaire. C'est un travail effectué sous une pression temporelle importante, le temps opératoire étant précieux vis-à-vis du patient anesthésié, du chirurgien lui-même et des plannings d'occupation des blocs opératoires. La tâche chirurgicale comporte également de nombreux risques d'échec, ceux-ci pouvant induire des conséquences lourdes pour le patient et pour le chirurgien notamment.

La tâche chirurgicale est une tâche experte et complexe [33]. Les chirurgiens manipulent des tissus fragiles, glissants, plus ou moins déformables, attachés les uns aux autres. L'unicité de chaque corps ajoute à la complexité de la tâche chirurgicale. En effet, les chirurgiens ont en mémoire lors d'une opération l'étude préliminaire qu'ils en ont fait : des données sur le patient, des imageries et la planification de l'opération. Mais chaque patient étant différent, chaque opération est différente. Tous les paramètres d'une opération ne peuvent pas être anticipés grâce aux techniques d'imagerie médicale et à l'étude pré-opératoire. Les chirurgiens doivent donc pouvoir s'adapter très rapidement aux nombreux changements de dernière minute qui peuvent survenir. Ils doivent de plus faire preuve d'une grande dextérité manuelle.

1.1.2 Contexte d'utilisation : bloc opératoire

Dans un bloc opératoire, un certain nombre de règles, d'usages et de contraintes techniques régissent le comportement du chirurgien et du personnel de bloc [33, 73]. Par exemple dans un bloc opératoire, le chirurgien est presque le seul maître à bord. Il a autorité sur tout le personnel de bloc sauf l'anesthésiste, au même niveau hiérarchique que lui.

La contrainte fondamentale de la tâche chirurgicale est l'*asepsie* de ceux qui y participent. Les chirurgiens et leurs assistants se lavent les mains et s'habillent avec des vêtements stériles afin d'éviter toute contamination du patient par des agents infectieux. Tout ce qui entre dans le *champ opératoire*, proche de la partie opérée du patient, est stérile : objets et personnes. Dans un bloc opératoire, la circulation de l'air est contrôlée afin qu'aucune poussière ne vienne souiller une plaie. Pour préserver cette asepsie, le chirurgien ne doit toucher ni personne ni objet non-stérile, sous peine de recommencer un processus de stérilisation de 2 heures, temps de refroidissement compris.

Une deuxième contrainte forte du bloc opératoire est le manque d'espace. En effet, de nombreuses machines et un personnel conséquent limitent les déplacements et l'espace disponible. Les chirurgiens s'éloignent donc le moins possible de la table d'opération, où tous les outils nécessaires leur sont apportés et où leur asepsie est protégée.

Cet ensemble de règles et d'usages constituent un contexte d'interaction particulier.

1.1.3 Utilisateur : chirurgien

Le chirurgien endosse la responsabilité en cas d'échec de l'opération. Cette pression et les spécificités de la tâche chirurgicale génèrent du stress. La charge cognitive et émotionnelle d'un chirurgien au bloc est considérable. Les chirurgiens et la tâche chirurgicale elle-même exigent donc des machines et du personnel de bloc la plus grande réactivité, la plus grande fiabilité et la plus grande efficacité.

Cassell (1987) [33] a révélé des traits psychologiques caractéristiques de ce métier. L'arrogance, le courage, l'activisme, le leadership et la maîtrise technique sont des traits souvent observés parmi les chirurgiens [33]. Cassell souligne le rôle important des notions de contrôle et de certitude dans le bloc opératoire. Le chirurgien doit manifester contrôle, certitude et capacité de décision : les urgences doivent être gérées, les imprévus maîtrisés et toute opportunité exploitée [33]. Les autres acteurs de l'opération, humain ou machine, doivent éviter toute action qui remettrait en cause la confiance du chirurgien dans son outillage et dans le contrôle qu'il exerce sur son environnement et sur le déroulement de l'opération. Enfin, le positivisme de l'ensemble de l'équipe de bloc est nécessaire pour permettre au chirurgien d'exercer pleinement son métier [33].

1.2 Chirurgie augmentée

1.2.1 Chirurgie Assistée par Ordinateur

La Chirurgie Assistée par Ordinateur (CAO, en anglais CAS pour Computer Assisted Surgery) est un domaine de recherche né dans les années 1990 pour des applications en neurochirurgie [22]. Aujourd'hui, ils sont plus largement utilisés, en neurochirurgie, en orthopédie et en oto-rhino-laryngologie (ORL) par exemple. La France est actuellement le deuxième fournisseur de systèmes de CAO au monde, juste derrière l'Allemagne. Un des systèmes les plus utilisés est le robot *da Vinci* (figure 1.1, https://www.intuitivesurgical.com/products/davinci_surgical_system/), qui peut être manipulé par un chirurgien pour réaliser une chirurgie endoscopique complète sans toucher le patient. De nombreux autres systèmes existent, comme le système *ROBODOC* (http://www.robodoc.com/professionals_howitworks.html), le système *CyberKnife* (<http://www.cyberknife.com/technology>) et le système *OrthoPilot®*, que nous décrivons ci-après.

Nous distinguons deux types de systèmes de CAO, ceux *passifs* et ceux *actifs*, comme établi par Dubois et al. en 1999 [43]. Un système passif de navigation a pour finalité de fournir de l'information utile aux chirurgiens durant une opération chirurgicale. Il n'effectue aucun geste à la place du chirurgien, contrairement aux systèmes actifs de co-intervention ou de télé-intervention comme le robot *da Vinci* (figure 1.1). Un système passif de navigation fournit des informations sur le patient (par exemple, des données



FIGURE 1.1 – Un chirurgien pratiquant une opération grâce au système actif de télé-chirurgie robotique da Vinci. Figure issue de <http://adexsol.com/Le-robot-chirurgical-Da-Vinci>.

chirurgicales ou chimiques, des imageries enregistrées auparavant, ou des mesures réalisées au cours de l'opération) mais il a également la capacité de réaliser des simulations à partir de ces données et de guider le chirurgien dans ses actes pour lui permettre d'être plus précis. Ces systèmes d'information permettent de se passer d'appareillages mécaniques contraignants ou de mesures supplémentaires et coûteuses, comme l'Imagerie à Raisonance Magnétique (IRM).

1.2.2 OrthoPilot® : un système de navigation

Le système OrthoPilot® (figure 1.2) est un système passif de navigation spécialement conçu pour les interventions orthopédiques. Il est développé par Aesculap, l'entreprise à l'origine de cette thèse CIFRE.

Un OrthoPilot® aide les chirurgiens orthopédistes à placer les prothèses de hanche et de genou. Pour cela, le système a besoin de construire des modèles géométriques des articulations opérées. Il permet au chirurgien de prendre et d'enregistrer des mesures prises directement sur les structures anatomiques du patient, une fois que celles-ci sont accessibles, c'est-à-dire *pendant* l'opération. Cela permet au système de construire des modèles géométriques d'une grande précision, que l'étude pré-opératoire et l'imagerie médicale ne permettent pas d'atteindre. Un OrthoPilot® connaît les dimensions précises d'un grand nombre de prothèses de genou et de hanche. Grâce à ces informations, il lui est possible de simuler la pose d'une prothèse. Le chirurgien utilise ces informations pour déterminer la taille de la prothèse, plusieurs tailles existant pour chaque modèle,

et son positionnement optimal en regard des données biomécaniques enregistrées par l'OrthoPilot®. Le meilleur positionnement pour une prothèse est celui qui respecte au mieux les propriétés mécaniques et dynamiques de l'articulation opérée. La récupération du patient n'en est que plus rapide. C'est pourquoi un OrthoPilot® permet également au chirurgien d'enregistrer des propriétés dynamiques des articulations, comme la trajectoire de certains os lorsque l'articulation est mobilisée. Enfin, il assiste les chirurgiens lors de la pose de la prothèse en guidant le positionnement des différents outils, comme les guides de coupe, les impacteurs ou les fraiseuses, qui permettent de préparer l'articulation et de fixer la prothèse.

L'intérêt de ce système est double. Premièrement, les processus opératoires sont accélérés. Ceci bénéficie aux patients vis-à-vis des anesthésies et des risques infectieux, mais aussi aux centres de soins vis-à-vis de l'occupation des blocs opératoires et enfin aux chirurgiens vis-à-vis du temps opératoire. Deuxièmement, plus la prothèse est bien placée, plus le patient récupère vite et bien, ce qui réduit en particulier les soins post-opératoires.

OrthoPilot® est composé d'une structure roulante, d'un ordinateur avec un écran tactile (pour l'interaction pré- ou post-opératoire ou en per-opératoire avec un(e) infirmier(ère) non stérile) et d'un bras articulé permettant de mobiliser un Polaris Spectra de NDI



FIGURE 1.2 – OrthoPilot® d'Aesculap. Figure fournie par Aesculap.

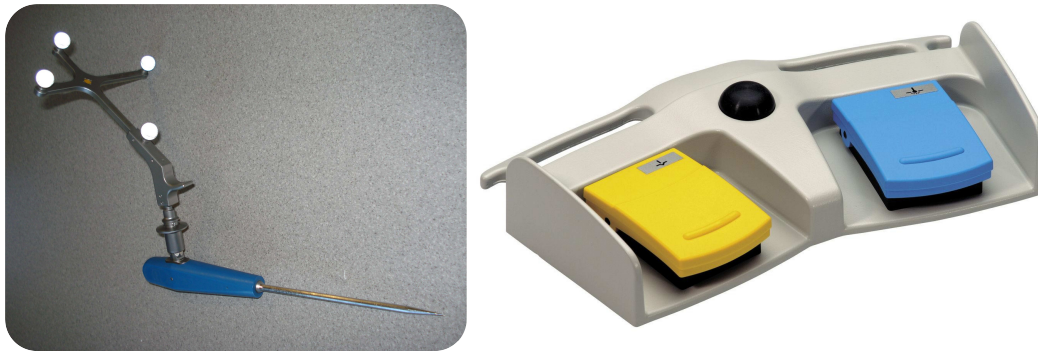


FIGURE 1.3 – À gauche, le palpeur sur lequel est clippé un objet-rigide équipé de 4 boules réfléchissant la lumière infra-rouge. À droite, la pédale médicale d'un OrthoPilot®. Elle est de plus équipée d'un arceau, non visible sur l'image, permettant de la déplacer en la soulevant avec la pointe du pied.

(<https://www.ndigital.com/medical/products/polaris-family/>), contenant deux caméras infra-rouge et un émetteur infra-rouge (figure 1.2).

Le Polaris Spectra permet de repérer et d'orienter dans l'espace les outils chirurgicaux prévus à cet effet. Des *objets-rigides* en métal, portant chacun quatre boules qui réfléchissent très bien la lumière infra-rouge, sont clippés sur ces outils également métalliques (figure 1.3). Les différentes "constellations" formées par les boules réfléchissantes sont repérées et différenciées par le Polaris Spectra, qui calcule également leur orientation dans l'espace. Les outils et les corps-rigides sont stérilisables, ce qui permet aux chirurgiens de les manipuler sans risque de contamination et d'interagir pendant l'opération avec un OrthoPilot®.

Un OrthoPilot® dispose d'un autre dispositif d'interaction : une pédale médicale à trois boutons (figure 1.3) reliée par câble à l'ordinateur. De nombreux systèmes de CAO utilisent ce périphérique pour l'interaction chirurgien-ordinateur. La pédale médicale est historiquement présente dans les blocs opératoires pour contrôler des outils comme les bistouris électriques ou, en chirurgie orthopédique, les perceuses-fraiseuses et les scies oscillantes ou réciproques. Elle présente les avantages d'être relativement mobile grâce à un arceau, simple d'utilisation, reconnaissable au toucher et surtout, elle préserve l'asepsie des chirurgiens, dont les membres inférieurs ne sont pas stériles et peuvent toucher un tel objet. Lors d'une intervention naviguée avec un OrthoPilot®, elle est utilisée de nombreuses fois pour déclencher et valider les phases d'acquisition de données. Grâce aux trois boutons et à la distinction entre appui long (dépassant 3 secondes) et appui court, le chirurgien dispose d'un vocabulaire d'interaction de 6 appuis-pédale différents.

Parmi les outils stérilisables à disposition du chirurgien, le palpeur (figure 1.3) est le plus utilisé au cours d'une intervention. Il permet au chirurgien de désigner un point de l'espace avec la pointe de l'outil et d'enregistrer ce point par un appui-pédale court. Une grande partie des données permettant la construction du modèle géométrique de l'articulation opérée est ainsi enregistrée par le chirurgien, lors d'*acquisitions statiques*. Des corps-rigides fixés au fémur et au tibia du patient au début de l'opération fournissent

un repère local pour la construction et l'utilisation du modèle géométrique. Au cours de la construction du modèle géométrique de l'articulation, des *acquisitions cinématiques* sont également réalisées. Pour ce faire, le chirurgien manipule les membres du patient, tandis que l'OrthoPilot® enregistre des données concernant l'articulation mise en mouvement ou mise en charge (comme si le patient était debout).

1.3 Application-cible : Total Knee Arthroplasty (TKA) version 5

Le système OrthoPilot® est majoritairement utilisé pour la pose de prothèses totales de genou. Les prothèses totales de genou sont constituées d'une partie fémorale et d'une partie tibiale. Le logiciel dédié à cette opération se nomme TKA pour Total Knee Arthroplasty (arthroplastie totale du genou). La version actuelle est la version 5, la version 6 sera publiée courant 2018. TKA 5 est l'application-cible de nos recherches chez Aesculap.

Le processus opératoire est partagé en 6 phases principales :

1. Paramétrage du modèle biomécanique du genou opéré
2. Coupe tibiale
3. Planification de la pose de la partie fémorale
4. Coupe fémorale
5. Vérification des propriétés biomécaniques de l'articulation
6. Finalisation de l'opération

Lors de la première phase, de nombreuses données sont enregistrées au cours d'acquisition statiques et cinématiques. Une fois le modèle biomécanique paramétré, le chirurgien effectue la coupe tibiale nécessaire à la pose de la partie tibiale de la prothèse. Ensuite, le chirurgien interagit avec un *écran de planification* lui permettant de simuler autant de fois que nécessaire le positionnement de la partie fémorale de la prothèse. Il choisit ainsi la taille finale de la prothèse et définit le positionnement à réaliser. L'OrthoPilot® le guide ensuite lors de la pose de la partie fémorale. Pour cela, l'OrthoPilot® montre le positionnement effectif et le positionnement idéal de chaque outil de préparation du fémur. Le chirurgien peut ainsi se conformer au maximum au positionnement de partie fémorale de prothèse planifié. Puis l'articulation reconstituée est testée afin de s'assurer que le positionnement de la prothèse est réussi et que ses propriétés biomécaniques sont respectées. Si ce n'est pas le cas, la finalisation de l'opération est reportée et d'autres tailles de prothèse peuvent être testées.

La planification de la pose de la partie fémorale de la prothèse se fait grâce à l'écran de planification (figure 1.4). Cet écran requiert une interaction complexe avec l'OrthoPilot® : c'est le seul écran de l'application qui nécessite la manipulation d'un curseur, comme

avec une souris. C'est pourquoi faciliter l'interaction avec cet écran en particulier est l'objectif de nos travaux de recherche présentés dans ce mémoire.

L'écran de planification (figure 1.4) est constitué de :

- 8 boutons ronds, essentiellement répartis le long des bords de l'écran. Ceux-ci permettent de contrôler l'application en général et l'enchaînement entre les différents écrans.
- Des informations générales affichées en haut et en bas de l'écran.
- Une représentation de face (vue frontale) du genou déplié du patient, à gauche, et une représentation de face du genou plié à 90°, à droite.
- Une représentation simplifiée de côté (vue sagittale) de l'articulation, en haut, au milieu de l'écran.
- Un encadré blanc permettant de choisir la taille de la prothèse, au centre de l'écran.
- Un curseur jaune en forme de viseur, en bas à droite de l'écran sur la figure 1.4. Juste au-dessus, une petite image de l'outil chirurgical utilisé par le chirurgien pour cet écran, ici le palpeur.

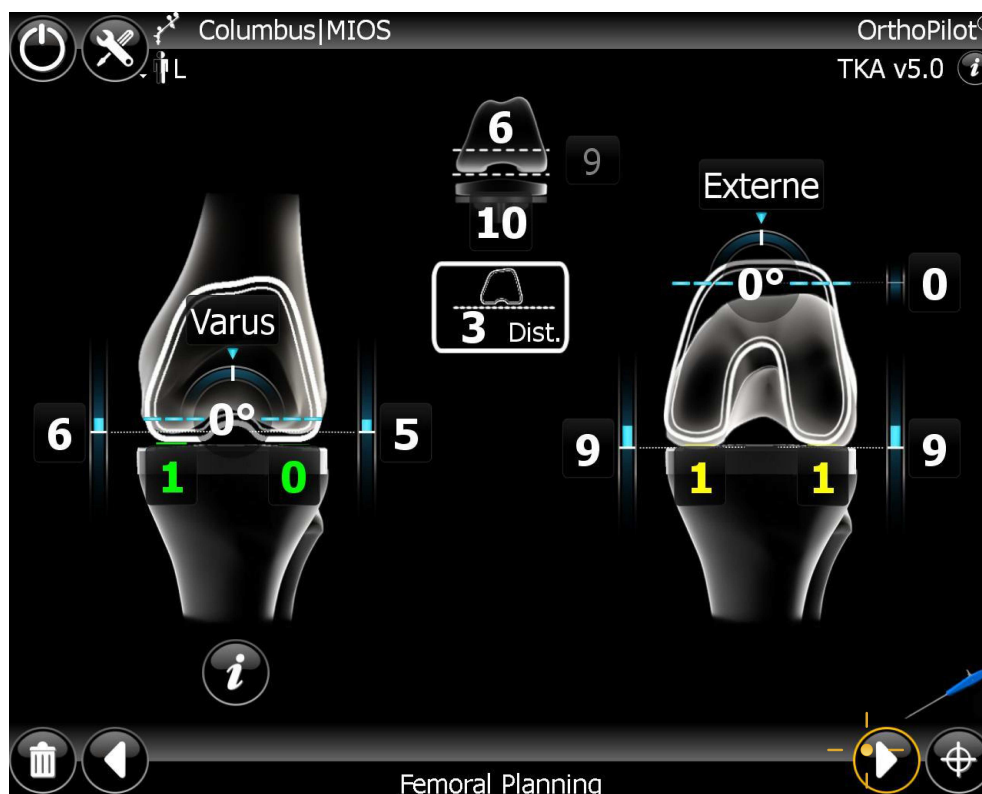


FIGURE 1.4 – Écran de planification de la pose de la partie fémorale d'une prothèse totale de genou "Colombus|MIOS", du logiciel TKA 5 d'Aesculap. Les 10 valeurs modifiables sont en blanc, les valeurs simulées sont en jaune ou vert (code couleur pouvant aller jusqu'au rouge), les 8 boutons de l'interface sont ronds et blancs. Le curseur est le point jaune entourée de 4 ailettes jaunes en bas à droite de l'écran, le bouton "suivant" est désigné. Figure fournie par Aesculap.

Les nombres en blanc présents sur cet écran désignent des données de positionnement de la partie fémorale de la prothèse, sauf le nombre au centre (3 à la figure 1.4) qui désigne la taille de la prothèse. Le chirurgien peut les sélectionner et les modifier pour simuler différents positionnements.

Les nombres en verts et en jaune constituent des données anatomiques calculées par le logiciel à partir des données de positionnement réglées par le chirurgien et du modèle géométrique de l'articulation. Le chirurgien ne peut pas les sélectionner, mais ces nombres l'informent sur les caractéristiques finales de l'articulation, si la partie fémorale de la prothèse est positionnée ainsi.

Modifier les données de positionnement en blanc, modifie également les données anatomiques, qui suivent un code de couleurs donnant une évaluation du positionnement réglé par le chirurgien (vert est un positionnement correct, jaune est un positionnement à corriger si possible).

Le but du chirurgien est de trouver le positionnement et la taille de prothèse qui optimiseront la récupération du patient et les propriétés biomécaniques finales de l'articulation opérée. Ce positionnement idéal n'est pas toujours réalisable, selon l'état initial et les caractéristiques de l'articulation opérée. Cependant, une bonne planification de la pose de la partie fémorale de la prothèse est indispensable pour la réussite de l'opération. En effet, lorsque les coupes osseuses nécessaires à la pose de la prothèse ont commencé, le retour en arrière n'est plus possible.

Pour sélectionner un bouton ou une donnée de positionnement à régler, le chirurgien amène le curseur sur l'élément choisi et valide son choix avec un appui-pédale, comme avec une souris. Cette interaction permettant de sélectionner une cible à l'écran avec un curseur est appelée *pointage*. Avec l'OrthoPilot®, le chirurgien ne dispose pas de souris car une souris n'est pas stérilisable. Il contrôle le curseur grâce à une technique de pointage à distance développée par Aesculap, le Virtual Pointer, que nous présentons au chapitre suivant. Cette technique est utilisée pour d'autres logiciels de l'OrthoPilot®, notamment pour la pose de prothèses de hanche. Faciliter le pointage pour l'OrthoPilot® constitue l'objectif applicatif de nos travaux de recherche.

Conclusion

Dans ce premier chapitre d'introduction, nous avons présenté le contexte applicatif de nos recherches : le contexte chirurgical, le système OrthoPilot® et l'écran de planification du logiciel TKA 5. Faciliter le pointage pour l'OrthoPilot® constitue l'objectif applicatif de nos travaux de recherche chez Aesculap. Au chapitre suivant, nous présentons les modèles de performance du pointage, le pointage à distance et les techniques de pointage à distance développées spécifiquement pour le contexte chirurgical, dont le Virtual Pointer d'Aesculap.

Première partie

État de l'art

Chapitre 2

Pointage

Sommaire

<i>Introduction</i>	18
2.1 Modèles de performance	19
2.1.1 Loi de Fitts	19
2.1.2 Distracteurs et modèle de Blanch et al.	19
2.1.3 Théorie cinématique et loi de puissance	20
2.1.4 Steering Law	21
2.1.5 Modèle de l'impulsion initiale optimisée	22
2.1.5.1 Sous-mouvements	23
2.1.5.2 Phases du mouvement	24
2.1.5.3 Optimisation des sous-mouvements	25
2.1.6 Loi de Hick-Hyman	26
2.2 Espace visuel et espace moteur	27
2.3 Pointage à distance	28
2.3.1 Ray-casting	29
2.3.1.1 Principe de l'interaction	29
2.3.1.2 Cinq degrés de liberté	30
2.3.1.3 Modèle de performance	31
2.3.1.4 Limites du ray-casting	32
2.3.2 Pointage main libre	32
2.4 Pointage à distance en contexte chirurgical	34
2.4.1 Interaction chirurgien-ordinateur et pointage	34
2.4.2 Virtual Pointer	36
2.4.3 Pointage main libre avec le Xtion	37
<i>Conclusion</i>	39

Introduction

Le geste de *pointage* est une tâche élémentaire de l'Interaction Homme-Machine (IHM), qu'il s'agisse d'une tablette interactive, d'un smartphone, d'une montre connectée, d'un ordinateur de bureau ou d'autres objets interactifs. C'est un geste *déictique* [74], c'est-à-dire un geste qui *montre* quelque chose, comme un enfant montre du doigt ce qu'il désire, effectué dans un objectif de communication avec un ordinateur. Typiquement, désigner une icône grâce à une souris en plaçant le curseur sur cette icône, sélectionner d'un toucher du doigt une icône sur une tablette ou un smartphone, sont des gestes de pointage.

La difficulté d'un geste de pointage est principalement déterminée par la longueur du geste à faire et la précision requise, c'est-à-dire la taille de la cible visée. La durée du geste et les erreurs possibles, comme désigner la mauvaise cible ou manquer la cible visée, dépendent de sa difficulté. Ces deux grandeurs (durée et nombre d'erreurs) définissent la performance d'un geste de pointage. L'objectif des travaux de recherche, particulièrement en IHM, est donc de réduire la durée et les erreurs des gestes de pointage dans toute leur variété.

Dans ce chapitre, nous présentons les principaux modèles de performance de la tâche de pointage, ainsi que les concepts nécessaires pour aborder au chapitre suivant les techniques d'aide au pointage. Les modèles de performance présentés permettent de prédire les performances des utilisateurs à partir de paramètres mesurables de la tâche de pointage réalisée. Nous présentons ensuite des techniques pour le pointage à *distance*, c'est-à-dire à une distance ne permettant pas de toucher l'ordinateur, ainsi que les modèles de performance et les concepts-clés qui y sont associés. Enfin, nous présentons les techniques de pointage à distance spécialement conçues pour la chirurgie augmentée, qui est le domaine applicatif de nos travaux (chapitre 1).

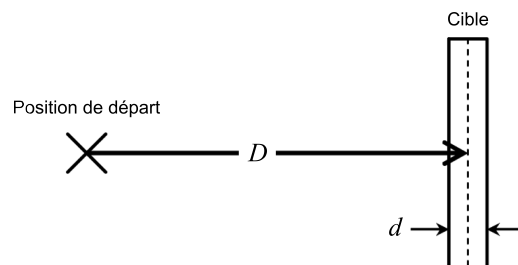


FIGURE 2.1 – Loi de Fitts [46] : distance D séparant le point de départ du centre de la cible et largeur d de la cible. Figure modifiée de [124].

2.1 Modèles de performance

2.1.1 Loi de Fitts

Fitts étudie le pointage et propose en 1954 [46] un modèle communément appelé *loi de Fitts*. Cette loi permet de prédire la durée d'une tâche de pointage rapide (T), c'est-à-dire réalisée sous pression temporelle, en fonction de la distance séparant initialement le curseur du centre de la cible (D) et de la taille de la cible (d) (figure 2.1). La formulation suivante est celle généralement acceptée depuis les travaux de MacKenzie (1992) [79] :

$$T = a + b \cdot \log_2 \left(\frac{D}{d} + 1 \right) \quad (2.1)$$

où a et b sont des coefficients calculés empiriquement pour les conditions matérielles spécifiques de l'expérimentation. La loi de Fitts établit ainsi une relation linéaire entre la durée de la tâche T et le terme logarithmique de la partie droite de l'équation 2.1. Ce terme logarithmique est nommé *Indice de Difficulté (ID)* de la tâche :

$$ID = \log_2 \left(\frac{D}{d} + 1 \right) \quad (2.2)$$

La loi de Fitts souligne ici le compromis entre vitesse et précision du geste. En effet, le terme D/d exprime mathématiquement le fait que plus le geste est rapide, moins il est précis. Une tâche requérant une grande précision nécessitera une grande application de l'utilisateur et donc un ralentissement du geste lorsque le curseur s'approche de la cible.

La loi de Fitts est très largement utilisée en IHM [12, 24, 34, 53, 54, 79, 84, 126] car même si la loi originelle est formulée pour une tâche de pointage en une dimension (figure 2.1), elle s'applique en pratique à nombre de tâches de pointage en deux dimensions [12, 34, 53, 54, 84, 126]. Plusieurs travaux [2, 26, 76, 145] ont donné lieu à des variantes de la loi de Fitts. Celles-ci décrivent des conditions particulières de la tâche de pointage [2, 76, 145] ou prennent en compte d'autres paramètres de la tâche, comme le modèle proposé par Blanch et al. [26].

2.1.2 Distracteurs et modèle de Blanch et al.

Le modèle proposé en 2011 par Blanch et al. [26] introduit un nouveau paramètre de la tâche de pointage dans la loi de Fitts : la *densité de distracteurs*. Plusieurs études [34, 53, 75] ont montré que cette grandeur impacte les performances du pointage. Les distracteurs sont des cibles potentielles de la tâche de pointage, mais elles ne sont pas la cible visée par le pointage (figure 2.2). À l'utilisation d'un ordinateur, les distracteurs sont tous les objets cliquables qui ne sont pas la cible du pointage, par exemple, dans un menu hiérarchique, les éléments que l'utilisateur ne souhaite pas sélectionner. La densité

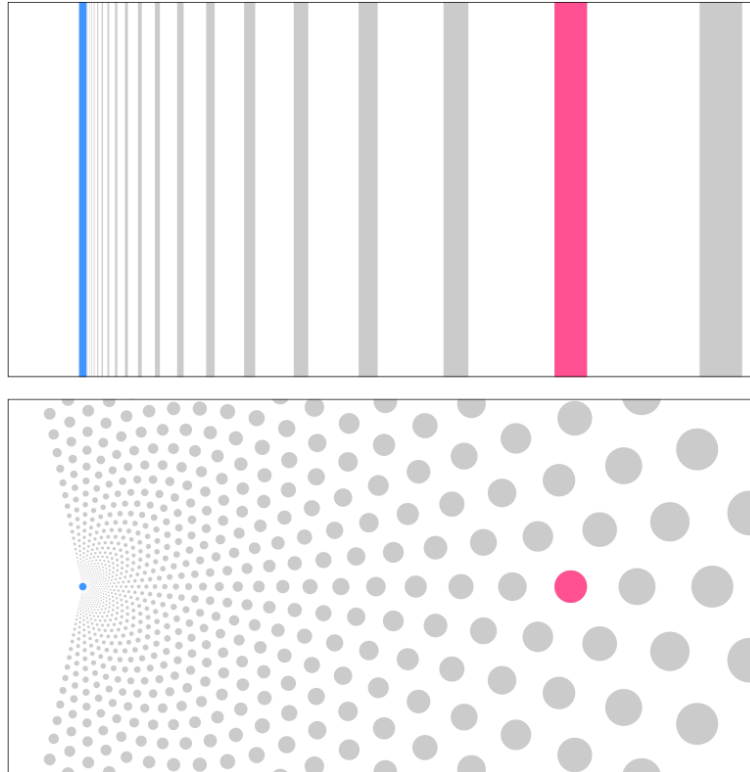


FIGURE 2.2 – Deux tâches de pointage ($ID = 4$) avec distracteurs (densité de distracteurs $\rho = 0,25$). En haut, une tâche de pointage à une dimension, en bas une tâche à deux dimensions. Les distracteurs sont en gris, la zone de départ (à gauche) est en bleu et la cible (à droite) est en rose. Figure issue de [26].

de distracteurs désigne la proportion de l'espace qu'ils occupent à l'écran (figure 2.2). La densité maximale 1 est atteinte lorsque les cibles occupent tout l'écran, tandis que la densité minimale 0 est atteinte si uniquement la cible du pointage est affichée. Blanch et al. proposent une variante de la loi de Fitts qui prend en compte la densité de distracteurs ρ pour prédire la durée d'un mouvement de pointage selon la formulation suivante :

$$T = a + b \cdot \log_2 \left(\frac{D}{d} + 1 \right) - c \cdot \log_2 \left(\frac{1}{\rho} \right) \quad (2.3)$$

où a , b et c sont des coefficients calculés empiriquement pour les conditions expérimentales. Les auteurs présentent le terme c/b comme la *sensibilité aux distracteurs* de la technique de pointage utilisée.

2.1.3 Théorie cinématique et loi de puissance

Alimi et Plamondon proposent en 1997 [107] un nouveau modèle de performance des gestes de pointage : la théorie cinématique. Pour cela, il modélise les structures anatomiques (notamment les muscles) sollicitées lors d'un geste de pointage comme des systèmes linéaires fonctionnant en synergie pour atteindre la cible. Le résultat de ce fonctionnement s'observe au travers de la vitesse du curseur, qui suit une loi *log-normale*.

Cette loi permet de décrire avec précision la grande variété des profils de vitesse observés expérimentalement [107]. Plamondon déduit de cette loi log-normale une variante de la loi de Fitts, qui est plus générique dans ses conditions d'application et qui peut être approximée par une loi de puissance :

$$\ln(T) = a + b \cdot \ln\left(\frac{D}{d}\right) \quad (2.4)$$

Dans cette formulation comme dans la loi de Fitts (équation 2.1), D et d représentent respectivement la distance initiale entre le curseur et le centre de la cible, et la taille de la cible ; a et b sont des coefficients calculés empiriquement pour chaque condition expérimentale. Le compromis vitesse-précision à l'œuvre lors d'un pointage s'exprime dans la loi de puissance comme dans la loi de Fitts, par le terme D/d dans l'équation 2.4.

2.1.4 Steering Law

Accot et Zhai proposent en 1997 la *steering law* [1] (la "loi de conduite"), un modèle de performance de la tâche de *franchissement d'un tunnel en deux dimensions*. La tâche correspondante consiste à suivre avec le curseur un chemin prédéfini d'une largeur qui peut varier (figure 2.3). Ce modèle est inspiré de la loi de Fitts et il permet de prédire le temps de parcours du chemin. Par exemple, il permet de prédire le temps de sélection d'un élément dans un menu hiérarchique nécessitant de ne pas sortir de l'élément courant pour accéder à un de ses éléments fils (figure 2.3 c). Dans le cas d'un chemin "rectangulaire" de longueur A et de largeur W (figure 2.3 a), ce modèle exprime la durée T de la tâche de parcours de la façon suivante :

$$T = a + b \cdot \frac{A}{W} \quad (2.5)$$

où a et b sont deux constantes calculées empiriquement. Cette formulation ressemble à la loi de Fitts si l'on apparente la longueur A du chemin avec la distance D à la cible et la largeur W du chemin à la taille d de la cible. Ce modèle se généralise pour des chemins non-rectilignes dont la largeur varie au cours du parcours (figure 2.3 b et c). Dans sa forme générale, la *steering law* s'exprime :

$$T = a + b \int_C \frac{ds}{W(s)} \quad (2.6)$$

où a et b sont deux constantes calculées empiriquement, T est la durée moyenne du parcours, C le chemin paramétré par s et $W(s)$ la largeur du chemin en s . L'analogie avec la loi de Fitts s'étend au compromis vitesse-précision du pointage, avec le terme A/W dans l'équation 2.5 et le terme $ds/W(s)$ dans l'équation 2.6.

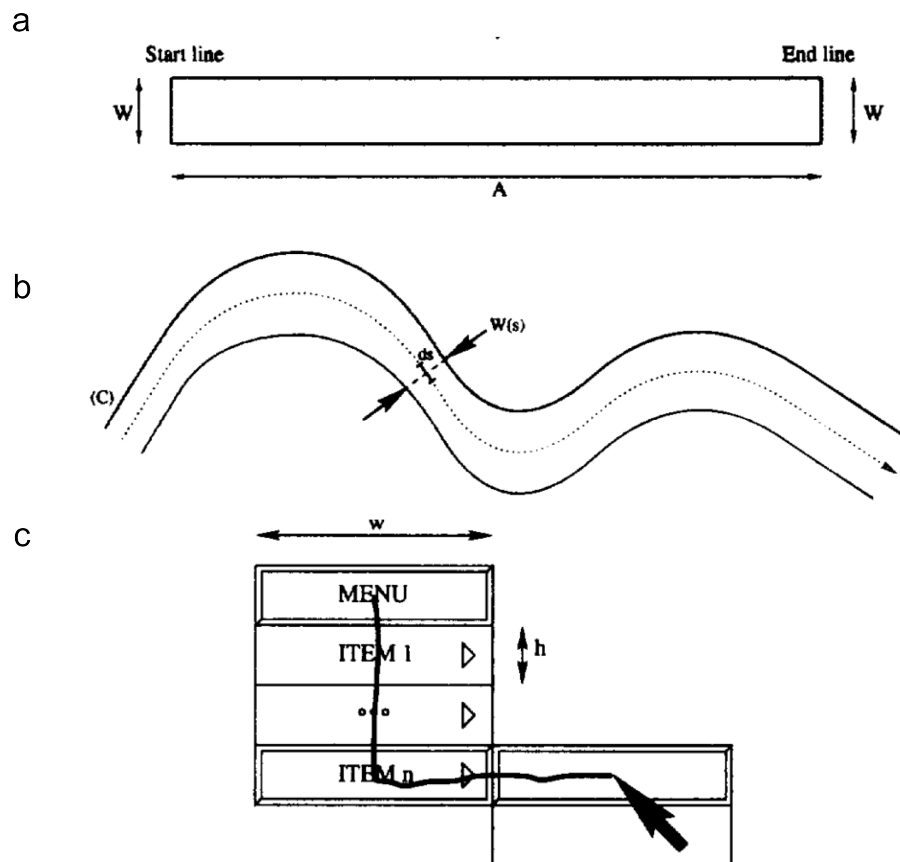


FIGURE 2.3 – Trois types de tunnels à deux dimensions : un tunnel rectangulaire (a), un chemin de largeur variable (b) et un parcours dans un menu hiérarchique (c). Figure modifiée de [1].

2.1.5 Modèle de l'impulsion initiale optimisée

Le *Modèle de l'Impulsion Initiale Optimisée* (MIIO) est proposé par Meyer en 1988 [88]. C'est un modèle du mouvement humain qui décrit les processus d'optimisation à l'œuvre dans un geste de pointage. C'est pourquoi ce modèle a beaucoup inspiré nos travaux.

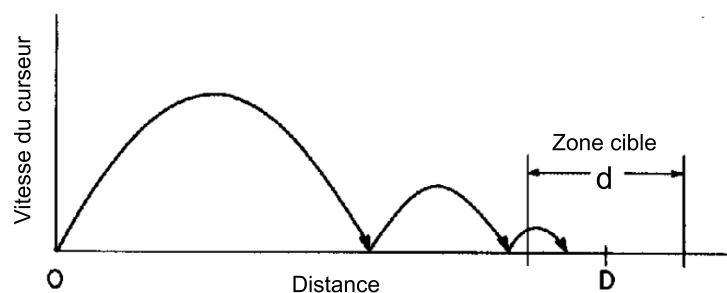


FIGURE 2.4 – Sous-mouvements successifs permettant d'atteindre la cible (de largeur d). Ces sous-mouvements s'observent sur le profil de vitesse du curseur. Figure modifiée de [88].

2.1.5.1 Sous-mouvements

Le MIIO reprend les bases posées par les travaux de Crossman et Goodeve (1983) [40]. Il considère le geste de pointage comme une succession de *sous-mouvements* qui tendent tous vers le but final : atteindre la cible (figure 2.4). En effet selon ce modèle, l'utilisateur "programme" mentalement des mouvements idéaux atteignant tous la cible à coup sûr, pour les transmettre aux différents muscles impliqués dans le mouvement, par exemple, les muscles du bras pour un mouvement de la souris. Cependant, la transmission des ordres du cerveau aux muscles est *bruitée* par des perturbations aléatoires dues à l'activité nerveuse et cérébrale : le *bruit neuro-moteur*. Le premier mouvement idéal programmé devient en réalité un mouvement imparfait, qui manque souvent la cible. Un deuxième mouvement imparfait corrige alors le premier et ainsi de suite jusqu'à ce que la cible soit atteinte (figure 2.4). Le terme de *sous-mouvements* désigne leur aspect constitutif du mouvement conscient de pointage.

Les sous-mouvements sont théoriquement détectables en observant la vitesse du curseur et ses accélérations. La figure 2.5 illustre ce fait. Elle contient trois courbes, de haut en bas : l'évolution de la *position* du curseur au cours du temps, l'évolution de la *vitesse* du curseur au cours du temps et enfin l'évolution de l'*accélération* du curseur au cours du temps. Les unités sont respectivement le degré d'angle ($^{\circ}$), le degré par seconde ($^{\circ}/s$) et le degré par seconde par seconde ($^{\circ}/s/s$) car :

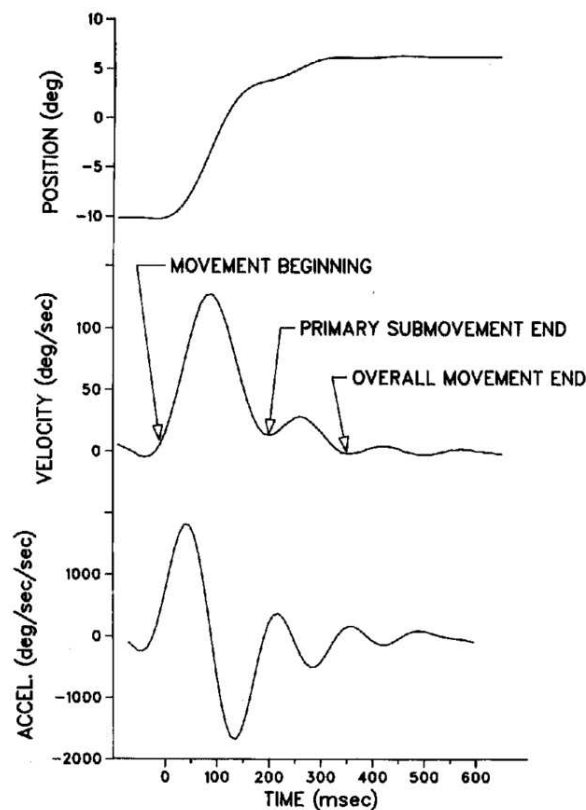


FIGURE 2.5 – Sous-mouvements détectables au travers de l'évolution de la vitesse et de l'accélération du curseur. Figure issue de [88].

- la figure représente une tâche de pointage effectuée par rotation du poignet,
- chaque grandeur représentée est la dérivée temporelle de la grandeur représentée au-dessus.

La figure 2.5 fournit également trois repères sur la courbe de vitesse du curseur, de gauche à droite : le début du mouvement de pointage, la fin du premier sous-mouvement et la fin du mouvement de pointage, lorsque la cible est atteinte. Au début de chaque sous-mouvement, la vitesse du curseur augmente jusqu'à un maximum puis diminue. Puis, au sous-mouvement suivant, le curseur accélère de nouveau. Chaque sous-mouvement crée donc un pic de vitesse (un maximum local) et un changement de signe de l'accélération du curseur, tous deux détectables *en temps réel*, c'est-à-dire en IHM, dans un laps de temps imperceptible par l'utilisateur. Le mouvement représenté par la figure 2.5 contient donc deux sous-mouvements.

En pratique, vitesse et accélération sont bruitées par les imperfections des mouvements de pointage et des dispositifs de pointage (figure 2.6). Ces grandeurs sont donc lissées pour rendre la détection fiable et stable. Cela consiste à utiliser des transformations mathématiques empruntées aux sciences de l'Automatique pour supprimer ces bruits.

2.1.5.2 Phases du mouvement

Le MIOO définit trois phases du mouvement de pointage (figure 2.7) :

1. La *phase préliminaire* précède le mouvement. L'utilisateur prend conscience du geste à réaliser et le planifie : il prépare et calibre les signaux nerveux à transmettre aux muscles concernés, en fonction de la cible à atteindre.
2. La *phase balistique* est constituée du premier sous-mouvement, le *sous-mouvement balistique*. Ce terme illustre une particularité de ce sous-mouvement : le curseur est *projeté* vers la cible, comme lors d'un tir, sans rétro-contrôle au cours du sous-mouvement balistique. Ce dernier n'est pas corrigé en fonction de la trajectoire effective du curseur, que l'utilisateur perçoit. L'utilisateur contrôle le sous-mouvement balistique *en boucle ouverte*, terme technique apporté par les sciences de l'Automatique pour désigner ce type de contrôle.

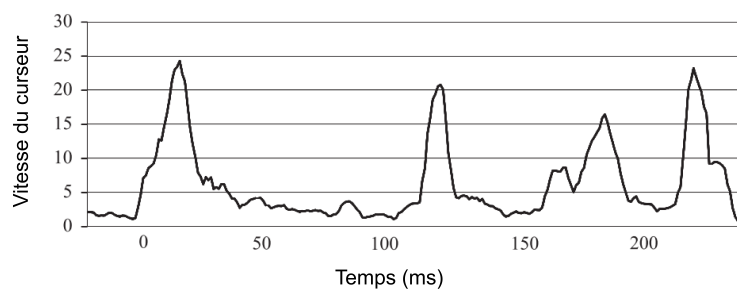


FIGURE 2.6 – Profil bruité de la vitesse instantanée du curseur. Les traitements de lissage permettent d'en obtenir des profils lissés, comme à la figure 2.5. Figure modifiée de [10].

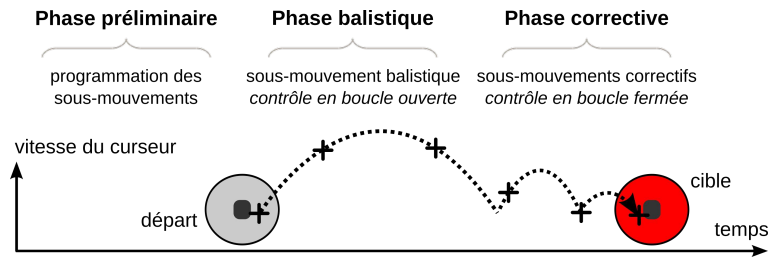


FIGURE 2.7 – Geste de pointage en trois phases : préliminaire, balistique et corrective.

3. La *phase corrective* est constituée de tous les sous-mouvements nécessaires pour corriger la position du curseur après le sous-mouvement balistique. Ces *sous-mouvements correctifs* ne sont pas nécessaires si le sous-mouvement balistique atteint la cible : c'est une phase facultative si le pointage est facile (lorsque l'*ID* est faible). Dans les autres cas, elle désigne toute la fin du geste de pointage. L'utilisateur exerce durant cette phase un contrôle *en boucle fermée* des sous-mouvements, différent du contrôle *en boucle ouverte* de la phase balistique. L'utilisateur ajuste constamment ses sous-mouvements correctifs en fonction des informations qu'ils perçoit de la situation, comme la position du curseur, sa vitesse, la distance restante à parcourir ou la direction à suivre. Le terme de *contrôle en boucle fermée* désigne cette capacité de réinjecter dans le processus de contrôle de l'information sur la situation courante.

Au cours du geste, l'utilisateur, qui exerce différents types de contrôle sur les sous-mouvements, utilise donc différemment l'information visuelle fournie par le système interactif.

2.1.5.3 Optimisation des sous-mouvements

Le principal processus d'optimisation à l'œuvre selon le MIIO consiste à programmer le sous-mouvement balistique pour que les deux premiers sous-mouvements (un sous-mouvement balistique et un sous-mouvement correctif) atteignent la cible à coup sûr, comme à la figure 2.5. Ce processus est au centre de ce modèle, d'où le nom de Modèle de l'Impulsion Initiale Optimisée. Il a été mis en évidence en comparant des tâches de pointage *à l'aveugle*, l'utilisateur ne voyant pas la cible avant de commencer le geste, avec des tâches de pointage similaires, l'utilisateur voyant la cible dès le début. Afficher ou non la cible durant la phase préliminaire modifie l'optimisation du sous-mouvement balistique. En effet, celui-ci est contrôlé *en boucle ouverte* : son optimisation dépend donc de la cible perçue avant de commencer le geste. Si, pendant la phase préliminaire, la cible n'est pas visible, cette optimisation devient approximative. Si la cible est visible, le sous-mouvement balistique est optimisé selon la taille visible de la cible. Par exemple, si le seul sous-mouvement balistique a toutes les chances d'atteindre la cible visible, il peut être programmé plus rapide et moins précis que si la tâche est difficile. Nous appelons cet effet sur le sous-mouvement balistique *effet d'optimisation du geste balistique*.

Les effets de cette optimisation sont observables si l'*ID* de la tâche est assez bas pour qu'un seul sous-mouvement correctif suffise à atteindre la cible. Si la tâche est difficile avec un *ID* élevé, l'optimisation des deux premiers sous-mouvements est compensée par les sous-mouvements correctifs suivants et l'effet d'optimisation du geste balistique n'est plus visible directement. Ce fonctionnement permet cependant à l'utilisateur d'atteindre la cible dans un temps assez court, même lorsque la difficulté de la tâche augmente.

Nous ne détaillons pas la description mathématique des processus d'optimisation du MIO [88]. Nous notons néanmoins que le MIO met en relation le terme D/d (lié à l'*ID* de la tâche et augmentant avec la difficulté) avec la probabilité qu'un geste de pointage contienne des sous-mouvements correctifs. Ainsi, ce modèle formalise mathématiquement le fait que plus l'*ID* augmente, plus la probabilité que le geste contienne plusieurs sous-mouvements correctifs augmente.

Le MIO exprime le compromis vitesse-précision du geste de pointage par la relation :

$$T = a + b \cdot \sqrt{\frac{D}{d}} \quad (2.7)$$

où a et b sont des constantes positives calculées empiriquement pour les conditions expérimentales. Les simulations de ce modèle réalisées par Meyer et al. suggèrent cependant une meilleure approximation du temps de pointage, pour les mouvements contenant de multiples sous-mouvements, par la loi de puissance :

$$T = a + b \cdot \sqrt[n]{\frac{D}{d}} = a + b \cdot \left(\frac{D}{d}\right)^{\frac{1}{n}} \quad (2.8)$$

n étant le nombre de sous-mouvements.

2.1.6 Loi de Hick-Hyman

La loi de Hick-Hyman [68, 69] modélise la tâche de sélection d'un objet parmi n objets et permet de prédire sa durée. Le dispositif expérimental initial consiste en une matrice (une grille) de lampes associées à des boutons poussoirs. Les lampes s'allument de façon aléatoire et le but est d'appuyer sur le bouton correspondant le plus vite possible. Hick conclut que la durée (RT pour *Reaction Time*) de cette tâche augmente de façon logarithmique avec le nombre n de lampes, toutes ayant une probabilité égale de s'allumer [68] :

$$RT = a + b \cdot \log_2(n) \quad (2.9)$$

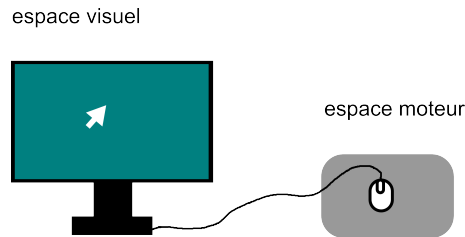


FIGURE 2.8 – Espace visuel et espace moteur.

Dans cette formulation, confirmée par les travaux de Hyman [69] sur un dispositif expérimental différent, a et b sont également des coefficients calculés empiriquement pour les conditions matérielles de chaque expérience.

2.2 Espace visuel et espace moteur

Pour étudier les techniques matérielles et logicielles de pointage, il est nécessaire de distinguer *espace moteur* et *espace visuel*.

L'espace moteur est l'espace dans lequel sont effectués les mouvements de l'utilisateur. Il s'agit en général de mouvements de la main ou des doigts. Ce peut être la zone de déplacement d'une souris (figure 2.8), l'écran tactile d'un smartphone, le pavé tactile d'un ordinateur portable ou une zone de l'espace en trois dimensions, par exemple. L'espace visuel correspond à l'affichage de l'ordinateur, là où se déplace l'avatar du dispositif de pointage : le curseur. Ce peut-être un écran fixe (figure 2.8), mobile, un ensemble d'écrans ou une partie d'un mur où l'image est projetée, par exemple. Espace moteur et espace visuel peuvent être confondus, dans le cas d'un smartphone ou d'une tablette interactive par exemple.

Une première approche voudrait que la transformation entre l'espace moteur et l'espace visuel soit l'identité, c'est-à-dire que les mouvements du curseur soient identiques aux mouvements du dispositif de pointage. C'est le cas d'un certain nombre de dispositifs de pointage, comme les écrans tactiles, pour lesquels espace moteur et espace visuel sont confondus. Cependant, il est intéressant de modifier la trajectoire du curseur, notamment en manipulant sa vitesse, pour faciliter le pointage. Cela permet notamment de limiter la taille de l'espace moteur pour minimiser le débrayage de la main, aussi noté *clutching*, c'est-à-dire le repositionnement de la souris ou du doigt pour parcourir l'espace visuel en recommençant le même geste. Un exemple de manipulation de la vitesse du curseur consiste à amplifier les grandes accélérations (phase balistique du pointage), identifiées comme une volonté de l'utilisateur de parcourir une grande distance sur l'écran, tout en amortissant les petites accélérations du curseur (phase corrective du pointage), identifiées comme une volonté de l'utilisateur de sélectionner une cible. Ainsi, l'utilisateur accède plus rapidement aux cibles distantes et il bénéficie d'une meilleure précision pour sélectionner les petites cibles.

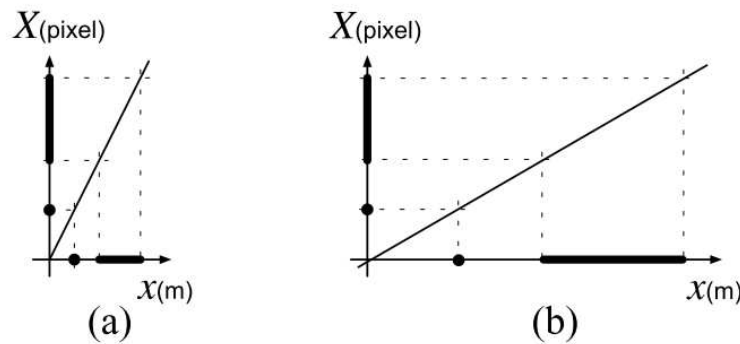


FIGURE 2.9 – Deux exemples de CD-gain constants : (a) un CD-gain haut et (b) un CD-gain bas. $X(\text{pixel})$ représente la distance dans l'espace visuel et $x(m)$ la distance parcourue dans l'espace moteur. Les lignes épaisses sur les axes des deux graphiques représentent une cible ou un déplacement dans l'espace. Figure issue de [24].

Le *Control-Display gain* (CD-gain), "le gain Moteur-Visuel", caractérise cette transformation appliquée à un instant t à la vitesse du curseur entre l'espace moteur et l'espace visuel. Cette grandeur désigne le facteur multiplicatif appliqué à la distance parcourue dans l'espace moteur par le dispositif de pointage x pour obtenir la distance parcourue dans l'espace visuel par le curseur X . Bien que la figure 2.9 et l'équation 2.10 laissent supposer une mesure du CD-gain en pixel/m, le CD-gain doit être en réalité sans dimension, afin de ne pas dépendre de la densité de pixels de l'espace visuel.

$$CD\text{-}gain = \frac{X}{x} \quad (2.10)$$

En 2008, Casiez et al. analysent en détail l'impact du CD-gain sur les performances de pointage [32]. Puis en 2011, Casiez et al. [31] étudient en détail les techniques de CD-gain dynamiques utilisées par nos systèmes d'exploitation.

2.3 Pointage à distance

Les techniques matérielles et logicielles de *pointage à distance* permettent à un utilisateur de contrôler un curseur tout en étant à quelques mètres de l'ordinateur. Bien que l'ordinateur soit hors de portée de l'utilisateur, celui-ci peut ainsi tout-de-même désigner et sélectionner des cibles à l'écran. De nombreuses techniques de pointage à distance existent [13, 15, 36, 44, 51, 81, 102, 104, 117, 133, 135], évaluées notamment par les travaux de Nancel et al. [92–95], et ont donné lieu à plusieurs taxonomies [15, 81, 133]. Nous décrivons dans cette section deux techniques communément utilisées, y compris dans nos travaux (chapitres 5 et 6) : la première est appelée "*ray-casting*" et la seconde "*pointage main libre*".

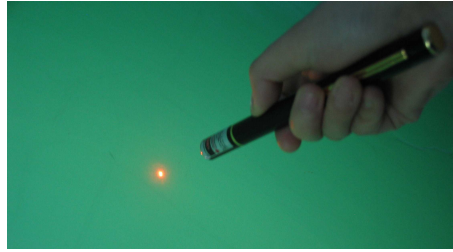


FIGURE 2.10 – Un pointeur laser permet de désigner un point d'une surface éloignée. Figure issue de http://fr.warnlaser.com/yellow_laser_pointer/polar_series.html.

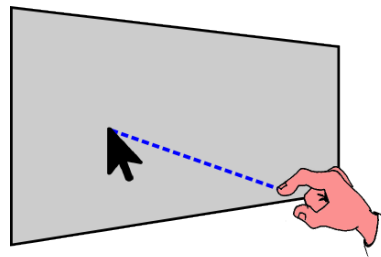


FIGURE 2.11 – Principe du ray-casting. Un axe du dispositif manipulable (ici, la main de l'utilisateur) est projeté sur l'écran et définit la position du curseur.

2.3.1 Ray-casting

2.3.1.1 Principe de l'interaction

Le *ray-casting* [76] (le "lancer de rayon") reproduit la manipulation d'un pointeur laser (figure 2.10). L'utilisateur peut alors montrer un point précis (geste déictique) sur une surface éloignée en y projetant le rayon du pointeur. Quand elle n'est pas utilisée en réalité virtuelle [10], une technique de ray-casting comprend une *technique de suivi* et un *dispositif manipulable*. La technique de suivi localise et oriente dans l'espace le dispositif manipulable. Un axe de ce dernier est ensuite projeté virtuellement sur l'écran et son point d'intersection avec l'écran définit la position du curseur (figure 2.11). En manipulant cette projection, il est possible de modifier le CD-Gain pour faciliter le pointage.

Les techniques de suivi et les dispositifs manipulables sont variés. La technique de suivi est le plus souvent optique. Cependant, la technique Myo-pointing [62] utilise une technique de suivi inertiel : un bracelet contenant des capteurs inertiels (accéléromètres et gyroscopes) que l'utilisateur porte à l'avant-bras et qui est donc également le dispositif manipulable de la technique.

Lorsque la technique de suivi est optique, le dispositif manipulable peut être *actif*, s'il est producteur d'ondes à destination du système de suivi, ou *passif*, s'il est seulement réflecteur d'ondes. Par exemple, la Wiimote (la manette de jeu de la console Nintendo Wii) est, entre autres fonctionnalités, le dispositif manipulable du système de ray-casting de la console (figure 2.12). Elle est active : elle émet des ondes à destination de la console qui localise la Wiimote dans l'espace (figure 2.12). En revanche, la technique



FIGURE 2.12 – Exemple de ray-casting : Wiimote utilisée pour un jeu vidéo. Figure issue de [15].

implémentée par Vogel et Balakrishnan en 2005 [133] utilise le système de suivi Vicon (www.vicon.com) et des marqueurs passifs fixés sur la main de l'utilisateur, qui est le dispositif manipulable de la technique. Ces techniques utilisent également différentes façons de valider la sélection de la cible, comme réaliser un geste de la main [62, 133], maintenir le curseur sur la cible pendant un laps de temps ou appuyer sur une touche du clavier ou sur la gâchette de la Wiimote [15].

2.3.1.2 Cinq degrés de liberté

Le ray-casting est une technique à 5 *Degrés De Libertés* (DDL), par opposition à la souris par exemple, qui n'en a que deux. Les DDL sont une notion issue de la Mécanique. Ils désignent les dimensions de l'espace moteur servant à situer et, éventuellement, orienter le dispositif de pointage. La souris évolue sur une table, espace à deux dimensions. Elle a donc deux DDL. Le dispositif manipulable d'un système de ray-casting évolue en général dans un espace à 3 dimensions. Pour connaître son axe de projection et donc son orientation, deux autres dimensions sont nécessaires. Ce sont ses deux axes de rotations orthogonaux à l'axe projeté. Le ray-casting est donc une technique à 5 DDL : trois translations et deux rotations. Un changement de valeur sur chacun de ces axes indépendamment des autres entraîne un changement de la position finale du curseur. Seule une rotation autour de l'axe projeté est sans effet pour la position du curseur.

La sélection d'une cible par pointage se compose de deux sous-tâches : le pointage de la cible et la validation de la sélection. Les travaux de Zhai et al. (1996) [143] ont montré que les techniques à 5 DDL comme le ray-casting peuvent favoriser la *parallélisation* des deux sous-tâches : la validation commence avant que le pointage soit terminé.

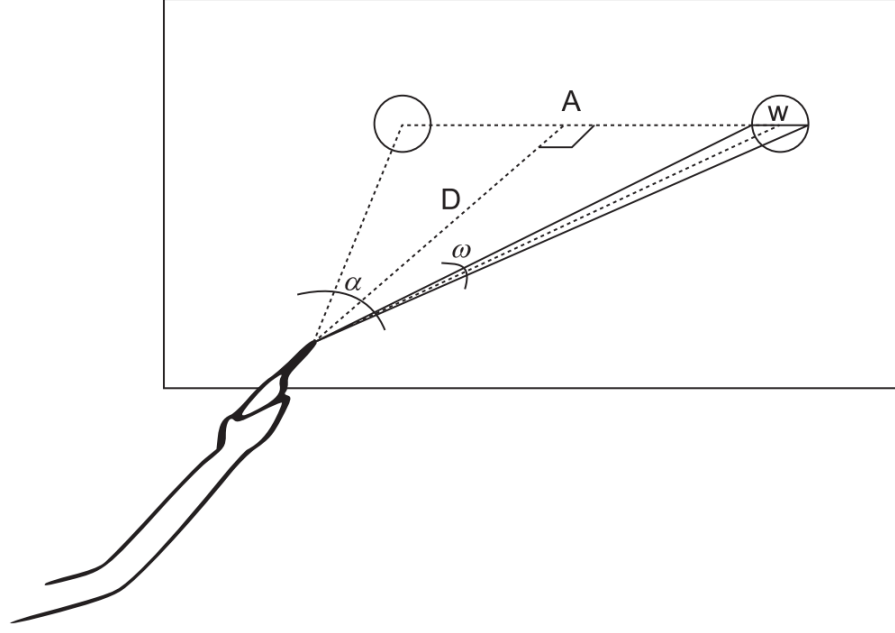


FIGURE 2.13 – Le modèle de Kopper et al. utilise les grandeurs angulaires de la tâche de pointage : α désigne la distance angulaire à parcourir jusqu'à la cible et ω désigne la taille angulaire de la cible.

2.3.1.3 Modèle de performance

Wingrave et Bowman [137] identifient la distance angulaire entre le curseur et la cible visée (α sur la figure 2.13) comme un facteur déterminant de la durée du pointage. Kopper et al. proposent sur cette base un modèle de performance du ray-casting [76] qui exprime la durée du pointage T en fonction de l'amplitude angulaire de la distance à parcourir jusqu'au centre de la cible α et de la taille angulaire de la cible ω :

$$T = a + b \cdot \left[\log_2 \left(\frac{\alpha}{\omega^k} + 1 \right) \right]^2 \quad (2.11)$$

Dans cette formulation, a , b et k sont déterminés empiriquement pour chaque condition expérimentale. Kopper et al. [76] obtiennent la meilleure modélisation avec $k = 3$. Une analogie avec la loi de Fitts est possible en considérant le terme logarithmique de l'équation 2.11 comme l'indice de difficulté ID de MacKenzie (équation 2.1). Ce terme désigne alors un Indice de Difficulté pour le ray-casting $ID_{ray-casting}$ et l'équation 2.11 représente comment la durée de la tâche T est déterminée par régression linéaire de $ID_{ray-casting}$ pour chaque condition expérimentale :

$$ID_{ray-casting} = \left[\log_2 \left(\frac{\alpha}{\omega^k} + 1 \right) \right]^2 \quad (2.12)$$

Le compromis vitesse-précision s'exprime alors pour le ray-casting par le terme α/ω^k : des gestes plus rapides seront moins précis et vice-versa.

2.3.1.4 Limites du ray-casting

Un problème inhérent au ray-casting est la perte de précision due aux tremblements naturels et inconscients de la main [76, 91]. Ce tremblement est amplifié par une distance accrue entre le dispositif manipulable et l'écran. Par exemple, une variation d'angle de 1° du dispositif manipulable déplace le curseur de 3,5 cm si l'écran est à 2 m de distance et de 7 cm si l'écran est à 4 m [91]. Ces variations involontaires limitent la précision du ray-casting. Sélectionner à 4 m de distance une cible de 4 cm, la taille typique d'une icône sur un bureau, est difficile. Plusieurs techniques, que nous présentons au chapitre suivant, ont été conçues pour pallier ce problème incontournable.

Un deuxième problème du ray-casting est la fatigue induite par une utilisation prolongée [99]. Ce problème est bien connu des chirurgiens utilisant le Virtual Pointer, que nous présentons ci-après, dont le dispositif manipulable entièrement métallique est relativement lourd.

Un troisième problème décrit par Bowman et al. [28], récurrent parmi les systèmes de pointage en général, est appelé *effet Heisenberg*. Cet effet désigne un décalage accidentel du curseur causé par l'action de validation de la sélection. Par exemple, si l'utilisateur doit cliquer avec le pied sur une pédale à boutons tout en pointant une cible avec une Wiimote, le léger déséquilibre que cause cette action peut faire bouger sa main et, par conséquent le curseur.

2.3.2 Pointage main libre

Au delà du ray-casting, nous définissons une catégorie de techniques de pointage à distance appelée *pointage main libre*. Une technique de *pointage main libre* permet à un

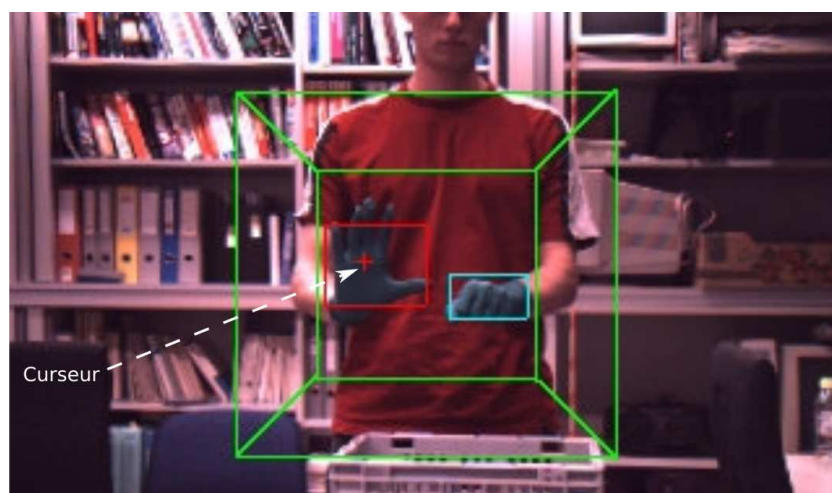


FIGURE 2.14 – Pointage main libre à 3 DDL. Suivi des mains : repérage des mains (rectangles rouge et bleu) dans un espace moteur en trois dimensions (cube vert). Ce suivi permet de réaliser une technique de pointage main libre à 3 DDL : le curseur (croix rouge) est contrôlé par la main droite de l'utilisateur. Figure modifiée de [51].

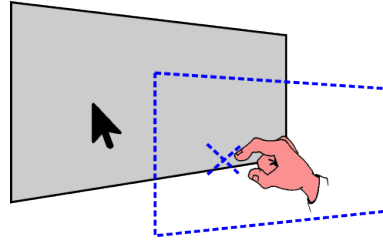


FIGURE 2.15 – Espace visuel et espace moteur d'une technique de pointage main libre.

utilisateur de déplacer un curseur à distance en déplaçant sa main vide dans l'espace [51, 58, 110, 133] (figure 2.14). Ce type de techniques de pointage à distance est particulièrement intéressante pour l'interaction chirurgien-ordinateur car elles permettent d'éviter tout contact entre les mains du chirurgien et un autre objet pour l'interaction. L'asepsie des mains du chirurgien est ainsi protégée et aucun objet stérilisable n'est requis pour l'interaction.

Une technique de pointage main libre utilise une technique de suivi des mains ("hand-tracking" en anglais) en considérant seulement les déplacements de la main de l'utilisateur pour définir la position du curseur. Le curseur se comporte comme s'il était le reflet de la main sur l'écran (figure 2.15). La main de l'utilisateur est vide mais peut-être équipée de marqueurs [133] ou d'un gant [136]. Elle est donc le dispositif manipulable d'une technique de pointage main libre (figure 2.14).

Les mouvements de la main déterminent les mouvements du curseur : si la main bouge dans une direction, le curseur bouge dans la même direction. L'orientation de la main n'est pas utilisée par le logiciel. Le pointage main libre est donc une technique à 3 DDL. L'espace moteur est un volume dans l'espace, typiquement un parallélépipède rectangle [51, 110]. Lorsque l'espace visuel est en deux dimensions [51, 52, 133], l'axe normal à l'écran, qui donne la dimension de profondeur, est ignoré ou utilisé à une autre fin que le déplacement du curseur, par exemple pour effectuer un geste de validation du pointage comme le "Push-and-Click" de Gräetzel et al. [51, 52]. S'il n'est pas utilisé du tout, c'est une technique à 2 DDL. L'espace moteur est alors une surface dans l'espace, typiquement un rectangle dans un plan vertical autour de la main de l'utilisateur (figure 2.15).

Les nombreuses techniques de pointage main libre se différencient essentiellement par leurs techniques de suivi des mains. Celles-ci utilisent des algorithmes de vision par ordinateur et des périphériques de détection divers variés, comme des caméras de plus ou moins haute résolution [51, 125, 134, 136], des caméras infra-rouges et des marqueurs placés sur la main de l'utilisateur [133], des capteurs inertiels [23, 118] ou un Kinect [44, 98, 110] par exemple (figure 2.16).

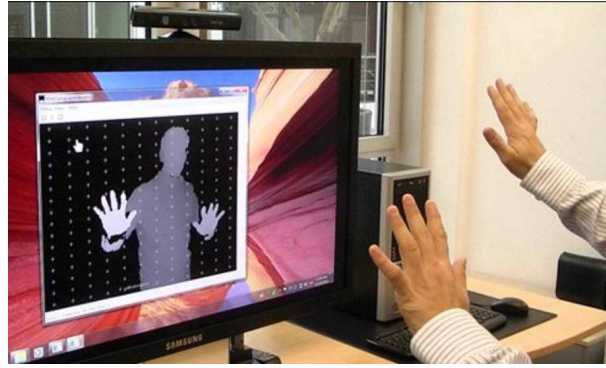


FIGURE 2.16 – Un Kinect est posé sur l'écran de l'utilisateur, qui affiche l'image de profondeur produite par le Kinect. Figure modifiée de https://www.youtube.com/watch?v=Fe_9-XW4GKQ.

Le Kinect de Microsoft (<http://www.xbox.com/fr-FR/xbox-one/accessories/kinect>) met à la portée du grand public le pointage main libre pour un coût raisonnable (environ 149\$ en 2010). Ce périphérique, développé initialement pour la console de jeu Xbox, permet notamment un suivi des mains et du squelette des joueurs, qui se tiennent de 0,8 à 2,1 m du Kinect posé au pied de l'écran de télévision. Il contient plusieurs capteurs : une caméra couleur, des microphones et, surtout, une caméra infra-rouge accompagnée d'un émetteur infra-rouge. Ce dernier couple forme une *caméra de profondeur* fournissant un *image de profondeur* en niveau de gris, chaque nuance de gris codant pour le pixel correspondant la distance qu'à pu parcourir la lumière infrarouge avant d'être réfléchi et diffusée par un objet. L'image de profondeur ainsi obtenue est un dispositif puissant pour détecter les mains et les corps des joueurs (figure 2.16) car elle contient une troisième dimension : la profondeur. De plus, elle requiert moins de traitements informatiques, comme la suppression du fond de l'image par exemple, qu'une image couleur de la scène pour être exploitable pour le suivi des mains et la détection de gestes [44, 98, 110].

Il n'existe pas à notre connaissance de modèle de performance du pointage main libre. Cependant, plusieurs travaux [94, 133] montrent que la durée du pointage avec un dispositif main libre semble suivre la loi de Fitts (équation 2.1).

2.4 Pointage à distance en contexte chirurgical

2.4.1 Interaction chirurgien-ordinateur et pointage

L'interaction chirurgien-ordinateur est soumise à l'ensemble des contraintes de la tâche chirurgicale, particulièrement à l'*asepsie* du chirurgien et à sa réticence à s'éloigner de la table d'opération (chapitre 1). Les dispositifs d'interaction classiques, comme les claviers, les souris, les écrans tactiles, les manettes de jeu ou les joysticks, ne sont pas stérilisables (immersion d'au moins 18 minutes à 134 ° C en atmosphère saturée de vapeur d'eau dans les autoclaves chargés de stériliser le matériel chirurgical).

Une solution est d'envelopper un de ces dispositifs dans un sac plastique stérile, mais deux inconvénients majeurs se posent. Premièrement, en cas de rupture du sac plastique, l'asepsie du chirurgien est menacée [64]. Deuxièmement, la manette de jeu ou la souris ainsi préparée doit rester à portée du chirurgien afin qu'il ne se déplace pas jusqu'à l'ordinateur. Ce périphérique supplémentaire encombre l'environnement proche du chirurgien, déjà occupé par les artefacts et les autres acteurs de l'intervention. De plus, la souris nécessite une surface plane assez grande. Enfin, l'emballage d'un dispositif d'interaction dans un sac plastique entraîne une surcharge de travail pour les instrumentistes. Une deuxième solution est d'interagir indirectement avec l'ordinateur, en donnant des instructions à un(e) infirmier(ère) ou un(e) assistant(e) non stérile, qui interagit avec l'ordinateur à la place du chirurgien en utilisant les périphériques de pointage classiques (souris ou écran tactile). Bien que pratiquée, avec tous les problèmes de communication possibles, les chirurgiens préfèrent néanmoins garder le contrôle direct de l'ordinateur afin d'être plus rapide et précis.

Pour ces raisons, l'interaction chirurgien-ordinateur en contexte per-opératoire se fait à distance et sans contact dans la grande majorité des cas [100]. Ainsi, le chirurgien peut rester autour de la table d'opération et l'asepsie de ses mains gantées est dûment protégée. Plusieurs études [71, 86, 87, 100] ont été menées sur le comportement des chirurgiens et de leurs assistants dans des contextes chirurgicaux particuliers, afin de comprendre leurs besoins réels en terme d'interaction à distance et l'impact de ce type d'interaction sur leur pratique professionnelle. Ces études confirment l'adéquation des techniques d'interaction à distance.

Nombreuses études [23, 44, 49, 51, 52, 112, 118, 125, 134], encouragées à partir de 2010 par l'apparition du Kinect, utilisent une technique de suivi des mains pour l'interaction chirurgien-ordinateur. Ces études focalisent principalement sur la détection de gestes et l'interaction gestuelle [23, 44, 49, 98, 112, 118]. La plupart de ces études incluent néanmoins aussi une technique de pointage main libre pour interagir avec des icônes sur un écran distant [23, 49, 51, 52, 112, 118, 125, 134]. Gratzel et al. [51, 52] conçoivent une technique de suivi des mains incluant le pointage main libre parmi ses fonctionnalités (figure 2.14). Ils utilisent pour cela une caméra stéréoscopique et leur objectif est clairement de proposer une souris "sans-contact" aux chirurgiens. Le clic est remplacé par un geste de *push* effectué en direction de la caméra, *Push-and-Click*, ou par une temporisation (intervalle de temps pendant lequel le curseur est maintenu sur la cible), *Wait-and-Click*. Wachs et al. [134] réalisent une technique d'interaction gestuelle en utilisant une caméra couleur classique pour le suivi des mains. Leur système permet au chirurgien de naviguer parmi un ensemble d'images médicales. Soutschek et al. [125] conçoivent également une souris "sans-contact" (une technique de pointage main libre à 3 DDL) avec une *caméra temps de vol* ("Time-of-Flight camera"), un périphérique utilisant le même principe que la caméra de profondeur du Kinect pour fournir une image 3D de la scène. La fonctionnalité de pilotage du curseur n'est pas la seule implémentée, l'interaction gestuelle occupant encore une grande part du potentiel de la technique.

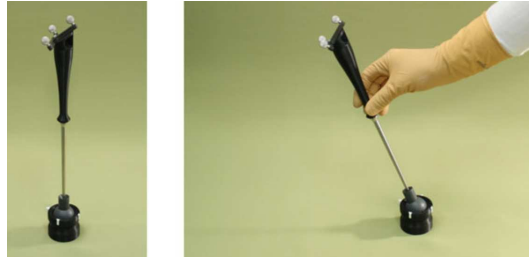


FIGURE 2.17 – Technique de joystick en bloc opératoire de Onceanu et al. [102]. Le chirurgien manipule la sonde pour contrôler un curseur et l'enfonce dans la base pour cliquer. Figure issue de [102].

Oceanu et al. [102] créent un nouveau dispositif d'interaction chirurgien-ordinateur en utilisant un objet stérilisable déjà dans les salles d'opération, une sonde chirurgicale. Ils conçoivent une base dans laquelle la sonde s'insère (figure 2.17). Celle-ci fournit un retour de force et la possibilité de cliquer avec la sonde en enfonceant celle-ci dans la base. Cette technique utilise le principe du joystick (fonction isométrique avec un signal proportionnel à la force appliquée) pour contrôler un curseur. Son étude expérimentale [102] s'intéresse aux performances de pointage : celles-ci sont moins bonnes que celles d'une souris dans un sac plastique, mais meilleures que l'interaction indirecte via un assistant.

Schwarz et al. [23, 118] présentent un système d'interaction gestuelle basé sur la pose de capteurs inertiels sur le chirurgien, réglant ainsi les problèmes d'occlusion, récurrents avec les dispositifs de détection optique comme ceux présentés précédemment. Avec un de ces capteurs placé sur chaque main, une technique de pointage main libre est ainsi réalisable. Cependant, l'objectif de l'étude est l'interaction gestuelle, le chirurgien pouvant choisir et personnaliser le vocabulaire de gestes dont il dispose.

De nombreuses techniques, postérieures à 2010, utilisent un Kinect [44, 49, 98, 112]. Gallo et al. (2011) [49] s'intéresse ainsi à l'interaction gestuelle avec les mains et les bras, Ruppert et al. (2012) [112] utilisent seulement les gestes des mains, tandis que Ebert et al. (2012) [44] proposent un système d'interaction gestuelle avec les doigts. Réaliser une technique de pointage à distance n'est pas l'objectif principal de ces études [23, 49, 51, 52, 112, 118, 125, 134]. Au contraire, Nouei et al. [98] conçoivent un système d'interaction complet incluant la fonctionnalité de pointage, l'utilisateur contrôlant alors le curseur avec le bout d'un doigt. Leur système d'interaction chirurgien-ordinateur inclut également des capteurs RFID permettant de localiser les acteurs de la chirurgie.

2.4.2 Virtual Pointer

Pour permettre au chirurgien de contrôler un curseur à distance et d'utiliser ainsi l'écran de planification présenté au chapitre 1 précédent, Aesculap a développé le *Virtual Pointer*. Cette technique de ray-casting a été certifiée et validée par les autorités de contrôle des dispositifs médicaux et chirurgicaux. Elle a fait ses preuves partout où OrthoPilot® est



FIGURE 2.18 – Chirurgiens utilisant le Virtual Pointer pour interagir avec l'OrthoPilot®. Figure fournie par Aesculap.

utilisé pour la pose de prothèses totales de genou. Le Virtual Pointer est donc le dispositif de ray-casting d'OrthoPilot® (figure 2.18).

Pour contrôler le curseur, le chirurgien manipule le palpeur (figure 1.3) en dirigeant sa pointe vers l'écran. Il valide ses sélections par un appui-pédale. Le suivi optique du palpeur, réalisé par le Polaris Spectra, permet d'atteindre une précision de ± 1 mm et 1° à la pointe du palpeur dans toutes les directions de l'espace. Au niveau logiciel, le Virtual Pointer amortit les tremblements humains en filtrant et en lissant les positions du palpeur. Cela introduit entre le geste et le déplacement du curseur une courte latence, négligeable à l'usage.

Nous utilisons le Virtual Pointer [38, 57–59] dans les évaluations expérimentales présentées dans le chapitre 6. L'utilisateur se tient alors à environ 1,8 m de l'écran. Les éléments du Virtual Pointer que sont le Polaris Spectra, avec un support déployable, la pédale médicale et le palpeur équipé d'un corps-rigide, sont utilisés et ne constituent pas un OrthoPilot® complet.

2.4.3 Pointage main libre avec le Xtion

L'équipe de Recherche et Développement d'Aesculap a implémenté une technique de pointage main libre en utilisant un dispositif proche du Kinect, le périphérique Asus Xtion Pro Live (figure 2.19 à gauche, https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/). Celui-ci dispose de capteurs strictement identiques à ceux du Kinect, sauf pour les microphones [50]. Au niveau logiciel, le Xtion est piloté par l'algorithme de suivi des mains par défaut du logiciel OpenNI 2 (<http://openni.ru/index.html>) s'appuyant sur



FIGURE 2.19 – À gauche, dispositif Asus Xtion Pro Live. À droite, suivi de la main rendu possible par l'utilisation des logiciels OpenNI 2 et Nite 2. Figure modifiée de https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/ et <https://www.youtube.com/watch?v=hl2tONHC4oI>.

l'intergiciel Nite 2 (<http://openni.ru/files/nite/index.html>). La figure 2.19 illustre, à droite, le suivi des mains ainsi réalisé. L'activation/désactivation du pointage main libre est réalisé par un geste de push de l'utilisateur. Cette technique restera cependant un prototype et ne sera pas inclut aux logiciels du système OrthoPilot®.

Nous utilisons cette technique de pointage main libre à plusieurs reprises dans nos travaux de recherche chez Aesculap, notamment pour les expériences présentées aux chapitres 5 et 6 [58]. En effet, un Xtion est aisément transportable et offre un bon champ de vision de la scène. Pour nos expériences, il est posé sur une table, au pied de l'écran, connecté par USB à l'ordinateur, et l'utilisateur est à 2 m de l'écran. Le CD-gain est égal à 1 : le curseur reproduit sans amplification ou amortissement les mouvements de la main de l'utilisateur, réalisés dans un plan vertical parallèle à l'écran (figure 2.15). Cette technique de pointage à distance relativement bon marché est cependant limitée en latence et en précision par les capteurs du Xtion [50]. Elle est moins réactive et moins précise que la technique de ray-casting Virtual Pointer.

Conclusion

Dans ce chapitre 2 sont présentés les concepts et les principaux modèles de performance du pointage et de la sélection. Ils permettent de prédire le temps de réalisation de la tâche en fonction des conditions expérimentales et des caractéristiques de la tâche à réaliser (taille de la cible ou du chemin, nombre de cibles ou distance à parcourir). Nous distinguons espace moteur, où est effectué le geste de l'utilisateur, et espace visuel, où se déplace le curseur, afin de comprendre les manipulations de CD-gain possibles pour faciliter l'interaction. Le pointage à distance fait l'objet d'une troisième partie, dans laquelle le ray-casting et le pointage main libre sont présentés en détail. Enfin, nous détaillons les techniques de pointage à distance développées spécifiquement pour le contexte per-opératoire, et en particulier celles que nous utilisons pour nos expériences présentées aux chapitres 5 et 6 : le Virtual Pointer et le pointage main libre avec un Asus Xtion Pro Live. Les bases conceptuelles du pointage posées et les dispositifs de pointage présentés, nous pouvons explorer les techniques d'aide au pointage et particulièrement les techniques d'expansion de cibles dans le chapitre suivant.

Chapitre 3

Aide au pointage

Sommaire

Introduction	41
3.1 Manipuler la trajectoire du curseur	42
3.1.1 Réduire l'espace moteur	42
3.1.2 Déformer l'espace moteur	43
3.1.3 Multiplier les curseurs	45
3.2 Agrandir les cibles	46
3.2.1 Qu'est-ce qu'une technique d'expansion de cibles ?	46
3.2.1.1 Algorithme d'expansion et tessellation de Voronoï	46
3.2.1.2 Aide visuelle : les indices de l'expansion	48
3.2.2 Dans l'espace visuel seulement	48
3.2.3 Dans l'espace moteur seulement	51
3.2.4 Dans l'espace moteur et dans l'espace visuel	51
3.2.4.1 Expansions motrices et visuelles de cibles	51
3.2.4.2 Curseurs étendus	53
3.3 Sélectionner en plusieurs temps	58
3.4 Aide au pointage à distance	60
3.5 Quelle approche pour le contexte chirurgical ?	64
Conclusion	66

Introduction

Au cours du chapitre précédent, nous avons présenté le pointage, les concepts et les modèles de performance qui y sont associés, ainsi que le pointage à distance, en particulier le ray-casting et le pointage main libre. Les fondements posés, nous présentons dans ce chapitre, de façon structurée selon trois paradigmes, les techniques logicielles visant à faciliter le pointage. Nous détaillons particulièrement les techniques d'expansion de cibles, qui agrandissent dans l'espace moteur les cibles potentielles pour faciliter leur sélection.

La grande majorité des techniques présentées sont applicables au pointage à distance, comme au pointage "ordinaire" avec la souris. La loi de Fitts (chapitre 2) définit deux façons non-exclusives de réduire la difficulté, et donc la durée, d'une tâche de pointage [12]. La première est de réduire la distance entre la position initiale du curseur et la cible. La seconde est d'agrandir la cible. Ces deux opérations offrent à l'utilisateur un meilleur compromis entre vitesse et précision du pointage : il peut accélérer son geste sans faire plus d'erreurs. Pour faciliter le pointage, les techniques existantes utilisent principalement trois paradigmes : *manipuler la trajectoire du curseur*, *agrandir les cibles* et *sélectionner la cible en plusieurs temps*.

Nous présentons une revue de ces techniques d'aide au pointage, en détaillant particulièrement les techniques d'expansion de cibles, qui constituent le sujet de nos travaux. Nous détaillons enfin notre choix des techniques d'expansion de cibles comme approche la plus adéquate pour faciliter le pointage à distance en conditions chirurgicales.

3.1 Manipuler la trajectoire du curseur

Manipuler la trajectoire du curseur pour faciliter le pointage est l'idée sous-jacente aux techniques de CD-gain dynamiques décrites au chapitre précédent (section 2.2). De nombreuses autres techniques logicielles facilitent la tâche de pointage par modification de la trajectoire du curseur. Certaines réduisent l'espace moteur, d'autres le déforment en manipulant le CD-gain, l'inertie du curseur, ou en attribuant aux cibles un pouvoir d'attraction sur le curseur. Enfin, certaines techniques multiplient les curseurs.

3.1.1 Réduire l'espace moteur

Un mouvement de *drag* ("traîner") consiste à déplacer le curseur tout en maintenant le bouton de la souris enfoncé. Il peut être initié sur le bureau de l'ordinateur (sans objet sélectionné), ou sur un objet avec l'intention de le déplacer. *Drag-and-Pop* ("traîne-et-apparaît", 2003) et *Drag-and-Pick* ("traîne-et-cueille", 2003) [18] utilisent la direction du mouvement de drag pour amener à proximité du curseur des avatars des cibles éloignées (figure 3.1). La technique à employer est déterminée par le geste de drag de l'utilisateur : s'il déplace un objet, c'est *Drag-and-Pop*, si le drag est initié sur le bureau pour un pointage, c'est *Drag-and-Pick*. *Push-and-Throw* ("pousse-et-lance", 2003) et *Drag-and-Throw* ("traîne-et-lance", 2003) [65] permettent de "lancer" une icône vers une autre, par exemple un fichier à supprimer vers la corbeille, au moment où l'utilisateur relâche le bouton de la souris. Il n'a pas besoin de réaliser le geste de *drag* complet jusqu'à la cible : l'icône lancée se déplace instantanément dans la direction que l'utilisateur a choisie. Cette approche, développée pour réduire le clutching avec des écrans de grande taille, donnera par la suite plusieurs techniques [37] dont *Push-and-Pop* ("Pousse-et-apparaît", 2005), qui combine efficacement *Push-and-Throw* et *Drag-and-Pop*. Ces techniques permettent d'omettre une partie de l'espace moteur séparant le curseur des cibles.

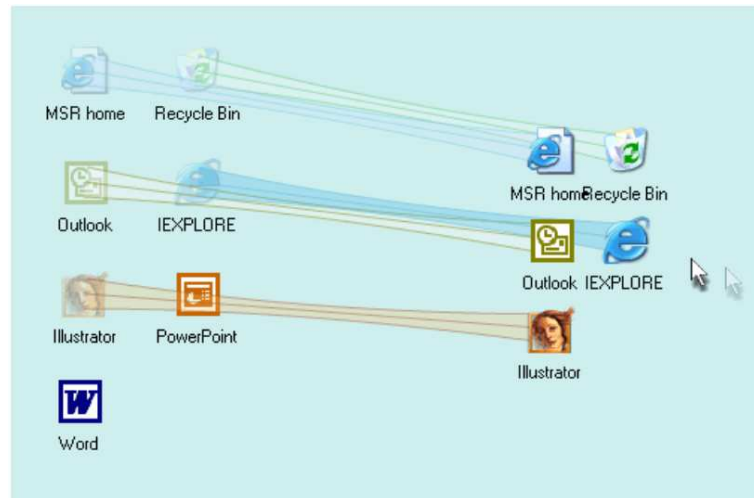


FIGURE 3.1 – Drag-and-Pick amène près du curseur un avatar de toutes les cibles qui sont dans la direction d'un mouvement de drag initié sur le bureau. Figure issue de [18].

Object Pointing (le "pointage d'objets", 2004) [56] propose d'omettre l'espace séparant les cibles. Ainsi, le curseur se déplace à l'intérieur des cibles. Lorsqu'il atteint un bord de la cible, il "saute" à la cible suivante dans la direction de son mouvement.

Toutes ces techniques sont qualifiées de *target-aware*, c'est-à-dire qu'elles ont besoin de connaître au minimum la position des cibles pour fonctionner. Les premières nécessitent également des informations sur les correspondances entre les objets déplacés et les cibles, afin de ne présenter que des cibles valables pour chaque objet déplacé. *Object Pointing* nécessite également la taille de chaque cible.

Delphian Desktop (le "bureau delphien", 2005) [11] prédit la position finale du curseur à partir de la direction et de la vitesse du sous-mouvement balistique. Le curseur est alors téléporté à la position prédite. Cette technique permet de parcourir de grandes distances à l'écran très rapidement. *Delphian Desktop* est une technique *target-agnostic* : connaître la position des cibles n'est pas nécessaire pour faciliter le pointage.

3.1.2 Déformer l'espace moteur

MAGIC Pointing (pour Manual And Gaze Input Cascaded Pointing, soit "pointage en cascade regard et main", 1999) [144] lie la position du curseur au regard de l'utilisateur grâce à un *eye-tracker*, un dispositif qui capte la direction du regard de l'utilisateur. Cela permet d'amener le curseur à proximité des cibles que l'utilisateur regarde. Le geste de pointage consiste ensuite à ajuster la position du curseur sur la cible, une grande partie de l'espace moteur ayant été parcourue par le regard, qui se déplace beaucoup plus vite que les autres parties du corps mais qui n'est pas assez stable pour désigner les petites cibles sans ambiguïté.

Semantic Pointing (le "pointage sémantique", 2004) [24] utilise la "sémantique" de chaque pixel, c'est-à-dire : appartient-il à une cible ou à un espace vide ? Cette information est utilisée pour déformer l'espace moteur en manipulant le CD-gain. Dans l'espace moteur, *Semantic Pointing* dilate les cibles et contracte les espaces vides. Dans l'espace visuel, cette technique a pour effet de ralentir le curseur quand il survole une cible et de l'accélérer lorsqu'il survole un espace vide. Cette déformation est continue : il n'y a pas de "saut" du curseur. Son mouvement est progressivement accéléré ou ralenti selon la sémantique du pixel survolé.

Les techniques *Sticky Targets* ("cibles collantes", 2006) [6, 80] manipulent également le CD-gain pour créer une impression d'adhérence du curseur aux cibles. Ces techniques sont très proches de *Semantic Pointing*, mais a contrario elles ne diminuent le CD-gain que lorsque le curseur est au-dessus d'une cible. Le curseur est ainsi ralenti et il semble coller à la cible. Ces techniques aident les utilisateurs à atteindre la précision requise pour réussir le pointage. Les erreurs de pointage sont plus rares, ainsi que les *overshoots*, c'est-à-dire les pointages aboutissant au-delà de la cible.

Les techniques de champs de force ("Force Fields", 2006) [6] et de cibles gravitationnelles ("Target Gravity", 2006) [15, 16] utilisent les lois de la physique classique pour déformer l'espace moteur. Elles attribuent aux cibles un champ de force attractif simulant, respectivement, la force électromagnétique et la force gravitationnelle. Ainsi, les cibles attirent le curseur, modifiant directement sa trajectoire. A mesure qu'il s'approche, l'attraction s'intensifie. Ces techniques facilitent ainsi la sélection des petites cibles et luttent contre les tremblements de la main et l'effet Heisenberg (chapitre précédent, section 2.3.1.4). C'est pourquoi elles sont étudiées en détail pour le pointage à distance par Bateman et al. en 2010 [15].

Glide Cursor (le "curseur glissant", 2014) [21] déforme l'espace moteur en attribuant au curseur des caractéristiques d'objets réels : une inertie, comme s'il avait une masse, et des forces de frottement sur l'espace moteur. Ainsi, le curseur continue son mouvement quand le geste de l'utilisateur s'arrête. Selon les forces de frottement simulées, le curseur "glisse" jusqu'au bord de l'écran ou ralentit progressivement. Cette technique réduit ainsi le *clutching* en minimisant le nombre de gestes de pointage pour atteindre les cibles distantes. La technique est très bien acceptée par les utilisateurs, qui la préfèrent aux accélérations logicielles du curseur, et est particulièrement adaptée aux surfaces tactiles comme les pavés tactiles des ordinateurs portables.

Tandis que *MAGIC Pointing*, *Semantic Pointing*, les techniques *Sticky Targets* et celles de champs de force sont toutes des techniques *target-aware*, *Glide Cursor* n'a pas besoin de connaître l'emplacement des cibles pour faciliter le pointage.

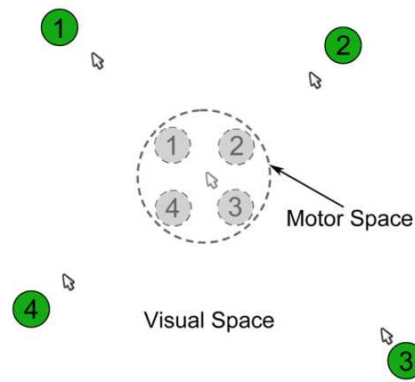


FIGURE 3.2 – Satellite Cursor permet à l'utilisateur de choisir implicitement un curseur et une cible par la direction du mouvement de pointage réalisé. Figure issue de [140].

3.1.3 Multiplier les curseurs

Multiplier les curseurs permet de faciliter le pointage en fractionnant l'espace moteur. L'utilisateur choisit alors le curseur le plus proche de la cible. *Ninja Cursor* (le "curseur-ninja", 2008) [75] ajoute des curseurs à intervalle régulier sur l'ensemble de l'écran, créant ainsi une matrice de curseurs. L'utilisateur choisit idéalement le curseur le plus proche de la cible, mais tous les curseurs se déplacent simultanément au cours du pointage. Ce comportement est source d'ambiguïtés si plusieurs curseurs survolent simultanément plusieurs cibles : quelle cible l'utilisateur veut-il désigner ? Un algorithme basé sur des temporisations (courts temps d'attente) permet de désambiguïser le choix de l'utilisateur [75].

Rake Cursor (le "curseur-rateau", 2009) [25] utilise également une grille régulière de curseurs. En revanche, la désambiguïisation est réalisée grâce à une autre modalité, le suivi des yeux, qui capte en temps réel la direction du regard de l'utilisateur. Celui-ci choisit le curseur actif en le regardant.

Satellite Cursor (le "curseur-satellite", 2010) [140] propose un curseur par cible et les place à proximité des cibles de telle façon que pour chaque curseur, la cible est dans une direction différente (figure 3.2). L'utilisateur choisit implicitement le curseur qu'il utilise par la direction de son mouvement de pointage. Ce mécanisme de désambiguïisation peut cependant être mis en défaut par certaines configurations des cibles.

Parmi ces techniques de multiplication des curseurs, seule *Satellite Cursor* est une technique *target-aware*, qui nécessite donc de connaître la position des cibles.

3.2 Agrandir les cibles

3.2.1 Qu'est-ce qu'une technique d'expansion de cibles ?

Selon le MIO (chapitre précédent, section 2.1.5), le contrôle en boucle fermée des sous-mouvements correctifs doit permettre aux utilisateurs de profiter d'une cible qui s'agrandit, même si c'est inattendu et au dernier moment, à la toute fin du pointage. Les travaux de McGuffin et Balakrishnan (2002 et 2005) [83, 84], relayés par ceux de Zhai et al. (2003) [142] et Cockburn et al. (2006) [35], ont vérifié cette prédiction. Ils ont montré qu'une telle expansion facilite le pointage, même si le curseur a déjà parcouru 90% de la distance le séparant initialement du centre de la cible. Ils ont aussi démontré que pour prédire avec la loi de Fitts (chapitre précédent, équation 2.1) la durée du pointage d'une cible étendue, il convient d'utiliser la taille étendue de la cible, d_E , au lieu de sa taille initiale d . Ce dernier point a été vérifié par de nombreuses études [34, 53, 126]. Une technique d'*expansion de cibles* facilite le pointage en agrandissant les cibles. L'expansion peut être uniquement visuelle, uniquement motrice ou visuelle et motrice. L'*algorithme d'expansion* de la technique détermine la taille étendue des cibles et l'*aide visuelle* traduit dans l'espace visuel cette expansion.

3.2.1.1 Algorithme d'expansion et tessellation de Voronoï

Un algorithme d'expansion définit une manière d'étendre les cibles et détermine leurs tailles étendues. Un exemple consiste à doubler la taille des cibles. Avec un tel algorithme, il est probable que des recouvrements se produisent entre cibles proches (figure 3.4). Ceci crée une ambiguïté lors de la désignation des pixels de ce recouvrement : quelle cible servent-ils à désigner ? La tessellation de Voronoï [41] permet de partager l'espace sans ambiguïté entre plusieurs sites (ensemble discret de points). Elle a ainsi résolu le problème du recouvrement pour plusieurs techniques d'expansion [34, 53, 77]. La représentation graphique de cette répartition de l'espace est appelé *diagramme de Voronoï* (figure 3.3).

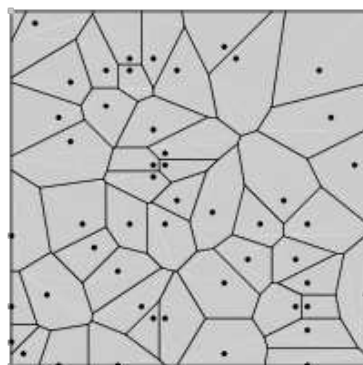


FIGURE 3.3 – Tessellation de Voronoï définissant un partage de l'espace selon l'emplacement des sites (points noirs). Le graphe obtenu, appelé diagramme de Voronoï, contient une cellule polygonale par site. Figure issue de <http://mathworld.wolfram.com/VoronoiDiagram.html>

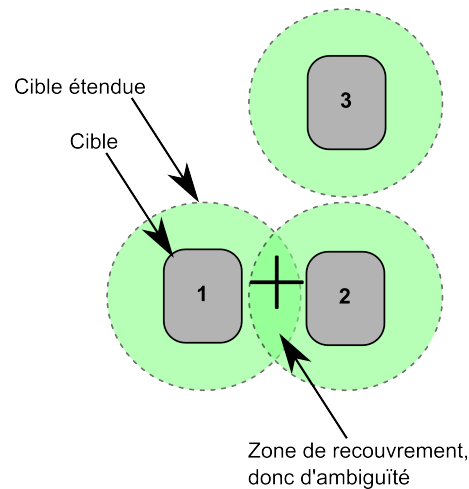


FIGURE 3.4 – Exemple de cas où l'expansion des cibles entraîne un recouvrement entre les zones étendues des cibles, et donc une ambiguïté si l'utilisateur clique dans cette zone.

L'espace est fractionné en autant de polygones convexes, appelés *cellules de Voronoï*, qu'il y a de sites. Un diagramme de Voronoï est donc composé uniquement de segments de lignes droites (figure 3.3) et les cellules englobent les sites. La tessellation de Voronoï est utilisée en expansion de cibles de la façon suivante : la position des cibles détermine les sites à considérer. Les cellules de Voronoï obtenues déterminent la taille étendue des cibles. Cliquer dans une cellule permet de sélectionner l'unique cible qui est à l'intérieur de cette cellule.

La tessellation de Voronoï présente une particularité très utile pour l'expansion de cibles : elle définit pour chaque site une cellule qui contient tous les points de l'espace plus proches de ce site que des autres sites [41]. Autrement dit, pour tous les points d'une cellule, le site le plus proche est celui qui est dans la cellule. Les cellules sont donc des zones de "plus proche voisinage" pour les sites. Cependant, la tessellation de Voronoï présente également deux inconvénients essentiels pour l'expansion de cibles :

1. L'expansion est limitée par la densité de cibles. Quand des cibles sont très proches les unes des autres, les tailles de leurs cellules sont limitées par la proximité des cibles voisines. Si la taille étendue d'une cible est à peine supérieure à sa taille initiale, le bénéfice de l'expansion est faible. Certaines techniques d'expansion, notamment *Starburst* [19] ("éclatement stellaire", 2008) que nous décrivons ci-après, compensent ce problème en utilisant un autre algorithme d'expansion.
2. Une incohérence entre l'espace visuel et l'espace moteur apparaît lorsqu'une cible déborde de sa cellule de Voronoï. En effet dans une interface graphique, les cibles ne sont généralement pas des points mais des formes géométriques, le plus souvent des rectangles. Comme la tessellation de Voronoï considère uniquement le centre des cibles et non leur forme, certaines configurations de cibles génèrent des "débordements" de cibles sur des cellules de Voronoï voisines. Cela se produit par exemple dans le cas de cibles voisines de tailles très différentes et suffisamment

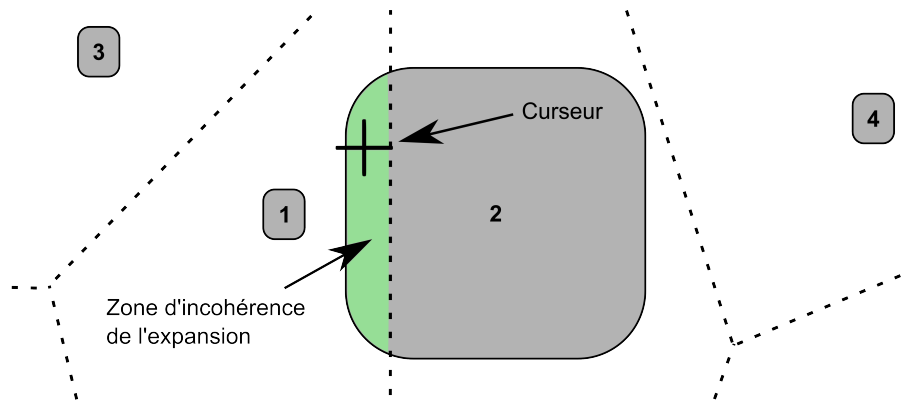


FIGURE 3.5 – Dans certaines configurations de cibles, une cible (ici la cible 2) peut déborder sur une cellule de Voronoï qui n'est pas la sienne (ici, celle de la cible 1).

rapprochées (figure 3.5). Les agencements de cibles classiques, comme le standard ISO 9241-9 d'évaluation des techniques de pointage, ne permettent pas de rendre compte de ce problème car les cibles sont de formes et de tailles similaires. Ce problème d'incohérence de l'expansion est complexe et n'apparaît que dans des configurations de cibles particulières. Il n'est abordé dans aucune des études ou des techniques d'expansion de la littérature présentées par la suite dans ce chapitre. Nous ne l'aborderons pas non plus dans ce mémoire. En effet, la quasi-totalité des expériences présentées dans ce mémoire utilisent le standard ISO 9241-9 ou un agencement de cibles issu de l'application d'Aesculap TKA 5, ceux-ci ne générant pas d'incohérence. Dans les cas où le problème apparaît (cibles temporairement agrandies au chapitre 5 section 5.1.1), c'est la cible concernée (celle qui dépasse de sa cellule de Voronoï) qui est prioritaire à la sélection et qui est désignée lorsque le curseur la survole, même s'il survole également la cellule de Voronoï d'une cible voisine.

3.2.1.2 Aide visuelle : les indices de l'expansion

Une expansion des cibles dans l'espace moteur peut être représentée de diverses manières dans l'espace visuel. L'*aide visuelle* d'une technique d'expansion désigne tous les indices visuels qui permettent à l'utilisateur d'exploiter l'expansion des cibles. Les aides visuelles existantes sont très diverses, de la représentation directe des cibles étendues [19], à la modification de l'aspect du curseur [53]. À l'opposé, pour les techniques manipulant la trajectoire du curseur (section 3.1), peu d'aides visuelles sont définies : en effet, le mouvement du curseur (ou sa duplication) suffit en général à rendre compte des déformations de l'espace moteur.

3.2.2 Dans l'espace visuel seulement

Certaines techniques d'expansion ne modifient pas l'espace moteur [35, 61, 83, 106] : l'utilisateur dispose exactement des mêmes pixels pour désigner chaque cible avec le

curseur. Les cibles sont agrandies uniquement dans l'espace visuel. Elles s'étendent alors lorsque le curseur s'approche ou les survole. Selon la loi de Fitts (chapitre 2, équation 2.1), le temps de sélection ne devrait pas être modifié par une expansion uniquement visuelle, car d et D dans l'équation 2.1 restent identiques. Cependant, l'agrandissement de la cible à la moitié de la durée du geste (lorsque 90 % de la distance D est parcourue) aide l'utilisateur à déclencher la validation plus rapidement [35]. La loi de Fitts ne permet donc pas de prédire des différences de temps de sélection dues à des différences d'aides visuelles ou à des événements se produisant après l'entrée du curseur dans la cible.

Plusieurs techniques de "lentilles", comme les *Fisheye Views* (les "vues en œil de poisson", 2002) [61] ou les *Sigma Lenses* ("lentilles sigma") de Pietriga et Appert (2008) [106], déforment l'espace visuel autour du curseur pour agrandir le cible survolée (figure 3.6). Elles proposent alors plusieurs façons de gérer les cibles voisines (figure 3.7). Tandis que la plupart des expansions visuelles de cibles sont des techniques *target-aware*, les techniques de Fisheye View et les Sigma Lenses sont *target-agnostic*.

De nombreuses autres techniques de lentilles ont été conçues, comme *WidgetLens* de Agarwal et Stuerzlinger [5] ou *Pointing Lenses* de Ramos et al. [109], mais leurs mécanismes d'expansion ne sont pas strictement que visuels et ne concernent pas forcément les cibles. Les problématiques spécifiques aux techniques de lentilles concernent majoritairement la présentation simultanée des informations centrales (du focus) et des informations de contexte [9], en particulier les problèmes de continuité visuelle.

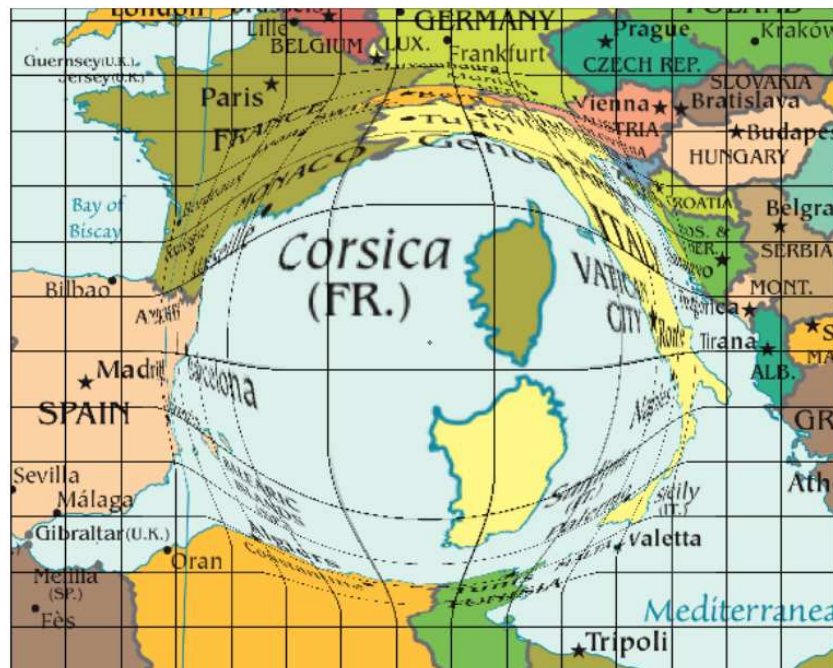


FIGURE 3.6 – Espace visuel déformé autour du curseur (ici invisible, mais situé au centre de l'image) : l'espace est grossi à proximité du curseur, contracté à moyenne distance, et intact à grande distance. Figure issue de [106].

Les expansion de cibles uniquement visuelles génèrent cependant des erreurs de sélection [35, 61] dues à la confusion entre taille visuelle et taille motrice des cibles (figure 3.7 cas *a*, *c* et *d*). Les utilisateurs croient pouvoir cliquer n'importe où dans la cible étendue pour la sélectionner, ce qui est inexact, par exemple dans les cas *a*, *c* et *d* de la figure 3.7 : l'expansion n'est *que* visuelle. Vermeulen et al. [132] parlent alors de *fausse affordance* : un objet (ici la cible) suggérant une utilisation (ici de sa taille) qui est inexacte. Le problème est équivalent au problème d'incohérence entre espace visuel et espace moteur évoqué à la section 3.2.1.1.

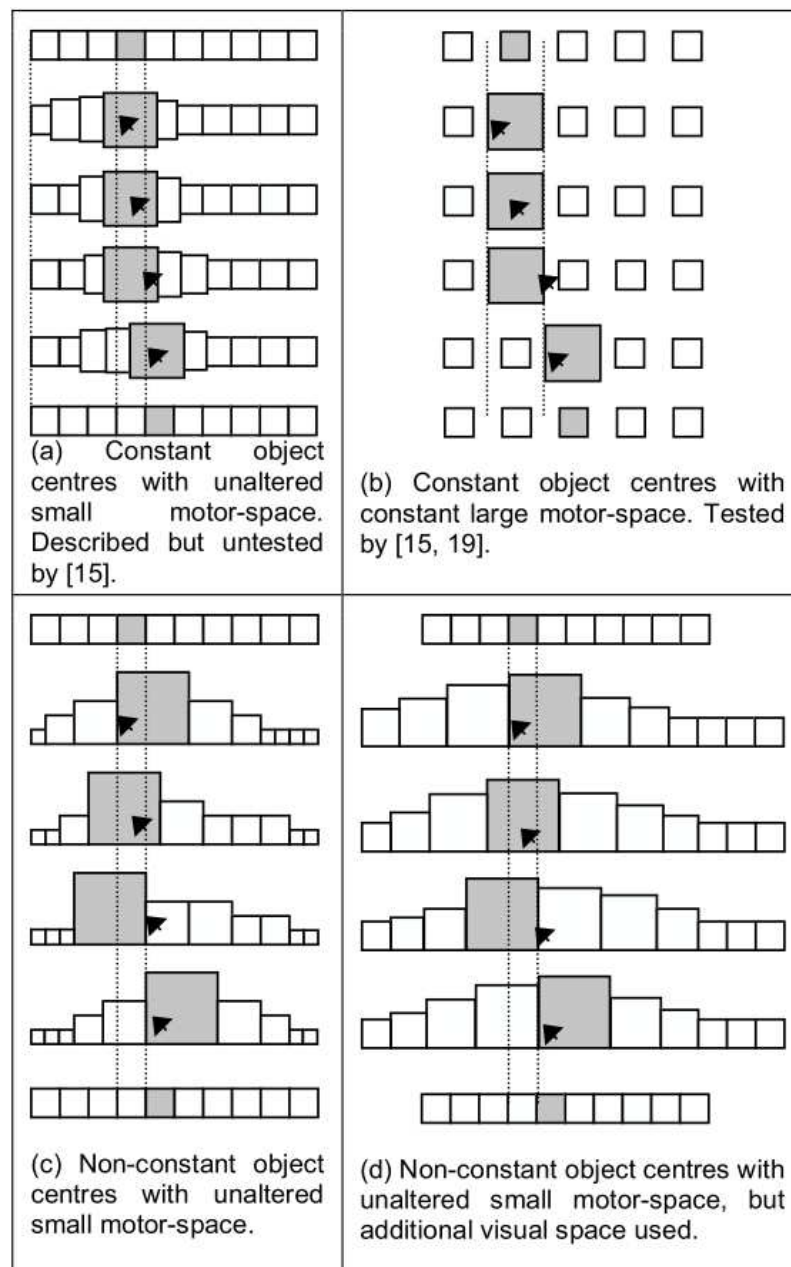


FIGURE 3.7 – Plusieurs façons d'agrandir une cible uniquement dans l'espace visuel.
Figure issue de [35].

3.2.3 Dans l'espace moteur seulement

L'expansion des cibles peut avoir lieu dans l'espace moteur sans qu'il n'y ait de trace de cette expansion dans l'espace visuel. L'utilisateur peut donc cliquer à côté de la cible et la sélectionner tout-de-même. Cependant, un problème majeur se pose alors : l'utilisateur ne peut pas savoir *où* cliquer exactement en dehors de la cible pour la sélectionner. Une telle technique d'expansion de cibles sans aide visuelle s'apparente à une technique de correction automatique des erreurs de pointage : lorsque le clic n'est pas assez précis mais est dans une certaine zone, la cible est tout-de-même sélectionnée.

Les techniques d'aide au pointage qui déforment l'espace moteur (section 3.1) réalisent une forme d'expansion de cibles uniquement dans l'espace moteur. En ralentissant le curseur au-dessus des cibles, leurs tailles dans l'espace moteur sont étendues. C'est le cas notamment des techniques Semantic Pointing et Sticky Target décrites précédemment. Avec ces techniques, seul le mouvement du curseur permet d'observer l'expansion, aucune aide visuelle supplémentaire ne montre les tailles étendues des cibles.

3.2.4 Dans l'espace moteur et dans l'espace visuel

3.2.4.1 Expansions motrices et visuelles de cibles

Lorsqu'un algorithme d'expansion et aide visuelle agissent de concert, l'utilisateur peut exploiter l'expansion de cibles en toute connaissance de cause : à chaque pixel survolé par le curseur, il sait si une cible est *désignée* ou si le curseur survole un espace vide. De plus, il sait également *quelle* cible est désignée, c'est-à-dire laquelle est sélectionnée s'il valide la sélection. *Toutes* les techniques que nous présentons ensuite dans ce chapitre sont *target-aware*.

Nous nommons *Cell Painting* et *TARGET* deux techniques d'expansion visuelle et motrice de cibles typiques. Ces deux techniques utilisent la tessellation de Voronoï pour étendre les cibles. La première, Cell Painting, consiste à afficher avec une couleur semi-transparente la taille étendue de la cible désignée, soit sa cellule de Voronoï (figure 3.8).

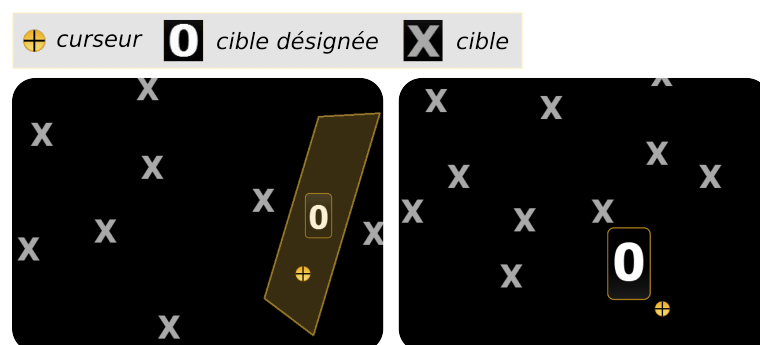


FIGURE 3.8 – À gauche, Cell Painting affiche la taille étendue de la cible. À droite, TARGET affiche la cible agrandie.

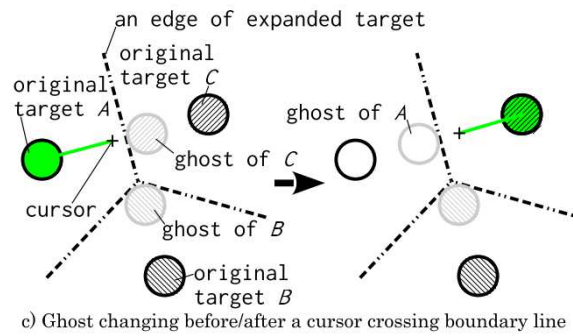


FIGURE 3.9 – Ghost-hunting utilise la tessellation de Voronoï en affichant seulement le "fantôme" de chaque cible. Figure issue de [77].

La seconde, TARGET, consiste à grossir la cible lorsque le curseur survole sa cellule de Voronoï (figure 3.8). Avec TARGET, ce grossissement de la cible ne représente pas la taille de sa cellule, mais il constitue une aide visuelle permettant de savoir quelle cible est désignée. La confusion possible entre cible grossie (dans l'espace visuel) et cible étendue (dans l'espace moteur) crée une fausse affordance susceptible de causer de nombreuses erreurs de sélection. Avec Cell Painting en revanche, l'expansion motrice de la cible est fidèlement représentée dans l'espace visuel (figure 3.8).

Ghost-hunting ("la chasse aux fantômes") [77] utilise également la tessellation de Voronoï, mais son aide visuelle est très différente de celles de *Cell Painting* et *TARGET*. Ghost-hunting affiche un avatar de chaque cible, appelé *fantôme* (figure 3.9), que l'utilisateur peut pointer pour sélectionner la cible. Ce fantôme reste dans la cellule de la cible, mais il indique à tout moment à l'utilisateur le plus court pointage possible pour sélectionner la cible. Les fantômes se déplacent donc le long des bords des cellules de Voronoï, tant que leurs cibles ne sont pas désignées (figure 3.9).

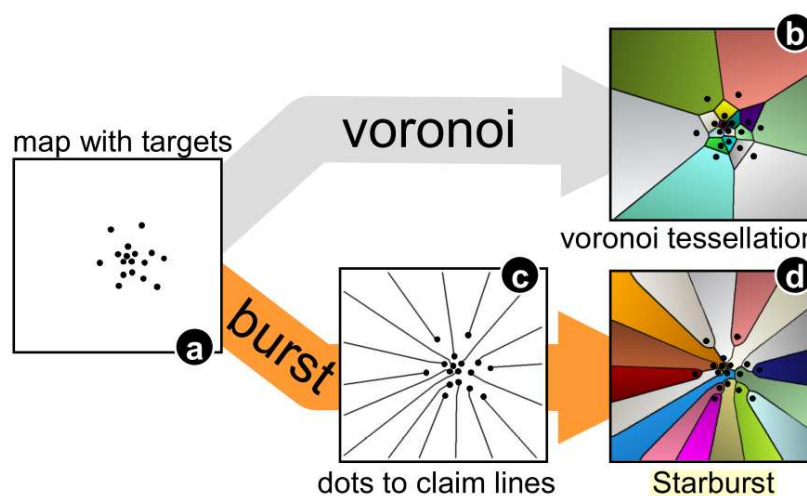


FIGURE 3.10 – Starburst permet d'obtenir des zones de sélection équilibrées entre les cibles (d), même lorsque celles-ci sont concentrées en agglomérats (a), contrairement à la tessellation de Voronoï (b). Figure issue de [19].

Starburst ("éclatement stellaire", 2008) [19] pallie aux limitations de la tessellation de Voronoï en utilisant un autre algorithme d'expansion. Cette technique s'adresse au pointage de cibles concentrées en agglomérats, c'est-à-dire très proches les unes des autres. Dans ce cas, la tessellation de Voronoï n'apporte qu'une expansion minimale des cibles au centre d'un agglomérat (figure 3.10 *b*). L'algorithme *Starburst* répartit l'ensemble de l'espace disponible entre les cibles mais, contrairement à la tessellation de Voronoï, il produit des lignes courbes et des polygones concaves (figure 3.10 *d*). Il repère tout d'abord les agglomérats de cibles et crée une ligne de fuite à partir de chaque cible de l'agglomérat (figure 3.10 *c*). Il étend ensuite ces lignes de fuites pour obtenir des zones utilisées comme des tailles étendues des cibles (figure 3.10 *d*). La répartition de l'espace moteur obtenue est plus égalitaire entre les cibles. Elle est fidèlement représentée à l'écran par l'aide visuelle. Les travaux de Baudisch et al. [19] ont montré que l'algorithme *Starburst* facilite mieux le pointage des cibles en agglomérats que la tessellation de Voronoï.

3.2.4.2 Curseurs étendus

Avec un curseur traditionnel, l'utilisateur ne désigne qu'un seul point sur l'écran, ce qui permet d'éviter les ambiguïtés de sélection : la cible désignée est celle qui contient ce point (figure 3.11 à gauche). Les *Area Cursor* (les "curseurs étendus", 1994) [72, 138, 141] sont des curseurs qui ont une aire de sélection plus grande que le point traditionnel (figure 3.11 au centre). L'utilisateur peut ainsi sélectionner une cible si elle se trouve dans l'aire de sélection du curseur. Les curseurs étendus constituent un type de techniques d'expansion visuelle et motrice des cibles : le but du pointage étant de réduire à zéro la distance D entre le curseur et la cible, attribuer une partie de D au curseur revient à attribuer une partie de D à la cible. Avec un curseur étendu comme avec une expansion visuelle et motrice, par exemple la technique *Cell Painting* présentée ci-dessus, cliquer à proximité d'une cible peut suffire à la sélectionner, le besoin de précision dans le pointage est réduit. Toutes les techniques de curseur étendu que nous présentons consistent en des expansions de cibles, les tailles des cibles étendues étant souvent calculables. Ils fournissent par leur forme une aide visuelle représentant partiellement mais avec justesse les tailles étendues des cibles. Les curseurs étendus ne manipulent pas la trajectoire du curseur.

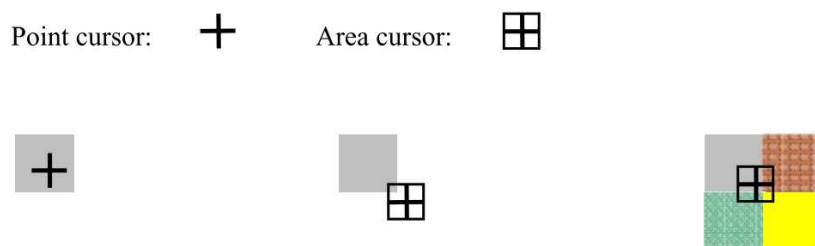


FIGURE 3.11 – À gauche, sélection avec un curseur traditionnel : le point doit être sur la cible. Au centre, sélection avec un *Area Cursor* : une partie du curseur doit être sur la cible. À droite, ambiguïté avec *Area Cursor* : la cible désignée est celle qui se trouve sous le point central (retour au curseur traditionnel). Figure issue de [12].

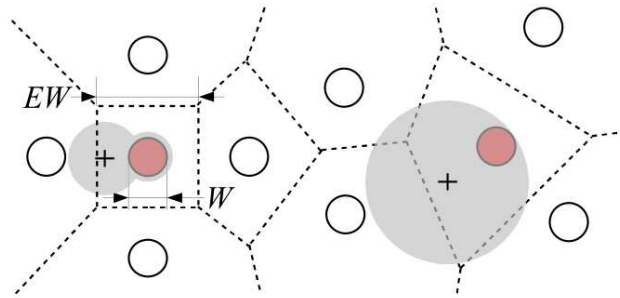


FIGURE 3.12 – Bubble Cursor utilise la tessellation de Voronoï et affiche une bulle autour du curseur (les lignes en pointillé sont uniquement explicatives). La bulle englobe tant que possible la cible la plus proche (à droite) sans toucher les cibles voisines (à gauche). Figure issue de [53].

La forme du curseur étendu, carrée par exemple [72, 138, 141], peut créer des ambiguïtés en survolant plusieurs cibles simultanément (figure 3.11 à droite). *Bubble Cursor* (le "curseur-bulle", 2004) [53] règle ce problème d'ambiguïtés en exploitant la tessellation de Voronoï à l'aide d'un curseur étendu rond ayant l'aspect d'une bulle (figure 3.12). La bulle grossit ou rétrécit pour être constamment en contact avec uniquement la cible la plus proche. L'utilisateur peut donc sélectionner la cible la plus proche, celle qui se trouve dans la même cellule de Voronoï que le curseur (figure 3.12). Le problème des ambiguïtés posé par un Area Cursor carré est donc résolu. Si possible, la bulle s'étend pour englober totalement la cible la plus proche. Sinon, c'est-à-dire si une autre cible l'empêche de s'étendre autant, la bulle limite sa taille pour ne pas toucher cette autre cible (figure 3.12).

Bubble Cursor est une technique qui a été répliquée dans de nombreuses études [34, 45, 77, 90, 126, 129, 131]. Elle a donné lieu à plusieurs variantes [4, 34, 47, 78, 90, 129, 131]. Parmi celles-ci, *Lazy Bubble* (la "bulle paresseuse", 2008) et *Cone Cursor* (le "curseur-cône", 2008) [78] utilisent aussi une bulle mais ayant un comportement différent. Le première ne s'étend pour englober la cible la plus proche, que si la deuxième cible la plus proche est assez éloignée (figure 3.13 a). La deuxième se comporte de la même manière, mais elle laisse également sa bulle s'accrocher à la dernière cible désignée (figure 3.13 b). Ces choix créent une expansion moins importante des cibles (figure 3.13 a, b et c), mais offrent la possibilité d'utiliser le curseur comme un curseur traditionnel. Leur principal avantage est d'utiliser une bulle qui grossit ou rétrécit plus progressivement que Bubble Cursor, dont les changements de taille parfois brutaux gênent les utilisateurs [34, 53, 67].

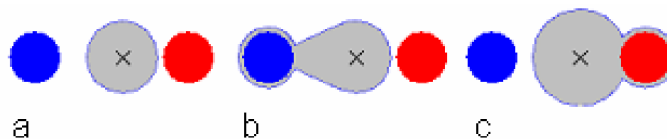


FIGURE 3.13 – (a) Lazy Bubble, (b) Cone Cursor et (c) Bubble Cursor. Figure issue de [53].

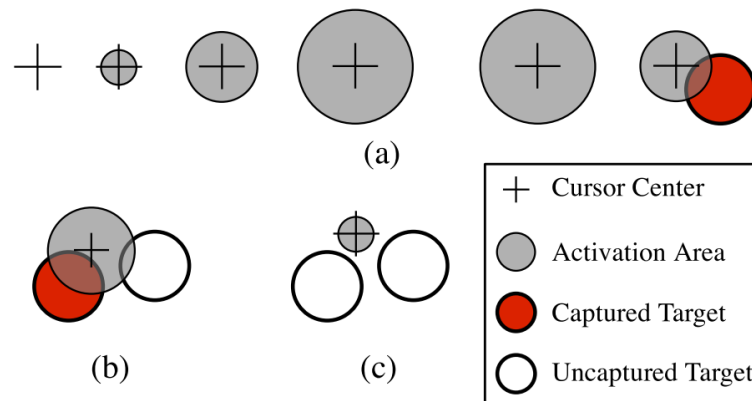


FIGURE 3.14 – (a) La bulle de Dynaspot est couplée à la vitesse du curseur. (b) En cas d'ambiguïté, la cible la plus proche est désignée (Voronoi). (c) La sélection au-pixel est possible tant que la bulle ne touche aucune cible. Figure issue de [34].

Dynaspot [34] est également une variante de Bubble Cursor. Dynaspot autorise deux modes : un mode *expansion* et un mode *normal*, équivalent au curseur traditionnel. Dans le mode *expansion*, Dynaspot est un Area Cursor rond, comme Bubble Cursor, et les cibles sont étendues. Contrairement à Bubble Cursor, la bulle de Dynaspot ne s'adapte pas à la proximité des cibles, elle grossit jusqu'à sa taille maximale quand le curseur accélère (figure 3.14 a). Quand le curseur s'arrête, elle rétrécit après un délai d'environ 300 ms (figure 3.14 a). Limiter la taille de la bulle répond à un défaut de Bubble Cursor, dont la bulle grossit exagérément quand la densité de cibles est faible, causant une gêne visuelle [34, 53, 67]. La bulle de Dynaspot crée en revanche un problème d'ambiguïté si elle recouvre plusieurs cibles (figure 3.14 b). La cible sélectionnée est alors la plus proche du curseur, au centre de sa bulle, ce qui revient à utiliser la tessellation de Voronoï pour désambiguïser la sélection.

Dans le mode *normal*, les cibles ne sont pas étendues, le curseur désigne un point de l'espace. Cela permet de sélectionner un point hors des cibles (sélection au-pixel) et d'accéder à d'autres interactions que le pointage, comme une sélection multiple (figure 3.14 c). Lorsque tout l'espace libre est partagé entre les cibles, comme avec Bubble Cursor et la tessellation de Voronoï, les interactions avec cet espace sont inaccessibles. Dans le cas d'un système d'exploitation avec un bureau par exemple, les fonctionnalités du bureau sont perdues. Le mode normal de Dynaspot évite ce problème.

La contribution de Dynaspot réside dans le mécanisme de *changement de mode implicite*. Dynaspot change de mode en fonction de la vitesse du curseur. Au-dessus d'une certaine vitesse, les cibles sont étendues, l'expansion est active. L'accélération du curseur est considérée comme une volonté de pointage. Quand le curseur s'arrête, il retourne progressivement au mode normal (figure 3.14 a). Ainsi, aucune action supplémentaire de l'utilisateur n'est requise : c'est le principe du changement de mode *implicite*. Plusieurs techniques de curseur étendu sont améliorables avec ce principe [90, 126] (chapitre 5 section 5.4).

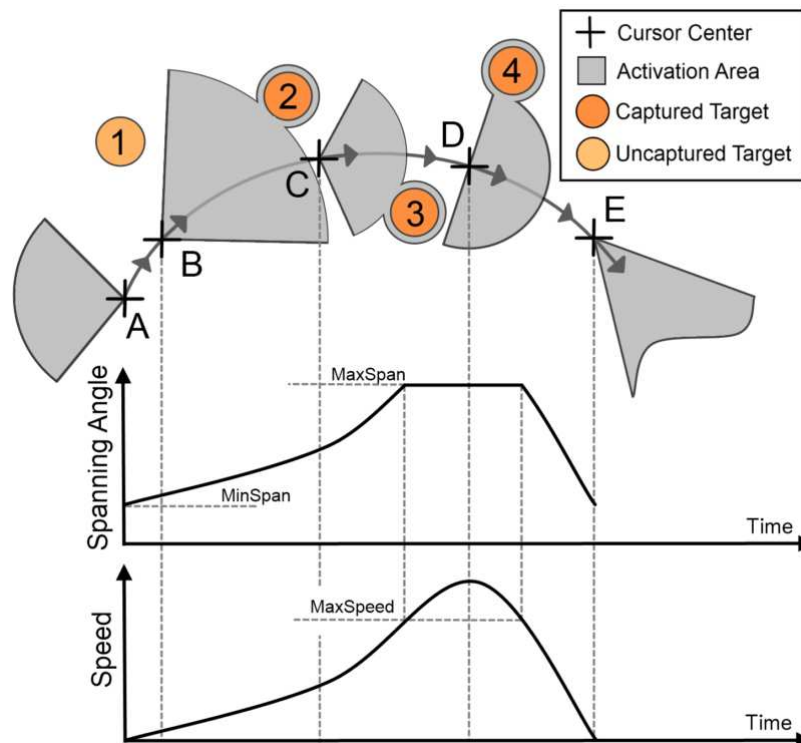


FIGURE 3.15 – L'angle d'ouverture d'IFC change dynamiquement avec la vitesse du curseur et son orientation est déterminée par la direction du mouvement. Figure issue de [126].

Parmi elles, *Implicit Fan Cursor* (IFC) (le "curseur-éventail implicite") [126] est un curseur étendu qui utilise la direction et la vitesse du curseur pour moduler son aire de sélection. Celle-ci a une forme d'éventail orienté dans la direction du mouvement (figure 3.15 : 1, puis 2, 3 et 4). La cible sélectionnée est celle que l'éventail touche. L'angle d'ouverture de l'éventail varie de 90° à 180° . Lorsque le curseur accélère, l'éventail s'ouvre de plus en plus (figure 3.15 : 2, 3 et 4), jusqu'à atteindre son ouverture maximale lorsque le curseur atteint 1200 pixel/s (figure 3.15 : 4). L'éventail s'étend en longueur pour toucher la cible qu'il désigne (figure 3.15 : 2, 3 et 4). Cette technique permet d'étendre les cibles de façon plus efficace qu'avec une tessellation de Voronoï, tout en évitant les ambiguïtés. Fan Cursor a été testé avec trois aides visuelles différentes (figure 3.16) :

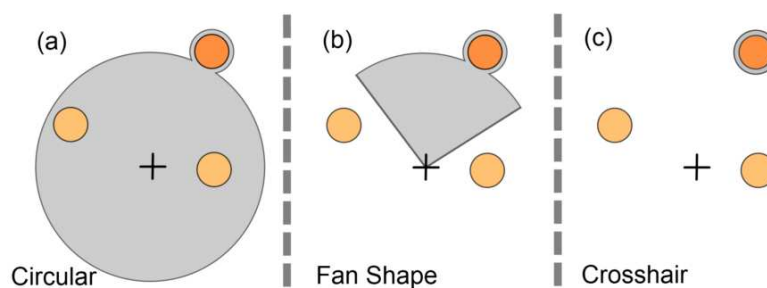


FIGURE 3.16 – Les 3 aides visuelles de Fan Cursor. Figure issue de [126]

1. l'éventail complet
2. une bulle, sur le principe de Bubble Cursor, mais adaptée aux spécificités d'IFC
3. pas d'aide visuelle, le curseur ne change pas d'aspect

Dans les trois cas, la cible désignée est mise en valeur par un contour coloré. Les résultats [126] ont montré que la troisième aide visuelle, la plus minimaliste, offre les meilleures performances, tant en vitesse de sélection qu'en nombre d'erreurs commises. Fan Cursor devient finalement Implicit Fan Cursor, l'éventail ayant disparu de l'espace visuel.

Bubble Lens (la "lentille-bulle", 2014) [90], autre variante de Bubble Cursor, propose une expansion en deux temps. Avec la tessellation de Voronoï, la densité de cibles est un critère majeur de la tâche de pointage car elle est intimement liée à la taille étendue des cibles. Tant que cette densité n'est pas trop élevée, Bubble Lens reste un Bubble Cursor et facilite efficacement le pointage (figure 3.17 a). Lorsque les cibles sont rassemblées en agglomérats denses, une deuxième expansion se produit en fin de geste pour faciliter la sélection (figure 3.17 c). Ainsi, Bubble Lens est une alternative de type "curseur étendu" à l'algorithme Starburst, permettant de traiter les cas où la tessellation de Voronoï n'apporte pas de bénéfice significatif.

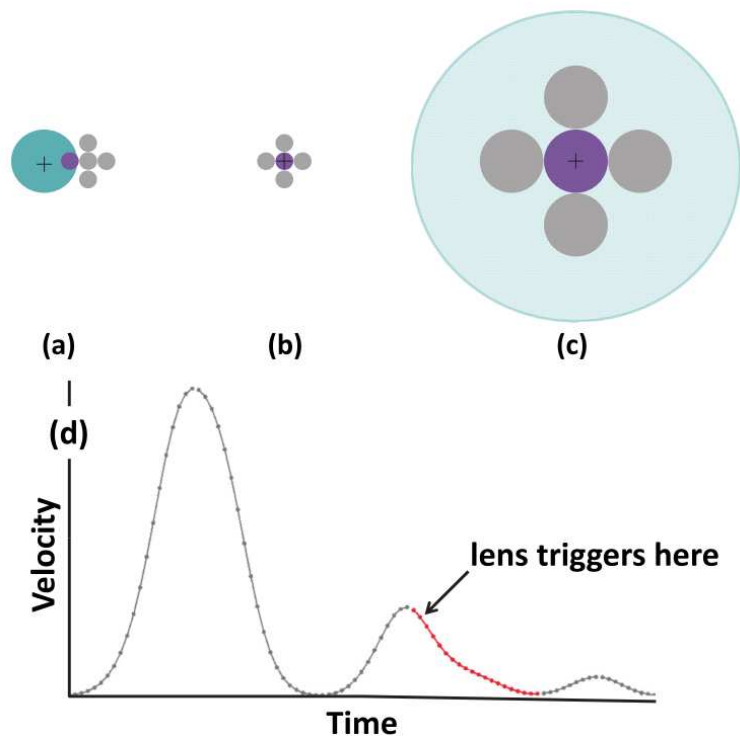


FIGURE 3.17 – (a) Au début du geste de pointage, Bubble Lens correspond à un Bubble Cursor. (b) Pour des cibles petites et rapprochées, Bubble Cursor n'est pas meilleur qu'un curseur traditionnel. (c) Automatiquement, Bubble Lens étend les cibles une seconde fois dans l'espace visuel et moteur. (d) La deuxième expansion est déclenchée sur la pente descendante du premier sous-mouvement correctif détecté. Ainsi, tandis que l'utilisateur corrige son mouvement vers la cible, l'expansion se déclenche automatiquement. Figure issue de [90].

La deuxième expansion est locale : elle concerne une zone ronde autour du curseur, d'où la métaphore de la lentille optique. Les cibles contenues dans la lentille sont agrandies et un algorithme préserve leur distance *bord-à-bord*. Ainsi, bien que les cibles sont étendues une deuxième fois, cela ne génère pas de problème d'ambiguïté (figure 3.17 c).

Cette deuxième expansion est déclenchée à la fin du geste seulement quand cela est nécessaire, grâce à une analyse constante des mouvements du curseur (figure 3.17 d). L'accomplissement d'une tâche de pointage difficile requiert la présence d'une phase corrective (chapitre 2 section 2.1.5), et donc de sous-mouvements correctifs. Ces sous-mouvements correctifs sont détectés sur un profil de vitesse lissé du curseur. Lorsque ce profil de vitesse présente des sous-mouvements correctifs, le ralentissement du premier sous-mouvement correctif est l'événement qui déclenche la deuxième expansion. Cette combinaison d'expansions repose donc uniquement sur l'analyse cinématique du curseur. Les études expérimentales [90] ont montré que Bubble Lens offre un gain de performance significatif par rapport à Bubble Cursor dans les cas difficiles de cibles petites et denses.

3.3 Sélectionner en plusieurs temps

Certaines techniques facilitent le pointage en répartissant sa difficulté par plusieurs sélections successives : elles proposent une succession de pointages plus faciles, jusqu'à ce qu'il ne reste plus d'ambiguïté sur la cible choisie.

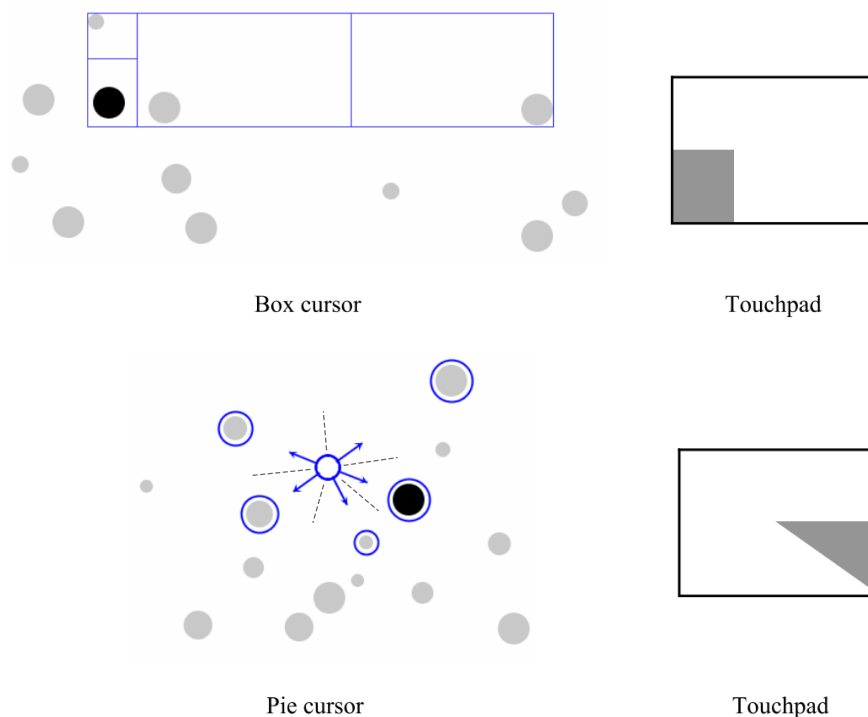


FIGURE 3.18 – À gauche, en haut, Box Cursor, et en bas, Pie Cursor. Dans les deux cas, la cible voulue est en noir. Elle est sélectionnée par un toucher dans la zone correspondante du pavé tactile de l'ordinateur portable (à droite). Figure modifiée de [67].

Par exemple, les *Cell-Cursor* (les "curseurs-cellules", 2007) [67] regroupent les cibles dans des cellules rectangulaires (variante *Box-Cursor*, le "curseur-boîte") ou dans des cellules en forme de tranches de disque (variante *Pie-Cursor*, le "curseur-tarte") (figure 3.18). L'utilisateur sélectionne une cellule, dont les cibles sont immédiatement regroupées en nouvelles cellules. L'utilisateur sélectionne ainsi les cellules jusqu'à ce que la cellule choisie ne contienne que la cible voulue. Conçues pour l'interaction tactile, les cellules correspondent à des zones du pavé tactile (figure 3.18). La tâche de sélection s'éloigne ainsi d'un pointage pour se rapprocher d'une succession de sélection de zones du pavé tactile. Cependant, les études expérimentales [67] montrent que la durée moyenne des sélections est significativement plus courte avec Bubble Cursor qu'avec les techniques de Cell-Cursor. Ce résultat est à rapprocher de précédentes études expérimentales présentant *TouchGrid* (la "grille de toucher", 2005) [66]. Les auteurs ont conclu qu'avec TouchGrid, la durée moyenne d'une tâche de pointage croît proportionnellement au nombre de touches nécessaires. Cela laisse peu de chances en termes de vitesse de sélection à ce genre de techniques face aux techniques ne nécessitant qu'un seul toucher.

D'autres techniques de sélection en plusieurs temps, les *Enhanced Area Cursors* (les "curseurs étendus améliorés", 2014) [45] facilitent efficacement le pointage de petites cibles pour des utilisateurs ayant des déficiences motrices, comme une maladie de Parkinson ou une tétraplégie par exemple. Ces utilisateurs ont de grandes difficultés à accomplir des pointages précis, lorsque la tâche nécessite des ajustements fins de la position du curseur pour désigner une cible petite.

Parmi ces techniques, *Click-and-Cross* ("clique-et-traverse") propose à l'utilisateur de contrôler un Area Cursor de forme ronde afin de sélectionner une *zone d'intérêt* autour

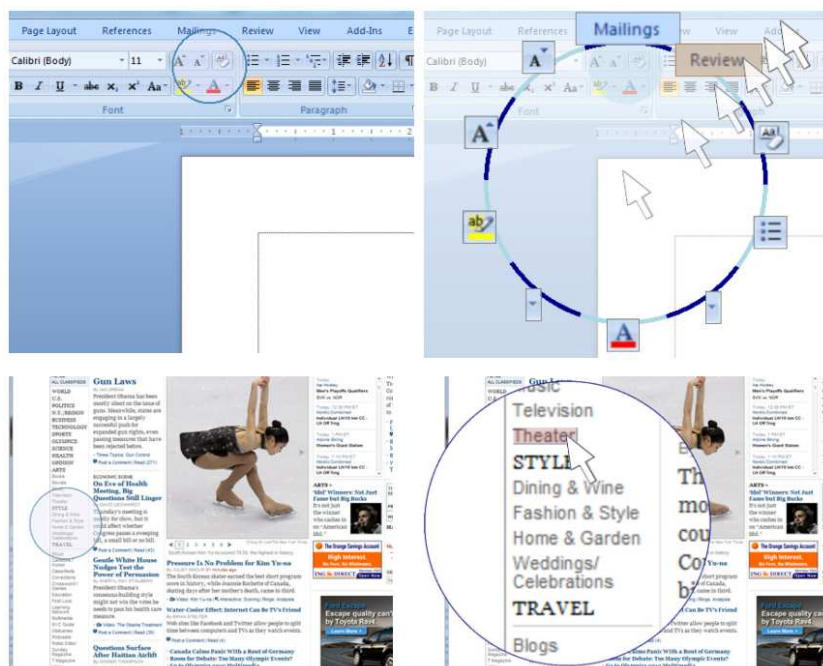


FIGURE 3.19 – En haut, Click-and-Cross et, en bas, Visual-Motor-Magnifier. Figure issue de [45].

de la cible. Les cibles contenues dans cette zone sont alors également réparties le long de sa périphérie : les cibles sont sur le bord de la zone d'intérêt (figure 3.19). L'utilisateur termine sa sélection en guidant le curseur vers la cible et en traversant l'arc de cercle de la zone (figure 3.19). Une autre de ces techniques, nommée *Visual-Motor-Magnifier* (la "loupe visuelle et motrice"), permet à l'utilisateur de définir de la même façon une zone d'intérêt ronde. Cette zone est ensuite uniformément grossie, comme avec une loupe, pour faciliter la sélection des petites cibles avec un second pointage, plus facile (figure 3.19). Les études expérimentales [45] impliquent des utilisateurs présentant des déficiences motrices. Les résultats montrent que ces deux techniques d'*Enhanced Area Cursor* apportent un gain de temps, d'efforts et de précision important. De plus, ces deux techniques sont unanimement préférées à Bubble Cursor [45].

3.4 Aide au pointage à distance

La plupart des techniques d'aide au pointage présentées dans ce chapitre peuvent être facilement adaptées au pointage à distance [15, 16, 104]. Nous présentons les études expérimentales qui s'intéressent à cette adaptation, puis nous présentons *Bubble Ray*, une technique spécialement développée pour le pointage à distance, mais néanmoins adaptable à d'autres types de pointage.

Parker et al. [104] comparent trois techniques d'expansion visuelle et motrice de cibles sur un dispositif proche du ray-casting (chapitre 2 section 2.3.1), le *TractorBeam* (le "rayon-tracteur"). Ce dispositif consiste en un écran disposé comme une table (*tabletop display*, "affichage dessus de table") et un stylet que l'on utilise sans contact avec l'écran pour contrôler le curseur à une courte distance (figure 3.20). Comme en ray-casting, l'axe principal du stylet est projeté sur la table pour définir la position du curseur. Le *TractorBeam* ne s'utilise pas à quelques mètres de l'écran, mais il présente de fortes similarités avec le ray-casting (principe de l'interaction, projection de l'axe du dispositif manipulable et 5 DDL). L'algorithme d'expansion est le même pour les trois techniques d'expansion de cibles comparées : à 90 % de D la distance à parcourir, le curseur entre dans l'espace moteur de la cible. L'utilisateur n'a plus qu'à valider la sélection. Cet

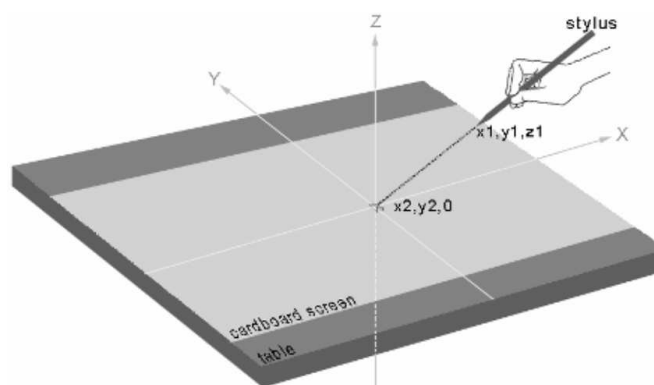


FIGURE 3.20 – Technique d'interaction TractorBeam. Figure issue de [104].

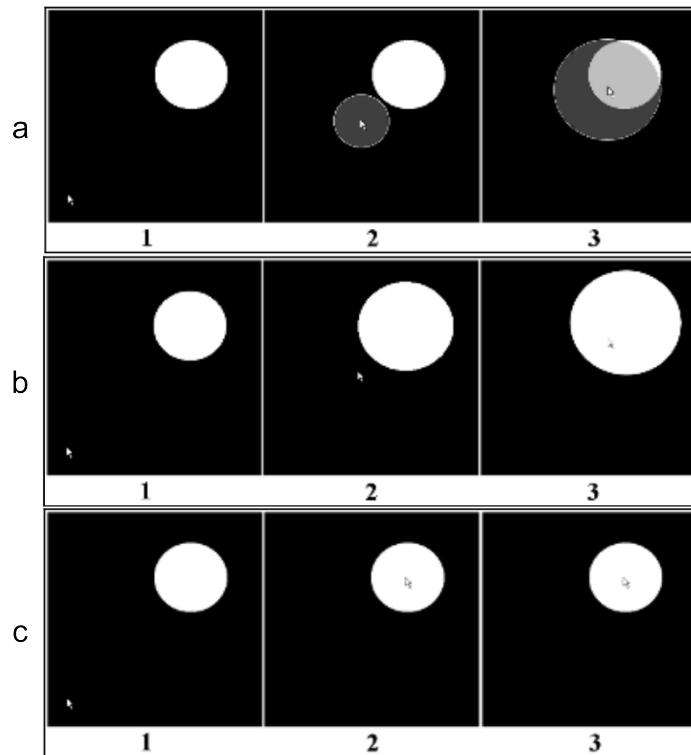


FIGURE 3.21 – Techniques d'expansion de cibles de l'étude TractorBeam [104] : (a) expand-cursor, (b) expand-target et (c) snap-to-target. En 1, le curseur n'a pas atteint le seuil des 90% de D , il n'est donc pas dans la taille étendue de la cible. En 2, il vient de passer le seuil : (a) le curseur commence à grossir, (b) la cible commence à grossir, (c) le curseur se téléporte sur la cible. En 3, le curseur est sur la cible initiale : (a) le curseur continue de grossir, (b) la cible continue de grossir, (c) le curseur reste sur la cible. Figure modifiée de [104].

l'algorithme définit donc la taille étendue des cibles de la façon suivante : la taille initiale est entourée d'une zone d'une largeur égale à 10 % de D , qui est également incluse dans la zone d'activation de la cible. Ajoutée à la taille initiale de la cible, cette zone qui l'entoure permet de la désigner avec le curseur. L'aide visuelle fournie diffère entre les trois techniques. Elle n'apparaît que quand le curseur entre dans la taille étendue de la cible. Les trois aides visuelles comparées avec l'algorithme d'expansion décrit ci-dessus sont les suivantes :

- *expand-cursor* (figure 3.21 a) affiche une bulle autour du curseur. Celle-ci grossit à mesure que le curseur approche du centre de la cible.
- *expand-target* (figure 3.21 b) agrandit visuellement la cible quand le curseur entre dans sa taille étendue, puis l'agrandit de plus en plus à mesure que le curseur approche du centre de la cible.
- *snap-to-target* (figure 3.21 c) téléporte le curseur au centre de la cible dès qu'il entre dans sa taille étendue.

Les résultats de l'étude montrent que l'aide visuelle *snap-to-target* obtient les meilleures performances et est préférée des utilisateurs.



FIGURE 3.22 – Jeu vidéo de tir avec une Wiimote utilisé par Bateman et al. [16] pour comparer 3 techniques d'aide au pointage. Les cibles sont mouvantes. Figure issue de [16].

Les travaux [15, 16] que nous présentons ensuite s'intéressent spécifiquement à faciliter le pointage à distance en ray-casting. Bateman et al. [16] comparent trois techniques permettant d'aider le joueur le moins bon dans un jeu vidéo de tir. La finalité est de contrebalancer les différences de niveaux trop importantes qui peuvent rendre le jeu lassant pour les joueurs les moins bons. Le dispositif de ray-casting est une Wiimote (figure 2.12). L'étude compare une technique d'expansion de cibles (un curseur étendu carré similaire à celui décrit à la figure 3.11), une technique de cibles collantes et une technique de cibles attractives. Les résultats montrent que le curseur étendu et les cibles attractives aident efficacement le joueur le moins bon. Le curseur étendu offre également les meilleurs gains de performance et il est perçu comme la technique d'aide au pointage la plus facile à utiliser, la plus efficace et la plus amusante.

Bateman et al. [15] étudient aussi les techniques de cibles gravitationnelles et de cibles collantes, avec pour principale motivation de pallier la perte de précision induite par les tremblements humains lors de l'utilisation d'une technique de ray-casting. Comme dans [16], les études expérimentales se font avec une Wiimote. L'objectif est de déterminer la force de gravité et l'adhérence des cibles permettant d'obtenir les meilleures performances de pointage. Ils comparent également différents indices de désignation de la cible selon plusieurs modalités de sortie : vibration de la Wiimote, production d'un son et changement de couleur de la cible. Les auteurs comparent plusieurs combinaisons multimodales de ces indices. Les résultats montrent que quelque soit l'indice ou la combinaison testée, cela n'a aucun effet significatif sur les performances du pointage. Enfin, les meilleures techniques de leurs trois premières expériences sont comparées dans une quatrième expérience. Les résultats montrent que les cibles très attractives et les cibles très adhérentes offrent les meilleures performances et ont la préférence des utilisateurs.

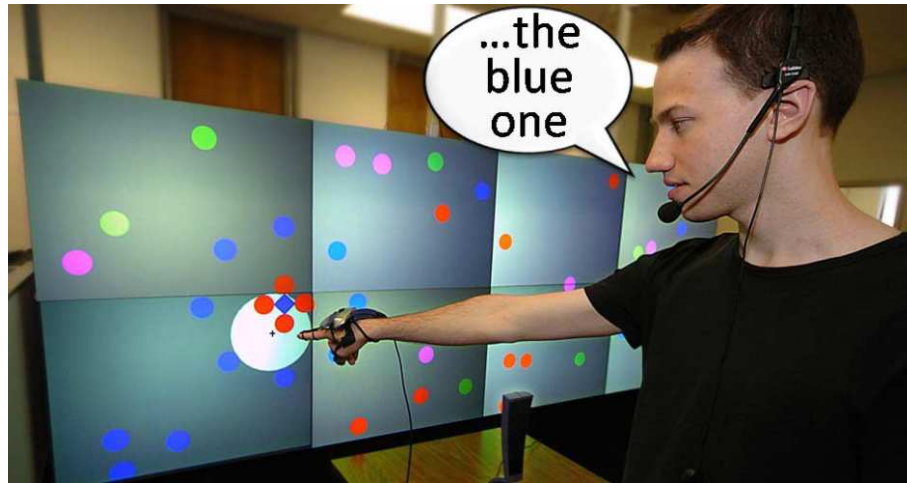


FIGURE 3.23 – Technique Speech-Filtered Bubble Ray : une technique multimodale.
Figure issue de [129].

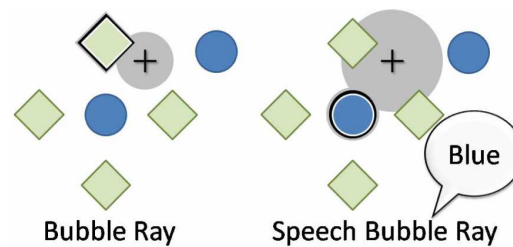


FIGURE 3.24 – À gauche, utilisée sans commande vocale, Bubble Ray est un Bubble Cursor pour le ray-casting. À droite, grâce à la commande vocale, les cibles vertes sont ignorées et l'expansion des cibles est plus importante. Figure issue de [129].

À notre connaissance, une seule technique logicielle d'aide au pointage a été spécifiquement conçue pour le pointage à distance. *Speech-Filtered Bubble Ray* (le "rayon-bulle filtré par la parole") [129] est une technique multimodale qui combine Bubble Cursor et une technique de reconnaissance vocale (figure 3.23) selon le principe suivant : lorsqu'il n'y a pas d'espace libre entre les cibles (cas des agglomérats de cibles), l'utilisateur verbalise une caractéristique de la cible, par exemple "la cible violette" ou "le fichier audio", et toutes les cibles ne correspondant pas à cette description sont ignorées par l'algorithme d'expansion. L'espace qui est occupé par les cibles ignorées est de nouveau disponible pour l'expansion et Bubble Cursor redevient efficace (figure 3.23). Utilisée sans la commande vocale, cette technique s'appelle *Bubble Ray* (figure 3.24).

L'étude expérimentale de cette technique [129] utilise le ray-casting et compare Speech-Filtered Bubble Ray, Bubble Ray et le ray-casting sans aide au pointage. Les résultats montrent que Speech-Filtered Bubble Ray est la technique la plus performante et la plus appréciée, suivie par Bubble Ray.

3.5 Quelle approche pour le contexte chirurgical ?

Les logiciels d'OrthoPilot® sont la cible applicative de nos travaux, et particulièrement l'écran de planification de TKA 5 (figure 3.25), déterminant pour une pose de prothèse totale de genou réussie chapitre (1 section 1.3). Nous choisissons donc parmi les approches existantes de l'aide au pointage, celle qui répond au mieux aux exigences de l'interaction chirurgien-ordinateur, compte-tenu des possibilités offertes par les logiciels OrthoPilot®. Ce choix s'appuie sur plusieurs constats.

Tout d'abord, les techniques de manipulation de la trajectoire du curseur posent plusieurs problèmes pour notre contexte applicatif :

1. Les chirurgiens préfèrent en général garder le contrôle de la situation et de leurs outils, comme nous l'expliquons au chapitre 2. Des techniques déviant ou ralentissant la trajectoire du curseur exercent sur celui-ci un contrôle qui échappe au chirurgien. Elles risquent donc d'être moins bien acceptées.
2. Les chirurgiens, conscients de la complexité et de la criticité de leur tâche, demandent des solutions parfaites, qui n'apportent que des avantages. Or, si au cours d'un pointage, le curseur est dévié ou ralenti à proximité de toutes les cibles survolées, la solution proposée est imparfaite. Elle peut même ralentir le geste si la densité de cibles est importante.
3. En terme de charge cognitive [128], les écarts entre geste planifié et geste effectif posent un réel problème vis-à-vis de la charge cognitive déjà importante du chirurgien, car ils créent une dissonance cognitive. Tandis que le chirurgien devient peu à peu expert dans la manipulation du système de pointage à distance, mobilisant déjà des ressources cognitives non-négligeables, de tels décalages risquent d'entraver son apprentissage et de créer un stress cognitif excessif. Cela peut avoir des conséquences dramatiques lorsque la tâche principale est aussi critique, car la mémoire de travail est théoriquement limitée et un stress cognitif excessif peut créer des problèmes de concentration et de mémoire. Nous pensons qu'une expansion des cibles mobilise moins de ressources cognitives qu'une manipulation de la trajectoire du curseur, car le lien entre les déplacements du curseur et les gestes du chirurgien reste inaltéré tout au long du pointage.

Deuxièmement, l'écran de planification (figure 3.25) dispose d'espace libre entre les cibles, comme de nombreux écrans des logiciels OrthoPilot® : les 18 cibles ne couvrent pas la totalité de l'écran comme une tessellation (un pavage). Une expansion motrice des cibles est donc possible sans agrandir l'espace moteur. Cet espace libre, inutile pour l'interaction, n'est cependant pas toujours présent dans nos logiciels du quotidien, comme les traitements de texte. Il existe principalement deux façons d'utiliser cet espace pour optimiser le compromis vitesse-précision : proposer des gestes plus grossiers ou proposer des gestes plus courts. Par exemple, Object Pointing ignore cet espace libre et réduit ainsi l'espace moteur. Le geste à effectuer est plus court mais la précision requise est identique : les tailles des cibles n'ont pas changé.

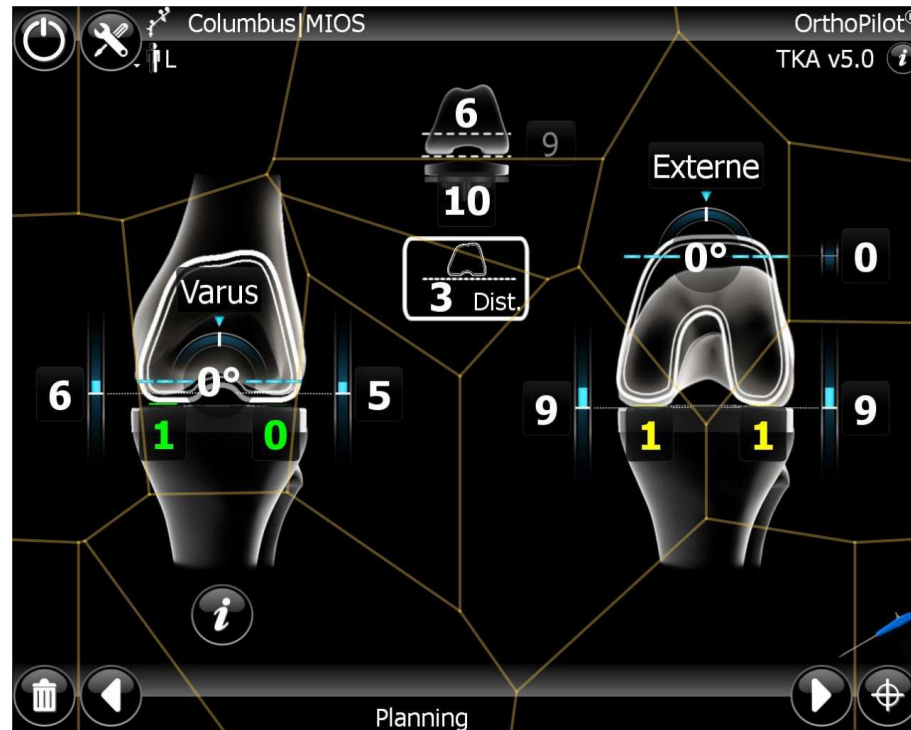


FIGURE 3.25 – La taille étendue des cibles est représentée par les lignes marron semi-transparentes. Figure issue de [58].

Une technique d'expansion de cibles attribue tout ou partie de cette espace libre aux cibles. La précision requise est moins importante et de plus, le geste à effectuer peut également être plus court. Nous favorisons les techniques d'expansion de cibles car les techniques de pointage à distance, comme le ray-casting, posent déjà des problèmes de précision (chapitre 2 section 2.3.1.4). Appliquer par exemple une tessellation de Voronoï à l'écran de planification de TKA 5 permet d'utiliser l'espace disponible sans problèmes d'ambiguïté (figure 3.25).

La position des cibles, nécessaire pour utiliser une technique d'aide au pointage *target-aware* (comme le sont les techniques d'expansion visuelle et motrice), peut être connue au sein d'un même logiciel. Il suffit au concepteur d'implémenter une fonctionnalité qui donne à chaque instant les caractéristiques spatiales des cibles actives. Il est préférable d'inclure cette fonctionnalité dès le pré-développement et la spécification du logiciel, mais cela n'est pas obligatoire. Comme nous le verrons au chapitre 9, elle peut être développée de façon ad-hoc. Développer un standard permettant aux logiciels de déclarer les cibles actives à chaque instant rendrait possible l'utilisation des techniques *target-aware* au niveau des systèmes d'exploitation multi-tâches.

Nous focalisons donc nos travaux sur les techniques d'aide au pointage les plus adéquates à l'interaction chirurgien-ordinateur et à l'utilisation de l'écran de planification avec le Virtual Pointer : les techniques d'expansion visuelle et motrice de cibles. Dans la suite de ce mémoire, nous les désignons sous les termes de *techniques d'expansion de cibles* ou seulement *techniques d'expansion*.

Conclusion

Dans ce chapitre, nous avons présenté une revue structurée des techniques d'aide au pointage, en détaillant particulièrement les techniques d'expansion motrice et visuelle de cibles, sujet de nos travaux. Parmi celles-ci, Bubble Cursor est devenue une référence pour l'évaluation de nouvelles techniques. Nous retiendrons également TARGET, Cell Painting, Ghost-hunting, Starburst, Dynaspot, IFC (Implicit Fan Cursor) et Bubble Lens, techniques d'expansion auxquelles nous faisons référence dans la suite de ce mémoire. Nous avons aussi présenté les travaux pour l'aide au pointage à distance, pour finir en expliquant notre choix d'étudier les techniques d'expansion de cibles pour le pointage à distance en contexte chirurgical. Les deux parties suivantes de ce mémoire présentent nos contributions concernant les techniques d'expansion de cibles tandis que la dernière partie, les applications de nos contributions au cas de l'interaction à distance chirurgien-ordinateur.

Deuxième partie

Conception d'aides visuelles pour l'expansion de cibles

Chapitre 4

Espace de conception

Sommaire

Introduction	69
4.1 Espace conceptuel	70
4.1.1 Trois axes pour les aides visuelles atomiques	70
4.1.1.1 Dynamicité	71
4.1.1.2 Observabilité de l'expansion	71
Aides visuelles explicites	72
Aides visuelles implicites	72
4.1.1.3 Élément augmenté	72
4.1.2 Combinaison d'aides visuelles atomiques	72
4.1.3 Notation matricielle d'une technique d'expansion de cibles	73
4.2 Exploration analytique de l'espace conceptuel	74
4.2.1 Aides visuelles <i>sur-curseur</i>	75
4.2.2 Aides visuelles <i>sur-espace</i>	75
4.2.3 Aides visuelles <i>sur-cible</i>	75
4.2.4 Aides visuelles combinées	76
Conclusion	76

Introduction

Dans une technique d'expansion motrice et visuelle de cibles, un *algorithme d'expansion* attribue plus d'espace aux cibles et une *aide visuelle* rend compte de cette expansion à l'utilisateur. Au cours des chapitres précédents, nous avons montré la grande diversité des aides visuelles, parfois pour le même algorithme d'expansion. Par exemple, Ghost-hunting [77] (figure 3.9), Bubble Cursor [53] (figure 3.12) et Dynaspot [34] (figure 3.14) utilisent la tessellation de Voronoï pour répartir l'espace libre entre les cibles. Leurs aides visuelles sont pourtant très différentes.

Des études [126, 131] ont mis en évidence la criticité de l'aide visuelle et son impact sur les performances de la technique d'expansion. Par exemple, l'étude de la technique Implicit Fan Cursor [126] montre des performances très différentes pour les trois aides visuelles étudiées, l'aide visuelle la plus minimaliste étant la plus performante.

Dans une première partie, nous présentons une de nos contributions [59] : un espace de conception permettant de décrire les aides visuelles selon trois axes de conception. À ces trois axes s'ajoute l'axe du temps avec la prise en compte des phases du mouvement. Ainsi, cet espace conceptuel permet de considérer les techniques d'expansion de cibles qui combinent plusieurs aides visuelles au cours d'un même mouvement de pointage. Nous proposons ensuite une notation matricielle qui permet de décrire synthétiquement les aides visuelles d'une technique d'expansion. Dans une deuxième partie, nous utilisons cet espace conceptuel pour classer l'ensemble des techniques d'expansion visuelle et motrice de cibles décrites dans l'état de l'art (chapitre précédent, section 3.2.4).

4.1 Espace conceptuel

4.1.1 Trois axes pour les aides visuelles atomiques

Nous introduisons la notion d'*aide visuelle atomique* pour décrire une aide visuelle élémentaire. Une aide visuelle atomique peut être combinée à d'autres aides visuelles atomiques au sein d'une même technique d'expansion de cibles. Pour décrire une aide visuelle atomique, nous définissons trois axes : la *dynamacité*, l'*observabilité de l'expansion* et l'*élément augmenté* (figure 4.1). Pour présenter ces axes de conception, nous prenons comme exemples les deux techniques Bubble Cursor et Starburst (figure 4.2). Tandis que la bulle de Bubble Cursor indique à tout moment quelle cible est désignée, Starburst affiche en permanence l'ensemble des tailles étendues des cibles.

Espace de conception : aides visuelles atomiques

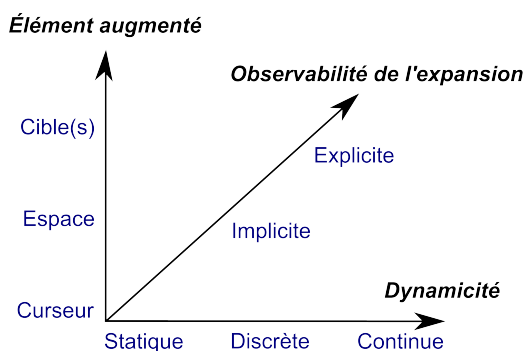


FIGURE 4.1 – Espace de conception pour les aides visuelles atomiques : trois axes de conception.



FIGURE 4.2 – À gauche, Bubble Cursor. À droite, Starburst. Les cibles sont les points d'intérêt sur la carte. Leur taille étendue est représentée par les lignes grises épaisses. Figure modifiée de [53] et [19].

4.1.1.1 Dynamicité

Des études précédentes [17, 132] définissent une aide visuelle comme *statique* ou *dynamique*. Cependant, cette distinction est insuffisante. En effet, certaines aides visuelles dynamiques sont *discrètes* tandis que d'autres sont *continues*. Comme expliqué dans [17, 132], nous distinguons les aides visuelles statiques de celles dynamiques. Selon cet axe, nous affinons l'aspect dynamique en distinguant les aides visuelles discrètes et celles continues. Cette distinction fait appel aux notions de discret et de continu empruntées aux Mathématiques : tandis que certaines aides visuelles changent d'apparence à des moments discrets dans le temps, par exemple lorsque le curseur change de cible désignée, d'autres aides visuelles, comme celle de Bubble Cursor, changent constamment d'apparence lorsque le curseur bouge. L'aide visuelle *discrète* est une aide visuelle qui change d'état par à-coups tandis que l'aide visuelle *continue* change tout au long du mouvement du curseur. Enfin l'aide visuelle *statique* ne change pas lors du geste de pointage : elle est statique, comme l'aide visuelle de Starburst. L'axe *Dynamicité* de notre espace de conception comprend donc trois valeurs : statique, discrète et continue.

4.1.1.2 Observabilité de l'expansion

L'axe *Observabilité de l'expansion* caractérise l'observabilité visuelle de l'expansion motrice de cibles. Les aides visuelles rendent observables la cible désignée, une ou plusieurs cibles étendues (par exemple celles autour du curseur) ou la répartition totale de l'espace moteur entre les cibles, comme Starburst (figure 4.2 à droite). Les aides visuelles existantes fournissent de l'information spatiale. Aucune approche symbolique (avec des lettres par exemple) n'a été étudiée pour l'aide visuelle d'une expansion de cibles, probablement car la tâche de pointage est elle-même spatiale. Nous décrivons les aides visuelles atomiques en distinguant l'observabilité *explicite* de l'expansion et l'observabilité *implicite* de l'expansion.

Aides visuelles explicites Lorsque la taille étendue d'une cible ou de plusieurs cibles sont directement observables par l'utilisateur (par des indices visuels comme des lignes ou des couleurs), l'aide visuelle est *explicite* : c'est une information spatiale qui indique à l'utilisateur *où* cliquer. En montrant une seule capture d'écran de la technique, il est possible d'observer directement la taille étendue d'une cible au moins, comme avec Starburst (figure 4.2 à droite).

Aides visuelles implicites Si l'utilisateur doit estimer la taille étendue d'une cible sans l'observer directement, alors l'aide visuelle fournie est *implicite*. Une aide visuelle implicite ne fournit d'information spatiale sur l'expansion que relativement à son état précédent. L'aide visuelle est alors dynamique (discrète ou continue) et elle indique à l'utilisateur *quand* cliquer lorsque le curseur se déplace. Par exemple, Bubble Cursor est implicite car aucune taille étendue de cible n'est affichée : quand la bulle atteint la cible, l'utilisateur peut sélectionner la cible en cliquant (figure 4.2 à gauche).

4.1.1.3 Élément augmenté

Plusieurs indices visuels (couleur, forme, taille...) définissent la forme concrète à l'écran d'une aide visuelle. De plus pour les expansions de cibles, une caractéristique discriminante de la forme de l'aide visuelle est l'*élément* de la tâche de pointage qui est visuellement *augmenté* : le curseur, la cible ou l'espace entre les deux.

Comme le curseur est un élément mobile, les aides visuelles *sur-curseur* comme Bubble Cursor (figure 4.2 à gauche) sont dynamiques par définition. Une aide visuelle *sur-espace* modifie l'espace entre les cibles en partie ou totalement, tandis qu'une aide visuelle *sur-cible* modifie l'apparence visuelle d'au moins une cible. Par exemple avec Starburst (figure 4.2 à droite), l'apparence visuelle des cibles n'est pas modifiée : c'est une aide visuelle *sur-espace*, par opposition par exemple à une aide visuelle *sur-cible* qui double la taille de la cible désignée, comme TARGET (chapitre 3 section 3.2.4).

L'espace de conception à trois axes ainsi obtenu (figure 4.1) définit 16 types d'aide visuelle atomique (figure 4.3), à utiliser comme base pour la description des techniques existantes et pour la conception de nouvelles techniques d'expansion. La figure 4.3 donne une représentation en deux dimensions de l'espace de conception, ce qui permet de visualiser les 16 types d'aide visuelle atomique et de procéder plus facilement au classement des techniques d'expansion de cibles de la littérature présentées au chapitre précédent.

4.1.2 Combinaison d'aides visuelles atomiques

Certaines techniques d'expansion de cibles combinent différentes aides visuelles atomiques. Pour décrire ces combinaisons, nous considérons l'enchaînement des aides visuelles au cours des trois phases du geste de pointage définies par le MIIO [88] (chapitre 2 section 2.1.5) :

Élément augmenté	Observabilité de l'expansion					
	Dynamicité					
	statique	implicite discret	continu	statique	explicite discret	continu
curseur	Impossible		Area Cursor, Bubble Cursor, Lazy Bubble, Cone Cursor, Bubble Lens, Dynaspot	Impossible		
espace			Ghost-hunting	Starburst, Bubble Lens		
cible(s)		TARGET, IFC, Dynaspot		Bubble Lens	Cell Painting	

FIGURE 4.3 – Représentation en deux dimensions de l'espace de conception des aides visuelles atomiques et classification des techniques existantes. Les techniques combinant plusieurs aides visuelles atomiques apparaissent dans plusieurs cases avec des liens fléchés. Les cases vides représentent des possibilités inexplorées d'aide visuelle.

1. la *phase préliminaire*, qui précède le mouvement de pointage,
2. la *phase balistique*, qui désigne le premier mouvement d'atteinte, le sous-mouvement balistique
3. et la *phase corrective*, qui n'a lieu que si le sous-mouvement balistique n'atteint pas la cible.

Si une technique utilise plusieurs aides visuelles atomiques simultanément, c'est-à-dire durant la même phase du mouvement, ces aides visuelles sont combinées de façon *parallèle*. Les combinaisons *séquentielles* décrivent alors l'enchaînement d'aides visuelles utilisées durant différentes phases consécutives du mouvement. Ainsi, les aides visuelles sont définies à la granularité de la phase du mouvement de pointage.

4.1.3 Notation matricielle d'une technique d'expansion de cibles

Une aide visuelle atomique correspond à une et une seule case dans la représentation en deux dimensions de l'espace de conception (figure 4.3). Une technique qui combine différentes aides visuelles correspond donc potentiellement à différentes cases dans cette représentation (figure 4.3). La comparaison de telles techniques à l'aide de la représentation en deux dimensions de l'espace de conception n'est donc pas simple. En réponse à ce problème, nous définissons une notation visuelle compacte pour décrire une technique d'expansion de cibles en se basant sur les caractéristiques des différentes aides visuelles atomiques qu'elle utilise et sur leur schéma de combinaison (figure 4.4).

Une matrice de 3 lignes et 3 colonnes représente une technique d'expansion de cibles. Les colonnes représentent les 3 phases du mouvements en suivant un axe temporel horizontal, en ordre chronologique de la gauche vers la droite (figure 4.4). Les lignes correspondent à l'axe *Élément augmenté* de l'espace de conception : du *curseur* (lignes du dessus) à la *cible* (ligne du dessous) en passant par l'*espace* entre les deux (ligne du milieu)

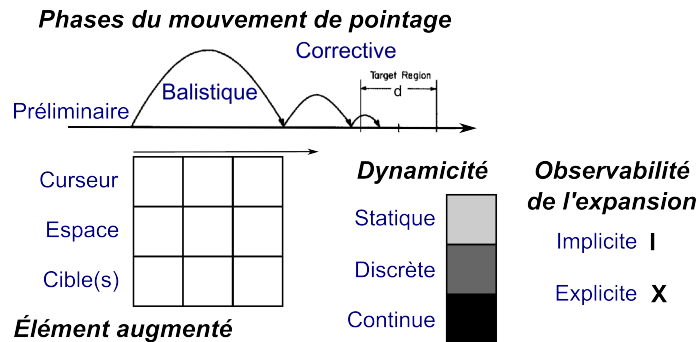


FIGURE 4.4 – Principe de la notation en matrice 3×3 . Les phases du mouvement sont représentées par les colonnes. L'axe *Dynamacité* est représenté par des niveaux de gris. L'axe *Observabilité de l'expansion* est représenté par une lettre : X ou I. L'axe *Élément augmenté* est représenté par les lignes.

(figure 4.4). Chaque cellule de la matrice représente ainsi un type d'élément augmenté durant une phase du mouvement de pointage. Un code de niveaux de gris représente l'axe *Dynamacité* : gris clair pour *statique*, gris foncé pour dynamique-*discret* et noir pour dynamique-*continu*. Enfin, pour l'axe *Observabilité de l'expansion* on utilise un X dans une cellule pour une aide visuelle *explicite* comme Starburst, et un I pour une aide visuelle *implicite* comme Bubble Cursor. Ceci permet de représenter les combinaisons d'aides visuelles plus facilement et plus efficacement.

4.2 Exploration analytique de l'espace conceptuel

La figure 4.3 représente notre classification selon leurs aides visuelles des techniques d'expansion de cibles de la littérature, que nous avons décrites au chapitre précédent, section 3.2.4. La figure 4.5 représente la notation matricielle de ces techniques. Nous abordons cette classification par l'axe *Élément augmenté*, ce qui nous conduit naturellement à la combinaison d'aides visuelles augmentant différents éléments.

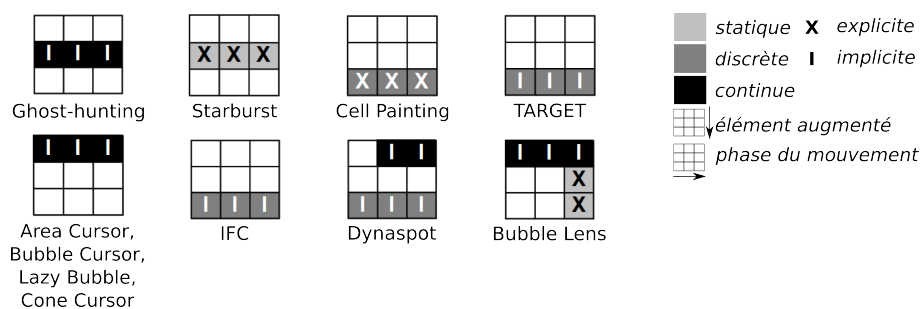


FIGURE 4.5 – Notation matricielle des techniques d'expansion de la littérature.

4.2.1 Aides visuelles *sur-curseur*

L'aide visuelle de Bubble Cursor définit une augmentation du *curseur* qui est *implicite* et *continue* (figure 4.3). Area Cursor, Lazy Bubble et Cone Cursor sont des variations le long du continuum qui va du traditionnel point de sélection au Bubble Cursor. Ces techniques utilisent également une aide visuelle *implicite* et *continue* en augmentant visuellement le *curseur* (figure 4.3) : au cours d'un mouvement de pointage, les utilisateurs savent *quand* cliquer pour sélectionner la cible. Leur notation matricielle est donc identique à celle de Bubble Cursor (figure 4.5).

4.2.2 Aides visuelles *sur-espace*

Starburst affiche la répartition entière de l'espace entre les cibles une fois, avant que le mouvement commence. Starburst illustre un type d'aide visuelle qui rend l'expansion des cibles explicitement et immédiatement observable. L'aide visuelle est donc une augmentation de l'*espace explicite* et *statique* (figure 4.3 et figure 4.5). Avec l'affichage d'un avatar pour chaque cible, Ghost-hunting utilise une augmentation de l'*espace implicite* et *continue* (figure 4.3 et figure 4.5) : les avatars sont dans l'espace entre les cibles, ils bougent lorsque le curseur bouge mais aucune cellule de Voronoï n'est affichée.

4.2.3 Aides visuelles *sur-cible*

Les aides visuelles mettant en valeur une cible ont été beaucoup étudiées [85, 142]. Parmi elles, *Cell Painting* remplit entièrement la cellule de Voronoï de la cible désignée avec une couleur semi-transparente. L'aide visuelle utilisée est donc une augmentation *explicite* et *discrète* de la *cible* (figure 4.3 et figure 4.5). En revanche, quand la cible est seulement agrandie, comme avec TARGET, ou mise en valeur en changeant de couleur [85, 142], l'aide visuelle est une augmentation de *cible implicite* et *discrète* (figure 4.3 et figure 4.5). En effet, contrairement à Cell Painting, ces techniques rendent visible la cible désignée mais pas sa taille étendue.

Implicite Fan Cursor (IFC) [126] était à l'origine une technique conçue comme une augmentation *implicite* et *continue* du *curseur* : Fan Cursor. Son étude expérimentale [126] a comparé Fan Cursor avec une version alternative qui utilise seulement la mise en valeur de la cible désignée. Les résultats ont montré que cette augmentation *discrète* et *implicite* de la *cible* (figure 4.3 et figure 4.5) est plus efficace que la technique initiale Fan Cursor.

4.2.4 Aides visuelles combinées

Dynaspot [34] combine deux aides visuelles complémentaires. Premièrement, la bulle, une aide visuelle *sur-curseur implicite* et *continue*, n'apparaît que quand le curseur bouge et indique le mode actif : *normal* ou *expansion*. Deuxièmement, la mise en valeur de la cible désignée est une aide visuelle *sur-cible implicite* et *discrète*. La technique Dynaspot utilise donc une combinaison *parallèle* de deux aides visuelles (figure 4.3 et figure 4.5).

Bubble Lens [90] se comporte comme un Bubble Cursor [53], avec en plus l'ajout d'un grossissement local quand les cibles sont petites et denses. Grâce au déclenchement cinématique, le grossissement ne se produit que durant la phase corrective du mouvement de pointage. Durant la phase de grossissement, les cibles et l'espace sont étendus dans une zone d'intérêt ronde par un algorithme d'expansion particulier, différent de la tessellation de Voronoï. Comme les cibles et l'espace sont représentés dans leur forme étendue, indépendamment des mouvements du curseur, le grossissement combine deux aides visuelles *explicites* et *statiques* : une expansion visuelle de l'*espace* et une expansion visuelle des *cibles*. Pour résumer, comme Bubble Cursor est encore actif durant le grossissement, deux algorithmes d'expansions et trois aides visuelles sont donc utilisés simultanément (figure 4.3 et figure 4.5).

Conclusion

Dans ce chapitre, nous avons présenté une première contribution : un espace de conception selon trois axes des aides visuelles atomiques des techniques d'expansion de cibles. Cet espace se complète d'une prise en compte des phases du mouvement de pointage et des combinaisons possibles d'aides visuelles par une technique d'expansion. Afin de rendre plus lisible une technique entière, en particulier si elle fournit plusieurs aides visuelles, nous avons proposé une notation compacte sous la forme de matrices de 3 lignes \times 3 colonnes. Nous avons finalement montré le pouvoir descriptif de ces éléments de conception en classant toutes les techniques d'expansion visuelle et motrice de cibles existantes présentées au chapitre précédent, section 3.2.4. Dans le chapitre suivant, nous montrons comment ces éléments de conception nous ont permis de créer de nouvelles techniques d'expansion de cibles.

Chapitre 5

Nouvelles techniques d'expansion

Sommaire

Introduction	78
5.1 Nouvelles techniques <i>sur-espace</i>	79
5.1.1 VTE	79
5.1.1.1 Rationnel de conception	79
5.1.1.2 Exploration de variantes de conception	81
Appareillage et participants	83
Protocole expérimental	83
Résultats	84
Discussion	85
5.1.2 MTE	85
5.1.3 eVTE	86
5.1.4 Conclusion des nouvelles techniques <i>sur-espace</i>	87
5.2 Nouvelles techniques <i>sur-curseur</i>	87
5.2.1 Expansion Lens	87
5.2.2 Rope Cursor	89
5.2.3 Conclusion des nouvelles techniques <i>sur-curseur</i>	91
5.3 Combinaisons d'aides visuelles : trois nouvelles techniques	92
5.4 Extensions	94
5.4.1 Améliorer l'aspect graphique	94
5.4.2 Changement de mode implicite	94
5.4.2.1 Fonctionnement du changement de mode implicite . .	94
5.4.2.2 Adaptation de Rope Cursor	95
5.4.2.3 Adaptation d'Expansion Lens	96
5.4.3 Autres algorithmes d'expansion de cibles	96
5.4.3.1 Voronoï en distances de Manhattan	96
5.4.3.2 Autres variantes de Voronoï	96
5.4.3.3 Tripler la taille des cibles	97
5.4.4 VTE to Expansion Lens	98
Conclusion	98

Introduction

Une technique d'expansion de cibles utilise un algorithme d'expansion, qui alloue de l'espace supplémentaire aux cibles, et une aide visuelle, qui informe l'utilisateur de cette expansion. L'analyse et la formalisation présentées au chapitre précédent permettent la description des techniques existantes et la création de nouvelles techniques d'expansion que nous présentons dans ce chapitre. Tandis que certaines de ces techniques utilisent une seule aide visuelle, d'autres les combinent pour adapter l'aide visuelle aux phases de la tâche de pointage. Dans ce chapitre, nous présentons trois techniques *sur-espace* – VTE [58], MTE [59] et eVTE [59] – deux techniques *sur-curseur* – Rope Cursor [59] et Expansion Lens [60] – et trois combinaisons d'aides visuelles [59] : VTetoROPE, VTetoCELL et VTetoTARGET. À part MTE, ces techniques ont été développées et évaluées avec la tessellation de Voronoï comme algorithme d'expansion. L'utilisation d'un autre algorithme d'expansion est néanmoins possible : la majorité des techniques peuvent être directement adaptées, comme le démontre la nouvelle technique MTE adaptée de VTE.

Tout d'abord, VTE est issu de la volonté d'utiliser une aide visuelle *explicite*, *statique* et *sur-espace*, comme Starburst. Sa conception est motivée par l'étude du Modèle de l'Impulsion Initiale Optimisée (MIIO) [88] et de la littérature. MTE est une variante de VTE utilisant un algorithme d'expansion différent. Nous avons ensuite exploré plusieurs autres variantes de VTE en nous reposant sur l'espace de conception. Ainsi l'exploration de l'axe *Dynamacité* a permis la conception d'eVTE, avec une dynamacité *continue*. L'exploration de l'axe *Élément augmenté* a ensuite permis de créer Expansion Lens, toujours dans le but d'améliorer VTE.

Dans la catégorie des aides visuelles *implicites*, *continues* et *sur-curseur*, caractéristique des techniques de curseur étendu (chapitre 3 section 3.2.4.2), nous proposons ensuite Rope Cursor, créé par exploration de l'axe *Observabilité de l'expansion* à partir de Bubble Cursor, une technique plusieurs fois répliquée [26, 34, 126, 131].

Enfin, le principe de combiner VTE avec une autre aide visuelle a donné naissance à trois techniques combinées : VTetoROPE, VTetoCELL et VTetoTARGET.

Élément augmenté	Observabilité de l'expansion					
	Dynamacité					
		<i>implicite</i>		<i>explicite</i>		
	<i>statique</i>	<i>discret</i>	<i>continu</i>	<i>statique</i>	<i>discret</i>	<i>continu</i>
<i>curseur</i>	Impossible		Rope Cursor	Impossible		Expansion Lens
<i>espace</i>				VTE, MTE		eVTE
<i>cible(s)</i>		TARGET			Cell Painting	

FIGURE 5.1 – Classification des techniques conçues au sein de notre espace de conception. Les flèches représentent les techniques combinées.

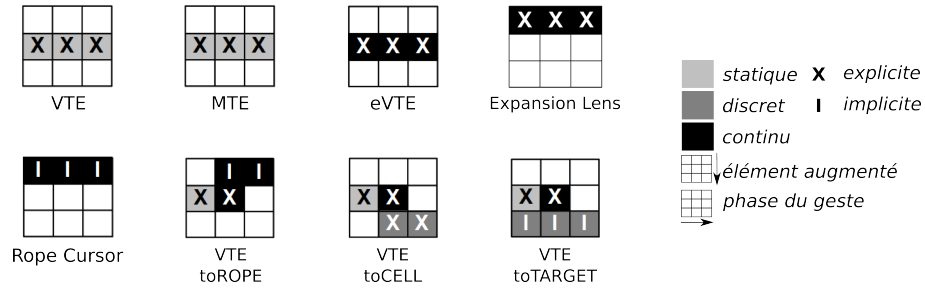
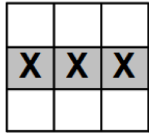


FIGURE 5.2 – Notation matricielle des techniques conçues.

La figure 5.1 représente la classification des techniques ainsi conçues dans l'espace de conception décrit au chapitre précédent. La figure 5.2 présente la notation matricielle, introduite aussi au chapitre précédent, de ces techniques. Nous illustrons les techniques conçues avec plusieurs images issues de nos différentes expériences. Celles-ci contiennent un curseur en forme de croix, une cible à atteindre contenant un 0 et d'autres cibles, les distracteurs (chapitre 2 section 2.1.2), contenant un *X* ou rien.

5.1 Nouvelles techniques *sur-espace*

5.1.1 VTE



Voronoi-based Target Expansion (VTE) [58] (l'"expansion de cibles basée sur Voronoï", figure 5.3) repose sur la tessellation de Voronoï, comme Bubble Cursor [53] et Ghost-hunting [77]. L'aide visuelle de VTE est *sur-espace*, *statique* et *explicite* (figures 5.1 et 5.2), comme Starburst [19]. Ainsi, VTE se démarque des techniques de la littérature qui exploitent la tessellation de Voronoï avec des aides visuelles *implicites*, comme Bubble Cursor et Ghost-hunting. La nouveauté de cette technique réside dans l'affichage constant du diagramme de Voronoï complet, et donc de l'ensemble des tailles étendues des cibles, dès le début de chaque geste de pointage. Contrairement à Starburst, qui a été testé sur des tablettes interactives, VTE est conçu comme une technique d'aide au pointage à distance (contexte applicatif décrit au chapitre 1). Contrairement à Starburst, il n'y a donc pas d'adaptation spéciale aux cibles petites et denses. En effet, les cibles petites et denses posent des problèmes de lisibilité et de précision lorsque l'utilisateur est à distance. L'algorithme d'expansion utilisé est donc la tessellation de Voronoï originelle.

5.1.1.1 Rationnel de conception

La conception de VTE repose sur un principe simple : afficher dès le début la taille étendue de la cible permet d'être plus performant. Ce principe prend pleinement son sens à la lumière du Modèle de l'Impulsion Initiale Optimisée (MIIO) [88] et conduit à trois fondements de cette technique :

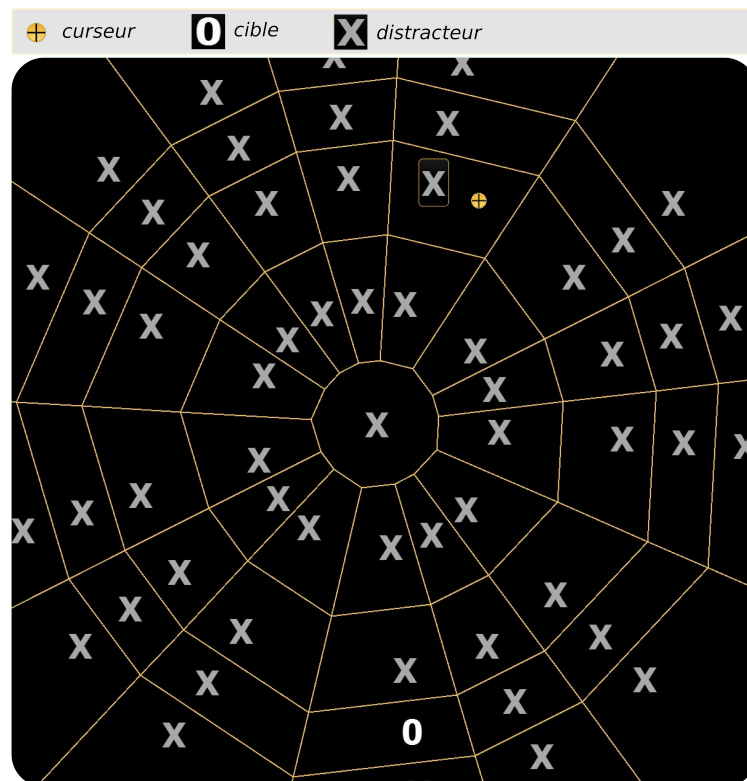


FIGURE 5.3 – VTE.

1. D'après le MIIO, connaître la taille étendue de la cible (sa taille "réelle" dans l'espace moteur) permet d'optimiser le mouvement de pointage dès le commencement de la tâche, c'est-à-dire dès la phase préliminaire et la programmation neuro-motrice du sous-mouvement balistique. En effet, ce modèle définit la variabilité du point de chute du sous-mouvement balistique, c'est-à-dire la dispersion des points de chute autour du point visé, comme proportionnelle à sa vitesse moyenne :

$$S = k \cdot \frac{D}{T} \quad (5.1)$$

S désigne l'écart-type du point de chute, D la distance parcourue, T la durée du sous-mouvement et k est une constante positive. Ainsi, plus le sous-mouvement balistique est rapide, moins il est précis et inversement. Les utilisateurs qui observent une cible plus grande pourront être moins précis autour du point de chute idéal, sans perdre pour autant en réussite de leur sous-mouvement balistique. Une plus grande dispersion du point de chute n'est pas un problème si la cible est plus grande. Les utilisateurs pourront donc programmer un sous-mouvement balistique moins précis et plus rapide, dès la phase préliminaire : un sous-mouvement balistique adapté à la taille étendue de la cible et non à sa taille originelle. D'après le MIIO, les sous-mouvements correctifs devraient bénéficier également d'une connaissance a priori de la taille étendue de la cible : ils pourront être programmés moins précis et plus rapides pour des raisons similaires.

2. Si le sous-mouvement balistique est plus efficace avec VTE qu'avec les techniques *implicites*, il pourrait également suffire dans de nombreux cas, la phase corrective devenant alors inutile. Ne pas avoir recours aux sous-mouvements correctifs apporte un bénéfice potentiel de 45 % de la durée totale du geste de pointage [142].
3. Même si la littérature a montré que les utilisateurs visent le barycentre de la cible [55] (ou de la cible étendue avec VTE), afficher explicitement les cellules permet de profiter des formes allongées des cellules de Voronoï pour trouver des points de sélection beaucoup plus proches et pour mieux assurer le clic au moment de la sélection. En effet, connaître les contours de l'espace disponible pour cliquer permet de savoir si le curseur est proche d'un bord de la cellule, et donc d'anticiper des erreurs de sélection.

Ainsi, les bénéfices potentiels de VTE sont multiples et sont particulièrement adaptés au pointage à distance, les dispositifs de pointage à distance souffrant de problèmes de précision et générant une fatigue non-négligeable de l'utilisateur (chapitre 2 section 2.3.1.4).

5.1.1.2 Exploration de variantes de conception

Pour concevoir VTE [58], nous avons étudié quelle forme est la meilleure pour afficher constamment le diagramme de Voronoï complet tout en restant visuellement simple et sans encombrer trop l'interface graphique originelle. Pour cela, nous avons étudié deux paramètres : la visualisation du diagramme de Voronoï et sa combinaison parallèle avec TARGET (chapitre 3 section 3.2.4), une aide visuelle supplémentaire, *implicite*, *discrète* et *sur-cible*, qui consiste à doubler le diamètre de la cible désignée (figure 5.4 *a* et *b*).

Nous avons considéré trois visualisations du diagramme de Voronoï, pour lesquelles nous avons choisi un thème de couleur (de bleu à rose, par le violet) afin de donner une coloration cohérente à la technique :

1. La première visualisation représente directement le diagramme complet avec des lignes semi-transparentes (figure 5.4 *a* et *b*). Elle est nommée EDGE ("arête" en anglais).
2. La deuxième visualisation est une carte de distance basée sur la position des cibles (figure 5.4 *c*). La distance entre chaque pixel violet et la cible la plus proche est codée par la transparence du pixel. La distance la plus grande séparant un pixel d'une cible est calculée dès le début pour définir le pixel le plus opaque. Ainsi les distances représentées utilisent toute la gamme de transparence disponible et les pixels proches des cibles sont transparents, ce qui permet de conserver une aide visuelle *sur-espace*. Cette visualisation présente le diagramme de Voronoï de façon plus fluide et continue que le diagramme brut de la première visualisation EDGE (figure 5.4 *a* et *b*) car elle ne contient aucune arête entre deux cellules, ni transition franche. Elle est nommée DMAP ("distance map" en anglais).

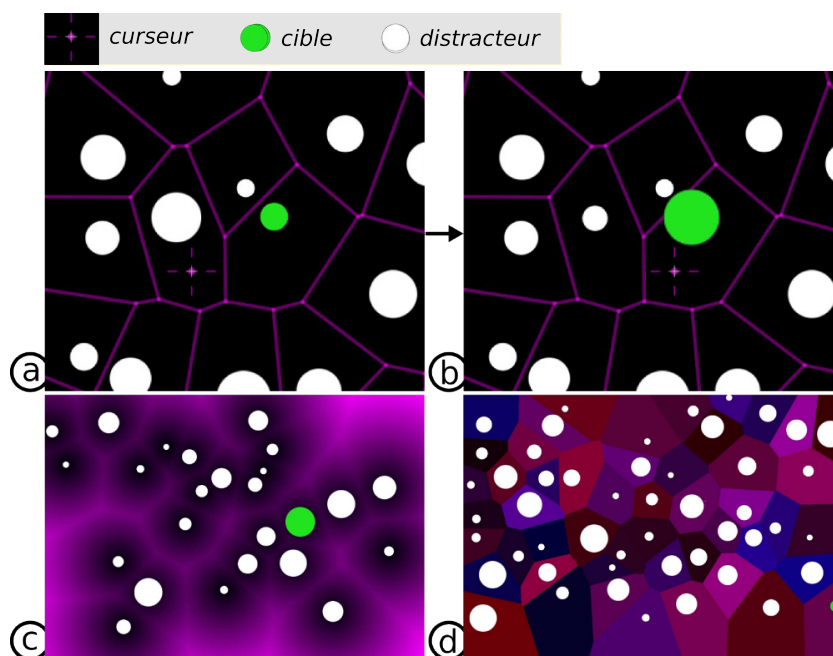


FIGURE 5.4 – Les trois variantes de VTE comparées et TARGET. Le fond d'écran est noir. Les images *a* et *b* illustrent TARGET (agrandissement de la cible désignée), combiné à la première visualisation : EDGE. Les images *c* et *d* illustrent respectivement DMAP et TILE.

3. La troisième visualisation est une représentation des cellules de Voronoï dans différentes couleurs semi-transparentes en arrière-plan des cibles afin de rester *sur-espace*. Cette représentation crée une sorte de pavage de l'écran (figure 5.4 *d*) grâce à un algorithme de génération de couleurs différentes et pseudo-aléatoires dans le thème de couleur choisi. Elle est nommée TILE ("carreau" en anglais).

Nous avons étudié si l'ajout d'un agrandissement de la cible désignée, comme TARGET (chapitre 3 section 3.2.4), pourrait améliorer les performances de VTE en apportant une aide visuelle dynamique et complémentaire à VTE (figure 5.4 *a* et *b*). En effet, TARGET agit au niveau de la cible (c'est une aide visuelle *sur-cible*), c'est-à-dire au centre des cellules de Voronoï, tandis que VTE définit les bords des cellules (une aide visuelle *sur-espace*). De plus, VTE ne propose aucun indice visuel à l'entrée du curseur dans une cellule. TARGET double la taille de la cible désignée dès que le curseur entre dans sa cellule de Voronoï. Elle apporte un indice clair de la cible désignée, qui change au cours du geste, et complète VTE : ainsi un utilisateur novice, à qui personne n'aurait expliqué le fonctionnement de VTE, pourrait comprendre rapidement son fonctionnement.

Nous avons donc combiné les trois variantes visuelles de VTE (figure 5.4) avec l'usage ou non de TARGET. Nous avons comparé expérimentalement les six techniques conçues avec TARGET seule (sans visualisation du diagramme) et avec une technique "contrôle" sans expansion de cibles ni aide au pointage notée CONTROL [58]. La comparaison entre TARGET seule et les variantes de VTE explore la possibilité que TARGET seule pourrait être plus efficace que toutes les variantes de VTE, ce qui contredirait le rationnel de

conception de VTE. Dans le cas contraire, ce serait un indice du bien fondé de ce rationnel de conception. L'expérience compare donc 8 techniques EDGE, EDGE+TARGET, DMAP, DMAP+TARGET, TILE, TILE+TARGET, TARGET et CONTROL.

Appareillage et participants Nous avons conduit cette étude pilote avec six participants bénévoles recrutés parmi les membres du laboratoire (5 femmes et 1 homme, dont 5 droitier(e)s et un gaucher(e), âgé(e)s de 25 à 31 ans). Les participants utilisent leur main dominante pour contrôler le curseur. Nous utilisons comme technique de pointage à distance le dispositif de pointage main libre utilisant le Asus Xtion Pro Live, décrit au chapitre 2 section 2.3.2. Le CD-gain du dispositif est de 1, ce qui signifie que le curseur se déplace à l'écran des mêmes distances que la main des participants dans le plan de l'écran, sans amplification ni amortissement des distances. Les participants se situent à 2 mètres de l'écran. Ils valident leurs sélections avec un appui court sur une pédale médicale fournie par Aesculap. Le programme développé pour l'expérience s'exécute sur un PC Quadcore sous Windows 7 cadencé à 2,3 GHz. L'affichage est un écran LCD de 20 pouces, à la résolution de 1280×1024 pixels.

Protocole expérimental La tâche consiste à sélectionner les cibles. Comme le montre la figure 5.4, les distracteurs sont des disques blancs et la cible est un disque vert. Les distracteurs ajoutent du réalisme à l'expérience par leur présence, mais l'utilisateur ne doit pas les sélectionner. Les distracteurs sont distribués aléatoirement dans l'espace, toutefois sans contact entre eux ou avec la cible. Leur taille est aléatoire entre 20 et 100 pixels. Si une sélection est faite en dehors de l'écran, un son indique que l'action est impossible. Lorsqu'une cible agrandie par TARGET "déborde" sur une cellule de Voronoï voisine, la priorité est donnée à la cible agrandie : si le curseur est dans la cible au moment de valider la sélection, la cible agrandie est sélectionnée. Les variables indépendantes sont :

- l'*Amplitude* du mouvement, c'est-à-dire la distance à parcourir jusqu'à la cible : 630 ou 840 pixels,
- la *Taille* de la cible : 40 (figure 5.4 d) ou 100 pixels (figure 5.4 c),
- le nombre de *Distracteurs* : 9, 24 (figure 5.4 c) ou 49 (figure 5.4 d),
- la *Technique* d'expansion de cibles : EDGE, EDGE+TARGET, DMAP, DMAP+TARGET, TILE, TILE+TARGET, TARGET ou CONTROL.

Ces variables sont totalement croisées dans un schéma d'expérience intra-sujet. 12 combinaisons de conditions (2 amplitudes \times 2 tailles de cibles \times 3 nombres de distracteurs) et 8 techniques sont donc présentées aux participants.

Pour chaque technique, les participants commencent par une phase d'entraînement : ils sélectionnent 2 jeux de cibles, un jeu correspondant à 12 sélections dans les 12 conditions. Ils sont ainsi familiarisés avec la nouvelle technique. Pour la phase d'expérimentation qui suit, les participants sélectionnent 3 blocs de 2 jeux de cibles. Pour chaque jeu de cibles, l'ordre des 12 conditions est aléatoire. Avant chaque jeu de 12 sélections, une *tâche de*

transition consiste à sélectionner une cible seule de 100 pixels. Les participants sont autorisés à se reposer à chaque tâche de transition. La cible de la tâche de transition est placée à gauche de l'écran. Cela permet de calculer la position de la prochaine cible avec des paramètres aléatoires de façon à ce que la cible soit entièrement affichée et à la distance voulue de la cible précédente. Les participants ne peuvent donc pas anticiper la position de la prochaine cible. Au mieux, ils peuvent remarquer une alternance entre les deux côtés de l'écran. L'expérience dure environ une heure pour chaque participant. Certains paramètres, comme les cartes de distance pour la technique DMAP, sont pré-calculés afin d'éviter les temps d'attente entre les sélections.

Résultats Les performances de pointage sont les temps de sélection et les taux d'erreurs. À la figure 5.5, les techniques sont ordonnées de gauche à droite en ordre croissant de temps de sélection (à gauche) et de taux d'erreurs (à droite). Pour comparer les temps de sélection entre les 8 techniques, nous utilisons le test de Mann-Whitney-Wilcoxon. Pour les taux d'erreurs, nous utilisons un test du χ^2 d'indépendance entre le succès d'une sélection et les 8 techniques.

Sans surprise, toutes les techniques d'expansion de cibles surpassent la technique CONTROL (3,14 s de temps de sélection moyen et 28,7 % d'erreurs) largement et significativement en temps de sélection et en taux d'erreurs ($p < 0,001$).

Taux d'erreurs CONTROL obtient un taux d'erreurs significativement supérieur à ceux de toutes les autres techniques ($p < 0,01$). EDGE et TILE sont les moins propices aux erreurs, avec un taux d'erreurs identique de 4,86 %. DMAP+TARGET a un taux d'erreurs de 6,48 % et TILE+TARGET de 7,11 %, tandis que les trois autres techniques ont un taux d'erreurs égal de 7,18 %.

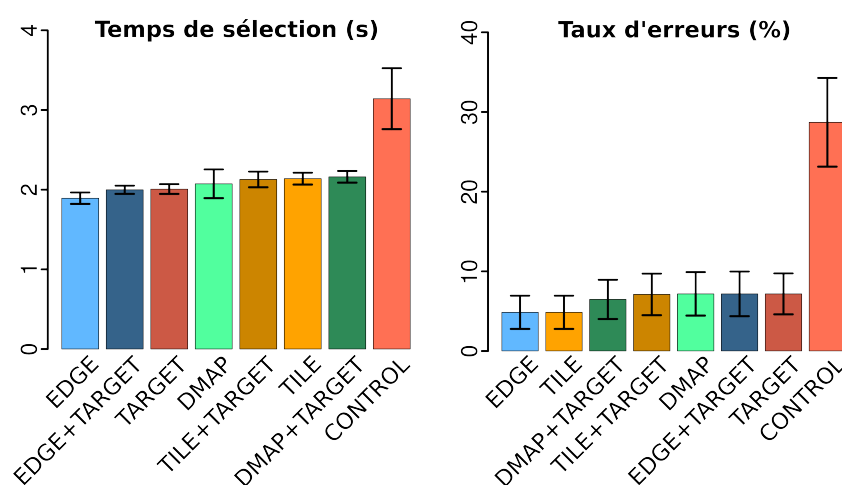


FIGURE 5.5 – Temps de sélection moyen et taux d'erreurs par technique, avec les intervalles de confiance à 95 %.

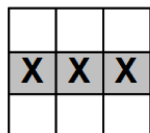
Temps de sélection Sans combinaison avec TARGET, les différences en temps de sélection moyen entre les 3 techniques aux visualisations différentes sont toutes significatives ($p < 0,01$). EDGE (1,98 s) est significativement la plus rapide des trois techniques ($p < 0,01$ avec TILE et DMAP) et la plus rapide de toutes les techniques comparées ($p < 0,01$ avec toutes les autres). De plus, TARGET (2 s) est significativement plus rapide que DMAP (2,07 s) et TILE (2,14 s).

DMAP est significativement plus rapide que DMAP+TARGET (2,16 s), tandis que la différence entre TILE et TILE+TARGET (2,139 s) n'est pas significative. Comme EDGE+TARGET est aussi significativement moins rapide que EDGE, nous concluons que la combinaison parallèle avec TARGET, *implicite*, *discrète* et *sur-cible*, n'apporte aucun gain substantiel en terme de performances de pointage à l'aide visuelle *explicite*, *statique* et *sur-espace* de VTE.

Discussion Les résultats de l'étude pilote ont montré les meilleurs temps de pointage pour EDGE suivie de EDGE-TARGET (figure 5.4 *a* et *b*), puis de TARGET sans visualisation du diagramme, suivies de toutes les autres variantes. En revanche, TARGET et EDGE-TARGET ont obtenu des taux d'erreurs élevés, contrairement à EDGE, qui a obtenu le plus faible taux d'erreurs. Quelque soit la variante de VTE, la combinaison parallèle avec TARGET n'a pas conduit à de meilleures performances.

Nous avons donc adopté EDGE, qui affiche le diagramme en lignes semi-transparentes, pour forme finale de VTE et nous avons abandonné la combinaison parallèle avec TARGET. L'aide visuelle atomique de VTE est donc *explicite*, *statique* et *sur-espace* (figure 5.1 et figure 5.2).

5.1.2 MTE



Manhattan Target Expansion (MTE) [59] ("expansion de cibles de Manhattan", figure 5.6) est une variante de VTE créée par exploration des algorithmes d'expansion dans le but de mieux intégrer le diagramme de Voronoï aux interfaces graphiques existantes. Or, celles-ci sont très souvent constituées de lignes verticales et horizontales et leurs éléments sont souvent répartis suivant ces deux directions.

MTE répartit l'espace entre les cibles grâce à un algorithme d'expansion spécifique : une tessellation de Voronoï calculée en distance de Manhattan (figure 5.6). Contrairement à la distance euclidienne utilisée couramment (ce que l'on entend communément par distance, ou distance "à vol d'oiseau"), la distance de Manhattan (ou distance rectilinéaire) est la distance la plus courte si l'on est contraint d'emprunter uniquement des chemins verticaux ou horizontaux, comme à Manhattan où les rues forment un quadrillage régulier du quartier. Mathématiquement, la distance de Manhattan entre deux points est la somme de la valeur absolue des différences entre les coordonnées cartésiennes des deux points.

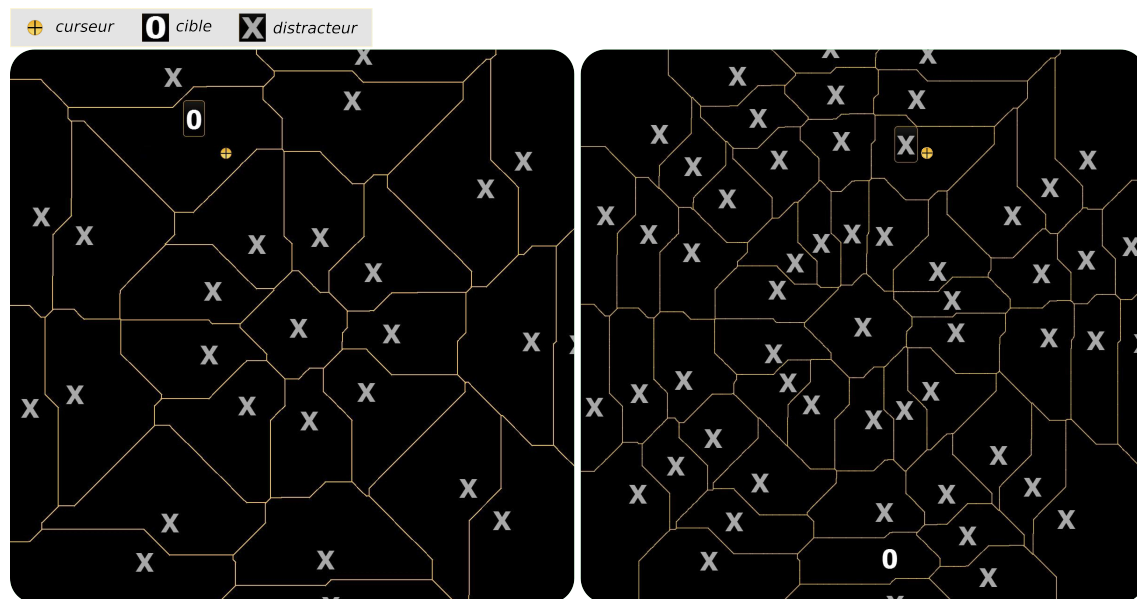
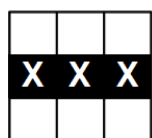


FIGURE 5.6 – MTE. À droite, la densité de cibles est plus élevée qu'à gauche.

Dans cette géométrie, les diagrammes de Voronoï n'incluent que des lignes verticales, horizontales ou orientées à 45° (figure 5.6). Le partage de l'espace résultant peut alors être plus esthétique (car plus de symétrie et plus de parallélisme) et plus cohérent que VTE avec l'interface graphique sous-jacente. MTE a les mêmes caractéristiques que VTE selon les axes de conception que nous avons définis (figure 5.1) : *explicite*, *statique* et *sur-espace* (figure 5.2).

5.1.3 eVTE



Erasable Voronoï-based Target Expansion (eVTE) [59] (le "VTE effaçable") est une variante majeure de VTE. eVTE est fondé sur un constat : avec VTE, l'affichage statique du diagramme de Voronoï peut cacher des informations utiles et perturber la perception visuelle et la lisibilité de l'interface graphique sous-jacente. Pour répondre à ce problème, nous avons exploré l'axe *Dynamacité* de notre espace de conception. Nous avons rendu VTE dynamique-continu en couplant la transparence du diagramme de Voronoï avec la vitesse du curseur à l'aide d'une fonction sigmoïde. Cette fonction permet des transitions douces à haute et à basse vitesse. À haute vitesse, le diagramme disparaît de façon progressive et continue pour laisser apparaître l'interface graphique et disparaître totalement au-delà de 1500 pixels/s (valeur empirique pour le pointage à distance avec une technique de ray-casting). Cette technique offre également la possibilité d'effacer temporairement le diagramme en secouant le curseur, d'où le nom de "VTE effaçable". En faisant disparaître le diagramme à haute vitesse seulement, les bénéfices escomptés apportés par l'affichage du diagramme complet perdurent. En effet, une grande vitesse de déplacement du curseur correspond au pic de vitesse de la phase balistique du mouvement, la seule phase sous contrôle "en boucle ouverte", c'est-à-dire la seule phase où les aides visuelles ne sont pas utilisées pour

corriger le mouvement, selon le MIIO [88]. Le diagramme reste donc visible pendant la phase préliminaire, le début de la phase balistique et réapparaît pour la phase corrective, durant laquelle la vitesse du curseur est nettement plus faible que durant le pic de vitesse balistique.

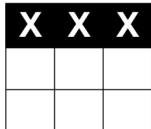
eVTE est une technique d'un nouveau type dans notre classification (figure 5.1) car elle est *explicite*, *continue* et *sur-espace* (figure 5.2). À notre connaissance, c'est une possibilité de conception jusque-là inexplorée.

5.1.4 Conclusion des nouvelles techniques *sur-espace*

VTE, MTE et eVTE ont été conçues avec une aide visuelle *sur-espace*, *statique* pour VTE et MTE et *continue* pour eVTE. Une faiblesse de VTE, qui encombre tout l'écran tout au long du geste avec un diagramme de Voronoï réalisé sur tout l'espace visuel, nous a incités à concevoir d'autres techniques comme eVTE. Pour poursuivre cette exploration, nous avons considéré les axes *Élément augmenté* et *Dynamacité* puis la combinaison d'aides visuelles.

5.2 Nouvelles techniques *sur-curseur*

5.2.1 Expansion Lens



Inspirés de VTE et ses variantes, nous avons conçu Expansion Lens [60] (la "lentille d'expansion", figure 5.7). L'objectif est de fournir une aide visuelle *explicite*, comme VTE, mais *sur-curseur*. Pour fournir une aide visuelle *sur-curseur* simple, nous avons choisi une zone de dimension fixe centrée sur le curseur, la lentille, dont la fonction est de révéler les tailles étendues des cibles autour du curseur (figure 5.8). Expansion Lens fonctionne donc comme une lentille magique qui révèle l'algorithme d'expansion sous-jacent. Avec Expansion Lens comme aide visuelle et la tessellation de Voronoï comme algorithme d'expansion, les utilisateurs peuvent observer à travers la lentille une portion du diagramme de Voronoï (figure 5.7).

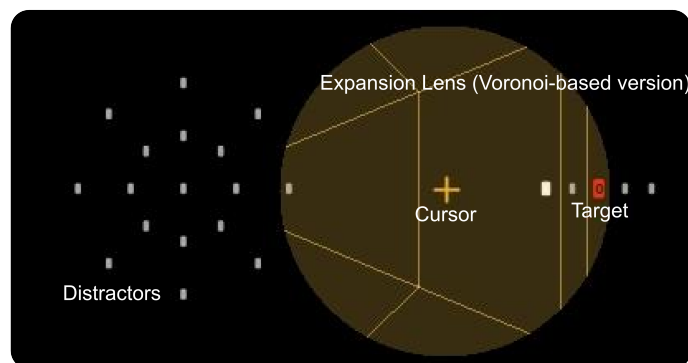


FIGURE 5.7 – Expansion Lens.

Les mécanismes d'optimisation possibles au début du geste grâce à VTE ne sont pas mis en jeu avec Expansion Lens car l'aide visuelle reste *sur-curseur*, néanmoins :

- Expansion Lens devrait offrir une bonne optimisation des sous-mouvements correctifs et une bonne prévention des erreurs grâce à son caractère *explicite*.
- Expansion Lens évite une éventuelle surcharge visuelle de l'écran – qui peut se produire avec l'affichage (comme VTE) d'un diagramme de Voronoï complet de façon *statique* – en utilisant une aide-visuelle *sur-curseur* et *continue* (changement de valeur sur les deux axes de conception *Dynamacité* et *Élément augmenté* de l'aide visuelle).

La lentille est d'une couleur semi-transparente afin que l'utilisateur puisse connaître à tout moment son étendue exacte. Comme la lentille est centrée sur le curseur, elle bouge constamment avec le curseur lorsque celui-ci bouge (figure 5.8). Ainsi, Expansion Lens utilise une aide visuelle d'un nouveau type et peuple une case vide de notre espace de conception : celle des aides visuelles *explicite*, *continue* et *sur-curseur* (figure 5.1).

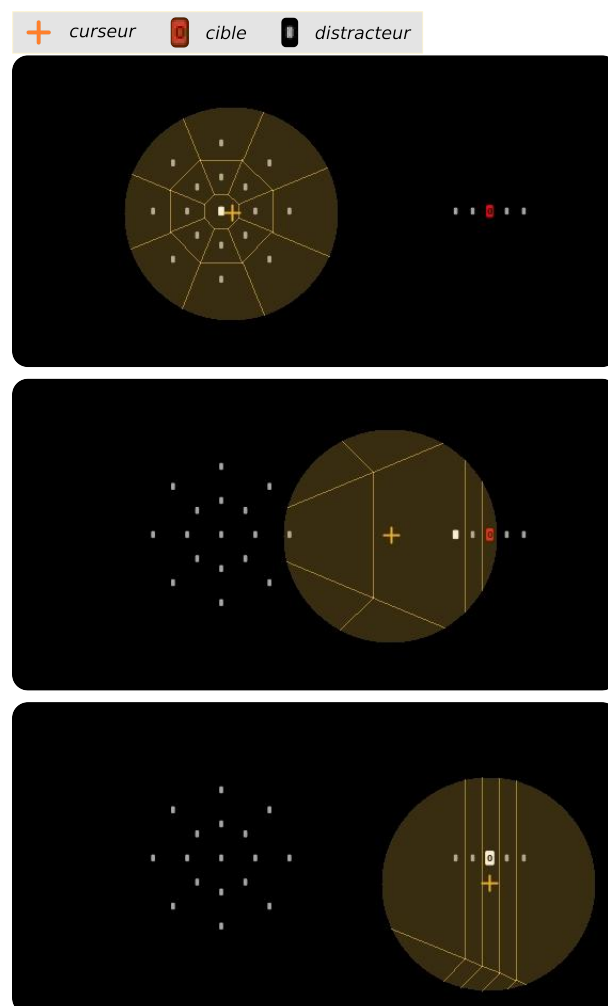
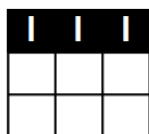


FIGURE 5.8 – Expansion Lens : l'utilisateur voit autour du curseur la répartition de l'espace moteur calculée par l'algorithme d'expansion. La cible désignée est blanche.

Nous avons conduit une étude expérimentale sur un ordinateur de bureau avec 10 participants afin d'explorer deux paramètres de conception : la taille et la forme de la lentille. Les résultats indiquent que les utilisateurs préfèrent une lentille ronde (8 participants sur 10) plutôt qu'une lentille carrée, et avec une taille intermédiaire (un diamètre entre 100 et 300 pixels pour une utilisation de bureau avec une résolution de 1920×1200). Les résultats de cette étude montrent également que la forme ronde est significativement plus rapide que la forme carrée. Nous pensons que la forme ronde permet de renforcer la métaphore de la lentille. La version finale d'Expansion Lens utilise donc une lentille ronde de 200 pixels de diamètre, comme à la figure 5.8.

5.2.2 Rope Cursor



Comme Expansion Lens, Rope Cursor [59] (le "curseur-corde") est une technique *sur-curseur* qui utilise la tessellation de Voronoï comme algorithme d'expansion. Son nom provient de la Ninja Rope, un accessoire disponible dans la série de jeux vidéos Worms. Cet accessoire très utile dans le jeu permet de se balancer de point d'attache en point d'attache, comportement que l'on peut reconnaître à l'utilisation de Rope Cursor.

Le rationnel de conception de Rope Cursor repose sur une réduction de la gêne visuelle créée par la bulle de Bubble Cursor [67, 126, 131] et une amélioration de l'*Observabilité de l'expansion* de cette technique. Pour cela, nous utilisons des lignes semi-transparentes au lieu d'une bulle semi-transparente (figure 5.9) : une ligne principale (la "corde") relie le curseur à la cible la plus proche. Puis des lignes supplémentaires plus fines (les "mini-cordes") indiquent par leur comportement la proximité des cibles avoisinantes.

L'algorithme qui régit le comportement des mini-cordes est illustré par la figure 5.10. Il est conçu pour apporter aux mini-cordes un comportement, comme celui d'un végétal, qui fait que les mini-cordes "poussent" en direction des cibles qui sont suffisamment proches. Le détail de cet algorithme est cependant moins trivial que son intention.

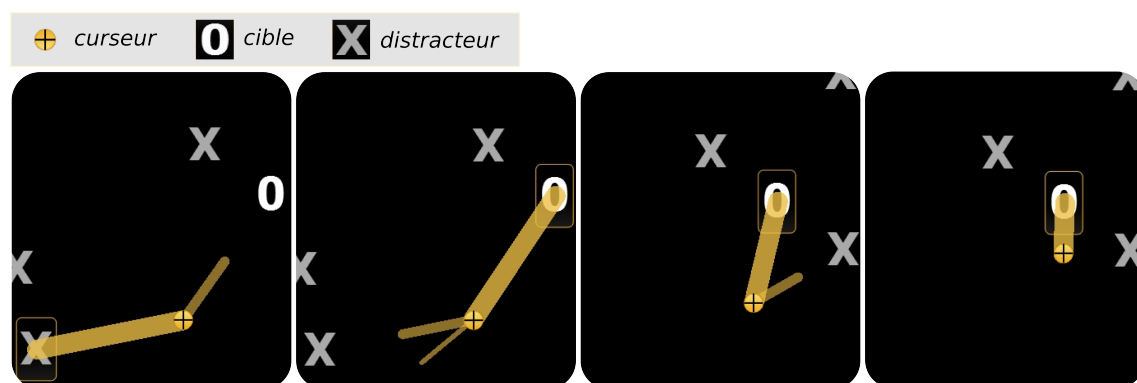


FIGURE 5.9 – Rope Cursor : la ligne principale indique la cible la plus proche (tessellation de Voronoï) tandis que les lignes secondaires indiquent les cibles voisines.

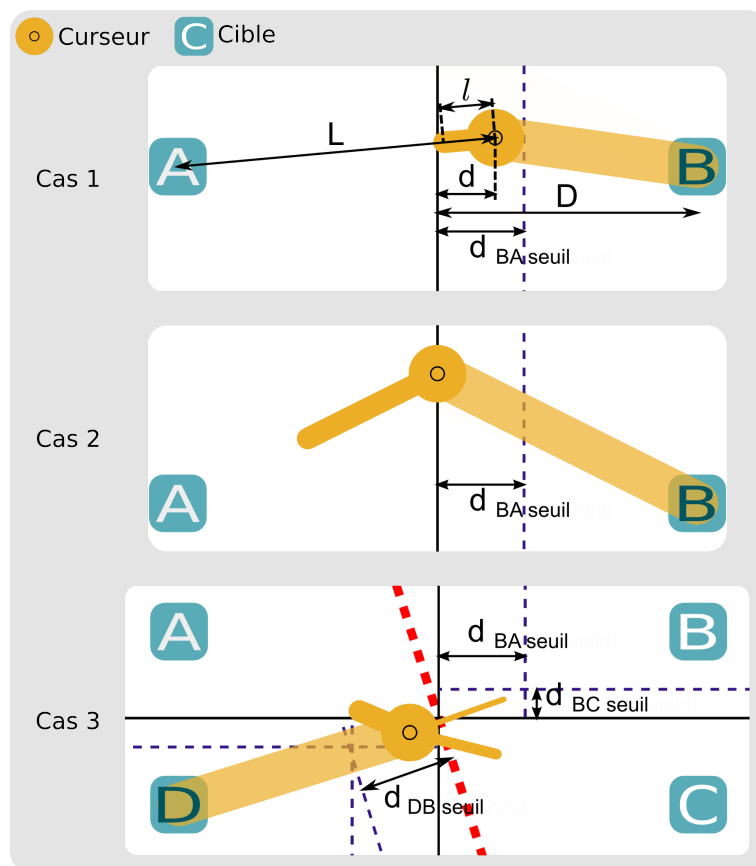


FIGURE 5.10 – Rope Cursor : la ligne principale indique la cible la plus proche (tessellation de Voronoï) tandis que les lignes secondaires indiquent les cibles voisines.

Quand le curseur est plus proche d'un bord de la cellule de Voronoï survolée qu'un certain seuil de distance (figure 5.10 cas 1 et 2 : $d < d_{BA\text{seuil}}$), une mini-corde apparaît sur le curseur. Nous avons défini arbitrairement sur la base de nos observations ce seuil de distance à un tiers de la distance séparant la cible désignée de la bordure correspondante (figure 5.10 : $d_{BA\text{seuil}} = D/3$). Les seuils de distance peuvent être différents dans une même cellule, comme on peut le voir à la figure 5.10 ($d_{BA\text{seuil}} \neq d_{BC\text{seuil}}$). Ceci est une conséquence des propriétés intrinsèques au diagramme de Voronoï. Quand la cellule d'une cible voisine partage seulement un coin avec la cellule survolée (figure 5.10 cas 3 : cibles D et B), le seuil de distance est égal à un tiers de la distance de la cible désignée au coin en question (figure 5.10 cas 3 : la ligne épaisse en pointillé représente la médiatrice du segment DB). La mini-corde est orientée dans la direction de la cible dont le bord de la cellule est proche du curseur. Ainsi, les mini-cordes désignent directement les cibles voisines dont le curseur s'approche. Par exemple à la figure 5.10 cas 1 et 2 : la mini-corde indique la cible A ; cas 3 : les mini-cordes indiquent les cibles A, B et C, dont les cellules sont proches.

Tant que le curseur s'approche du bord de la cellule, la mini-corde croît linéairement dans un intervalle de 0 à $L/2$ (figure 5.10 et 5.11). Ainsi, quand le curseur est sur le point de traverser le bord (figure 5.10 cas 2 et figure 5.11 troisième vignette), la mini-corde est longue comme la moitié de la distance entre le curseur et la cible voisine

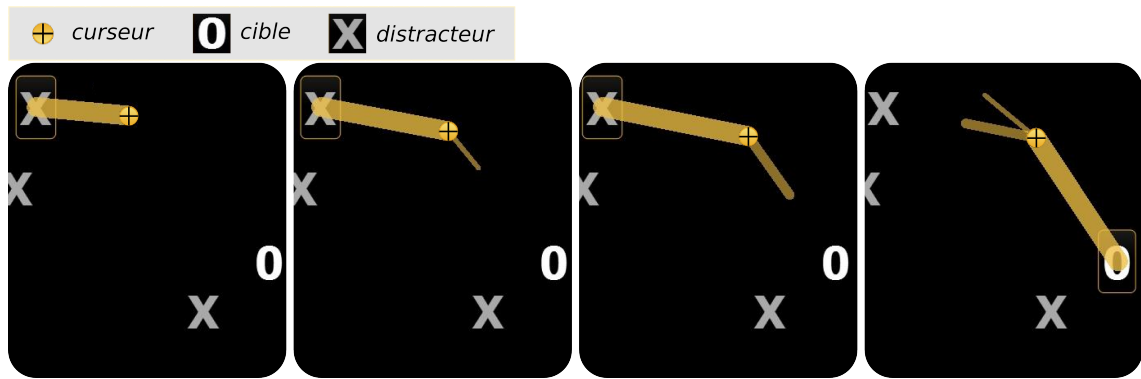


FIGURE 5.11 – Rope Cursor : la transition au changement de cible désignée est progressive. La mini-corde qui annonce le franchissement du bord de la cellule survolée s’allonge avant de devenir la corde principale. Elle grossit également car l’ordre de proximité à la cible encore désignée change. Lorsque le bord de la cellule survolée est franchi (vignette de droite), une nouvelle cible est désignée. La mini-corde la plus épaisse désigne alors la cible précédente comme la voisine la plus proche du curseur. Une deuxième mini-corde moins épaisse indique la présence et la proximité d’une cible qui n’est pas visible sur cette image.

désignée. La transition de franchissement du bord est ainsi rendue fluide (figure 5.11). Ce comportement des mini-cordes représente visuellement la distance restante jusqu’à la cellule de la cible voisine (figure 5.10 : d) car à tout moment, le rapport $\frac{l}{L/2}$ est égal au rapport $\frac{d_{BA_seuil}-d}{d_{BA_seuil}}$.

Au-delà de sa longueur indiquant la proximité des cellules voisines, l’épaisseur d’une mini-corde traduit l’ordre de proximité des cibles voisines : plus la mini-corde est large, plus la cible est proche du curseur. La largeur de chaque mini-corde est égale à la moitié de la largeur de la (mini-)corde immédiatement plus large : par exemple, 24 pixels pour la corde principale, 12, 6 et 3 pixels pour les mini-cordes, comme sur les figures 5.9, 5.10 cas 3 et 5.11.

Rope Cursor se classe dans la catégorie des techniques *implicites continues* et *sur-curseur* (figure 5.1 et 5.2) et des curseurs étendus (chapitre 3 section 3.2.4.2). Elle apporte cependant une meilleure *Observabilité de l’expansion* que Bubble Cursor, tout en restant *implicite*. En effet, les 3 mini-cordes (mais plus de mini-cordes est possible !) montrent la proximité de 3 cibles voisines simultanément (figure 5.10 : cas 3), tandis que Bubble Cursor est contraint d’ajuster sa bulle à la cible désignée et à une cible voisine seulement [53] (chapitre 3, section 3.2.4.2).

5.2.3 Conclusion des nouvelles techniques *sur-curseur*

Expansion Lens est la première technique *explicite, continue* et *sur-curseur*. Reposant sur une lentille centrée sur le curseur qui permet de voir les tailles étendues des cibles autour de celui-ci, elle répond à une volonté de garder uniquement une partie du diagramme de VTE et a été conçue par exploration des axes *Élément augmenté* et *Dynamacité*. Rope Cursor, une technique de curseur étendu, indique la cible la plus proche par une ligne

épaisse et indique la proximité d'autres cibles grâce à d'autres lignes dynamiques moins épaisses. C'est une technique très dynamique, *implicite*, *continue* et *sur-curseur*. Dans la section suivante, l'exploration de l'espace de conception repose sur la combinaison d'aides visuelles. Rope Cursor, Cell Painting et TARGET sont combinés à VTE pour améliorer les performances et l'acceptabilité de VTE.

5.3 Combinaisons d'aides visuelles : trois nouvelles techniques

Nous avons conçu trois autres nouvelles techniques d'expansion de cibles [59] (figures 5.12, 5.13 et 5.14) en cherchant à répondre au problème déjà traité lors de la conception d'eVTE et d'Expansion Lens : la surcharge visuelle causée par l'affichage permanent par VTE du diagramme de Voronoï entier. Nous avons cette fois-ci exploré la combinaison d'aides visuelles [59]. Comme VTE a théoriquement un grand potentiel d'amélioration du début du mouvement, nous gardons VTE tel quel durant la phase préliminaire et le tout début de la phase balistique seulement, c'est-à-dire jusqu'à ce que le mouvement de pointage soit commencé. Nous combinons ensuite l'aide visuelle de VTE avec trois aides visuelles différentes, que nous utilisons durant la phase balistique et la phase corrective : Rope Cursor (*implicite*, *continu*, *sur-curseur*), Cell Painting (*explicite*, *discret*, *sur-cible*) et TARGET (*implicite*, *discret*, *sur-cible*) (figure 5.1). Ainsi, nous maintenons le bénéfice de VTE sur le début du geste, tout en exploitant d'autres aides visuelles, avec chacune leurs bénéfices pour la fin du geste. Les trois techniques résultantes sont nommées respectivement VTetoROPE (figure 5.12), VTetoCELL (figure 5.13) et VTetoTARGET (figure 5.14).

Comme Dynaspot [34], le changement d'aide visuelle se produit au tout début de la phase balistique, dès que le geste de pointage a commencé. Dès que le curseur dépasse un certain seuil de vitesse (500 pixels/s pour un pointage à distance avec une technique de

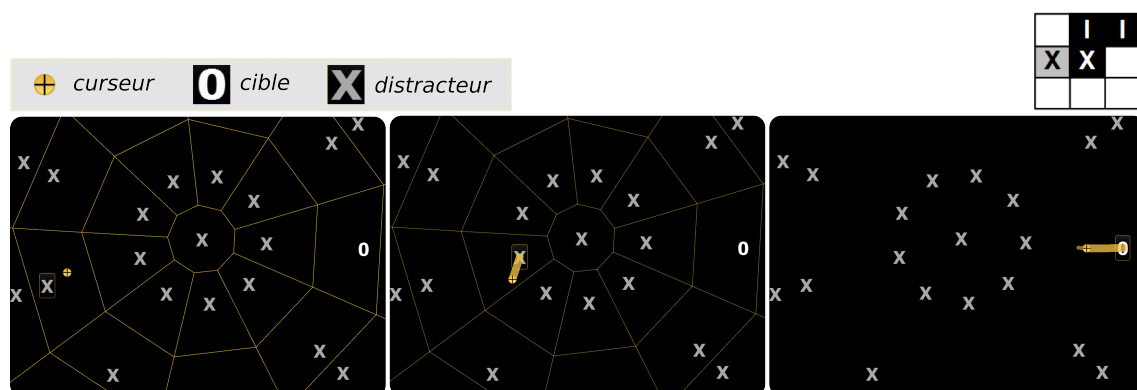


FIGURE 5.12 – VTetoROPE. De gauche à droite, la combinaison d'aides visuelles évolue au cours des phases du mouvement : phase préliminaire (VTE seulement), phase balistique (VTE disparaissant progressivement et Rope Cursor) et phase corrective (Rope Cursor seulement).

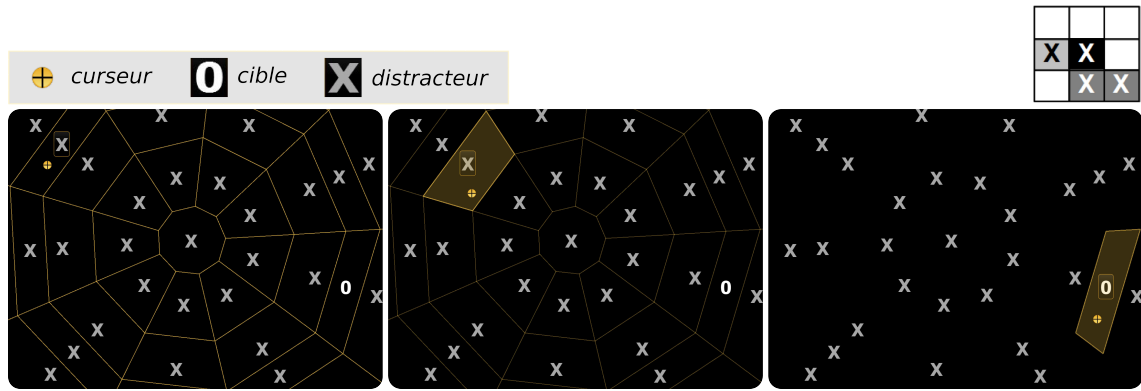


FIGURE 5.13 – VTEtoCELL. De gauche à droite, la combinaison d'aides visuelles évolue au cours des phases du mouvement : phase préliminaire (VTE seulement), phase balistique (VTE disparaissant progressivement et Cell Painting) et phase corrective (Cell Painting seulement).

ray-casting), le diagramme de Voronoï de VTE disparaît progressivement en 300 ms avec un effet visuel de fondu : sa transparence augmente de façon quadratique de l'opacité totale à la transparence totale. Ainsi, pendant 300 ms au cours de la phase balistique, ces techniques utilisent une version dynamique-continue de VTE, comme l'illustre la case noire au centre des matrices de la figure 5.2, matrices aussi rappelées aux figures 5.12, 5.13 et 5.14. Cette disparition du diagramme se produit parallèlement à l'apparition d'une autre aide visuelle : Rope Cursor ou Cell Painting. Dans VTEtoROPE et VTEtoCELL, Rope Cursor et Cell Painting apparaissent au début de la phase balistique (deux phases seulement), une fois que le mouvement a commencé (figures 5.12 et 5.13). Au contraire pour VTEtoTARGET, TARGET est active durant les trois phases du mouvement (figure 5.14). Nous explorons donc deux schémas de combinaison différents : une combinaison *séquentielle* d'aides visuelles avec VTEtoROPE et VTEtoCELL et une combinaison *parallèle* de VTE et TARGET avec VTEtoTARGET.

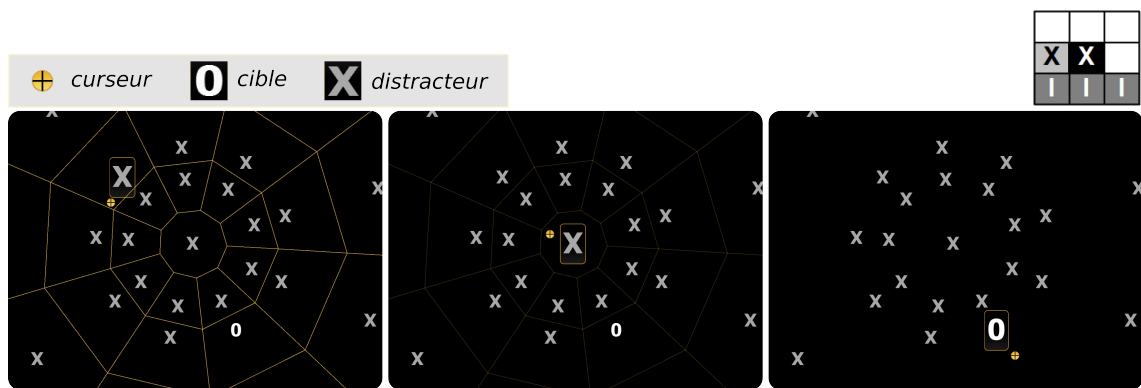


FIGURE 5.14 – VTEtoTARGET. De gauche à droite, la combinaison d'aides visuelles évolue au cours des phases du mouvement : phase préliminaire (VTE et TARGET), phase balistique (VTE disparaissant progressivement et TARGET) et phase corrective (TARGET seulement).

5.4 Extensions

L'espace de conception définit 16 types d'aides visuelles et nous avons montré son pouvoir génératif par la conception de nouvelles techniques. De nombreuses autres techniques sont à concevoir. Au-delà de l'exploration de l'espace de conception, nous avons identifié un ensemble d'extensions lors de la conception de ces nouvelles techniques.

5.4.1 Améliorer l'aspect graphique

Une étude sur l'apparence graphique des nouvelles techniques permettrait d'améliorer leur acceptabilité par les utilisateurs et leur intégrabilité aux interfaces graphiques existantes. En 2012, Fraher [47] a mené une telle démarche pour Bubble Cursor. Une telle étude pourrait s'avérer particulièrement bénéfique pour VTE, qui surcharge visuelle l'interface graphique existante. Plusieurs pistes existent, comme adapter les couleurs du diagramme de Voronoï à l'interface graphique sous-jacente (utiliser des couleurs inverses par exemple) ou encore moduler la transparence du diagramme différemment d'eVTE et des nouvelles techniques combinées. Une piste prometteuse est l'utilisation d'effets graphiques introduisant artificiellement une dimension de profondeur dans l'interface, par exemple les effets de relief, d'empreinte ou d'ombre.

5.4.2 Changement de mode implicite

Parmi les techniques présentées dans ce chapitre, les deux techniques *sur-curseur*, Rope Cursor et Expansion Lens, peuvent être aisément adaptées à une utilisation qui conserve la sélection au-pixel sans changer d'algorithme d'expansion. Pour cela, il convient d'adapter à ces techniques le mécanisme de changement de mode implicite défini par la technique Dynaspot [34]. Ces adaptations sont possibles pour Rope Cursor et Expansion Lens car ce sont des techniques *sur-curseur*, comme Dynaspot. L'étude de la technique Implicit Fan Cursor [126] propose également une telle adaptation. Pour ces techniques, le bénéfice de l'adaptation est le gain de la sélection au-pixel à faible vitesse du curseur. Pour VTE, MTE, eVTE, VTetoROPE, VTetoCELL et VTetoTARGET, l'adaptation au changement de mode implicite est plus complexe. Ces techniques n'augmentant pas le curseur durant la phase préliminaire du geste, le curseur ne peut pas servir de support visuel au changement de mode implicite, comme avec Dynaspot, Rope Cursor et Expansion Lens.

5.4.2.1 Fonctionnement du changement de mode implicite

Ce basculement entre le mode *normal*, dans lequel les cibles ne sont pas étendues dans l'espace moteur, et le mode *expansion*, dans lequel les utilisateurs profitent d'une expansion des cibles, repose sur un contrôle de la vitesse du curseur ("Speed" à la figure 5.15).

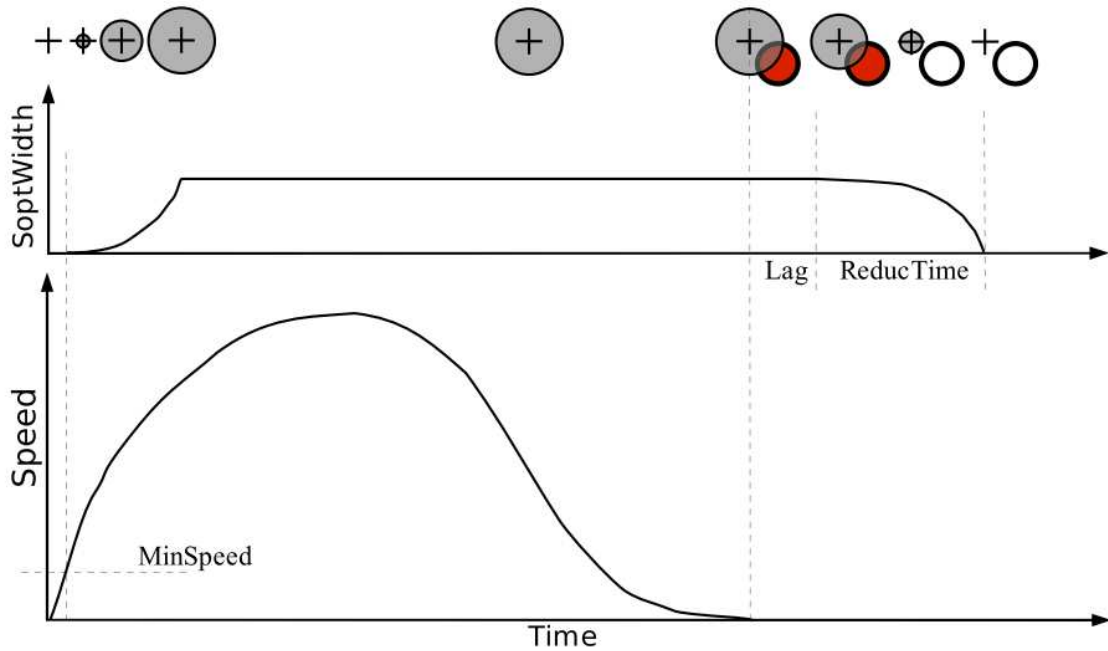


FIGURE 5.15 – Changement de mode implicite : la taille de la bulle de Dynaspot est une fonction de la vitesse du curseur. Figure issue de [34].

A basse vitesse, le mode *normal* est activé, la sélection est au-pixel. Lorsque le curseur accélère et franchit un seuil de vitesse de 100 pixels/s ("MinSpeed" à la figure 5.15), le mode *expansion* est activé progressivement : la bulle de Dynaspot et la taille étendue des cibles grossissent. À haute vitesse, les cibles sont totalement étendues et la bulle atteint sa taille maximale (32 pixels de diamètre).

Le principe de ce changement de mode est de détecter le début d'un geste de pointage à l'aide d'un seuil de vitesse (détection du sous-mouvement balistique) et d'activer ainsi l'expansion de cibles uniquement lors des gestes de pointage, qui ne constituent pas la totalité des interactions possibles avec un curseur. Le mode *expansion* est légèrement maintenu (300 ms) en fin de geste pour laisser à l'utilisateur le temps de cliquer ("Lag" et "ReducTime" à la figure 5.15).

5.4.2.2 Adaptation de Rope Cursor

L'adaptation de Rope Cursor que nous proposons consiste à appliquer la fonction qui régit la taille de la bulle de Dynaspot ("SpotWidth" à la figure 5.15), à la longueur de chaque (mini-)corde de Rope Cursor, en considérant à chaque fois la longueur de la (mini-)corde (calculée selon la méthode décrite à la section 5.2.2) comme la taille maximale de la bulle de Dynaspot. Le résultat serait une "poussée" rapide des cordes durant l'accélération du curseur vers leur cible respective et une rétractation des cordes 120 ms après l'arrêt du curseur, la rétractation durant 180 ms, de la même manière que Dynaspot [34]. Une étude plus approfondie permettrait d'optimiser pour Rope Cursor la durée avant la rétractation des cordes (*Lag* à la figure 5.15) et la durée de la rétractation elle-même (*ReducTime* à la

figure 5.15). Tandis que la bulle de Dynaspot se limite à 32 pixels de diamètre et limite ainsi la taille étendue des cibles, Rope Cursor ainsi modifié profiterait à pleine vitesse de toute la tessellation de Voronoï. Ce qu'apporte cette adaptation à Rope Cursor est alors la sélection au-pixel à basse vitesse et durant les phases de transition entre faible et haute vitesse (croissance et rétractation des cordes).

5.4.2.3 Adaptation d'Expansion Lens

Pour Expansion Lens, deux adaptations sont possibles. La première consiste à appliquer la fonction qui régit la taille de la bulle de Dynaspot à la transparence de la lentille : la lentille est invisible à faible vitesse du curseur (sélection au-pixel), elle devient pleinement visible à grande vitesse (mode *expansion*) et elle retourne progressivement à la transparence totale après l'arrêt du mouvement du curseur. La deuxième adaptation possible consiste à gérer la taille de la lentille exactement comme la taille de la bulle de Dynaspot. Pour ces deux adaptations, le mode *expansion* n'est pas activé lors des phases de transition entre faible et grande vitesse (mode *normal* : sélection au-pixel). Ainsi c'est uniquement lorsque la lentille a son opacité maximale ou sa taille maximale que le mode *expansion* est activé.

5.4.3 Autres algorithmes d'expansion de cibles

5.4.3.1 Voronoï en distances de Manhattan

Les aides visuelles présentées dans ce chapitre peuvent toutes être adaptées à un grand nombre d'algorithmes d'expansion. Par exemple, MTE est une adaptation directe de l'aide visuelle de VTE à la tessellation de Voronoï en distance de Manhattan. Nous avons adapté au cours de nos tests (chapitre 9) d'autres techniques à cet algorithme d'expansion, notamment Bubble Cursor, eVTE et Cell Painting (créant ainsi les techniques eMTE et MCell Painting). En effet, pour le comportement de Bubble Cursor, nous avons calculé les distances habituellement euclidiennes en distance de Manhattan. Rope Cursor peut être modifié de la même façon. Ensuite, comme il est possible de calculer le diagramme complet en distance de Manhattan, il est possible de l'afficher pour créer MTE, de dynamiser cet affichage pour créer eMTE et de calculer la cellule survolée pour créer MCell Painting. Ces deux techniques sont visibles au chapitre 9 à la figure 9.2. L'adaptation d'Expansion Lens suit le même principe : remplacer un diagramme par un autre. Il est alors possible de combiner ces aides visuelles, tant que l'algorithme d'expansion reste constant au cours du geste.

5.4.3.2 Autres variantes de Voronoï

La tessellation de Voronoï est, par définition, ouverte à d'autres métriques que la distance euclidienne [27, 111]. Le distance de Manhattan est un exemple mais il existe de

nombreuses autres variantes possibles [27], comme la distance de Mahalanobis [111]. Il est également possible d'approximer un diagramme de Voronoï. Le résultat est un diagramme laissant entre les cellules de l'espace libre utilisable pour une sélection au-pixel. De plus, pondérer les cibles selon un ou plusieurs critères, par exemple leur fréquence d'utilisation, produirait une nouvelle version du diagramme sans espace libre.

Enfin, une autre approche est d'adapter la position des cibles, pour obtenir un diagramme de Voronoï centroïdal, où cibles et tailles étendues des cibles ont le même centre de masse [41]. Cette adaptation de l'interface graphique apporterait plus de symétrie au diagramme et pourrait améliorer les performances de pointage [101, 122].

5.4.3.3 Tripler la taille des cibles

Un autre exemple bien différent concerne l'adaptation à un algorithme d'expansion dont le principe est simple : tripler la taille de la cible dans l'espace moteur. Un tel algorithme d'expansion laisse de l'espace libre entre les tailles étendues des cibles, si la densité de cibles n'est pas trop élevée. Il est alors possible de profiter de l'interaction au-pixel et de l'expansion de cibles (figure 5.16). L'adaptation de Rope Cursor à cet algorithme est possible sans changement fondamental dans son comportement : la corde principale n'apparaît alors que lorsque le curseur entre dans la taille étendue d'une cible et les mini-cordes réagissent de la même façon aux cibles voisines, en prenant en compte les mêmes grandeurs : distance au bord de la taille étendue de la cible voisine, distance à la cible voisine elle-même, ordre de proximité des cibles voisines. Lorsque le curseur survole un espace libre, où la sélection est au-pixel, en s'approchant d'une cible, une mini-corde apparaît en direction de la cible et grandit jusqu'à devenir la corde principale de Rope Cursor au moment où le curseur entre dans la taille étendue de la cible.

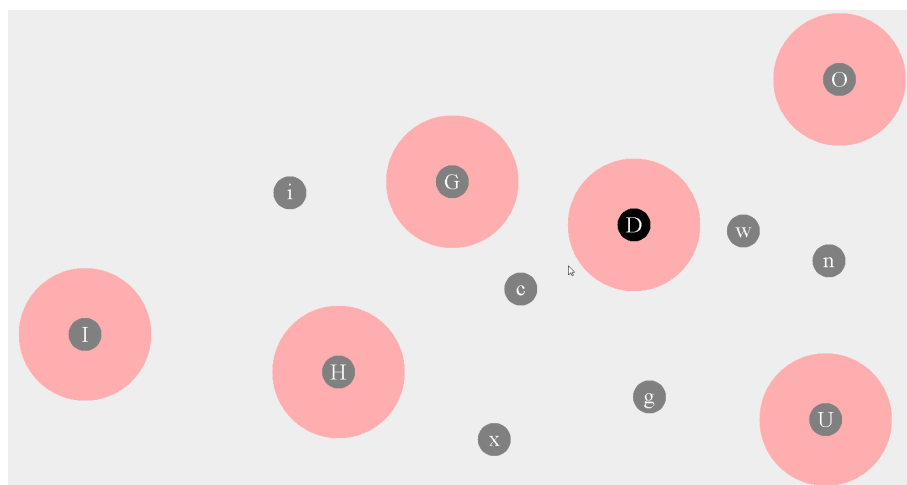


FIGURE 5.16 – Exemple d'aide visuelle avec un algorithme d'expansion consistant à tripler la taille des cibles. Les cibles sont grises et contiennent une lettre majuscule. La cible-objectif est noire. Leur taille étendue est en rose. Le fond d'écran est gris clair.

Figure modifiée de [4].

Pour les aides visuelles de VTE, eVTE et Expansion Lens, il est nécessaire de distinguer graphiquement l'espace libre de la taille étendue des cibles. Le principe de VTE est d'afficher la taille étendue de toutes les cibles dès le début du geste. Dans ce contexte, l'enjeu de conception est de décider comment présenter les cibles étendues. Une solution est d'utiliser une couleur semi-transparente pour représenter la taille étendue des cibles (figure 5.16). Avec VTE ainsi adapté, l'utilisateur verrait un halo semi-transparent autour de chaque cible. La lentille d'Expansion Lens peut alors être transparente et délimitée par une ligne. Au travers de la lentille, l'utilisateur visualise autour de chaque cible un halo semi-transparent qui indique sa taille étendue. Pour adapter eVTE à cet algorithme d'expansion, l'intervalle de variation de la transparence du diagramme doit être réduit pour aller de la transparence totale à la semi-transparence. En effet, avec eVTE ainsi adapté, c'est le halo semi-transparent autour de chaque cible qui disparaît lorsque la vitesse du curseur augmente.

Utiliser un tel algorithme d'expansion nécessiterait toutefois de traiter les problèmes d'ambiguïtés qui apparaissent lorsque les cibles sont trop proches. Sélectionner la cible la plus proche du curseur est la solution employée par Dynaspot [34] pour traiter ce problème. L'algorithme d'expansion de Dynaspot est donc, en réalité, une tessellation de Voronoï "tronquée" [41].

5.4.4 VTE to Expansion Lens

Une autre possibilité de combinaison intéressante d'aides visuelles est une combinaison séquentielle de VTE et d'Expansion Lens au cours du même geste, sur le modèle des techniques combinées présentées dans ce chapitre. Le diagramme de VTE, présent jusqu'au démarrage de la phase balistique, disparaîtrait au cours du geste balistique avec un effet de fondu, comme pour VTetoCELL par exemple. Seule resterait jusqu'à la fin de la sélection la portion de diagramme contenue dans la lentille, apportant ainsi une aide visuelle *explicite* autour du curseur pour optimiser l'éventuelle phase corrective. Cette possibilité de combinaison n'a pas été implémentée, mais elle présente l'intérêt d'une certaine continuité dans l'aide visuelle.

Conclusion

Dans ce chapitre, nous avons présenté notre contribution en termes de techniques de pointage, dans la catégorie des expansions visuelles et motrices de cibles. Huit techniques ont été présentées : VTE, MTE, eVTE, Expansion Lens, Rope Cursor et trois techniques combinées : VTetoROPE, VTetoCELL et VTetoTARGET.

Ces techniques ont été créées par exploration de l'espace de conception. Dans le cas de VTE, l'aide visuelle devait être *statique*, *explicite* et *sur-espace* car l'étude du MIIO laissait présager d'une grande efficacité de ce type d'aide visuelle. Cependant, la possible

gène visuelle créée par VTE a motivé la création de variantes par exploration des algorithmes d'expansion pour MTE, de l'axe *Dynamacité* pour eVTE et de l'axe *Élément augmenté* pour Expansion Lens. Rope Cursor est né de l'exploration de l'axe *Observabilité de l'expansion* à partir de Bubble Cursor. Les techniques combinées sont issues de l'exploration de la combinaison d'aides visuelles selon un schéma destiné à préserver les avantages de VTE, tout en libérant l'écran dès que le geste a commencé. De nombreuses adaptations de ces techniques conçues sont possibles.

Dans le chapitre suivant, nous expliquons comment nous avons expérimentalement comparé ces techniques et tiré des conclusions sur les techniques d'expansion en fonction des axes de conception.

Chapitre 6

Exploration expérimentale de l'espace de conception

Sommaire

Introduction	102
6.1 Expérience 1 : VTE	103
6.1.1 Introduction	103
6.1.2 Protocole expérimental	104
6.1.2.1 Participants et appareillage	104
6.1.2.2 Détails des techniques implémentées	104
6.1.2.3 Tâche et procédure	105
6.1.2.4 Conditions	106
6.1.3 Résultats : Performances de pointage	107
6.1.3.1 Méthode	107
6.1.3.2 Calcul de d_E	107
6.1.3.3 Comparaison des dispositifs de pointage	108
6.1.3.4 Performances pour le ray-casting	108
6.1.3.5 Performances pour le pointage main libre	109
6.1.4 Résultats : Préférences des utilisateurs	110
6.1.5 Discussion	111
6.1.6 Conclusion de l'expérience 1	112
6.2 Expérience 2 : Axes de conception	113
6.2.1 Introduction	113
6.2.2 Agencement de cibles pour des tessellations non-centroïdales	114
6.2.2.1 Décalage du point de chute des mouvements	114
6.2.2.2 Agencement des cibles	115
6.2.3 Protocole expérimental	117
6.2.3.1 Participants et appareillage	117
6.2.3.2 Détail des techniques implémentées	117
6.2.3.3 Tâche et procédure	118
6.2.3.4 Conditions	118

6.2.4	Résultats	119
6.2.4.1	Résultats quantitatifs	119
	Méthode	119
	Décalage du point de chute des mouvements	119
	Temps de sélection	119
	Taux d'erreurs	120
6.2.4.2	Scores SUS et résultats qualitatifs	120
6.2.5	Discussion	121
6.2.5.1	Axe <i>Dynamacité</i> : Minimiser la dynamicité	122
	Temps de sélection	122
	Taux d'erreurs	122
	Utilisabilité subjective	122
	Préférence des utilisateurs	123
6.2.5.2	Axe <i>Observabilité de l'expansion</i> : Expliciter l'expansion	123
	Décalage du point de chute des mouvements	123
	Taux d'erreurs	123
6.2.5.3	Axe <i>Élément augmenté</i>	124
6.2.6	Conclusion de l'expérience 2	124
	Conclusion	125

Introduction

Dans ce chapitre, nous présentons deux expériences. La première a pour objectif de mettre à l'épreuve nos hypothèses de conception de la technique VTE, présentée au chapitre précédent. La deuxième a pour objectif d'explorer expérimentalement plus avant les dimensions de notre espace de conception : 6 techniques sont comparées et nous étudions l'influence de l'aide visuelle sur les performances du geste de pointage. Nous présentons également notre contribution à l'étude expérimentale des techniques d'expansion de cibles avec la conception d'un nouvel agencement de distracteurs.

Les résultats de la première expérience valident les hypothèses de conception de VTE, fondées sur une étude approfondie du MIO [88], et nous apportent des recommandations sur les techniques de pointage à distance utilisées. Les résultats de la deuxième expérience sont analysés et discutés à la lumière des axes de conception définis au chapitre 4. Ils permettent de définir de nouvelles recommandations pour la conception des aides visuelles des techniques d'expansion de cibles et par là même permettent d'augmenter le pouvoir génératif de notre espace de conception.

Élément augmenté	Observabilité de l'expansion					
	Dynamacité					
	<i>statique</i>	<i>discret</i>	<i>continu</i>	<i>statique</i>	<i>discret</i>	<i>continu</i>
<i>curseur</i>	Impossible		Bubble Ray (Bubble Cursor)	Impossible		
<i>espace</i>				VTE		
<i>cible(s)</i>						

FIGURE 6.1 – Classification des techniques comparées par l'expérience 1 dans la représentation en deux dimensions de l'espace de conception défini au chapitre 4. Bubble Ray utilise la même aide visuelle que Bubble Cursor.

6.1 Expérience 1 : VTE

6.1.1 Introduction

L'objectif premier de cette expérience est de confronter nos hypothèses de conception de VTE [58] à l'étude expérimentale en comparant VTE à Bubble Ray [129] (figure 6.1). L'originalité de VTE réside dans les choix de conception de l'aide visuelle que cette technique utilise : une aide visuelle *explicite*, *statique* et *sur-espace*. Bubble Ray, présentée au chapitre 3 section 3.4, utilise une aide visuelle semblable à la technique Bubble Cursor, présentée au chapitre 3 section 3.2.4 : une aide visuelle *implicite*, *continue* et *sur-curseur* (figure 6.1). VTE et Bubble Ray utilisent en revanche le même algorithme d'expansion, la tessellation de Voronoï, ce qui nous permet de focaliser notre étude sur leurs aides visuelles.

Un objectif second de cette expérience est de comparer deux techniques de pointage à distance : le pointage main libre à l'aide du Xtion de Asus et le ray-casting à l'aide du Virtual Pointer, techniques développées par Aesculap et présentées au chapitre 2, respectivement section 2.3.2 et section 2.3.1. Nos hypothèses expérimentales sont donc les suivantes :

- H1 : VTE obtiendra de meilleures performances que Bubble Ray à la fois en vitesse de sélection et en taux d'erreurs, car son aide visuelle permet une meilleure optimisation du geste de pointage dès les premières phases du geste, comme expliqué en détail au chapitre précédent, section 5.1.1.
- H2 : VTE sera perçue plus utilisable et plus intuitive que Bubble Ray, car l'affichage statique de l'ensemble du diagramme devrait fournir aux utilisateurs un affichage plus stable que Bubble Ray et une meilleure sensation de contrôle, notamment au moment de la validation des sélections.
- H3 : la technique de ray-casting devrait obtenir de meilleures performances que la technique de pointage main libre car sa précision est meilleure, sa latence moindre (chapitre 2 section 2.3.2) et son utilisation engendre moins de fatigue physique

pour l'utilisateur. En effet, tandis que le pointage main libre nécessite un maintien et des déplacements de la main dans l'espace, le ray-casting nécessite seulement un maintien du dispositif et des mouvements du poignet.

6.1.2 Protocole expérimental

6.1.2.1 Participants et appareillage

Treize bénévoles de 25 à 67 ans (6 femmes et 7 hommes), dont 12 droitier(e)s, ont participé à l'expérience. Tous sont des utilisateurs quotidiens d'un ordinateur et tous utilisent leur main dominante pour contrôler le dispositif de pointage à distance. Deux d'entre-eux utilisent quotidiennement un dispositif de suivi des mouvements du corps (le Microsoft Kinect) pour des jeux vidéos.

Nous utilisons deux techniques de pointage à distance : le pointage main libre à l'aide du Xtion de Asus et le ray-casting à l'aide du Virtual Pointer, techniques présentées au chapitre 2, respectivement section 2.3.2 et section 2.3.1. Dans les deux cas, les participants valident leur sélection avec la pédale médicale d'Aesculap. La sélection est déclenchée au relâchement de la pédale afin de limiter la fatigue des participants causée par le maintien du pied en suspension.

La plate-forme logicielle d'Aesculap est responsable de la gestion des techniques de pointage à distance. Le programme développé pour l'expérience est en C++/Qt, exécuté sous Windows 7 sur un PC avec un processeur Quadcore cadencé à 2,3 GHz. L'affichage est assuré par un écran 20 pouces LCD à une résolution de 1280×1024 .

6.1.2.2 Détails des techniques implémentées

Nous avons reproduit la technique Bubble Ray, telle qu'elle a été présentée [53, 129] : l'algorithme qui régit la taille de la bulle est celui décrit dans l'étude Bubble Cursor [53]. La bulle s'étend pour englober la cible désignée (figure 6.2 a) : elle forme alors un contour de 6 pixels de large autour de la cible en question.

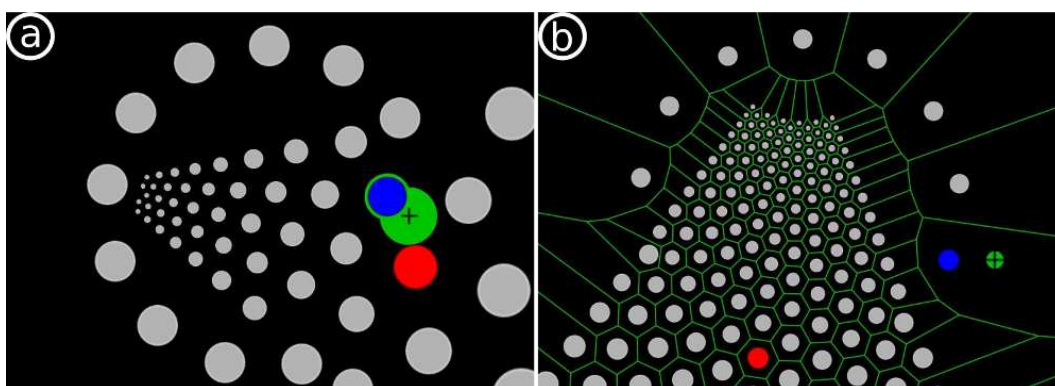


FIGURE 6.2 – Implémentation de (a) Bubble Ray et de (b) VTE.

L'aide visuelle de VTE représente les cellules de Voronoï avec des lignes vertes assez épaisses (2 pixels) pour être vues à une distance de deux mètres (figure 6.2 b).

6.1.2.3 Tâche et procédure

La tâche répétée est une sélection d'une cible en deux dimensions fortement inspirée du standard ISO 9241-9 [70] d'évaluation des techniques de pointage. La cible est un disque rouge, les distracteurs sont des disques gris et la cible désignée est un disque bleu (figure 6.2 et 6.3). Nous utilisons un fond noir. Le curseur est une petite croix noire entourée d'un petit disque vert (figure 6.2). Si un participant sélectionne un distracteur au lieu de la cible, le distracteur devient rouge pendant 100 ms. Si c'est la cible qui est sélectionnée, elle devient verte pendant 100 ms. Ainsi, les participants ont un retour visuel immédiat sur la validité de leur sélection : rouge signifie une erreur de sélection, vert signifie une sélection correcte. Après ces 100 ms, la tâche de pointage suivante est affichée.

Les participants sélectionnent successivement un ensemble de cibles placées sur un cercle, chaque cible "suivante" étant diamétralement opposée à sa précédente. Un cercle contient 13 cibles (figure 6.3). La première sélection constitue une *tâche de transition*, non comptabilisée dans les résultats et durant laquelle aucun distracteur n'est affiché (figure 6.3). Les 12 sélections de cibles suivantes sont enregistrées et constituent la mesure.

Pour ces 12 sélections, nous adaptons à notre protocole l'agencement de cibles proposé par Blanch et al. en 2011 [26]. L'algorithme de sa version deux dimensions, fourni par Renaud Blanch, définit en fonction de l'*ID* du geste, de l'*Amplitude* du geste et de la *Densité* de distracteurs la position relative des distracteurs : un ensemble de cercles concentriques de distracteurs de taille croissante en partant du centre. L'ensemble des distracteurs est ensuite centré sur la cible précédente, c'est-à-dire approximativement sur le point de départ du mouvement, et orienté précisément dans la direction du mouvement à effectuer (figure 6.3). Les distracteurs inférieurs à 5 pixels ou en dehors d'un angle de 60° centré sur la direction du mouvement à faire sont supprimés (figure 6.3). Deux distracteurs du cercle de ISO9241-9 sont également supprimés afin de conserver la géométrie de l'agencement des distracteurs (figure 6.3).

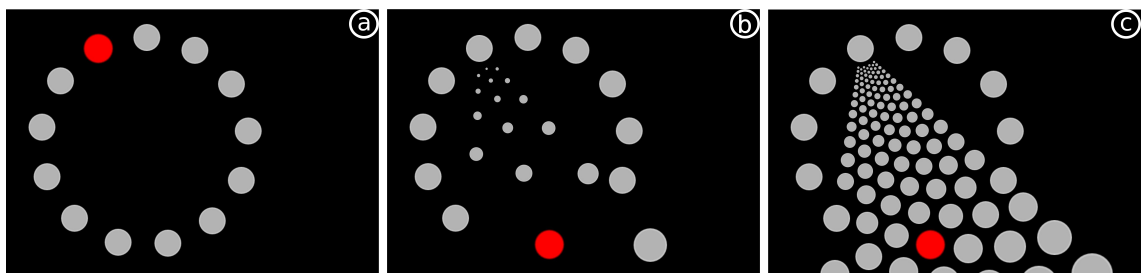


FIGURE 6.3 – Intégration de l'agencement de cibles de Blanch et al. [26] : (a) Les 13 cibles (tâche de transition). (b) Intégration des distracteurs de l'agencement de Blanch et al. avec une *Densité* = 0,1 et avec (c) avec une *Densité* = 0,6.

Grâce à la répétition dans la séquence des cibles et dans la position des distracteurs, les participants peuvent anticiper la position de chaque cible et chaque distracteur, conformément aux recommandations du standard ISO 9241-9 [70].

La sélection d'un distracteur est comptée comme une erreur. Le taux global d'erreurs par session est calculé et mis à jour à chaque sélection. Il est affiché pendant les tâches de transition avec un message indiquant si le participant doit ralentir ou accélérer pour respecter un objectif de 4 % d'erreurs. Les participants peuvent se reposer pendant les tâches de transition.

Enfin, entre deux sessions consécutives, les participants remplissent un questionnaire System Usability Scale (SUS) [14] (l'"échelle d'utilisabilité des systèmes") traduit en français à propos de la technique de pointage (*Dispositif* + *Technique*) qu'ils viennent d'utiliser. Ce questionnaire en 10 questions (annexe A), développé en 1986 par John Brooke, permet d'évaluer l'utilisabilité subjective d'un système informatique, c'est-à-dire la satisfaction globale de l'utilisateur. A la fin de l'expérience, les participants remplissent un questionnaire comparatif des 4 techniques de pointage de l'expérience (annexe B) et bénéficient de jus de fruits, de croissants et de confiture poire-gingembre *faite-maison*.

6.1.2.4 Conditions

Les variables indépendantes sont :

- le *Dispositif* de pointage à distance : *RayC* pour ray-casting ou *Main* pour pointage main libre,
- la *Technique* d'aide au pointage : *VTE* ou *Bubble Ray*,
- l'*Amplitude* du mouvement, c'est-à-dire la distance D à parcourir jusqu'à la cible dans l'équation de la loi de Fitts (chapitre 2, équation 2.1) : 600 ou 800 pixels,
- l'indice de difficulté ID de la tâche de pointage : 3, 3,5 ou 4,
- la *Densité* de distracteurs : 0,1, 0,3, 0,6 ou 0,9.

0,91 est la densité maximale possible avec cet agencement de distracteurs [26]. Nous utilisons la formule de Shannon (chapitre 2, équation 2.2) pour calculer la taille des cibles.

Nous utilisons un protocole intra-sujet en mesures répétées croisant totalement toutes les variables indépendantes :

- une session par $\text{Dispositif} \times \text{Technique}$ pour chaque participant, c'est-à-dire 4 sessions par participant, dont l'ordre est contrebalancé en carré latin.
- un bloc de 13 sélections (un cercle complet) par $\text{Amplitude} \times \text{ID} \times \text{Densité}$ par session, c'est-à-dire 24 blocs de sélections en ordre aléatoire par session.

Ainsi, nous avons enregistré 1152 sélections par participant. Un bloc de 13 sélections supplémentaire permet aux participants de se familiariser avec la nouvelle technique de pointage (*Dispositif* + *Technique*) au début de chaque session.

6.1.3 Résultats : Performances de pointage

6.1.3.1 Méthode

Sur 14976 sélections, nous avons retiré 40 anomalies évidentes. Dans cette section de résultats, nous utilisons le code suivant : * désigne une p-value inférieure à 0,05, ** désigne une p-value inférieure à 0,001. Afin d'éviter les effets de réplication, nous agrégeons les essais d'un même bloc (12 essais) en considérant leur moyenne géométrique, cette mesure étant la meilleure approximation du centre de la distribution [115]. Nous utilisons des tests t appariés avec une correction de Bonferroni [116, 139] pour comparer les temps de sélection et les taux d'erreurs entre les différentes conditions $Dispositif \times Technique$.

6.1.3.2 Calcul de d_E

Nous avons calculé un indice de difficulté effectif ID_E prenant en compte la taille étendue des cibles d_E et non la taille initiale des cibles d utilisée par l'indice de difficulté classique ID (chapitre 2, équation 2.2). Ainsi, nous respectons les recommandations du MIIO et des études précédentes sur les expansions de cibles [35, 83, 84, 142] quant à l'usage de la loi de Fitts pour les expansions de cibles.

Nous utilisons une approximation circulaire de la taille étendue de la cible (figure 6.4). Cela permet de considérer une seule variable, d_E le diamètre du cercle, comme la taille étendue de la cible, notamment dans les calculs utilisant la loi de Fitts. Comme l'agencement des distracteurs génère des cellules de Voronoï pseudo-hexagonales légèrement étirées dans la direction du mouvement à faire [26], la taille étendue d_E (segment $[AB]$ à la figure 6.4) est deux fois la distance du centre de la cible au bord de la cellule le plus proche (segment $[AO]$ à la figure 6.4), c'est-à-dire le bord entre la cible et le point de départ du mouvement. Le calcul de d_E a été possible grâce à la valeur r présentée dans les annexes de [26] pour décrire la génération de cet agencement de cibles.

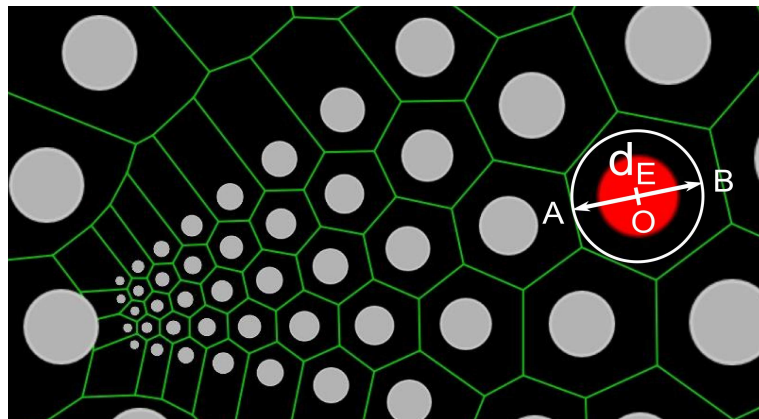


FIGURE 6.4 – Le sens du mouvement est de droite à gauche. Le cercle blanc représente la zone considérée comme la taille étendue de la cible.

6.1.3.3 Comparaison des dispositifs de pointage

Les résultats confirment l'hypothèse H3 : les temps de sélection et les taux d'erreurs reflètent les différences entre les dispositifs de pointage, à l'avantage du ray-casting avec le Virtual Pointer (figure 6.5) :

- le temps de sélection moyen global est de 2,35 s ($\sigma = 1,05s$) pour le pointage main libre avec le Asus Xtion et de 1,92 s ($\sigma = 0,764s$) pour le ray-casting avec le Virtual Pointer ($t = 28,52 *$).
- Le taux d'erreurs global est de 7,36 % pour le pointage main libre avec le Asus Xtion et de 4,30 % pour le ray-casting avec le Virtual Pointer ($t = 5,81 **$).

Appuyés par ces résultats, nous avons séparé la suite des résultats entre les deux dispositifs de pointage, en raison de leurs différences importantes en terme de précision, de latence et de fatigue induite (H3).

6.1.3.4 Performances pour le ray-casting

Le temps de sélection moyen global est de 1,87 s ($\sigma = 0,615s$) pour VTE et de 1,98 s ($\sigma = 0,885s$) pour Bubble Ray ($t = 6,528 *$), ce qui confirme H1 (figure 6.5 gauche). Une ANOVA en mesures répétées à 3 facteurs (*Technique* \times *Densité* \times *Amplitude*) sur le temps de sélection a montré des effets significatifs de la *Technique* ($F_{1,608} = 5,62 *$) et de la *Densité* ($F_{3,608} = 49,8 **$), mais aucun effet d'interaction significatif sur le temps de sélection. Ces effets reflètent la forte dépendance entre la *Densité* et l' ID_E de la tâche, ce dernier ayant une forte influence sur les temps de sélection.

Bien que ce soit une tâche de pointage à distance, nous avons pu modéliser précisément le temps de sélection comme une fonction linéaire de ID_E en utilisant la loi de Fitts à partir d'un modèle de régression (figure 6.6).

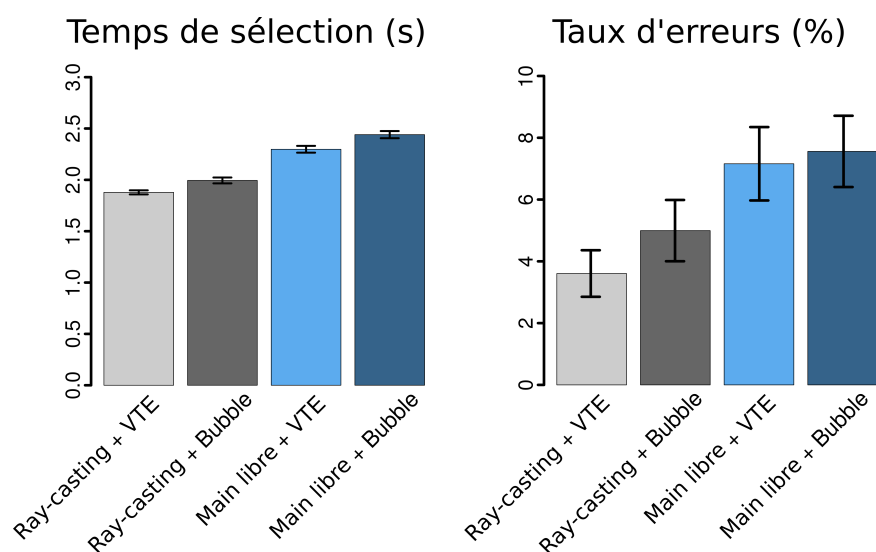


FIGURE 6.5 – Temps de sélection moyens et taux d'erreurs moyens par *Dispositif* \times *Technique*, avec les intervalles de confiance à 95 %.

	a	b	Adj. R²
RayC + VTE	.164	.566	.94
RayC + Bubble	.253	.575	.89
Main + VTE	-.425	.902	.94
Main + Bubble	-.200	.874	.97

FIGURE 6.6 – Coefficients de la loi de Fitts (a pour l'ordonnée à l'origine et b pour la pente en s/bit) pour chaque *Dispositif + Technique*.

Le taux d'erreurs est de 3,61 % pour VTE et 4,99 % pour Bubble Ray (figure 6.5 droite), cette différence étant significative ($t = 2,19^*$), cela confirme H1. Une ANOVA en mesures répétées à 3 facteurs (*Technique* \times *Densité* \times *Amplitude*) sur le taux d'erreurs par bloc d'essais a montré des effets significatifs de la *Technique* ($F_{1,608} = 5,38^*$), de l'*Amplitude* ($F_{1,608} = 4,21^*$) et de la *Densité* ($F_{3,608} = 25,1^{**}$) sur les taux d'erreurs.

6.1.3.5 Performances pour le pointage main libre

Le temps de sélection moyen est de 2,28 s ($\sigma = 1,02s$) pour VTE et de 2,42 s ($\sigma = 1,08s$) pour Bubble Ray ($t = 5,814^*$), ce qui confirme H1 (figure 6.5 gauche). Une ANOVA en mesures répétées à 3 facteurs (*Technique* \times *Densité* \times *Amplitude*) sur le temps de sélection a montré des effets significatifs de la *Technique* ($F_{1,608} = 5,90^*$), de l'*Amplitude* ($F_{1,608} = 6,02^*$) et de la *Densité* ($F_{3,608} = 66,6^{**}$), mais aucun effet d'interaction significatif sur le temps de sélection.

La modélisation du temps de sélection comme une fonction linéaire de ID_E en utilisant la loi de Fitts à partir d'un modèle de régression (figure 6.6) n'est pas optimale pour le dispositif main libre. En effet, l'apparition d'ordonnées à l'origine (associées au temps de réaction de l'utilisateur dans une tâche à ID nul) négatives est problématique pour interpréter ce modèle.

Les participants ont globalement respecté le taux d'erreurs objectif (4 % des sélections) avec le dispositif de ray-casting, mais ce n'est pas le cas avec le dispositif de pointage main libre (H3). Le taux d'erreurs est de 7,16 % pour VTE et 7,56 % pour Bubble Ray (figure 6.5, à droite), différence qui n'est pas significative ($t = 0,476, p = 0,634$). Une ANOVA en mesures répétées à 3 facteurs (*Technique* \times *Densité* \times *Amplitude*) sur le taux d'erreurs par bloc d'essais a montré des effets significatifs de l'*Amplitude* ($F_{1,608} = 22,2^{**}$), de la *Densité* ($F_{3,608} = 57,9^{**}$) et de l'interaction *Amplitude* \times *Densité* ($F_{3,608} = 4,40^*$) sur les taux d'erreurs, mais pas d'effet de la *Technique*.

Ce dernier effet de l'interaction *Amplitude* \times *Densité* sur les taux d'erreurs nous a permis de remarquer que la différence entre le taux d'erreurs en-dessous (4,93 %) et au-dessus (19,3 %) d'un seuil de ID_E de 3,5 est significative ($t = 10,9^{**}$). Ce seuil se retrouve dans les temps de sélection. La figure 6.7 présente l'indice de performance moyen IP comme une fonction de ID_E pour le pointage main libre avec le Asus Xtion : en-dessous du seuil de 3,5 qui correspond à une taille de cible étendue d_E d'environ 61,3 pixels, VTE surpasse significativement Bubble Ray. Au-dessus du seuil $ID_E = 3,5$, les deux techniques sont équivalentes.

Comme un tel seuil n'est pas observé pour les temps de sélection et les taux d'erreurs pour le dispositif de ray-casting, nous estimons que ce seuil montre la limite de précision de ce dispositif de pointage main libre avec le Asus Xtion Pro Live : une cible d'environ 60 pixels pour une sélection à 2 m de distance.

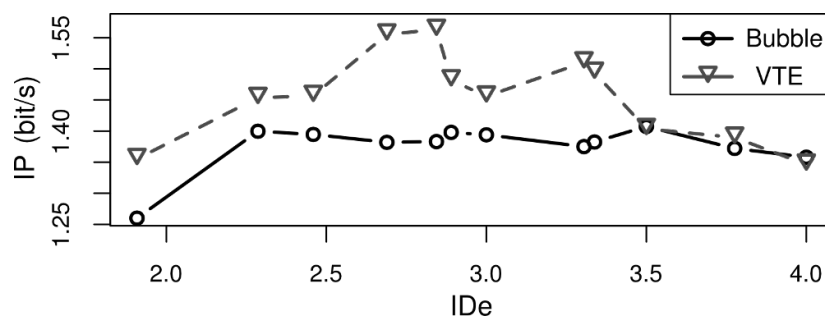


FIGURE 6.7 – Pointage main libre avec le Asus Xtion : Indice de performance moyen IP comme une fonction de l'indice de difficulté effectif ID_E : $IP = T/ID_E$, T étant le temps de sélection.

6.1.4 Résultats : Préférences des utilisateurs

Les questionnaires sur l'utilisabilité perçue par les participants des techniques de pointage *Dispositif* \times *Technique* confirment H2 et H3. Parmi 13 sujets, 6 préfèrent RayC+VTE, 3 préfèrent RayC+Bubble, 2 préfèrent Main+VTE et les 2 restant les trouvent équivalents. Nous avons demandé aux participants de choisir la technique la plus intuitive, indépendamment du dispositif de pointage à distance : 10 sujets désignent VTE comme la plus intuitive, 1 sujet choisi Bubble Ray et 2 ne se prononcent pas. Les raisons les plus communes pour le choix de VTE sont :

1. la visibilité et la stabilité des tailles étendues des cibles (4 participants)
2. VTE est moins perturbant que Bubble Ray (3 participants).

Enfin, les scores SUS moyens confirment également H2 et H3. RayC+VTE obtient un score moyen de 77,1, RayC+Bubble de 74,6, Main+VTE de 71,5 et Main+Bubble de 66,5, ce qui est un score faible.

6.1.5 Discussion

Les résultats de l'expérience confirment les hypothèses de conception de VTE et l'impact de l'aide visuelle sur les performances d'une technique d'expansion de cibles. De plus, les résultats évoquent également une préférence marquée des utilisateurs pour l'aide visuelle *statique*. Ces résultats confirment le rationnel de conception de VTE, détaillé au chapitre 5 :

1. *Le sous-mouvement balistique* : En voyant la taille étendue de la cible, l'utilisateur peut planifier un mouvement plus opportuniste car adapté à la taille de la cible dans l'espace moteur. Ainsi, l'impulsion nerveuse qui émane du système central neuromoteur programme dès le départ une rigidité du bras adaptée à la taille étendue de la cible dans les directions normale et tangentielle [130]. De plus, la confiance dans le succès de ce sous-mouvement est accrue. Le système neuromoteur humain trouve toujours le meilleur compromis vitesse/précision [130] et VTE offre un moyen d'exploiter cette capacité en fournissant toute l'information pertinente sur la cible dès la phase préliminaire du geste.
2. *La phase corrective* : Grâce à une connaissance a priori de la taille étendue des cibles, les utilisateurs peuvent optimiser leurs sous-mouvements correctifs. Nous pensons qu'ils peuvent exploiter consciemment et en toute confiance les bords et les coins de chaque cellule de Voronoï, ce qui s'avérerait probablement plus difficile dans le cas d'une aide visuelle *implicite* comme celle de Bubble Ray, qui ne représente pas les cellules de Voronoï [133]. Enfin, nous pensons que VTE offre aux utilisateurs un meilleur sentiment de contrôle, un point important pour l'utilisabilité de la technique.
3. *Les raisons des préférences* : Les scores SUS et les commentaires libres des participants suggèrent que VTE est plus simple à utiliser que Bubble Ray. En effet, les cellules de Voronoï sont des formes simples et régulières, faciles à percevoir et à comprendre. Chaque point de l'espace est contenu dans un polygone qui entoure la cible la plus proche. L'utilisateur peut aisément comprendre la propriété *où que je clique, la cible sélectionnée est la plus proche du curseur*, car deux cibles A et B voisines partagent un bord de cellule qui est sur la médiatrice du segment $[AB]$. De plus, avec VTE les utilisateurs perçoivent la taille étendue de toutes les cibles simultanément. Comme l'oubli est un mécanisme nécessaire à la cognition humaine, le système cognitif va filtrer l'information dont il a besoin et se focaliser sur la cible et sa cellule. Le reste est oublié. Ainsi, avec une représentation statique de l'information critique et spatiale, nous concentrons la charge de travail visuelle au début du geste et nous évitons d'avoir à gérer l'état de la bulle de Bubble Ray ensuite. Cette transformation se reflète dans les réponses des utilisateurs : VTE leur semble plus intuitive et moins perturbante que Bubble Ray.

6.1.6 Conclusion de l'expérience 1

Cette première expérience nous a permis de confirmer le bien-fondé des hypothèses de conception de VTE et son potentiel en terme de performances de pointage et de satisfaction des utilisateurs. En effet, la comparaison avec Bubble Ray, qui implémente Bubble Cursor pour le pointage à distance, est sans équivoque, tant au niveau des temps de sélection et des taux d'erreurs que des préférences utilisateurs et de l'utilisabilité perçue. Les résultats qualitatifs nous amènent à penser qu'utiliser VTE est plus confortable. L'aide visuelle *explicite*, *statique* et *sur-espace* offre donc des possibilités intéressantes et nous évaluerons dans l'expérience suivante des variantes de VTE. Ces résultats doivent cependant être modérés en raison de la disposition choisie des distracteurs, qui résulte en des cellules de Voronoï hexagonales aux tailles assez régulières. Il serait intéressant de tester par la suite des configurations moins favorables. Les résultats montrent également que le dispositif de ray-casting Virtual Pointer offre une meilleure utilisabilité que le dispositif de pointage main libre avec le Asus Xtion, tant en terme d'efficacité que de satisfaction des utilisateurs. Nous constatons par ailleurs que les dispositifs basés sur le suivi de la main sont plus utilisés pour des langages gestuels que pour le pointage [44, 51, 98]. Nous n'utiliserons donc dans l'expérience suivante que le Virtual Pointer.

Élément augmenté	Observabilité de l'expansion					
	Dynamicité					
		<i>implicite</i>		<i>explicite</i>		
	<i>statique</i>	<i>discret</i>	<i>continu</i>	<i>statique</i>	<i>discret</i>	<i>continu</i>
<i>curseur</i>	Impossible		Bubble Cursor, Rope Cursor	Impossible		
<i>espace</i>				VTE, MTE		eVTE
<i>cible(s)</i>		TARGET			Cell Painting	

FIGURE 6.8 – Classification des techniques comparées par l'expérience 2 dans la représentation en deux dimensions de l'espace de conception défini au chapitre 4. Les flèches représentent les techniques combinées, évaluées également.

6.2 Expérience 2 : Axes de conception

6.2.1 Introduction

Cette expérience a plusieurs objectifs, dont le premier est d'évaluer 6 techniques conçues par exploration analytique de l'espace de conception (chapitre 5) : Rope Cursor, eVTE, MTE, VTEtoROPE, VTEtoCELL et VTEtoTARGET. Expansion Lens, en revanche, est évaluée au cours d'une troisième expérience présentée au chapitre 8. Les techniques évaluées sont comparées à VTE, à Bubble Cursor (technique équivalente à Bubble Ray de l'expérience précédente) et à TARGET, technique *implicite*, *discrète* et *sur-cible* qui a montré de bonnes performances [126, 131], notamment lors de l'expérience de comparaison des variantes de VTE en phase de conception (chapitre 5). Ainsi, l'expérience compare au moins une technique de chaque valeur de chaque axe de l'espace de conception (figure 6.8), c'est-à-dire des techniques *sur-espace*, *sur-cible* et *sur-curseur*, des techniques *explicites* et *implicites* et des techniques de dynamique variable. Elle compare également une technique de chaque ligne et de chaque colonne de la représentation en deux dimensions de l'espace de conception (figure 6.8). Seule la colonne correspondant à la combinaison *implicite* et *statique* n'est pas représentée car nous n'en connaissons aucun représentant à ce jour. En effet, une aide visuelle n'apportant d'information spatiale ni par sa forme *explicite* directement, ni par son mouvement comme beaucoup de techniques *implicites*, devrait alors proposer un autre moyen d'aider l'utilisateur.

Nous justifions donc notre deuxième objectif : concevoir des recommandations pour la conception d'aides visuelles pour les expansions de cibles en se basant sur une analyse des résultats de l'expérience selon nos axes de conception. Toutes ces techniques, excepté MTE, utilisant le même algorithme d'expansion (Voronoi), le focus est sur l'impact des aides visuelles sur les performances de pointage et l'utilisabilité des techniques d'expansion de cibles.

Le troisième objectif consiste à analyser plus en profondeur les différences d'optimisation des gestes de pointage en fonction des aides visuelles. En effet, la première expérience semble confirmer que VTE permet une meilleure optimisation du geste grâce à la présence permanente du diagramme de Voronoi. Dans cette expérience, les techniques combinées VTEtoROPE, VTEtoCELL et VTEtoTARGET n'affichent les cellules de Voronoi qu'au début du geste de pointage. Or, cela suffit-il à ce que l'utilisateur prenne bien en compte la cellule de Voronoi et pas seulement la cible ? Pour explorer cette question, une nouvelle mesure inspirée de précédents travaux [55] est mise en place, et un nouvel agencement des distracteurs permettant de contrôler la géométrie des cellules de Voronoi est conçu. Cette nouvelle mesure permet de savoir si l'utilisateur a visé la cible ou bien sa taille étendue.

Nos hypothèses répondent à nos objectifs : H1 et H3 concernent le deuxième objectif, tandis que H2 intéresse le troisième :

- H1 : À l'intérieur d'une même cellule de Voronoï, lorsqu'une aide visuelle change constamment alors que la cible désignée ne change pas, elle produit du bruit et de la redondance, selon la théorie de l'information de Shannon [121]. Ces stimuli inutiles ralentiront la tâche de sélection. Les aides visuelles les plus dynamiques seront donc moins rapides que les aides visuelles moins dynamiques.
- H2 : Les techniques atomiques ou combinées utilisant une aide visuelle *explicite* dès la phase préliminaire seront plus rapides que les autres car les utilisateurs optimiseront leur sous-mouvement balistique en considérant la taille étendue des cibles.
- H3 : Les techniques atomiques et combinées utilisant une aide visuelle *explicite* à la fin du mouvement obtiendront de moindres taux d'erreurs que les techniques utilisant une aide visuelle *implicite*. En effet, en ayant la possibilité d'évaluer les distances entre le curseur et les bords de chaque cible étendue, les utilisateurs disposeront de moyens d'anticipation des erreurs plus efficaces.

6.2.2 Agencement de cibles pour des tessellations non-centroïdales

Afin de savoir si l'utilisateur a visé la cible ou bien sa taille étendue, nous concevons un nouvel agencement de cibles qui permet de moduler la géométrie des cellules de Voronoï et ainsi de mesurer le décalage du point de chute des mouvements de pointage qu'induisent les aides visuelles.

6.2.2.1 Décalage du point de chute des mouvements

Selon le MIIO [88] et comme le suggèrent les résultats de l'expérience précédente (section 6.1.3), les utilisateurs optimisent leurs gestes de pointage en fonction de la taille qu'ils perçoivent de la cible qu'ils visent. En conséquence, le centroïde de la cible (son centre de masse) est une bonne prédiction du point de chute du mouvement : Grossman et al. l'ont démontré [55] avec des cibles de forme arbitraire. En comparant la distance moyenne du point de chute des mouvements au centre de masse de la cible ($d_{toTarget}$) avec la distance moyenne du point de chute des mouvements au centre de masse de la taille *étendue* de la cible (d_{toCell}), c'est-à-dire au centre de masse de la cellule de Voronoï, nous définissons trois cas :

- si ($d_{toCell} < d_{toTarget}$), c'est une aide visuelle *centrée sur la cellule*.
- si ($d_{toCell} > d_{toTarget}$), c'est une aide visuelle *centrée sur la cible*.
- si ($d_{toCell} \approx d_{toTarget}$), c'est une aide visuelle *mixte*, car aucune tendance ne se dégage.

Le fait qu'une aide visuelle est *centrée sur la cellule* est une preuve expérimentale de la prise en compte de la taille étendue des cibles par les utilisateurs au cours du geste. Une aide visuelle *explicite* doit provoquer de tels décalages du point de chute : du centre de masse de la cible vers le centre de masse de la cellule, comme le suggère l'hypothèse H2.

Néanmoins le timing de l'aide visuelle est également primordial : il détermine le moment où les utilisateurs perçoivent les tailles étendues et donc le temps restant pour optimiser leur mouvement.

6.2.2.2 Agencement des cibles

Pour le cas présent d'expansions de cibles utilisant Voronoï, les agencements de cibles existants rendent difficilement observables les décalages des points de chute et nos trois cas (centré sur la cible, sur la cellule ou mixte). Souvent [34, 53, 90, 126, 129], la cellule de Voronoï de la cible est carrée : son centre de masse est donc confondu avec celui de la cible, ce qu'on appelle mathématiquement une tessellation de Voronoï *centroïdale* [41]. L'agencement géométrique décrit par Blanch et al. [26] que nous avons utilisé lors de la première expérience [58] crée des cellules de Voronoï pseudo-hexagonales, dont le centre de masse est très proche de celui de la cible. Enfin, dans les études de Bateman et al. [15], le décalage des centres de masse entre cellules et cibles suit toujours une unique direction radiale. Nous avons donc créé un nouvel agencement de cibles, qui permet de contrôler l'ampleur et la direction des décalages entre centres de masse de cellule et de cible. Ce nouvel agencement de cibles est conçu pour obtenir des cibles qui ne sont pas au centre de leur cellule de Voronoï. Il est entièrement reproductible et se base sur la tâche standard ISO 9241-9 [15, 70]. Sur la figure 6.9, les cibles placées par ce standard sont celles dont le centre de masse de la cellule est matérialisé par un petit carré blanc. Elles sont placées sur un cercle virtuel et deviennent chacune leur tour *la* cible à sélectionner.

Pour décaler le centre de masse des cellules de Voronoï par rapport aux cibles, c'est-à-dire par rapport aux "sites" dans la terminologie de Voronoï (chapitre 3 section 3.2.1.1), la méthode est en deux étapes : créer des cellules de Voronoï trapézoïdales puis décaler la cible et le centre de la cellule de Voronoï. Tout d'abord, un distracteur (facultatif) est ajouté au centre du cercle des cibles. La géométrie des cellules de Voronoï dépend ensuite du placement des autres sites autour de chaque cible. Nous définissons donc trois contraintes pour placer les distracteurs :

1. Les distracteurs sont placés sur des cercles concentriques au cercle ISO des cibles.
2. Chaque cercle concentrique contient le même nombre de distracteurs.
3. Les distracteurs de différents cercles sont alignés le long des rayons des cercles.

Ces contraintes définissent des cellules de Voronoï trapézoïdales (figure 6.9) : ceci simplifie le calcul de l'aire des cellules et d'autres grandeurs associées (hauteur du trapèze, petite et grande base notamment) et permet de garder une forme de cellule bien maîtrisée. Mais cela ne suffit pas en revanche à assurer que les cibles ne sont pas au centre de leur cellule.

Dans un diagramme de Voronoï, l'arête qui sépare deux cibles voisines A et B (le bord commun de leur cellule) est un segment de la médiatrice du segment $[AB]$. Si la distribution des cibles est régulière (rayons des cercles périodiques, égalité des angles entre les rayons contenant les cibles), chacune se retrouvera approximativement au centre de sa

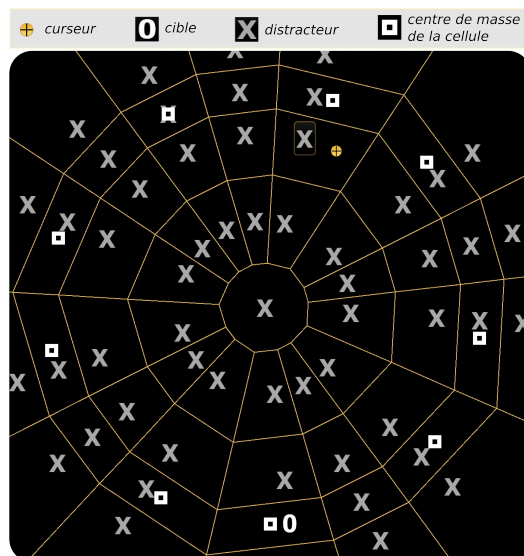


FIGURE 6.9 – Nouvel agencement de cibles. Les décalages entre centre de la cible et centre de masse de la cellule sont ici seulement tangentiels (pas de décalage radial) et dans les deux sens (horaire et anti-horaire).

cellule. C'est ce que l'on peut observer sur certaines cellules trapézoïdales de la figure 6.9 et dans nombre de travaux [34, 53, 90, 126, 129] dans lesquels la cible est entourée de 4 distracteurs, tous à la même distance d'elle, et se retrouve donc au centre d'une cellule de Voronoï carrée. Pour créer des cellules de Voronoï non-centroïdales, il faut donc introduire de l'irrégularité dans le placement des distracteurs. C'est pourquoi nous plaçons ensuite les distracteurs :

- sur des cercles avec des diamètres apériodiques et/ou
- sur des rayons qui sont dans des directions apériodiques.

Cela crée un décalage du centre de masse des cellules par rapport aux cibles suivant respectivement une *direction radiale* et/ou une *direction tangentielle*. Comme la tâche ISO 9241-9 place les cibles dans des directions périodiques, nous ajoutons les distracteurs sur les rayons intermédiaires une fois sur deux, soit sur la bissectrice d'une part du cercle ISO sur deux (figure 6.9).

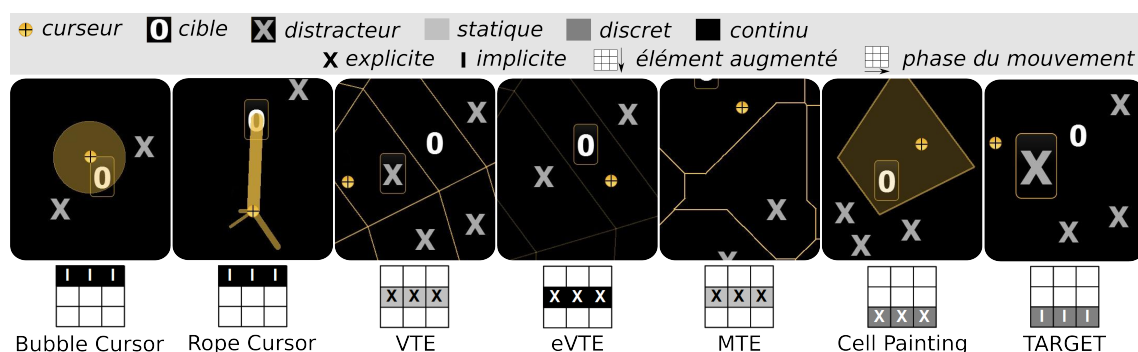


FIGURE 6.10 – Aides visuelles atomiques telles que développées pour l'expérience. Pour chaque aide visuelle, capture d'écran et notation matricielle.

6.2.3 Protocole expérimental

6.2.3.1 Participants et appareillage

Dix-huit bénévoles droitiers (6 femmes et 12 hommes) âgés de 22 à 51 ans ont participé à l'expérience. Ils sélectionnent les cibles à 1,8 m de l'écran, debout devant une table, avec le Virtual Pointer, technique de ray-casting d'Aesculap. Ils valident les sélections avec la pédale médicale fournie par Aesculap. L'application développée pour l'expérience tourne sous Windows 7, sur un PC Quadcore cadencé à 2,3 GHz. L'affichage est un écran LCD de 24" avec une résolution de 1920×1200 .

6.2.3.2 Détail des techniques implémentées

La figure 6.10 illustre l'ensemble des techniques développées pour l'expérience :

- Le comportement de la bulle de Bubble Cursor est reproduit à l'identique de l'original [53] et de l'expérience 1. La bulle en revanche, ne s'étend pas à entourer totalement les cibles de la même façon. Cette fonction de contour est réalisée par une mise en valeur de la cible décrite ci-dessous.
- L'épaisseur de la corde principale de Rope Cursor est de 24 pixels, celle des mini-cordes de 12, 6 et 3 pixels.
- La transparence du diagramme de eVTE varie de l'opacité totale à la transparence totale quand la vitesse du curseur varie de 0 à 1500 pixels/s, suivant une fonction sigmoïde de la vitesse du curseur. Une analyse des mouvements de pointage réalisée avec le Virtual Pointer a permis d'ajuster ces valeurs au cours de tests préliminaires.
- TARGET agrandit la cible d'un facteur de 1,618.
- Pour les aides visuelles combinées, VTetoROPE, VTetoCELL et VTetoTARGET, le seuil de vitesse du curseur déclenchant le changement d'aide visuelle est fixé à 500 pixels/s, valeur empirique déterminée lors de pré-tests. Le début du sous-mouvement balistique est ainsi détecté. De précédentes études [34, 131] ont évalué ce même seuil de 100 à 150 pixels/s. Leurs conditions expérimentales étaient cependant très différentes de celles-ci : configuration de bureau, dispositifs de pointage différents.

Avant d'appliquer ce seuil, la vitesse du curseur est filtrée : la vitesse prise en compte est la moyenne des 4 dernières vitesses instantanées. En effet, les tests préliminaires ont révélé :

- des pics de vitesse induits par effet Heisenberg lors de la validation de la sélection par un appui du pied,
- un bruit important dans le profil de vitesse instantanée dû aux tremblements naturels de la main amplifiés par la distance entre l'utilisateur et l'écran.

Le filtrage de la vitesse du curseur permet de lisser ces effets parasites et de révéler clairement la phase balistique de chaque geste.

6.2.3.3 Tâche et procédure

La tâche est une tâche de pointage standard ISO 9241-9 avec 9 cibles. La première est la cible de départ et les 8 autres constituent la mesure. La plate-forme logicielle C++ d'Aesculap gère le Virtual Pointer et définit le style *black-cockpit* (figures 6.9 et 6.10).

Les cibles affichent un zéro blanc, et les distracteurs un *X* gris. Une mince bordure jaune permet de reconnaître la cible désignée à tout moment (figure 6.10). Ce mécanisme de mise en valeur de la cible désignée est une aide visuelle minimale *implicite, discrète et sur-cible* (comme TARGET). Présente quelque soit la technique en cours de mesure, elle assure une bonne compréhension de l'interface graphique. Un son court confirme une sélection correcte. Quand un participant sélectionne un distracteur, cela provoque le remplacement de l'affichage de sa lettre *X* par celui du nombre d'erreurs de l'essai en cours pendant 100 ms. Les temps de sélection, les trajectoires du curseur, les profils de vitesse du curseur, les erreurs de sélection et les points de chute des mouvements sont recueillis et constituent les mesures. Les essais manqués sont retirés des mesures pour les temps de sélection.

La consigne aux participants est de terminer chaque session le plus vite possible. Avant de passer à la technique suivante, ils remplissent un questionnaire SUS [14] sur l'utilisabilité de la technique récemment expérimentée. Afin de les aider à se projeter mentalement en situation d'utilisation réelle de la technique, un contexte concret d'interaction à distance avec une télévision est proposé à leur imagination. Enfin, les participants classent les trois techniques qu'ils ont préférées à l'aide d'un rappel sur papier des techniques par des captures d'écran de chacune (annexe C). L'ordre des techniques est contre-balançé entre les participants en carré latin. L'expérience dure approximativement 40 minutes et commence avec une session de familiarisation au Virtual Pointer et à l'interface graphique de 5 minutes. Un verre de jus de fruit et une viennoiserie sont offerts aux participants à la fin de l'expérience.

6.2.3.4 Conditions

C'est un protocole intra-sujet de 9 techniques $\times (2 \times 2 \times 2)$ agencements de cibles utilisant l'agencement de cibles décrit ci-dessus et spécialement développé pour l'expérience. Deux cercles de distracteurs ont respectivement 400 et 1200 pixels de diamètre. Le diamètre du cercle de cibles varie : 600 ou 1000 pixels. Pour chacun de ces diamètres, un cercle additionnel de distracteurs de 800 pixels de diamètre est ajouté *ou non*, afin de gérer le décalage radial des cibles avec le centre de masse des cellules. Enfin, la dernière condition a trait au décalage des centre de masse des cellules des cibles sur la direction tangentielle : des distracteurs sont ajoutés *ou non* entre les cibles *une fois sur deux* (comme sur la figure 6.9) sur la bissectrice de l'angle entre deux cibles voisines.

La figure 6.9 illustre une de ces conditions : le cercle des cibles est de 1000 pixels de diamètre, le cercle additionnel de distracteurs est présent et les distracteurs permettant

le décalage tangentiel sont présents. Ainsi, les cibles présentent un décalage tangentiel, mais pas de décalage radial avec le centre de masse de leur cellule.

$2 \times 2 \times 2$ agencements de cibles sont ainsi présentés en ordre aléatoire à chaque session. 64 sélections (8 agencements \times 8 sélections) par technique et par participant sont enregistrées, pour un total de 10368 sélections (9 techniques \times 18 participants \times 64 sélections).

6.2.4 Résultats










6.2.4.1 Résultats quantitatifs

Méthode 149 sélection anormales (1,44 % des données) dues à des doubles clics ou à des difficultés techniques relatives au Virtual Pointer sont retirées de l'analyse. Dans cette section, nous utilisons la nomenclature suivante pour traduire la significativité des tests statistiques : * signifie $p < 0,05$; ** signifie $p < 0,01$; *** signifie $p < 0,0001$.

Pour comparer $d_{toTarget}$ et d_{toCell} pour chaque technique, nous avons considéré l'hypothèse H_0 : « $d_{toTarget} = d_{toCell}$ ». Ne pouvant pas assumer la normalité des données (tests de Shapiro-Wilk ***), nous choisissons parmi les tests statistiques non-paramétriques le test de Wilcoxon, les données étant appariées. Si ce test n'est pas significatif, H_0 n'est pas rejetée et le technique est classée comme « mixte ». Si H_0 est rejetée pour une technique, celle-ci sera classée centrée-cellule ou centrée-cible selon le signe de la différence observée entre $d_{toTarget}$ et d_{toCell} . La significativité des tests de Wilcoxon est reportée à la figure 6.11, colonne « Classification ». Les essais sont agrégés par bloc de 8 essais. Une ANOVA en mesures répétées à un facteur révèle un effet significatif de la technique sur le temps de sélection ($F_{8,1287} = 8,58$ ***). Les temps de sélection et les taux d'erreurs obtenus sont en conséquence comparés entre les techniques grâce à des tests t appariés utilisant la méthode de Bonferroni.

Décalage du point de chute des mouvements Comme le montre la figure 6.11 troisième colonne, MTE, VTE, VTEtoTARGET, VTEtoCELL, eVTE et VTEtoROPE sont mesurées *centrées sur la cellule*. Bubble Cursor et TARGET sont des techniques *centrées sur la cible* tandis que Rope Cursor est *mixte*.

Temps de sélection Les techniques sont ordonnées par ordre croissant de temps moyen de sélection : à la figure 6.11 de haut en bas du tableau et à la figure 6.12 (gauche) de gauche à droite. MTE est la technique la plus performante, plus rapide que Rope Cursor *, Bubble Cursor **, VTEtoCELL **, VTEtoTARGET **, eVTE *** et VTEtoROPE ***. En seconde position, VTE est plus rapide que VTEtoTARGET *, Bubble Cursor **, VTEtoCELL **, eVTE *** et VTEtoROPE **. En dernière position, VTEtoROPE est moins rapide que TARGET ** et Rope Cursor *.

Technique	Représentation	Classification	Temps de Sélection	Taux d'erreurs
MTE		centré/Cellule ***	1308 ms	5,13%
VTE		centré/Cellule ***	1310 ms	6,34%
TARGET		centré/Cible ***	1394 ms	8,09%
Rope Cursor		Mixte	1424 ms	6,03%
VTEtoTARGET		centré/Cellule ***	1431 ms	6,78%
Bubble Cursor		centré/Cible **	1440 ms	7,41%
VTEtoCELL		centré/Cellule ***	1456 ms	5,93%
eVTE		centré/Cellule ***	1477 ms	5,40%
VTEtoROPE		centré/Cellule **	1527 ms	7,61%








 explicite
  implicite
  statique
  discret
  continu
  élément augmenté
  phase du mouvement

FIGURE 6.11 – Pour chaque technique comparée : (1) sa notation matricielle (2) sa classification en fonction des points de chute des mouvements (3) son temps de sélection moyen (4) son taux d'erreurs moyen.

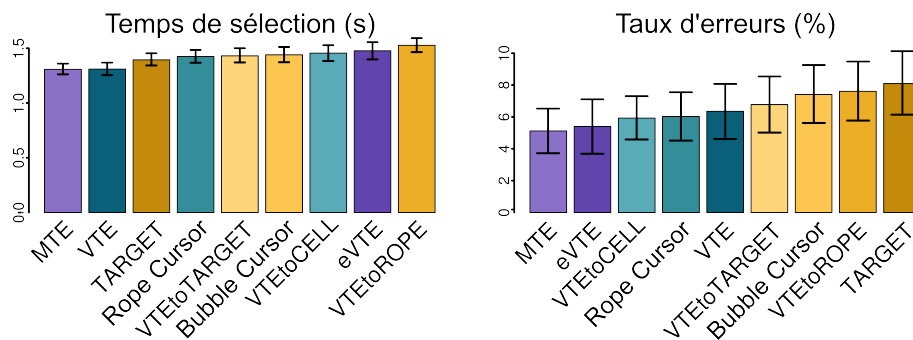


FIGURE 6.12 – À gauche, les temps de sélection moyens, à droite, les taux d'erreurs par technique.

Taux d'erreurs À la figure 6.12 à droite, les techniques sont ordonnées de gauche à droite en ordre croissant de taux d'erreurs. Les tests statistiques effectués n'ont révélé aucune différence significative entre les techniques.

6.2.4.2 Scores SUS et résultats qualitatifs

A la figure 6.13 à gauche, les techniques sont ordonnées de gauche à droite en ordre décroissant de score SUS moyen. VTE et TARGET sont perçues comme les techniques les plus utilisables, avec un score SUS moyen au-dessus de 84. Bubble Cursor et VTEtoROPE sont perçues au contraire comme les techniques les moins utilisables avec un score SUS moyen d'environ 72.

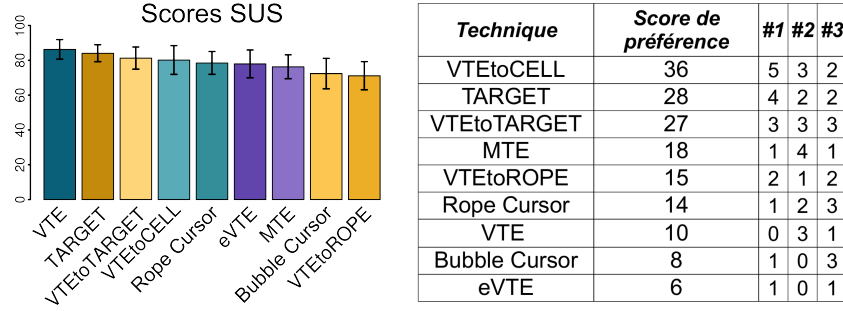


FIGURE 6.13 – A gauche, les scores SUS moyens avec les intervalles de confiance à 95 %. A droite, les scores de préférence et le nombre de participants ayant classé chaque technique 1^{ère}, 2^{nde} ou 3^{ème}.

Nous utilisons une approximation pour traiter les préférences exprimées par les participants : un score de préférence est calculé pour chaque technique sur la base des données qualitatives collectées. Ils sont classés en ordre décroissant du sommet au pied du tableau à droite de la figure 6.13. Le score de préférence (S) d'une technique s'exprime de la façon suivante :

$$S = 5 \times 1^{\text{ère}} + 3 \times 2^{\text{nde}} + 3^{\text{ème}} \quad (6.1)$$

où 1^{ère}, 2^{nde} et 3^{ème} sont le nombre de participants ayant classé la technique respectivement première, seconde ou troisième selon sa préférence. Les résultats sont similaires et le classement est identique en utilisant des coefficients plus faibles (respectivement 3, 2 et 1) dans la formulation 6.1 ci-dessus. Les participants ont préféré VTEtoCELL, puis TARGET et VTEtoTARGET. En revanche, VTE, Bubble Cursor et eVTE ont obtenu les scores de préférence les plus bas.

6.2.5 Discussion

MTE est la technique la moins propice aux erreurs et la plus rapide. La tessellation de Voronoï en distance de Manhattan semble donc un algorithme d'expansion prometteur. En distance de Manhattan, les arêtes du diagramme de Voronoï ne sont orientées que suivant 3 directions différentes, ce qui génère plus de motifs symétriques et périodiques qu'en distance euclidienne. Or, d'après les principes de la théorie gestaltiste [122], une plus grande symétrie axiale améliore la conception des interfaces. De plus, Oliva et al. [101] ont étudié la signification de la *complexité visuelle* pour les observateurs de scènes d'intérieur complexes. Ils ont ainsi montré que les observateurs trient les scènes par complexité visuelle en s'appuyant sur le nombre d'objets, les couleurs et les détails, l'espace libre et le désordre mais également sur la symétrie et l'organisation de l'information. Cette particularité qu'a MTE de proposer un rendu visuel d'une grande symétrie peut être à l'origine de ses bonnes performances. Cependant, il existe aussi une grande symétrie avec VTE. Une autre origine des bonnes performances de MTE peut résider dans la

forte présence de lignes horizontales et verticales, qui sont majoritaires dans les interfaces graphiques du quotidien.

Par rapport aux autres techniques, MTE souffre dans cette expérience d'un défaut d'effet d'apprentissage : c'est la seule technique qui utilise cet algorithme d'expansion, tandis que les autres partagent la tessellation de Voronoï en distance euclidienne. Avec les autres techniques, les utilisateurs apprennent donc d'une technique à l'autre à utiliser la tessellation en distance euclidienne, tandis qu'avec MTE, l'apprentissage est réduit à cette seule technique. Cette exception explique le score SUS passable de MTE. Les autres résultats observés sont exposés selon les axes de l'espace de conception.

6.2.5.1 Axe *Dynamacité* : Minimiser la dynamacité

Temps de sélection Les résultats de l'expérience montrent que LE TEMPS DE SÉLECTION CROÎT AVEC LA DYNAMACITÉ DE L'AIDE VISUELLE (H1). En effet, les trois techniques les moins dynamiques sont les plus rapides : MTE et VTE, les techniques *statiques*, sont les plus rapides et TARGET, *implicite* et *discrète*, est plus rapide que toutes les techniques utilisant une aide visuelle *continue*, qu'elle soit *implicite* ou *explicite* (les différences entre TARGET et Rope Cursor, VTEtoTARGET, VTEtoCELL et Bubble Cursor ne sont cependant pas significatives). De plus, eVTE et VTEtoCELL, les deux techniques *explicites* et dynamiques, sont significativement plus lentes que VTE et MTE, les deux techniques *explicites* et *statiques*.

Taux d'erreurs eVTE est moins propice aux erreurs que VTE. Une explication est que les aides visuelles *continues* fournissent constamment de l'information nouvelle et attirent le regard de l'utilisateur. D'un côté, la charge cognitive supplémentaire qui en résulte rend la technique plus lente. D'un autre côté, nous pensons que CANALISER L'ATTENTION DE L'UTILISATEUR PEUT RENDRE LA TECHNIQUE PLUS PRÉCISE SI L'INFORMATION UTILE EST FOURNIE AU BON MOMENT DU MOUVEMENT DE POINTAGE, c'est-à-dire durant la phase corrective pour eVTE.

Utilisabilité subjective Faisons un pas de plus dans la distinction statique/dynamique et définissons le *degré de dynamacité* d'une aide visuelle comme la *quantité de pixels modifiés* par l'aide visuelle (ou comme sa dérivée temporelle, un *flux de pixels modifiés*). Utilisons cette mesure pour classer les techniques en ordre croissant de degré de dynamacité : VTE, MTE, TARGET, VTEtoTARGET, VTEtoCELL, puis les techniques n'utilisant que des aides visuelles *continues*. Ainsi, excepté pour le cas de MTE expliqué précédemment, il existe une correspondance directe entre les scores SUS et le degré de dynamacité. Une explication est que LA DYNAMACITÉ D'UNE AIDE VISUELLE DIMINUE SON UTILISABILITÉ SUBJECTIVE CAR ELLE ACCROÎT LA CHARGE COGNITIVE DE L'UTILISATEUR. Cette charge cognitive additionnelle est déterminante lors de l'utilisation d'un dispositif de ray-casting à 5 DDL, qui occasionne déjà une charge cognitive significative.

Préférence des utilisateurs Les utilisateurs ont une préférence marquée pour les aides visuelles *discrètes*. VTEtoCELL, VTEtoTARGET et TARGET, les trois seules techniques utilisant une aide visuelle *discrète*, ont obtenu les trois meilleurs scores de préférence. Ce constat est en accord avec les résultats de précédents travaux [126, 131], qui montrent qu'une simple mise en valeur de la cible (aide visuelle *discrète*) permet de meilleures performances qu'une aide visuelle *continue* et *sur-curseur*. Ces résultats précédents, essentiellement quantitatifs, sont ainsi confirmés pour le pointage à distance et renforcés par des résultats qualitatifs et des mesures d'utilisabilité subjective.

6.2.5.2 Axe *Observabilité de l'expansion* : Expliciter l'expansion

Décalage du point de chute des mouvements Les résultats montrent une correspondance directe entre la nature *implicite* ou *explicite* de l'aide visuelle et le décalage des points de chute observé (figure 6.11). En effet, les techniques impliquant des aides visuelles *explicites* (toutes sauf TARGET, Bubble Cursor et Rope Cursor) sont centrées sur la cellule. Du côté des techniques *implicites*, TARGET et Bubble Cursor sont centrées sur la cible, tandis que Rope Cursor est mixte grâce aux mini-cordes, qui accroissent l'observabilité des bords de la cellule survolée sans les représenter directement.

Tant que les utilisateurs n'ont aucune information sur la taille étendue des cibles, ils se basent uniquement sur la taille des cibles. Grâce à l'information *explicite*, les utilisateurs visent la taille étendue de la cible et la technique est centrée sur la cellule. Ceci est cohérent avec le MIIO [88] : LES AIDES VISUELLES EXPLICITES ONT UN IMPACT DIRECT ET POSITIF SUR L'OPTIMISATION DE LA TÂCHE DE POINTAGE (H2). C'est pourquoi les deux techniques les plus rapides, MTE et VTE, sont centrées sur la cellule.

Ces résultats confirment également avec plus de précision le rationnel de conception de VTE. En effet, l'expérience 1 avait permis de mettre en évidence les différences d'optimisation du geste grâce à la mesure des différences de performance entre les techniques. Le lien de causalité entre optimisation et performance s'appuyait alors sur une interprétation théorique des différences observées à l'aide du MIIO [88]. Lors de cette deuxième expérience, les différences d'optimisation du geste sont rendues directement observables grâce à la mesure des décalages du point de chute. Cette expérience ajoute ainsi une preuve *numérique* du principe qui est à la base du rationnel de conception de VTE : afficher dès le début la taille étendue de la cible permet d'être plus performant, *car les utilisateurs visent alors la taille étendue de la cible*. En confirmant le rationnel de conception de VTE, un objectif de l'expérience est atteint.

Taux d'erreurs Les techniques utilisant une aide visuelle *explicite* durant la phase d'atterrissage du mouvement (fin de la phase balistique et phase corrective) sont les moins propices aux erreurs. Rope Cursor fait exception car son aide visuelle présente de l'information spatiale sur la proximité des cellules et des cibles voisines. Nous pensons qu'en fournissant de l'information spatiale, UNE AIDE VISUELLE EXPLICITE PERMET AUX

UTILISATEURS DE CONNAÎTRE LA DISTANCE ENTRE LE CURSEUR ET LES BORDS DE LA CELLULE SURVOLÉE ET D'ANTICIPER AINSI CERTAINES ERREURS (H3).

Le cas de Rope Cursor est démonstratif : la longueur des mini-cordes représente la distance entre le curseur et les bords des différentes cellules, sans toutefois afficher les cellules elles-mêmes. Cette aide visuelle *continue* et *implicite* lui permet d'obtenir un taux d'erreurs proche de celui des techniques *explicites*.

6.2.5.3 Axe *Élément augmenté*

VTetoROPE, la seule technique combinant des aides visuelles *sur-espace* et *sur-curseur*, est aussi la moins performante, avec un score SUS faible. Une explication est que ce changement d'élément augmenté est cognitivement trop coûteux. Au contraire, LA COMBINAISON D'UNE AIDE VISUELLE SUR-ESPACE AVEC UNE AIDE VISUELLE SUR-ESPACE OU SUR-CIBLE EST PROMETTEUSE et mérite une étude approfondie. En effet, VTetoTARGET et VTetoCELL ont toutes deux obtenu des résultats quantitatifs encourageants et une très bonne acceptabilité. Il serait de plus intéressant d'évaluer ces techniques combinées et eVTE dans le contexte concret d'une interface graphique réaliste, car leur rationnel de conception vise à réduire la surcharge visuelle d'un diagramme de Voronoï statique superposé à l'interface graphique.

6.2.6 Conclusion de l'expérience 2

Cette expérience avait pour buts d'évaluer les nouvelles techniques conçues par exploration de l'espace de conception, de concevoir des recommandations pour la conception d'aides visuelles pour les expansions de cibles, et d'approfondir notre étude des aides visuelles pour les techniques d'expansion de cibles, notamment en étudiant l'optimisation des gestes de pointage.

Parmi les nouvelles techniques, MTE a obtenu de meilleures performances que toutes les techniques utilisant la tessellation de Voronoï en distance euclidienne. Malgré sa caractéristique *implicite*, Rope Cursor est peu propice aux erreurs car la technique offre un mécanisme de prévention des erreurs efficace avec les mini-cordes.

Une nouvelle mesure permettant de savoir si les utilisateurs ont visé la cible ou bien sa taille étendue a été définie pour étudier l'optimisation des gestes de pointage. Faisant défaut dans la littérature, un nouvel agencement de cibles dédié à cette mesure a été créé. Celui-ci permet à la fois de réaliser une tâche de pointage standard ISO 9241-9 et de générer une tessellation non-centroïdale, c'est-à-dire d'introduire de façon contrôlée un décalage entre le centre des cibles et le centre de leurs cellules. Les utilisateurs visant naturellement le centre de l'objet qu'ils veulent atteindre [55], ce nouvel agencement permet de mesurer si les utilisateurs visent la cible ou la cellule. Cette nouvelle mesure et ce nouvel agencement ont permis une étude approfondie des différences d'optimisation

des gestes de pointage causées par les différences entre les aides visuelles des techniques étudiées. Ils ont également permis de confirmer le rationnel de conception de VTE.

Nous approchons le deuxième objectif de cette expérience grâce à l'analyse des résultats selon les axes de conception définis au chapitre 4. Cette analyse nous pousse à formuler plusieurs conseils pour la conception, bien que ceux-ci nécessitent de plus profondes analyses pour être pleinement confirmés par l'expérience, cette expérience 2 étant très exploratoire :

1. Le temps de sélection croît avec la dynamicité de l'aide visuelle.
2. Canaliser l'attention de l'utilisateur grâce à la dynamicité de l'aide visuelle peut rendre la technique plus précise si l'information utile est fournie au bon moment du mouvement de pointage.
3. La dynamicité d'une aide visuelle diminue son utilisabilité subjective en accroissant la charge cognitive de l'utilisateur.
4. Les utilisateurs préfèrent une aide visuelle *discrète* à une aide visuelle *continue*.
5. Les aides visuelles *explicites* ont un impact direct et positif sur l'optimisation du mouvement de pointage : les utilisateurs visent la cellule et non la cible.
6. Une aide visuelle *explicite* permet aux utilisateurs d'anticiper certaines erreurs car elle permet de connaître directement la distance entre le curseur et les bords de la cellule en fournissant de l'information spatiale.
7. La combinaison d'une aide visuelle *sur-espace* et d'une aide visuelle *sur-espace* ou *sur-cible* est prometteuse.

Conclusion

Ces deux expériences d'exploration de l'espace de conception confirment dans le cadre du pointage à distance le rationnel de conception de VTE et le principe tiré du MIO sur lequel il se base : les utilisateurs utilisent si possible la taille étendue de la cible pour optimiser leurs gestes. Tandis que l'expérience 1 conclue grâce à l'étude des performances de pointage, de l'utilisabilité subjective et des préférences des utilisateurs, l'expérience 2 met en perspective les performances de pointage selon les axes de conception et conclue notamment grâce à l'étude du point de chute des gestes. Ainsi, notre protocole expérimental s'est perfectionné de l'expérience 1 à l'expérience 2 par l'introduction d'une nouvelle mesure complémentaire aux traditionnelles mesures de performance.

Six nouvelles techniques d'expansion de cibles ont été évaluées. Parmi elles, MTE et Rope Cursor ouvrent de nouvelles pistes de réflexion en faisant figure d'exception parmi les techniques étudiées. De même TARGET, dont les performances talonnent celles de VTE malgré des caractéristiques très différentes dans deux expériences (l'exploration des variantes de conception de VTE – chapitre 5 section 5.1.1.2 – et l'expérience 2), montre

la nécessité d'approfondir notre étude des aides visuelles pour les expansions de cibles. Expansion Lens est évaluée au cours d'une troisième expérience présentée au chapitre 8.

L'espace de conception, par son pouvoir descriptif, a permis d'analyser les résultats avec une approche novatrice et fructueuse. Ces expériences concluent une démarche d'analyse des aides visuelles et de leurs performances. Dans la partie suivante, nous tirons de cette démarche et de ses conclusions une modélisation des performances des aides visuelles, progressant ainsi dans notre démarche conceptuelle d'un pouvoir descriptif puis génératif, à un pouvoir prédictif.

Troisième partie

Modèle de performance des aides visuelles

Chapitre 7

Modèle conceptuel de performance

Sommaire

Introduction	129
7.1 Information fournie par les techniques	131
7.2 Pertinence de l'information selon les phases orientées-buts	135
7.2.1 Phase de départ	135
7.2.2 Phase de transfert	136
7.2.3 Phase de validation	137
7.3 Complémentarités et limites	138
7.3.1 Définition des phases du geste de pointage	138
7.3.2 Caractérisation de l'information visuelle	139
7.3.3 Limites	139
7.4 Mise en œuvre du modèle : quantifier l'information	140
7.5 Validation analytique : aides visuelles de Fan Cursor	140
Conclusion	143

Introduction

Une technique d'expansion motrice et visuelle de cibles repose sur un *algorithme d'expansion* pour attribuer plus d'espace aux cibles et fournit une *aide visuelle* pour rendre compte de cette expansion à l'utilisateur. Au cours des chapitres précédents, nous avons constaté la grande diversité des aides visuelles existantes (chapitre 3) et possibles (chapitres 4 et 5).

Un nouvel espace de conception [59] (chapitre 4) permet de décrire ces aides visuelles indépendamment de l'algorithme d'expansion utilisé par la technique et des traits visuels de l'aide visuelle (couleur, forme, détails visuels). Ainsi, trois axes de conception ont été conçus – l'axe *Dynamacité*, l'axe *Observabilité de l'expansion* et l'axe *Élément augmenté* – et utilisés pour la conception de nouvelles techniques (chapitre 5).

Nos études expérimentales [58, 59] (chapitre 6), conduites avec des techniques de la littérature et des techniques conçues par exploration de l'espace de conception, ont permis (1) de confirmer la pertinence d'hypothèses de conception basées sur le Modèle de l'Impulsion Initiale Optimisée (MIIO) [88] comme cadre théorique pour la conception d'aides visuelles (chapitre 6, section 6.1.6) et (2) d'analyser l'impact de l'aide visuelle sur les performances d'une technique en fonction des caractéristiques de l'aide visuelle. Les recommandations (chapitre 6, section 6.2.6) résultant de l'analyse des performances en fonction des caractéristiques de l'aide visuelle constituent l'aboutissement de cette démarche théorique et expérimentale. Ces recommandations s'ajoutent aux contributions que constituent l'espace de conception, les nouvelles techniques et l'enrichissement de notre protocole expérimental. Les recommandations qui concluent l'expérience 2 du chapitre précédent constituent une première contribution pour lier les caractéristiques de l'aide visuelle avec les performances. Elles soulignent l'importance de la nature de l'information fournie à l'utilisateur (*explicite*, *implicite*, ou autre – voir Rope Cursor : chapitre 5, section 5.2.2) et de sa temporalité (dynamicité et timing).

Dans ce chapitre, nous décrivons un nouveau modèle conceptuel de performance [60] qui repose sur les deux notions de nature et de temporalité de l'information en les concrétisant par une analyse de la *pertinence* de l'information visuelle fournie par la technique et traitée par les utilisateurs. Selon la théorie de l'action de Norman [97, 132] (figure 7.1), l'information traitée par les utilisateurs est liée à leurs buts. Or, les buts des utilisateurs varient au cours d'une tâche de pointage. Nous définissons un séquençage en trois phases, basé sur les différents sous-buts de la tâche de pointage (figure 7.2). La *pertinence* de l'information fournie à l'utilisateur au cours de ces différentes phases orientées-but est le concept-clé de ce modèle. C'est cette notion qui apporte un pouvoir prédictif à ce modèle conceptuel des performances d'une technique d'expansion de cibles.

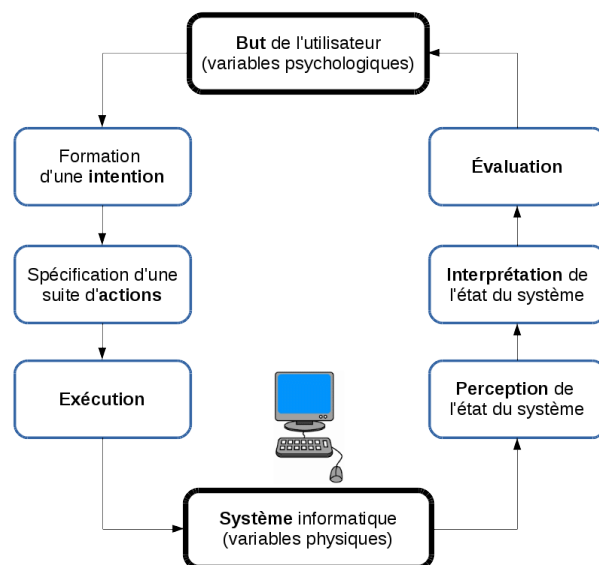


FIGURE 7.1 – Les sept étapes de la théorie de l'action selon D. Norman [97].

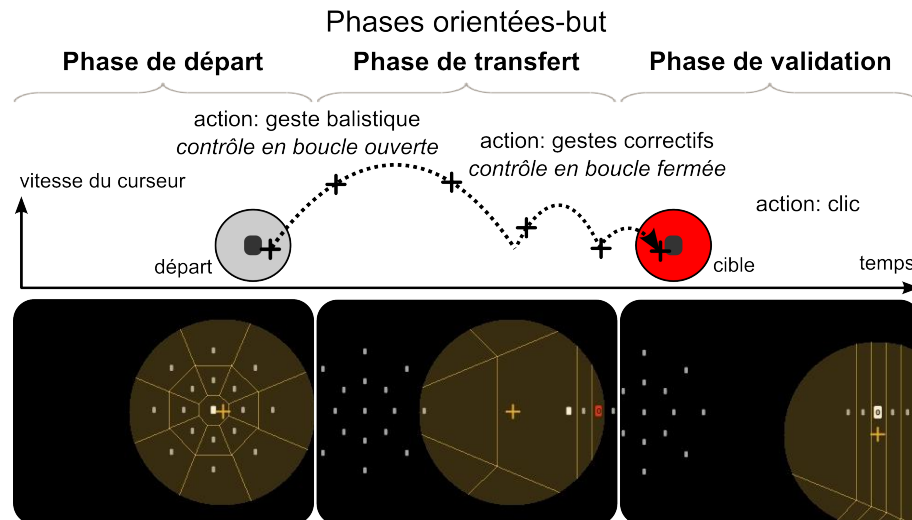


FIGURE 7.2 – Les trois phases orientées-but d’une tâche de sélection de cible, basées sur le Modèle de l’Impulsion Initiale Optimisée [88, 130] (chapitre 2 section 2.1.5) et la théorie de l’action de Norman [97]. Chaque phase est illustrée avec la technique Expansion Lens, décrite au chapitre 5.

Pour présenter notre analyse de performance des aides visuelles, nous décrivons tout d’abord l’information qu’une technique fournit. Deuxièmement, nous détaillons la division d’une tâche de pointage en trois phases orientées-but. Troisièmement, nous catégorisons l’information fournie par une technique d’expansion comme *primaire* ou *secondaire*, selon son degré d’utilité – sa pertinence – pour les décisions orientées par les buts de chaque phase. Enfin, nous expliquons comment l’information *primaire* et *secondaire* impacte la performance durant chaque phase orientée-but.

Dans une dernière section, nous identifions des limites de ce modèle conceptuel de performance et nous étudions sa complémentarité avec l’espace de conception. Enfin, nous présentons une manière d’opérationnaliser le modèle que nous illustrons en ré-interprétant certains résultats de la littérature par le prisme de ce modèle conceptuel de performance.

7.1 Information fournie par les techniques

Nous analysons l’information fournie à l’utilisateur par les techniques d’expansion de cibles. Une interface graphique pour le pointage affiche au minimum un curseur et une cible. Elle fournit ainsi les informations suivantes :

1. la position du curseur,
2. la position de la cible,
3. la forme de la cible.

Ces deux dernières informations permettent de percevoir l’aire d’activation de la cible.

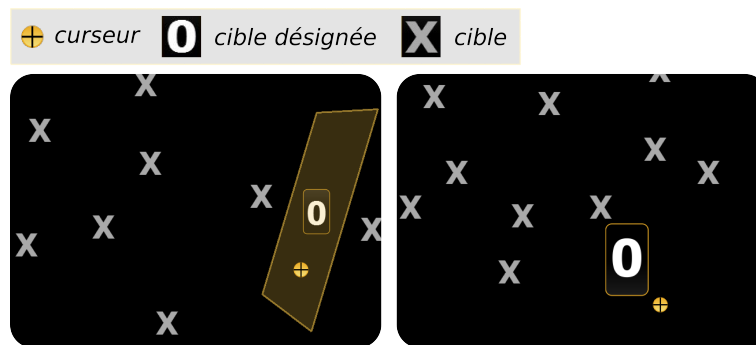


FIGURE 7.3 – À gauche, Cell Painting affiche la taille étendue de la cible. À droite, TARGET met en évidence la cible désignée en l'agrandissant.

Classiquement en pointage, le fait que le curseur est à l'intérieur de l'aire d'activation de la cible signifie que cette cible est désignée (le curseur la survole). Mais, quand nous utilisons une technique d'expansion de cibles, une cible peut être désignée sans que le curseur la survole (figure 7.3). Dans ce cas, une aide visuelle supplémentaire, comme un changement de couleur de la cible, est nécessaire pour informer les utilisateurs que la cible est désignée.

Au-delà de cette aide visuelle "minimale", une technique d'expansion de cibles fournit de différentes façons de l'information spatiale sur la taille étendue des cibles, comme le décrit l'axe *Observabilité de l'expansion* de l'espace de conception (chapitre 4). Cell Painting, une technique *explicite*, affiche la taille étendue de la cible désignée (figure 7.3 à gauche), tandis que TARGET, une technique *implicite*, affiche une version *agrandie* de la cible (figure 7.3 à droite). Ces deux techniques fournissent deux informations différentes : Cell Painting indique la forme d'une cellule de Voronoï (la forme étendue de la cible désignée) et TARGET indique la cible désignée en l'agrandissant. Cette cible agrandie peut parfois induire en erreur si elle empiète sur une cellule de Voronoï voisine. On parle alors de *fausse affordance* [132], un type d'inconsistance de l'interface graphique qui peut entraîner des erreurs de sélection.

Outre la taille étendue ou agrandie des cibles, les techniques d'expansion de cibles fournissent d'autres informations. Considérons par exemple Bubble Cursor (figure 7.4) :

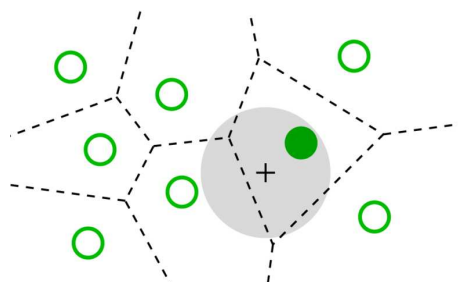


FIGURE 7.4 – Bubble Cursor se base sur un partage de l'espace disponible entre les cibles par une tessellation de Voronoï. Si elle peut, la bulle englobe totalement la cible désignée.

- La *position de la bulle* et sa *circonférence* permettent de connaître le centre de la bulle et, par conséquent, la position du curseur.
- Le *rayon de la bulle* indique la distance entre le curseur et la cible désignée. Cette distance varie dans l'intervalle défini par (1) la distance du curseur au point de la cible le plus proche et (2) la distance du curseur au point de la cible le plus éloigné, selon si la bulle peut englober ou non la cible désignée (figure 7.4).
- Le *contact de la bulle*, qui n'existe qu'avec une seule cible à la fois, indique la cible désignée.
- Selon les situations, cette information peut être complétée par la *résistance de la bulle au contact*, qui indique la cible la plus proche après la cible désignée. En effet, la bulle ajuste sa taille pour ne toucher qu'une seule cible à la fois, la plus proche du curseur. Son rayon est limité en longueur par la distance du curseur (centre de la bulle) au point le plus proche de la deuxième cible la plus proche du curseur. À l'usage, cet ajustement simule une résistance de la bulle au contact avec la deuxième cible la plus proche, ce qui permet d'identifier celle-ci.
- Dans certaines situations (par exemple, des cibles alignées et espacées), le *mouvement du bord de la bulle* opposé au bord en contact indique la distance entre le curseur et la cellule de Voronoï qui se trouve dans la direction du mouvement du curseur. En effet, la bordure de la bulle se déplace à deux fois la vitesse du curseur et dans la même direction (figure 7.5). Son déplacement peut donc être utilisé pour évaluer la distance restante à la cible précédente (ou suivante) dans la direction du mouvement. En pratique, cette information permet aux utilisateurs d'*anticiper* les contacts de la bulle avec les cibles.

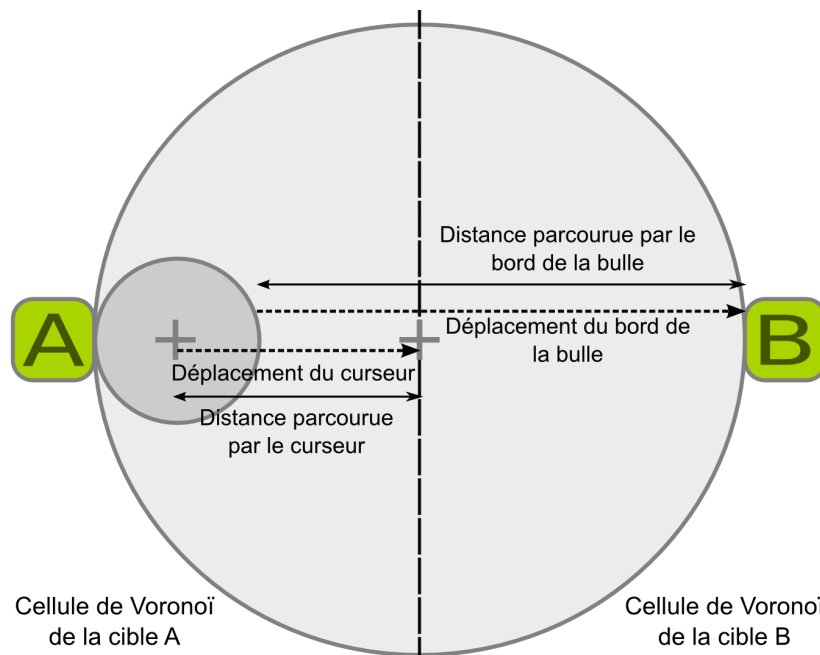


FIGURE 7.5 – Déplacement du bord de la bulle de Bubble Cursor : le bord de la bulle qui suit la direction du mouvement parcourt deux fois la distance que le curseur parcourt, entre la situation où Bubble Cursor désigne la cible A, à gauche, et la situation où Bubble Cursor atteint la cible B (curseur au centre, entre les deux cellules de Voronoï).

Aide Visuelle	Source d'Information	Information	Observabilité de l'Expansion	Primaire pour la Phase de Départ	Primaire pour la Phase de Transfert	Primaire pour la Phase de Validation
Bubble Cursor	Position de la bulle	Position du curseur	Implicite	Non	Oui	Oui
	Contact de la bulle	Cible désignée et voisines	Implicite	Non	Non	Oui
	Mouvement du bord de la bulle	Distance à la prochaine cible	Implicite	Non	Non	Oui
	Rayon de la bulle	Distance à la cible désignée	Implicite	Non	Non	Oui
IFC: Forme circulaire	Position de la bulle	Position du curseur	Implicite	Non	Oui	Oui
	Rayon de la bulle	Distance à la cible désignée	Implicite	Non	Non	Oui
IFC: Forme d'éventail	Position de l'éventail	Position du curseur	Implicite	Non	Oui	Oui
	Rayon de l'éventail	Distance à la cible désignée	Implicite	Non	Non	Oui
	Contact de l'éventail	Cible désignée	Implicite	Non	Non	Oui
	Angle d'ouverture de l'éventail	Vitesse du curseur	Implicite	Non	Non	Oui
	Angle d'ouverture et orientation de l'éventail	Cibles potentiellement désignées	Implicite	Non	Non	Oui
TARGET	Forme agrandie de la cible désignée	Forme agrandie de la cible désignée	Implicite	Non	Non	Non
	Agrandissement d'une cible	Cible désignée	Implicite	Non	Oui	Oui
Cell Painting	Forme étendue de la cible désignée	Forme étendue de la cible désignée	Explicite	Non	Non	Oui
	Coloration d'une cellule	Cible désignée	Implicite	Non	Non	Oui
VTE	Forme étendue de la cible-objectif	Forme étendue de la cible-objectif	Explicite	Oui	Oui	Oui
	Forme étendue des distracteurs	Forme étendue des distracteurs	Explicite	Non	Non	Non
eVTE	Forme étendue de la cible-objectif	Forme étendue de la cible-objectif	Explicite	Oui	Oui	Oui
	Forme étendue des distracteurs	Forme étendue des distracteurs	Explicite	Non	Non	Non
	Transparence du diagramme	Vitesse du curseur	Implicite	Non	Non	Oui
Expansion Lens	Position de la lentille	Position du curseur	Implicite	Non	Oui	Oui
	Contenu de la lentille	Forme étendue de certaines cibles	Explicite	Non	Non	Oui
Rope Cursor	Extrémité "curseur" de la corde principale	Position du curseur	Implicite	Non	Oui	Oui
	Extrémité "cible" de la corde principale et mini-cordes	Cible désignée et voisines	Implicite	Non	Non	Oui
	Mouvement de la 1 ^{ère} mini-rope	Distance à la prochaine cible	Implicite	Non	Non	Oui
	Longueur de la corde principale	Distance à la cible désignée	Implicite	Non	Non	Oui

FIGURE 7.6 – Analyse de l'information fournie par neuf techniques d'expansion de cibles. L'information secondaire durant la phase de transfert est en gras.

Nous avons analysé de la même façon l'information fournie par huit autres techniques d'expansion de cibles : Implicit Fan Cursor version curseur-bulle, Implicit Fan Cursor version curseur-éventail, TARGET et Cell Painting (les techniques issues de la littérature et décrites au chapitre 3), VTE, eVTE, Expansion Lens et Rope Cursor (des nouvelles techniques décrites au chapitre 5). Cette analyse est résumée à la figure 7.6, aux colonnes intitulées "Source d'information" et "Information".

7.2 Pertinence de l'information selon les phases orientées-buts

Nous classons l'information fournie par les techniques comme *primaire* ou *secondaire* selon les buts de l'utilisateur, qui varient au cours du geste. Les termes *primaire* et *secondaire* permettent de faire une distinction entre les degrés d'utilité de l'information pour atteindre le but courant. Nous définissons dans un geste de pointage trois phases consécutives (figure 7.2) qui sont basées sur l'évolution des buts des utilisateurs : la phase de *départ*, la phase de *transfert* et la phase de *validation*. Durant ces phases, l'information *primaire* est définie comme celle qui est nécessaire à l'utilisateur pour atteindre son but. Par opposition, l'information *secondaire* concerne des paramètres optionnels pour la bonne réalisation de cet but. La figure 7.6 résume cette analyse de la pertinence de l'information pour neuf techniques. Les sections suivantes définissent les trois phases et l'impact de l'information primaire et secondaire sur les performances pour chacune des trois phases.

7.2.1 Phase de départ

La phase de *départ* correspond à la phase préliminaire dans le modèle de Meyer et al. [88] (chapitre 2 section 2.1.5). Le but de l'utilisateur est d'initier le mouvement vers la cible. La phase de départ s'étend du début de la tâche de pointage jusqu'au début du sous-mouvement balistique. Le temps de départ inclut donc le temps nécessaire pour :

1. comprendre que la tâche a commencé,
2. traiter l'information liée aux paramètres du mouvement à réaliser (direction, longueur, précision requise),
3. programmer mentalement le mouvement,
4. commencer le mouvement du curseur, c'est-à-dire initier le sous-mouvement balistique.

L'action résultant de la phase de départ est le sous-mouvement balistique, qui est produit sous le contrôle *en boucle ouverte* de l'utilisateur. Durant cette phase, l'information *primaire* est :

1. l'information signifiant que la tâche est commencée : un événement particulier ou, dans le cas de sélections séquentielles rapides, un retour visuel signifiant que la sélection précédente est terminée.
2. l'information nécessaire pour préparer le mouvement [88, 97] : la position de la cible, sa forme et, si possible, sa forme étendue.

Nous ne traitons pas la position du curseur, qui serait une information *primaire*, car nous faisons l'hypothèse que les utilisateurs connaissent *a priori* la position du curseur, c'est-à-dire avant de commencer la tâche. Les autres informations, comme la forme étendue des cibles non visées (les distracteurs), sont catégorisées comme *secondaires* pour la phase de départ.

L'information *primaire* est particulièrement nécessaire si les utilisateurs n'ont aucune connaissance préliminaire de la cible. Le besoin de traiter cette information est moins pressant lorsque la cible est bien connue, comme l'icône de la corbeille ou du navigateur internet. Le standard ISO 9241-9 [70] pour évaluer les techniques de pointage met les utilisateurs dans cette situation de familiarité avec l'interface : l'information *primaire* est connue car les cibles sont identiques et apparaissent toujours dans le même ordre, la séquence de leur position étant répétée au cours des conditions. Après quelques blocs d'essais, l'utilisateur sait à chaque instant où va apparaître la cible suivante.

Suite à l'étude des travaux de Hick (1952) [68] (chapitre 2), Hyman (1953) [69] et Seow (2005) [120], nous considérons dans ce modèle conceptuel de performance que plus il y a d'information à traiter, plus les utilisateurs ont besoin de temps pour la comprendre et l'utiliser. Sans connaissance préliminaire de la cible, les utilisateurs ont besoin de temps pour traiter l'information *primaire*. En conséquence, ce modèle conceptuel prédit que **l'information *primaire* allonge la durée de la phase de départ.**

7.2.2 Phase de transfert

La seconde phase est la phase de *transfert*. Durant cette phase, le but des utilisateurs est de gérer le geste pour stabiliser le curseur à l'intérieur de la taille étendue de la cible. Cette phase couvre donc le mouvement du curseur depuis le début du sous-mouvement balistique jusqu'à la *dernière* entrée du curseur dans la taille étendue de la cible. En effet, dans le cas typique d'un sous-mouvement balistique trop long, le curseur traverse la taille étendue de la cible mais la phase de transfert se poursuit car le but n'est pas atteint : l'utilisateur devra corriger son geste pour ramener le curseur dans la zone visée.

La phase de transfert inclut le sous-mouvement balistique et les potentiels sous-mouvements correctifs, que l'utilisateur contrôle *en boucle fermée* (figure 7.2). Durant cette phase aussi, l'information nécessaire à l'utilisateur pour atteindre son but constitue l'information *primaire*. Sous contrôle *en boucle fermée*, il s'agit de l'information utilisée comme *retour visuel* par l'utilisateur pour gérer le mouvement du curseur jusqu'à son entrée dans la taille étendue de la cible. Elle permet d'évaluer la progression du mouvement vers la cible. L'information *primaire* pour la phase de transfert inclut :

- la position du curseur,
- les caractéristiques visibles de la cible (position, taille ou taille étendue) [88],
- un éventuel mécanisme de signalement de la cible désignée, indispensable pour profiter de l'expansion de cibles lorsque la taille étendue de la cible n'est pas visible, comme avec une technique caractérisée d'*implicite* dans notre espace de conception.

Toute autre information fournie par la technique est *secondaire*.

Nous avons constaté lors de pré-tests que la phase de transfert est souvent la phase orientée-but la plus longue dans une tâche de pointage. Nous expliquons la plupart des différences de performance observées entre les techniques d'expansion de cibles [34, 35, 59, 126] par l'impact de l'information *secondaire* sur le temps de transfert. En particulier, nous détaillons dans la section 7.5 ci-après notre interprétation des résultats expérimentaux ayant servi à déterminer la meilleure aide visuelle pour la technique Implicit Fan Cursor [126].

En raison du contrôle *en boucle fermée* des sous-mouvements correctifs, l'information fournie est traitée pour en extraire l'information *primaire* de l'information *secondaire*. L'information *primaire* et *secondaire* a donc un impact sur le *temps de transfert*. En particulier, et cela constitue la deuxième prédiction du modèle : **l'information *secondaire* allonge la durée de la phase de transfert.**

7.2.3 Phase de validation

La troisième et dernière phase est celle de *validation*. Le but de l'utilisateur est de valider correctement la sélection. Cette phase commence lorsque le curseur est arrivé dans une position lui permettant de finaliser la sélection, c'est-à-dire lorsque le curseur est dans la taille étendue de la cible. Elle se termine lorsque la sélection est validée, par un clic par exemple. Lors d'études expérimentales, une validation incorrecte lors de la phase de validation provoque une erreur car la cible n'est pas sélectionnée. Les expériences 1 et 2 (chapitre 6) ont d'ailleurs montré que les techniques *explicites*, qui affichent la taille étendue de la cible, sont moins propices aux erreurs que les techniques *implicites*.

Durant cette phase, les utilisateurs ont besoin de s'assurer que l'action de validation va finaliser la tâche de sélection avec succès. Cognitivement, ils cherchent deux certitudes :

1. le curseur est dans la taille étendue de la cible,
2. le curseur est suffisamment éloigné des bords de la taille étendue de la cible pour qu'un potentiel déplacement du curseur pendant l'action de validation (effet Heisenberg, chapitre 2 section 2.3.1.4) n'en affecte pas le résultat.

L'information *primaire* inclut donc :

- la position du curseur,
- la proximité des tailles étendues des cibles voisines, à défaut la proximité des cibles voisines,

— la taille étendue de la cible, à défaut sa taille d'origine.

Les autres informations concernant les autres cibles sont *secondaires* durant cette phase.

Dans la phase de validation comme dans les autres phases du geste, les utilisateurs ont besoin de temps pour traiter l'information, ce qui amène à la prédiction du modèle conceptuel concernant cette phase : **l'information *primaire* allonge le *temps de validation*.**

L'expérience 2 (chapitre 6) nous a permis de formuler des recommandations aux concepteurs et de constater le fort impact de l'aspect explicite/implicite de l'aide visuelle sur les taux d'erreurs (les techniques explicites sont moins propices aux erreurs). Ne pouvant faire de lien entre l'aspect explicite/implicite des informations fournies et leur aspect primaire/secondaire durant la phase de validation, le modèle ne nous permet pas de formuler de prédiction sur les taux d'erreurs des techniques d'expansion. Cette recommandation issue de l'expérience 2 est complémentaire au modèle conceptuel de performance afin de formuler des prédictions sur l'ensemble des performances du pointage, durée et taux d'erreurs.

Un compromis vitesse-précision est à l'œuvre dans la phase de validation, bien qu'elle ne soit pas modélisée par la loi de Fitts. En effet, tandis que l'information *explicite* sur la cible et ses voisines, *primaire* pour cette phase, limite les erreurs de sélection (chapitre 6, expérience 2 [59]), elle ajoute au temps de validation le temps nécessaire pour la traiter.

7.3 Complémentarités et limites

7.3.1 Définition des phases du geste de pointage

Ce modèle de performance considère trois phases orientées-but dans un geste de pointage. Ces phases sont basées sur l'évolution des buts de l'utilisateur au cours du geste (théorie de l'action de Norman [97], figure 7.1). En cela, ce modèle adopte sur le geste de pointage un point de vue complémentaire au point de vue *cinématique* de Meyer et al. [88], Nieuwenhuizen et al. [96], Plamondon et al. [107] et de l'espace de conception défini au chapitre 4. En effet, l'espace de conception considère que la troisième phase est définie par le ralentissement du curseur à l'approche de la cible et la présence éventuelle de sous-mouvements correctifs. Ce point de vue est *cinématique* car les phases du geste sont définies en fonction des profils de vitesse et d'accélération du curseur. Par opposition, la troisième phase de ce modèle conceptuel est définie par la présence stable du curseur dans la taille étendue de la cible et par le changement de but consécutif à cet événement, indépendamment des profils de vitesse du curseur. Les événements de transition entre deux phases orientées-but ne sont pas des événements *cinématiques* (par exemple un ralentissement du curseur), mais des événements *cognitifs*.

7.3.2 Caractérisation de l'information visuelle

Pour chaque phase orientée-but, la pertinence de l'information visuelle est définie en fonction des buts des utilisateurs (théorie de l'action de Norman [97], figure 7.1) et du type de contrôle (*en boucle ouverte* ou *en boucle fermée*) exercé par l'utilisateur sur son geste de pointage (Modèle de l'Impulsion Initiale Optimisée [88]). Ainsi, l'analyse des aides visuelles, fondée sur la pertinence de l'information visuelle pour chaque phase, conduit à faire la distinction entre information *primaire* ou *secondaire*.

Cette analyse étend la caractérisation de l'information visuelle au-delà de l'axe *Observabilité de l'expansion* de l'espace de conception (chapitre 4). En effet, cet axe ne permet de caractériser l'information visuelle que comme *explicite* ou *implicite* : l'affichage ou non de la taille étendue de la cible. Au contraire analyser toutes les informations fournies par l'aide visuelle et les caractériser comme primaire ou secondaire permet de mieux détailler l'information visuelle. Par exemple, l'information *explicite* est détaillée en fonction des cibles dont elle affiche la taille étendue : la cible désignée, la cible objectif, les cibles non-visées (figure 7.6). L'information implicite est également détaillée en informations élémentaires : la cible désignée, la distance à la cible ou la vitesse du curseur par exemple (figure 7.6).

7.3.3 Limites

Ce modèle de performance n'est qu'une première étape vers la compréhension de l'utilisation de l'information visuelle par les utilisateurs lors d'une tâche de pointage avec une technique d'expansion de cibles. En effet, il définit des phases orientées-but *indépendantes* et *séquentielles* : les effets d'anticipation d'une phase à l'autre et de parallélisation des phases ne sont pas pris en compte.

Par exemple, ce modèle ne permet pas de prendre en compte l'*effet d'optimisation du geste balistique*, qui n'est observable que lors de tâches de pointage ayant un Indice de Difficulté (*ID*) assez bas. En effet, l'information *explicite* permettant à l'utilisateur d'optimiser ses premiers sous-mouvements pour la cible étendue est *primaire* pour la phase de départ et l'allonge. En revanche, elle permet à l'utilisateur de mieux optimiser la phase de transfert et de la raccourcir. Ce modèle conceptuel de performance ne permet pas de modéliser cet *effet d'optimisation du geste balistique*, qui relie une information *primaire* pour la phase de départ à la durée de la phase de transfert.

De plus, nous n'avons considéré dans cette analyse que des techniques d'expansion de cibles fournissant la même aide visuelle durant les trois phases du geste. Les combinaisons d'aides visuelles, comme celles présentées au chapitre 5 et évaluées au chapitre 6, n'entrent pas dans le champ d'application du modèle conceptuel, car l'expérience 2 (chapitre 6) a montré un coût de changement entre les aides visuelles (coût cognitif et temporel) trop important. Étendre cette analyse aux techniques combinant différentes aides visuelles nécessite de modéliser ce coût de changement.

7.4 Mise en œuvre du modèle : quantifier l'information

Les différentes informations que présentent une technique à l'utilisateur sont identifiées en menant une analyse détaillée de la technique, comme l'analyse de Bubble Cursor présentée à la section 7.1 ou celle de huit autres techniques résumée à la figure 7.6. Cependant, cette analyse ne suffit pas à opérationnaliser le modèle conceptuel de performance pour faire des prédictions de performance. En effet, il faut pouvoir comparer l'information fournie par les techniques au cours du geste de pointage pour pouvoir comparer deux techniques d'expansion de cibles. Or, comparer l'information fournie implique au préalable de la quantifier, au minimum de façon relative.

Pour mettre en œuvre le pouvoir prédictif de ce modèle, il est donc nécessaire de quantifier l'information. L'analyse des techniques, comme celle de neuf techniques à la figure 7.6, permet de dénombrer les différentes informations fournies par les techniques.

Nous choisissons donc dans un premier temps comme approximation pour quantifier l'information, le nombre d'informations différentes identifiées par l'analyse des techniques. Nous revenons dans les perspectives à ces travaux au chapitre 9.3.4 sur la quantification des informations fournies. La section suivante fournit un exemple d'utilisation du modèle conceptuel ainsi opérationnalisé à travers l'interprétation de résultats expérimentaux de la littérature.

7.5 Validation analytique : aides visuelles de Fan Cursor

Les différentes formes de la technique Implicit Fan Cursor [126] présentées au chapitre 2 sont analysées grâce au modèle conceptuel de performance. Dans l'étude de Su et al. (2014), une expérience est réalisée afin de déterminer la forme de Fan Cursor la plus performante en terme de performance de pointage. Les prédictions du modèle conceptuel de performance correspondent pleinement aux résultats de cette expérience, ce qui apporte une première validation analytique de ce modèle.

Fan Cursor a été conçue avec trois alternatives visuelles, c'est-à-dire trois aides visuelles différentes, utilisant toutes le même algorithme d'expansion spécifique à cette technique (figure 7.7).

- L'aide visuelle minimale de Fan Cursor : une croix permettant de situer le curseur et un mécanisme de mise en valeur de la cible désignée. Cela constitue également une aide visuelle minimale pour toute technique d'expansion de cibles.
- Le curseur-bulle : une bulle circulaire centrée sur le curseur croît jusqu'à atteindre la cible désignée. Cette bulle s'ajoute à l'aide visuelle minimale et constitue la première alternative.
- Le curseur-éventail : la part de la bulle circulaire qui est dirigée vers la cible désignée est ajoutée à l'aide visuelle minimale. Le curseur résultant a une forme d'éventail, qui s'ouvre et se ferme selon la vitesse du curseur. Lorsque le curseur

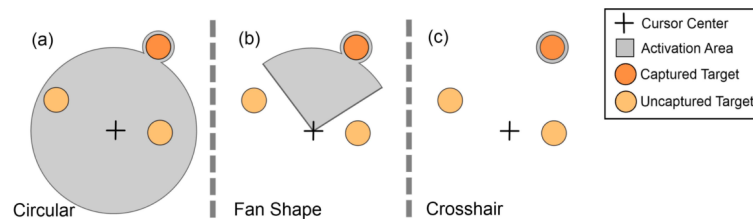


FIGURE 7.7 – Aides visuelles de Fan Cursor : (a) le curseur-bulle, (b) le curseur-éventail et (c) l'aide visuelle minimale. Figure modifiée de [126].

accélère, l'éventail s'ouvre jusqu'à 180° maximum. Lorsque le curseur ralentit, l'éventail se referme jusqu'à 90° d'ouverture minimum. L'éventail est toujours dirigé vers la cible désignée, c'est-à-dire dans la direction du mouvement du curseur.

L'analyse de ces aides visuelles a permis d'identifier les informations qu'elles fournissent. Cette analyse est résumée à la figure 7.6. L'aide visuelle minimale n'apparaît pas à la figure 7.6 car elle constitue une base présente dans les trois aides visuelles : un curseur et une mise en valeur de la cible désignée.

Les trois variantes de Fan Cursor sont comparées lors d'une expérience constituée de tâches de pointage ISO 9249-9 (900 tâches de pointage étant enregistrées) accomplies dans des conditions de bureau sur un ordinateur avec un écran 24 pouces, une souris et la configuration par défaut de Microsoft Windows 7 de la souris. L'étude est présentée comme une étude préliminaire (seulement six participants, tous droitiers).

Considérant l'analyse des aides visuelles de Fan Cursor, les prédictions du modèle de performance sont les suivantes :

1. Pour la phase de départ : aucune différence n'est observable entre les trois alternatives visuelles car aucune d'elles ne fournit d'information *primaire*.
2. Pour la phase de transfert : comme le curseur-bulle fournit plus d'informations *secondaires* que l'aide minimale (figure 7.6), le curseur-bulle est plus lent que l'aide minimale et, comme le curseur-éventail fournit plus d'informations *secondaires* que le curseur-bulle (figure 7.6), le curseur-éventail est plus lent que le curseur-bulle.
3. Pour la phase de validation : comme le curseur-bulle fournit plus d'informations *primaires* que l'aide minimale (figure 7.6), le curseur-bulle est plus lent que l'aide minimale et, comme le curseur-éventail fournit plus d'informations *primaires* que le curseur-bulle (figure 7.6), le curseur-éventail est plus lent que le curseur-bulle.
4. Concernant les erreurs, le curseur-éventail indique les cibles voisines à proximité par son ouverture et son orientation (figure 7.6). Comme Rope Cursor, cette information *implicite* permet une meilleure prévention des erreurs. Le curseur-éventail génère donc moins d'erreurs que le curseur-bulle. L'aide visuelle minimale doit causer encore plus d'erreurs car elle fournit l'information primaire minimale et *implicite*, sans aucun indice sur la proximité des cibles voisines. Cependant, ce manque d'information peut aussi conduire l'utilisateur à poursuivre son geste vers la cible pour s'assurer d'une validation correcte, plaçant ainsi le curseur loin

des bords de la taille étendue de la cible. Dans ce cas, l'aide visuelle minimale génère moins d'erreurs que les autres aides visuelles.

L'étude de Su et al. montre également que toute l'information *secondaire* durant la phase de transfert est bien traitée par les participants. En effet, l'étude [126] rapporte que :

"Certains participants ont exprimé que le curseur-éventail et le curseur-bulle entraînent tous deux de la distraction visuelle, due à leur taille changeant constamment. Ils attirent l'attention de l'utilisateur, causant des changements de focus superflus entre le curseur et les cibles. Pour le curseur-éventail en particulier, l'aire d'activation en forme d'éventail les distrait de la sélection de la cible désirée pour se concentrer sur la manipulation de l'orientation du curseur."

Texte original : *"Some participants commented that both the fan-shape and circular cursors caused visual distraction, due to their continuous changing sizes. They drew user attention, causing unnecessary focus switching between the cursor and the targets. For the fan-shape cursor in particular, the fan-shape activation area distracted them from the selection of the desired target to the manipulation of the cursor orientation."*

Les prédictions du modèle conceptuel concernant le temps total de sélection (pour le geste entier) peuvent être synthétisées de la façon suivante : le curseur-bulle est plus lent que l'aide minimale et plus rapide que le curseur-éventail.

Les résultats de l'expérience confirment les prédictions de notre modèle (figure 7.8) : l'aide minimale est la plus rapide, suivie par le curseur-bulle et enfin, le curseur-éventail. La prédiction concernant les erreurs est également confirmée par l'expérience : le curseur-éventail génère moins d'erreurs que le curseur-bulle.

Les résultats rapportent également que l'aide minimale génère moins d'erreurs que les deux autres variantes de Fan Cursor, malgré une information *primaire* moins importante durant la phase de validation (figure 7.8). La pauvreté de l'information primaire durant la phase de validation peut donc pousser l'utilisateur à rapprocher le curseur de la cible pour assurer sa validation. Or, dans le cas de Fan Cursor, cela éloigne le curseur de la taille étendue des autres cibles en raison de son algorithme d'expansion particulier [126].

L'aide visuelle minimale étant la plus performante, c'est elle qui est choisie pour la forme définitive de Fan Cursor, qui devient alors Implicit Fan Cursor.

Display Style	Mean Movement Time	Error Rate
Fan-shape	851ms	9.7%
Circular	844ms	13.6%
Implicit (Crosshair only)	819ms	7.3%

FIGURE 7.8 – Résultats de la comparaison des aides visuelles pour Fan Cursor. De gauche à droite, les colonnes présentent l'aide visuelle, le temps moyen de mouvement et le taux d'erreur. De haut en bas, les aides visuelles sont le curseur-éventail, le curseur-bulle et l'aide visuelle minimale. Figure issue de [126].

Conclusion

Dans ce chapitre, nous avons présenté un nouveau modèle conceptuel de performance des gestes de pointage effectués avec une technique d'expansion de cibles. Ce modèle est basé sur un découpage du geste en trois phases orientées-but et sur une analyse de l'information visuelle fournie par les techniques d'expansion de cibles. Le concept-clé de ce modèle conceptuel est la pertinence de l'information visuelle en fonction des phases orientées-but du geste. Il permet de caractériser l'information visuelle comme *primaire* ou *secondaire* pour chaque phase. Le modèle conceptuel de performance considère alors l'impact de cette information sur la durée de chaque phase orientée-but.

Ce modèle est complémentaire à l'espace de conception présenté au chapitre 4. Il permet de mieux détailler les informations explicites et implicites fournies par les aides visuelles que l'axe *Observabilité de l'expansion*. Il adopte de plus un point de vue *cognitif* sur les phases du pointage, complémentaire au point de vue *cinématique* des travaux de la littérature [88, 96, 107]. Les trois phases orientées-but décrivent la tâche de pointage différemment, notamment la phase de validation, des trois phases du MIO (chapitre 2 section 2.1.5) utilisées par l'espace de conception. Cependant, certaines limites de ce modèle conceptuel ont été identifiées, comme la séquentialité et l'indépendance des phases orientées-but. Ce modèle conceptuel est une première tentative de modélisation des performances d'une technique d'expansion de cibles en fonction de l'information fournie par l'aide visuelle. De nombreuses améliorations sont possibles.

Nous opérationnalisons le modèle en quantifiant l'information, au moins de façon relative, par le nombre d'informations différentes identifiées par l'analyse de chaque technique d'expansion de cibles. Cette approximation permet une ré-interprétation des résultats de l'expérience préliminaire réalisée pour déterminer la meilleure aide visuelle d'Implicit Fan Cursor. Les prédictions du modèle pour cette expérience sont confirmées par les résultats, ce qui apporte une première validation analytique du modèle conceptuel de performance.

Dans le chapitre suivant, nous confrontons ce modèle conceptuel de performance à l'expérience en ré-interprétant les résultats de l'expérience 2 (chapitre 6) et en réalisant une expérience qui compare plusieurs techniques d'expansion de cibles pour lesquelles nous avons mené une analyse complète de l'information que leur aide visuelle fournit à l'utilisateur.

Chapitre 8

Validation expérimentale du modèle conceptuel de performance

Sommaire

Introduction	146
8.1 Analyse de l'expérience 2	146
8.1.1 Introduction	146
8.1.2 Analyse des mouvements	148
8.1.3 Résultats et discussion	148
8.1.3.1 Méthode	149
8.1.3.2 Phase de transfert	149
8.1.3.3 Phase de validation	150
8.1.4 Conclusion de l'analyse de l'expérience 2	151
8.2 Expérience 3	152
8.2.1 Introduction	152
8.2.2 Protocole expérimental	153
8.2.2.1 Participants et appareillage	153
8.2.2.2 Détail des techniques implémentées	154
8.2.2.3 Tâche et procédure	154
8.2.2.4 Conditions	155
8.2.3 Analyse des mouvements	156
8.2.4 Résultats et discussion	156
8.2.4.1 Méthode	156
8.2.4.2 Phase de départ	157
8.2.4.3 Phase de transfert	158
8.2.4.4 Phase de validation	159
8.2.4.5 Résultats globaux	160
Résultats quantitatifs	160
Loi de Fitts	160
Résultats qualitatifs	162
8.2.5 Conclusion de l'expérience 3	162

Introduction

Le chapitre précédent présente un nouveau modèle de performance de la tâche de pointage avec une technique d'expansion de cibles. Ce modèle catégorise les informations fournies par les techniques comme *primaires* ou *secondaires* selon leur pertinence vis-à-vis des différents buts de l'utilisateur au cours de son geste de pointage. Il prédit ensuite l'impact de ces informations sur la durée de chaque phase orientée-but du geste. Le modèle est opérationnalisé grâce à une approximation de l'information fournie par les techniques : nous considérons pour mesurer l'information le nombre d'informations différentes que l'analyse au préalable des techniques a pu dénombrer.

Dans ce chapitre, nous mettons ce modèle à l'épreuve grâce à une analyse des résultats de l'expérience 2 du chapitre 6 et grâce à une nouvelle expérience impliquant 5 techniques, toutes utilisant la tessellation de Voronoï comme algorithme d'expansion : Expansion Lens, Cell Painting, TARGET, VTE et Bubble Cursor (figure 8.5). Le fait de considérer des techniques reposant toutes sur la tessellation de Voronoï garantit que les différences de performance ne sont dues qu'aux différences dans les aides visuelles.

8.1 Analyse de l'expérience 2

8.1.1 Introduction

Le but de cette analyse des résultats de l'expérience 2 [59] (chapitre 6 section 6.2) est de confronter les prédictions du modèle conceptuel de performance à un premier jeu de données expérimentales.

Pour cela, nous sélectionnons tout d'abord, parmi les 9 techniques d'expansion de cibles que compare cette expérience, celles qui entrent dans le champ d'application du modèle, c'est-à-dire celles qui ne changent pas d'aide visuelle durant un geste de pointage. En effet, comme détaillé au chapitre précédent, ces techniques mettent en jeu un coût de changement d'aide visuelle que le modèle conceptuel de performance ne modélise pas à l'heure actuelle. De plus, seules les techniques utilisant la tessellation de Voronoï en distance *euclidienne* sont étudiées dans cette analyse. Nous considérons donc cinq parmi les neuf techniques : VTE, eVTE, TARGET, Bubble Cursor et Rope Cursor. Les informations fournies par ces techniques sont analysées et reportées à la figure 8.1.

Lors de l'expérience 2, les participants utilisent le Virtual Pointer, un système de ray-casting présenté au chapitre 2 section 2.3.1, pour contrôler le curseur et une pédale médicale pour valider leurs sélections avec le pied. Les détails du protocole expérimental de cette expérience sont présentés au chapitre 6 section 6.2.3.

Aide Visuelle	Source d'Information	Information	Observabilité de l'Expansion	Primaire pour la Phase de Départ	Primaire pour la Phase de Transfert	Primaire pour la Phase de Validation
Bubble Cursor	Position de la bulle	Position du curseur	Implicite	Non	Oui	Oui
	Contact de la bulle	Cible désignée et voisines	Implicite	Non	Non	Oui
	Mouvement du bord de la bulle	Distance à la prochaine cible	Implicite	Non	Non	Oui
	Rayon de la bulle	Distance à la cible désignée	Implicite	Non	Non	Oui
TARGET	Forme agrandie de la cible désignée	Forme agrandie de la cible désignée	Implicite	Non	Non	Non
	Agrandissement d'une cible	Cible désignée	Implicite	Non	Oui	Oui
VTE	Forme étendue de la cible-objectif	Forme étendue de la cible-objectif	Explicite	Oui	Oui	Oui
	Forme étendue des distracteurs	Forme étendue des distracteurs	Explicite	Non	Non	Non
eVTE	Forme étendue de la cible-objectif	Forme étendue de la cible-objectif	Explicite	Oui	Oui	Oui
	Forme étendue des distracteurs	Forme étendue des distracteurs	Explicite	Non	Non	Non
	Transparence du diagramme	Vitesse du curseur	Implicite	Non	Non	Oui
Rope Cursor	Extrémité "curseur" de la corde principale	Position du curseur	Implicite	Non	Oui	Oui
	Extrémité "cible" de la corde principale et mini-cordes	Cible désignée et voisines	Implicite	Non	Non	Oui
	Mouvement de la 1 ^{ère} mini-rope	Distance à la prochaine cible	Implicite	Non	Non	Oui
	Longueur de la corde principale	Distance à la cible désignée	Implicite	Non	Non	Oui

FIGURE 8.1 – Analyse de l'information fournie par cinq des techniques d'expansion de cibles de l'expérience 2. L'information catégorisée comme secondaire durant la phase de transfert est en gras. Partie répétée de la figure 7.6

Pour interpréter les résultats de l'expérience 2, nous définissons deux hypothèses pour chaque phase orientée-but, sauf la phase de départ. En effet, la position de chaque cible est facilement prédictible par l'utilisateur tout au long de l'expérience car la séquence des positions des cibles est fixe : la cible suivante est toujours *quasiment* dans la direction opposée à celle du geste en cours. La phase de départ consiste alors à vérifier la fin de la tâche précédente, que ce soit une erreur ou une bonne sélection, et à initier le geste balistique dans la direction presque opposée. Dans cette configuration, la prise d'information *primaire* par l'utilisateur avant d'initier le sous-mouvement balistique est optionnelle car l'information *primaire* est déjà connue et mémorisée lors des essais précédents et lors des blocs d'entraînement permettant aux utilisateurs de se familiariser avec les différentes techniques. La phase de départ n'est donc pas étudiée dans cette analyse.

D'après le modèle conceptuel de performance, le temps de transfert croît avec l'information *secondaire* fournie durant la phase de transfert et le temps de validation croît avec l'information *primaire* fournie durant la phase de validation. Nos hypothèses sont donc les prédictions du modèle conceptuel de performance pour ces phases du geste :

- H1 : VTE et TARGET seront les techniques les plus rapides dans la phase de transfert car elles fournissent moins d'informations *secondaires* à l'utilisateur durant cette phase qu'eVTE, Rope Cursor et Bubble Cursor (figure 8.1).
- H2 : Rope Cursor et Bubble Cursor seront les techniques les plus lentes dans la phase de transfert car elles fournissent plus d'informations *secondaires* à l'utilisateur durant cette phase qu'eVTE, VTE et TARGET (figure 8.1).
- H3 : VTE et TARGET seront les techniques les plus rapides dans la phase de validation car elles fournissent moins d'informations *primaires* à l'utilisateur qu'eVTE, Rope Cursor et Bubble Cursor durant cette phase (figure 8.1).
- H4 : Rope Cursor et Bubble Cursor seront les techniques les plus lentes dans la phase de validation car elles fournissent plus d'informations *primaires* à l'utilisateur durant cette phase qu'eVTE, VTE et TARGET (figure 8.1).

8.1.2 Analyse des mouvements

Les données expérimentales de l'expérience 2 sont analysées de la façon suivante. Les trois phases orientées-but sont détectées au cours de chaque essai. Comme elles sont consécutives, la fin d'une phase signifie le début de la suivante :

- La phase de départ commence lorsque la cible précédente a été sélectionnée. La nouvelle cible apparaît alors.
- La phase de départ termine quand le curseur quitte la cellule de Voronoï de départ (taille étendue de la cible de départ). Cette approximation du début du sous-mouvement balistique nous permet d'éviter le problème des gestes irréguliers, comme ceux contenant plusieurs pics de vitesse successifs en début de geste par exemple. Ce type de gestes est courant avec le Virtual Pointer, très sensible aux mouvements de la main.
- La phase de transfert se termine quand le curseur entre dans la cellule de Voronoï de la cible-but pour la dernière fois.
- La phase de validation se termine quand la cible est sélectionnée. La phase de départ de la tâche suivante commence alors.

Afin de détecter les sous-mouvements balistiques et correctifs de façon cinématique, la vitesse du curseur considérée est la moyenne de 5 valeurs consécutives de la vitesse instantanée (calculée grâce à deux positions successives du curseur). Le profil de vitesse moyenne ainsi obtenu est ensuite lissé avec un filtre à noyau gaussien ($\sigma = 2$). Dans le profil lissé de vitesse moyenne ainsi obtenu, les minima locaux sont utilisés pour définir le début des différents sous-mouvements.

8.1.3 Résultats et discussion

Dans cette section, les résultats de l'expérience 2 sont présentés et discutés à la lumière du modèle de performance présenté précédemment au chapitre 7.

8.1.3.1 Méthode

Nous analysons les données expérimentales avec la même méthode que celle présentée au chapitre 6 section 6.2.4.1. Les essais incorrects, c'est-à-dire ceux contenant une erreur, sont retirés des données pour l'analyse des durées des phases orientées-but. Les temps de sélection, les durées des phases orientées-but et les nombres de sous-mouvements sont comparés statistiquement entre les techniques d'expansion de cibles grâce à des tests t appariés utilisant la méthode de Holm-Bonferroni. Aux figures 8.2 et 8.3, les résultats présentés sont ordonnés de la gauche vers la droite en ordre croissant. Dans cette section, nous utilisons la nomenclature suivante pour traduire la significativité des tests statistiques : *ns* signifie $p > 0,05$; * signifie $p < 0,05$; ** signifie $p < 0,01$; *** signifie $p < 0,0001$.

8.1.3.2 Phase de transfert

Concernant la phase de transfert, les résultats observés confirment l'hypothèse H1 (figure 8.2 à gauche) : VTE et TARGET sont les techniques les plus rapides (pour VTE : * avec TARGET et Bubble Cursor, ** avec Rope Cursor et *** avec eVTE ; et pour TARGET : *ns* avec Bubble Cursor et Rope Cursor, et *** avec eVTE).

VTE est significativement plus rapide que TARGET * car VTE bénéficie de l'*effet d'optimisation du geste balistique* grâce à l'information *explicite* et *primaire* qu'elle fournit durant la phase de départ. En effet, la grande majorité des gestes de pointage de cette expérience contiennent un sous-mouvement balistique et pas de sous-mouvements correctifs (figure 8.2 à droite). Cet effet d'optimisation s'exprime donc pleinement dans les tâches de pointage de cette expérience pour VTE et eVTE, les seules techniques fournissant des informations *explicites* et *primaires* durant la phase de départ.

L'hypothèse H2 n'est pas vérifiée car eVTE obtient un temps moyen de transfert significativement plus long que Rope Cursor ** et Bubble Cursor ***, alors qu'eVTE profite, contrairement à Rope Cursor et Bubble Cursor, de l'*effet d'optimisation du geste balistique*. Ces résultats sont contradictoires avec le modèle de performance et suggèrent que d'autres effets dégradent les performances d'eVTE. Nous proposons deux façons de l'expliquer :

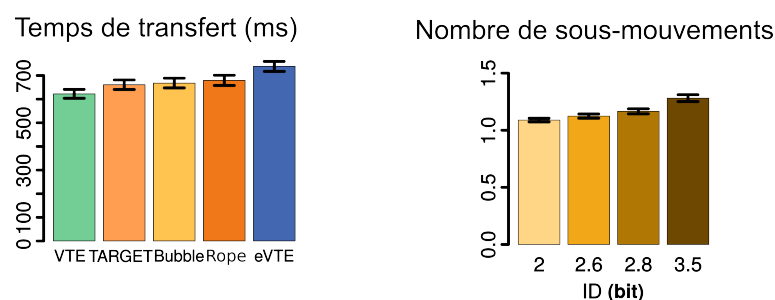


FIGURE 8.2 – Résultats quantitatifs de l'expérience 2 - Phase de transfert. À gauche, le temps de transfert par technique. À droite, le nombre de sous-mouvements par *ID*.

- Avec eVTE, la transparence du diagramme de Voronoï est liée à la vitesse du curseur. Or, ce diagramme contient de l'information *primaire* pour toutes les phases du geste. Il est donc possible que la disparition du diagramme lors de l'accélération du geste balistique provoque chez l'utilisateur un sentiment de retenue et de précaution, qui se concrétise par un ralentissement du geste dans le but de conserver l'affichage de l'information *primaire*.
- Cette expérience a permis de constater au chapitre 6 section 6.2.4.1 que la dynamicité des aides visuelles a un effet négatif sur le temps de sélection. Il est possible que la dynamicité *basée sur la vitesse du curseur* ait un impact négatif très important, eVTE étant la seule technique dans ce cas. Cette hypothèse est à mettre en relation avec les résultats de l'expérience de Su et al. comparant les aides visuelles de la technique Implicit Fan Cursor [126]. Cette expérience, ré-interprétée avec le modèle de performance au chapitre précédent section 7.5, montre que l'aide visuelle la moins efficace est aussi celle qui utilise une dynamicité *basée sur la vitesse du curseur* : le curseur-éventail (l'angle d'ouverture de l'éventail dépend de la vitesse du curseur).

8.1.3.3 Phase de validation

Les résultats concernant la phase de validation confirment l'hypothèse H4 mais ne confirment pas l'hypothèse H3 (figure 8.3). En effet, Rope Cursor et Bubble Cursor sont les techniques les plus lentes pour la validation (pour Rope Cursor et Bubble Cursor : *** avec VTE et eVTE, ** avec TARGET). La différence entre elles n'est pas significative. En revanche, TARGET n'est pas plus rapide qu'eVTE (figure 8.3), contrairement aux prédictions du modèle conceptuel de performance (H3). En effet, VTE et eVTE sont significativement les techniques les plus rapides lors de cette phase (pour VTE et eVTE : ** avec TARGET et *** avec Rope Cursor et Bubble Cursor).

Nous expliquons cette performance non prédite d'eVTE par un effet d'anticipation présent aussi avec VTE. VTE et eVTE, deux techniques *explicit*es affichent la cellule de la cible durant tout le geste, y compris la phase de transfert. En conséquence, l'utilisateur voit le curseur s'approcher de la cellule de la cible et du bord à franchir pour y entrer. Connaissant les caractéristiques de son propre mouvement de pointage, l'utilisateur peut alors se préparer à valider la sélection dès l'entrée du curseur dans la cellule visée. Ce mécanisme d'anticipation de l'action de validation n'est pas fourni avec les techniques

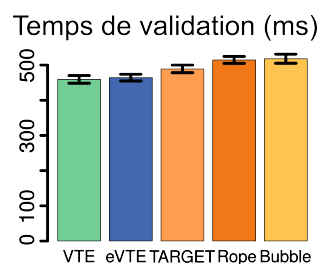


FIGURE 8.3 – Temps de validation par technique.

implicites, qui n'affichent pas les bords de la cellule de la cible. En revanche, certaines techniques *implicites* fournissent également des moyens d'anticiper l'entrée du curseur dans la cellule de la cible, comme Rope Cursor et Bubble Cursor, avec respectivement la longueur de la corde (*rope*) principale et le mouvement du bord de la bulle de Bubble Cursor, qui indiquent tous deux la distance restante jusqu'à la prochaine cible. Cependant, ces mécanismes fournissent l'information adéquate au dernier moment du geste et de façon très dynamique, contrairement au diagramme de Voronoï de VTE et eVTE. Il est possible que ces dernières caractéristiques en fassent des mécanismes d'anticipation moins efficaces ou plus difficile d'utilisation que celui fourni par VTE et eVTE. De plus, le ray-casting permet à l'utilisateur de paralléliser les deux phases, le transfert du curseur et l'action de validation [143]. Avec le Virtual Pointer et la pédale médicale, les actions de transfert et de validation ne sont pas réalisées avec le même membre de l'utilisateur. Cela montre que les phases orientées-but d'un geste peuvent être parallélisées, particulièrement en pointage à distance. Dans ces conditions, les utilisateurs peuvent pleinement profiter des mécanismes d'anticipation de l'action de validation que fournissent les techniques. Le modèle conceptuel de performance ne modélise pas les effets dus aux interactions entre les phases d'un même geste. En effet, il considère que les phases orientées-buts sont séquentielles et indépendantes. Enrichir le modèle pour prendre en compte les dépendances entre les phases constitue un axe de recherche qui fait suite à ces travaux et que nous décrivons dans le chapitre 9.3.4.

8.1.4 Conclusion de l'analyse de l'expérience 2

L'analyse de l'expérience 2 et la relecture des données expérimentales par le prisme des prédictions du modèle ont permis de constater que les résultats expérimentaux soutiennent le modèle conceptuel de performance. En effet, les prédictions du modèle sont globalement confirmées par les mesures de performance, ce qui apporte un élément de validité externe à ce modèle.

Cette nouvelle analyse des résultats expérimentaux par phase orientée-but permet également de mettre en exergue trois effets : l'impact de l'*effet d'optimisation du geste balistique*, le ralentissement d'eVTE en phase de transfert ainsi que les mécanismes d'anticipation permettant aux utilisateurs de paralléliser les actions de transfert et de validation. Tous ces effets ne sont actuellement pas pris en compte par le modèle conceptuel de performance, qui se base sur des phases orientées-but successives et indépendantes en ne considérant que l'information *primaire* ou *secondaire* pour chaque phase. Enrichir ce modèle pour prendre en compte ces effets fait partie de nos perspectives de recherche du chapitre 9.3.4.

Dans la section suivante, une nouvelle expérience dédiée à l'évaluation du modèle est présentée. Contrairement à l'expérience 2, celle-ci est réalisée dans des conditions de bureau proches d'une utilisation classique d'un ordinateur.

Aide Visuelle	Source d'Information	Information	Observabilité de l'Expansion	Primaire pour la Phase de Départ	Primaire pour la Phase de Transfert	Primaire pour la Phase de Validation
Bubble Cursor	Position de la bulle	Position du curseur	Implicite	Non	Oui	Oui
	Contact de la bulle	Cible désignée et voisines	Implicite	Non	Non	Oui
	Mouvement du bord de la bulle	Distance à la prochaine cible	Implicite	Non	Non	Oui
	Rayon de la bulle	Distance à la cible désignée	Implicite	Non	Non	Oui
TARGET	Forme agrandie de la cible désignée	Forme agrandie de la cible désignée	Implicite	Non	Non	Non
	Agrandissement d'une cible	Cible désignée	Implicite	Non	Oui	Oui
Cell Painting	Forme étendue de la cible désignée	Forme étendue de la cible désignée	Explicite	Non	Non	Oui
	Coloration d'une cellule	Cible désignée	Implicite	Non	Non	Oui
VTE	Forme étendue de la cible-objectif	Forme étendue de la cible-objectif	Explicite	Oui	Oui	Oui
	Forme étendue des distracteurs	Forme étendue des distracteurs	Explicite	Non	Non	Non
Expansion Lens	Position de la lentille	Position du curseur	Implicite	Non	Oui	Oui
	Contenu de la lentille	Forme étendue de certaines cibles	Explicite	Non	Non	Oui

FIGURE 8.4 – Analyse de l'information fournie par cinq des techniques d'expansion de cibles de l'expérience 3. L'information catégorisée comme secondaire durant la phase de transfert est en gras. Partie répétée de la figure 7.6

8.2 Expérience 3

8.2.1 Introduction

Une expérience a été réalisée [60] dans le but principal d'évaluer le modèle conceptuel de performance présenté au chapitre précédent (chapitre 7). Cette expérience a cependant trois objectifs :

1. Le premier objectif est de confronter notre modèle de performance à l'expérience.
2. Le second objectif concerne l'assertion issue de l'expérience 2 (chapitre 6) complémentaire au modèle de performance pour faire une prédiction sur les taux d'erreurs : "Les techniques *explicites* sont moins propices aux erreurs que les techniques *implicites*." Cette assertion a été vérifiée expérimentalement pour des tâches de pointage à distance de bas Indice de Difficulté (*ID*) (figure 8.2 à droite). Dans cette expérience, nous la ré-évaluons pour des tâches de pointage dans une configuration de bureau et à haut *ID*.
3. Le troisième est d'évaluer Expansion Lens, une nouvelle technique présentée au chapitre 5 section 5.2.1.

Les techniques choisies pour cette expérience répondent à ces objectifs. En effet comme le montre la figure 8.4, Expansion Lens, Cell Painting et VTE sont des techniques *explicites*, tandis que TARGET et Bubble Cursor sont des techniques *implicites* (deuxième objectif). De plus, avec cet ensemble de techniques, nous pourrions observer les effets de l'information *primaire* et *secondaire* sur toutes les phases orientées-but du geste de pointage, y compris la phase de départ. Cela signifie que toutes les valeurs possibles dans les trois colonnes de droite de la figure 8.4 sont couvertes par les techniques choisies.

Un objectif secondaire de l'expérience est de vérifier que la phase de transfert est la plus longue des trois phases orientées-but, constat que nous avons fait de façon informelle lors de l'analyse de l'expérience 2.

Les hypothèses de l'expérience font écho à ses objectifs. Le modèle de performance est évalué en première intention par ses prédictions pour chaque phase orientée-but :

- H1 : Pour la phase de départ, VTE sera la technique la plus lente (objectif 1). En effet, seule cette technique fournit de l'information *primaire* durant cette phase (figure 8.4).
- H2 : Pour la phase de transfert, Expansion Lens sera la technique la plus rapide (objectifs 1 et 3) car cette technique fournit l'information *secondaire* la plus limitée (figure 8.4) durant cette phase. En effet, TARGET présente une information *secondaire* via une aide visuelle *discrète* et *sur-cible* (chapitre 4). En attirant le regard par sa dynamisme (chapitre 6 section 6.2.6), cette aide visuelle ralentit cette phase du geste. VTE quant à elle présente une information *secondaire* (l'ensemble des tailles étendues des cibles) beaucoup plus riche qu'Expansion Lens, ce qui ralentit cette phase en fournissant plus d'information à analyser pour l'utilisateur.
- H3 : Pour la phase de validation, TARGET sera la technique la plus rapide (objectif 1). En effet, seule TARGET fournit une seule information *primaire* et *implicite*. Or, nos pré-tests suggèrent que l'information *explicite* est plus longue à analyser que l'information *implicite*.
- H4 : Le taux d'erreurs des techniques *explicites* sera inférieur à celui des techniques *implicites* (objectifs 2 et 3).

8.2.2 Protocole expérimental

8.2.2.1 Participants et appareillage

Vingt volontaires (8 femmes et 12 hommes), dont l'âge varie entre 22 et 49 ans (l'âge moyen des volontaires est de 28,8 ans), ont participé à l'expérience. Parmi eux, un(e) seul(e) est gaucher(ère). L'expérience est réalisée avec un ordinateur portable MSI GT60 doté d'un processeur Quadcore cadencé à 2,3 GHz, de Windows 7 et de la configuration par défaut de la souris. L'affichage est son écran, de dimension 15 pouces Full HD avec une résolution de 1920×1080 pixels. La souris utilisée est une Razer Abyssus de 2014. Le programme de test est écrit en Qt/C++.

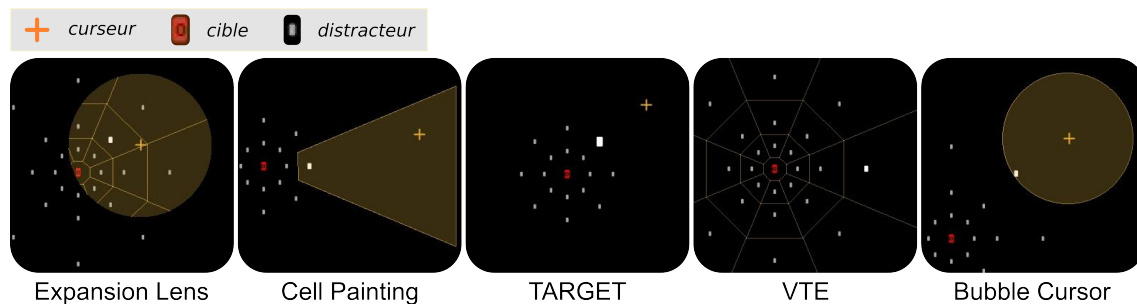


FIGURE 8.5 – Techniques d'expansion de cibles implémentées pour l'expérience 3.

8.2.2.2 Détail des techniques implémentées

Expansion Lens fait 200 pixels de diamètre (figure 8.5). Cell Painting utilise une couleur semi-transparente pour mettre en valeur la cellule de Voronoï survolée (figure 8.5). TARGET, VTE et Bubble Cursor ont été implémentées comme dans l'expérience 2 (chapitre 6).

8.2.2.3 Tâche et procédure

Afin d'étudier la phase de départ et particulièrement l'impact de l'information *primaire* sur le temps de départ, les participants ne doivent pas connaître *a priori* la position de la cible. Celle-ci ne doit pas être anticipée. Cette exigence diffère de plusieurs études [34, 53, 58, 59, 126] et des expériences 1 et 2 du chapitre 6.

Pour contraindre les participants à regarder la cible avant de commencer le geste, la tâche est une tâche de pointage ISO 9241-9 [70] modifiée de la façon suivante : huit cibles sont régulièrement distribuées le long d'un cercle (figure 8.6) ; Une cible de départ est au centre du cercle ; Avant toute sélection d'une autre cible, les participants sélectionnent la cible de départ ; L'ordre des huit cibles de chaque tâche ISO est aléatoire. Comme l'ordre des cibles est aléatoire, les participants ont besoin de localiser la cible avant de commencer leur geste de pointage.

Les cibles et les distracteurs ont une apparence très similaire : un rectangle gris de 8 pixels de haut et 4 pixels de large (figure 8.6). Le fond d'écran est noir. A tout instant, la cible désignée est blanche. La cible-objectif, que les participants doivent atteindre, est rouge, contient un zéro noir et est agrandie (1,5 fois plus grande que les autres) (figure 8.6). Le curseur est une petite croix jaune.

Les participants doivent réaliser chaque tâche le plus vite possible, tout en limitant leur taux d'erreurs aux 4 % nominaux (un essai incorrect pour 24 essais corrects). La sélection doit être correcte pour passer à l'essai suivant. Les participants utilisent les 5 techniques l'une après l'autre, l'ordre des techniques étant contrebalancé en carré latin. Pour chaque technique, un bloc d'entraînement de 8 cibles (une tâche ISO) permet aux participants de se familiariser avec la nouvelle technique. Après chaque technique, les participants

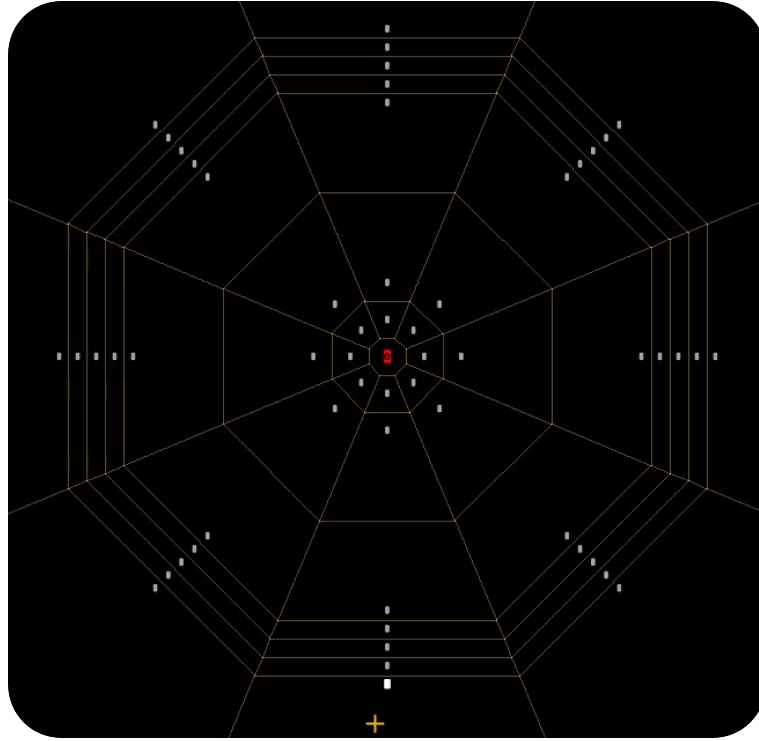


FIGURE 8.6 – Illustration des conditions expérimentales de l’expérience 3 avec l’agencement de cibles. La densité D des cibles est 1, la taille étendue des cibles est 16 pixels, l’amplitude des gestes est 252 pixels, la technique utilisée est VTE, le curseur est la petite croix jaune.

répondent à une enquête de type NASA-TLX simplifiée [63] (annexe D). Ensuite, ils classent les 5 techniques par ordre de préférence en disposant d’un résumé visuel sur papier des 5 techniques (une capture d’écran de l’expérience pour chaque technique). L’expérience dure environ 45 minutes.

8.2.2.4 Conditions

L’expérience respecte un schéma d’expérience intra-sujet de 5 techniques \times ($2 \times 2 \times 3$) agencements de cibles. Nous utilisons l’agencement de cibles créé pour l’expérience 2 (chapitre 6). Les différents agencements de cibles sont définis par trois paramètres :

1. deux amplitudes de mouvement A : 252 et 504 pixels,
2. deux largeurs de cellule de Voronoï W_X (taille étendue des cibles) : 8 et 16 pixels,
3. trois densités de distracteurs D : 0, 1 et 2.

De cette façon, les ID des tâches de pointage sont environ 4, 5 et 6 bits. Ces ID sont supérieures aux ID des tâches de pointage de l’expérience 2, qui sont entre 2 et 3,5 bits (figure 8.2). Ils répondent ainsi à l’objectif 2 de l’expérience.

Augmenter la densité de distracteurs D consiste à ajouter 3 cibles sur le chemin du curseur : 2 avant le cercle des cibles et une après (à la figure 8.6, $D = 1$). Les 12 agencements de cibles ($2 \times 2 \times 3$) sont présentés en ordre aléatoire pour chaque participant et pour chaque technique.

Quatre-vingt-seize sélections de cibles (12 agencements $\times 8$ sélections) par technique et par participant sont enregistrées. Cela représente un total de 9600 sélections enregistrées (96 sélections $\times 5$ techniques $\times 20$ participants).

8.2.3 Analyse des mouvements

Les gestes de pointage sont analysés de la même façon que pour l'analyse de l'expérience 2 précédemment présentée dans ce chapitre (section 8.1.2). Les trois phases orientées-but sont détectées au cours de chaque essai de la façon suivante :

- La phase de départ commence lorsque la cible de départ a été sélectionnée.
- La phase de départ termine quand le curseur quitte la cellule de Voronoï de départ (taille étendue de la cible de départ), qui mesure 32 pixels de large dans les 8 directions cardinales, car elle a la forme d'un octogone (figure 8.6).
- La phase de transfert se termine quand le curseur entre dans la cellule de Voronoï de la cible-objectif pour la dernière fois.
- La phase de validation se termine quand la cible est sélectionnée.

Les sous-mouvements balistiques et correctifs sont également détectés, de la même manière que pour l'analyse de l'expérience 2 (section 8.1.2).

8.2.4 Résultats et discussion

8.2.4.1 Méthode

Cent dix-sept essais anormaux (1,23 % des données), conséquences de doubles clics et de problèmes techniques, sont tout d'abord retirés des données. Les essais incorrects, c'est-à-dire ceux contenant une erreur, sont retirés des données pour l'analyse des temps de sélection et des durées des phases orientées-but. Les temps de sélection, les durées des phases orientées-but, les nombres de sous-mouvements et les réponses au questionnaire NASA-TLX sont comparés statistiquement entre les techniques d'expansion de cibles grâce des tests t appariés utilisant la méthode de Holm-Bonferroni. Les taux d'erreurs sont comparés, comme pour les expériences 1 et 2 (chapitre 6), grâce à des tests t appariés utilisant la méthode de Holm-Bonferroni après une agrégation par bloc de 8 essais. Les classements de préférence exprimés par les utilisateurs, qui sont beaucoup moins nombreux en termes d'individus statistiques, sont comparés grâce à des tests de Wilcoxon appariés.

De la figure 8.7 à la figure 8.9, les résultats présentés sont ordonnés de la gauche vers la droite en ordre croissant. Dans cette section, nous utilisons la nomenclature suivante pour traduire la significativité des tests statistiques : *ns* signifie $p > 0,05$; * signifie $p < 0,05$; ** signifie $p < 0,01$; *** signifie $p < 0,0001$.

Une ANOVA en mesures répétées à un facteur révèle un effet significatif de la technique d'expansion de cible utilisée sur le temps de sélection ($F_{4,9496} = 23,8$ ***). Dans cette section, les résultats de l'expérience sont présentés et discutés à la lumière du modèle de performance (chapitre 7).

8.2.4.2 Phase de départ

La phase de départ dure en moyenne 351 ms ($\sigma = 234$ ms).

VTE est la seule technique à fournir de l'information *primaire* durant la phase de départ (figure 8.4). En conséquence, sa phase de départ est plus longue que la phase de départ de toutes les autres techniques (*** avec toutes). Pour les autres techniques en revanche, aucune différence significative n'est observable dans la durée de la phase de départ. Ces résultats confirment l'hypothèse H1 et soutiennent le modèle de performance : *le temps de départ croît avec l'information primaire*. Cela est cependant à tempérer car une seule technique fournissant de l'information *primaire* durant cette phase est utilisée.

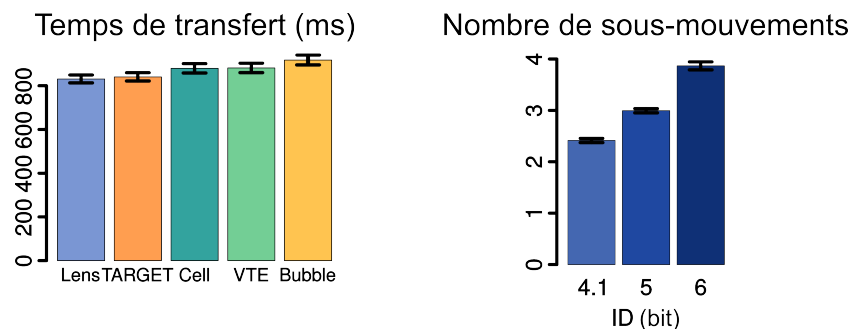


FIGURE 8.7 – Résultats quantitatifs de l'expérience 3 - Phase de transfert. À gauche, le temps de transfert par technique. À droite, le nombre de sous-mouvements par *ID*.

8.2.4.3 Phase de transfert

La phase de transfert dure en moyenne 900 ms ($\sigma = 497$ ms).

Expansion Lens et TARGET sont significativement les deux techniques les plus rapides dans la phase de transfert (figure 8.7 à gauche). Il n'y a pas de différence significative dans le temps de transfert entre Expansion Lens et TARGET et entre Cell Painting et VTE. Cependant, toutes les autres différences sont significatives. Ces résultats confirment l'hypothèse H2 et soutiennent le modèle de performance : *le temps de transfert croît avec l'information secondaire*. En effet, Bubble Cursor par exemple fournit dans cette phase plusieurs informations qui sont *secondaires* (figure 8.4, valeurs *Non* dans la colonne "Primaire pour la phase de transfert"). Elle est en conséquence la technique la plus lente dans cette phase. Le seul résultat contredisant le modèle concerne Cell Painting et VTE, mais la différence mesurée n'est pas significative. Quantifier numériquement l'information fournie par les techniques (par exemple en bits) permettrait une comparaison et une analyse plus fine de ces résultats. Ceci constitue un objectif une perspective à nos travaux décrite dans le chapitre 9.3.4.

Les résultats des expériences 1 et 2 du chapitre 6 concernant la technique VTE suggèrent que cette technique, la plus rapide lors de ces expériences, a également la phase de transfert la plus courte. Or, dans cette expérience, VTE n'obtient pas le temps de transfert le plus court (figure 8.7 à gauche). Ce fait s'explique de la façon suivante. Cette expérience est constituée de tâches de pointage d'Indice de Difficulté *ID* important (figure 8.7 à droite), contrairement à l'expérience 2 (figure 8.2 à droite). Par exemple, dans l'expérience 2, l'*ID* des tâches de pointage varie entre 2 et 3,5 et les gestes de pointage enregistrés ne contiennent dans leur grande majorité qu'un seul sous-mouvement, le sous-mouvement balistique. En comparaison, la plupart des essais enregistrés pour cette troisième expérience incluent plus de deux sous-mouvements (figure 8.7), conséquence directe de la plus grande difficulté des tâches de pointage. La différence observée dans les performances de VTE est due à l'*effet d'optimisation du geste balistique* : VTE en bénéficie lors des expériences précédentes grâce au faible *ID* des tâches de pointage. Mais, lors de tâches d'*ID* plus importants, cet effet est compensé et neutralisé par l'occurrence de plus d'un sous-mouvement correctif (figure 8.7 à droite). Dans cette expérience, le temps

de transfert de VTE est ainsi déterminé principalement par l'information *secondaire* fournie à l'utilisateur, en accord avec le modèle de performance.

8.2.4.4 Phase de validation

La phase de validation dure en moyenne 425 ms ($\sigma = 159$ ms).

Comme prédit par le modèle de performance, les temps de validation des différentes techniques sont très différents des temps de transfert (figure 8.8 à gauche). Les résultats confirment l'hypothèse H3 car TARGET est plus rapide dans cette phase que les autres techniques : *ns* avec VTE ($p = 0,09$), *** avec les autres techniques. Sauf pour Expansion Lens, les résultats soutiennent les prédictions du modèle de performance : *l'information primaire augmente le temps de validation*. Expansion Lens est néanmoins plus lente dans cette phase que Bubble Cursor alors que la prédiction était contraire. Cette différence n'est pas significative mais elle suggère que l'information *explicite* est plus longue à traiter que l'information *implicite*. La différence non-significative observée entre TARGET et VTE et les résultats de pré-tests renforcent cette hypothèse.

Les résultats montrent également qu'avec les techniques *explicites*, moins de sous-mouvements sont initiés dans la phase de validation, c'est-à-dire à l'intérieur de la cellule de la cible, qu'avec les techniques *implicites* (ce résultat n'est pas illustré dans ce mémoire). Ces différences ne sont pas significatives entre toutes les techniques cependant, il s'agit là d'un fort indice expérimental que *voir les bords de la zone visée permet d'être plus confiant pour cliquer* et ainsi, d'éviter des sous-mouvements correctifs additionnels.

Les taux d'erreurs observés (figure 8.8 à droite) confirment H4, même si les différences ne sont généralement pas significatives. Expansion Lens est moins propice aux erreurs que Bubble Cursor (*). De plus, si l'on rassemble les techniques par leur aspect explicite ou implicite, le groupe des techniques explicites est significativement (**) moins propice aux erreurs que le groupe des techniques implicites. Cela confirme le résultat de l'expérience 2 (chapitre 6) complémentaire au modèle pour prédire les taux d'erreurs et étend ce résultat aux tâches de pointage à *ID* important : les techniques *explicites* sont moins propices aux erreurs que les techniques *implicites*. Ce résultat nous permet d'atteindre le deuxième objectif de cette expérience.

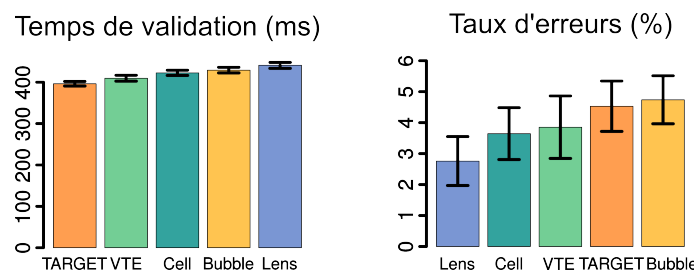


FIGURE 8.8 – Résultats quantitatifs de l'expérience 3 - Phase de validation. À gauche, le temps de validation par technique. À droite, le taux d'erreurs par technique.

8.2.4.5 Résultats globaux

Résultats quantitatifs Premièrement, ces résultats répondent à un objectif secondaire de cette expérience : la phase de transfert est plus longue que la phase de départ (***) et que la phase de validation (***), quelque soit la technique et l'*ID* de la tâche. C'est donc la phase qui a le plus d'influence sur la durée totale du geste. En conséquence, Expansion Lens et TARGET, les deux techniques les plus rapides dans la phase de transfert (figure 8.7 à gauche), sont également les deux techniques les plus rapides sur l'ensemble du geste (figure 8.9 à gauche). Leur différence n'est pas significative mais elles sont toutes deux significativement plus rapides que les autres techniques (***) avec Bubble Cursor et VTE, * pour Expansion Lens et ** pour TARGET avec Cell Painting). Cell Painting est également plus rapide que Bubble Cursor (*).

Expansion Lens étant la technique la moins propice aux erreurs (figure 8.8 à droite) et la deuxième plus rapide 8.9 à gauche), ces résultats répondent au deuxième objectif de cette expérience : Expansion Lens est une technique d'expansion de cibles offrant un bon compromis entre vitesse et précision.

Loi de Fitts La figure 8.10 représente le temps total de sélection (ensemble du geste) de chaque technique comme une fonction affine de l'*ID*, fonction dont l'ordonnée à l'origine et la pente sont indiquées à la figure 8.11. Ces droites ont été obtenues par régression linéaire, en appliquant la loi de Fitts à nos conditions expérimentales (variables indépendantes A et W_X). Pour toutes les techniques utilisées, les données sont bien modélisées par la loi de Fitts : les valeurs de R^2 ajusté pour l'*ID* sont acceptables (figure 8.11). Cependant, la qualité de cette modélisation est limitée par la négativité du coefficient a (ordonnée à l'origine) des droites de régression. En effet, ce facteur représente physiquement le temps de réaction de l'utilisateur, temps qui lui est nécessaire pour effectuer le pointage lorsque l'*ID* est de 0 (Amplitude du geste nulle et taille de la cible infinie). Cette modélisation suggère des temps de réaction négatifs, ce qui est physiquement impossible.

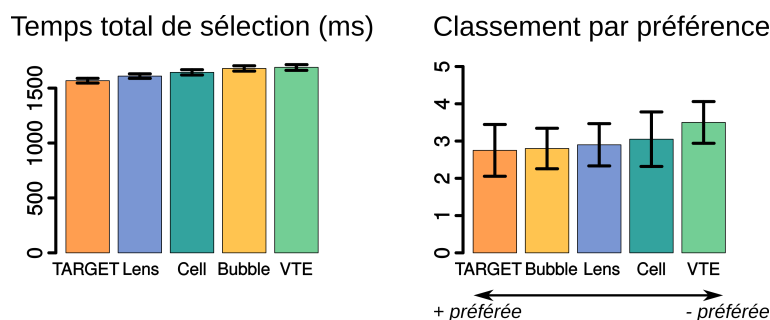
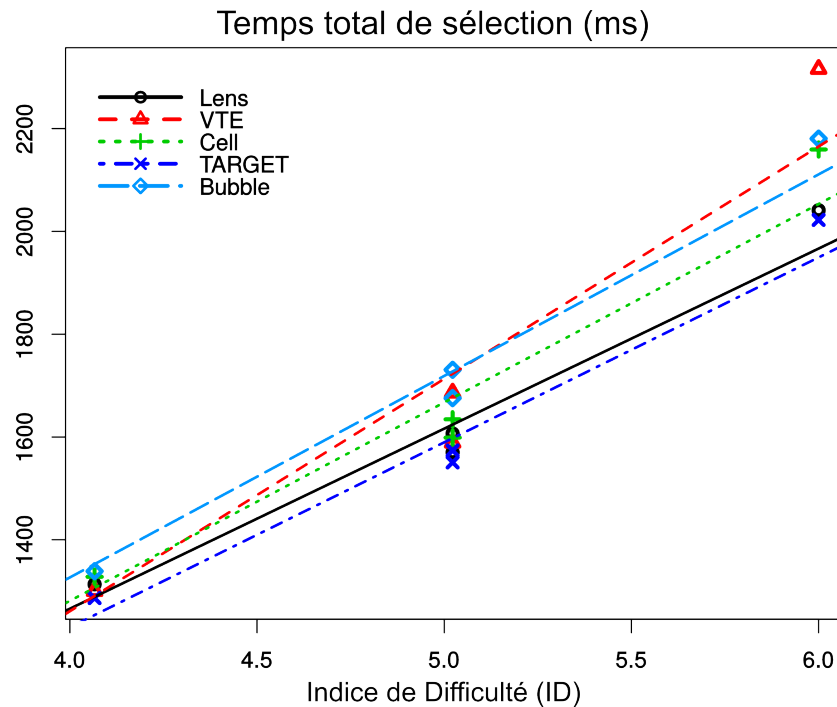


FIGURE 8.9 – Résultats quantitatifs globaux de l'expérience 3. À gauche, le temps total de sélection par technique. À droite, le classement par préférence par technique.

FIGURE 8.10 – Temps total de sélection par technique et par ID .

La pente importante de VTE (colonne b à la figure 8.11) illustre bien comment la rapidité de cette technique se dégrade lorsque l' ID de la tâche augmente et qu'elle ne bénéficie plus de l'*effet d'optimisation du geste balistique*. Les pentes de TARGET et Expansion Lens, les deux techniques les plus rapides, sont proches. La différence entre leurs temps totaux de sélection est relativement constante pour chaque ID , mais elle diminue légèrement lorsque l' ID augmente. Le modèle de performance permet d'expliquer cela de la façon suivante. Expansion Lens fournit durant la phase de validation plus d'information *primaire* que TARGET (figure 8.4). Cela crée une différence de temps de validation constante entre les deux techniques (figure 8.8 à gauche), même lorsque l' ID augmente. Cependant, la phase de transfert représente une part de plus en plus importante du geste à mesure que l' ID augmente. Leur différence de temps de validation a donc de moins en moins de poids sur le temps total de sélection lorsque l' ID augmente et les performances des deux techniques se rapprochent. TARGET a donc une pente légèrement plus forte qu'Expansion Lens dans cette modélisation avec la loi de Fitts.

Technique	$T = a + b.ID$		
	$a [ms]$	$b [ms/bit]$	Adj. R^2
Expansion Lens	-135	350	0.94
TARGET	-209	360	0.95
Cell Painting	-263	386	0.93
Bubble Cursor	-242	392	0.96
VTE	-547	452	0.91

FIGURE 8.11 – Expérience 3 : Coefficients de régression linéaire du temps total de sélection (T) par l' ID , calculé avec la taille étendue des cibles (W_X) : Ordonnée à l'origine (a), pente (b) et R^2 ajusté (colonne de droite).

Résultats qualitatifs Aucune différence significative entre les techniques n'existe dans le classement par ordre de préférence (figure 8.9 à droite). Cependant, TARGET est la technique préférée, suivie dans l'ordre par Bubble Cursor, Expansion Lens et Cell Painting. VTE a obtenu le score de préférence le plus faible.

Selon les résultats des enquêtes de type NASA-TLX simplifiées, Expansion Lens est classée la meilleure technique par les utilisateurs en :

- *Exigence Mentale*,
- *Exigence Temporelle*,
- *Effort* et
- *Frustration*.

Elle est également classée seconde en *Performance* et en *Exigence physique*. Bien que les différences entre les techniques ne soient pas significatives pour ces classements, cela indique un bon accueil de la technique par les utilisateurs.

8.2.5 Conclusion de l'expérience 3

Nous rappelons que les trois objectifs principaux de cette expérience sont :

1. mener une première évaluation du modèle de performance présenté au chapitre précédent (chapitre 7),
2. montrer que les techniques *explicites* sont moins propices aux erreurs que les techniques *implicites* pour des tâches de pointage à Indice de Difficulté (*ID*) élevé réalisées dans une configuration de bureau, et
3. évaluer Expansion Lens, une technique d'expansion de cibles présentée au chapitre 5 (section 5.2.1),

Ces trois buts sont atteints. Les trois prédictions du modèle de performance (hypothèses H1, H2 et H3 : une par phase orientée-but) ont été vérifiées par l'expérience, ce qui apporte un élément de validité externe supplémentaire au modèle prédictif de performance.

Expansion Lens s'est révélée être une technique d'expansion de cibles présentant un bon compromis entre vitesse d'utilisation et précision de la sélection, tout en évitant le problème de la surcharge visuelle que présentent VTE, eVTE et les techniques combinées présentées au chapitre 5 réduisant cette surcharge par des moyens différents. Ce résultat soutient par ailleurs le rationnel de conception d'Expansion Lens (chapitre 5).

Les résultats de l'expérience ont montré que les techniques *explicites* sont moins propices aux erreurs que les techniques *implicites* pour des tâches de pointage à *ID* élevé réalisées dans une configuration de bureau. Ces résultats complètent les résultats de l'expérience 2 du chapitre 6, valides pour des tâches de pointage à distance à faible *ID*.

Enfin, un objectif secondaire de cette expérience est de vérifier que la phase de transfert est la plus longue des trois phases orientées-but d'un geste de pointage. Les résultats montrent que la phase de transfert est la plus longue des trois phases orientées-but dans ces conditions expérimentales. Compte-tenu du modèle de l'impulsion initiale optimisée (chapitre 2) [88], ce résultat est probablement valable pour toutes les tâches de pointage à partir d'un certain niveau de difficulté de la tâche. La véracité de cette assertion reste cependant à démontrer pour des conditions expérimentales différentes.

Conclusion

L'analyse de deux expériences [59, 60] présentées dans ce chapitre apporte une première validation expérimentale du modèle conceptuel de performance présenté au chapitre précédent. Elles permettent également de constater les limites de ce modèle, qui ne prend pas en compte plusieurs phénomènes observés dans les résultats expérimentaux. Le modèle évalué par ces études expérimentales reste une première tentative de modélisation des performances qui peut ainsi être améliorée de plusieurs façons.

Cette première confrontation expérimentale du modèle de performance est basée sur une approximation permettant de quantifier l'information fournie par les techniques : nous considérons le nombre d'informations distinctes identifiées lors de l'analyse des aides visuelles. Ce premier pas pour quantifier l'information fournie par les techniques a permis de faire des prédictions quant aux résultats expérimentaux. Ces derniers montrent néanmoins les limites de cette approximation et questionnent sur le traitement de l'information par les utilisateurs lors d'un geste de pointage. L'utilisateur a-t-il besoin de plus de temps pour traiter l'information *explicite* que pour traiter l'information *implicite* ? Quel est l'impact de la dynamique d'une source d'information sur le temps de traitement de l'information qu'elle fournit ?

Notre travail de recherche nous a conduit à poursuivre ce travail de quantification. Ceci consiste à enrichir le modèle pour produire des prédictions numériques du temps de sélection d'une cible pour des conditions données et pour une technique d'expansion de cibles donnée, à l'image de la loi de Fitts [79] ou du Modèle de l'Impulsion Initiale Optimisée [88] (chapitre 2). Nous présentons au chapitre 9.3.4 les premières pistes entreprises.

Quatrième partie

Domaine applicatif : la chirurgie augmentée

Chapitre 9

Techniques d'interaction avec OrthoPilot® d'Aesculap

Sommaire

Introduction	168
9.1 Interviews de deux chirurgiens	169
9.1.1 Méthode	169
9.1.1.1 Première phase : Mise en situation	169
Logiciel de test et techniques implémentées	169
Exploration par axe de conception	172
Procédure	172
9.1.1.2 Deuxième phase : Co-conception de techniques	173
Matériel et conditions	173
Procédure	174
9.1.2 Résultats des interviews	174
9.1.2.1 Résultats des mises en situation	174
Géométrie	175
Axe <i>Dynamicité</i>	176
Axe <i>Observabilité de l'expansion</i>	176
Axe <i>Élément augmenté</i>	176
9.1.2.2 Résultats des sessions de co-conception	177
Améliorer TARGET	177
Aides visuelles <i>continues</i>	177
9.1.3 Conclusion des interviews	178
9.2 Total Knee Arthroplasty (TKA) version 6	178
9.3 Medical TapTap	180
9.3.1 Existant : Pédale médicale	181
9.3.2 Interaction gestuelle podale	182
9.3.2.1 Bracelet Myo	182
9.3.2.2 Interaction gestuelle podale	183
9.3.3 <i>Medical TapTap</i>	186

9.3.3.1	Vocabulaire gestuel	186
9.3.3.2	Détection des gestes	187
	Analyse des signaux inertiels	187
	Apprentissage automatique	188
	Détection des gestes	188
9.3.4	Conclusion	189
Conclusion		189

Introduction

Dans ce chapitre, nous présentons le travail réalisé chez Aesculap auprès de chirurgiens-partenaires, qui a permis l'intégration de nos résultats de recherche au logiciel TKA 6 (sortie prévue courant 2018). Pour le transfert de nos travaux au sein des logiciels d'Aesculap, la prise en compte des spécificités du contexte applicatif est nécessaire. Notre analyse de ce contexte nous a conduit à l'étude des techniques d'expansion de cibles. En effet, les techniques d'expansion de cibles ne modifient pas la trajectoire du curseur et le chirurgien en garde un contrôle direct et exclusif. Comme expliqué au chapitre 1, le sentiment de contrôle est important. De plus, l'écran de planification de TKA 5 contient de l'espace entre les zones d'interaction, condition nécessaire aux techniques d'expansion de cibles. Cet espace peut donc être mis à profit pour étendre les cibles existantes et faciliter le pointage. Enfin, le principe de l'expansion de cibles est simple et n'ajoute pas de charge cognitive supplémentaire. Ce dernier point est crucial dans un contexte de charge cognitive et émotionnelle élevée des utilisateurs, les chirurgiens.

Afin de déterminer quelles techniques d'expansion de cibles seraient les mieux acceptées par les chirurgiens dans le contexte du logiciel TKA, nous avons présenté les différentes techniques et particulièrement les différentes aides visuelles explorées au cours des chapitres précédents (du chapitre 4 au chapitre 7) à des chirurgiens. Pour cela, nous avons interviewé en situation deux chirurgiens partenaires d'Aesculap : le docteur Panisset, orthopédiste spécialiste du genou et le docteur Bettega, orthopédiste spécialiste de la chirurgie maxillo-faciale. Ce faible nombre de participants à l'étude est dû à la disponibilité des chirurgiens, dont le temps est rare et précieux. Nous présentons dans ce chapitre ces interviews et le résultat final intégré au logiciel TKA 6 développé par Aesculap.

Dans un deuxième temps, nous présentons une nouvelle technique d'interaction pour valider une sélection dans le cas de la chirurgie augmentée. Cette technique utilise un vocabulaire de gestes de pieds détectés par analyse des données d'accélération, y compris gravitationnelle, captées par le bracelet Myo.

9.1 Interviews de deux chirurgiens

Afin de recueillir l'avis de chirurgiens, nous avons conçu une méthode d'interview en deux phases : une session d'évaluation subjective de techniques d'expansion de cibles et une session de co-conception de techniques d'expansion de cibles. Nous décrivons tout d'abord cette méthode et nous analysons ensuite les résultats des deux interviews réalisés par le spectre de l'espace de conception des techniques d'expansion de cibles décrit au chapitre 4.

9.1.1 Méthode

La méthode d'interview utilisée a été conçue pour les objectifs suivants :

1. Explorer systématiquement, et non pas individuellement, les techniques d'expansion de cibles avec les chirurgiens.
2. Extraire des recommandations pour la conception de telles techniques pour la chirurgie augmentée.
3. Explorer la conception directe par l'utilisateur de techniques d'expansion de cibles.

Ces objectifs ont structuré les interviews en deux phases distinctes. La première met les utilisateurs en situation pseudo-réelle, c'est-à-dire en laboratoire mais avec des conditions ressemblant aux conditions chirurgicales réelles. L'entreprise Aesculap dispose pour cela d'une salle avec une table similaire à une table d'opération et un OrthoPilot® complet, équipé de la technologie Virtual Pointer (chapitre 2 section 2.3.1) qui permet de réaliser des tâches de pointage à distance. Cette installation ne reproduit pas les conditions de stress mental et émotionnel, ni la présence du personnel de bloc. Il est cependant demandé aux chirurgiens de se "mettre en situation" mentalement et d'adopter au maximum la même attitude que lors d'une opération réelle. La seconde phase de l'interview est un travail de co-conception de techniques autour d'un bureau.

Afin d'explorer la conception de techniques d'expansion de cibles de façon systématique et non individuelle, et éviter ainsi une évaluation individuelle et indépendante d'un grand ensemble de techniques, nous basons les deux phases de l'interview sur l'espace de conception décrit au chapitre 4. Dans la section suivante, nous décrivons le protocole expérimental de ces deux phases.

9.1.1.1 Première phase : Mise en situation

Logiciel de test et techniques implémentées Seul l'écran de planification du logiciel TKA5 (figure 1.4 au chapitre 1 reprise en figure 9.1) est utilisé car c'est une interface graphique qui nécessite le plus d'actions de pointage. De plus, cet écran est très bien connu par l'un des deux chirurgiens. Cet écran est modifié de façon à permettre aux chirurgiens de passer d'une technique de pointage à l'autre (figure 9.1) : les deux boutons en bas de l'écran sont dédiés à cette fonction.



FIGURE 9.1 – Écran de planification du logiciel TKA 5 d'Aesculap, modifié pour les besoins des interviews. Les cibles sont en blanc, les nombres en gris ou vert ne sont pas des cibles : ce sont des valeurs calculées par le système en fonction des nombres en blanc. Le curseur est le point jaune entourée de 4 ailettes. Les boutons ronds en bas à droite et à gauche de l'écran permettent de naviguer entre les techniques d'un même jeu de techniques. Le bouton "+" central permet d'activer/désactiver l'expansion de cibles. La technique d'expansion utilisée est Target Highlight : un contour jaune autour du chiffre 1 proche du curseur indique la cible désignée.

Les techniques d'expansion de cibles décrites au chapitre 5 et évaluées au chapitre 6 sont implémentées dans la plate-forme logicielle C++/Qt de l'OrthoPilot® d'Aesculap (figure 9.2). À ces techniques, est ajoutée une technique mettant simplement en valeur la cible sélectionnée en colorant son contour en jaune. Cette technique constitue une aide visuelle minimale présente avec toutes les autres techniques et assure la cohérence de l'interface graphique. Elle est nommée "Target Highlight" dans la suite de la présentation de ces interviews et illustrée à la figure 9.1.

De plus, les résultats de l'expérience 2 (chapitre 6) suggèrent un bon potentiel pour la technique MTE (chapitre 5). Celle-ci utilise la tessellation de Voronoï *en distance de Manhattan* comme algorithme d'expansion, contrairement aux autres techniques présentées au chapitre 5, qui utilisent la tessellation de Voronoï *en distance euclidienne*. Cette technique a été la plus rapide lors de l'expérience 2 (chapitre 6). De plus, son rationnel de conception évoque la possibilité que cet algorithme d'expansion, qui produit des cellules de Voronoï constituées uniquement de lignes verticales, horizontales, ou à 45°, soit mieux accepté par les utilisateurs car les lignes du diagramme de Voronoï *en distance*

de *Manhattan* suivent la même direction que la plupart des lignes de force de l'interface graphique, comme les bords de l'écran, les bords des fenêtres, les menus ou encore l'alignement des textes et des icônes. Afin de tester l'acceptabilité de cet algorithme et de le comparer à l'algorithme initial *en distance euclidienne* indépendamment de la dynamicité des techniques étudiées, nous avons conçu deux techniques supplémentaires :

- eMTE est une adaptation directe d'eVTE à la tessellation *en distance de Manhattan*. La transparence du diagramme varie de la même façon qu'eVTE, seul l'algorithme d'expansion change.
- MCell Painting est une adaptation directe de Cell Painting à la tessellation *en distance de Manhattan*. La cellule de Voronoï survolée par le curseur est colorée de façon semi-transparente.

La création de ces deux techniques spécifiquement pour ces interviews font partie des possibilités d'extensions décrites au chapitre 5, section 5.4, démontrant ainsi par l'exemple la faisabilité des extensions proposées. Les techniques Rope Cursor et Bubble Cursor ont également été adaptées à la tessellation *en distance de Manhattan* mais leur étude est moins intéressante ici car la différence avec les techniques originelles *en distance euclidienne* est quasiment imperceptible puisque les cellules ne sont pas directement observables à l'écran.

À part MTE, eMTE et MCell Painting, toutes les autres techniques implémentées utilisent la tessellation de Voronoï *en distance euclidienne* comme algorithme d'expansion.

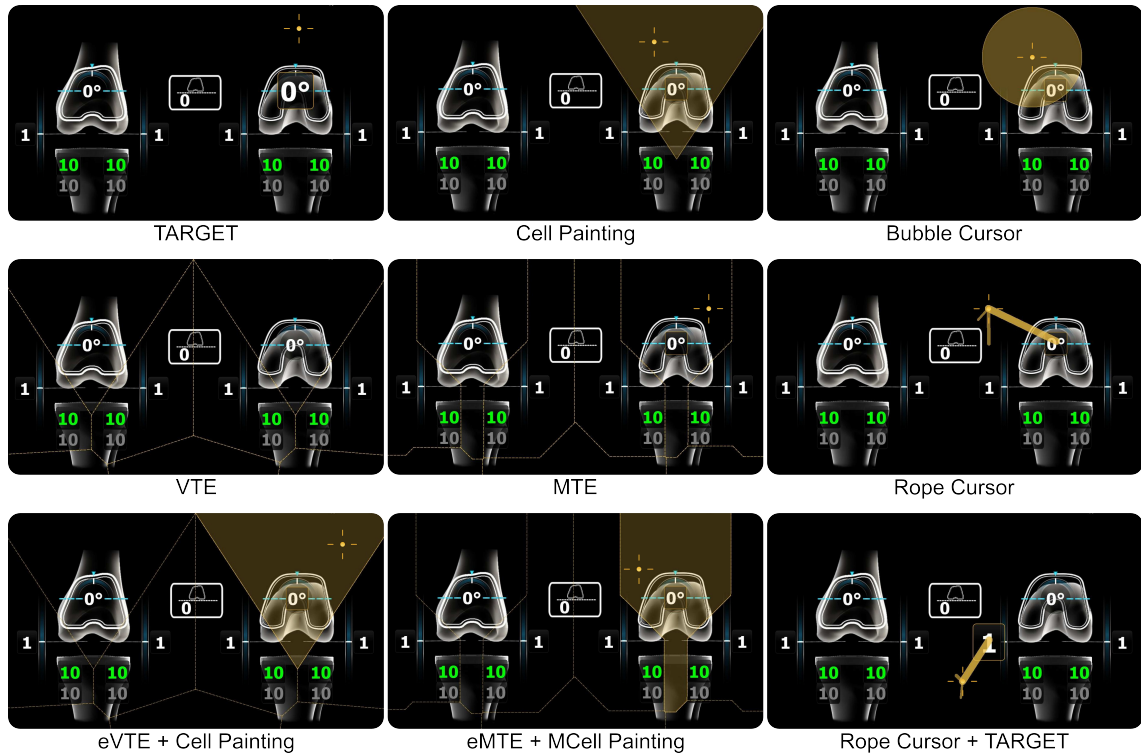


FIGURE 9.2 – Techniques d'expansion de cibles implémentées pour les interviews. La technique Target Highlight est illustrée à la figure 9.1.

Pour les techniques combinées, les combinaisons sont parallèles : aucune aide visuelle ne disparaît pendant le mouvement. Les chirurgiens utilisent les techniques à 2 mètres environ de l'OrthoPilot®. Ils utilisent le Virtual Pointer décrit au chapitre 2 et cliquent grâce à la pédale médicale de l'OrthoPilot®.

Exploration par axe de conception Pour structurer l'exploration des techniques grâce à l'espace de conception des techniques d'expansion de cibles, chaque axe est exploré indépendamment, mais sans rendre cette structure de l'exploration explicite à l'utilisateur. Un jeu de techniques est donc défini pour chaque axe (figure 9.2) :

- Jeu 1 : variations sur l'axe *Dynamacité* : VTE (statique), Cell Painting (discrète), VTE + Cell Painting (statique + discrète), eVTE (continue), eVTE + Cell Painting (continue + discrète).
- Jeu 2 : variations sur l'axe *Observabilité de l'expansion* : TARGET (implicite), Bubble Cursor (implicite à plusieurs informations), Rope Cursor (implicite à plusieurs informations), Cell Painting (explicite), VTE + Cell Painting (explicite + explicite).
- Jeu 3 : variations sur l'axe *Élément augmenté* : Target Highlight (sur-cibles), VTE + Cell Painting (sur-espace + sur-cibles), Bubble Cursor (sur-curseur), VTE (sur-espace), Rope Cursor (sur-curseur), TARGET (sur-cibles), VTE + TARGET (sur-espace + sur-cibles), Rope Cursor + TARGET (sur-curseur + sur-cibles).

Enfin, un quatrième jeu de techniques vise à comparer les tessellations de Voronoï *en distance euclidienne* et *en distance de Manhattan*, indépendamment de la dynamacité des techniques :

- Jeu 4 : variations sur l'algorithme d'expansion : MTE (manhattan + statique), VTE (euclide + statique), eMTE + MCell Painting (manhattan + discrète + continue), eVTE + Cell Painting (manhattan + discrète + continue)

L'ordre initial de présentation des jeux de techniques et des techniques de chaque jeu est inversé entre les deux chirurgiens, qui naviguent ensuite librement entre les techniques.

Procédure Le principe de l'expansion de cibles est décrit au chirurgien, ainsi que le fonctionnement du logiciel. Il est simplement précisé que différentes aides visuelles sont possibles pour une même expansion. Les quatre jeux de techniques sont présentés aux deux chirurgiens. Pour un jeu de techniques donné, les chirurgiens peuvent librement passer d'une aide visuelle à une autre autant de fois que nécessaire. La tâche consiste à utiliser chaque technique pour sélectionner au moins une fois toutes les cibles sélectionnables, c'est-à-dire les 7 nombres affichés en blanc à l'écran (figure 9.1).

Durant chaque jeu de techniques, les chirurgiens sont filmés et tous leurs commentaires ou expressions verbales sont consignées et utilisées comme données qualitatives. À la fin de chaque jeu de techniques, les chirurgiens classent les techniques du jeu par ordre de préférence en commentant leurs choix. Ils notent ensuite les techniques sur des échelles

de Likert en 5 points correspondant aux critères suivants : rapide, distrayant, efficace, fatiguant, facile, dérangeant, joli, intuitif, frustrant et satisfaisant. Cette première phase des interviews dure environ une heure.

9.1.1.2 Deuxième phase : Co-conception de techniques

Matériel et conditions La deuxième phase des interviews est consécutive à la première après une courte pause. C'est une phase de co-conception, qui a lieu autour d'un bureau. Les chirurgiens sont filmés et leurs commentaires rassemblés comme données qualitatives. Le matériel utilisé, illustré à la figure 9.3, est constitué d'un stylo et de feuilles de papier :

- Un schéma de l'espace de conception décrit au chapitre 4,
- un diagramme représentant un mouvement de pointage et ses phases cinématiques,
- un jeu de cartes représentant les aides visuelles atomiques utilisées durant la première phase de l'interview.

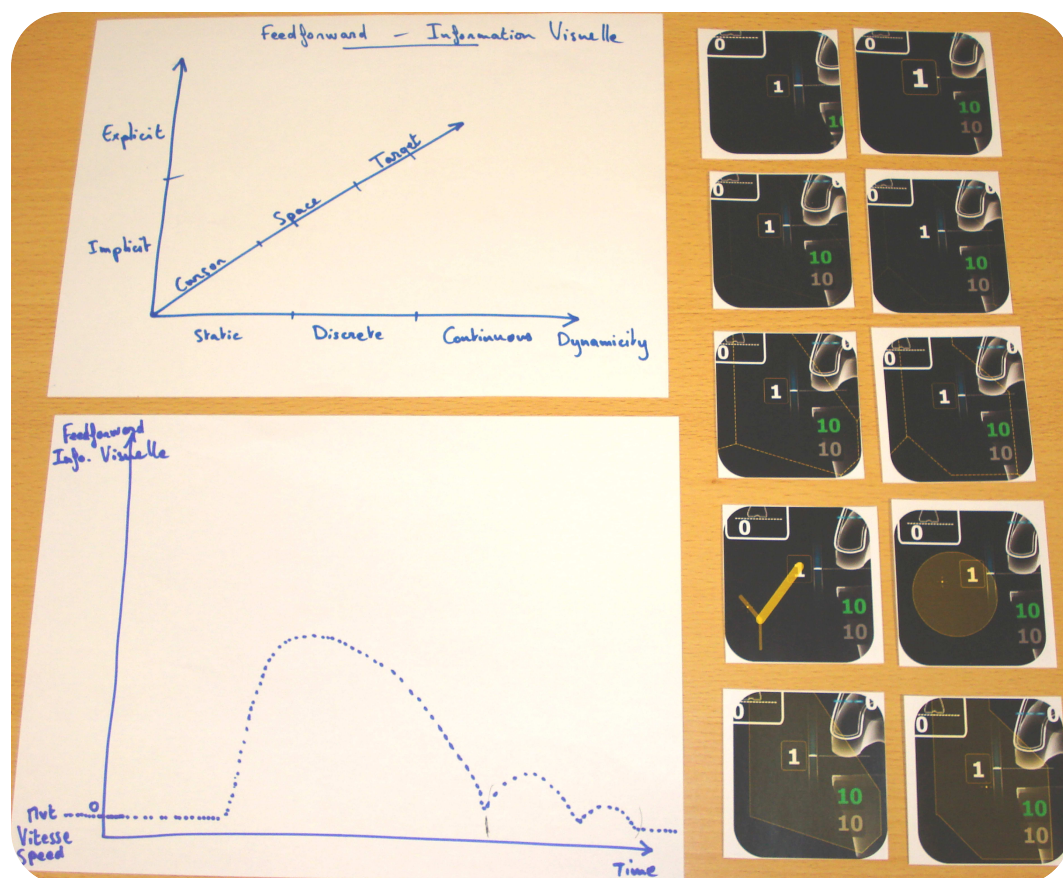


FIGURE 9.3 – Matériel papier pour la co-conception. En haut à gauche, le schéma de l'espace de conception. En bas à gauche, le diagramme d'un mouvement de pointage. À droite, les cartes des aides visuelles atomiques.

Procédure L'espace de conception et ses trois axes sont tout d'abord présentés et expliqués aux chirurgiens. Après cet exposé, la première tâche des chirurgiens est de proposer de nouvelles techniques d'expansion de cibles pertinentes pour leur contexte d'application. Pour cela et si les idées manquent, un point de départ suggéré est de considérer chaque axe pour faire évoluer leurs techniques d'expansion favorites. Le stylo, le schéma de l'espace de conception et les cartes, ainsi que des feuilles vierges de papier peuvent être utilisés pour créer de nouvelles techniques.

Ensuite, le mouvement de pointage et ses différentes phases cinématiques sont présentés et expliqués aux chirurgiens. La notion de combinaison des aides visuelles est également expliquée, détaillée (combinaisons *séquentielles* et/ou *parallèles*) et illustrée par certaines techniques manipulées durant la première phase de l'interview. La seconde tâche des chirurgiens est alors de proposer de nouvelles combinaisons d'aides visuelles pertinentes pour leur contexte d'application en plaçant les cartes sur le diagramme représentant les trois phases cinématiques du geste de pointage. Cette deuxième phase dure environ 30 minutes.

9.1.2 Résultats des interviews

Sont présentés dans cette section les résultats des interviews des deux chirurgiens orthopédistes séniors qui ont participé à l'étude. Dans les conclusions de l'expérience 2 (Chapitre 6 section 6.2 et 6.2.6), les techniques *statiques* et *explicites* sont les plus efficaces. Cependant, la mise en situation professionnelle, particulièrement pour ce profil d'utilisateurs, peut évidemment fournir des résultats différents de ceux obtenus en laboratoire avec des tâches abstraites.

9.1.2.1 Résultats des mises en situation

Afin de structurer notre analyse par rapport à l'espace de conception (chapitre 4), les résultats sont présentés et discutés par axe de conception. Les verbatim des chirurgiens sont cités EN PETITES MAJUSCULES.

Pour chaque technique atomique, chaque technique combinée et chaque valeur sur les axes de conception, un *classement moyen de préférence normalisé* et un *score moyen normalisé des échelles de Likert* sont calculés. Pour le *classement moyen de préférence normalisé*, les classements de préférences sont moyennés puis normalisés entre eux. Ainsi, la technique obtenant 1 (ou 100%) au *classement moyen normalisé de préférence* est la technique ayant été en moyenne préférée par les chirurgiens. Pour le *score moyen normalisé des échelles de Likert*, les différents critères choisis sont pondérés par 1 ou -1 selon leur aspect positif (comme "rapide") ou leur aspect négatif (comme "fatigant"). Les critères pondérés sont ensuite moyennés puis normalisés entre eux. La technique obtenant 1 (ou 100%) au *score moyen normalisé des échelles de Likert* est la technique ayant obtenu en moyenne les meilleures notes pour les critères positifs et les moins bonnes pour les

critères négatifs. Ces calculs permettent de dégager des données quantitatives pour des techniques qui ont parfois été notées plusieurs fois au cours des 4 jeux de techniques par chaque utilisateur, parfois de façon différente, et pour des ensembles de techniques (les techniques *statiques*, les techniques *implicites*, etc). Les résultats ainsi obtenus sont présentés à la figure 9.4.

Géométrie Lors du jeu 4, les chirurgiens ont préféré les cellules de Voronoï SIMPLES de la géométrie euclidienne aux cellules DÉFORMÉES et TORTUEUSES de la géométrie rectilinéaire. Les résultats quantitatifs calculés confirment cette tendance : les préférences et les scores obtenus par MTE et eMTE + MCell Painting sont respectivement inférieurs aux préférences et aux scores obtenus par VTE et eVTE + Cell Painting (figure 9.4).

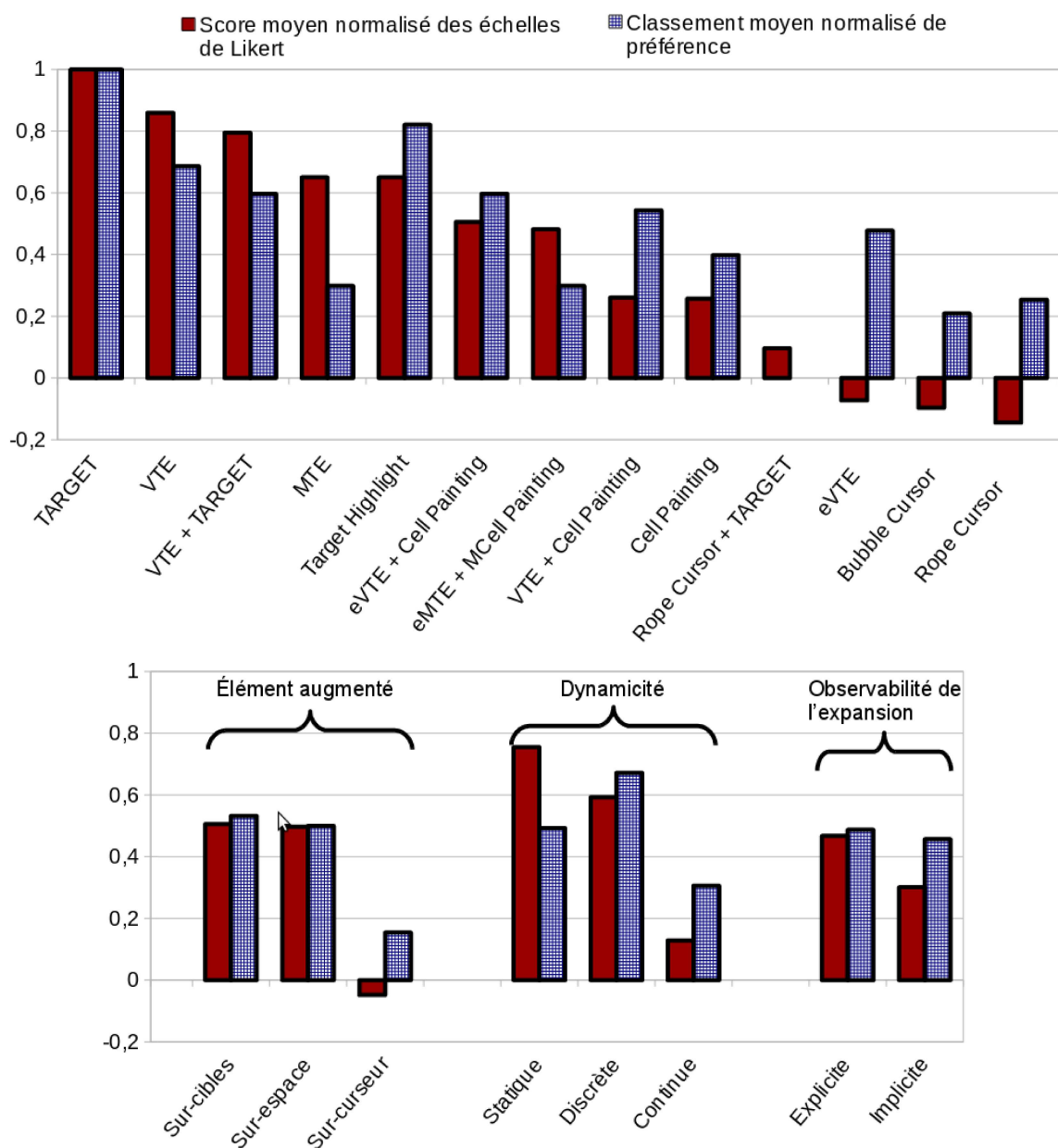


FIGURE 9.4 – Résultats quantitatifs des interviews.

Axe *Dynamacité* Les résultats et les commentaires des chirurgiens montrent que la complexité subjective d'une technique d'expansion augmente avec sa dynamacité : les aides visuelles *continues*, comme eVTE, Bubble Cursor et Rope Cursor, sont qualifiées de DÉSAGRÉABLES par les chirurgiens, à cause du BRUIT VISUEL qu'elles génèrent. Nous pensons que ce bruit visuel est dû aux mouvements continus et permanents de l'aide visuelle qui, entre deux cibles désignées, n'apportent aucune information nouvelle et utile. Vis-à-vis du modèle de performance présenté au chapitre 7, il s'agit d'information *secondaire* pendant la phase de transfert.

Lors du Jeu 1 de techniques, les deux chirurgiens ont trouvé eVTE distrayant et désagréable à utiliser car, selon leurs commentaires, la transparence du diagramme change constamment. Lorsque Bubble Cursor ou Rope Cursor sont apparus lors du Jeu 2, ils ont immédiatement exprimé leur surprise et leur rejet. Ils n'ont, de plus, pas trouvé de différence notable entre Bubble Cursor et Rope Cursor. Par contre, ils ont tous les deux trouvé ces techniques TROP COMPLIQUÉES et même agaçantes.

Les techniques *discrètes*, qualifiées d'UTILES ET LOGIQUES, furent beaucoup plus appréciées. TARGET est la technique préférée et la mieux classée (figure 9.4). Target Highlight est aussi bien appréciée et notée, ainsi que VTE + TARGET (figure 9.4). Un des deux chirurgiens trouve que VTE seule (non combinée) manque de QUELQUE CHOSE QUI CONFIRME LE CHANGEMENT DE CIBLE, en d'autres termes une aide visuelle *discrète* mettant en exergue la cible désignée. En revanche, VTE est décrit comme SIMPLE CAR STATIQUE.

Axe *Observabilité de l'expansion* Lors du Jeu 1, un chirurgien a préféré voir le diagramme de Voronoï complet plutôt qu'une seule cellule – IL MANQUE DES REPÈRES : il considérerait qu'il était crucial de voir toutes les tailles étendues des cibles. Lors du jeu 2, les deux chirurgiens ont cependant finalement préféré TARGET, une technique *implicite* ne rendant pas les cellules de Voronoï observables. Ils ont finalement conclu que voir les cellules de Voronoï n'était pas nécessaire pour réussir à sélectionner efficacement les cibles : sans le diagramme de Voronoï, ils VONT INTUITIVEMENT VERS LA CIBLE. Ce résultat est à rapprocher des résultats de l'expérience 2 concernant la distance des clics par rapport au centre des cibles et des cellules de Voronoï : les techniques ne fournissant aucune information *explicite* sont *centrées-cibles* (chapitre 6 section 6.2.4.1).

Axe *Élément augmenté* Les techniques *sur-cibles* sont bien notées et appréciées par les chirurgiens (figure 9.4). En effet, TARGET, Cell Painting et MCell Painting présentent un avantage sur les techniques *sur-espace* et *sur-curseur* : elles illustrent le principe même de l'expansion de cibles, soit en montrant directement comment la cible survolée est étendue, soit en agrandissant cette dernière, ce qui illustre le fait que sa taille est étendue. Les techniques *sur-espace* sont également appréciées (figure 9.4) pour la complétude de l'information qu'elle fournissent sur l'expansion des cibles. Les techniques *sur-curseur* ont en revanche été rejetées par les chirurgiens (figure 9.4), principalement à

cause de leur *dynamicité continue*. On peut supposer à la vue de ces résultats que comparé aux cibles ou à l'espace entre elles, le curseur n'est pas un support où l'utilisateur s'attend à trouver de l'information sur l'expansion des cibles.

Les combinaisons de techniques *sur-espace* et *sur-cible* ont été passablement appréciées pour la complémentarité des informations qu'elles fournissent (taille étendue de la cible et cible désignée). La combinaison Rope Cursor + TARGET (aides visuelles *sur-curseur* et *sur-cible*), apparue au Jeu 3, est en revanche la moins appréciée de toutes les techniques présentées (figure 9.4). Finalement, les chirurgiens trouvèrent que combiner plusieurs aides visuelles produit une technique inutilement compliquée.

9.1.2.2 Résultats des sessions de co-conception

Améliorer TARGET Le résultat le plus notable est que quelque soit l'axe de conception exploré, les deux chirurgiens se sont focalisés sur TARGET, considérée à l'issue de la mise en situation comme la meilleure technique. Leurs propositions consistaient donc à augmenter le facteur de grossissement de TARGET, augmenter le contraste entre la cible désignée et les autres cibles à l'aide d'autres couleurs pour la cible et pour le symbole (ou le nombre) qu'elle contient, ou encore à élargir la bordure de la cible désignée. Leur but est de rendre ainsi l'aide visuelle plus saillante. Pour le choix de nouvelles couleurs, les deux chirurgiens ont suggéré le vert car durant la mise en situation, les cibles non-sélectionnables en vert ont fortement attiré leur regard (figure 9.1). Enfin, aucun des deux chirurgiens n'a souhaité proposer de combinaison d'aides visuelles, considérant globalement qu'une aide visuelle SIMPLE ET DÉPOUILLÉE est particulièrement adaptée aux conditions chirurgicales.

Aides visuelles continues Les sessions de co-conception ont permis aux chirurgiens d'exprimer plus en détail leurs impressions sur l'utilisation de techniques *continues*. Nous pensons que cela a été rendu possible grâce à la description de l'espace de conception et de la cinématique des gestes de pointage. Ces apports théoriques ont incité les chirurgiens à adopter une approche plus *analytique* des techniques expérimentées.

Tous deux ont exprimé spontanément que ces techniques attirent le regard et requièrent plus de concentration que les autres techniques, confirmant que les techniques *continues* utilisent plus de ressources cognitives que les techniques *discrètes* ou *statiques*. Un des deux chirurgiens a également exprimé que contrairement à VTE, eVTE perturbe la perception de la structure de l'interface graphique, ce que l'on peut imputer à sa *dynamicité continue*, contrairement à VTE qui est *statique*. Enfin, un des chirurgiens a soutenu que Bubble Cursor et Rope Cursor ralentissent la tâche de pointage car leur aide visuelle *continue* (un ÉLÉMENT DE CONTINUITÉ) et *sur-curseur* encourage l'utilisateur à amener le curseur très proche de la cible visée avant de cliquer, malgré le fait que la cible soit déjà désignée. Or ce mouvement supplémentaire est inutile compte-tenu de l'expansion : le curseur est déjà dans la cellule de Voronoï de la cible. Ce chirurgien expliqua qu'il

attendait instinctivement que la bulle (ou la corde) rétrécisse et arrête de bouger avant de cliquer. Nous pensons que contrairement aux tâches de pointage de cette étude, des tâches qui nécessitent une grande précision, comme réaliser une ponction péricardique, tâche étudiée par Emmanuel Dubois (2001) [42], pourraient bénéficier d'aides visuelles *continues* et *sur-curseur*, celles-ci n'étant pas limitées à l'expansion de cibles.

9.1.3 Conclusion des interviews

Du point de vue méthodologique : Pour ces interviews, une méthode spécifique d'étude a été créée et mise en place afin de mener une exploration systématique de l'espace de conception des techniques d'expansion de cibles (chapitre 4). Chaque jeu de techniques d'expansion expérimentées par les chirurgiens a été construit en cherchant à décliner chaque axe de conception en considérant des valeurs différentes. Les sessions de co-conception ont ensuite permis d'approfondir notre étude de leurs besoins et de leurs ressentis. Cette méthode est probablement applicable à d'autres espaces de conception de la littérature. Un défaut majeur de cette étude est le faible nombre de sujets. Il est cependant difficile d'obtenir du temps pour l'étude des interfaces graphiques auprès de personnes aussi demandées que les chirurgiens.

Du point de vue conception concernant les techniques : Les résultats montrent cependant que les principes de simplicité et d'efficacité sont dominants dans l'évaluation d'interfaces graphiques pour la chirurgie augmentée, cela ayant été directement exprimé par les deux chirurgiens ayant pris part à l'étude. En conséquence, les augmentations *discrètes* des cibles semblent plus adéquates à ce contexte applicatif, car elles sont perçues plus intuitives et confortables que les techniques *continues* ou *statiques*. Comme la dynamisme des aides visuelles *continues* attire le regard et génère une augmentation sensible de la charge cognitive des chirurgiens, ces dernières semblent clairement inadéquates à la facilitation des tâches de sélection de cibles en contexte *per-opératoire*. Une telle mise en situation des techniques d'expansion de cibles complète efficacement les études en laboratoire présentées précédemment (chapitre 6 et 8) en enrichissant nos recommandations de facteurs prédictifs pour chaque catégorie de techniques d'expansion de cibles.

Dans la section suivante, nous décrivons comment l'ensemble des études expérimentales en laboratoire, les recommandations de conception issues de l'expérience 2 (chapitre 6 section 6.2.6) et les résultats de ces interviews ont été exploités pour améliorer la sélection de cibles dans la version 6 du logiciel TKA dédié à la pose de prothèses totales de genou.

9.2 Total Knee Arthroplasty (TKA) version 6

Les travaux ont été transférés au sein d'un produit d'Aesculap. Une technique d'expansion de cibles basée sur la tessellation de Voronoï a été intégrée au sein du logiciel TKA, figure de proue des logiciels de l'OrthoPilot®. En effet, TKA version 5 est actuellement le logiciel OrthoPilot® le plus utilisé par les chirurgiens dans le monde entier. Cette

amélioration de l'interface homme-machine de l'OrthoPilot® sera publiée en même temps que la version 6 de TKA, c'est-à-dire courant 2018.

De nombreuses aides visuelles ayant été implémentées dans la plate-forme logicielle C++/Qt de l'OrthoPilot®, les ingénieurs d'Aesculap ont pu tester et évaluer de nombreuses variantes de l'expansion de cibles basée sur la tessellation de Voronoï, avant de présenter cette amélioration aux autres pôles de compétences d'Aesculap, comme le marketing et le hardware. L'aide visuelle choisie est TARGET, un choix motivé par les résultats de nos travaux expérimentaux et des interviews réalisées avec les deux chirurgiens orthopédistes Dr. Bettega et Dr. Panisset.

L'écran de planification de la pose de la partie fémorale d'une prothèse totale de genou ainsi amélioré est présenté à la figure 9.5. Comme dans la version 5 de TKA, le Virtual Pointer est utilisé pour contrôler le curseur à distance.

L'interface graphique a évolué de la version 5 à la version 6 de TKA : le curseur est maintenant un cercle bleu entouré de 4 ailettes. Le bleu est utilisé dans l'interface graphique pour colorer les cibles désignées (figure 9.5). Cette évolution permet de visualiser sans peine les liens existant entre plusieurs grandeurs représentées sur cet écran. En effet, modifier la valeur de certaines cibles modifie également la valeur d'autres cibles. À la figure 9.5, la cible désignée est le 10 sur fond bleu. Elle est reconnaissable à sa taille supérieure aux autres cibles, comme l'implique l'utilisation de TARGET, et à la présence de deux boutons gris qui contiennent chacun une flèche verticale. Ces deux boutons représentent

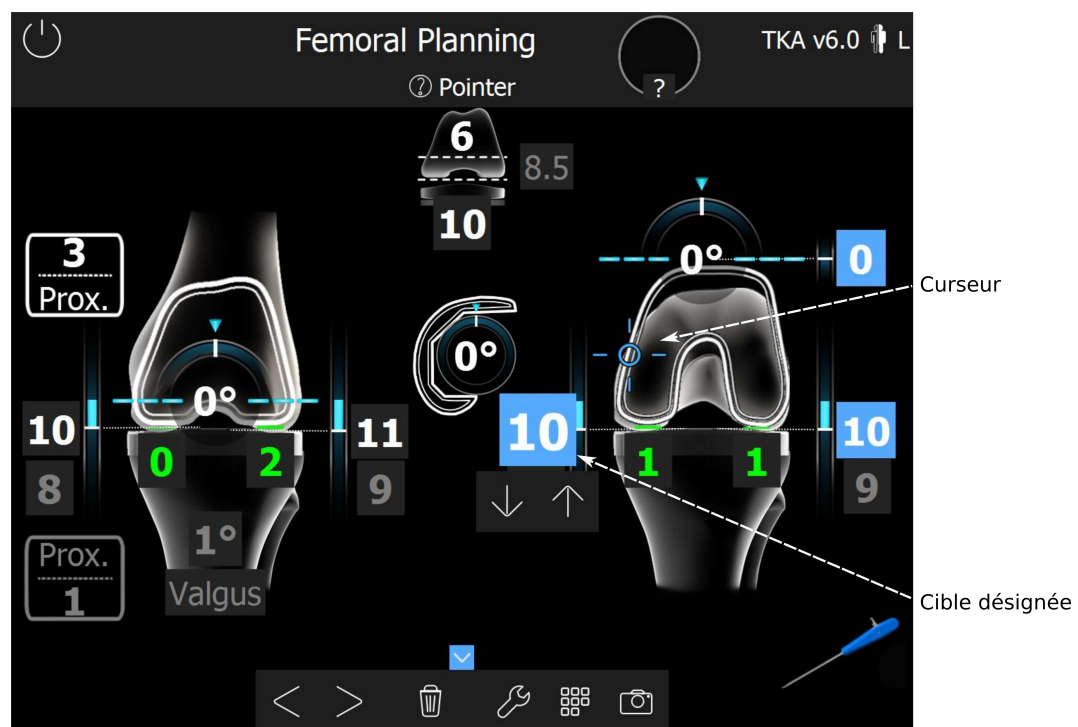


FIGURE 9.5 – Écran de planification du logiciel TKA 6. Le curseur est un cercle bleu entouré de 4 ailettes. La cible désignée est le nombre 10 sur fond bleu proche du centre de l'écran.

les gestes à effectuer pour modifier la valeur de la cible, une fois que la sélection est validée par un appui avec la pédale médicale de l'OrthoPilot®. Les autres cibles sur fond bleu représentent des grandeurs directement liées à celle que représente la cible désignée.

Le fait que les membres décisionnaires d'Aesculap SAS ont validé l'intégration de TARGET au logiciel TKA 6 constitue un retour très positif sur nos recherches, par un transfert effectif vers un produit.

Ce transfert d'une technique d'expansion dans un système de Chirurgie Assistée par Ordinateur (CAO) a nécessité une phase de certification. En effet les logiciels de l'OrthoPilot® sont des logiciels embarqués et certifiés. Une fois installés au bloc opératoire, ils ne sont ni mis-à-jour ni corrigés. Ils sont certifiés par des organismes tels que la FDA (Food and Drugs Administration) aux États-Unis, la PMDA (Pharmaceutical and Medical Device Agency) au Japon ou encore le TÜV (Technischer ÜberwachungsVerein, association d'inspection technique en français) pour l'Union Européenne. Les fournisseurs de systèmes de CAO tels qu'Aesculap s'engagent à respecter un ensemble de règles et de méthodes de conception, de validation et de documentation, qui sont conçues pour assurer la fiabilité des systèmes de CAO, qualité nécessaire pour ces systèmes critiques compte-tenu des risques inhérents à toute chirurgie. Les organismes certificateurs audient les entreprises, vérifient la documentation et éventuellement les systèmes concernés. Sans la certification délivrée par ces organismes, il est strictement interdit d'installer et d'utiliser un système de CAO.

9.3 Medical TapTap

Au delà du pointage d'une cible, nous avons étudié la confirmation de la sélection d'une cible, qui actuellement est faite avec une pédale médicale. Cela nous a amenés à créer Medical TapTap, une technique d'interaction mobile au pied.

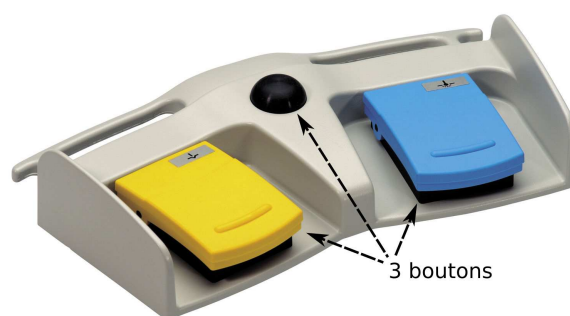


FIGURE 9.6 – Pédale médicale de l'OrthoPilot®. Elle est de plus équipée d'un arceau, non visible ici, permettant de la déplacer en la soulevant avec la pointe du pied.

9.3.1 Existant : Pédale médicale

De nombreux systèmes de CAO utilisent une pédale médicale pour l'interaction chirurgien-ordinateur (9.6). Souvent relié par câble aux systèmes de CAO, ce périphérique est historiquement présent dans les blocs opératoires pour contrôler des outils comme les bistouris électriques ou, en chirurgie orthopédique, les perceuses-fraiseuses et les scies oscillantes ou réciproques. La pédale médicale présente les avantages d'être relativement mobile, simple d'utilisation, reconnaissable au toucher et surtout, elle préserve l'asepsie des chirurgiens, dont les membres inférieurs ne sont pas stériles et peuvent toucher un tel objet.

Pour l'OrthoPilot®, la pédale est un périphérique incontournable pour l'interaction chirurgien-ordinateur. En effet, lors d'une chirurgie, la pédale est utilisée de nombreuses fois car elle dispose de trois boutons (figure 9.6). Grâce à la distinction entre appui long (dépassant 3 secondes) et appui court, elle propose un vocabulaire de 6 commandes, souvent suffisant pour les interactions simples comme l'acquisition d'un point anatomique. Lorsque l'interaction nécessite un dispositif de pointage, par exemple pour planifier la pose de la partie fémorale d'une prothèse totale de genou (section précédente), le chirurgien utilise le Virtual Pointer, qui permet de contrôler un curseur avec le palpeur par ray-casting (chapitre 2 section 2.4.2) et la pédale pour confirmer la sélection.

La mobilité des pédales médicales est cependant améliorable, car un chirurgien bouge autour de la table d'opération. Il peut s'éloigner de la pédale et avoir du mal à la retrouver sous la table d'opération et à la ramener vers lui pour l'utiliser. Ces difficultés peuvent également apparaître à un moment où le chirurgien ne peut pas se déplacer en raison des opérations chirurgicales en cours – s'il est contraint de garder en main certaines structures anatomiques par exemple.

Pour compléter notre démarche d'amélioration de l'interaction chirurgien-ordinateur dédiée aux tâches de pointage, nous avons étudié la phase de confirmation de la sélection faite avec une technique de pointage. La technique d'interaction *Medical TapTap* que nous présentons dans cette section vise à remplacer la pédale, pour la validation de la sélection des cibles mais aussi pour la spécification de commandes à l'OrthoPilot®. En effet *Medical TapTap*, une technique d'interaction gestuelle du pied, a le potentiel de remplacer totalement la pédale médicale pour l'interaction chirurgien-ordinateur, tout en bénéficiant d'une mobilité supérieure à celle de la pédale. Cette technique utilise pour la détection des gestes le Myo armband (figure 9.7), un bracelet pouvant être porté au poignet ou à la cheville. Nous n'avons cependant pas connaissance d'utilisation podale de ce périphérique. Le premier prototype de cette technique a été implémenté chez Aesculap par Sylvain Guy, stagiaire ENSIMAG 2A durant l'été 2015.



FIGURE 9.7 – Bracelet Myo. Image issue de <https://www.myo.com/techspecs>.

9.3.2 Interaction gestuelle podale

9.3.2.1 Bracelet Myo

Le bracelet Myo (<https://www.myo.com/>) mesure 1,14 cm d'épaisseur (figure 9.7) et 19 cm de diamètre, extensibles à 34 cm. Il pèse 93 g. Outre sa structure élastique, il est composé de 8 blocs enrobés d'acier chirurgical se plaçant au contact de la peau de l'utilisateur. Ces blocs sont les électrodes de capteurs électromyographiques, qui détectent l'activité électrique des muscles de l'utilisateur.

Le Myo est également équipé de 9 capteurs inertiels : un accéléromètre à 3 axes, un gyroscope à 3 axes et un magnétomètre à 3 axes. Il envoie ses données par technologie sans-fil Bluetooth. Son processeur ARM Cortex M4 permet de produire des retours haptiques sous forme de vibrations plus ou moins longues. Le Myo est un périphérique autonome qui se charge par micro-USB. Sa batterie lui permet de fonctionner une journée entière sans rechargement. Thalmic Labs Incorporation (<https://www.thalmic.com/>), fournit avec le bracelet Myo un kit de développement logiciel pour systèmes Windows, Mac, iOS et Android.

Le Myo a été utilisé par Haque et al. [62] pour concevoir un dispositif de pointage mains-libres sur le modèle de celui présenté par Vogel et Balakrishnan en 2005 [133]. Les auteurs montrent ainsi qu'un dispositif *grand public* équipé de capteurs électromyographiques et de capteurs inertiels tel que le Myo permet d'implémenter à moindre coût un système de pointage à distance utilisable. D'autres travaux [82, 108] utilisent ce périphérique pour l'interaction gestuelle avec les mains dans des contextes d'utilisation spécifiques, respectivement en voiture et pour la réalité virtuelle. Dans nos travaux, le Myo est utilisé à la cheville pour définir une technique d'interaction gestuelle podale.

9.3.2.2 Interaction gestuelle podale

De nombreux travaux ont exploré les interactions gestuelles aux pieds. En 2004, Pakkanen et Raisamo [103] étudient le potentiel des interactions podales pour les tâches spatiales (déplacer un objet virtuel par exemple) ne nécessitant pas une grande précision. Pour cela, ils utilisent un trackball de grande taille utilisé soit avec le pied, soit avec la main. Leurs résultats expérimentaux montrent que les interactions podales conviennent à de telles tâches, à condition que celles-ci ne soient pas temporellement fortement contraintes, c'est-à-dire que l'utilisateur ait suffisamment de temps pour utiliser la technique d'interaction au pied, moins rapide que la technique d'interaction à la main. L'acceptabilité de leur système d'interaction podale est cependant assez faible, selon les auteurs, en raison de la nouveauté de ce type d'interaction. Plusieurs travaux s'intéressent aux interactions podales pour des contextes d'utilisation variés.

Plusieurs études sur l'interaction au pied sont destinées à contrôler un smartphone resté dans la poche. Crossan et al. [39] conçoivent un langage de tapotements des pieds détectés par le pack de capteurs JAKE (<http://code.google.com/p/jake-drivers>), qui contient accéléromètres, magnétomètres et système de communication à distance Bluetooth. Cette technique permet d'interagir avec un smartphone sans le sortir de sa poche. Les auteurs comparent cette utilisation sans contact visuel aux utilisations classiques : smartphone en main et smartphone en poche, sorti pour chaque tâche. Ils montrent ainsi la faisabilité et la pertinence de telles techniques pour les interactions courtes (moins de 4 tapotements des pieds). Alexander et al. [8] étudient aussi les gestes et les déplacements du pied pour l'interaction mains libres avec un smartphone. Les auteurs fournissent un ensemble de recommandations pour la conception de techniques d'interaction gestuelle avec le pied, par exemple utiliser un vocabulaire gestuel assez restreint ou éviter les coups de pied vers l'arrière avec le talon pour définir un geste.

Considérant un autre contexte d'utilisation que celui de l'interaction avec un smartphone, Sangsuriyachot et Sugimoto [113] détectent les translations et les rotations des pieds avec un périphérique de leur conception. L'objectif est d'interagir simultanément avec les mains et les pieds avec une table interactive.

Pour interagir debout avec un ordinateur, Saunders et al. [114] présentent la technique d'interaction *Tap-Kick-Click*. Le vocabulaire gestuel est composé de séquences de *taps* (figure 9.8 en haut) – des tapotements de tout ou partie du dessous du pied sur le sol – et de *kicks* – des petits coups de la pointe du pied. Les gestes sont effectués dans une zone semi-circulaire autour de chaque pied (figure 9.8 en bas). Les deux zones (une pour chaque pied) sont partagées en plusieurs zones cibles, typiquement 3 zones par pied. Dans cette étude, la détection des *taps* et des *kicks* repose sur la caméra de profondeur d'un Kinect (chapitre 2 section 2.3.2) et des capteurs de force placés sur les chaussures de l'utilisateur.

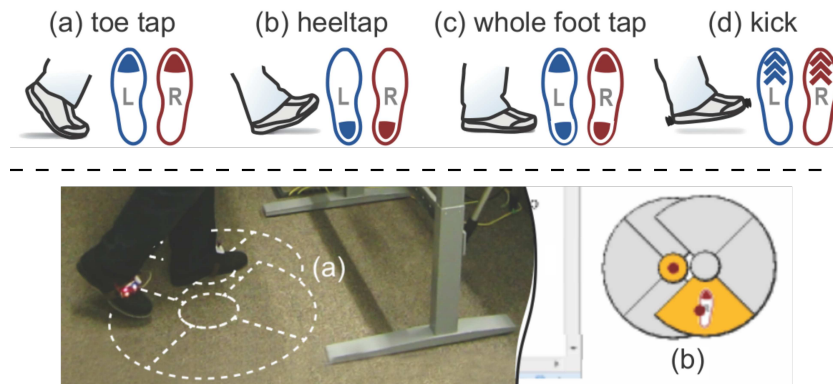


FIGURE 9.8 – En haut, les 4 actions du pieds de la technique Tap-Kick-Click et leurs icônes respectifs : (a) *tap* de la pointe, (b) *tap* du talon, (c) *tap* du pied entier et (d) *kick*. En bas, (a) les zones cibles autour de chaque pied, (b) le retour visuel d'un *tap* de la pointe du pied droit dans la zone arrière, tandis que le pied gauche est dans la zone de repos, au centre des cibles. Image modifiée de [114].

Enfin, les *Kickables*, présentés par Schmidt et al. [117], sont des objets tangibles conçus spécifiquement pour l'interaction podale. Les kickables peuvent reproduire de nombreux contrôles standards, comme les boutons, les interrupteurs (va-et-vient), les curseurs ou les molettes, en contraignant par leur forme les mouvements possibles et en guidant ainsi les utilisateurs lors de l'interaction (figure 9.9).

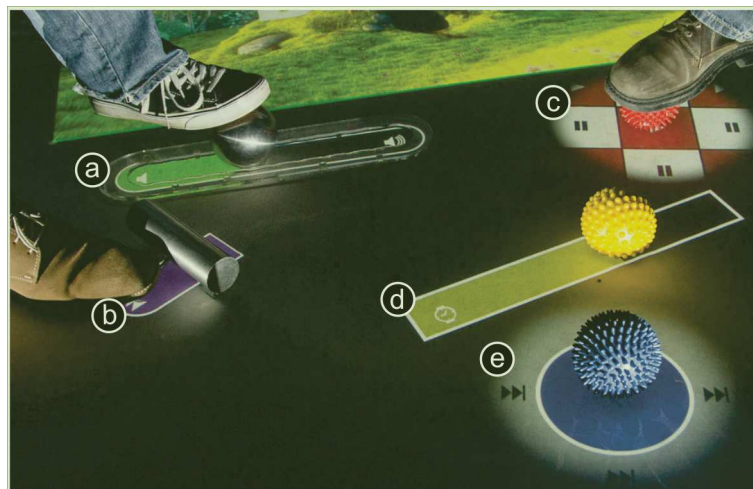


FIGURE 9.9 – Exemples de contrôles reproduits par 5 types de kickables : (a) curseur guidé (à une dimension), (b) interrupteur guidé, (c) sélecteur d'un élément d'une tessellation, (d) curseur non-guidé (potentiellement à deux dimensions, comme une souris), (e) bouton par franchissement. Image issue de [117].

Au delà de différents contextes d'utilisation de l'interaction aux pieds, des travaux plus génériques, soulignant une certaine maturité de cet axe de recherche, ont visé à établir un espace de conception de gestes au pied. Nous relevons deux espaces de conception [48, 119].

Pour étudier l'espace de conception des gestes du pied, Scott et al. [119] considèrent une détection optique très précise (figure 9.10). Cet espace contient 4 mouvements de base du pied (figure 9.10) :

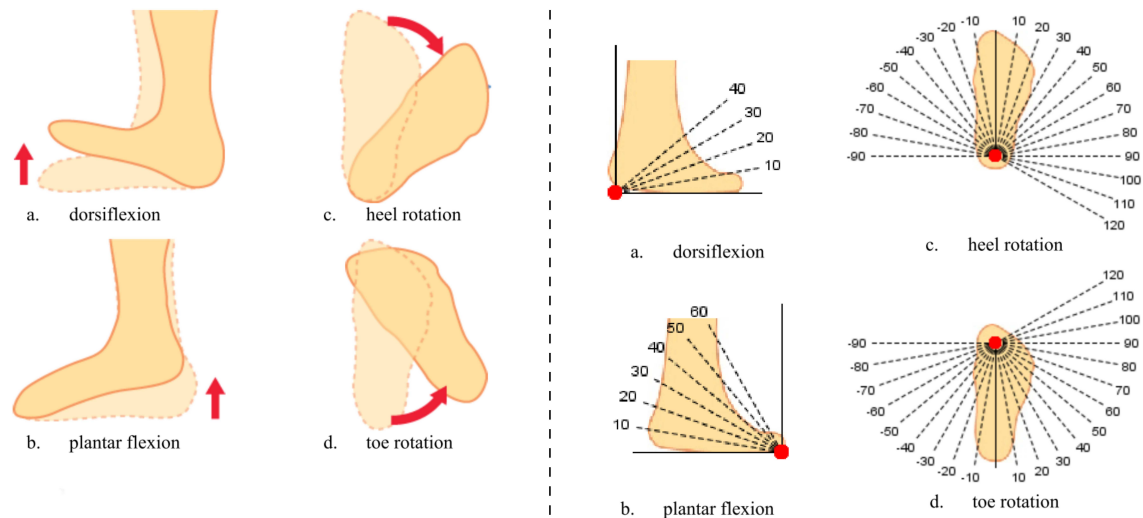


FIGURE 9.10 – À gauche, les 4 mouvements de base du pied définis par Scott et al. [119]. À droite, l'espace d'interaction exploré : chaque intervalle angulaire définit une cible à atteindre. Image issue de [119].

- lever la pointe du pied ou *dorsiflexion*,
- lever le talon ou *plantar flexion*,
- tourner le pied autour de la cheville ou *heel rotation* et
- tourner le pied autour de la pointe du pied ou *toe rotation*.

À partir de ces 4 mouvements, ils définissent un éventail de cibles angulaires (figure 9.10) et soulignent le fort potentiel de la *dorsiflexion* et de la *plantar flexion*. Les auteurs utilisent également les capteurs inertiels d'un smartphone laissé dans la poche de l'utilisateur pour détecter des gestes du pied. Ils concluent par des recommandations pour la conception de vocabulaires gestuels pour le pied.

Enrichissant l'espace des mouvements du pied, Fukahori et al. [48] proposent un vocabulaire de gestes aux pieds (figure 9.11) qui exploitent la pression. Ceux-ci sont détectés grâce à des chaussettes équipées de 8 capteurs de pression chacune. L'utilisateur déclenche des commandes en changeant la distribution de pression de ses pieds sur le sol (figure 9.11). Cette gestuelle est d'une grande discrétion, quasi-invisible pour un observateur

1. Item selection					2. Map navigation				
Left	Right	Up	Down	Select	Move viewpoint leftward	Move viewpoint rightward	Move viewpoint upward	Move viewpoint downward	Rotate map clockwise
2. Map navigation			3. Media control				4. Phone control		
Rotate map counter-clockwise	Zoom in	Zoom out	Play/pause	Previous track	Next track	Volume up	Volume down	Answer phone call	Ignore phone call
4. Phone call		5. Browser navigation				6. Gesture recognition			
Answer phone call (call with vibration)	Ignore phone call (call with vibration)	End phone call	Home screen	Previous page	Next page	Focus address bar	Activate	Deactivate	

FIGURE 9.11 – Exemple de langage gestuel proposé par Fukahori et al. [48] et établi par une démarche de conception centrée-utilisateur. Image issue de [48].

non-averti. Les auteurs définissent par une démarche de conception centrée-utilisateur un langage gestuel permettant d'effectuer 29 commandes courantes avec un ordinateur (figure 9.11) inspirées des travaux de Alexander et al. [8].

Nous retenons de ces travaux une grande variété dans les dispositifs utilisés pour capter l'interaction au pied. Les études détaillent la détection des gestes et fournissent des informations utiles pour les algorithmes de détection. Plusieurs contextes d'utilisation ont été identifiés et explorés. La majorité de ces contextes impliquent une interaction sans contact visuel. Cet apport est central pour l'interaction chirurgien-ordinateur en contexte per-opératoire.

9.3.3 *Medical TapTap*

Nous décrivons *Medical TapTap*, une nouvelle technique d'interaction gestuelle au pied développée spécifiquement pour le contexte per-opératoire. *Medical TapTap* est conçu dans le but de remplacer la pédale médicale de l'OrthoPilot® tout en restant constamment accessible par le chirurgien. Un chirurgien utilisant *Medical TapTap* porte un bracelet Myo à la cheville de son pied dominant. S'il est porté par-dessus une chaussette, le Myo ne pose aucun problème de confort, même pour une utilisation prolongée. La connexion sans-fil Bluetooth entre le Myo et l'OrthoPilot® permet au Myo de fonctionner dans une zone d'une demi-douzaine de mètres centrée autour de l'OrthoPilot®. En contexte réel, le chirurgien se tient typiquement à distance entre 2 et 4 mètres de l'OrthoPilot®. *Medical TapTap* permet donc, grâce au Myo, de concrétiser cet objectif d'accessibilité constante : le chirurgien peut se déplacer avec le Myo et spécifier des commandes à l'OrthoPilot® depuis n'importe quel point du bloc opératoire. *Medical TapTap* peut être doublée si le chirurgien porte un Myo à chaque pied. Cette première version est toutefois monopodale.

9.3.3.1 Vocabulaire gestuel

Medical TapTap utilise un vocabulaire de 6 gestes du pied, la pédale médicale de l'OrthoPilot® proposant un vocabulaire de 6 appuis-pédale. Ce vocabulaire gestuel s'inspire à la fois de l'espace de conception de Scott et al. [119] et de la technique *Tap-Kick-Click* [114]. Il est constitué des gestes suivants :

- le double tap de la pointe du pied, talon posé,
- le double tap du talon, pointe du pied posée,
- la rotation du pied autour du talon vers la droite (la pointe du pied se déplaçant),
- la rotation du pied autour du talon vers la gauche,
- la rotation du pied autour de la pointe vers la droite (la cheville se déplaçant) et
- la rotation du pied autour de la pointe vers la gauche.

Les rotations sont des mouvements "aller-retour" pour lesquels le pied part d'une position de repos et y revient. Ces gestes doivent être réalisés en moins d'une seconde pour être détectés. Ils sont faciles à réaliser et génèrent une faible fatigue musculaire.

9.3.3.2 Détection des gestes

La détection des gestes du pied implique (au niveau matériel) seulement l'accéléromètre à 3 axes et le gyroscope à 3 axes du Myo. En effet, les capteurs electromyographiques du Myo ne sont pas utilisables car la chaussette du chirurgien gêne le contact direct des électrodes avec la peau. De plus, les travaux de Crossan et al. [39] et de Scott et al. [119] montrent que des capteurs inertiels suffisent pour la détection des gestes constituant le vocabulaire présenté ci-dessus.

Au niveau logiciel, l'OrthoPilot® utilisant Microsoft Windows, notre programme de détection des gestes utilise le kit de développement fourni par Thalmic Labs Incorporation avec le langage C++ et la bibliothèque logicielle GRT (Gesture Recognition Toolkit). La bibliothèque GRT est open-source et développée par Nick Gillian du MIT (Massachusetts Institute of Technology, aux États-Unis). C'est une bibliothèque d'algorithmes d'apprentissage automatique dédiée à la reconnaissance de gestes en temps réel. La bibliothèque GRT est de plus codée en C++, qui est le langage de programmation des applications OrthoPilot®.

Analyse des signaux inertiels Les 6 signaux, un pour chaque axe de l'espace (X, Y et Z), fournis par l'accéléromètre et le gyroscope du Myo sont analysés par fenêtres successives de 1 seconde de large, toutes les 20 millisecondes, le Myo envoyant ces données à l'ordinateur à une fréquence de 50 Hz. Ils sont tout d'abord filtrés : un filtre passe-bas élimine les signaux de haute fréquence, typiquement non-imputables à un mouvement humain, et un filtre passe-haut élimine les signaux de basse fréquence, qui sont trop lents pour faire partie des gestes recherchés (par exemple, l'accélération gravitationnelle, constante et sans rapport avec un geste). Un ensemble de caractéristiques temporelles et fréquentielles sont ensuite extraites des signaux filtrés. Pour chaque signal, nous utilisons dans le domaine temporel :

- la moyenne,
- l'écart-type,
- les extremums,
- la durée entre minimum et maximum,
- le ratio minimum/maximum,
- la corrélation avec les signaux d'autres axes,
- l'énergie court-terme (égale à la somme des carrés de l'accélération),
- le maximum de l'auto-corrélation et
- le signe du pic.

Dans le domaine fréquentiel, nous utilisons :

- la moyenne de la transformée de Fourier,
- les extremums de la transformée de Fourier,
- l'énergie spectrale.

Ces caractéristiques des signaux ne sont calculées que si un pic d'accélération, typique d'un geste volontaire, est détecté. En effet, une des difficultés de l'interaction gestuelle est la distinction entre les gestes volontaires destinés à l'interaction et ceux communément effectués comme la marche ou la rotation du chirurgien sur lui-même pour saisir un outil.

Apprentissage automatique Les algorithmes d'apprentissage automatique nécessitent une phase d'entraînement préalable à la détection d'actions comme les gestes du vocabulaire décrit ci-dessus. Durant cette phase, les données reçues (en l'occurrence les caractéristiques calculées pour chaque signal fourni par les capteurs inertiels du Myo) sont utilisées par l'algorithme pour construire un modèle des actions à reconnaître, qu'il s'agisse de gestes ou de paroles (la reconnaissance de parole étant un des domaines de l'informatique utilisant massivement ce type d'algorithmes). Ces modèles "appris" sont spécifiques à chaque algorithme d'apprentissage automatique et sont utilisés pour détecter en temps réel les actions modélisées.

Après comparaison avec plusieurs algorithmes d'apprentissage automatique fournis par la bibliothèque GRT – ANBC : Adaptive Naive Bayes Algorithm, HMM : Hidden Markov Model et DWT : Dynamic Time Warping – le Support Vector Machine (SVM) s'est révélé l'algorithme le mieux adapté à la détection des gestes du vocabulaire de Medical TapTap. En effet, celui-ci permet d'atteindre des taux de détection élevés (supérieurs à 90 %) pour un nombre d'entraînements raisonnable (5 exemplaires de chaque geste). Lors d'une première utilisation de Medical TapTap, un chirurgien "entraîne" MedicalTapTap (et donc le SVM) pendant environ 3 minutes, ce qui est chirurgicalement acceptable. Cet apprentissage personnalisé offre un meilleur taux de détection que l'utilisation de modèles génériques des gestes, chaque chirurgien ayant une façon personnelle et unique de réaliser les gestes du vocabulaire. Chaque geste est ainsi répété, enregistré et analysé au moins 5 fois. Un plus grand nombre de répétitions améliore la précision de la détection, dans une certaine limite toutefois. Afin d'améliorer la détection, la marche est présentée au SVM comme un geste à part entière et 20 secondes de marche sont également enregistrées et modélisées par le SVM. Une interface graphique permet la réalisation de cette phase d'entraînement, ainsi que des réglages du SVM.

Détection des gestes La détection obtenue avec seulement 5 exemplaires de chaque geste est très satisfaisante. Les ingénieurs d'Aesculap, qui connaissent les problématiques et les exigences de la chirurgie considèrent la détection comme suffisante pour un développement plus approfondi de Medical TapTap (pour une détection encore plus robuste notamment) et une éventuelle utilisation per-opératoire. Une présentation orale de Medical TapTap à plusieurs chirurgiens a recueilli des avis positifs, voire enthousiastes.

Malgré ces retours encourageant, Medical TapTap est à l'heure actuelle un prototype expérimental, prouvant la faisabilité et la pertinence de cette technique d'interaction pour la chirurgie augmentée. Une expérience en laboratoire permettrait une étude approfondie des performances de détection et de l'utilisabilité de Medical TapTap. Une étude en bloc opératoire, en contexte opératoire réel (ou simulé, lors de sessions *tests-cadavres*) permettraient une étude approfondie de l'acceptabilité de la technique par les chirurgiens.

9.3.4 Conclusion

Nous avons décrit une nouvelle technique d'interaction gestuelle dédiée à la chirurgie augmentée, Medical TapTap. En utilisant les capteurs inertiels d'un bracelet Myo, qui pourrait être remplacé par un dispositif plus simple, Medical TapTap modélise et détecte ensuite en temps réel un vocabulaire de 6 gestes du pied. Deux Myo permettraient d'étendre ce vocabulaire à plus de 12 gestes, pour permettre la spécification de nombreuses commandes directes de l'OrthoPilot®. Medical TapTap avec un Myo atteint néanmoins son objectif de conception en fournissant un vocabulaire de commandes directes équivalent à celui fourni par une pédale médicale telle que celle de l'OrthoPilot®. En particulier MedicalTapTap permet la validation d'une sélection faite avec une de nos techniques de pointage. Medical TapTap présente de plus l'avantage crucial de permettre d'interagir depuis n'importe quel point du bloc opératoire, surpassant ainsi la pédale médicale en terme de mobilité et d'accessibilité par le chirurgien. Le prototype réalisé mériterait cependant de plus amples investigations et des améliorations, comme un dispositif matériel dédié plus facile à mettre à la cheville, une détection encore plus précise ainsi que des études d'acceptabilité et de performance en laboratoire puis en bloc opératoire.

Conclusion

Dans ce chapitre, nous avons présenté notre travail d'intégration de nos recherches aux logiciels d'Aesculap, en particulier l'OrthoPilot®, un système de CAO. Afin de compléter notre étude en laboratoire des techniques d'expansion de cibles nous avons interviewé deux chirurgiens orthopédistes utilisateurs d'OrthoPilot®. Ces interviews sont structurées en deux parties. La première est une phase de mise en situation, pendant laquelle le chirurgien utilise plusieurs techniques d'expansion intégrées à l'écran de planification simplifié du logiciel TKA 5. La deuxième partie est une phase de co-conception, pendant laquelle le chirurgien, après un exposé de notre espace de conception (chapitre 4), propose de nouvelles techniques d'expansion de cibles. Les résultats de nos travaux expérimentaux en laboratoire et de ces interviews chez Aesculap ont motivé l'intégration de la techniques d'expansion de cibles TARGET au logiciel TKA 6, prochaine version du logiciel phare d'Aesculap annoncée pour 2018.

Après ces travaux focalisés sur le pointage à distance, nous avons étudié les techniques d'interaction gestuelle au pied permettant de remplacer la pédale médicale pour confirmer les sélections de cibles et pour donner des commandes directes à OrthoPilot®. Nous avons ainsi conçu Medical TapTap, une technique d'interaction podale mobile exploitant les capteurs inertiels du bracelet Myo porté à la cheville. Cette technique propose un vocabulaire minimal de 6 gestes du pied, extensible à 12 gestes si le chirurgien porte un bracelet Myo à chaque cheville. La poursuite de ces travaux s'oriente tout d'abord vers une évaluation expérimentale de cette technique d'interaction.

Le fait de faire partie de l'équipe de Recherche et Développement d'Aesculap a permis d'avoir accès au logiciel TKA, à l'OrthoPilot®, d'avoir des retours professionnels de CAO et de conduire des interviews avec deux chirurgiens : ceci a permis un transfert effectif de nos travaux vers un produit commercialisé.

Conclusion et perspectives

Sommaire

Conclusion générale	191
Perspectives à court-terme	193
Perspectives à long-terme	199

Conclusion générale

Pointer une cible avec un curseur, puis confirmer la sélection (par un clic ou autre action équivalente), est une tâche quasi-omniprésente dans l'utilisation de tout système informatique. Des contextes privés aux contextes professionnels ou publics les plus variés, pointer une cible peut s'avérer difficile : elle est parfois petite, lointaine, voire mobile. De plus, les utilisateurs sont plus ou moins performants dans la manipulation des dispositifs de pointage.

Faciliter la sélection de cibles est un enjeu majeur de la recherche scientifique en Interaction Homme-Machine. Dans ce but, une technique d'expansion de cibles alloue à celles-ci plus d'espace interactif. La technique communique à l'utilisateur l'expansion calculée par son algorithme d'expansion via une aide visuelle. Nos travaux ont été consacré aux aides visuelles employées par ces techniques. Financés par une bourse CIFRE avec l'entreprise Aesculap SAS, fournisseur du système de chirurgie augmentée OrthoPilot®, nos travaux sur l'expansion de cibles visent à améliorer l'interaction chirurgien-ordinateur en bloc opératoire.

Notre travail de recherche s'adresse en premier lieu aux concepteurs de techniques d'interaction. Nous présentons tout d'abord un espace de conception des aides visuelles des techniques d'expansion (chapitre 4). Cet espace à trois dimensions (trois axes de conception) introduit de nouveaux concepts fondateurs inspirés de l'étude des techniques existantes. La vocation de cet espace de conception est double. Outre la description des aides visuelles existantes et futures, son exploration suggère de multiples possibilités de création d'aides visuelles. C'est cette exploration qui a permis la création de plusieurs nouvelles techniques d'expansion de cibles (chapitres 5 et 9) : VTE, eVTE, MTE, Expansion Lens, Rope Cursor, eMTE, MCell Painting, ainsi que plusieurs techniques utilisant une des nombreuses combinaisons possibles d'aides visuelles.

L'étude expérimentale de ces techniques (chapitre 6) nous a permis de proposer un ensemble de recommandations pour les concepteurs et nous a incités à approfondir l'étude des performances. Le modèle conceptuel de performance proposé (chapitre 7) a pour finalité de prédire les performances des techniques d'expansion par l'étude des informations fournies par leurs aides visuelles. La pertinence de ces informations selon les différentes phases d'une tâche de sélection d'une cible est au cœur de ce modèle.

Au cours d'une démarche de validation expérimentale (chapitre 8), nous formulons grâce au modèle plusieurs prédictions sur les performances relatives de techniques d'expansion. Tandis que les résultats expérimentaux ne contredisent pas notre modèle, cette validation reste insuffisante et le modèle limité par la non-prise en compte de plusieurs phénomènes neuromoteurs mis en jeu dans le pointage d'une cible. Cette modélisation est cependant un premier pas significatif vers la compréhension de l'impact des aides visuelles sur les performances des techniques d'expansion de cibles.

Un travail de transfert des travaux aux problématiques d'Aesculap a permis l'amélioration du pointage de cibles à distance dans la prochaine version du logiciel phare de l'OrthoPilot®, TKA. Nous avons également conçu une nouvelle technique d'interaction gestuelle au pied, dédiée en première intention aux chirurgiens. Les gestes du pied sont détectés par un dispositif porté équipé de capteurs inertiels. Cette technique a pour vocation de remplacer la pédale médicale, dispositif omniprésent en chirurgie augmentée, mais à la mobilité limitée.

L'ensemble de ces contributions a donné lieu aux publications suivantes :

- M. Guillon, F. Leitner, and L. Nigay. Static Voronoi-based target expansion technique for distant pointing. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 41–48. ACM, 2014
- M. Guillon, F. Leitner, and L. Nigay. Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2777–2786. ACM, 2015
- M. Guillon, F. Leitner, and L. Nigay. Target expansion lens: It is not the more visual feedback the better! In *Proceedings of the 2016 International Working Conference on Advanced Visual Interfaces*, pages 52–59. ACM, 2016
- C. Coutrix, W. Delamare, M. Guillon, T. Kurata, F. Leitner, L. Nigay, and T. Vincent. Techniques de pointage à distance: Cibles numériques et cibles physique. In *UbiMob2014: 10èmes journées francophones Mobilité et Ubiquité*, pages 5–10, 2014
- M. Guillon, F. Leitner, and L. Nigay. Démonstration de VTE: une technique de pointage à distance. In *IHM'14, 26e conférence francophone sur l'Interaction Homme-Machine*, pages 8–9, 2014

Dans une dernière partie, nous développons plusieurs pistes de recherche pour la poursuite de ce travail de recherche, dont la portée se veut à la fois générale pour les techniques de facilitation de la sélection de cibles et locale pour le contexte per-opératoire.

Perspectives à court-terme

Enrichir l'espace de conception et étudier les algorithmes d'expansion

L'espace de conception que nous proposons au chapitre 4 concerne uniquement les aides visuelles des techniques d'expansion de cibles. Il est cependant possible de l'enrichir afin qu'il puisse caractériser entièrement une technique d'expansion. Pour cela, il convient de caractériser également les algorithmes d'expansion.

Nous envisageons deux axes de conception des algorithmes d'expansion. Un premier axe caractériserait le remplissage de l'espace libre par l'algorithme, certains algorithmes utilisant l'ensemble de l'espace disponible, comme la tessellation de Voronoï, et certains algorithmes laissant de l'espace libre pour d'autres interactions. Un deuxième axe se focaliserait sur les recouvrements possibles, ou non, entre les cibles étendues.

Certaines études [26, 34, 53] comparent des techniques d'expansion entières sans distinguer aides visuelles et algorithmes d'expansion. De nouvelles études figeant l'aide visuelle, comme nous avons figé l'algorithme d'expansion sur la tessellation de Voronoï, permettraient d'étudier les performances des différents algorithmes d'expansion indépendamment des aides visuelles employées, selon ces nouveaux axes de conception.

Au delà de la caractérisation de l'algorithme d'expansion, l'espace de conception peut aussi être étendu pour caractériser plus finement les aides visuelles. En particulier, une piste de recherche serait l'ajout d'un axe pour caractériser la source de la dynamique de l'aide visuelle. Celle-ci peut être la position du curseur, comme avec Bubble Cursor et Rope Cursor, la vitesse du curseur, comme avec Dynaspot et eVTE, ou encore le vecteur vitesse du curseur, comme avec Implicit Fan Cursor.

Enfin, la construction des deux espaces de conception pour les aides visuelles et pour les algorithmes d'expansion permettrait d'étudier dans un second temps les liens qui existent entre les deux espaces de conception et les interdépendances entre aide visuelle et algorithme d'expansion.

Mesurer la dynamique des aides visuelles et son impact sur les performances

Sur la dynamique des aides visuelles, l'utilisation d'un dispositif de suivi du regard (eye-tracking) permettrait de mesurer à quel point les aides visuelles dynamiques distraient les utilisateurs et impactent leurs performances.

De plus, les résultats de l'expérience 2 ont apporté quelques pistes pour quantifier la dynamique des aides visuelles (chapitre 6) et passer ainsi d'un espace discret de valeurs (*statique*, *discret* et *continu*) à un espace continu, à l'appui d'une unité de mesure adéquate. Une première possibilité consiste à considérer le nombre total de pixels ayant changé de couleur durant une tâche de pointage "modèle" ou durant un enchaînement

de tâches de pointage. Cette mesure est calculable a posteriori comme mesure expérimentale sur les données d'une expérience. Une autre mesure consisterait à considérer le nombre total de pixels modifiés par unité de temps durant des tâches de pointage. Nous avons commencé à explorer ces pistes en mettant en place ces mesures de la dynamicité sur les données expérimentales de l'expérience 2, mais les résultats obtenus ne nous ont pas permis de conclure. Ils ont en revanche été la source de nombreuses réflexions et ont participé à l'élaboration des bases du modèle conceptuel de performance présenté au chapitre 7. En effet, ces résultats ont montré que la performance d'une aide visuelle n'est pas influencée que par sa dynamicité, mais plutôt par son apport sémantique : l'information fournie par l'aide visuelle. Cette information étant parfois fournie par des éléments dynamiques, ces grandeurs sont concrètement liées.

Du modèle conceptuel à un modèle numérique de performance

Plusieurs limites au modèle conceptuel de performance présenté au chapitre 7 ont été identifiées. Une possibilité de poursuite de ce travail est d'enrichir ce modèle en prenant en compte les phénomènes neuromoteurs impliqués par une tâche de pointage, comme l'effet d'optimisation du geste balistique (chapitre 2), l'anticipation de l'action de validation, la parallélisation des sous-tâches du pointage, et le coût temporel d'un changement d'aide visuelle au cours du geste.

Le modèle de performance que nous avons présenté permet des prédictions relatives sur les performances des techniques d'expansion, c'est-à-dire des prédictions du type "telle technique sera plus rapide que telle autre". Nous considérons pour cela les informations fournies par les aides visuelles et plus particulièrement les informations *secondaires* fournies durant la phase de transfert, phase la plus longue et donc la plus significative en terme de durée d'une tâche de sélection. Une piste de recherche que nous avons déjà initiée consiste à faire évoluer ce modèle vers un modèle numérique de performance, c'est-à-dire un modèle permettant des prédictions chiffrées sur la durée moyenne d'une sélection de cible avec une technique d'expansion. Ces prédictions absolues (ne concernant pas que la comparaison de techniques) sont dépendantes des conditions expérimentales, comme les prédictions faites par la loi de Fitts par exemple. Pour ce type de modèle, la nécessité d'une régression linéaire implique une première analyse de plusieurs tâches de pointage afin de formuler des prédictions pour d'autres tâches de pointage *dans les mêmes conditions expérimentales*. L'aspect prédictif et numérique d'un tel modèle se concrétiserait alors sous la forme d'une variante de la loi de Fitts prenant en compte les informations *secondaires* fournies durant la phase de transfert.

Les résultats expérimentaux du chapitre 8 suggèrent que la vitesse de sélection est impactée différemment par l'information *explicite* et l'information *implicite*. Quantifier *numériquement* (par exemple en bits) les informations fournies par les techniques d'expansion est donc nécessaire pour construire un modèle prédictif *numérique*.

Quantifier l'information

Pour quantifier *numériquement* l'information, une première approche consiste à calculer l'entropie de Shannon [121] (chapitre 2) des différents stimuli visuels que fournissent l'interface graphique et les techniques d'expansion de cibles à l'utilisateur. Cette entropie est additive. L'entropie de Shannon H d'une cible sur un écran de 1920×1080 pixels, soit l'information "espace occupé par la cible", est, comme celle de l'information "position du curseur", égale à $\log_2(1920 \times 1080)$, soit environ 21 bits. Pareillement, l'information "cette cible est désignée parmi n_{cibles} cibles" est donnée par le calcul de $\log_2(n_{cibles})$ et l'information "distance la plus courte jusqu'à la cible" est donnée par le calcul de $\log_2(\sqrt{1920^2 + 1080^2})$ (elle est la réponse à la question : "parmi toutes les distances possibles, soit de zéro à la diagonale de l'écran, laquelle est la plus courte jusqu'à la cible ?"), soit environ 11 bits. Il est possible de calculer ainsi en bits l'entropie de Shannon de l'ensemble des informations recensées par l'analyse des techniques d'expansion et récapitulées à la figure 7.6. Cependant, se pose le problème des grandeurs adéquates pour ces calculs : le pixel est-il vraiment une grandeur adéquate pour quantifier l'information traitée par l'utilisateur ? L'utilisateur ne perçoit pas forcément l'écran comme une matrice de pixels. Pareillement, quelle unité d'angle choisir pour quantifier l'information fournie par l'ouverture de l'éventail de Fan Cursor ? Le problème soulevé par ces questions est celui du lien entre l'information *fournie* par l'interface et l'information *perçue* et interprétée par l'utilisateur. D'autres informations, comme "cette cible est désignée parmi n_{cibles} cibles", sont moins problématiques car le nombre de cibles est *a priori* une grandeur adéquate pour évaluer l'information traitée par l'utilisateur, tant que le nombre de cibles reste raisonnable.

Variante de la loi de Fitts

Nous avons commencé à étudier un modèle numérique en considérant comme unités de l'information le bit et le pixel. La loi de Fitts [46, 79] (chapitre 2) modélise la phase de transfert en prenant en compte l'information *primaire* car celle-ci est nécessaire pour réaliser une tâche de pointage. Elle inclut donc dans son calcul du temps de transfert *sans technique d'expansion de cibles* le temps nécessaire pour traiter l'information *primaire*. Le temps nécessaire pour traiter l'information *secondaire* s'ajoute à ce temps de transfert. Pour modéliser cet impact négatif de l'information *secondaire* sur le temps de transfert, nous ajoutons un troisième terme à la loi de Fitts. Le temps de transfert est allongé proportionnellement à l'entropie de Shannon de l'information *secondaire*. Ce premier choix de la proportionnalité est consécutif à l'étude des travaux de Hick (1952), de Hyman (1953) et de Seow (2005) [68, 69, 120] (chapitre 2). La loi résultant de cette démarche se formule ainsi :

$$T_{transfert} = a + b \cdot \log_2 \left(\frac{D}{d} + 1 \right) + c \cdot \frac{D}{d} \cdot H \quad (9.1)$$

où T (en ms) est le temps de transfert, D la distance à la cible étendue, d la taille de la cible étendue et H (en bits) l'entropie de Shannon de l'information *secondaire* fournie par la technique d'expansion durant la phase de transfert. Le coefficient $\frac{D}{d}$ ajouté au troisième terme de cette loi est issu du Modèle de l'Impulsion Initiale Optimisée [88] (chapitre 2). Il est une estimation de la probabilité d'occurrence d'une phase de contrôle *en boucle fermée* du geste de pointage en fonction des caractéristiques de la tâche. Ce coefficient estime également la probabilité que la tâche requiert des sous-mouvements correctifs.

Les résultats obtenus en appliquant a posteriori ce modèle aux données expérimentales (expériences 2 et 3) sont globalement encourageants : les coefficients de détermination (R^2 et R^2 ajusté) obtenus avec ce modèle sont supérieurs à ceux obtenus avec la loi de Fitts à des exceptions près, notamment VTE qui permet un effet d'optimisation du geste balistique très fort dans l'expérience 2 et non modélisé par la loi de Fitts. De plus, le critère AICc [7, 29, 30] (une mesure de la qualité relative des modèles statistiques basée sur la théorie de l'information [121]) est plus bas pour ce modèle que pour la loi de Fitts, ce qui signifie que la probabilité que la loi de Fitts soit un meilleur modèle que le modèle numérique élaboré est faible (10^{-13} pour l'expérience 3 et 10^{-5} pour l'expérience 2).

Les problèmes posés par l'analyse de l'information fournie par les techniques, par sa quantification et par l'interprétation des résultats (qu'est-ce que le coefficient c dans la formule 9.1 ?) rendent ce modèle très incomplet et nécessite une étude plus approfondie. La difficulté majeure à résoudre dans cette démarche de modélisation réside dans la prise en compte des stratégies de haut-niveau de l'utilisateur dans la quantification de l'information visuelle traitée par celui-ci, c'est-à-dire dans le calcul de l'entropie.

Réaliser des expériences in-situ

Au chapitre 9, nous avons présenté l'interview conduit avec deux chirurgiens, au cours desquelles nous avons recueilli des retours sur les techniques d'expansion qui sont complémentaires à ceux des expériences en laboratoire (chapitres 6 et 8).

Le passage des cibles abstraites aux cibles concrètes par l'intégration des techniques à un logiciel professionnel, la mise en situation avec ces utilisateurs particuliers et la confrontation avec leur regard dûment orienté, nous ont permis non seulement d'apporter à Aesculap des réponses situées et averties, mais également d'affiner notre compréhension des attentes des chirurgiens et de la manière dont les techniques d'expansion sont appréhendées pour une utilisation concrète.

Pour cela, nous avons reproduit tant que possible le contexte per-opératoire. Cependant, des éléments caractéristiques de ce contexte fondamentalement critique nous firent défaut, comme le stress de l'opération, le regard des collègues, l'urgence de l'imprévu, la présence des épaisseurs de gants, des tuniques et des masques, l'éventuelle souillure des mains et des outils, dont la palpeur du Virtual Pointer, ou encore la fatigue d'une journée de travail avancée. Malgré ces manquements, ces interviews furent riches d'enseignements.

Une piste de recherche consiste à poursuivre cette démarche en organisant des expériences similaires dans un contexte réel, c'est-à-dire lors d'opérations chirurgicales – ce que l'on nomme des *expériences cliniques*. Nous avons travaillé régulièrement avec Aesculap à la mise en place de telles expériences, mais les contraintes légales et les coûts engendrés ont rendu la chose inaccessible pour plusieurs raisons :

1. Les prototypes sont mal acceptés dans un contexte si critique [102]. Les législations en vigueur reflètent cette réticence en rendant les essais de ce type extrêmement coûteux : aucun logiciel non certifié par les organismes compétents ne doit être utilisé en bloc opératoire. Or, cette certification nécessite une documentation considérable ainsi qu'un travail approfondi d'analyse de risques et de tests (chapitre 9). Les exigences de sécurité sont de plus rarement revues à la baisse.
2. Pour trouver suffisamment de chirurgiens participant à l'expérience, il faudrait démarcher très largement les collaborateurs d'Aesculap dans le monde entier, mettre en place le logiciel de test sur chaque site et revenir a posteriori recueillir les données expérimentales.
3. La sélection de cibles n'est pas la tâche principale du chirurgien. Au cours d'une opération, un nombre limité de sélections peuvent être enregistrées car il s'agit également de ne pas entraver le déroulement normal de l'opération. Aussi un grand nombre d'interventions sont nécessaires pour pouvoir extraire des données expérimentales des résultats quantitatifs significatifs. De plus, pour ces raisons, concevoir un protocole de test aussi rigoureux qu'en laboratoire nécessite de nombreuses adaptations et la présence du concepteur à chaque opération reste quasiment indispensable pour s'assurer de la bonne utilisation du logiciel de test et des techniques d'expansion.

Une solution par défaut consiste à utiliser la réalité virtuelle, qui a fait ses preuves auprès des élèves-chirurgiens comme outil pédagogique. A priori plus accessible, la pertinence de la réalité virtuelle pour nos objectifs reste à explorer dans le détail.

Améliorer et évaluer Medical TapTap

Données EMG et algorithmes de détection

Au chapitre 9, nous avons présenté Medical TapTap, nouvelle technique d'interaction qui utilise les capteurs inertiels du bracelet Myo pour détecter un ensemble de gestes du pied destinés à l'interaction gestuelle mobile avec l'OrthoPilot® d'Aesculap. Le Myo a été utilisé pour la détection de gestes précisément car c'est un bracelet élastique et qu'il dispose de capteurs inertiels et d'un émetteur Bluetooth. Un autre prototype plus rudimentaire constitué uniquement de capteurs inertiels, d'un émetteur Bluetooth et d'un microcontrôleur Arduino (<https://www.arduino.cc/>), a été abandonnée pour le Myo pour des raisons pratiques. Cependant, ce prototype aurait pu réaliser une détection similaire avec des performances probablement très proches.

Cela permet de rappeler que le Myo offre d'autres capteurs que ceux utilisés actuellement par Medical TapTap et que ceux-ci pourraient être exploités pour une détection plus précise et potentiellement un vocabulaire gestuel plus large. Le bracelet Myo est constitué de huit blocs articulés contenant chacun, du côté intérieur, des électrodes. Celles-ci sont au contact de la peau de l'utilisateur (si la chaussette ne l'empêche pas) et permettent d'observer en temps réel l'activité électrique des muscles situés sous la peau, du mollet en l'occurrence.

Une piste d'amélioration de Medical TapTap est d'utiliser ces données électromyographiques (EMG) en complément des données inertielles pour augmenter le pouvoir d'expression de cette technique. Cette amélioration permettrait d'augmenter le nombre de gestes disponibles pour l'interaction et même potentiellement de proposer des gestes détectés uniquement grâce aux données EMG. La recherche en informatique sur les algorithmes d'apprentissage automatique produit également de nombreuses avancées et l'amélioration de Medical TapTap grâce à de nouveaux algorithmes reste une piste de recherche pertinente, notamment en vue de réduire le rapport *temps d'entraînement/taux de détection* de cette technique.

Évaluer Medical TapTap

Plusieurs expériences mériteraient d'être menées pour évaluer Medical TapTap. Pour confirmer l'intérêt de cette technique, il convient de répondre à la question suivante :

"Medical TapTap est-il vraiment plus efficient que la pédale médicale ?"

L'aspect tangible de la pédale reste intéressant malgré son manque de mobilité et son taux de détection d'un appui-pédale est de 100 %. Medical TapTap, comme tout dispositif de détection automatique de gestes, n'offre pas une détection parfaite. Si le taux de détection est élevé (supérieur à 90%), cela peut ne pas invalider la technique pour un contexte d'utilisation privé (commande d'objets connectés) ou public (interaction sur le parcours d'un musée par exemple). En revanche, dans un bloc opératoire, l'efficacité des dispositifs d'interaction est un critère majeur d'acceptation. Seule une confrontation aux chirurgiens, dans des conditions les plus proches possibles des conditions réelles de la chirurgie, permettrait d'estimer la validité du compromis *gain de mobilité/perte d'efficacité* engendré par la transition de la pédale vers Medical TapTap et ainsi de conclure sur la pertinence de cette technique en contexte chirurgical critique.

Perspectives à long-terme

Manipuler la trajectoire du curseur ou étendre les cibles ?

Dans le chapitre 3, nous présentons les techniques d'aide au pointage en trois catégories principales :

1. les techniques de manipulation de la trajectoire du curseur,
2. les techniques d'expansion de cibles et
3. les techniques de sélection en plusieurs temps.

Toutes visent à réduire la distance du curseur à la cible visée et/ou à augmenter la taille de la cible. À plusieurs occasions [16, 26, 53], des techniques de différentes catégories sont comparées en terme d'acceptabilité et de performance. Les résultats montrent en majorité une meilleure acceptabilité et de meilleures performances pour les techniques d'expansion de cibles que pour les techniques de manipulation du curseur. Ce constat a motivé en partie notre choix de se focaliser sur les techniques d'expansion. Nous pensons cependant que pour conclure sur ces deux types de techniques, des travaux doivent être menés.

La manipulation automatique du curseur par l'ordinateur crée un découplage entre la trajectoire programmée par l'utilisateur (dans l'espace moteur) et la trajectoire finale (dans l'espace visuel). Ce découplage crée chez l'utilisateur une dissonance cognitive qui pénalise ce type de techniques dans de nombreux cas d'utilisation par rapport aux techniques d'expansion de cibles. Les techniques de CD-Gain dynamique ont cependant été massivement et définitivement adoptées par les systèmes d'exploitation les plus populaires. Des travaux en ce sens permettraient d'éclaircir cette question et d'estimer à quel point cette dissonance cognitive est préjudiciable selon les caractéristiques du découplage entre espace moteur et visuel.

Investiguer les variations de charge cognitive avec l'ISPIf

Lors de l'amélioration de techniques d'interaction existantes, un souci récurrent des concepteurs est de limiter la charge cognitive supplémentaire que cette amélioration va générer chez l'utilisateur. Ce questionnement est d'autant plus nécessaire que le contexte applicatif est critique, pour l'avionique ou la chirurgie par exemple. Certains effets doivent être évités, comme distraire l'utilisateur de sa tâche principale, entraîner des problèmes temporaires de mémorisation, ou générer un stress cognitif préjudiciable. La mesure de la charge cognitive est donc d'un intérêt majeur pour l'étude des techniques d'interaction homme-machine. Cependant, la charge cognitive est un concept difficilement mesurable objectivement. Selon la théorie de la charge cognitive [128], celle-ci dépend de la complexité des informations présentées à l'utilisateur ainsi que des liens entre les différentes informations et entre les différentes sources d'information. Elle peut être comprise comme

la proportion de la mémoire de travail de l'utilisateur accaparée par l'utilisation d'un système d'information [89, 127]. Une pratique simple à mettre en œuvre consiste à utiliser des questionnaires, comme le NASA Task-Load Index [63] pour obtenir une estimation subjective de la charge cognitive.

Les avancées dans le domaine des Interfaces Cerveau-Ordinateur (ICO ou BCI en anglais pour "Brain-Computer Interfaces") permettent d'étudier de plus en plus finement l'activité cérébrale d'utilisateurs accomplissant des tâches diverses. Récemment, l'Imagerie Spectroscopique Proche Infrarouge fonctionnelle (ISPIf, ou fNIRS en anglais pour "functional Near InfraRed Spectroscopy") consiste à estimer l'activité cérébrale en mesurant l'oxygénation d'une zone du cerveau. Une plus grande oxygénation signifie une plus grande activité cérébrale dans la zone étudiée. Cette technique d'ICO est abordable et mobile, contrairement à d'autres techniques d'ICO comme l'IRM fonctionnelle. Plusieurs travaux l'utilisent déjà pour moduler en temps réel des techniques d'interaction. Par exemple, Solovey et al. [123] présentent en 2012 Brainput, un système d'interaction humain-robot qui utilise l'ISPIf pour détecter les différents états cognitifs de l'utilisateur et adapter le comportement du robot. En 2014, Afergan et al. [4] présentent une variante de Bubble Cursor qui module la taille étendue des cibles en fonction de l'activité cérébrale détectée. Ces résultats indiquent que l'ISPIf peut être partie prenante d'une technique d'interaction.

Une piste de recherche sur les techniques d'expansion de cibles, les techniques de facilitation du pointage ou même plus largement les techniques de pointage, est d'utiliser l'ISPIf pour mesurer objectivement la charge cognitive à l'utilisation de différentes techniques. En effet, l'ISPIf est une des rares technologies capable de fournir en temps réel une mesure objective de la charge cognitive [3, 105]. Elle représente une opportunité pour la recherche en IHM, particulièrement pour les systèmes critiques, d'ajouter à la palette des outils de mesure de performance une dimension éminemment pertinente.

Valider la sélection d'une cible via d'autres modalités d'interaction

Une dernière piste de recherche que nous présentons est une suite directe des travaux menés. Parmi les trois phases d'une tâche de sélection de cible, la phase d'atteinte de la cible par le curseur (notée pointage au chapitre 2 et phase de transfert selon le modèle conceptuel de performance présenté au chapitre 7), a été intensivement étudiée, ainsi que les techniques visant à l'optimiser en agrandissant la zone active des cibles. La phase de validation de la sélection, a également fait l'objet d'un travail de recherche, notamment lors de la création du modèle de performance (chapitre 7), mais également par la création de Medical TapTap (chapitre 9). Une suite possible de ce travail consiste à étudier plus avant la phase de validation de la sélection et les techniques d'interaction multimodale adaptées au contexte chirurgical pour valider une sélection de cible. Les techniques d'interaction gestuelle comme Medical TapTap font partie des techniques candidates mais de nombreuses autres techniques d'interaction, existantes ou à créer, peuvent être pertinentes pour une utilisation per-opératoire.

Annexe A

System Usability Scale

	Pas du tout d'accord	Tout-à-fait d'accord
Je pense que j'aimerais utiliser ce type de système fréquemment.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Je trouve que ce système est inutilement complexe.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
J'ai trouvé que le système était facile à utiliser.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Je pense que j'aurais besoin de l'aide d'une personne expérimentée pour utiliser ce système.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
J'ai trouvé que les différentes fonctions du système étaient bien intégrées.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
J'ai trouvé qu'il y avait trop d'incohérences dans ce système.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
J'imagine que plusieurs personnes apprendront vite à utiliser le système.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
J'ai trouvé que le système était pesant, lourd, lors de son utilisation.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Je me suis senti en confiance lors de l'utilisation du système.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
J'ai eu besoin d'apprendre beaucoup de choses avant d'utiliser le système.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

FIGURE A.1 – Exemple de réponse d'un participant de l'expérience 2 à la version française du questionnaire SUS (System Usability Scale) [14]. Cette version est utilisée dans les expériences 1 et 2.

Annexe B

Questionnaire comparatif de l'expérience 1

Questionnaire final comparatif des 4 couples *DispositifDePointage* \times *TechniquesDePointage* de l'expérience 1.

Questionnaire

Classez les 4 techniques (*ex-æquo possibles*):

- avec un pointeur + zonage statique (P+ZS)
- avec un pointeur + curseur dynamique (P+CD)
- à main nue + zonage statique (MN+ZS)
- à main nue + curseur dynamique (MN+CD)

Par ordre de préférence (votre/vos préféré(s) en premier) :

- 1.
- 2.
- 3.
- 4.

De la plus rapide à la plus lente :

- 1.
- 2.
- 3.
- 4.

Indépendamment du système de pointage (main nue ou pointeur), vous trouvez que la technique du **zonage statique** est (*cochez la case adéquate*) :

- Utile

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Efficace

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Perturbante

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Agréable

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Aidante à la perception des cibles

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Fatigante

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Satisfaisante

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

Indépendamment du système de pointage (main nue ou pointeur), vous trouvez que la technique du **curseur dynamique** est :

- Utile

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Efficace

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Perturbante

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Agréable

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Aidante à la perception des cibles

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Fatigante

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Satisfaisante

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

Que proposeriez-vous pour améliorer ces techniques ?

- Zonage statique :

- Curseur dynamique :

Laquelle vous semble la plus intuitive (naturelle) ? Pourquoi ?

Avez-vous eu des stratégies ?

- Zonage statique :

- Curseur dynamique :

Indépendamment des techniques d'aide visuelle (zonage statique et curseur dynamique), vous trouvez que le **pointeur** est :

- Efficace

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Agréable

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Fatigant

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Satisfaisant

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Précis

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Rapide

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Pratique

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

Indépendamment des techniques d'aide visuelle (zonage statique et curseur dynamique), vous trouvez que le **hand-tracking** (suivi de la main nue) est :

- Efficace

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Agréable

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Fatigant

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Satisfaisant

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Précis

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Rapide

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

- Pratique

« Pas d'accord du tout »

« Tout-à-fait d'accord »

--	--	--	--	--

Ne pas remplir SVP

ID :

H :

Séq :

Autre :

Annexe C

Rappel visuel des techniques de l'expérience 2

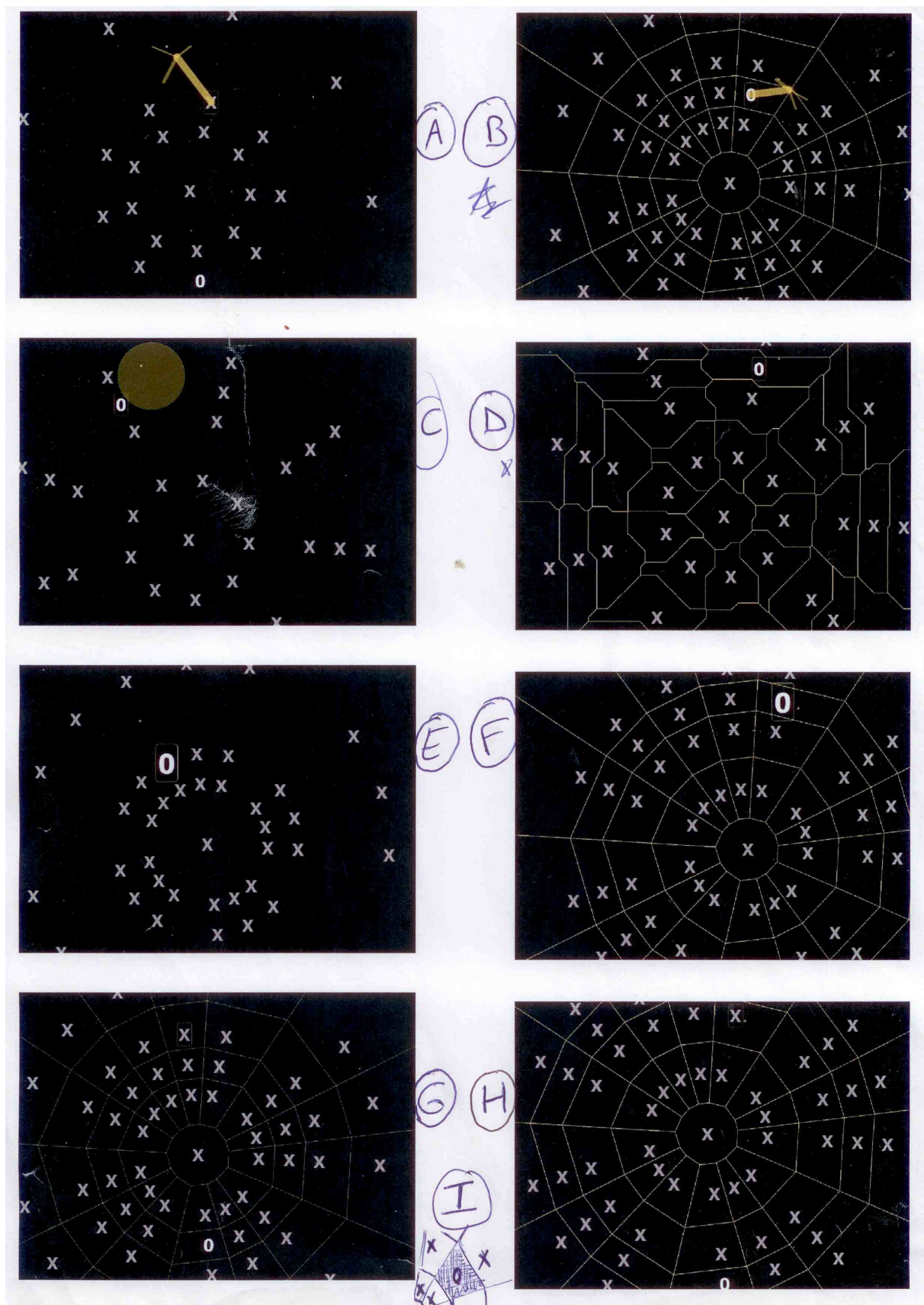


FIGURE C.1 – Rappel visuel des techniques d'expansion de cibles utilisées dans l'expérience 2. Ce rappel a permis aux participants de classer les techniques par leur lettre sans les confondre.

Annexe D

Version simplifiée du NASA-TLX (Task Load Index)

<p>Exigence Mental</p> <p>Fiable Elevée</p>	<p>Effort</p> <p>Fiable Elevée</p>
<p>Exigence physique</p> <p>Fiable Elevée</p>	<p>Performance</p> <p>Fiable Elevée</p>
<p>Exigence temporelle</p> <p>Fiable Elevée</p>	<p>Frustration</p> <p>Fiable Elevée</p>

FIGURE D.1 – Ce court questionnaire (moitié de la page) est extrait du NASA Task Load Index [63]. Il permet d'évaluer la charge de travail ressentie par les participants lors de l'utilisation des différentes techniques d'expansion de cibles.

Bibliographie

- [1] J. Accot and S. Zhai. Beyond fitts' law : models for trajectory-based hci tasks. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 295–302. ACM, 1997.
- [2] J. Accot and S. Zhai. Refining fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 193–200. ACM, 2003.
- [3] D. Afergan, E. M. Peck, E. T. Solovey, A. Jenkins, S. W. Hincks, E. T. Brown, R. Chang, and R. J. Jacob. Dynamic difficulty using brain metrics of workload. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3797–3806. ACM, 2014.
- [4] D. Afergan, T. Shibata, S. W. Hincks, E. M. Peck, B. F. Yuksel, R. Chang, and R. J. Jacob. Brain-based target expansion. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 583–593. ACM, 2014.
- [5] B. Agarwal and W. Stuerzlinger. Widgetlens : A system for adaptive content magnification of widgets. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, page 2. British Computer Society, 2013.
- [6] D. Ahlström, M. Hitz, and G. Leitner. An evaluation of sticky and force enhanced targets in multi target situations. In *Proceedings of the 4th Nordic conference on Human-computer interaction : changing roles*, pages 58–67. ACM, 2006.
- [7] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6) :716–723, 1974.
- [8] J. Alexander, T. Han, W. Judd, P. Irani, and S. Subramanian. Putting your best foot forward : investigating real-world mappings for foot-based gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1229–1238. ACM, 2012.
- [9] C. Appert, O. Chapuis, and E. Pietriga. High-precision magnification lenses. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 273–282. ACM, 2010.
- [10] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3) :121–136, 2013.

- [11] T. Asano, E. Sharlin, Y. Kitamura, K. Takashima, and F. Kishino. Predictive interaction using the delphian desktop. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 133–141. ACM, 2005.
- [12] R. Balakrishnan. “beating” fitts’ law : virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, 61(6) :857–874, 2004.
- [13] A. Banerjee, J. Burstyn, A. Girouard, and R. Vertegaal. Pointable : an in-air pointing technique to manipulate out-of-reach targets on tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 11–20. ACM, 2011.
- [14] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human-Computer Interaction*, 24(6) :574–594, 2008.
- [15] S. Bateman, R. L. Mandryk, C. Gutwin, and R. Xiao. Analysis and comparison of target assistance techniques for relative ray-cast pointing. *International Journal of Human-Computer Studies*, 71(5) :511–532, 2013.
- [16] S. Bateman, R. L. Mandryk, T. Stach, and C. Gutwin. Target assistance for subtly balancing competitive play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2355–2364. ACM, 2011.
- [17] O. Bau and W. E. Mackay. Octopocus : a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 37–46. ACM, 2008.
- [18] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, A. Zierlinger, et al. Drag-and-pop and drag-and-pick : Techniques for accessing remote screen content on touch-and pen-operated systems. In *Proceedings of INTERACT*, volume 3, pages 57–64, 2003.
- [19] P. Baudisch, A. Zotov, E. Cutrell, and K. Hinckley. Starburst : a target expansion algorithm for non-uniform target distributions. In *Proceedings of the working conference on Advanced visual interfaces*, pages 129–137. ACM, 2008.
- [20] M. Beaudouin-Lafon. 50 ans d’interaction homme-machine : retours vers le futur. *Interstices*, 2016.
- [21] M. Beaudouin-Lafon, S. Huot, H. Olafsdottir, and P. Dragicevic. Glidecursor : Pointing with an inertial cursor. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 49–56. ACM, 2014.
- [22] A. Benabid, P. Cinquin, S. Lavalley, J. Le Bas, J. Demongeot, and J. De Rougemont. Computer-driven robot for stereotactic surgery connected to ct scan and magnetic resonance imaging. *Stereotactic and Functional Neurosurgery*, 50(1-6) :153–154, 1987.

- [23] A. Bigdelou, L. Schwarz, and N. Navab. An adaptive solution for intra-operative gesture-based human-machine interaction. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 75–84. ACM, 2012.
- [24] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing : improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 519–526. ACM, 2004.
- [25] R. Blanch and M. Ortega. Rake cursor : improving pointing performance with concurrent input channels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1415–1418. ACM, 2009.
- [26] R. Blanch and M. Ortega. Benchmarking pointing techniques with distractors : adding a density factor to fitts’ pointing paradigm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1629–1638. ACM, 2011.
- [27] J.-D. Boissonnat, F. Nielsen, and R. Nock. Bregman voronoi diagrams. *Discrete & Computational Geometry*, 44(2) :281–307, 2010.
- [28] D. A. Bowman, C. A. Wingrave, J. Campbell, V. Ly, and C. Rhoton. Novel uses of pinch glovesTM for virtual environment interaction techniques. *Virtual Reality*, 6(3) :122–129, 2002.
- [29] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference : a practical information-theoretic approach*. Springer Science & Business Media, 2003.
- [30] K. P. Burnham and D. R. Anderson. Multimodel inference : understanding aic and bic in model selection. *Sociological methods & research*, 33(2) :261–304, 2004.
- [31] G. Casiez and N. Roussel. No more bricolage ! : methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 603–614. ACM, 2011.
- [32] G. Casiez, D. Vogel, R. Balakrishnan, and A. Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction*, 23(3) :215–250, 2008.
- [33] J. Cassell. On control, certitude, and the “paranoia” of surgeons. *Culture, medicine and psychiatry*, 11(2) :229–249, 1987.
- [34] O. Chapuis, J.-B. Labrune, and E. Pietriga. Dynaspot : speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1391–1400. ACM, 2009.
- [35] A. Cockburn and P. Brock. Human on-line response to visual and motor target expansion. In *Proceedings of Graphics Interface 2006*, pages 81–87. Canadian Information Processing Society, 2006.

- [36] A. Cockburn, P. Quinn, C. Gutwin, G. Ramos, and J. Looser. Air pointing : Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies*, 69(6) :401–414, 2011.
- [37] M. Collomb, M. Hascoët, P. Baudisch, and B. Lee. Improving drag-and-drop on wall-size displays. In *Proceedings of Graphics interface 2005*, pages 25–32. Canadian Human-Computer Communications Society, 2005.
- [38] C. Coutrix, W. Delamare, M. Guillon, T. Kurata, F. Leitner, L. Nigay, and T. Vincent. Techniques de pointage à distance : Cibles numériques et cibles physique. In *UbiMob2014 : 10èmes journées francophones Mobilité et Ubiquité*, pages 5–10, 2014.
- [39] A. Crossan, S. Brewster, and A. Ng. Foot tapping for mobile interaction. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, pages 418–422. British Computer Society, 2010.
- [40] E. Crossman and P. Goodeve. Feedback control of hand-movement and fitts’ law. *The Quarterly Journal of Experimental Psychology*, 35(2) :251–278, 1983.
- [41] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations : Applications and algorithms. *SIAM review*, 41(4) :637–676, 1999.
- [42] E. Dubois. *Chirurgie augmentée : un cas de réalité augmentée ; Conception et réalisation centrées sur l'utilisateur*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2001.
- [43] E. Dubois, L. Nigay, J. Troccaz, O. Chavanon, and L. Carrat. Classification space for augmented surgery. an augmented reality case study. In *Humancomputer interaction, INTERACT’99 : IFIP TC. 13 International Conference on Human-Computer Interaction, 30th August-3rd September 1999, Edinburgh, UK*, volume 1, page 353, 1999.
- [44] L. C. Ebert, G. Hatch, M. J. Thali, and S. Ross. Invisible touch—control of a dicom viewer with finger gestures using the kinect depth camera. *Journal of Forensic Radiology and Imaging*, 1(1) :10–14, 2013.
- [45] L. Findlater, A. Jansen, K. Shinohara, M. Dixon, P. Kamb, J. Rakita, and J. O. Wobbrock. Enhanced area cursors : reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, pages 153–162. ACM, 2010.
- [46] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6) :381, 1954.
- [47] R. Fraher. Cursor dynamics : aesthetic exploration of the bubble cursor. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction : Making Sense Through Design*, pages 765–766. ACM, 2012.

- [48] K. Fukahori, D. Sakamoto, and T. Igarashi. Exploring subtle foot plantar-based gestures with sock-placed pressure sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3019–3028. ACM, 2015.
- [49] L. Gallo, A. P. Placitelli, and M. Ciampi. Controller-free exploration of medical image data : Experiencing the kinect. In *Computer-based medical systems (CBMS), 2011 24th international symposium on*, pages 1–6. IEEE, 2011.
- [50] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, and P. Arias. Metrological evaluation of microsoft kinect and asus xtion sensors. *Measurement*, 46(6) :1800–1806, 2013.
- [51] C. Graetzel, T. Fong, S. Grange, and C. Baur. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care*, 12(3) :245–257, 2004.
- [52] S. Grange, T. Fong, and C. Baur. M/oris : a medical/operating room interaction system. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 159–166. ACM, 2004.
- [53] T. Grossman and R. Balakrishnan. The bubble cursor : enhancing target acquisition by dynamic resizing of the cursor’s activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–290. ACM, 2005.
- [54] T. Grossman and R. Balakrishnan. A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(3) :435–459, 2005.
- [55] T. Grossman, N. Kong, and R. Balakrishnan. Modeling pointing at targets of arbitrary shapes. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 463–472. ACM, 2007.
- [56] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing : a complement to bitmap pointing in guis. In *Proceedings of Graphics Interface 2004*, pages 9–16. Canadian Human-Computer Communications Society, 2004.
- [57] M. Guillon, F. Leitner, and L. Nigay. Démonstration de VTE : une technique de pointage à distance. In *IHM’14, 26e conférence francophone sur l’Interaction Homme-Machine*, pages 8–9, 2014.
- [58] M. Guillon, F. Leitner, and L. Nigay. Static Voronoi-based target expansion technique for distant pointing. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 41–48. ACM, 2014.
- [59] M. Guillon, F. Leitner, and L. Nigay. Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2777–2786. ACM, 2015.

- [60] M. Guillon, F. Leitner, and L. Nigay. Target expansion lens : It is not the more visual feedback the better ! In *Proceedings of the 2016 International Working Conference on Advanced Visual Interfaces*, pages 52–59. ACM, 2016.
- [61] C. Gutwin. Improving focus targeting in interactive fisheye views. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 267–274. ACM, 2002.
- [62] F. Haque, M. Nancel, and D. Vogel. Myopoint : Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3653–3656. ACM, 2015.
- [63] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index) : Results of empirical and theoretical research. *Advances in psychology*, 52 :139–183, 1988.
- [64] B. Hartmann, M. Benson, A. Junger, L. Quinzio, R. Röhrig, B. Fengler, U. W. Färber, B. Wille, and G. Hempelmann. Computer keyboard and mouse as a reservoir of pathogens in an intensive care unit. *Journal of clinical monitoring and computing*, 18(1) :7–12, 2004.
- [65] M. Hascoët. Throwing models for large displays. In *HCI’03 : 11th International Conference on Human Computer Interaction*, volume 2, pages pp–73. British HCI Group, 2008.
- [66] M. Hertzum and K. Hornbæk. Touchgrid : Touchpad pointing by recursively mapping taps to smaller display regions. *Behaviour & Information Technology*, 24(5) :337–346, 2005.
- [67] M. Hertzum and K. Hornbæk. Input techniques that dynamically change their cursor activation area : A comparison of bubble and cell cursors. *International Journal of Human-Computer Studies*, 65(10) :833–851, 2007.
- [68] W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4(1) :11–26, 1952.
- [69] R. Hyman. Stimulus information as a determinant of reaction time. *Journal of experimental psychology*, 45(3) :188, 1953.
- [70] I. ISO. 9241-9 ergonomic requirements for office work with visual display terminals (vdts)-part 9 : Requirements for non-keyboard input devices (fdis-final draft international standard). *International Organization for Standardization*, 2000.
- [71] R. Johnson, K. O’Hara, A. Sellen, C. Cousins, and A. Criminisi. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3323–3332. ACM, 2011.

- [72] P. Kabbash and W. A. Buxton. The “prince” technique : Fitts’ law and selection using area cursors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 273–279. ACM Press/Addison-Wesley Publishing Co., 1995.
- [73] P. Katz. Ritual in the operating room. *Ethnology*, 20(4) :335–350, 1981.
- [74] A. Kendon. *Gesture : Visible action as utterance*. Cambridge University Press, 2004.
- [75] M. Kobayashi and T. Igarashi. Ninja cursors : using multiple cursors to assist target acquisition on large screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 949–958. ACM, 2008.
- [76] R. Kopper, D. A. Bowman, M. G. Silva, and R. P. McMahan. A human motor behavior model for distal pointing tasks. *International journal of human-computer studies*, 68(10) :603–615, 2010.
- [77] C. Kuwabara, K. Yamamoto, I. Kuramoto, Y. Tsujino, and M. Minakuchi. Ghost-hunting : a cursor-based pointing technique with picture guide indication of the shortest path. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, pages 85–86. ACM, 2013.
- [78] J. Laukkanen, P. Isokoski, and K.-J. R  ih  . The cone and the lazy bubble : two efficient alternatives between the point cursor and the bubble cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 309–312. ACM, 2008.
- [79] I. S. MacKenzie. Fitts’ law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1) :91–139, 1992.
- [80] R. L. Mandryk and C. Gutwin. Perceptibility and utility of sticky targets. In *Proceedings of graphics interface 2008*, pages 65–72. Canadian Information Processing Society, 2008.
- [81] D. C. McCallum and P. Irani. Arc-pad : absolute+ relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 153–156. ACM, 2009.
- [82] M. McCullough, H. Xu, J. Michelson, M. Jackoski, W. Pease, W. Cobb, W. Kalescky, J. Ladd, and B. Williams. Myo arm : swinging to explore a ve. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, pages 107–113. ACM, 2015.
- [83] M. McGuffin and R. Balakrishnan. Acquisition of expanding targets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 57–64. ACM, 2002.

- [84] M. J. McGuffin and R. Balakrishnan. Fitts' law and expanding targets : Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(4) :388–422, 2005.
- [85] M. J. McGuffin and R. Balakrishnan. Fitts' law and expanding targets : Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(4) :388–422, 2005.
- [86] H. M. Mentis, K. O'Hara, G. Gonzalez, A. Sellen, R. Corish, A. Criminisi, R. Trivedi, and P. Theodore. Voice or gesture in the operating room. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 773–780. ACM, 2015.
- [87] H. M. Mentis, K. O'Hara, A. Sellen, and R. Trivedi. Interaction proxemics and image use in neurosurgery. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 927–936. ACM, 2012.
- [88] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright, and J. Keith Smith. Optimality in human motor performance : Ideal control of rapid aimed movements. *Psychological review*, 95(3) :340, 1988.
- [89] R. Moreno and B. Park. Cognitive load theory : Historical development and relation to other theories. *Cognitive load theory*, pages 9–28, 2010.
- [90] M. E. Mott and J. O. Wobbrock. Beating the bubble : using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 733–742. ACM, 2014.
- [91] B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting at a distance : measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 33–40. ACM, 2002.
- [92] M. Nancel. *Designing and combining mid-air interaction techniques in large display environments*. PhD thesis, Paris 11, 2012.
- [93] M. Nancel, O. Chapuis, E. Pietriga, X.-D. Yang, P. P. Irani, and M. Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 831–840. ACM, 2013.
- [94] M. Nancel, E. Pietriga, O. Chapuis, and M. Beaudouin-Lafon. Mid-air pointing on ultra-walls. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(5) :21, 2015.
- [95] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 177–186. ACM, 2011.

- [96] K. Nieuwenhuizen, D. Aliakseyeu, and J.-B. Martens. Insight into goal-directed movements : beyond fitts' law. *Human-Computer Interaction-INTERACT 2009*, pages 274–287, 2009.
- [97] D. A. Norman. *The psychology of everyday things*. Basic books, 1988.
- [98] M. T. Nouei, A. V. Kamyad, A. R. Soroush, and S. Ghazalbash. A comprehensive operating room information system using the kinect sensors and rfid. *Journal of clinical monitoring and computing*, 29(2) :251–261, 2015.
- [99] J.-Y. Oh and W. Stuerzlinger. Laser pointers as collaborative pointing devices. In *Graphics Interface*, volume 2002, pages 141–149, 2002.
- [100] K. O'Hara, G. Gonzalez, A. Sellen, G. Penney, A. Varnavas, H. Mentis, A. Criminisi, R. Corish, M. Rouncefield, N. Dastur, et al. Touchless interaction in surgery. *Communications of the ACM*, 57(1) :70–77, 2014.
- [101] A. Olivia, M. L. Mack, M. Shrestha, and A. Peeper. Identifying the perceptual dimensions of visual complexity of scenes. In *Proceedings of the Cognitive Science Society*, volume 26, 2004.
- [102] D. Onceanu and A. J. Stewart. Direct surgeon control of the computer in the operating room. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 121–128. Springer, 2011.
- [103] T. Pakkanen and R. Raisamo. Appropriateness of foot interaction for non-accurate spatial tasks. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1123–1126. ACM, 2004.
- [104] J. Parker, R. Mandryk, M. Nunes, and K. Inkpen. Tractorbeam selection aids : improving target acquisition for pointing input on tabletop displays. *Human-Computer Interaction-INTERACT 2005*, pages 80–93, 2005.
- [105] E. M. Peck, D. Afergan, B. F. Yuksel, F. Lalooses, and R. J. Jacob. Using fnirs to measure mental workload in the real world. In *Advances in physiological computing*, pages 117–139. Springer, 2014.
- [106] E. Pietriga and C. Appert. Sigma lenses : focus-context transitions combining space, time and translucence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1352. ACM, 2008.
- [107] R. Plamondon and A. M. Alimi. Speed/accuracy trade-offs in target-directed movements. *Behavioral and brain sciences*, 20(02) :279–303, 1997.
- [108] A. Rajavenkatanarayanan, Y. V. Surathi, A. R. Babu, and M. Papakostas. Myo-drive : A new way of interacting with mobile devices. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 18. ACM, 2016.

- [109] G. Ramos, A. Cockburn, R. Balakrishnan, and M. Beaudouin-Lafon. Pointing lenses : facilitating stylus input through visual-and motor-space magnification. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 757–766. ACM, 2007.
- [110] G. Ren and E. O’Neill. 3d marking menu selection with freehand gestures. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 61–68. IEEE, 2012.
- [111] R. Richter and M. Alexa. Mahalanobis centroidal voronoi tessellations. *Computers & Graphics*, 46 :48–54, 2015.
- [112] G. C. S. Ruppert, L. O. Reis, P. H. J. Amorim, T. F. de Moraes, and J. V. L. da Silva. Touchless gesture user interface for interactive image visualization in urological surgery. *World journal of urology*, 30(5) :687–691, 2012.
- [113] N. Sangsuriyachot and M. Sugimoto. Novel interaction techniques based on a combination of hand and foot gestures in tabletop environments. In *Proceedings of the 10th asia pacific conference on Computer human interaction*, pages 21–28. ACM, 2012.
- [114] W. Saunders and D. Vogel. Tap-kick-click : Foot interaction for a standing desk. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 323–333. ACM, 2016.
- [115] J. Sauro and J. R. Lewis. Average task times in usability tests : what to report ? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2347–2350. ACM, 2010.
- [116] E. Schmider, M. Ziegler, E. Danay, L. Beyer, and M. Bühner. Is it really robust ? *Methodology*, 2010.
- [117] D. Schmidt, R. Ramakers, E. W. Pedersen, J. Jasper, S. Köhler, A. Pohl, H. Rantzs, A. Rau, P. Schmidt, C. Sterz, et al. Kickables : tangibles for feet. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3143–3152. ACM, 2014.
- [118] L. A. Schwarz, A. Bigdelou, and N. Navab. Learning gestures for customizable human-computer interaction in the operating room. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 129–136. Springer, 2011.
- [119] J. Scott, D. Dearman, K. Yatani, and K. N. Truong. Sensing foot gestures from the pocket. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, pages 199–208. ACM, 2010.
- [120] S. C. Seow. Information theoretic models of hci : a comparison of the hick-hyman law and fitts’ law. *Human-Computer Interaction*, 20(3) :315–352, 2005.
- [121] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1) :3–55, 2001.

- [122] K. Smith-Gratto and M. M. Fisher. Gestalt theory : a foundation for instructional screen design. *Journal of Educational Technology Systems*, 27(4) :361–371, 1999.
- [123] E. Solovey, P. Schermerhorn, M. Scheutz, A. Sassaroli, S. Fantini, and R. Jacob. Brainput : enhancing interactive systems with streaming fnirs brain input. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 2193–2202. ACM, 2012.
- [124] R. W. Soukoreff, J. Zhao, and X. Ren. The entropy of a rapid aimed movement : Fitts’ index of difficulty versus shannon’s entropy. In *IFIP Conference on Human-Computer Interaction*, pages 222–239. Springer, 2011.
- [125] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.
- [126] X. Su, O. K.-C. Au, and R. W. Lau. The implicit fan cursor : a velocity dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 753–762. ACM, 2014.
- [127] J. Sweller. Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2) :123–138, 2010.
- [128] J. Sweller and P. Chandler. Why some material is difficult to learn. *Cognition and instruction*, 12(3) :185–233, 1994.
- [129] E. Tse, M. Hancock, and S. Greenberg. Speech-filtered bubble ray : improving target acquisition on display walls. In *Proceedings of the 9th international conference on Multimodal interfaces*, pages 307–314. ACM, 2007.
- [130] G. P. van Galen and W. P. de Jong. Fitts’ law as the outcome of a dynamic noise filtering model of motor control. *Human movement science*, 14(4) :539–571, 1995.
- [131] L. Vanacken, T. Grossman, and K. Coninx. Multimodal selection techniques for dense and occluded 3d virtual environments. *International Journal of Human-Computer Studies*, 67(3) :237–255, 2009.
- [132] J. Vermeulen, K. Luyten, E. van den Hoven, and K. Coninx. Crossing the bridge over norman’s gulf of execution : revealing feedforward’s true identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1931–1940. ACM, 2013.
- [133] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 33–42. ACM, 2005.
- [134] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler. Gestix : a doctor-computer sterile gesture interface for dynamic environments. In *Soft Computing in Industrial Applications*, pages 30–39. Springer, 2007.

- [135] J. Wagner, M. Nancel, S. G. Gustafson, S. Huot, and W. E. Mackay. Body-centric design space for multi-surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1299–1308. ACM, 2013.
- [136] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. In *ACM transactions on graphics (TOG)*, volume 28, page 63. ACM, 2009.
- [137] C. Wingrave and D. Bowman. Baseline factors for raycasting selection. In *Proceedings of HCI International*, 2005.
- [138] A. Worden, N. Walker, K. Bharat, and S. Hudson. Making computers easier for older adults to use : area cursors and sticky icons. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 266–271. ACM, 1997.
- [139] K. Yanati. Statistical methods for hci research, 2014. <http://yatani.jp/teaching/doku.php?id=hcistats:start>.
- [140] C. Yu, Y. Shi, R. Balakrishnan, X. Meng, Y. Suo, M. Fan, and Y. Qin. The satellite cursor : achieving magic pointing without gaze tracking using multiple cursors. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, pages 163–172. ACM, 2010.
- [141] S. Zhai, W. Buxton, and P. Milgram. The “silk cursor” : investigating transparency for 3d target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 459–464. ACM, 1994.
- [142] S. Zhai, S. Conversy, M. Beaudouin-Lafon, and Y. Guiard. Human on-line response to target expansion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 177–184. ACM, 2003.
- [143] S. Zhai, P. Milgram, and W. Buxton. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 308–315. ACM, 1996.
- [144] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 246–253. ACM, 1999.
- [145] X. Zhang, H. Zha, and W. Feng. Extending fitts’ law to account for the effects of movement direction on 2d pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3185–3194. ACM, 2012.