

THÈSE  
présentée pour obtenir le grade de  
DOCTEUR DU CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

Discipline : Informatique

préparée au Centre d'étude et de recherche en Informatique du CNAM  
dans le cadre de l'**EDITE de Paris**

Rodrigo ANDRADE BOTELHO DE ALMEIDA

Titre :

CONTRIBUTION AUX TECHNIQUES POUR ENRICHIR  
L'ESPACE MOTEUR ET L'ESPACE VISUEL  
DES DISPOSITIFS D'INTERACTION BUREAUTIQUE

Soutenue le 6 novembre 2009

**Directeur de thèse :**

M. CUBAUD Pierre                      Professeur au CNAM

**Rapporteurs :**

M. FEKETE Jean-Daniel            Directeur de Recherche à l'INRIA Saclay - Île-de-France  
M<sup>me</sup> HASCOËT Mountaz           Maître de Conférences HDR au LIRMM - Université Montpellier II

**Président du jury :**

M. NATKIN Stéphane                Professeur Titulaire de Chaire au CNAM

**Examineurs :**

M. LECOLINET Eric                   Maître de Conférences à l'ENST Paris  
M. TOPOL Alexandre                Maître de Conférences au CNAM



# Remerciements

Je remercie tout d'abord Stéphane Natkin pour avoir accepté de présider mon jury. C'est aussi grâce à la confiance qu'il m'a accordée en m'acceptant en DEA Conception d'Applications Multimédia que j'ai pu commencer mon aventure dans le monde de l'interaction homme-machine. Je remercie mon directeur de thèse, Pierre Cubaud, pour l'encadrement très attentif et disponible qu'il m'a dispensé pendant ces années au CNAM. Son enthousiasme pour la recherche, sa transdisciplinarité et son sens du dialogue m'ont inspiré et m'ont aidé à assumer et à développer mes propres idées. Je remercie également Alexandre Topol pour la générosité du temps qu'il a passé m'aidant à surmonter les obstacles rencontrés et partageant son savoir avec moi.

Je remercie les membres de mon jury, Mountaz Hascoët et Jean-Daniel Fekete, dont j'admire beaucoup les travaux, d'avoir accepté d'être rapporteurs et de la lecture soigneuse qu'ils ont faite de cette thèse ; leurs remarques ont été très constructives. Il en est de même pour Eric Lecolinet, qui a accepté de faire partie du jury et qui, à plus d'une reprise, m'a fait part avec sincérité de ses impressions quant à mes travaux de recherche.

Un grand merci aux amis qui m'ont tant aidé pendant la thèse - Thiago, Pedro et Shuo-Hsiu - ainsi qu'à Fatima-zahra et à Françoise pour leur aide dans la relecture ! Je remercie également toutes les personnes de la Bibliothèque centrale du CNAM avec qui j'ai travaillé avec grand plaisir pendant le doctorat. Je tiens à remercier plus particulièrement Mireille Le Van Ho, directrice de la bibliothèque, et mes collègues du CNUM : Françoise, Geneviève, Claire et Jean-Paul. Je ne pourrais pas ne pas citer mes amis et collègues du laboratoire qui ont rendu ce périple moins solitaire et plus enrichissant : Julien, Jérôme, Viviane, Jean-Paul, Jean-Frédéric, José, Chen, Guillaume, Xiang Qiu, Areti, Amélie, Hélène, Nicolas, Romain, Patrice. Mais aussi les amis en dehors du laboratoire qui m'ont beaucoup motivé et inspiré : Silvia, Otavio, Jean-Baptiste et Vincent !

C'est enfin aux personnes de ce dernier paragraphe que je consacre mes remerciements les plus chaleureux. A mon frère Eduardo pour son appui moral et pratique (je pense aux statistiques) et pour partager avec moi sa passion pour la science. A ma compagne Anne, pour sa compréhension, pour ses nombreuses aides qui m'ont permis d'achever cette entreprise et pour l'encouragement infailible et quotidien qu'elle m'a offert pendant tout le parcours. A mes parents pour le soutien inconditionnel qu'ils m'ont apporté depuis que j'ai décidé de poursuivre mes études en France. Même étant très loin, ils ont été très présents et indispensables pour que ce voyage se termine bien.



# Résumé

De nombreux travaux ont montré que, à l'origine des limitations de l'interaction bureautique, il y a un manque à la fois d'espace moteur et d'espace visuel. Cette thèse explore des moyens pour optimiser l'usage de ces espaces.

D'une part, à partir du constat que l'on contrôle la position et l'orientation d'un objet par un geste naturel, cette thèse étudie les bénéfices que peut offrir une souris dotée d'un capteur de rotation. Cette « souris rotative » permet à l'utilisateur de maîtriser avec aisance trois variables continues d'une tâche informatique. Un état de l'art présente des aspects perceptifs et moteurs des actions en question et les particularités ergonomiques et techniques d'un tel dispositif. Deux techniques d'interaction - visant à faciliter des tâches métier récurrentes - sont proposées : le réglage « quasi-intégral » et la « palette satellitaire ». Par ailleurs, une évaluation expérimentale compare la performance d'une souris rotative avec celle d'une souris traditionnelle.

D'autre part, ce travail se penche sur les questions de la visualisation de documents dans le contexte des bibliothèques numériques. D'abord, il examine l'apport et la faisabilité technique de l'utilisation d'un dispositif d'affichage immersif pour la navigation dans un catalogue de titres virtuel. Puis, afin de faciliter l'inspection massive d'un lot de pages numérisées, il avance des techniques de visualisation zoomables et multi-focales. Ces dernières permettent, dans une recherche d'anomalies, de saisir vite les caractéristiques visuelles de quelques centaines de pages. Et cela grâce à un va-et-vient entre la vue d'ensemble et la navigation panoramique des détails.

**Mots clés :** interaction 3D, dispositifs d'entrée, techniques d'interaction, visualisation de documents, bibliothèques numériques, dispositifs d'affichage immersif



# Abstract

Past research has suggested that among the reasons for the limitations of present desktop interaction style is the lack of both motor and visual space. The goal of this thesis is to optimize the use of such spaces.

Based on the fact that one can control an object's position and orientation through a natural movement, the first main contribution of this thesis is to explore the advantages of enhancing the sensing of the standard mouse through a rotation sensor. This « rotary mouse » allows one to easily control three continuous variables of a computer task. A survey presents the perceptual and motor issues of some rotary manipulations and also the technical and ergonomic requirements of such device. Two interaction techniques, aimed to simplify repetitive tasks, are proposed : the « nearly-integral selection » and the « satellite palette ». Furthermore, an experimental evaluation compares the performance of the rotary mouse with that of a standard one.

The other main contribution of this work is to investigate document visualization issues in the context of digital libraries. First, it analyses the advantages and the technical feasibility of integrating an immersive display to an interface aimed to support navigation in a virtual catalog. Second, in order to inspect the quality of a batch of digitized pages, it explores some zoomable and multi-focal visualization techniques. The overview and the panoramic detail browsing enabled by such techniques try to help users, which have to identify the flaws resulted from the digitization process, to quickly grasp the visual characteristics of a large set of pages.

**Keywords :** 3D interaction, input devices, interaction techniques, document visualization, digital libraries, immersive display devices





# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Limitations des interfaces actuelles . . . . .	2
1.2	Espace moteur et espace visuel . . . . .	5
1.2.1	Optimiser l'espace moteur de la souris . . . . .	5
1.2.2	Elargir l'espace visuel des bibliothèques numériques . . . . .	9
1.3	Contexte de la thèse . . . . .	10
1.4	Organisation de la thèse . . . . .	12
1.5	Publication associées . . . . .	14
<b>2</b>	<b>Etat de l'art de la manipulation rotative : dispositifs et aspects humains</b>	<b>17</b>
2.1	Pistes pour l'exploration de la rotation . . . . .	18
2.1.1	Configurations temporelles des manipulations rotatives . . . . .	18
2.1.2	Manipulation multidimensionnelle . . . . .	19
2.1.3	Manipulation « absolue » . . . . .	23
2.1.4	Les types de mouvements manuels . . . . .	25
2.1.5	Analogie avec le bi-manuel . . . . .	29
2.2	Contrôles rotatifs . . . . .	31
2.2.1	Types de dispositifs rotatifs . . . . .	31
2.2.2	Propriétés matérielles . . . . .	33
2.2.3	<i>Mappings</i> . . . . .	34
2.3	Les périphériques d'entrée rotatifs . . . . .	35
2.3.1	Caractériser les périphériques d'entrée manuels . . . . .	35
2.3.2	Les souris enrichies . . . . .	40
2.3.3	Les souris rotatives . . . . .	44
2.4	Des interfaces graphiques rotatives . . . . .	56

<b>3 Manipulation de documents dans un espace 3D</b>	<b>59</b>
3.1 Des techniques 3D pour la gestion de fenêtres . . . . .	60
3.1.1 Une correspondance directe entre dispositif et fenêtre . . . . .	61
3.2 Construction d'un prototype de souris rotative . . . . .	62
3.3 Une étude comparative . . . . .	64
3.3.1 Hypothèses . . . . .	66
3.3.2 Méthode . . . . .	67
3.3.3 Résultats . . . . .	71
3.3.4 Discussion . . . . .	74
3.4 Conclusion et travaux futurs . . . . .	78
<b>4 Manipulation quasi-intégrale de <i>widgets</i> rotatifs</b>	<b>81</b>
4.1 Interaction réelle et avancée . . . . .	83
4.1.1 Boutons linéaires et boutons rotatifs . . . . .	84
4.2 Réglages graphiques appauvris . . . . .	84
4.2.1 Multiplexage temporel des contrôles physiques . . . . .	85
4.2.2 Réglage linéaire . . . . .	86
4.3 Une technique à trois degrés de liberté . . . . .	88
4.3.1 Transition quasi-intégrale . . . . .	88
4.3.2 Le prototype de la souris rotative . . . . .	89
4.3.3 Premiers essais . . . . .	90
4.3.4 Un modèle à trois états . . . . .	91
4.3.5 Visualisation des entrées et mise au point . . . . .	92
4.3.6 Retours de l'utilisation . . . . .	93
4.4 Travaux en perspective : sélectionner avec une grande précision . . . . .	93
4.4.1 Espace moteur et espace graphique . . . . .	94
4.4.2 Réglage rotatif de grande précision . . . . .	94
4.5 Conclusion . . . . .	96
<b>5 Une technique pour le changement d'outils à l'aide d'une palette rotative locale</b>	<b>97</b>
5.1 Problématique . . . . .	98
5.1.1 Contextes d'utilisation des palettes graphiques . . . . .	99
5.1.2 Inconvénients des palettes graphiques . . . . .	100

5.2	Approches connexes . . . . .	101
5.2.1	Survol des concepts . . . . .	101
5.2.2	Sélection via un capteur supplémentaire . . . . .	103
5.3	La palette satellitaire . . . . .	104
5.3.1	Propriétés de la technique . . . . .	105
5.3.2	Un prototype rotatif et « cliquable » . . . . .	106
5.3.3	Manipulation rotative . . . . .	107
5.3.4	Suivi visuel de la sélection . . . . .	107
5.3.5	Nombre d'outils . . . . .	109
5.4	Etude exploratoire : une palette de couleurs . . . . .	110
5.5	Travaux en perspective : activation gestuelle des outils . . . . .	112
5.5.1	Nombre et configuration des secteurs . . . . .	113
5.6	Conclusion . . . . .	114
<b>6</b>	<b>Navigation et immersion dans une collection de documents numériques</b>	<b>117</b>
6.1	Au-delà du catalogue de titres . . . . .	118
6.1.1	Exploration d'un catalogue et découverte fortuite . . . . .	119
6.1.2	Représentation visuelle des documents . . . . .	120
6.1.3	Organisation spatiale de la collection . . . . .	122
6.2	Expérience visuelle enrichie . . . . .	123
6.3	Un poste de consultation et lecture immersif . . . . .	126
6.3.1	Fonctionnement du périphérique d'affichage immersif . . . . .	126
6.3.2	Rendu d'une image grand angle . . . . .	126
6.3.3	Immersion visuelle dans la collection . . . . .	128
6.3.4	Affichage final et qualité visuelle . . . . .	130
6.3.5	Une stratégie « focus plus contexte » . . . . .	131
6.3.6	Implémentation du deuxième prototype . . . . .	132
6.4	Conclusion et travaux futurs . . . . .	133
<b>7</b>	<b>Techniques de visualisation pour le contrôle qualité de lots d'images numérisées</b>	<b>135</b>
7.1	Contrôle qualité des pages numérisées . . . . .	136
7.2	Vue globale et vue détaillée . . . . .	138
7.3	Visualisation massive statique d'une série de volumes . . . . .	141
7.4	Interface Grille de détails . . . . .	143

7.4.1	Interface Web . . . . .	144
7.5	Mur de pages . . . . .	146
7.5.1	Stratégies de navigation . . . . .	147
7.5.2	Scène et contrôle de la caméra . . . . .	148
7.5.3	Implémentation . . . . .	150
7.5.4	Retour des utilisateurs . . . . .	151
7.5.5	Mieux contrôler la vitesse du panorama. . . . .	151
7.6	Panorama de détails . . . . .	152
7.6.1	Paramètres de navigation . . . . .	153
7.6.2	Interface de navigation . . . . .	153
7.7	Conclusions . . . . .	154
<b>8</b>	<b>Conclusions et perspectives</b>	<b>157</b>
8.1	Contributions . . . . .	158
8.2	Perspectives . . . . .	160

# Liste des tableaux

3.1	Résultats pour chaque condition : durée moyenne, écart type et le nombre de fois où chaque technique a été la plus rapide . . . . .	72
4.1	Résumé des différences entre les caractéristiques de la manipulation d'un bouton rotatif et d'un bouton linéaire. . . . .	84
5.1	Résumé des recommandations données dans Kroemer et Grandjean (1997) concernant la taille des boutons et l'écart en degrés entre les options d'un <i>switch</i> rotatoire. . . . .	109
7.1	Liste de problèmes de numérisation en fonction du moment où ils apparaissent . . . . .	137
7.2	Résumé des caractéristiques de l'« échantillonnage visuel » procuré par chaque méthode de la Grille de détails. . . . .	147



# Table des figures

1.1	Continuum des contextes d'usage de l'informatique . . . . .	1
2.1	Les cinq types de configuration décrivant la temporalité d'une action composée . . . . .	19
2.2	Rotation et déplacement simultanées d'un objet . . . . .	20
2.3	Schémas d'une tâche intégrale et d'une tâche séparable . . . . .	21
2.4	Deux figures illustrant la « prise forte » et la « prise fine » . . . . .	26
2.5	La variation dans l'implication des doigts pendant une prise fine selon le diamètre de l'objet saisi . . . . .	27
2.6	Les types de mouvements manuels intrinsèques . . . . .	28
2.7	Illustration du modèle « pas rotatif » . . . . .	29
2.8	Illustration du modèle « pivotement digital » . . . . .	30
2.9	<i>Mappings</i> et correspondances utilisés par les dispositifs rotatifs . . . . .	34
2.10	Types de retour d'une opération de rotation . . . . .	35
2.11	Le modèle à trois états indiquant les états et les transitions d'une souris et d'une tablette tactile . . . . .	37
2.12	Les propriétés des dispositifs physiques avancées par Lipscomb et Pique . . . . .	39
2.13	Les souris enrichies présentées dans ce chapitre . . . . .	41
2.14	Le <i>Padmouse</i> . . . . .	42
2.15	Le <i>Rockin'Mouse</i> . . . . .	44
2.16	Les variétés de souris rotatives classées selon le type de technique utilisée pour la captation de l'angle de lacet . . . . .	45
2.17	Le dispositif <i>Scrubwheel</i> . . . . .	46
2.18	La « souris à double-boule » . . . . .	47
2.19	La coque d'une souris rotative . . . . .	48
2.20	Le <i>TwistMouse</i> . . . . .	50

---

2.21	Le dispositif <i>Toolstone</i> . . . . .	51
2.22	Le <i>VideoMouse</i> . . . . .	52
2.23	La souris à trois DDL <i>Cylo</i> . . . . .	54
3.1	Prototype développé d'une souris à trois DDL à partir de deux souris sans fil . . . . .	62
3.2	Evolution des valeurs $X_1$ , $Y_1$ , $X_2$ et $Y_2$ pour une rotation libre de $3 \times 360^\circ$ . . . . .	63
3.3	Les quatre techniques d'interaction utilisées dans l'étude comparative . . . . .	66
3.4	Copies d'écran du logiciel d'évaluation de la tâche de manipulation trois DDL d'une fenêtre . . . . .	68
3.5	Le <i>setup</i> expérimental et la flèche indiquant l'orientation du capteur de la souris rotative ronde . . . . .	70
3.6	Variation de la durée moyenne de chaque essai au fil des 20 essais . . . . .	71
3.7	Les durées moyennes des essais et l'écart type pour chaque technique d'interaction . . . . .	72
3.8	Histogrammes des durées des essais pour les techniques 3DM et 3DR . . . . .	73
3.9	Histogrammes des durées des essais pour les techniques 2DW et 2DM . . . . .	74
3.10	Fréquence des notes données par les participants à chaque technique d'interaction . . . . .	75
3.11	L'avis de participants sur la technique qu'ils ont considérée « la moins fatigante » et l'avis des participants sur la technique qui a procuré « la plus grande maîtrise » sur l'objet virtuel. . . . .	76
4.1	Cadre de travail des ingénieurs du don . . . . .	82
4.2	Exemple de manipulation précise basée sur les doigts . . . . .	83
4.3	Un exemple d'interface physique et d'interface graphique pour le contrôle des paramètres sonores . . . . .	86
4.4	Le dispositif <i>Wacom 4D Mouse</i> dans une coque ronde de polystyrène . . . . .	89
4.5	Le modèle à trois états utilisé par la technique pour faciliter l'activation spatiale . . . . .	91
4.6	Interface de visualisation pour le calibrage de la technique . . . . .	92
4.7	<i>Widgets</i> rotatifs associés à des paramètres couleur d'un objet de la fenêtre . . . . .	93
5.1	Palettes ouvertes dans <i>Photoshop</i> . . . . .	100



5.2	Les trois DDL captés par le dispositif et la correspondance avec les mouvements de la « palette satellitaire » sur l'écran . . . . .	105
5.3	Schémas présentant le fonctionnement de la palette satellitaire . . . . .	106
5.4	Le prototype de la souris rotative fait à partir du <i>4DMouse</i> . . . . .	107
5.5	La palette satellitaire en action . . . . .	108
5.6	La palette satellitaire avec quatre, huit, seize et trente-deux zones. . . . .	110
5.7	La souris ronde d' <i>Apple</i> . . . . .	111
5.8	Exemple de sélection d'un outil par la technique d'« activation gestuelle » . . . . .	113
5.9	Configuration des zones pour la palette satellitaire basée sur une sélection angulaire . . . . .	114
6.1	Un « nuage de mots-clés » et un « carrousel 3D » de couverture de livres . . . . .	120
6.2	Copies d'écran du site <i>Google Books</i> et du logiciel de gestion personnelle de livres <i>Delicious</i> . . . . .	121
6.3	Deux livres de taille différents côte à côte . . . . .	122
6.4	L'image finale formée à partir des quatre images intermédiaires . . . . .	127
6.5	Les quatre « canaux » utilisés pour créer l'image grand angle et le rendu final . . . . .	128
6.6	Copies d'écran du catalogue immersif avec l'affichage traditionnel et avec le rendu à deux étapes qui crée un champ de vision de 180° . . . . .	129
6.7	Un utilisateur navigue dans le catalogue immersif projeté sur la <i>VisionStation</i> . . . . .	131
6.8	Solution focus plus contexte utilisée pour augmenter la résolution de la <i>VisionStation</i> . . . . .	132
6.9	La configuration bi-écran et détail sur la qualité visuelle des mêmes éléments affichés sur la <i>VisionStation</i> et sur l'écran auxiliaire . . . . .	133
7.1	Les 32 500 pages de <i>La Nature</i> affichées sur une même image . . . . .	142
7.2	Les 32.500 pages de <i>La Nature</i> affichées sur une même image ; un volume par ligne . . . . .	143
7.3	Exemple d'une image master issue d'un volume numérisé par un prestataire du <i>CNUM</i> . . . . .	144
7.4	Formats des images et étapes du traitement dans le <i>CNUM</i> . . . . .	145
7.5	Copies d'écran de l'interface de contrôle qualité « Grille de détails » : miniatures et détails . . . . .	146

7.6	Copies d'écran de l'interface de contrôle qualité « Grille de détails » : images master et pages de différents volumes . . . . .	147
7.7	Une métaphore décrivant les trois DDL de la caméra du <i>Mur de pages</i>	149
7.8	Le <i>Mur de pages</i> affiché dans deux configurations d'affichage différentes . . . . .	150
7.9	Schéma avec les caméras virtuelles de la maquette « Panorama de détails » . . . . .	152
7.10	Maquette interactive du « Panorama de détails » fait sur <i>Processing</i>	153
8.1	La centrale inertielle à six DDL <i>Sparkfun v4</i> . . . . .	160

# Introduction

La problématique de cette thèse s'inscrit dans le cadre de l'*interaction bureautique* ; il s'agit du contexte d'usage où la durée et l'attention consacrées à la réalisation des tâches sont importantes. En effet, ce contexte est marqué par des *tâches cognitives intensives* (O'Hara et Sellen, 1997). Voici d'autres caractéristiques de cette situation d'usage :

- l'utilisateur travaille assis ;
- les périphériques d'entrée sont génériques et se trouvent sur un bureau ;
- plusieurs applications sont conçues pour répondre à des besoins professionnels précis (édition vidéo, création graphique, mixage sonore, mise en page de textes, outils pour la gestion des comptes, etc.) ;
- les tâches imposent des opérations très répétitives (surtout au plan gestuel) ;
- le temps nécessaire à la réalisation d'une tâche est une contrainte importante.

La figure 1.1 présente quelques repères dans un continuum de situations d'usage de l'informatique. Cet espace avance vers une complexification croissante des tâches, ce qui exige également une concentration de plus en plus grande de la part de l'utilisateur.

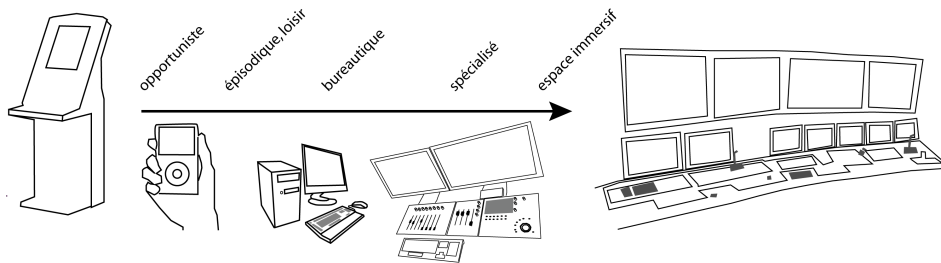


FIG. 1.1: Continuum des contextes d'usage de l'informatique.

Dans les dernières années, les améliorations techniques ont beaucoup privilégié l'interaction mobile. Les téléphones portables et les PDA convergent vers le format d'ordinateur de poche disposant d'un système opérationnel assez complet et d'un grand nombre d'applications. Ces dispositifs sont dotés d'écrans tactiles (souvent capables de détecter simultanément plus d'un point de contact) et, dans les grandes villes, sont connectés de manière presque permanente à l'internet. Malgré l'introduction de nombreuses techniques d'interaction visant à faciliter la saisie de texte, la sélection de commandes et l'optimisation graphique de petites écrans, la taille réduite de l'espace d'entrée et de sortie de ces dispositifs ne les rend pas confortables pour une utilisation longue et intensive. Dans l'idée du *Calm Computing* (Weiser, 1991), ces dispositifs sont utiles pour permettre à l'utilisateur de rester, même en déplacement, intégré à la trame de l'univers informationnel numérique, c'est-à-dire d'accéder à sa messagerie, de recevoir des informations de contexte associées à l'endroit où il se trouve (grâce au GPS), de pouvoir consulter sporadiquement certains documents, etc.

L'interaction bureautique tend ainsi à rester le cadre *privilégié* pour travailler (que ce soit pour la création ou pour la consultation) avec ses documents numériques (du moins pour les documents *visuels*). En effet, on est habitué à avoir son poste de travail placé dans un endroit précis et organisé d'une façon donnée. On peut configurer cet espace pour se servir avec plus de confort et d'efficacité des périphériques d'entrée. En outre, ce lieu physique est utile pour que l'on se mette en relation avec ses collègues, pour que l'on dispose ses documents papier sur les surfaces autour, etc.

## 1.1 Limitations des interfaces actuelles

L'interaction bureautique est aujourd'hui synonyme d'interfaces graphiques. La souris et le clavier sont les dispositifs d'entrée les plus communs dans cette configuration. La « manipulation directe » décrit le style d'interaction (apparu à la fin des années 80) où la boucle entrée-sortie est plus courte et essentiellement graphique (Shneiderman, 1987). Dans les interfaces actuelles, les résultats des actions de l'utilisateur sont immédiatement restitués de façon continue, les commandes de l'application sont visibles et cliquables (la syntaxe est donc simplifiée), les documents peuvent être édités tel qu'il se présenteront à la fin<sup>1</sup>, etc. Le *WIMP* (acronyme pour *Windows, Icons, Menus, and Pointer*) désigne les éléments centraux des interfaces graphiques. Les documents entrent dans l'espace de travail et sont manipulés via des intermédiaires graphiques comme la fenêtre, l'icône, le menu et le curseur. Les icônes sont des représentations graphiques des documents ou des applications. Les documents sont « ouverts » dans des fenêtres et celles-ci - pouvant être réduites, chevauchées et placées côte à côte - constituent une mé-

---

<sup>1</sup>Les applications qui présentent les documents de telle façon sont identifiées comme *WYSIWYG*. Acronyme pour « ce que vous voyez est ce que vous obtenez ».

taphore intuitive pour qu'on travaille avec plus d'un document (ou plus d'une application) en même temps.

Les interfaces graphiques orientées par ces principes ont rendu possible une utilisation beaucoup plus accessible à des utilisateurs novices. Cependant, face au nombre grandissant de données numériques et face à la diversité de tâches que l'on doit actuellement réaliser avec l'ordinateur, les interfaces du type WIMP semblent de moins en moins adaptées aux conditions d'interaction actuelles. Dourish (2001), par exemple, dresse une critique assez complète sur les limitations des paradigmes d'interaction dominants et suggère que de nouvelles approches pourraient favoriser une interaction moins en série, moins dépendante de la modalité visuelle, moins gourmande de concentration et plus facile à partager avec d'autres utilisateurs qui se trouvent sur le même environnement.

Comme remarquent Gaver (1989), concernant la modalité sonore, ou Snibbe et al. (2001), concernant la modalité haptique, les interfaces actuelles s'appuient presque exclusivement sur la modalité visuelle. Puisque la vision est un canal d'acquisition d'informations privilégié chez l'humain, l'aspect graphique de l'interface tend à garder sa place principale. Toutefois, des interfaces qui tirent profit des autres modalités peuvent faire appel à l'attention périphérique de l'utilisateur, c'est-à-dire qu'elles peuvent fournir des informations auxiliaires sur le contexte de la tâche et réserver à l'attention visuelle uniquement les opérations centrales.

Par ailleurs, Sellen et al. (2002) observent que les interfaces bureautiques sont faites pour que l'utilisateur se serve d'une seule main alors que l'utilisation de deux mains correspond très souvent à la façon la plus naturelle de travailler (Guiard, 1987). Enfin, O'Hara et Sellen (1997) observent que la lecture (ainsi que la prise de notes) de documents numériques est plus gourmande d'attention et produit des résultats moins efficaces que la même activité réalisée avec des documents en papier.

**Gestion de plusieurs documents numériques.** En effet, la gestion de plusieurs documents dans les interfaces WIMP est une problématique ancienne. Les fenêtres des documents chevauchent lorsqu'elles sont nombreuses. Passer d'un document, ou d'une application, à un autre n'est pas une opération simple. Elle ne peut pas être réalisée sans que l'utilisateur focalise son attention sur cette opération. Plusieurs études ont été réalisées dans le but d'offrir un moyen facile de transiter entre plusieurs documents dans l'espace graphique réduit des écrans.

A titre d'exemple, les travaux de Perlin et Fox (1993) et Bederson et Hollan (1994) - introduisant les « interfaces zoomables » (ZUI<sup>2</sup>) - peuvent être mentionnés. Ces interfaces permettent qu'on visualise un espace de travail beaucoup plus grand que celui offert par une interface graphique traditionnelle. Tous les documents peuvent être réduits pour qu'ils apparaissent ensemble dans l'écran et, inversement, on peut zoomer sur les zones de cet espace sur lesquelles on souhaite

---

<sup>2</sup>Acronyme pour « *Zoomable User Interface* ».

travailler. Par ailleurs, Topol (2000) propose une interface où les fenêtres sont « immergées » dans un espace 3D pour que, grâce au zoom intégré et au changement de perspective, on puisse facilement trouver des configurations graphiques où plusieurs fenêtres apparaissent en même temps. D'autre part, Beaudouin-Lafon (2001) explore des techniques d'interaction, qui s'inspirent de la façon dont on travaille lorsqu'on dispose de plusieurs documents en papier, pour palier la difficulté de passer d'une fenêtre à l'autre. En effet, plusieurs techniques dites *post-WIMP*, comme le *BumpTop* (Agarawala et Balakrishnan, 2006), proposent de mélanger des éléments des interfaces zoomables ou 3D dans des métaphores d'interaction plus ambitieuses afin d'offrir plus de souplesse dans la manipulation des documents numériques.

**Espace visuel appauvri.** La configuration de la partie « matérielle » des interfaces graphiques est, elle aussi, remise en cause puisqu'elle permet une utilisation très limitée de l'espace visuel de l'utilisateur. D'autant plus que celui-ci peut passer un nombre important d'heures par jour en travaillant à l'ordinateur sous ces conditions appauvries. La différence entre cette configuration informatique et l'activité analogue réelle est flagrante puisque lorsqu'on travaille avec des documents en papier, tout l'espace qui nous entoure peut servir pour structurer l'information.<sup>3</sup> Sur les murs, on colle des rappels ou des diagrammes ; sur les meubles voisins, on laisse les dossiers qu'on vient d'utiliser, dont on compte parler à un collègue, etc. ; sur le bureau, la façon dont on étale les documents, comme on crée une hiérarchie avec des piles et prend en compte la topologie du bureau aide à ce que l'on décharge sa mémoire. Il s'agit donc d'un espace dense en information et qui peut être configuré de manière fine (Malone, 1982).

Cet espace visuel insuffisant a une conséquence sur l'augmentation du nombre d'opérations que l'utilisateur devra réaliser avec le périphérique d'entrée. De fait, plus l'écran est réduit, plus d'opérations avec la souris (ou le clavier) l'utilisateur devra réaliser pour afficher le document qui ne l'est pas (puisque'il est « minimisé » ou couvert par une autre fenêtre). Ainsi, la tendance naturelle serait d'augmenter l'espace d'affichage de l'information numérique. Les approches *complètement immersives* (casques de réalité virtuelle ou *cave*) sont peu appropriées à une situation de travail quotidien car ils coupent trop l'utilisateur du contact avec le monde réel (documents papier, collègues de travail, etc.) et sont visuellement fatigants pour des durées d'utilisation importantes. Il existe cependant nombreux projets envisageant une *augmentation visuelle* de l'espace de travail d'une façon moins envahissante.

Dans le « bureau augmenté » proposé par Wellner (1991), plusieurs surfaces de la pièce où se trouve l'utilisateur peuvent servir comme surface de projection

---

<sup>3</sup>Sur le plan visuel, la configuration écran-souris traditionnelle dont on se sert actuellement équivaldrait au travail dans un bureau avec des dimensions plus réduites. C'est-à-dire, un bureau doté d'une surface qui offre peu d'espace pour que l'on place ses documents secondaires autour du document principal.

et afficher une couche d'information numérique. Cela fait que l'espace physique et l'information numérique *partagent le même espace visuel* et, voire même interagissent entre eux. D'autres études, comme celles de Grudin (2001) ou Baudisch et Good (2002), évoquent les bénéfices de l'élargissement du périphérique d'affichage. Cependant, Baudisch et Good constatent qu'un problème récurrent des grandes surfaces de projection est la faible résolution visuelle, ce qui peut les rendre peu appropriées à une tâche pour laquelle les détails visuels des documents sont importants. Concernant les surfaces d'affichage composées par plusieurs écrans, un des inconvénients est créé par les bordures entre les écrans qui introduisent des ruptures dans l'espace visuel (Mackinlay et al., 2003).

## 1.2 Espace moteur et espace visuel

L'étude de l'interaction bureautique proposée ici est divisée en deux axes principaux. D'une part, il sera étudié les apports d'une souris dotée de la captation de la rotation afin de rendre certaines tâches plus fluides. D'autre part, les goulots d'étranglement de la visualisation massive d'images sont analysés et deux solutions pilotes pour les tâches de visualisation de grandes quantités d'images numérisées sont proposées.

Ces deux types de tâche (la manipulation de documents et l'exploration visuelle) correspondent de façon analogue à des actions que l'on réalise dans des activités réelles. Dans la première partie, cette thèse examine des techniques permettant d'aboutir aux résultats que l'on obtient habituellement lorsqu'on agit sur des objets avec sa main. Dans la deuxième partie, les techniques permettant d'aboutir aux effets que l'on obtient par la « vision déambulatoire » (Gibson, 1979) sont étudiées. Il s'agit de la démarche de compréhension visuelle de l'environnement dans laquelle le sujet se sert des mouvements des yeux, du cou, du corps en déplacement. On peut également utiliser ses mains pour aider la visualisation ou la compréhension visuelle d'un objet.<sup>4</sup>

### 1.2.1 Optimiser l'espace moteur de la souris

L'*espace moteur* de l'interaction est l'espace gestuel dans lequel l'utilisateur manipule le périphérique d'entrée. Dans le cas de la souris traditionnelle, il s'agit de la région de la table à l'intérieur de laquelle on la déplace. Mais outre ses dimensions, cet espace peut être également caractérisé par la granularité des mouvements qu'il offre (s'ils sont plutôt fins ou grossiers) ; il peut permettre des mouvements plus ou moins expressifs, selon les types de données captées par la souris.

---

<sup>4</sup>Par exemple, on a tendance à tourner, à éloigner et à suivre avec le doigt un schéma difficile à comprendre ou une carte dense en informations. Kirsh et Maglio (1994) identifient ce type de geste comme étant des actions *épistémiques*.

La souris est un périphérique très performant ; Balakrishnan et al. (1997) et Moggridge (2006) ont identifié quelques-uns de ses points forts :

- on repose sa main sur la souris gardant la prise appropriée pour la déplacer ou pour cliquer ;
- manipulation stable ;
- dispositif non-volatile : lorsqu'on la lâche, elle garde la même position physique et le même état logique ;
- mécanisme de « débrayage » naturel (voir *clutching* ci-dessous) ;
- on peut manipuler ses deux axes simultanément ;
- c'est un périphérique relatif : on déplace le curseur sur un espace graphique qui peut être plus grand ou plus petit que la surface physique sur laquelle la souris se déplace.

En plus des atouts listés ci-dessus, la souris est aujourd'hui le produit résultant des efforts concentrés de plus de quatre décennies de *design*, ergonomie et électronique (Buxton, 2008).

### Limitations de la souris traditionnelle

Malgré tous les points positifs cités, ce périphérique comporte un certain nombre d'inconvénients inhérents. Quels sont-ils ? Dans quelles tâches ces obstacles interviennent-ils ? Reprenons quelques points évoqués plus haut.

**Clutching.** Ce mécanisme de débrayage est en réalité l'effet de bord d'un déplacement important du curseur. Lorsqu'on déplace la souris, et ce mouvement seul ne suffit pas pour que le curseur atteigne la cible, la souris doit être remise à une position neutre pour que l'on puisse donner suite au mouvement du curseur. Cette limitation du mouvement peut être due soit à une manque d'espace sur le bureau, soit à des contraintes dynamiques des membres impliqués. Le *clutching* devient une opération fatigante si on doit l'exécuter avec fréquence.

**Empreinte sur le bureau.** L'espace que l'on doit réserver sur le bureau pour le déplacement de la souris est plus grand que celui d'autres périphériques équivalents, comme le *trackball* ou le *trackpad*. Les dimensions de cette « empreinte » sur le bureau ont un impact sur le *clutching* puisque moins il y a de surface disponible où l'on peut faire rouler la souris, plus souvent on devra la soulever et la re-positionner.

**Captation limitée à deux degrés de liberté.** La souris est un périphérique doté d'une captation limitée à deux degrés de liberté (DDL<sup>5</sup>) - on ne peut contrôler de façon continue et simultanée que deux paramètres. De plus, ce dispositif capte

---

<sup>5</sup>Dans l'espace tridimensionnel, un objet solide a six degrés de liberté (DDL), c'est-à-dire que les transformations géométriques qu'il peut subir peuvent être décomposées dans des opérations de translation (déplacement le long des axes X, Y ou Z) et de rotation (variation angulaire autour de ces mêmes axes). Lorsqu'un dispositif peut capter ces six variations, on dit qu'il s'agit d'un dispositif à six DDL. La souris est un périphérique à deux DDL puisqu'elle ne peut capter que les translations limitées au plan de la surface sur laquelle on la fait rouler.



uniquement les translations - si l'on souhaite faire tourner un objet sur l'écran, il n'y a pas de correspondance naturelle entre cette transformation virtuelle et les mouvements que l'on doit réaliser.

**Faible résolution temporelle des muscles impliqués.** La manipulation de la souris est assurée par tout le système moteur des membres supérieurs dominants (comprenant le poignet, le coude et l'épaule). Les muscles de ces articulations sont capables d'exécuter des mouvements avec une fréquence temporelle et une finesse spatiale moins importantes que le sont ceux des doigts (Zhai et al., 1997). Des manipulations qui font appel à ces muscles fins peuvent être plus économes et permettre des mouvements plus diversifiés (Elliott et Connolly, 1984).

**Inconvénients ergonomiques.** Une étude ergonomique réalisée par la société *Wacom* (GEC, 1998) révèle que les excès moteurs qui nuisent aux articulations des doigts, du poignet et de l'avant-bras sont dus au type de tâche, à la forme du dispositif, à sa hauteur, à la position relative du dispositif et à la position des boutons. Un type d'opération exigeant souvent des mouvements ulnaires trop excessifs est celle du tracé d'une ligne horizontale (en mode « *drag* »). Un périphérique pouvant être saisi avec la paume de la main perpendiculaire au plan de travail, comme le stylet, permet une prise plus neutre pour les articulations que celle permise par la souris.

**Manipulation du curseur dépendant d'un suivi visuel continu.** La souris est un périphérique d'entrée relatif. De ce fait, lorsqu'on la déplace, son mouvement doit être guidé non pas par sa position sur le bureau mais par la position du curseur graphique sur l'écran (Fitzmaurice et Buxton, 1997). Ainsi, durant les déplacements du curseur, l'utilisateur ne pourra pas faire rouler la souris tout en portant son attention visuelle sur un autre élément de l'interface. En effet, le retour kinesthésique ressenti par l'utilisateur durant ce type de manipulation ne peut presque rien lui apporter sur l'action logique qu'il exécute. De ce fait, même si l'utilisateur réalise souvent et depuis longtemps la sélection d'une commande de menu, il ne pourra jamais se passer du retour visuel afin d'accomplir cette sélection (Buxton, 2007).

**Tâches dont l'exécution est pénible.** Certaines études se sont penchées sur l'insuffisance de la souris dans la réalisation de certaines tâches. Des techniques, se basant sur la souris, ont été proposées en vue de contourner ces problèmes. Les contextes problématiques les plus analysés sont les suivants :

- utilisation fréquente d'outils et les allers-retours en vue de les acquérir. Exemple : entre les documents d'un côté et les menus et les palettes de l'autre (Bier et al., 1993) ;
- contrôle de plus de deux variables d'une tâche (Hinckley et al., 1997) ;
- sélection d'un item dans un ensemble de taille très importante (Masui et al., 1995; Appert et Fekete, 2006) ;
- sélection de cibles très petites. Des approches « multi-échelle » avec la séparation entre l'espace graphique et l'espace moteur ont été explorées (voir,

par exemple, Ayatsuka et al., 1998; Bourgeois et Guiard, 2002; Elmqvist et Fekete, 2008).

### Capter la rotation de la souris

Les inconvénients cités en haut peuvent être attribués au faible débit de communication entre la souris traditionnelle et les applications actuelles. Je propose d'augmenter la captation de la souris par la prise en compte de la rotation autour de l'axe vertical (par rapport à la surface de déplacement), comme un moyen d'élargir ce canal de communication.

Les apports *immédiats* de cette augmentation de la manipulation basée sur la souris traditionnelle sont les suivants :

- mise en parallèle de certaines actions composées requérant plus de deux paramètres ;
- contrôle plus intuitif d'un paramètre graphique angulaire (par exemple, la rotation d'un dessin ou d'un *widget*) ;
- manipulations *absolues*<sup>6</sup> ou *quasi-absolues* grâce au fait que la rotation de la souris peut être absolue même si son déplacement reste relatif ;
- réduction du nombre de *clutchings* grâce au fait que les opérations de rotation n'engendrent pas de déplacement ;
- une plus grande variabilité des gestes et diversité des membres impliqués (tous les cinq doigts de la main peuvent intervenir) dans la manipulation.

Cette thèse s'est également appuyée sur des travaux issus de l'ergonomie et de la psychologie afin de comprendre d'autres conséquences ainsi que les limitations de ce type de manipulation. Des exemples de ces aspects à prendre en compte sont :

- les amplitudes maximale et minimale des gestes de rotation (Pheasant et Haslegrave, 2006) ;
- la coordination entre les mouvements de rotation et de déplacement (Jacob et Sibert, 1992; Wang et al., 1997) ;
- les « stratégies motrices » mises en oeuvre par les doigts et le poignet lors de ce type de mouvement (Elliott et Connolly, 1984; Mackenzie et Iberall, 1994).

### Construction d'un prototype de souris rotative

Comme nous le verrons dans la section suivante (1.3), la motivation de départ de la thèse pour étudier l'interaction basée sur une souris rotative était dans l'utilisation de ce dispositif pour la manipulation de documents dans une scène 3D. La souris rotative permet un contrôle naturel de cette tâche. C'était suite à

---

<sup>6</sup>Le terme « absolu » fait référence à la possibilité d'exécuter une opération sans avoir recours exhaustif au feed-back graphique mais en se servant d'informations haptiques et kinesthésiques.

l'état de l'art du domaine qu'il a été constaté qu'un dispositif de ce type pourrait faciliter les opérations d'autres tâches métier traditionnelles. J'ai cherché donc à comprendre quelles seraient les caractéristiques de l'« espace de conception » pour des techniques d'interaction basées sur ce type de périphérique d'entrée.

Le centre d'intérêt de la thèse était ainsi l'*utilisation* qu'on peut faire de ce dispositif et non pas la *conception* d'un nouveau dispositif. Cependant, étant donnée l'inexistence d'un dispositif avec les caractéristiques nécessaires, j'ai dû construire des prototypes.<sup>7</sup> Il a été également nécessaire d'étudier comment certains aspects physiques du prototype (par exemple la symétrie, l'emplacement du bouton, les dimensions, etc.) ont un impact sur la manipulation et comment ils gêneraient ou contribueraient aux techniques proposées.

Au cours de la thèse, deux prototypes de souris rotatives ont été utilisés. Le premier était réalisé à partir du couplage mécanique de deux mini-souris optiques sans fil. L'orientation relative de ce nouveau dispositif assemblé était calculée à partir des différences entre les données envoyées par la souris droite et par la souris gauche. Le deuxième prototype était une adaptation de la souris *4D Mouse* de la société *Wacom*. Ce prototype s'utilise sur une tablette magnétique *Wacom* qui fournit son orientation absolue avec une précision importante. Cependant, puisque ces données comportent beaucoup de bruit, ils subissent un filtrage. Cela introduit un délai important dans l'envoi de ces données.<sup>8</sup> Enfin, vers la fin de la thèse, l'utilisation d'une centrale inertielle couplée à un magnétomètre a commencé à être étudiée (voir les « perspectives » dans le chapitre 8.2). La faisabilité d'un prototype basé sur ces technologies demeure toutefois incertaine.

## 1.2.2 Elargir l'espace visuel des bibliothèques numériques

Les corpus d'informations numériques auxquels on peut accéder via le Web s'agrandissent à toute vitesse.<sup>9</sup> La gestion de cette importante masse de données graphiques est une problématique liée aussi aux documents personnels de chaque utilisateur : ses collections de photos, ses sites visités, ses morceaux vidéo enregistrés, etc. Les bibliothèques numériques sont des entrepôts de stockage et d'accès à ces documents. L'« existence » numérique de ces documents présente toutefois un inconvénient important. Il ne suffit pas que ce document soit interrogeable pour qu'il existe aux yeux de l'utilisateur. Si l'on regarde la façon dont sont organisés les sites des bibliothèques numériques, on peut aisément constater que la plupart des livres n'est pas immédiatement « visible ». Comment peut-on trouver les documents que l'on cherche de manière imprécise et comment peut-on *découvrir*

---

<sup>7</sup>J'ai été en relation avec M. Lawrence Crew, concepteur de la souris *Cylo* (voir chapitre 2.3.3), pour voir la possibilité d'utiliser un de ses prototypes de souris rotative pour réaliser les évaluations des techniques conçues. Cela n'a malheureusement pas été possible.

<sup>8</sup>J'ai également été en relation avec un ingénieur de *Wacom* dans l'espoir de trouver un moyen de contourner ce délai bien que je n'aie pas eu de succès.

<sup>9</sup>*Google* comptait vingt-huit millions de pages Web en 1998, un milliard en 2000 et un trillion en juillet 2008. <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>

des documents intéressants qui ne sont pas connus d'avance ? Des représentations plus graphiques de ces documents offrent une navigation plus intuitive et plus efficace dans ces corpus (Rao et al., 1995). De même, il serait intéressant d'explorer une vision élargie de cet espace de documents pour que l'utilisateur puisse se servir de tout son champ de vision, ce qui pourrait faciliter la compréhension de cet espace. Comment les techniques de visualisation immersive pourraient être appliquées à la visualisation d'un catalogue de documents numériques ?

D'autre part, cette foule de données graphiques pose des problèmes au « bibliothécaire numérique » et non pas seulement au « lecteur numérique ». Dans le cas des bibliothèques numériques qui produisent des fac-similés des volumes numérisés, un contrôle qualité des fac-similés doit être réalisé pour assurer leur validité : des pages peuvent être manquantes, répétées, mal photographiées, les fichiers peuvent avoir été échangés par erreur, etc. Le nombre de pages régulièrement numérisées étant très important, il est nécessaire d'explorer des méthodes pour que tout l'ensemble puisse être inspecté de façon optimale. Les outils d'archivage et de visualisation de photos numériques ne sont pas faits pour prendre en compte les particularités de ces documents. Par exemple, les pages d'un même volume ou d'une même série possèdent des éléments graphiques communs (bordure, taille du texte, type d'illustration, etc.). Cette caractéristique n'est pas vraie pour une collection de photos personnelles. Comment fournir des outils qui facilitent la visualisation exhaustive de grands lots de fac-similés ? Les problématiques concernant la visualisation du document dans ces deux contextes sont nombreuses et elles seront à nouveau évoquées dans la section suivante (1.3).

### 1.3 Contexte de la thèse

Cette thèse a pour but l'approfondissement de la compréhension de certaines problématiques de l'interaction bureautique. Toutefois, ses motivations de départ se situent dans le domaine plus spécifique des bibliothèques numériques. Elle a été réalisée au sein de l'équipe « Interaction pour lire et jouer » (au laboratoire Cedric du CNAM), qui a mené plusieurs études antérieures concernant l'interaction avec le document numérique. Quelques-uns de ces travaux sont évoqués ici puisqu'ils mettent en contexte les questions explorées dans cette thèse.

Concernant le processus de numérisation et la manière de représenter le document numérique, Cubaud (2006) estime que les méthodes traditionnelles (extraction du texte de chaque page ou à partir des fac-similés des pages) « sous-échantillonnent » le document. Le document est considéré comme étant une suite de caractères ou de pages et on ignore le fait qu'il est aussi un objet (un objet doté d'un volume, d'une taille absolue, de parties qui interagissent entre elles). Cubaud et al. (2005) se sont penchés sur l'insuffisance de la numérisation traditionnelle dans le cas spécifique des livres à systèmes<sup>10</sup> et ont proposé un modèle

---

<sup>10</sup>Il s'agit de livres scientifiques ou ludiques où des parties mobiles articulées peuvent former un

pour décrire le fonctionnement des parties articulées de ces documents. D'autre part, la numérisation par fac-similé (la mise à plat du livre) atteint aussi ses limites lorsque les pages du volume ont été elles-mêmes transformées en volumes, c'est-à-dire abîmées, à cause de l'action du temps ou de l'humidité. Une numérisation des courbures des pages permet de mieux restituer l'information imprimée, voire de corriger la déformation sur la représentation numérique (Cubaud et al., 2004).

Des travaux de l'équipe ont contribué à l'architecture et à l'implémentation d'environnements 3D interactifs. Cubaud et Topol (2001) ont développé en VRML une interface 3D de bibliothèque numérique et ont proposé des évolutions de ce langage de description pour qu'il prenne en compte de comportements plus interactifs (Cubaud et al., 2000). Dupire et al. (2005) ont étudié la faisabilité de se servir de moteurs de jeu vidéo commerciaux pour faciliter la conception d'une bibliothèque numérique 3D disposant ainsi de fonctionnalités que l'on retrouve dans les jeux vidéo (par exemple, rendu graphique performant, moteur physique pour la détection de collisions, etc).

En ce qui concerne la consultation des documents, Cubaud et Topol ont proposé un *widget* de lecture de documents dans un environnement 3D. Il s'agit d'un trépied avec deux pages qui peut être déplacé sur un plan et qui dispose des boutons pour donner accès à d'autres fonctionnalités, comme le feuilletage, la rotation du document et le zoom de la page (voir une description du fonctionnement dans Almeida et al., 2006). Plusieurs trépieds peuvent se trouver dans cet environnement et, en ajustant leur profondeur et leur orientation, l'utilisateur peut en visualiser plusieurs en même temps. Bien que le déplacement du trépied sur le plan soit associé directement au mouvement (X,Y) de la souris traditionnelle, sa rotation ne peut pas être directement contrôlée par la souris. La prise en compte naturelle de ces trois variables (déplacement sur le plan et rotation autour de l'axe vertical) par le périphérique d'entrée a été ma motivation de départ pour l'exploration et l'évaluation des techniques basées sur une souris rotative.

D'autres travaux sont consacrés à la conception d'interfaces permettant la visualisation d'une collection de documents numériques. Dans Cubaud et al. (1998) puis dans Cubaud et al. (2002), un espace cylindrique pour naviguer dans une collection est proposé ; chaque document est représenté par l'image du dos de la reliure du volume original. Topol (2002) a exploré différentes géométries de scène afin que la position du document renseigne l'utilisateur sur la pertinence de celui-ci par rapport à une requête de recherche lancée. Mon investigation sur la faisabilité technique de l'utilisation d'un dispositif immersif pour les bibliothèques numériques s'est basée sur les interfaces conçues dans ces études.

Par ailleurs, les membres de l'équipe ont participé activement dans la création de deux bibliothèques numériques : l'Association des bibliophiles univer-

---

volume (afin de représenter un bâtiment, par exemple) ou peuvent être glissées pour cacher/dévoiler une partie d'une figure, comme dans le cas de certains livres d'anatomie.

sels (ABU)<sup>11</sup> (Cubaud et Girard, 1998) et la bibliothèque numérique du CNAM (CNUM)<sup>12</sup> (Cubaud et Deblock, 2003). Dans le CNUM, dont le fonds est composé principalement par des ouvrages de sciences et techniques du XVIII, XIX et XX siècles, l'utilisateur consulte les fac-similés des pages des volumes originaux. J'ai travaillé pendant ma thèse pour le CNUM et cette expérience m'a permis de connaître en détail les obstacles associés au contrôle qualité de grands lots d'image (plus de 50.000 pages mises en ligne par an), à la validité des métadonnées et à la cohérence entre les informations qui se trouvent (a) sur les documents originaux, (b) sur les fac-similés issues de la numérisation et (c) sur les métadonnées. J'estime que cette connaissance en interne des bibliothèques numériques m'a également permis d'apercevoir les défis qu'auront prochainement les utilisateurs ordinaires dus à l'accumulation d'une quantité grandissante de documents personnels. L'étude sur les interfaces pour la visualisation d'images dans un contexte de contrôle qualité de lots de pages numérisées s'appuie largement sur cette expérience.

## 1.4 Organisation de la thèse

Les chapitres 2 à 5 traitent de la question de la manipulation de la souris rotative, tandis que les chapitres 6 et 7 abordent la problématique de la visualisation pour les bibliothèques numériques.

Le chapitre 2 couvre un large spectre de domaines qui interviennent dans l'action de manipuler une souris rotative. Tout d'abord, des concepts de la psychologie motrice et perceptive, ainsi que de l'ergonomie sont présentés. Ils sont utiles pour la compréhension de l'action de tourner un objet avec la main. Ensuite, certains dispositifs exclusivement rotatifs, comme les potentiomètres rotatifs, seront évoqués. Les propriétés physiques de ces contrôles, l'usage qu'on en fait et les contraintes qu'ils imposent sont décrits. Puis, une liste de propriétés, issues de taxonomies des périphériques d'entrée, est présentée ; elles permettent de caractériser certains aspects importants des souris rotatives. Quelques « souris enrichies », l'ensemble dans lequel se trouvent les souris rotatives, sont décrites et un état de l'art des souris rotatives est réalisé. Enfin, nous visiterons quelques interfaces graphiques qui utilisent la rotation.

Le chapitre 3 présente une étude dans laquelle une souris rotative sert à contrôler la position et l'orientation d'une fenêtre dans une scène 3D. D'autres travaux qui ont proposé de contrôler le positionnement et l'orientation d'un document dans un espace sont évoqués. Une évaluation contrôlée d'une tâche de « *docking* » d'une fenêtre a été réalisée comparant quatre conditions de manipulation : (1 et 2) deux configurations de coques pour cette souris trois DDL, (3) la manipulation par molette et (4) la manipulation par le clic bouton suivi du déplacement vertical.

---

<sup>11</sup><http://abu.cnam.fr>

<sup>12</sup><http://cnum.cnam.fr>

La méthodologie de cette évaluation se base sur l'étude réalisée par Hinckley et al. (1997).

Dans le chapitre 4, la manipulation de potentiomètres graphiques dans les applications sonores est examinée. Ce chapitre décrit comment ces opérations de réglage de *widgets* rotatifs, effectuées des dizaines de fois par minute, sont peu économes au niveau gestuel et peu intuitives par rapport au feed-back qu'elles offrent. Une solution pour rendre ce réglage plus fluide est proposée : la souris contrôle le *widget* directement par sa rotation lorsque le curseur se trouve sur la zone de ce dernier. Les apports de cette mise en correspondance naturelle et les problèmes rencontrés dans un premier prototype sont évoqués ainsi que les solutions proposées pour les surmonter. Ce chapitre se termine en présentant les perspectives d'une deuxième technique de sélection par la rotation qui permettrait la sélection d'une valeur dans un ensemble de taille importante.

Le chapitre 5 présente une contribution de cette thèse pour la problématique du changement d'outils (crayon, ligne, insertion de texte, etc.), notamment dans les applications de création graphique. Dans ce chapitre, la technique « palette satellitaire » est proposée : une palette d'outils orbite autour du curseur et son orientation est associée à l'orientation de la souris. On active un outil en tournant la souris et en plaçant l'outil souhaité sous le curseur. Cette opération permet que le curseur graphique ne quitte pas la zone d'intérêt de la tâche en question et évite également que l'utilisateur se déconcentre visuellement (en détournant son regard vers une autre zone de la fenêtre). Ce chapitre présente également les principes d'un mode avancé de cette technique - permettant la sélection d'outils via des raccourcis gestuels - destiné aux utilisateurs expérimentés.

Un prototype pour la navigation immersive dans un catalogue de livres numérisés est présenté dans le chapitre 6. Les facteurs qui rendent difficile la consultation des catalogues de documents numériques sont d'abord recensés. Ensuite, des arguments en faveur de l'utilisation d'*indices visuels réels* pour représenter les documents sont avancés. Dans le prototype développé, les documents sont disposés dans un espace cylindrique à l'intérieur duquel l'utilisateur peut facilement se déplacer à l'aide d'un dispositif isométrique. Un dispositif d'affichage immersif individuel, la *Visionstation*, a été utilisé afin de renforcer la compréhension de cet espace par l'utilisateur. Les obstacles à l'utilisation de ce dispositif pour ce type de tâche sont rapportés et une solution est proposée : la mise en place d'une configuration *focus plus contexte*, à partir du couplage d'un deuxième écran. L'objectif original de ce travail n'était pas d'évaluer l'interface mais de vérifier la faisabilité technique d'adapter le dispositif immersif à une tâche de navigation ayant été antérieurement explorée par Cubaud et al. (1998). Ce travail a été réalisé en 2004/2005 et constitue la première exploration de cette thèse. De ce fait, la problématique abordée et certaines contraintes techniques identifiées à ce moment-là sont aujourd'hui un peu dépassées. Cette exploration a tout de même été utile pour repérer les contraintes techniques de l'utilisation de ce type d'affichage dans

un contexte de bibliothèque numérique ainsi que pour développer les bases techniques pour un des prototypes décrit dans le chapitre 7.

Enfin, le chapitre 7 introduit la problématique de la visualisation pour le contrôle qualité des images issues d'un programme de numérisation de grands corpus de livres. Ce chapitre apporte une première contribution à ce sujet en décrivant les caractéristiques de la tâche et ses particularités qui la rendent peu facile d'être réalisée lorsqu'on se sert d'applications traditionnelles pour la consultation de photos. Des techniques d'interaction réalisées dans cette thèse sont présentées ; elles visent à faciliter la conciliation entre la visualisation du détail de certaines pages et la compréhension de la totalité de *l'ensemble de pages* (c'est-à-dire le volume ou, dans certains cas, toute une série de volumes). Ces explorations comprennent une interface Web - utilisée au quotidien dans la bibliothèque numérique du CNAM - et deux prototypes zoomables qui répondent, chacun, à certains des besoins identifiés. Ces derniers doivent encore être évalués formellement à la suite de cette thèse.

## 1.5 Publication associées

Durant cette thèse, j'ai pu participer à quelques conférences et journées d'études pour présenter mon travail. J'ai aussi cosigné un certain nombre de publications :

- R. Almeida et P. Cubaud. Visualisations de contrôle pour la numérisation massive. *Actes du 12e Colloque international sur le document numérique (CIDE 12)*. Montréal, Canada, 21-23 Octobre, pp. 177-188, Eurovia, 2009.
- R. Almeida, P. Alessio, P. Cubaud et A. Topol. A Visualization Technique for Quality Control of Massive Digitization Programs. *Proceedings of ECDL'09*. Corfu, Greece. pp. 150-155, Springer Verlag, 2009.
- R. Almeida. Palettes satellitaires : faciliter les changements d'outils avec la rotation de la souris. *Actes de la conférence IHM'08*. Metz, 2-5 Septembre, pp. 173-176, ACM Press, 2008.
- R. Almeida. Utilisation d'un périphérique à trois DLL pour la manipulation de *widgets* rotatifs. *Actes des Communications informelles de l'IHM'07*. Paris, 12-15, Novembre 2007.
- R. Almeida et P. Cubaud. Nearly-integral Manipulation of Rotary Widgets. *Proceedings of INTERACT'07*. Rio de Janeiro, Brazil. 14-17, Septembre, pp. 489-492, Springer Verlag, 2007.
- R. Almeida et P. Cubaud. Supporting 3D Window Manipulation with a Yawing Mouse. *Proceedings of NordiCHI'06*, Oslo, Norway, pp. 477-480, ACM Press, Octobre 2006.
- R. Almeida, P. Cubaud, J. Dupire, S. Natkin et A. Topol. Experiments towards 3D Immersive Interaction for Digital Libraries. *Proceedings of Edutainment'2006*. Hangzhou, China, Springer Verlag, pp. 1348-1357, 2006.
- R. Almeida, P. Cubaud, J. Dupire et A. Topol. Métadonnées et interactions



riches pour les bibliothèques numérisées. Revue *Document numérique*, 9(2) : 18, pp. 83-109, 2006. Lavoisier.

- R. Almeida, P. Cubaud, J. Dupire et A. Topol. En hommage à Ramelli : un dispositif immersif de consultation de bibliothèques numériques. *Actes de la conférence H2PTM'05 : Hypermedias Hypertexts, Products, Tools and Methods*. Hermes Science Publications, St Denis, France, 2005.
- R. Almeida et J. Dupire. Un dispositif de visualisation 3D immersif pour les documents numériques . *Actes des Rencontres des Jeunes Chercheurs en IHM (RJC-IHM'04)*, Lacanau, 20-22 Octobre, pp. 3-6, 2004.



# Etat de l'art de la manipulation rotative : dispositifs et aspects humains

*Ce chapitre présente un panorama des aspects humains et techniques impliqués dans la manipulation rotative. Quelques facteurs moteurs et perceptifs liés à l'action d'imprimer une rotation à un objet avec la main sont décrits. Par ailleurs, un aperçu de la diversité des « souris enrichies » est fourni et un état de l'art des souris rotatives est ensuite avancé.*

La captation de la rotation de la souris est un moyen d'élargir le débit de communication entre l'utilisateur et les applications bureautiques. L'ajout de la captation d'un degré de liberté (DDL) supplémentaire à la souris traditionnelle permet que ce nouveau dispositif puisse « hériter » de la plupart des points forts de la souris traditionnelle. En effet, la rotation de la souris autour de l'axe perpendiculaire au plan sur lequel on la déplace a peu d'impact sur l'utilisation habituelle (c'est-à-dire, l'action de la déplacer sur ce même plan). Ceci ne serait pas vrai si la captation de la rotation était, par exemple, autour de l'axe horizontal au plan. Les facteurs impliqués dans la manipulation rotative sont multiples. Il convient d'examiner, entre autres questions, la façon dont la main se charge de la rotation d'un objet, quels doigts agissent sur l'objet et la manière dont le geste de rotation interagit avec un geste de déplacement. Par ailleurs, une première analyse des contrôles rotatifs et des souris dotées d'un capteur de rotation nous fournit des pistes de réflexion. Ce chapitre propose de situer les souris rotatives dans un domaine d'étude plus large : celui des dispositifs d'entrée manuels, pour lesquels nombreuses taxonomies ont été créées, et des « souris enrichies », c'est-à-dire des souris couplées à des capteurs divers leur permettant de prendre en compte da-

vantage de types de manipulation.

## 2.1 Pistes pour l'exploration de la rotation

La rotation d'objets et d'instruments est une action quotidienne que l'on exécute facilement plusieurs fois par jour de façon peu attentive. Ces actions de rotation sont toutefois bien plus difficiles à réaliser sur des objets virtuels. En effet, pourquoi le temps nécessaire pour orienter un objet virtuel est-il beaucoup plus important que celui nécessaire pour orienter un objet réel, comme une tasse de café? Ware et Rose (1999) estiment qu'un grand nombre de facteurs peuvent être à l'origine de cette disparité, à savoir :

- l'éventuel désaccord entre la forme et l'aspect visuel de l'objet tourné,
- l'écart spatial entre l'objet physique et l'objet virtuel,
- le degré de pratique que possède l'utilisateur pour cette manipulation,
- le délai du feed-back graphique,
- l'utilisation des deux mains pour orienter un objet,
- la qualité graphique (le degré de réalisme visuel) de l'objet virtuel manipulé,
- le nombre de DDL et de contraintes de la tâche,
- le besoin de lâcher et reprendre l'objet au cours de la manipulation afin de trouver une prise plus confortable,
- d'autres exigences de la tâche, comme la précision demandée pour l'orientation finale.

Il est important de remarquer qu'une partie des études expérimentales liées à cette question ont été menées dans le cadre des environnements immersifs. Certaines questions sont ainsi moins pertinentes lorsqu'elles sont posées hors de ce contexte-là.<sup>1</sup>

### 2.1.1 Configurations temporelles des manipulations rotatives

Lorsqu'on effectue un geste pour tourner un objet, ce geste peut être fusionné à un geste de déplacement de cet objet. Néanmoins, ce *geste composé* ne pourra être réalisé que si une série de conditions existent. L'idée d'un « espace perceptif » de la tâche fournit quelques pistes pour analyser ces opérations composées (Simon, 1969; Norman, 1993). Cet espace perceptif est la représentation que l'on fait d'une action que l'on va réaliser ; par exemple, changer la position et l'orientation d'un objet. Les éléments visuels dont on dispose au moment de réaliser cette opération peuvent la *simplifier* au niveau cognitif. De même, les concepts relatifs aux manipulations bi-manuelles sont utiles dans ce contexte. Des analogies peuvent être

---

<sup>1</sup>Par exemple, le problème de l'utilisateur ne pas voir sa main durant la manipulation est lié au fait que lorsqu'on utilise un casque de réalité virtuelle, la main restera invisible sauf si on choisit de la représenter dans l'environnement virtuelle. De même, le capteur utilisé dans ces expériences est tenu suspendu par la main de l'utilisateur, ce qui est différent du type d'utilisation pris en considération dans cette thèse (c'est-à-dire, le capteur reste posé sur une surface horizontale).

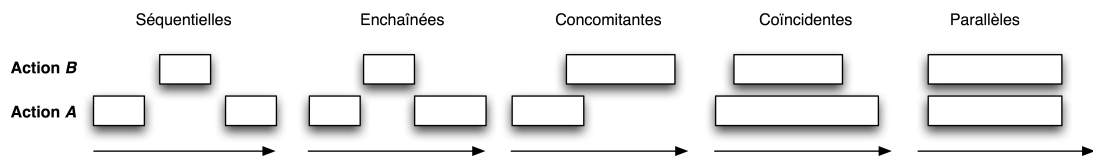


FIG. 2.1: Les cinq types de configuration décrivant la temporalité d'une action composée.

dressées entre les sous-opérations d'un geste composé et les actions coordonnées entre les deux mains (Guiard, 1987; Sellen et al., 2002).

Dans une étude pionnière, Buxton et Myers (1986) ont observé que, dans un cadre de travail bi-manuel, les utilisateurs réalisent de façon séquentielle les manipulations qui n'ont pas de rapport l'une avec l'autre (par exemple, dessiner une ligne et attraper l'outil pour l'opération suivante). Cependant, peu à peu, l'utilisateur commence à entrelacer ces deux opérations dans le temps jusqu'à ce qu'elles atteignent un certain degré de parallélisme.

Les différentes combinaisons temporelles entre les actions des deux mains ont été étudiées et décrites par Sellen et al. (2002) puis par Bailly et al. (2005). Ces derniers proposent quatre schémas descriptifs de la temporalité des actions bi-manuelles : *séquentielles*, *concomitantes*, *coïncidentes* et *parallèles* (voir figure 2.1). Ces distinctions sont utiles pour analyser les actions composées. L'*enchaînement*, une cinquième combinaison que je propose, nuance la différence entre les actions séquentielles et concomitantes :

- *séquentielles* : il n'y a pas de chevauchement entre les actions du type A et celles du type B. De plus, il y a un intervalle de transition d'une action à l'autre ;
- *enchaînées* : similaire au schéma séquentiel sauf qu'il n'y a pas d'intervalle de transition entre les deux actions. Bien qu'il n'y ait pas de parallélisme, ce type de schéma peut favoriser des actions plus fluides comme celle étudiée dans le chapitre 4 ;
- *concomitantes* : les actions sont essentiellement enchaînées mais il y a une courte période de chevauchement entre A et B au moment de la transition ;
- *coïncidentes* : l'action A est contenue dans l'action B (elle commence après B et finit avant que B ne finisse) ;
- *parallèles* : les actions A et B commencent et finissent en même temps.

### 2.1.2 Manipulation multidimensionnelle

Dans l'espace moteur, « il y aurait autant de dimensions que nous avons de muscles » (Poincaré, 1902). Ainsi, plus nombreux et diversifiés sont les muscles impliqués dans la manipulation d'un dispositif, plus nombreuses sont les dimensions que ce dispositif permet à l'utilisateur de contrôler dans l'application informatique. L'idée est donc d'étudier comment on peut contrôler davantage de

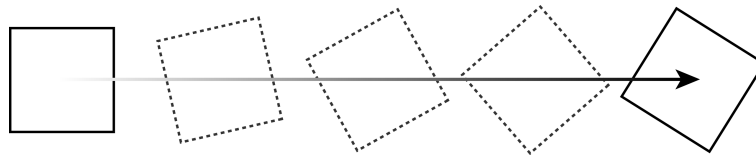


FIG. 2.2: L'objet en question dispose d'une orientation à sa position de départ qui est différente de celle à sa position d'arrivée. Il est tourné en même temps qu'on le déplace.

variables dans une applications. Etendre le dispositif d'entrée pour qu'il capte d'autres DDL est une manière ; pour cela on examine si, sur le plan gestuel, on peut réaliser ces actions plus complexes avec facilité et avec dextérité. Le déplacement et la rotation d'un objet en est un cas.

On peut orienter un objet en même temps qu'on le déplace (fig. 2.2). Si cet objet est un périphérique d'entrée, le mouvement va affecter non seulement la configuration spatiale de cet objet mais aussi les paramètres de l'application. La configuration spatiale de ce périphérique est ainsi associée à une ou plusieurs variables logiques dans cette application. Dans ce contexte informatique, le but de la manipulation n'est plus de donner une nouvelle configuration spatiale à l'objet mais d'attribuer de nouvelles valeurs aux variables contrôlées.

Une opération qui consiste à faire varier plusieurs paramètres continus peut être appelée « multidimensionnelle ». Le nombre de variables qu'on peut contrôler simultanément avec un périphérique dépend de trois aspects : perceptif, moteur et de la configuration des transducteurs<sup>2</sup> du périphérique d'entrée.

### La structure perceptive des manipulations multidimensionnelles

Dans une application informatique, l'objet d'intérêt qu'on manipule peut présenter divers attributs visuels et sonores, comme la taille, la couleur, la position et le volume sonore. Chaque combinaison d'attributs (par exemple, *couleur-taille* ou *volume-position*) peut être perçue comme un attribut composé ou comme des attributs distincts. Pour qu'on puisse passer d'une configuration logique  $E_1$  à une configuration  $E_2$  par un « geste simple », il est nécessaire que les variables de  $E_1$  soient perçues comme un attribut composé (voir figure 2.3). C'est-à-dire que malgré le fait que ce changement fasse varier  $n$  variables, on perçoit un seul changement. Un attribut composé constitue un espace perceptif « intégral » tandis que des attributs perçus de façon isolée constituent un espace perceptif « séparable » (Jacob et Sibert, 1992; Jacob et al., 1994).<sup>3</sup>

<sup>2</sup>Comme dans Wanderley et al. (2000), dans cette thèse, les termes « transducteur » et « capteur » sont utilisés de façon similaire, mais le premier désigne plutôt le principe tandis que le second désigne le dispositif physique.

<sup>3</sup>Puisque les termes « intégral » et « séparable » indiquent des prédispositions perceptives plutôt que motrices ou mécaniques, ils seront employés pour désigner les *tâches* mais pas les dispositifs.

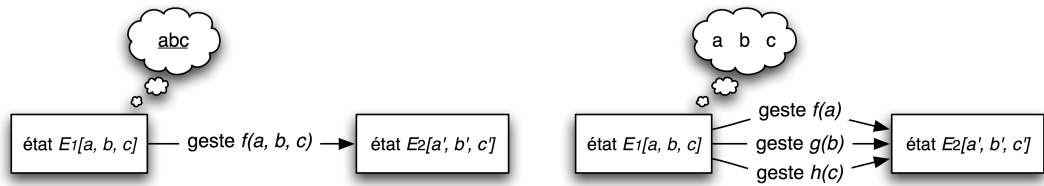


FIG. 2.3: A gauche, une tâche où les trois variables sont perçues comme un seul attribut et modifiées par un seul geste. A droite, une tâche où les trois variables sont perçues comme des attributs distincts et on les modifie avec trois gestes. Pour Jacob et Sibert (1992), la première situation caractérise une tâche intégrale tandis que la seconde caractérise une tâche séparable.

L'idée d'un espace perceptif avec des dimensions séparables ou intégrales a été d'abord proposée par Garner (1974). Toutefois, ses expériences portaient uniquement sur la reconnaissance d'attributs visuels dans des images fixes. Cette distinction a été ensuite étendue et adaptée au domaine de l'interaction par Jacob et Sibert. Si l'on prend comme exemple un écran d'ordinateur affichant un carré rouge, la taille et la couleur du carré sont des attributs séparables. Sa position (sur les axes X et Y de l'écran) et sa taille sont des attributs intégraux.

Dans l'étude de Jacob et Sibert, les participants devaient exécuter une tâche dite séparable (déplacer et changer la brillance d'un carré) et une tâche dite intégrale (déplacer un carré et changer sa taille). Les dispositifs utilisés étaient une souris (deux DDL) et un capteur magnétique (uniquement les trois DDL de translation étaient captés). Les résultats suggèrent que des tâches intégrales à  $n$  variables sont effectuées plus rapidement lorsqu'on dispose d'un périphérique d'entrée avec les mêmes  $n$  DDL.<sup>4</sup> A l'inverse, une tâche séparable produit un meilleur résultat lorsqu'on se sert d'un périphérique qui contrôle un seul DDL à la fois. Un périphérique pouvant capter plusieurs DDL sera approprié pour ce contexte s'il est doté de contraintes mécaniques permettant de manipuler un DDL à la fois sans activer accidentellement les autres DDL. Ainsi, un dispositif ne doit pas être considéré comme étant *supérieur* à un autre, mais plutôt *plus adapté* qu'un autre pour une tâche donnée.

Toutefois, ces constatations doivent être analysées avec prudence car la façon dont on manipule un dispositif résulte de plusieurs facteurs. Par exemple, n'y a-t-il pas certains couples ou triplets de DDL qui peuvent être plus facilement manipulés ensemble que d'autres ? La manipulation de trois variables via la translation du dispositif ou via sa rotation imposent le même degré de difficulté ?

<sup>4</sup>Il faut bien sûr que, sur le plan moteur, ces  $n$  DDL puissent être manipulés simultanément.

### Déplacer et orienter un objet

Dans les travaux de Jacob et Sibert (1992) cités en haut, la variation de la position du dispositif est analysée mais la variation de son orientation ne l'est pas. Lorsqu'on dit qu'il est possible de contrôler simultanément trois DDL d'un objet, l'un de ces DDL peut-être l'orientation (autour d'un axe quelconque)? Peut-on, par exemple, simultanément déplacer et tourner un objet sur un plan? En effet, il ne serait pas étonnant d'affirmer que l'on peut déplacer et tourner en même temps une tasse de thé sur la table. Mais ces opérations sont-elles vraiment simultanées? Avec quel degré de simultanéité?

Wang et al. (1998) ont étudié les aspects temporels d'une tâche où le participant déplace un cube en bois afin de le positionner et de l'orienter à l'intérieur d'un cube virtuel translucide. Une importante période de chevauchement entre la phase de déplacement et la phase de rotation du cube a été observée. Le cube cible, seul élément virtuel de l'expérience, était vu à l'aide de lunettes stéréoscopiques et apparaissait sur la même surface que celle où se trouvait le cube en bois. La main du participant, le cube réel et le cube virtuel partageaient ainsi le même espace haptique et visuel. Ces résultats ont été analysés à l'aide de deux notions : *coïncidence* et *interdépendance*.

Le déplacement et l'orientation d'un objet sont des opérations *coïncidentes* (voir figure 2.1). La phase d'orientation survient toujours en même temps que la phase de déplacement alors que normalement cette dernière commence et finit seule. La phase d'orientation est donc *contenue* dans la phase de déplacement. C'est d'ailleurs la durée de la seconde qui détermine la durée totale de la manipulation.

Par ailleurs, l'orientation et le déplacement sont des opérations *interdépendantes*. Deux opérations A et B sont interdépendantes lorsque l'*entrée* de A (par exemple, la position ou l'orientation que l'on souhaite atteindre) influe sur la *sortie* de B, c'est-à-dire sur la façon dont B est exécutée (Wang et al., 1998). Cela signifie que lorsque la durée de l'une varie, la durée de l'autre varie également. Par exemple, en augmentant l'écart entre l'angle de départ et l'angle d'arrivée du cube, la durée de la phase d'orientation a augmenté et, par conséquent, la phase de déplacement a été également prolongée. D'autre part, si la phase de déplacement est réduite (par exemple, en adoptant une position d'arrivée plus proche de la position de départ), la phase d'orientation commencera plus tôt. Le mouvement est peut-être reprogrammé de cette manière pour que l'orientation termine avant la fin du déplacement.

La configuration des mouvements dans la condition décrite ci-dessus est similaire à celle de la condition partiellement dépourvue de feed-back visuel (où le participant ne voit pas sa main ni le cube manipulé, mais uniquement le cube virtuel). Cette piste est pertinente pour la conception d'interfaces qui se servent de *mappings* absolus. Ce type de correspondance pourrait permettre à l'utilisateur de



réaliser des réglages de façon moins dépendante du feed-back visuel. Ce manque de retour visuel engendre néanmoins plus d'erreurs (plus dans le positionnement que dans l'orientation) et une durée d'exécution plus longue. Cela peut suggérer que, guidées uniquement par le canal haptique, les actions de rotation sont plus robustes que celles de déplacement.

Enfin, la tâche étudiée par Wang et al. (1998) était très peu « virtuelle » ; elle a donc un rapport limité avec le cadre d'interaction bureautique qui nous intéresse. Il me paraît, par exemple, difficile de conclure par cette expérience que la manipulation d'un capteur, associé à un objet graphique sur l'écran, produirait les mêmes résultats. Ces résultats pourraient être mieux exploités dans la conception d'*interfaces tangibles* ou de la réalité virtuelle où les espaces visuel et moteur sont les mêmes. Cette étude est tout de même importante dans le cadre de cette thèse puisque la façon dont on manipule le cube dans l'expérience est identique à celle dont on se servirait pour manipuler une souris rotative. Elle offre des pistes sur la temporalité, sur l'interférence entre les mouvements et sur le rôle des informations haptiques.

### 2.1.3 Manipulation « absolue »

Le terme « absolu » peut désigner deux stratégies de manipulation rotative qui permettent à l'utilisateur d'être moins dépendant du feed-back visuel : la *manipulation pré-programmée* et la *manipulation matérielle*.

La manipulation est dite « pré-programmée » lorsque le mouvement devient une habilité algorithmique qui n'a besoin d'aucun type de retour durant le geste (Viviani, 1994). Ce type de manipulation peut être considéré comme un « contrôle en boucle ouverte ». L'avantage de ce type de manipulation est illustré par l'expérience de Ware et Rose (1999). Dans cette étude qui consistait à orienter un objet réel, les manipulations des participants se sont montrées plus performantes lorsque l'orientation d'arrivée était toujours la même. Cela suggère que les muscles « apprennent » l'orientation finale et l'on peut ainsi atteindre celle-ci plus facilement.

La manipulation est dite « matérielle » lorsque l'utilisateur l'exécute guidé par sa proprioception et par les indications tactiles « codées en dur » sur le dispositif. Hinckley et al. (1994) ont observé que les *repères haptiques passifs* rendent plus facile ce type de manipulation. La manipulation est matérielle car elle ne se base pas sur des indications dynamiques (en général, graphiques) rendues par l'application. L'utilisateur se sert du « toucher actif » (Gibson, 1966). En effet, il se sert à la fois des indications kinesthésiques (la perception de la position relative des articulations) et tactiles pour atteindre l'orientation ciblée.

### Les référentiels spatiaux

Les « référentiels spatiaux » désignent l'ensemble de repères qui permettent à l'individu de situer son corps (ou un de ses membres) dans l'espace. En réalité, des référentiels différents peuvent guider nos actions. Ware et Arsenault (2004) évoquent quatre référentiels importants lorsqu'on manipule un objet virtuel sur l'écran.

Premièrement, le *référentiel égocentrique* situe l'espace par rapport à la position et à l'orientation de la tête de l'observateur. Puisqu'on observe ce qui entoure son corps via des mouvements panoramiques et d'inclinaison, ce cadre de référence fournit un système de coordonnées polaires centré sur la tête. Deuxièmement, le *référentiel géométrique* est défini par l'axe de la gravité, les lignes droites d'une pièce ou d'un meuble. L'approche de la perception écologique avancée par Gibson (1979) souligne également l'importance de ce type de repère basé sur les éléments environnant le sujet. Troisièmement, il y a un *référentiel de l'objet* lorsque la forme de l'outil manipulé présente une symétrie bien définie et l'utilisateur identifie clairement l'axe autour duquel il peut imprimer une rotation à cet outil. La rotation envisagée autour de cet « axe canonique » est plus facilement imaginable et programmable. Enfin, le *référentiel haptique* est déterminé par la configuration mécanique des membres qui manipulent un objet. Les impressions tactiles, la configurations angulaires des articulations reliant la main jusqu'au tronc et la force exercée par les muscles impliqués fournissent les données pour ce référentiel (Gibson, 1966).

### Perception kinesthésique et haptique

Gibson (1966) introduit le concept de « toucher actif » pour désigner l'action exploratoire des mains du sujet sur un objet. Lorsqu'on saisit un objet inconnu, ses mains naturellement le touchent, le tournent et se déplacent sur lui pour repérer les détails de forme et de texture. Hinckley et al. (1994) s'est appuyé sur ces actions pour explorer les pistes haptiques passives ; il montre comment une application peut être enrichie grâce à l'adoption d'un *mapping* absolu et d'un *prop*<sup>5</sup> dont la forme est bien adaptée à la tâche. La forme particulière de cet outil permet qu'on le remette dans une orientation donnée de façon rapide et peu coûteuse en ressources cognitives.

Par ailleurs, la forme asymétrique du dispositif peut faciliter sa prise en main. L'irrégularité de l'objet offre des coins et des extrémités pouvant être plus facilement saisis et immobilisés. Un autre avantage de l'asymétrie, lorsqu'elle est associée à une correspondance absolue : elle permet à l'utilisateur de connaître l'état logique de l'application en touchant cet objet physique.<sup>6</sup> Grâce à cette utilisation

---

<sup>5</sup>Il s'agit ici d'un objet physique qui « habille » le capteur de position-orientation. Voir Bowman et al. (2004) pour une présentation plus complète de ce type d'objet.

<sup>6</sup>Ceci est étudié par Rekimoto et Sciammarella (2000) avec la *Toolstone*.

*riche* du canal haptique, l'utilisateur peut consacrer sa vision à d'autres aspects de la tâche qu'il réalise. En revanche, cette asymétrie peut offrir une prise moins confortable dans certaines orientations que dans d'autres. Par exemple, un objet rectangulaire dont l'un des côtés est beaucoup plus long que l'autre pourrait être facile à saisir uniquement par son côté court. Il serait alors contraint de contorsionner son poignet afin d'accéder au côté court lorsque celui-ci n'est pas directement abordable.

Un moyen de comprendre les bénéfices du mapping absolu est de voir les différences entre la manipulation d'un objet réel et d'un objet virtuel. Comme évoqué au début de ce chapitre, Ware et Rose (1999) ont étudié ces différences. La manipulation est facilitée lorsqu'on voit sa main et l'objet que l'on contrôle. Le mouvement est plus rapide et plus précis que lorsqu'on ne voit ni sa main ni l'objet. Deuxièmement, le fait qu'il y ait un désaccord entre les informations haptiques et les informations visuelles sur l'objet virtuel ne s'est pas avéré un problème (il s'agissait de contrôler la position d'une manette rectangulaire virtuelle via un capteur sphérique). Cependant, la forme sphérique n'est pas une forme comme les autres, puisqu'elle ne communique aucune orientation. Les résultats seraient-ils les mêmes si le capteur disposait de la même forme de l'objet virtuel mais d'une orientation différente de celle de l'objet virtuel ?

Un objet dont la forme est complètement symétrique (par exemple, cylindrique ou décagonale) a l'avantage d'être saisi toujours de la même façon, avec une prise convenable. Ces formes sont néanmoins dépourvues de pistes haptiques indiquant son orientation. De plus, dans le cas d'un objet sphérique, la surface peut se montrer glissante et l'absence de coins peut la rendre difficile à saisir.

#### 2.1.4 Les types de mouvements manuels

Certaines études ont décrit les séquences de mouvement des doigts lorsque la main manipule, dans son intérieur, des objets relativement petits (Elliott et Connolly, 1984; Mackenzie et Iberall, 1994). Ces descriptions sont pertinentes pour cette thèse car les dispositifs rotatifs présentent cette caractéristique, à savoir d'être réorientés par les doigts, à l'intérieur de la main, et pas uniquement déplacés par celle-ci. La manipulation d'un objet via l'action des doigts n'est pas la seule possibilité. Napier (1956) distingue deux types principaux de préhension d'objets : la « prise forte » et la « prise fine » (voir figure 2.4).<sup>7</sup>

Dans la *prise forte*, l'objet reste immobilisé à l'intérieur de la main grâce au contact avec une grande surface de sa paume. Cette configuration permet à la main de saisir l'objet de façon ferme et stable : la saisie d'un marteau ou d'une raquette de tennis en sont des exemples. En contrepartie, cette configuration de la main permet des mouvements moins variés pour manipuler l'objet en question. En effet, si l'on souhaite déplacer cet objet, ce sera par des mouvements du

---

<sup>7</sup>En anglais, respectivement, « *power grip* » et « *precision grip* ».

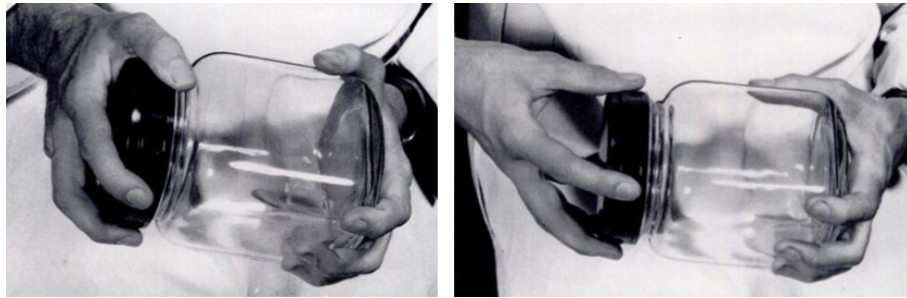


FIG. 2.4: A gauche, les deux mains se servent d'une prise forte pour ouvrir le pot. A droite, la main gauche passe à une prise fine pour tourner le bouchon plus rapidement. Figure extraite de Napier (1956).

poignet, de l'avant-bras et du bras.

Lorsqu'on se sert de la *prise fine*, on saisit l'objet entre les doigts et le pouce dans un mouvement de pincement. Elliott et Connolly notent que la prise fine ne procure pas obligatoirement une plus grande précision ou délicatesse mais plutôt une plus grande *variété* de mouvements. Cette variété peut, à son tour, contribuer à l'exécution de mouvements plus précis. Elle peut aussi favoriser des manipulations plus convenables et économes, minimisant l'implication des membres supérieurs. Ces types de mouvements utilisent une action coordonnée entre les doigts afin de changer la configuration spatiale de l'objet dans la main. Cela se fait au détriment de la stabilité et de la force de la saisie. Le nombre de doigts impliqués dans chaque manipulation varie en fonction de la taille de l'objet en question (voir figure 2.5).

Les mouvements réalisés lorsqu'on saisit l'objet avec une prise fine sont des « mouvements intrinsèques ». <sup>8</sup> L'objet manipulé reste à l'intérieur de la main et ce sont surtout les muscles de celle-ci qui sont employés. Des manipulations quotidiennes mais complexes, comme l'écriture avec un stylo, sont souvent des combinaisons de mouvements extrinsèques et intrinsèques. Comme illustre la figure 2.6, les mouvements intrinsèques peuvent être repartis dans trois catégories : *synergies simples*, *synergies réciproques* et *modèles*<sup>9</sup> *séquentiels*.

Les actions synergiques diffèrent des actions séquentielles par le fait que, dans les premières, les doigts impliqués agissent simultanément (en se déplaçant et se re-positionnant) tandis que dans les secondes, les doigts se relayent dans le temps pour que, par exemple, lorsque l'un retourne à sa position neutre, l'autre tient l'objet. Ainsi, les modèles séquentiels se basent sur une coordination par relais et engendrent des mouvements discontinus. Lorsqu'on tourne plusieurs fois un objet dans la main sans le poser, comme une balle de tennis, les cinq doigts sont

---

<sup>8</sup>En opposition à « mouvements extrinsèques », où les objets sont manipulés grâce aux muscles extérieurs à la main.

<sup>9</sup>Elliott et Connolly (1984) utilisent le terme « *patterns* », pour lequel je n'ai pas trouvé une traduction satisfaisante en français. Je l'ai remplacé par « modèle » ou « type », selon le contexte.

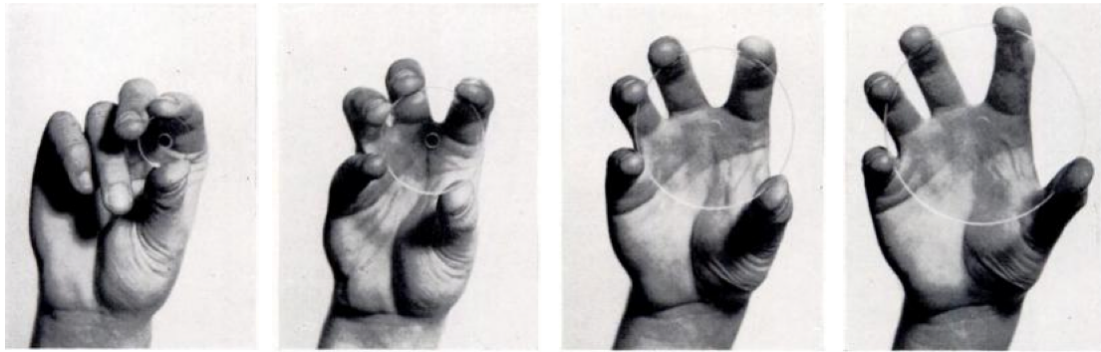


FIG. 2.5: La variation dans l'implication des doigts pendant une prise fine selon le diamètre de l'objet saisi. Figure extraite de Napier (1956).

impliqués et agissent en relais. Ce type de manipulation s'appelle « pas rotatif ».

Dans les synergies simples, tous les doigts exécutent un même mouvement (qu'il soit fléchisseur ou extenseur). Cette classe de mouvements peut être illustrée par quelques exemples : presser un ballon de caoutchouc (« pression »), arracher une écharde (« pincement »), écrire un chiffre avec un stylo (on l'appelle « trépied dynamique » puisque le stylo est stabilisé par trois doigts).

Dans les synergies réciproques, les doigts impliqués présentent des mouvements dissemblables ou réciproques. Un doigt peut exécuter un mouvement extenseur tandis que les autres réalisent un mouvement fléchisseur. Cette classe de mouvement est subdivisée selon le type de mouvement prédominant effectué par les doigts : fléchisseurs/extenseurs ou abduction/adduction.<sup>10</sup> Voici quelques exemples de cette catégorie : tourner un écrou avec les doigts (« roulage digital<sup>11</sup> »), tourner une monnaie afin de lire ce qui est écrit sur le bord de la face (« pivotement digital » - il s'agit d'une version à plus grande échelle, impliquant plus de doigts, du *roulage digital*), remonter une montre (« roulage radial »).

Le modèle « pas rotatif » est évoqué avec intérêt par Zhai et al. (1996) et par Hinckley et al. (1997) pour décrire la façon habile dont les utilisateurs manipulent un capteur magnétique à six DDL emballé dans une sphère (voir fig. 2.7). Cependant, ce modèle décrit la manipulation d'un objet dans l'espace dépourvu de tout point d'appui, ce qui est différent de la manipulation d'une souris rotative où cette dernière repose toujours sur un plan. Dans ce dernier cas, les doigts sont libres pour engager tous en même temps le mouvement de rotation et se repositionner tous en même temps également.

C'est finalement le modèle *pivotement digital*<sup>12</sup>, de type synergique réciproque, qui est le plus approprié pour décrire la manipulation d'une souris rotative (voir fig. 2.8). Dans les contextes explorés dans cette thèse, l'utilisateur se sert de ce type

---

<sup>10</sup>L'abduction des doigts est le mouvement qui écarte les doigts entre eux dans un même plan tandis que l'adduction est le mouvement de les rapprocher les uns des autres.

<sup>11</sup>Le terme « digital » est utilisé ici pour exprimer l'implication des doigts dans l'action.

<sup>12</sup>En anglais, « rock ».

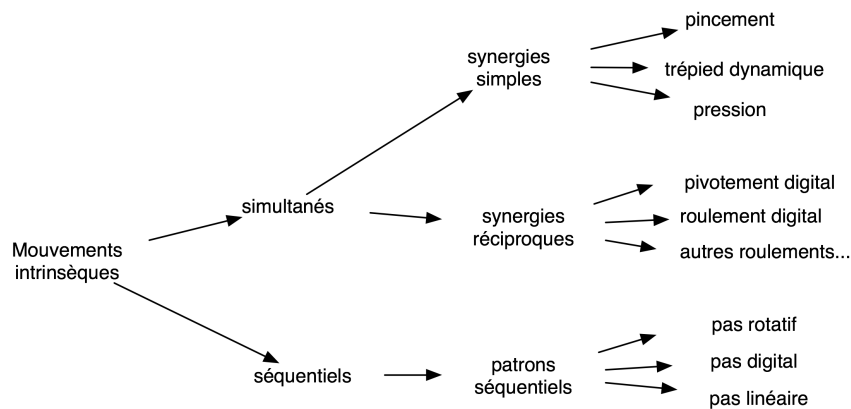


FIG. 2.6: Les types de mouvements manuels intrinsèques selon (Elliott et Connolly, 1984).

de mouvement de façon itérative. Il manie le dispositif plusieurs fois, jusqu'à ce qu'il atteigne une nouvelle orientation qui ne pourrait pas être atteinte par un seul geste.

Ainsi, j'utiliserai le terme « pas angulaire » pour désigner chaque étape du mouvement itératif de *pivotement digital*. Pour sortir le dispositif d'une orientation  $\Theta_{actuel}$  et le mettre à  $\Theta_{cible}$ , on doit le tourner de  $\Delta_{\Theta}$ . Pour cela, la main emploiera itérativement une succession de manipulations parcourant  $\Delta_{\theta_1}, \Delta_{\theta_2}, \dots, \Delta_{\theta_n}$ . Le terme « pas angulaire » désigne chacun de ces gestes intermédiaires désignés ici par  $\Delta_{\theta_i}$ .

**L'apport des doigts.** La taille d'un objet est également un facteur important concernant la façon dont on le manipule. Certains travaux évoquent le potentiel des dispositifs petits car ils favorisent des manipulations plus riches (voir, par exemple, Card et al., 1991; Zhai et al., 1996; Moggridge, 2006). En effet, en faisant appel aux groupes des muscles fins, ces dispositifs permettraient des mouvements plus rapides et précis que ceux réalisables par les muscles du bras. Cette hypothèse est étayée par le fait que ces muscles sont très bien représentés dans le cortex moteur et sensoriel. Il est important de remarquer que les doigts jouent un rôle important dans la manipulation surtout lorsqu'il s'agit d'une rotation. Le déplacement « linéaire » d'un objet, même si saisi par une prise fine, est réalisé principalement via des mouvements du poignet, du coude et de l'épaule. De même, les mouvements linéaires réalisables par les doigts, tels que décrits par les modèles pincement ou pression, sont d'une amplitude très limitée.

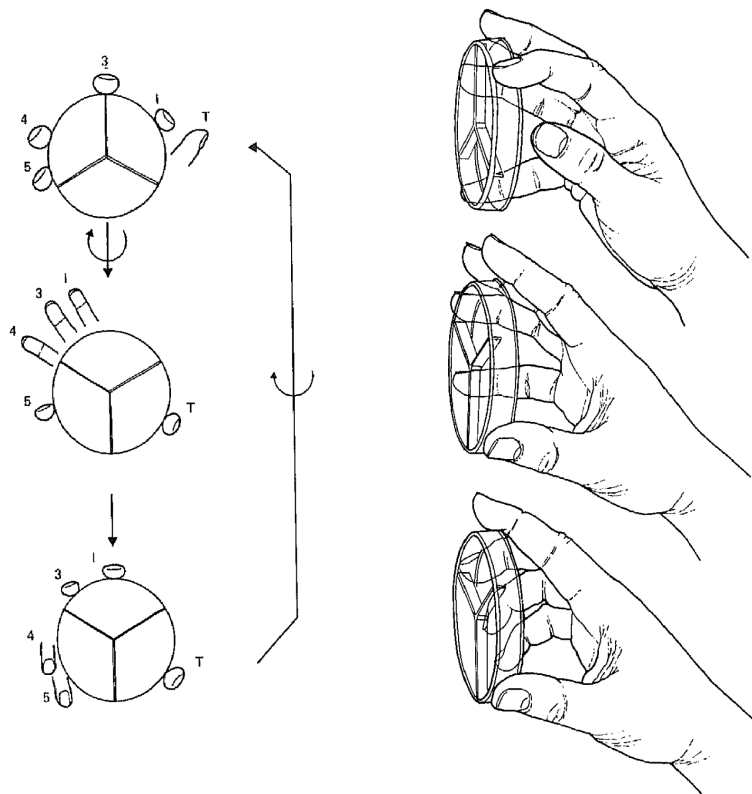


FIG. 2.7: Illustration du modèle « pas rotatif » dans une rotation de  $120^\circ$ . A gauche, les positions des doigts dans trois étapes de ce mouvement ; à droite, les postures de la main correspondantes à ces trois étapes. Figure extraite de Elliott et Connolly (1984).

### 2.1.5 Analogie avec le bi-manuel

Le cadre théorique décrivant l'action bi-manuelle comme étant coopérative et asymétrique a beaucoup influencé la recherche en IHM, notamment la réalité virtuelle et les interfaces *post-wimp*. Guiard (1987) observe que la main gauche (non-dominante) et la main droite (dominante) peuvent travailler ensemble soit de manière *symétrique*, comme lorsqu'on tire une corde avec les deux mains, soit de manière *asymétrique*, comme lorsqu'on découpe un papier avec un ciseau. Parmi les actions asymétriques, les mains peuvent avoir des rôles indépendants ou elles peuvent coopérer l'une avec l'autre. Dans ce dernier cas, la main droite et la main gauche se voient attribuer des rôles distincts l'une de l'autre. D'une part, la main gauche crée un *référentiel spatio-temporel* pour la main droite : la première commence l'action avant celle de la seconde et finit après celle-ci. Elles entretiennent ainsi une relation temporelle *coïncidente*. D'autre part, sur le plan spatial, la main gauche saisit l'objet en question, le déplace et l'oriente de sorte qu'elle crée un *espace d'action relatif* pour la main droite. C'est à l'intérieur de cet espace plus restreint que la main droite peut plus facilement intervenir, exécutant des mouve-

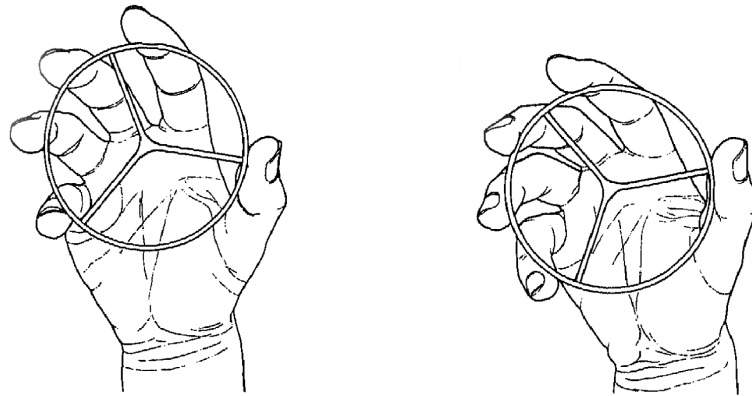


FIG. 2.8: Illustration du modèle « pivotement digital » : à gauche, rotation dans sens anti-horaire ; à droite, dans le sens horaire. Figure extraite de Elliott et Connolly (1984).

ments plus courts et plus rapides que ceux de la main gauche. L'analogie proposée par l'auteur pour décrire cette action coordonnée est celle d'un bras articulé à deux niveaux où le premier niveau raffine et réduit l'espace d'action du second.

Le contrôle de la rotation est aussi caractérisé par une action coordonnée entre des membres différents. J'identifie deux analogies possibles entre le système bi-manuel de Guiard et les manipulations rotatives : une action coopérative entre les doigts et une action interdépendante entre les gestes de déplacement et de rotation. Ces analogies sont utiles puisqu'elles pourront permettre d'étudier l'espace de conception de techniques pour une souris rotative en s'appuyant sur les techniques et sur les outils conceptuels issus des travaux d'interaction bi-manuelle.

D'une part, comme vu dans la section 2.1.4, les doigts impliqués dans la rotation d'un objet coopèrent de façon coordonnée. Cette collaboration est plutôt symétrique dans le cas de la rotation et se manifeste sous deux formes : soit ils font tous le même mouvement en même temps, comme lorsqu'on tourne une tasse de thé sur la table (le modèle *pivotement digital*), soit ils se rassemblent en deux ou trois groupes qui agissent de façon séquentielle (le *pas rotatif*). Dans ce second cas, la coopération est plus évidente car les groupes de doigts doivent se relayer pour tenir l'objet en l'air. Mais dans les deux cas, les doigts agissent ensemble pour que l'objet reste à l'intérieur de la main et pour qu'il tourne autour d'un centre stable.

D'autre part, nous avons vu plus haut que, lorsqu'on tourne et déplace un objet, ces deux composantes du geste entretiennent une relation temporelle *coïncidente* similaire à celle observée dans les actions bi-manuelles. Sur le plan spatial, la rotation est relative à la trajectoire de l'objet et non pas à un point fixe dans l'espace. A l'instar de la main gauche dans les manipulations bi-manuelles, le geste de déplacement définit le cadre temporel et spatial du geste de rotation (bien que ces actions soient interdépendantes et non pas coopératives).



## 2.2 Contrôles rotatifs

Une souris rotative peut être vue comme un dispositif hybride doté des caractéristiques d'une souris et de celles d'un bouton rotatif. La littérature en IHM est riche en études sur la première mais il n'en est pas de même pour le second. Cette section présente un aperçu des dispositifs rotatifs utilisés dans l'informatique de bureau ou dans d'autres contextes.

### 2.2.1 Types de dispositifs rotatifs

Les dispositifs de rotation sont utilisés dans l'informatique bureautique mais aussi dans d'autres contextes également, comme dans les équipements de l'audiovisuel, dans les cockpits de véhicules ou dans les bornes d'information placés dans les gares de train et métro.

**Boutons rotatifs.** Il s'agit de contrôles isotoniques. Une fois tournés, ils restent dans l'orientation où ils ont été lâchés. C'est le type le plus répandu de contrôle rotatif : ils se présentent sous plusieurs tailles sur les tables de mixage, télévisions, radios, variateurs de lumière, etc. (Jacomy, 2002). Dans les débuts de l'interaction graphique, le terminal graphique utilisé pour le *Sketchpad* était doté des boutons rotatifs qui servaient, entre autres, à contrôler le niveau de zoom de l'image affichée (Sutherland, 1964). On peut noter deux points forts de ces contrôles : l'espace réduit qu'ils requièrent sur un bureau ou sur une console (comparé à un potentiomètre linéaire) et la précision de réglage qu'ils procurent (grâce à la friction et au fait que les doigts sont impliqués dans la manipulation). Par ailleurs, des boutons plus grands sont utilisés dans les tables d'édition vidéo pour naviguer dans les morceaux enregistrés et, de façon multi-fonctionnelle, dans le contexte de l'informatique bureautique (comme le dispositif *Powermate* de la société *Griffin*). On consacre à ces dispositifs un design poussé et on mise sur le confort du toucher et de la friction que l'utilisateur sent lors du maniement. Enfin, il existe des boutons rotatifs programmables pouvant jouer le rôle de plusieurs boutons. Par exemple, le *iDrive*<sup>13</sup>, utilisé dans certains modèles de voitures *BMW*, peut dynamiquement simuler de la friction et des contraintes mécaniques.

**Molette.** La molette (présente sur la souris mais aussi sur d'autres contrôles) est souvent considérée comme un dispositif rotatif malgré le peu de « feed-back angulaire » qu'elle fournit. Ce dispositif sert à incrémenter ou à décrémenter la valeur d'un paramètre via des coups successifs qu'on y applique. Même si la partie principale de ce dispositif tourne autour d'un axe, on ne perçoit pas la « distance parcourue » et on ne sent à peine qu'il s'agit d'une rotation. Cela est dû au fait qu'une grande partie de la roue est cachée, ce qui empêche l'utilisateur de voir et sentir où est passée la partie de la roue qu'il vient de pousser. De même, le fait

---

<sup>13</sup>Voir URL : [http://www.bmw.com/com/en/insights/technology/technology\\_guide/articles/idrive.html](http://www.bmw.com/com/en/insights/technology/technology_guide/articles/idrive.html)

(le 15 avril 2009).

que la roue soit cachée empêche l'utilisateur de la saisir par un mouvement de pincement afin d'avoir plus de contrôle d'aller et de retour. Ainsi, on se sert de mouvements extenseurs et fléchisseurs d'un seul doigt pour manipuler cet objet (le premier étant moins agile que le second). Dans les bornes d'achat de ticket de la société *RATP*, on trouve aussi une grande molette servant à parcourir linéairement les options du menu affiché.

**Boutons isométriques.** Les boutons isométriques peuvent soit n'offrir aucun feed-back haptique (raideur très élevée) soit offrir un feed-back élastique (comme les *joysticks* du type manche à balai d'avion). C'est le cas de certains dispositifs six DDL, comme le *Space Navigator* de la société *3D Connexion*, ou de certains contrôles que l'on trouve aussi dans les tables d'édition vidéo. Bien que ces dispositifs offrent un feed-back très réduit et ne permettent pas d'accomplir le geste de rotation, on peut quand même le saisir comme s'il s'agissait d'un potentiomètre (par une *prise fine*) et moduler la pression qu'on y exerce avec précision. Les dispositifs de ce type sont idéaux lorsqu'on veut contrôler le paramètre vitesse d'un changement plutôt que la position d'un élément (Zhai et al., 1997). Par exemple, lorsqu'on parcourt un morceau sonore accélérant et ralentissant la lecture de temps en temps. Il en est ainsi puisqu'il suffit d'y exercer une torsion donnée pour que la valeur correspondante à cette force soit envoyée en continu et de façon continue à l'application. Un mouvement continu avec un bouton rotatif serait non seulement impossible (car il serait interrompu à chaque opération de rengagement des doigts) mais aussi très fatigant. En revanche, les dispositifs isométriques sont peu intuitifs pour une tâche de sélection, comme pointer un objet sur l'écran.

**Piste infinie.** Le meilleur exemple en est la *Clickwheel* - l'interface de navigation proposée par *Apple* dans l'*iPod* classique. Cette rotation ne concerne pas la rotation d'un objet ou d'une partie mais plutôt le glissement du doigt le long d'une piste circulaire. Ce type de manipulation permet le contrôle d'un paramètre continu via un geste ininterrompu et sans bornes qui se dispense de toute opération de rengagement ou de *clutching*. Elle est donc moins fatigante, puisqu'on ne fait pas de mouvements de re-positionnement, et plus efficace, puisqu'on peut réaliser un réglage sans aucune rupture. Ce type d'action pourrait inspirer des techniques d'interaction plus riches au niveau gestuel, comme pour contrôler la navigation dans une bande vidéo ou dans bien d'autres cas (Hsu et al., 2008). La platine vinyle, telle que utilisée par un disc jockey (comme lors d'un geste de *scratch*), est un cas limite de ce type d'interface. Dans ce cas, c'est le bras, et non pas le doigt, qui travaille et le geste de rotation ne parcourt qu'une partie du cercle.

### 2.2.2 Propriétés matérielles

Voici quelques propriétés de nature mécanique et électronique des dispositifs rotatifs qui ont un impact important dans leur utilisation.<sup>14</sup>

**Bornes et nombre de tours.** Les dispositifs sans bornes, comme la molette et le *Powermate*, peuvent être tournés un nombre infini de fois. Ils peuvent être ainsi associés à différents paramètres puisqu'ils n'ont pas besoin d'être remis à une configuration neutre à chaque fois qu'on les associe à un nouveau paramètre. Les potentiomètres rotatifs possèdent souvent des bornes mécaniques et celles-ci limitent leur rotation à une amplitude qui peut aller d'un tour (dans certains cas moins que ça) jusqu'à dix tours. L'avantage du tour unique est qu'une orientation donnée peut être directement associée à une valeur logique ou analogique déjà connue ; on peut retrouver cette valeur se basant uniquement sur les repères tactiles disponibles sur le contrôle. Quant aux potentiomètres à dix tours, ils ont l'avantage de disposer d'un espace angulaire absolu avec dix fois plus de valeurs qu'un potentiomètre à un seul tour (voir l'exploration de ces questions dans Bjorklund, 2001). Le *iDrive* peut être programmé pour simuler tous ces trois cas.

**Forme.** Les dispositifs rotatifs sont plus communément ronds, ou possèdent une forme polygonale régulière (par exemple, décagonale). Certains boutons rotatifs sont légèrement asymétriques (par exemple, en forme de flèche). Cela sert comme indication tactile et visuelle du curseur de sélection (lorsque les valeurs possibles de sélection sont indiquées autour du bouton). Les textures et les rayures qui couvrent les côtés de ces boutons servent à rendre le toucher plus agréable et à augmenter l'adhérence avec le doigt. De même, grâce au toucher, l'utilisateur peut différencier un bouton d'un autre (McCormick, 1970).

**Dimensions.** Le rayon des boutons rotatifs varie entre quelques millimètres et 10 cm. La prise utilisée pour saisir le dispositif change en fonction de sa taille. Un bouton très petit demande une prise avec deux doigts et une manipulation de type *roulage digital*. Installé sur une surface horizontale, il peut être pris par le haut ou latéralement. Les dispositifs plus grands engagent plus de doigts et peuvent demander aussi des mouvements du poignet. De même, ils exigent une prise par le haut ; le modèle de mouvement employé est alors le *pivotement digital*.

**Emplacement.** Les boutons sont plus généralement installés sur un plan horizontal (à la hauteur d'une table) ou sur un plan vertical (face à l'utilisateur). Dans ce dernier cas, l'utilisateur le manipule en se servant de la rotation de l'avant-bras plutôt que des mouvements du poignet ou des doigts.

**Captation et résolution.** Le dispositif peut capter la rotation de façon *absolue* ou *relative*. La molette ou le *Powermate* détectent la différence de valeur entre deux

---

<sup>14</sup>Les boutons rotatifs prennent une place importante dans les manuels de *Human Factors* mais, dans ce type d'ouvrage, on y analyse surtout comment la forme, la taille, la texture et la couleur de ces objets peuvent être utilisées pour qu'un opérateur distingue facilement les boutons les uns des autres dans des conditions de tension (McCormick, 1970). Or ce problème ne survient que lorsque l'utilisateur dispose de plus d'un dispositif, ce qui arrive peu souvent aux utilisateurs dans un contexte bureautique.

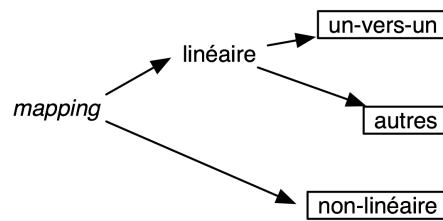


FIG. 2.9: Mappings et correspondances utilisés par les dispositifs rotatifs.

états (l'orientation relative) tandis que la plupart des potentiomètres peuvent, à tout moment, fournir une valeur correspondante à leur état. Néanmoins, un dispositif rotatif peut disposer d'une correspondance arbitraire entre les valeurs qu'il fournit et sa configuration spatiale initiale. Cette correspondance arbitraire reste possible tant que les erreurs de captation sont peu importantes. Un bouton de re-initialisation peut être utilisé pour corriger l'erreur accumulée lorsqu'elle devient trop importante et rétablir ainsi une nouvelle correspondance.<sup>15</sup>

Par ailleurs, la *résolution angulaire* désigne la plus petite variation d'orientation que le dispositif est capable de détecter. Une résolution élevée est importante dans un contexte de micro-réglages de nuances d'un paramètre (par exemple, éclairage ou volume sonore) ou dans le cas où on sélectionne un élément dans un grand espace de choix. Lorsqu'il s'agit d'un bouton à un seul tour, le nombre de cibles possibles doit être plus petit que le nombre de mesures captées.

Enfin, la *résolution temporelle* désigne la fréquence à laquelle la variation angulaire est captée. Un dispositif doté d'une fréquence faible peut ne pas être capable de détecter certains changements rapides. Par exemple, un dispositif absolu que l'on tourne puis que l'on retourne immédiatement à son orientation initiale peut ne capter que son orientation de début et celle de fin ; c'est-à-dire, l'action que l'on a réalisée ne sera pas détectée. Un cas encore plus extrême : si le dispositif se trouve dans une orientation  $15^\circ$  et si on le tourne rapidement vers  $200^\circ$ , le capteur « ignore » si la rotation a été faite dans le sens horaire ou dans le sens anti-horaire. Dans certaines applications, il est important de pouvoir faire cette distinction.

### 2.2.3 Mappings

Le *mapping* désigne souvent la fonction de transfert entre la valeur captée par le périphérique et le résultat produit par l'application (voir fig. 2.9). Dans un *mapping* linéaire, une variation de  $n$  degrés, envoie une valeur logique  $kn$ . En revanche, le *mapping* non-linéaire établit une relation qui prend en compte, outre la distance angulaire parcourue, la vitesse de ce changement (entre autres paramètres possibles). Cela permet par exemple que, par un petit mouvement de rotation, on

<sup>15</sup>On doit dans ce cas ramener le bouton à une configuration spatiale correcte vis-à-vis de la nouvelle valeur qu'on lui attribue. Je me suis servi de ce mécanisme dans le prototype que nous avons utilisé lors de l'expérience décrite dans le chapitre 3.

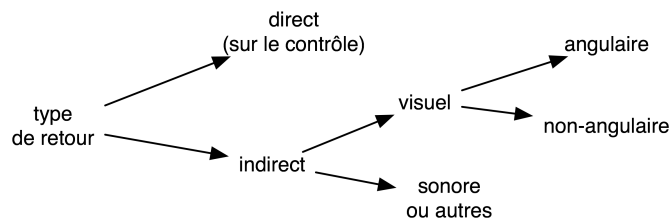


FIG. 2.10: Types de retour qui guident l'utilisateur dans une manipulation rotative.

avance de façon très « granulaire », tandis que, par un mouvement plus large et plus rapide, on avance de façon moins granulaire

Le terme *mapping* peut désigner aussi, entre autres, la correspondance qui s'établit entre la géométrie physique d'un dispositif (ou objet capté) et la géométrie d'un objet graphique. La « correspondance » indique alors le type de rapport entretenu entre l'état physique du dispositif et l'état du paramètre associé. Lorsque l'orientation initiale du dispositif est connue et le *mapping* est linéaire, on sait qu'une orientation donnée correspond à un paramètre donné. Lorsque cette relation entre l'orientation physique et le paramètre ne change pas, il s'agit d'une correspondance *absolue*. Si cette relation peut varier, il s'agit d'une correspondance *relative*. Dans le cas du dispositif contrôler un paramètre graphique angulaire via un *mapping* un-vers-un, l'orientation du dispositif et celle du paramètre sont toujours la même (voir fig. 2.10). Ceci procure l'avantage de transformer l'orientation du dispositif physique en une sorte d'interface de sortie<sup>16</sup> du paramètre auquel il est associé (Fitzmaurice, 1996).

## 2.3 Les périphériques d'entrée rotatifs

Cette section traite des souris rotatives. Toutefois, ces dispositifs sont d'abord situés dans l'univers plus général des périphériques d'entrée manuels. Premièrement, nous passons en revue quelques caractéristiques proposées par les taxonomies des dispositifs manuels. Puis, quelques exemples de « souris augmentées » sont présentés en mettant en valeur ce qui les différencie des périphériques traditionnels. Enfin, un état de l'art des souris rotatives est proposé.

### 2.3.1 Caractériser les périphériques d'entrée manuels

Qu'est-ce qu'un périphérique d'entrée ? Selon la définition proposée par Baecker et Buxton (1987), « *basically, an input device is a transducer from the physical*

---

<sup>16</sup>En simplement regardant ou touchant le dispositif, l'utilisateur peut connaître l'état actuel du paramètre de l'application associé à ce dispositif.

*properties of the world into logical parameters of an application* »<sup>17</sup>. Cette « propriété physique du monde » peut être la position du dispositif, son déplacement, la vitesse à laquelle il se déplace, son orientation, le fait d'être soulevé ou posé, le fait d'être cliqué, etc. Mais, dans le cas de cette thèse, connaître quelles sont les propriétés pouvant être captées et le type de valeur logique fournie ne suffit pas pour comprendre les particularités d'un dispositif.

En effet, l'analyse des taxonomies de dispositifs manuels facilite l'identification des particularités des souris rotatives par rapport à d'autres dispositifs manuels qui captent plus de deux DDL. Ces taxonomies sont par ailleurs utilisées dans la modélisation des dispositifs d'entrée afin de créer une couche d'abstraction matérielle. Elles peuvent permettre, entre autres, aux utilisateurs avec des besoins spéciaux de se servir plus facilement de périphériques non standards (Wang et Mankoff, 2003), aux utilisateurs avancés de créer des configurations de travail optimales (Dragicevic, 1998) ou aux utilisateurs en déplacement de changer de périphérique selon le contexte avec un minimum de re-configuration (Lingrand et al., 2005).

### Comment décrire le périphérique d'entrée ?

Les particularités d'un dispositif d'entrée peuvent être exprimées de différentes façons, pouvant être plus analytiques à partir de ses « briques de base » ou plus illustratives sans un souci de fournir une description logique exacte. Voici quelques façons de comprendre ces dispositifs.

**Le périphérique est un dispositif logique.** Dans cette interprétation, deux périphériques différents correspondent à un même dispositif logique si les informations qu'ils fournissent à l'application sont du même type (Foley et al., 1990). Il y a cinq dispositifs logiques : « *locator* », indique une position ou une orientation (le stylo sur une tablette magnétique) ; « *pick* », sélectionne un objet affiché sur l'écran (la souris) ; « *valuator* », indique une valeur réelle (un potentiomètre rotatif) ; « *clavier* » ; et « *bouton de choix* », change le mode d'interaction dans l'application (la touche SHIFT).

**Le périphérique est une fonction.** Cette fonction, en s'appuyant sur certains paramètres fournis et sur certaines variables d'état, convertit des valeurs d'un ensemble de départ vers d'autres valeurs. Selon Card et al. (1990), un dispositif d'entrée peut être défini par un *sextuple* :

$$\text{dispositif} = \{M, In, S, R, Out, W\} \quad (2.3.1)$$

où  $M$  est un opérateur de manipulation,  $In$  est le domaine d'entrée,  $S$  est l'état présent du dispositif,  $R$  est une fonction qui détermine la résolution de la correspondance entre

---

<sup>17</sup> « (...) d'une manière générale, un dispositif d'entrée est un transducteur qui fait correspondre des propriétés physiques du monde à des paramètres logiques d'une application. » (traduction de l'auteur).

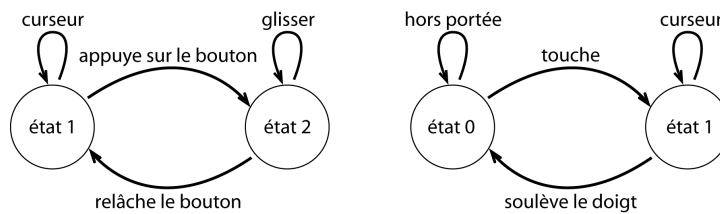


FIG. 2.11: Le modèle à trois états indiquant les états et les transitions d'une souris (gauche) et d'une tablette tactile (droite).

le domaine d'entrée et le domaine de sortie, *Out* est le le domaine de sortie et *W* est un ensemble générique de propriétés du dispositif.

**Le périphérique est un assemblage de transducteurs.** Un « opérateur de composition » peut caractériser la façon dont les différents transducteurs sont réunis dans le périphérique d'entrée (Card et al., 1990). Il y a trois types de composition : par *fusion*, par *juxtaposition* ou par *connexion*. Lorsque, par un seul geste, on peut contrôler des transducteurs distincts (qui se trouvent sur un même périphérique), ces transducteurs sont fusionnés (par exemple, la souris). La composition juxtaposée produit des dispositifs dont la disposition des transducteurs ne permet pas à l'utilisateur de les contrôler tous simultanément et de façon naturelle. Il s'agit d'une composition uniquement spatiale (par exemple, la molette sur la souris). Deux dispositifs, *A* et *B*, sont connectés lorsque *A* agit sur l'application via *B*. La souris et le curseur graphique sont des dispositifs connectés (considérant le curseur comme étant un dispositif logique).

**Le périphérique est une machine à état.** Buxton (1990) a proposé trois états (HORS DE PORTEE, CURSEUR et GLISSER<sup>18</sup>) qui explicitent les différences entre les dispositifs manuels standards. Dans la figure 2.11, la souris et le stylo magnétique sont décrits par ce modèle. Cette description associe les changements d'état « matériel » du dispositif<sup>19</sup> à des modes dans l'application graphique. De ce fait, ce modèle décrit non seulement le périphérique mais également les rôles distincts que ses transducteurs peuvent jouer dans le contexte d'une application. Ce type de représentation est actuellement utilisé pour décrire nombreuses des techniques d'interaction, parfois assez complexes (comme le font, par exemple, Forlines et al., 2006).

**Le périphérique d'entrée n'est pas qu'une entité logique.** Plus qu'une « boîte noire », il est une interface physique, dotée de contraintes mécaniques, entre l'utilisateur, doté de contraintes biomécaniques, et l'application. Certains aspects ignorés, concernant la façon dont l'utilisateur manipule le dispositif sur le plan physique, constituent sa dimension « pragmatique » (Buxton, 1983). Le *nulling problem* illustre cette dimension : deux potentiomètres, un rotatif et un linéaire, four-

<sup>18</sup>En anglais : « *out of range* », « *tracking* » et « *dragging* ».

<sup>19</sup>Dans l'exemple de la figure 2.11, ces transitions sont : le bouton enfoncé ou relâché et le fait que le signal du dispositif soit capté ou qu'il soit hors de portée.

nissent les mêmes types de données. Lorsqu'on utilise le potentiomètre linéaire pour contrôler plus d'une variable, celui-ci devra être remis à une position neutre afin d'être associé à une deuxième variable. En revanche, le potentiomètre rotatif pourra, pourvu qu'il soit sans bornes, être dissocié et associé à plusieurs variables sans qu'il soit nécessaire de le repositionner à chaque changement.

### Des propriétés pour caractériser les dispositifs rotatifs

Certains auteurs ont proposé des « espaces de caractérisation » afin de présenter les combinaisons de propriétés d'un dispositif. Le tableau de Buxton (1983) axée partiellement sur les aspects techniques des dispositifs, met l'accent aussi sur la dimension pragmatique (type de geste requis, espace occupé, manipulation directe ou indirecte, mains libres ou avec un intermédiaire). Card et al. (1990) présentent un espace de caractérisation similaire à celui de Buxton mais communiquant avec plus de précision certains aspects techniques du dispositif : la résolution de chaque transducteur, la propriété physique qu'il capte (force ou position, linéaire ou rotative, absolue ou relative), le type de composition qui lie les transducteurs dans un même dispositif (voir sec. 2.3.1), etc. Enfin, Lipscomb et Pique (1993) proposent une classification, plus pertinente que les autres pour l'étude des souris rotatives, qui s'appuie sur les aspects physiques, mécaniques et pragmatiques des dispositifs existants (voir fig. 2.12). Voici quelques propriétés utiles pour décrire les dispositifs du type souris rotative (isotoniques, posées sur une surface) :

**Imbrication ou orthogonalité :** Un dispositif captant plusieurs DDL peut être « imbriqué » ou « orthogonal ». Il est imbriqué<sup>20</sup> lorsque ses transducteurs ne sont pas « isolés » les uns par rapport aux autres. Par exemple, un dispositif captant la position en  $(X, Y, Z)$  et l'orientation en  $\theta_Y$  qui subit une rotation autour de  $\theta_Y$ , aura ses axes  $(X, Z)$  pivotés. De ce fait, un mouvement de translation imprimé à un moment  $t_1$  sur ce dispositif fournira des données en  $(X, Z)$  différentes d'un mouvement identique réalisé à  $t_0$ . Lorsque ces capteurs sont isolés, ils se trouvent dans une configuration « orthogonale » : l'orientation d'un transducteur n'influence pas la captation de l'autre.

**Empreinte sur le bureau :** Indique à la fois l'espace que le dispositif prend sur le bureau mais aussi l'espace libre nécessaire pour le manipuler confortablement.

**Bornes :** Un dispositif avec bornes ne peut pas être manipulé dans un même sens indéfiniment. Un dispositif de translation présente toujours des bornes puisque son déplacement est limité soit par l'espace de travail, soit par le fil, soit par la distance jusqu'où le bras de l'utilisateur peut aller. Certains dispositifs rotatifs, en revanche, ne présentent pas de bornes, c'est-à-dire on peut le tourner dans un même sens sans jamais atteindre une limite mécanique.

---

<sup>20</sup>En anglais, « *nested devices* ».



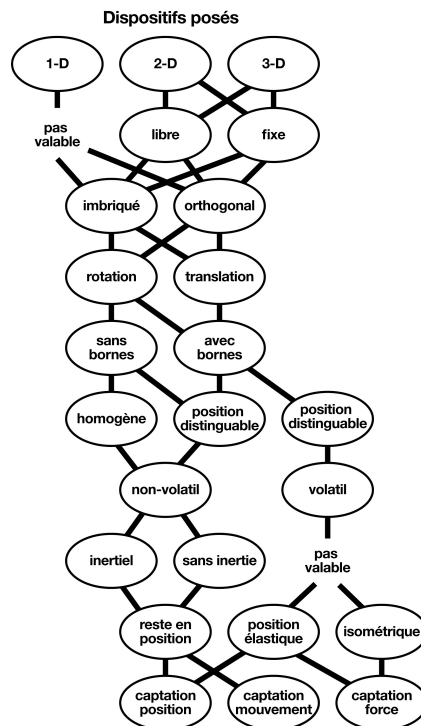


FIG. 2.12: Les propriétés des dispositifs physiques avancées par Lipscomb et Pique (1993).

**Homogénéité** : Un dispositif est homogène lorsqu'on est incapable de reconnaître quelle était sa configuration spatiale à un moment précédent. La molette de la souris et la boule du *trackball* en sont des exemples. A l'inverse, un *dispositif à position distinguable* peut être remis à une configuration déjà connue. Un capteur magnétique six DDL, un stylo à tablette magnétique et un potentiomètre d'un tour disposent tous de cette propriété. Les dispositifs qui nécessitent un mécanisme de *clutching*, comme la souris, présentent des positions distinguables aussi. Toutefois, les configurations spatiales du dispositif ne sont pas associées de manière fixe aux états logiques de l'application.

**Inertie** : Un dispositif est inertiel s'il continue en mouvement lorsqu'on le lâche suite à une manipulation. Certains *trackballs* sont inertiels, de même que la première souris à roue (Moggridge, 2006). Cette propriété est peu commune et pourrait engendrer des techniques d'interaction innovantes, tel que le feuilletage exploratoire d'un document. La plupart des dispositifs sont non-inertiels, c'est à dire qu'ils s'arrêtent dès qu'on les libère.

**Volatilité** : Un dispositif non-volatil conserve, lorsqu'il est lâché, sa position mécanique et ses valeurs de sortie. Une souris à tablette magnétique ou un *slider* en sont des exemples. Un dispositif volatil, à l'inverse, ne présente pas ces deux propriétés (ou du moins l'une d'entre elles). Par exemple, lorsqu'on libère le bâton d'un *joystick* isométrique, sa position mécanique semble ne pas subir de changement. Sa valeur de sortie, cependant, ne restera pas la

même. De même pour un écran tactile, la valeur de sortie change dès que l'utilisateur lève son doigt de l'écran. La volatilité est encore plus évidente lorsqu'il s'agit d'un *dispositif soulevé* tel qu'un stylo magnétique. Dans ce cas, lorsque l'utilisateur le libère, ni la valeur de sortie ni sa position mécanique ne sont conservées.

**Position du dispositif :** Comment le dispositif réagit-il mécaniquement à la manipulation ? Les dispositifs soulevés sont saisis et restent complètement ou partiellement suspendus pendant qu'on les utilise. Le stylo optique en est un exemple. Les **dispositifs portés** ne sont pas eux non plus en contact avec la surface de travail. Toutefois ils ne sont pas saisis, ils sont accrochés à une partie du corps de l'utilisateur. En revanche, la souris et le *trackball* sont déplacés mais ils *restent tels quels* lorsqu'on les lâche. Certains dispositifs, comme les *joysticks* élastiques, peuvent aussi être déplacés mais ils possèdent une *position neutre*, à laquelle ils retournent dès que l'utilisateur lâche le manche. Enfin, les dispositifs isométriques ne présentent aucune modification repérable dans leur configuration mécanique lorsqu'on les manipule ou lorsqu'on les libère.

**Ressources motrices de l'utilisateur :** Quels sont les membres auxquels les différents dispositifs manuels font appel ? Cette distinction en fonction du dispositif peut être illustrée lorsqu'on veut dessiner une ligne avec une souris ou avec un pavé tactile (Lingrand et al., 2005). Avec la souris, on clique sur le bouton et on fait le chemin souhaité avec la main, tandis que, avec le pavé tactile, les deux mains sont souvent utilisées, l'indicateur de la main gauche clique sur le bouton et le garde enfoncé, tandis que l'indicateur de la main droite parcourt le chemin sur le pavé.

### 2.3.2 Les souris enrichies

La littérature en IHM dispose de nombreuses propositions de souris enrichies. Avant d'analyser les souris rotatives, nous verrons quelques prototypes non rotatifs qui illustrent d'autres possibilités pour enrichir la captation de la souris traditionnelle. Ces souris - séparées ici dans deux catégories : *couplées* et *augmentées* (voir figure 2.13) - conservent la plupart des atouts de la souris traditionnelle. Il s'agit de périphériques manuels « posés » et déplaçables (donc isotoniques<sup>21</sup>).

#### Les « souris couplées »

Certaines souris (commerciales ou prototypes) sont dotées d'un dispositif secondaire couplé à sa coque. La souris à molette en est l'exemple le plus répandu. La molette peut être considérée comme étant un dispositif distinct de la souris

---

<sup>21</sup>C'est pourquoi les dispositifs *isométriques* comme le *SpaceMouse* ou le *GlobeFish* (Froehlich et al., 2006) ne sont pas mentionnés.

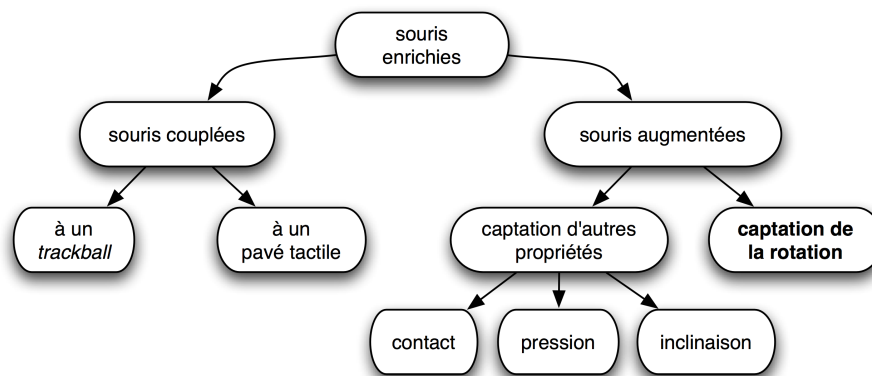


FIG. 2.13: Les « souris enrichies » présentées dans ce chapitre séparées dans deux catégories : les « souris couplées » et les « souris augmentées ».

car, bien que cela ne soit pas impossible, on ne la tourne pas naturellement en même temps qu'on déplace la souris. Il s'agit d'un dispositif  $2+1D$ <sup>22</sup> puisqu'on peut soit la déplacer, en contrôlant ainsi deux variables continues, soit tourner sa molette, en contrôlant ainsi une variable continue. De manière générale, les souris couplées sont des dispositifs  $2+nD$  et peuvent être vues comme étant le produit d'une juxtaposition de transducteurs. Un des avantages de ces dispositifs vient du fait que les doigts de l'utilisateur se trouvent en permanence sur la coque de la souris. L'utilisateur peut ainsi contrôler d'autres paramètres continus (zoom, défilement d'un document) sans avoir besoin ni de changer le mode d'interaction, ni de lâcher la souris afin d'acquérir un autre dispositif. Ce couplage réduit ainsi le *temps d'acquisition* de ces dispositifs secondaires et favorise une manipulation séparable de sous-ensembles de transducteurs. C'est-à-dire que l'on peut contrôler un transducteur sans activer accidentellement les autres.

**Couplage d'une boule.** Le *TrackMouse* est une souris étendue  $2+2D$ , disposant d'une boule de *trackball* assemblée à sa coque (Martin et Raisamo, 2004). Ce prototype a été créé pour répondre à un besoin précis : celui de contrôler deux curseurs graphiques. Le curseur secondaire, contrôlé par la boule, sélectionne l'outil de travail tandis que le curseur principal, associé à l'outil sélectionné, est contrôlé par le mouvement traditionnel de la souris. Ce dispositif ne s'est pas toutefois montré plus performant que la technique *Toolglass* (dans une évaluation contrôlée). De même, le *Trackmouse* a engendré un faible degré de parallélisme entre les manipulations des deux curseurs (Isokoski et al., 2007).

Parmi les dispositifs commercialisés, la souris *Mighty Mouse* d'Apple<sup>23</sup> propose, à la place de la molette, une mini-boule qui permet qu'on la roule librement le long de deux axes. Il s'agit donc aussi d'un dispositif  $2+2D$ . Dans une application tableur, tel que *Microsoft Excel*, on roule la mini-boule pour choisir la zone du

<sup>22</sup>Reprenant la terminologie utilisée par Buxton (1986) et Card et al. (1990)

<sup>23</sup><http://www.apple.com/mightymouse/design.html>



FIG. 2.14: Le *Padmouse*. Photo extraite de Balakrishnan et Patel (1998).

tableur affichée sur la fenêtre et on déplace la souris pour positionner le curseur sur la cellule voulue. L'utilisateur peut ainsi économiser des allers-retours vers les ascenseurs et déplacer la souris uniquement pour réaliser la tâche de sélection et d'édition cellule. Bien que ce dispositif semble identique au *TrackMouse*, la boule de ce dernier est bien plus grande que celle du premier. La micro-boule du *Mighty Mouse* peut servir pour choisir une valeur dans un petit ensemble de possibilités mais elle demande beaucoup trop de « coups » pour, par exemple, déplacer le curseur d'un bout à l'autre de l'écran.

**Couplage d'un pavé tactile.** Le *PadMouse* (fig. 2.14) est une souris sur laquelle un pavé tactile a été assemblé (Balakrishnan et Patel, 1998). Dans une configuration bi-manuelle du type *toolglass* (Bier et al., 1993), l'utilisateur manipule le *PadMouse* avec la main gauche afin de positionner les palettes graphiques et il utilise son doigt (index ou majeur) pour faire des « marques<sup>24</sup> ». Ces « marques » fonctionnent comme des raccourcis-clavier pour lancer des commandes fréquemment utilisées (enregistrer le document, zoomer d'un pas). Le vocabulaire de marques proposées pour le *PadMouse* permet à l'utilisateur d'indiquer confortablement jusqu'à 32 commandes.

#### Les « souris augmentées »

Les souris augmentées se servent de nouveaux capteurs en vue d'enrichir le vocabulaire gestuel de la manipulation ordinaire. Cette augmentation peut permettre que les nouveaux DDL soient manipulés avec les anciens DDL de la souris de façon parallèle, concomitante ou enchaînée.

**DétECTION DU CONTACT.** Hinckley et Sinclair (1999) ont proposé une souris capable de détecter, via un capteur de capacitance, lorsque l'utilisateur est en contact avec la coque de la souris. Ce capteur ne change en rien la forme de la coque (une peinture conductible est appliquée sur les parties devant être sensibles au contact),

---

<sup>24</sup>Il s'agit de gestes composés par un ou deux traits sur le pavé tactile. Le terme est issu des *Marking Menus* (Kurtenbach et Buxton, 1994).

ce qui permet qu'elle soit manipulée comme une souris ordinaire. La détection du contact fournit un état supplémentaire aux états de base de la souris, similaire à l'état « hors portée » des tablettes magnétiques (Buxton, 1990). Grâce à cette donnée, l'application peut savoir si l'utilisateur est prêt à interagir (sa main est posée sur la souris) ou s'il est plutôt en train de regarder « passivement » l'écran (sa main n'est pas en contact). Dans ce dernier cas, l'application peut, par exemple, cacher toute la partie interactive de l'interface (boutons, ascenseurs, menus) pour que, dans la fenêtre, il y ait plus de pixels consacrés à l'affichage du document.

**Captation de la pression du clic.** Certaines souris augmentées disposent d'un capteur de pression sous les boutons. Le clic est ainsi capté, non pas comme une activation discrète, mais comme une action à deux, voire plusieurs, niveaux. Par exemple, la souris *Pop-through Mouse* s'inspire des boutons des appareils photos, dont un clic léger sert à faire la mise au point tandis qu'un clic jusqu'au bout prend la photo (Zelevnik et al., 2001). En s'appuyant sur ce type de captation, la technique *Glimpse* propose d'utiliser le clic léger pour montrer à l'utilisateur ce qui va se passer s'il clique jusqu'au bout sur une option (Forlines et al., 2005). En revanche si, au lieu d'enfoncer le bouton, on le relâche, l'action n'est pas confirmée et elle subit un *undo* (avec un retour haptique). Dans d'autres prototypes, le bouton peut avoir une sensibilité à plusieurs niveaux (Omata et al., 2007). L'utilisateur peut ainsi sélectionner le niveau de zoom d'une loupe virtuelle selon la force appliquée sur le bouton. La captation à deux niveaux présente cependant un important avantage par rapport à la captation à plusieurs niveaux : elle fournit une indication haptique claire lorsque l'utilisateur atteint le niveau « clic léger ».

**Captation de la pression sur la coque.** Dans d'autres prototypes, les capteurs de pression sont installés sur des parties de la coque de la souris autres que sur le bouton. Cechanowicz et al. (2007) ont exploré deux versions de prototypes : doté d'un ou de deux capteurs de pression. L'utilisateur peut choisir le mode (couleur, transparence) ou moduler un paramètre continu (degré de transparence) de l'outil qu'il utilise selon la force avec laquelle il *saisit* la souris. Toutefois, le contrôle de la pression n'est pas homogène dans les deux sens, c'est-à-dire qu'on dispose de plus de sensibilité pour moduler la pression en serrant la main qu'en la relâchant. Une souris qui capte la pression sur deux zones de sa coque a donc été proposée. Dans ce cas, la force appliquée sur le deuxième capteur peut remplacer la modulation associée à la relâche du premier. Un autre moyen de contourner le problème de la sensibilité pour moduler la pression a été proposé par Kim et al. (2008) dans la forme d'une souris gonflable. Ce dispositif, qui capte la pression exercée sur sa coque, résout le problème de l'absence de retour haptique passif : l'utilisateur sent la dépression de la coque lorsqu'il la presse.

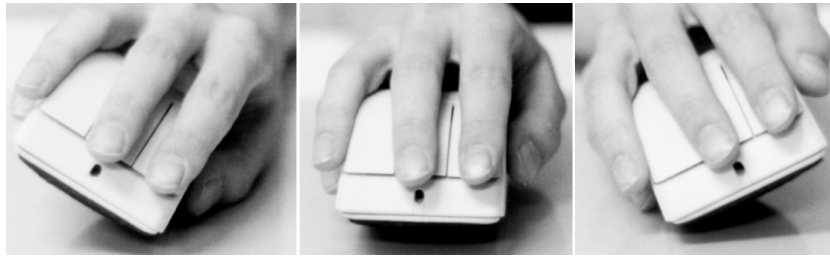


FIG. 2.15: Le *Rockin' Mouse* incliné latéralement (figures gauche et droite) et en position de repos (figure du centre). Photo extraite de Balakrishnan et al. (1997).

**Captation de l'inclinaison.** Le *Rockin' Mouse* (fig. 2.15) est une souris augmentée qui capte l'inclinaison autour de l'axe de la longitude et de la latitude de la souris (Balakrishnan et al., 1997). Il s'agit donc d'un dispositif à quatre DDL. Les bords de l'avant, de l'arrière et des deux côtés de sa base sont arrondis pour qu'on puisse facilement l'incliner. Toutefois, la partie centrale de cette base reste plate faisant en sorte que, lorsque la souris se trouve inclinée, puis ensuite relâchée, elle revient à une inclinaison neutre. Il s'agit donc d'un dispositif volatil et auto-centrant sur les axes de l'inclinaison. Cette caractéristique permet que ces axes soient utilisés pour contrôler la vitesse de changement d'un paramètre (le défilement d'un texte, par exemple). Toutefois, l'idée principale de ses concepteurs est que cette configuration permet à l'utilisateur de contrôler simultanément trois DDL (via un geste composé de déplacement et d'inclinaison).

### 2.3.3 Les souris rotatives

Dans cette section, les souris dotées d'un capteur de rotation proposées à ce jour sont présentées (voir fig. 2.16). Pour chaque dispositif, il sera indiqué<sup>25</sup> : les aspects techniques (mécanisme de rotation et le procédé de captation), les questions d'usage (la problématique de départ et les types de tâches envisagées par le concepteur), et les limites du dispositif. Tout d'abord, une mise au point sur le terme « souris rotative » : il s'agit de périphériques d'entrée manuels que l'on déplace sur une surface horizontale. La rotation est toujours autour de l'axe perpendiculaire à cette surface. Ce périphérique peut contrôler, au minimum, trois DDL (deux translations et une rotation). Sa « mécanique de rotation » peut se présenter sous deux formes : elle est *indécomposable* lorsque tout le dispositif est tourné comme un objet solide ; *décomposable*, lorsqu'une partie du dispositif est tournée tandis qu'une autre partie reste immobile.

---

<sup>25</sup>Les souris rotatives présentées ici ont été décrites par leur concepteurs dans des degrés de détail très variés. De ce fait, les descriptions qui suivent sont parfois asymétriques les unes par rapport aux autres et peu satisfaisantes pour permettre une comparaison entre les dispositifs.

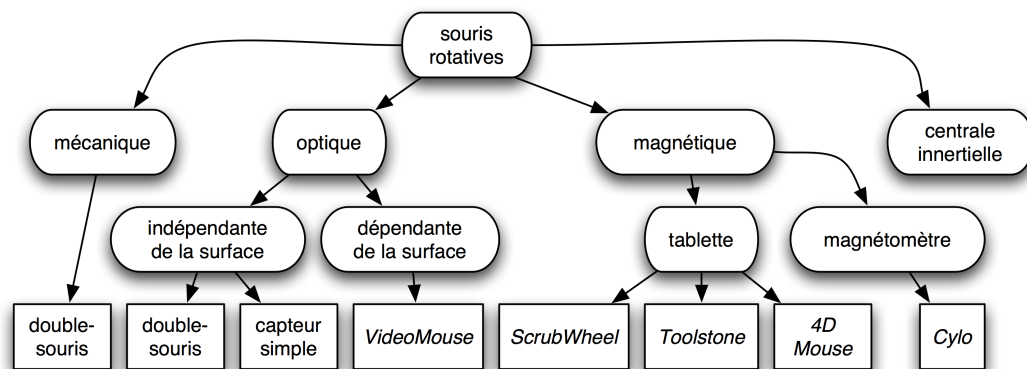


FIG. 2.16: Les variétés de souris rotatives classées selon le type de technique utilisée pour la captation de l'angle de lacet. Les boîtes rectangulaires représentent les dispositifs existants qu'ils soient des produits commerciaux ou des prototypes de recherche.

### Le Mobile scrubwheel

Le *Scrubwheel*<sup>26</sup> est un dispositif rotatif, en état de prototype, développé par la société *Wacom Technologies* (Sugiyama et al., 2000). Ce dispositif est destiné à des techniques d'interaction qui pourraient faciliter la navigation dans des médias numériques temporels, comme les vidéos ou les fichiers sonores (Fitzmaurice, 1996; Fitzmaurice et Buxton, 1999).

Le prototype est composé de deux parties concentriques, un disque central et un anneau périphérique, et doté de deux capteurs : un capteur de position placé sur le disque central et un deuxième capteur placé sur l'anneau (voir fig. 2.17). Ce deuxième capteur donne sa position angulaire par rapport au capteur du disque central. Le *scrubwheel* est utilisé sur une tablette graphique *Wacom* et cela permet une captation de position absolue. Un roulement mécanique permet qu'on tourne l'anneau indépendamment du disque central. En appliquant une pression des bords vers le centre, l'utilisateur peut, néanmoins, accroître la friction entre ces deux parties et les faire tourner ensemble. Le *scrubwheel* est un dispositif sans bornes. Cela permet à ce dispositif de contrôler la naviguer dans un espace « infini » puisqu'on peut associer chaque tour de l'anneau à un sous-ensemble différent de cadres de la vidéo. La forme du dispositif s'inspire du *jog wheel*, sorte de grand bouton rotatif que l'on trouve, par exemple, dans les tables d'édition vidéo analogiques.

**Edition vidéo.** Dans un cadre d'édition vidéo, le *scrubwheel* a été informellement testé comme un outil pour passer d'un morceau vidéo à l'autre et pour naviguer dans le temps à l'intérieur de chaque morceau. Dans cette application, plusieurs fenêtres de vidéo sont affichées sur l'écran. Le *scrubwheel* contrôle un curseur traditionnel et ce dernier active la vidéo sur laquelle il est. Le curseur de

<sup>26</sup> « To scrub » veut dire, en français, « frotter ».



FIG. 2.17: A gauche, le dispositif *Scrubwheel*. A droite, schéma du dispositif : l'anneau périphérique, contenant le capteur C2, tourne autour de l'anneau central, qui contient le capteur C1. C1 fournit la position du dispositif sur la tablette magnétique tandis que C2 fournit une orientation (relative à C1). Photo extraite de Fitzmaurice (1996).

lecture de cette vidéo devient alors associé à la rotation de l'anneau : la rotation horaire ou anti-horaire de l'anneau avance ou recule la vidéo. Cette méthode d'activation spatiale dispense l'utilisateur de réaliser une opération explicite, comme le clic, pour démarrer la navigation d'une vidéo donnée.

Le *scrubwheel* est un dispositif inertiel. On peut « lancer » l'anneau et il continuera à tourner (et à faire avancer la vidéo) jusqu'à ce qu'il s'arrête quelques millisecondes, voire quelques secondes après. Cela semble offrir de fait une méthode ludique et alternative de naviguer dans ce type de document. Cette navigation peut être en vitesse accélérée, en rembobinant ou en explorant lentement les cadres précédents et suivants. Les utilisateurs qui ont pu le tester ont apprécié cette fonctionnalité et certains sont allés plus loin dans l'exploration de la manipulation inertielle : ils se sont servis de deux mains pour manipuler la *scrubwheel* - l'une garde le disque central dans une position fixe, tandis que l'autre donne de nouvelles impulsions à l'anneau périphérique pour qu'il ne s'arrête pas de tourner.

**Limitations et perspectives.** Un des inconvénients de la forme actuelle du *scrubwheel* est l'emplacement du bouton. Il se trouve sur l'anneau et cette situation oblige les utilisateurs qui l'ont testé à le chercher lorsqu'ils veulent appuyer dessus. Fitzmaurice suggère que le bouton devrait se trouver plutôt sur le disque central. Deux autres configurations *scrubwheel* pouvant enrichir davantage son apport sont suggérées. Premièrement, le dispositif pourra être déplacé sur une surface d'affichage horizontale, afin de réduire la distance entre « l'espace d'entrée » et « l'espace visuel ». Les deux parties du dispositif sont déjà translucides et cela permettrait que l'utilisateur visionne, sous la *scrubwheel*, la vidéo qu'il navigue. Deuxièmement, un cadre d'interaction bi-manuel avec deux *scrubwheels* est envisagé. Pour l'édition d'une vidéo, un dispositif pourrait être associé à la temporalité de la vidéo source et l'autre à celle de la vidéo de destination.



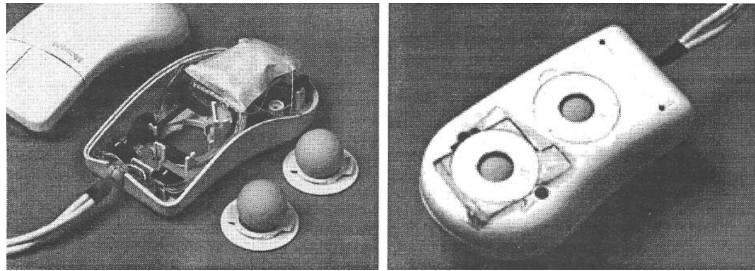


FIG. 2.18: La « souris à double-boule » : coque ouverte montrant l'emplacement des deux boules (gauche) et la base de la souris (droite). Photos extraites de MacKenzie et al. (1997).

### Les doubles-souris

Les différents projets qui se sont servis de doubles-souris rotatives sont présentés dans l'ordre chronologique. Par ailleurs, un des prototypes construits dans le cadre de cette thèse a été également fait à partir de l'assemblage de deux mini-souris optiques et sans fil (Almeida et Cubaud, 2006) ; il est décrit en détail dans le chapitre 3.

**La « souris à double-boule ».** Ce travail pionnier présente une souris à boule dont la coque a été modifiée afin d'héberger une deuxième boule et les pièces électroniques nécessaires pour la captation de son mouvement (MacKenzie et al., 1997; Zhai et MacKenzie, 1998)<sup>27</sup>. Ce prototype a été conçu à partir de la constatation que si l'on dispose de deux capteurs de position sur un plan, on peut calculer la rotation autour de ce plan. Toutefois, la façon dont on dérive la position angulaire n'est pas mentionnée. La forme du prototype est celle d'une souris filaire traditionnelle. Ce prototype présente deux contraintes qui empêchent l'utilisateur de le tourner librement : (1) il ne peut être saisi que dans une orientation<sup>28</sup> et (2) il est filaire (ce qui gêne l'utilisateur lorsqu'il le tourne de plus de 180 °).

Le prototype a été conçu en vue de faciliter les tâches d'interaction qui nécessitent de régler une variable angulaire (par exemple, la rotation d'un dessin dans un logiciel d'édition graphique).<sup>29</sup> L'utilisateur doit appuyer sur une touche du clavier pour que la captation de la rotation soit activée. Cela fournit un mécanisme explicite pour que la souris puisse être repositionnée à une orientation neutre lorsqu'on souhaite tourner un objet d'une plus grande amplitude que le poignet ne permet. La touche CONTROL installe une correspondance angulaire 1 : 1 entre la souris et l'objet logique tandis que la touche ALT installe une correspondance 5 : 1. Cette deuxième correspondance compense l'amplitude réduite du mouvement du

<sup>27</sup>Son nom en anglais est « *Two-Ball Mouse* ».

<sup>28</sup>Celle qui est confortable pour la main.

<sup>29</sup>En effet, dans les applications de dessin, la rotation d'un objet est réalisée de manière indirecte : (1) en saisissant dans la palette d'attributs la valeur en degrés ou (2) en sélectionnant le bord du dessin puis en faisant une trajectoire circulaire avec le curseur.

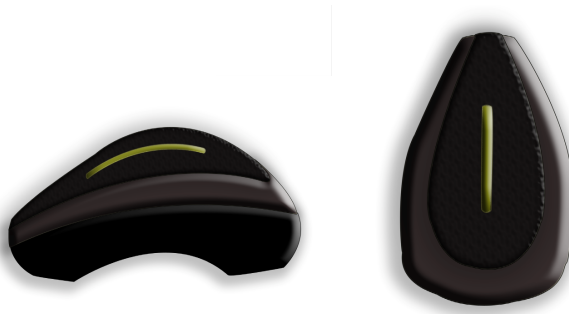


FIG. 2.19: La coque de la souris rotative proposée : vue latérale (gauche) et vue du haut (droite). Figure extraite de Fallman et al. (2007).

poignet et une grande rotation peut être alors effectuée sans que l'on soit obligé de recentrer la souris.

**Couplage de deux souris à l'aide d'une boîte à outils.** Olwal et Feiner (2004) présentent un prototype de double-souris rotative à trois DDL sans fil. Ce dispositif illustre comment un nouveau dispositif pouvant capter plusieurs DDL peut être facilement pris en compte au niveau logiciel par la boîte à outils que les auteurs ont développée. Le API proposé permet de créer des interfaces à des « dispositifs composés » (comme deux souris collées l'une à l'autre) pour que ces nouveaux dispositifs se comportent comme des « dispositifs natifs »<sup>30</sup>. Ces prototypes assemblés pourraient ainsi être rapidement construits et testés, ce qui accélérerait la conception de nouveaux périphériques d'entrée et techniques d'interaction.

**Forme et résolution d'une double-souris.** Fallman et al. (2007) ont réalisé un travail se focalisant sur les aspects de forme de la coque d'une souris rotative et sur la résolution des capteurs optiques utilisés. Deux mini-souris optiques ont été utilisées<sup>31</sup> ; leurs capteurs ont été enlevés et connectés à un micro-contrôleur programmable. Ce dernier se sert d'une connexion sans fil pour communiquer avec l'ordinateur hôte. Le prototype étudié dispose d'une précision de 5040 mesures dans une rotation de 360 degrés, ce qui donne 14 mesures par degré. Cette précision permet que l'on contrôle la rotation complète d'un objet graphique via un mouvement de petite amplitude (comme les mouvements d'abduction et d'adduction du poignet). Par exemple, si l'on ne dispose que de 30 degrés d'amplitude pour tourner confortablement la souris, on pourra tout de même disposer de 420 mesures.

---

<sup>30</sup>C'est-à-dire, pour que le programmeur de la technique n'ait pas besoin de différencier un dispositif à six DDL composé à partir de trois dispositifs deux DDL d'un dispositif à six DDL composé d'un seul capteur six DDL.

<sup>31</sup>Le capteur optique ADNS-2610 intégré aux mini-souris Belkin utilisées dans cette expérience est doté d'une résolution de 400 CPI (*counts per inch*) et pouvant capter des déplacements jusqu'à une vitesse de 30 cm par seconde.

Pour la définition des aspects de forme du dispositif, les auteurs se sont appuyés sur un cadre conceptuel appelé *Product Semantics*. Cette approche met l'accent sur les impressions que les propriétés physiques du dispositif provoquent chez les utilisateurs quant à son origine, à son but et à la manière de s'en servir. Dans une séance de maquettage, les auteurs ont développé une coque de souris trois DDL (fig 2.19). Cette coque, d'abord esquissée sur papier et ensuite construite sous forme de maquettes en polystyrène, est similaire à celle d'une souris classique mais elle porte un trait qui va de son arrière jusqu'à son devant. De même, la base de cette souris n'est pas plate, mais courbée. Les auteurs argumentent que ces deux aspects suggèrent à l'utilisateur le fait que ce dispositif permet de contrôler des paramètres angulaires.

Un atelier « focus groupe » a été organisé avec un groupe d'utilisateurs avancés (graphistes). Les retours obtenus dans cet atelier suggèrent que les dispositifs trois DDL seraient plus facilement acceptés s'ils conservaient la forme d'une souris ordinaire et permettaient des manipulations rotatives dans certains contextes précis (par exemple, dessiner une courbe dans *Adobe Illustrator* ou réorienter une carte dans *Google Maps*). Ces mouvements rotatifs seraient courts mais dotés d'une grande précision, ce qui compenserait la forme « mono-orientée » de la coque du dispositif.

**Le *TwistMouse*.** Le *TwistMouse* est un prototype de souris trois DDL construit à partir de l'assemblage mécanique de deux souris optiques traditionnelles (Hannagan, 2007). Selon les auteurs, ce dispositif enrichi doit pouvoir répondre à trois contextes d'usage : (1) manipulation souris classique, (2) manipulation  $2D+1D$  (la variation angulaire joue le rôle de la molette) et (3) manipulation trois DDL (on l'utilise pour tourner un objet en même temps qu'on le déplace).

Deux souris filaires ont été collées, l'une devant l'autre (voir fig. 2.20). La moitié arrière de la souris d'avant est sciée et enlevée pour que le tout ne soit pas trop long. Le bouton, la molette et la partie arrière de la souris arrière ont été conservés. Les deux souris se connectent à un hub USB qui se trouve à l'intérieur de la nouvelle double coque.<sup>32</sup> L'avantage de cette configuration vient du fait que sa partie arrière, qui n'a presque pas été touchée, permet que le prototype soit manipulé comme une souris normale lorsqu'on ne souhaite pas le tourner.

Une étude d'usabilité a été conduite auprès de 24 participants pour vérifier quel mode de rotation d'objets graphiques est le plus « performant ». Trois configurations ont été testées : le « cliquer+glisser », la rotation de la molette et la rotation de la souris (avec un *mapping* de 1 : 2 entre le mouvement de rotation et la rotation du graphique). Pour ces trois configurations, les utilisateurs se sont servis du même prototype décrit. La tâche consistait à déplacer des carrés, depuis l'extérieur

---

<sup>32</sup>Les bibliothèques de fonctions de deux projets similaires, *MouseSim9* (<http://www.mit.edu/~srimano/research/mushaca/AndNowMushaca/install.html>) et *CPNMouse* (<http://cpnmouse.sourceforge.net>), ont été utilisées dans l'implémentation du pilote du *TwistMouse*.



FIG. 2.20: Le *TwistMouse* : vue de la partie supérieure (en haut) et vue de la base (en bas). Photos extraites de Hannagan (2007).

d'une grille, vers les cases cibles dans cette grille et de les orienter correctement. La « performance » a été mesurée selon trois critères : (1) la rapidité dans l'exécution de la tâche, (2) l'aboutissement de la tâche (si achevée, avec quelle précision ?) et (3) la préférence des participants. Le *TwistMouse* rotatif ne s'est pas montré ni plus performant que les deux modes comparés en matière d'achèvement de la tâche ni en matière de précision du réglage. En ce qui concerne la rapidité, il s'est avéré moins performant que la souris à molette. Enfin, les participants ont préféré réaliser la manipulation par la souris à molette. Certains participants ont suggéré qu'une coque moins longue et plus large serait plus confortable. D'autres participants ont critiqué le fait que, dans la configuration rotative, ils tournaient involontairement les carrés au moment de les déplacer.

### Le *Toolstone*

Le *Toolstone*, une petite boîte rectangulaire de 2,5 x 4 x 5 cm (fig. 2.21, droite), est un dispositif à cinq DDL multi-fonctions (Rekimoto et Sciammarella, 2000). Sa taille réduite et sa technologie non-filaire contribuent au fait que l'on puisse le tourner facilement sur un plan et le retourner selon plusieurs orientations. Les dimensions différentes de chaque côté aident à ce que chaque face soit facilement identifiable par le toucher et fonctionne comme un point d'accroche pour la rotation.

C'est la face du haut de la boîte qui indique la fonction à laquelle il est associé. Six fonctions principales sont possibles puisque toutes les configurations spatiales sont permises. En effet, ce dispositif présente un mécanisme innovant pour faciliter l'utilisation de plusieurs outils avec un même périphérique : au lieu d'utiliser des modes temporels, dans lesquels la souris reste associée à l'outil (exemples : pinceau, loupe) jusqu'à ce qu'on clique sur un autre, le mode activé du *Toolstone* est représenté par la face supérieure de la boîte.

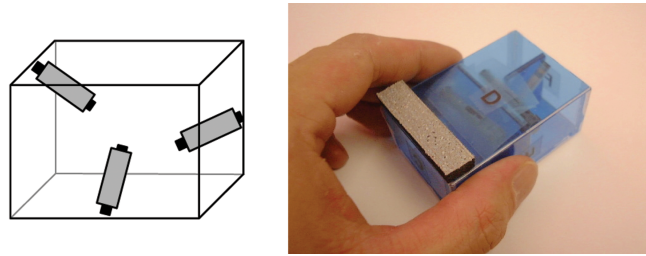


FIG. 2.21: Le dispositif *Toolstone* : schéma indiquant l'emplacement des trois bobines à l'intérieur de la boîte du dispositif (gauche) et photo du prototype testé (droite). Figure et photo extraites de Rekimoto et Sciammarella (2000).

**Captation et correspondances.** La configuration spatiale du *Toolstone* est captée par une tablette magnétique *Wacom*, sur laquelle il est utilisé. Ce dispositif contient, à l'intérieur de sa boîte, trois bobines disposées de telle sorte que l'application peut savoir à chaque instant : (1) la position de la boîte sur la tablette, (2) la face de la boîte touchant la tablette, (3) l'orientation absolue de la boîte (par rapport à la tablette), l'inclinaison (autour des axes X et Z) de la boîte (voir fig. 2.21, gauche). En réalité, uniquement le signal d'une bobine est capté à la fois mais cela suffit pour pouvoir identifier la bobine (et donc la face associée) et son inclinaison.<sup>33</sup> Un pilote a été créé pour interpréter les données générées par la tablette et les convertir dans des événements *STONE* qui arrivent dans des paquets avec les informations énumérées ci-dessus.

Bien qu'il soit capable de capter les variations de position sur les cinq DDL, la manipulation n'est pas homogène d'un DDL à l'autre. Les mouvements les plus confortables et ceux captés intégralement sont ceux effectués sur le plan (déplacement X, Z et rotation autour de Y). Outre le fait qu'il n'est pas confortable de garder la boîte inclinée, la captation de l'inclinaison est plafonnée à 45° dans chaque sens de chaque axe (au-delà de cet angle, le dispositif active le mode associé à la face inclinée la plus proche). Le fait d'être *volatil* sur les axes d'inclinaison, et disposant donc d'une position d'inclinaison neutre, fait que la manipulation de ces deux DDL soit plus efficace lorsqu'elle est associée à un *mapping* de vitesse (Zhai et al., 1997). Le prototype ne dispose pour l'instant pas de boutons.

**Exemples d'utilisation.** Selon les auteurs, le *Toolstone* pourrait être utilisé dans une interaction bi-manuelle. La main dominante, munie d'un stylet, s'occuperait des opérations de dessin, tandis que la main gauche, avec le *Toolstone*, sélectionnerait la palette d'outils (chaque face correspondrait à une palette). Une fois la palette sélectionnée, l'outil pourrait être sélectionné via la rotation du dispositif. Par ailleurs, il pourrait être utilisé pour contrôler l'orientation d'une caméra dans une scène 3D.

Les concepteurs du *Toolstone* envisagent de faire évoluer sa forme pour qu'il

---

<sup>33</sup>Les bobines sont distinguables les unes des autres grâce à la résonance du signal propre à chacune.

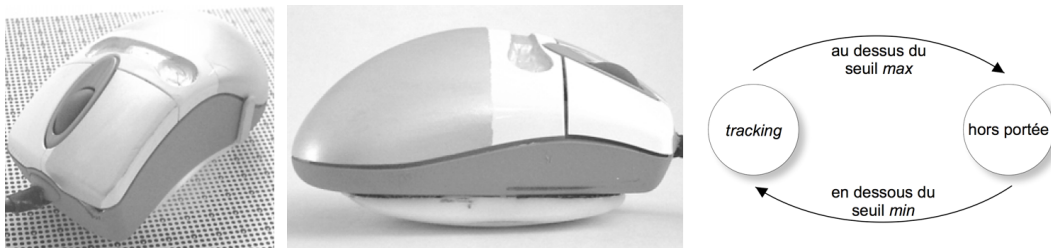


FIG. 2.22: Le *VideoMouse* : vue du haut (gauche) et vue latérale montrant la base arrondie du dispositif (centre). A droite, diagramme d'état illustrant le basculement entre les modes « *clutching* » (« hors portée ») et « *tracking* ». Photos extraites de Hinckley et al. (1999).

soit plus facilement retournable et inclinable. Par exemple, en arrondissant les côtés comme le *Rockin'Mouse* (Balakrishnan et al., 1997). Des géométries différentes seraient également étudiées pour que l'*affordance* du dispositif soit différente selon la façon dont on le saisit (Norman, 2002). Le *mapping* serait alors plus ou moins sensible selon la finesse de la prise effectuée. Enfin, un scénario d'usage dans lequel des utilisateurs avancés disposeraient de plusieurs *Toolstones* est suggéré. Chaque dispositif serait associé à une application ou à un ensemble d'outils nécessaires à l'accomplissement d'une tâche à l'intérieur d'une application. Ces divers périphériques varieraient dans la forme et dans l'étiquetage visuel rendant les transitions entre outils une opération tangible. C'est-à-dire, on changerait de dispositif physique afin de changer de dispositif logique.

### Le *VideoMouse*

Le *VideoMouse* est une souris pouvant capter six DDL, formé par une caméra tournée vers le bas et une coque de souris traditionnelle, boutons compris, couvrant la caméra (Hinckley et al., 1999). En raison de la taille de cette camera, ce dispositif est légèrement plus haut qu'une souris ordinaire (fig. 2.22). En revanche, sa base est plus petite que celle d'une souris, se limitant à une zone centrale plate avec les bords arrondis. Cette configuration, inspirée du *Rockin'Mouse*, permet à l'utilisateur de l'incliner plus facilement autour des axes X et Z. De plus, le fait que le centre de la base soit plus dense que sa périphérie permet au dispositif de revenir à une inclinaison nulle lorsqu'il est lâché incliné.

**Captation.** Le *VideoMouse* doit être manipulé sur un tapis où un motif spécial est imprimé. C'est grâce à ce motif, composé par une grille avec des points de repère non-ambigus, que les mouvements linéaires et l'orientation du *VideoMouse* peuvent être calculés. La camera éclaire la grille à l'aide de six LED rouges et l'image résultante est numérisée par une carte de capture vidéo. La dérivation des six DDL qui indiquent la *pose* de la camera est faite par un deuxième ordinateur. Le système peut ainsi déduire les mesures absolues, c'est-à-dire relatives au tapis, de la position, de l'orientation sur le plan, de l'inclinaison et de la hauteur du

*VideoMouse*. Les informations de l'orientation et de position sont *imbriquées* et les données issues du déplacement du dispositif sont toujours corrigées par rapport à son orientation.

**Mécanisme de débrayage et manipulation verticale.** Un mécanisme de *clutching implicite* est émulé afin que le *VideoMouse* puisse être utilisé comme un périphérique relatif (fig. 2.22, droite). Puisque la hauteur du *VideoMouse* est mesurée, on peut détecter lorsqu'on soulève le dispositif. La solution adoptée prend en compte deux paramètres de hauteur, un *seuil minimal* et un *seuil maximal*.<sup>34</sup> Le dispositif est en mode *clutching* lorsque sa hauteur dépasse le seuil maximal et il est réassocié au curseur lorsqu'il est en-dessous du seuil minimal (voir fig. 2.22, droite). S'il ne disposait que d'un seul seuil, le système oscillerait entre les deux états lorsque ses paramètres d'entrée se trouveraient dans le voisinage du seuil (le simple tremblement de la main pourrait engendrer cette oscillation). Un mécanisme de débrayage *explicite* (garder la touche `CONTROL` appuyée) a aussi été implémenté afin de tirer profit de la captation de la position verticale de la souris. La manipulation en hauteur du dispositif présente des bornes en bas (le tapis) et en haut (la mise au point de la caméra est fixe et limitée à une distance de 2,5 cm). En pratique, lorsque l'utilisateur contrôle la position verticale d'un objet graphique, il peut le monter mais pas le descendre par rapport à sa hauteur de départ.

**Correspondances.** Le *VideoMouse* a été informellement testé dans une tâche de sélection et d'alignement de cubes dans une scène 3D. Trois correspondances entre la rotation du dispositif et celle d'un cube sélectionné ont été proposées : 1 : 1, 1 : 4 et un contrôle par vitesse. En raison de sa forme, qui privilégie une seule orientation pour la prise, et par le fait qu'il soit filaire le *VideoMouse* n'offre pas une rotation sans bornes. Ainsi, le fil et la posture fixe de la main sont gênantes lorsqu'on essaie de faire un mouvement plus large disposant d'un *mapping* 1 : 1 car le mouvement sera limité à la rotation que le poignet peut imprimer. Dans le contrôle par vitesse, au fur et à mesure que l'on dévie le *VideoMouse* de son orientation « neutre », la souris augmente la vitesse à laquelle le cube tourne. Toutefois, comme le *VideoMouse* ne dispose pas d'un « mécanisme auto-centrant » pour cet axe (permettant à l'utilisateur de revenir facilement à l'orientation neutre), ce contrôle n'était pas facilement maîtrisé par les utilisateurs.

Une technique alternative et bi-manuelle a été proposée pour contrôler la rotation d'un objet : garder la souris fixe avec une main et tourner le tapis avec l'autre. L'utilisateur peut alors tourner en même temps le tapis et la souris (pouvant ainsi atteindre une amplitude angulaire plus importante). Par ailleurs, dans une scène 3D, l'utilisateur peut contrôler, d'une part, l'orientation de la scène via la rotation du tapis et, d'autre part, l'orientation d'un objet dans la scène via la rotation de la souris.<sup>35</sup>

---

<sup>34</sup>La gestion du basculement entre deux états à l'aide de deux seuils distincts est un mécanisme d'hystérésis.

<sup>35</sup>Pourvu que les bords du tapis disposent d'un capteur de contact par lequel l'application sait que l'on agit sur le tapis.



FIG. 2.23: La souris à trois DDL *Cylo* : vue latérale supérieure (gauche) et vue de la base (centre). A droite, le prototype qui a été construit. Figures et photo extraites de Crew (2002).

### La souris *Cylo*

La souris *Cylo* est un dispositif rotatif rond (75 mm de diamètre) et sans fil (Crew, 2002). Ce dispositif est décrit par un projet très détaillée de sa coque, de sa mécanique et de son électronique (fig. 2.23, gauche et centre). Un prototype fonctionnel de ce dispositif a également été construit (fig. 2.23, droite) et, bien qu'il ne dispose ni de la forme ni de la mécanique du projet, son électronique et son implémentation sont suffisamment performantes pour montrer la faisabilité de l'idée.<sup>36</sup> Ce prototype capte le déplacement par une caméra ordinaire d'une souris et se sert d'un magnétomètre (*solid-state compass*) pour capter son orientation absolue (par rapport au champ magnétique terrestre). Selon l'auteur, ce périphérique pourrait être utilisé dans les contextes les plus variés, comme les jeux, la CAO, les interfaces de production multimédia et toutes autres applications qui font appel à la molette (par exemple, défilement d'un texte ou changement du facteur d'agrandissement d'une image).

**Référentiels et initialisation.** Les capteurs de mouvement et d'orientation de la souris *Cylo* sont *imbriqués*. Cependant, puisque le dispositif connaît de manière précise son orientation absolue, les axes de translation sont corrigés régulièrement (pour qu'ils soient relatifs à l'écran et non pas à la souris). De même, le prototype est doté d'un bouton d'initialisation angulaire et on doit l'activer lorsque la configuration de l'espace de travail change. Cette fonction associe son référentiel absolu (nord de la Terre) au référentiel de l'espace du bureau. Ainsi, avant de cliquer sur le bouton, l'utilisateur tourne le dispositif afin d'aligner la flèche du bouton avec le « nord » de son espace de travail (par exemple, l'écran).

**Symétrie de la souris et disposition des boutons.** Cette souris, disposant d'une coque symétrique, prend en compte l'idée qu'un dispositif rotatif ne doit pas disposer d'une « orientation canonique ». De même, il ne doit pas privilégier, du point de vue ergonomique de la prise, une orientation donnée plutôt qu'une autre. Le prototype communique avec l'ordinateur hôte par un canal radio et l'absence

---

<sup>36</sup>L'auteur indique qu'une limitation importante de ce prototype est la faible résolution temporelle de la communication avec l'ordinateur hôte. Cela semble nuire à la fluidité des manipulations.



de fil permet de le tourner sans contraintes. Concernant le clic, ce projet innove en proposant deux nouveaux types de boutons afin qu'ils puissent être cliqués quelle que soit l'orientation de la souris. Le premier s'actionne en appuyant sur la souris du haut vers le bas sur n'importe quelle région de sa coque. Le second s'actionne par une prise « en pince », c'est-à-dire qu'on doit presser latéralement la souris sur deux points, de l'extérieur vers le centre.

**Manipulation décomposable.** Concernant le déplacement et la rotation de la souris, le projet prévoit que l'utilisateur puisse facilement les opérer soit séparément, soit simultanément. Une « rotation accidentelle » du dispositif serait inévitable si la souris était constituée par une seule pièce mécanique. C'est la raison pour laquelle ce projet prévoit des roulements mécaniques ; le mouvement de rotation est ainsi pris en compte par un anneau latéral (détaché de la base du dispositif).<sup>37</sup>

### Captation angulaire à partir d'une souris optique simple

La technique de captation dont disposent les souris optiques actuelles est insuffisante pour calculer les variations angulaires opérées par l'utilisateur. Poston et al. (2006) proposent d'enrichir ce procédé de captation afin que sa rotation puisse être estimée. D'après les auteurs, la captation de ce DDL pourrait servir, entre autres, à rendre plus naturelles les opérations de rotation d'objets graphiques dans un éditeur de dessin. Par ailleurs, cette prise en compte de la rotation pourrait être utilisée pour corriger les mouvements 2D de déplacement du curseur (en mesurant et compensant la variation angulaire involontaire qui survient le long d'un mouvement linéaire de pointage). Ainsi, si l'« accélération » du curseur est désactivée et si l'utilisateur ne réalise pas des mouvements de *clutching*, la correspondance entre la position du curseur et celle de la souris serait « quasi-absolue ».<sup>38</sup>

**Captation du déplacement relatif et obstacles.** Une souris optique traditionnelle utilise une micro camera CCD qui capte 1000 images par seconde. Ces images capturent une zone carré de 1 mm de côté et disposent d'une très faible définition, à savoir 18x18 pixels. Pour le calcul du déplacement relatif, on se sert de certains « points d'intérêt » qui se trouvent dans cette image. Ces points d'intérêt sont issus des détails de la surface sur laquelle la souris se déplace. Toutefois, un point ne peut être « suivi » que si, lors de l'échantillonnage, il « devient » un élément graphique faisant au moins un pixel.

On estime que les mouvements de rotation sont réalisés à une vitesse très faible, tel que 10° par seconde. A cause de cette lenteur et de la taille réduite de la fenêtre d'acquisition, la captation traditionnelle ne suffit pas pour repérer la variation angulaire. Par exemple, une rotation de 10° qui dure une seconde sera échantillonnée sur 1000 images. Dans ces conditions, il n'y a aucun point d'intérêt

<sup>37</sup>Il s'agit d'une solution similaire à celle adoptée par les concepteurs du *scrubwheel*.

<sup>38</sup>Par exemple, en déplaçant la souris d'un point  $A_{physique}$  à un point  $B_{physique}$  et puis de  $B_{physique}$  de retour à  $A_{physique}$ , le curseur, à la fin, se trouverait à nouveau sur le point  $A_{graphique}$ .

qui varie plus d'un centième de pixel (donc, pas de « pixels connus » dont les trajectoires peuvent être suivies). Ce n'est qu'au bout de 100 images qu'une variation angulaire peut être détectée. Toutefois, puisque la rotation vient souvent accompagnée d'une translation, les points d'intérêt de l'image  $I_n$  ne seront plus présents dans l'image  $I_{n+100}$ .

**Calcul de la rotation.** Afin de surmonter ce problème, les auteurs ont mis au point un procédé de calcul de la variation angulaire qui (1) utilise tous les pixels de l'image, plutôt que d'analyser uniquement quelques points d'intérêt et (2) inspecte les variations le long d'une série plus longue d'images, plutôt que d'observer les variations uniquement entre l'image précédente et l'image courante. Une approche originale a été adoptée pour explorer les problèmes de captation et de calcul de la variation angulaire. Ils ont utilisé une grande photo d'une surface en bois (d'une résolution similaire à celle des images obtenues par le capteur de la souris) et ont ensuite créé un *viewport* de 18x18 pixels qui pouvait être déplacé et tourné sur cette image. Cette simulation de la caméra de la souris leur a permis de rapidement tester la faisabilité du procédé et de l'adapter.

L'intervention nécessaire pour intégrer cette technique à la souris de base porte uniquement sur certaines composantes électroniques et algorithmiques du système optique : le micro-contrôleur de la souris devra être remplacé pour être capable de traiter ce flot de pixels beaucoup plus important. Quant à la forme de la souris, les auteurs ne trouvent pas que l'utilisation d'une coque traditionnelle pour une souris qui capte la rotation soit gênante ou apporte des contraintes. Au contraire, ils reconnaissent comme un atout le fait que, à part la puce qui doit être rajoutée à la souris traditionnelle, toutes les autres pièces de la souris peuvent rester telles quelles.

## 2.4 Des interfaces graphiques rotatives

Cette dernière section passe rapidement en revue quelques expériences où la rotation est représentée dans l'interface graphique. La rotation d'objets qui nous entoure est une des opérations quotidiennes que l'on réalise lorsqu'on souhaite reconnaître un objet, en chercher des détails ou le manipuler. On peut aussi tourner les yeux ou la tête afin de reconnaître ou de mieux observer un objet. Par ailleurs, un nombre important de panneaux de contrôle et d'affichage d'informations (analogique et numérique) communique des valeurs continues ou discrètes à travers la variation angulaire d'un curseur.

Pourtant, la rotation d'objets dans les interfaces graphiques est peu utilisée. Quelques exemples d'interfaces qui se servent de ce type de transformation pour enrichir la présentation de documents sont présentés dans les paragraphes suivants.

Il y a deux types de documents affectés par la rotation. D'une part, les *docu-*

*ments visuels*, où l'objet d'intérêt est visuel et la rotation lui est directement appliquée. C'est le cas des photos, cartes, fenêtres et modèles 3D. D'autre part, les *instruments rotatifs* sont ceux qui agissent sur des *documents indirects*, c'est-à-dire des documents qui ne sont pas transformés par la rotation. Il peut s'agir d'une propriété d'un document qui ne peut pas être *directement* ajustée. Des exemples sont la transparence d'une image, le volume d'un morceau d'un fichier son. Il y a donc des instruments rotatifs qui jouent le rôle d'intermédiaires entre le périphérique d'entrée (qu'il soit rotatif ou pas) et les propriétés du document. Le potentiomètre graphique en est un exemple : il peut simuler un *switch* rotatif ou un potentiomètre à réglage continu.

### Utilité de la rotation d'un document

La rotation de documents visuels peut avoir une des fonctions ci-dessous :

**Gestion des pixels.** Il peut avoir deux types de manipulation rotative qui contribuent à ce que l'utilisateur optimise l'espace de son écran. D'une part, la rotation d'un document autour de l'axe vertical produit des transformations de perspective. Cette transformation conserve des propriétés importantes de l'image tournée. Le document occupera moins de pixels sur une dimension tout en restant visuellement intelligible. Ainsi, au fur et à mesure que l'utilisateur met le document en perspective, ce dernier devient une icône de l'image originale. D'autre part, la rotation autour de l'axe perpendiculaire à l'écran est plus commune. Cette fonctionnalité, disponible dans tous les logiciels de conception graphique, peut aussi être utilisée comme un moyen d'explorer visuellement des documents graphiques. L'utilisateur peut opérer cette rotation sur des documents « empilés » afin d'exposer une partie du document qui se trouve couvert. Lorsqu'une fenêtre A chevauche une fenêtre B, le fait de tourner A, dévoile les coins de B. Cela donne à l'utilisateur un moyen de le sélectionner B ou de voir des indices visuels relatifs à son contenu. Cette manière de travailler avec des piles de documents est mise en place lorsqu'il s'agit de documents papier et cela a inspiré une technique d'interaction analogue (Beaudouin-Lafon, 2001; Agarawala et Balakrishnan, 2006).

**Différencier les documents.** La rotation d'un document peut être utile non pas pour la distorsion qu'elle engendre dans un document mais pour le contraste qu'elle crée entre deux documents voisins. Ainsi, sur un écran où plusieurs images sont disposées l'une à côté de l'autre, le fait de pouvoir varier l'orientation de quelques unes est un moyen de les identifier. Cette variation permet, d'un coup d'oeil, de repérer un ensemble d'éléments différents des autres. Par ailleurs, Kruger et al. (2004) ont identifié deux rôles de la rotation du document dans les interfaces *tablettop* qui entrent dans cette catégorie : la *coordination*, lorsque l'orientation d'un document indique à qui il « appartient » et la *communication*, lorsqu'on tourne le document vers un interlocuteur dont on veut attirer l'attention.

**Lecture et exploration « épistémique ».** La compréhension de certaines images

demande un effort important de la part de l'utilisateur (Kruger et al., 2004). C'est le cas de certaines photos chargées de détails, schémas et cartes. Prenons l'exemple d'un plan de rues d'un quartier ; plus son orientation est différente de celle du référentiel de l'utilisateur, plus du temps passera avant que l'utilisateur ne puisse s'en servir (Darken et Cevik, 1999). De ce fait, la possibilité de tourner des documents visuellement complexes contribuerait pour les rendre plus « lisibles » et réduire ainsi la charge cognitive que l'on dépense pour réorienter mentalement l'image (Shepard et Metzler, 1971). Ce type de manipulation, par laquelle on cherche à mieux comprendre un objet, s'appelle une action « épistémique » (Kirsh et Maglio, 1994).

**Exploration d'un volume.** La surface continue d'un volume virtuel peut être utilisée comme une surface d'affichage des documents. La distorsion visuelle imposée par le volume aux images sert comme un filtrage visuel ; elle en privilégie quelques-unes au détriment des autres. L'exploration du volume via sa rotation favorise une navigation fluide dans cet ensemble d'images. Ce type de transformation offre un moyen rapide de redéfinir quelles images sont dans le *focus* et lesquelles sont, déformées ou cachées, dans la périphérie (comme dans le *XGL-Cube*<sup>39</sup> ou dans le *Carousel View* de Wang et al. 2005).

---

<sup>39</sup><http://www.novell.com/products/desktop10/features/xgl/>

# Manipulation de documents dans un espace 3D

*Ce chapitre traite de la manipulation de fenêtres dans un environnement 3D. Une évaluation comparant quatre techniques pour la rotation et le déplacement d'une fenêtre dans un espace 3D est décrite. Les résultats sont analysés et des questions liées aux gestes de manipulation et à la forme de la coque de la souris rotative sont discutées.*

L'une des tendances des interfaces de bureau actuelles est d'intégrer des métaphores et des techniques 3D au sein du paradigme 2D conventionnel. Cette approche se trouve appliquée dans des projets comme le *Looking Glass*, des interfaces du *Mac OS X* (*Cube*, *CoverFlow*, *iChat*) ainsi que dans des éléments du *Microsoft Windows Vista*. Un nombre important de projets de recherche a également été réalisé dans cette direction, tel que dans le *Web Forager* (Card et al., 1996), le *Task Gallery* (Robertson et al., 2000), le *3D Desktop* (Topol, 2000), le *Metisse* (Chapuis et Roussel, 2005) ou le *BumpTop* (Agarawala et Balakrishnan, 2006). Cette démarche est facilitée par la démocratisation de l'accès aux cartes graphiques dotées d'une mémoire graphique dédiée, ce qui a permis que le rendu en temps réel d'images en 3D devienne une opération banale.

Malgré la réussite de ces interfaces du point de vue graphique, les nouvelles dimensions qui y sont exploitées requièrent des métaphores d'interaction appropriées. Ces nouveaux *widgets* 3D proposent ou bien une manipulation peu intuitive de ces degrés de liberté (DDL) supplémentaires ou bien une animation pré-définie qui change la configuration du *widget* d'une position canonique à une autre. Des techniques et des dispositifs adaptés à contrôler ces nouveaux DDL permettraient à l'utilisateur de se servir des documents d'une façon plus directe et efficace. C'est pourquoi j'ai exploré l'utilisation d'un dispositif à trois DDL pour la manipulation naturelle d'une fenêtre dans un espace 3D.

Dans ce chapitre, tout d'abord la problématique de la manipulation de docu-

ments dans un espace tridimensionnel est approfondie. Ensuite, un périphérique d'entrée rotatif (pouvant capter trois DDL), qui sert à déplacer et à orienter une fenêtre dans un espace tridimensionnel, est présenté. Ce chapitre détaille le prototype de souris rotative utilisé mettant l'accent sur les aspects qui sont nouveaux par rapport aux dispositifs décrits dans le chapitre 2. Puis, l'évaluation réalisée est décrite. Dans cette expérience contrôlée, les performances de quatre techniques ont été comparées dans l'exécution d'une tâche de placement d'une fenêtre dans un espace 3D. Deux techniques étaient basées sur la souris traditionnelles et deux autres sur un prototype de souris rotative - dans l'une la coque de la souris rotative était ronde, dans l'autre elle était rectangulaire. Enfin, les résultats sont discutés et quelques conclusions tirées de cette étude sont présentées.

### 3.1 Des techniques 3D pour la gestion de fenêtres

L'étude rapportée dans ce chapitre s'est focalisée sur la métaphore du document dans les bureaux 3D.<sup>1</sup> L'insertion de fenêtres dans un espace 3D permet à l'utilisateur d'optimiser son espace écran grâce à la possibilité de visualiser facilement plusieurs fenêtres en même temps (voir aussi le chap. 2.4). En effet, la métaphore d'espace 3D offre un moyen de rendre plus « malléable » la configuration graphique des fenêtres sur l'écran. Cela peut être réalisé via l'agrandissement, la réduction ou la rotation des fenêtres. Au lieu d'être soit couverte soit visible, la fenêtre peut occuper moins d'espace, devenant plus petite ou se rétrécissant sur une dimension grâce à une distorsion de perspective. En bref, ces transformations géométriques qui peuvent être appliquées aux fenêtres permettent à l'utilisateur de gérer son espace de travail de manière plus *nuancée*. Il s'agit ainsi d'une alternative au style d'interaction des « fenêtres chevauchantes ».

La gestion de fenêtres concurrentes est une ancienne problématique de recherche en informatique. Des techniques qui ne sont pas 3D ont, bien sûr, déjà été proposées. Par exemple, *Pad++* est un bureau « zoomable » - l'utilisateur peut agrandir ou réduire son espace de travail (Bederson et Hollan, 1994). Dans les « fenêtres élastiques », les fenêtres ne chevauchent pas ; elles sont positionnées les bords des unes contre les bords des autres. En tirant ses bords vers l'extérieur ou vers le centre de l'écran, l'utilisateur peut facilement re-configurer l'espace de l'écran accordé à chaque fenêtre (Kandogan et Shneiderman, 1996). Enfin, des techniques d'interaction ponctuelles - par exemple, plier un bord de la fenêtre ou la pivoter légèrement - ont également été proposées en vue de permettre à l'utilisateur d'obtenir des indices visuels sur les fenêtres couvertes (Beaudouin-Lafon, 2001).

Parmi les approches 3D, peu d'attention a été consacrée à des techniques qui permettent à l'utilisateur de régler de façon intuitive et granulaire l'orientation

---

<sup>1</sup>Un espace de travail virtuel où les documents peuvent être déplacés non seulement le long des axes X et Y de l'écran mais aussi en profondeur.

des fenêtres. Par exemple, dans *Web Forager*, l'un des ancêtres de la visualisation 3D de documents, on ne mentionne pas la façon dont la rotation du document est prise en compte. Apparemment, un document peut être mis uniquement dans deux orientations : 0 ou 60°.

De même, dans *Task Gallery*, interface qui simule une galerie dont les tableaux sur les murs représentent les fenêtres des applications, une fenêtre ne peut avoir que deux orientations : elle est collée soit au mur latéral, soit au mur qui est au fond du couloir. C'est-à-dire, elle apparaît droite ou orientée de 45°. L'interface *Looking Glass* semble permettre quatre orientations définies en avance. D'autres interfaces, comme le bureau immersif de (Topol, 2000) ou Métisse (Chapuis et Rousset, 2005), proposent un contrôle plus fin et continu basé sur le déplacement de la souris. Cependant, la manipulation devient peu intuitive : on doit indiquer un changement de mode - par une touche du clavier ou un bouton de la souris - puis, ensuite, la rotation de la fenêtre est associée au mouvement de la souris le long de l'axe X ou Y avec un *mapping* arbitraire.

Il est vrai que la manipulation d'un objet dans un espace 3D peut être vue comme une opération à six DDL. Dans Métisse, par exemple, on peut aussi pivoter la fenêtre autour de l'axe orthogonal à l'écran. Toutefois, dans les interfaces citées ci-dessus, c'est la rotation autour de l'axe vertical qui est prise en compte le plus souvent. Le fait que la fenêtre soit contrainte au « sol » rend plus naturel la correspondance de celle-là avec la souris, qui repose et se déplace toujours sur un plan. L'étude présentée dans ce chapitre a restreint les mouvements dans cet espace à trois DDL : la translation le long des axes de profondeur et horizontal (droite-gauche) ; la rotation autour de l'axe vertical.

### 3.1.1 Une correspondance directe entre dispositif et fenêtre

Dans la plupart des projets de manipulation de documents dans un espace 3D, deux aspects des opérations impliquées ne sont pas traités en profondeur : l'intégralité de la tâche et la mise en place d'une technique permettant un contrôle angulaire fin et rapide.

Un contrôle de la rotation direct et intuitif me semble une condition essentielle pour la gestion optimale de ces opérations. Les techniques de rotation basées sur une correspondance arbitraire, comme celles des projets ci-dessus, imposent une surcharge cognitive à l'utilisateur ; il doit concentrer son attention visuelle sur la rotation qu'il imprime à la fenêtre par son mouvement linéaire. Ce qui devrait être banal, comme orienter un papier sur une table, devient gourmand de concentration et demande parfois des mouvements précis.

Par ailleurs, ces techniques séparent les opérations de déplacement et de rotation de la fenêtre. Elles imposent une manipulation via des opérations séquentielles à une tâche qui est perçue comme intégrale (Jacob et al., 1994).<sup>2</sup> Or, non

---

<sup>2</sup>Les tâches intégrales et séparables sont expliquées plus en détail dans le chapitre 2.1.2.

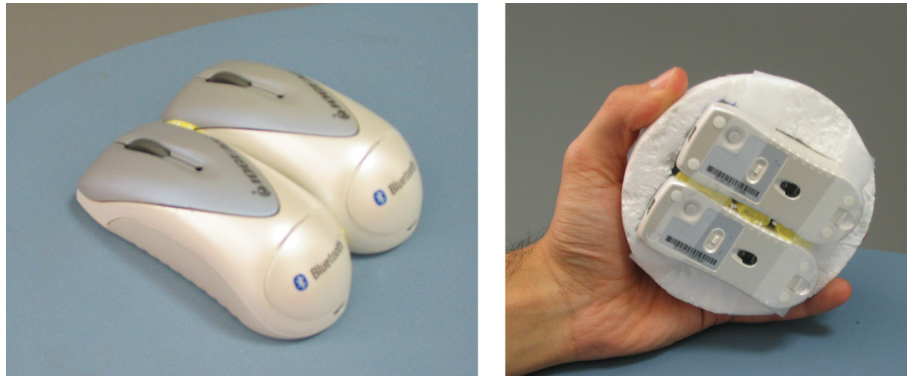


FIG. 3.1: Prototype développé d'une souris à trois DDL à partir de deux souris sans fil.

seulement la position et l'orientation d'un objet peuvent être perçues comme des attributs intégraux mais, de plus, l'étude de Wang et al. (1998) montre que lorsqu'on déplace et on tourne un objet physique, cette action comporte une importante phase de simultanéité.

Une technique d'interaction - basée sur une souris rotative et destinée à la manipulation trois DDL de fenêtres - peut donc contribuer à faciliter ces opérations. Ce dispositif rotatif contrôle les trois paramètres spatiaux de la fenêtre de façon fine, simultanée et via une correspondance naturelle.

## 3.2 Construction d'un prototype de souris rotative

Afin de tester les hypothèses sur l'adéquation d'un prototype à trois DDL au type d'opération décrite en haut, j'ai construit un prototype de souris rotative. Ce prototype a été réalisé à partir de deux mini-souris *Bluetooth* sans fil. Ces deux dispositifs ont été collés, la latérale de l'un à celle de l'autre (voir figure 3.1).

Normalement, lorsque connectées, les deux souris seraient associées à un seul curseur et le système *X11* ne montrerait pas leurs événements comme étant issus de périphériques distincts. En modifiant le fichier *XF86Config*, on associe les flots de données de chaque souris à des noms de périphériques<sup>3</sup>. La bibliothèque *XInput*, une extension de *X11* qui gère les périphériques non-standards, est utilisée dans notre application en C. Elle nous permet de récupérer séparément le couple de coordonnées  $(x, y)$  de chaque souris.

Les coordonnées  $(X_1, Y_1)$  - pour la souris gauche - et  $(X_2, Y_2)$  - pour la souris droite - sont initialisées à 0. La rotation est calculée à partir de l'écart qui se crée

---

<sup>3</sup>Ces flots de sortie sont récupérables via le fichier qui est activé dans le répertoire */dev/* lorsqu'on connecte la souris (ou lors de l'initialisation du système). Malheureusement, pour les souris *Bluetooth*, le fichier utilisé souvent change de nom à chaque fois que la souris se connecte.



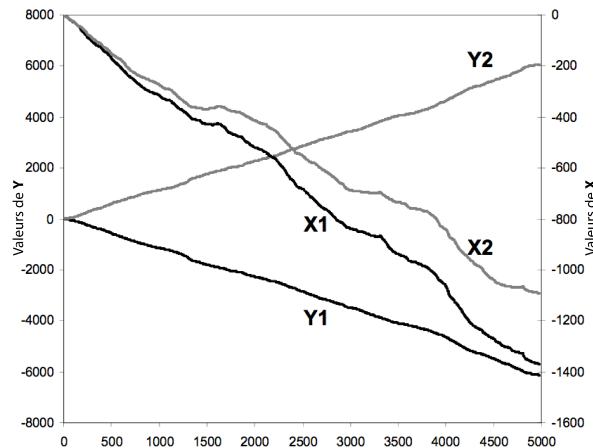


FIG. 3.2: Evolution des valeurs  $X_1$ ,  $Y_1$ ,  $X_2$  et  $Y_2$  pour une rotation libre de  $3 \times 360^\circ$  (4983 mesures).

entre  $Y_1$  et  $Y_2$ <sup>4</sup>. Lorsque la souris rotative est tournée en sens horaire,  $Y_1$  croît tandis que  $Y_2$  décroît.

La figure 3.2 montre la variation des coordonnées de la souris durant une rotation libre de  $720^\circ$ . Une interpolation linéaire s'applique bien à ces quatre coordonnées. La différence entre  $Y_1$  et  $Y_m$  (la moyenne de  $Y_1$  et  $Y_2$ ) est une valeur proportionnelle à l'angle de lacet  $\theta$ . Le coefficient de proportionnalité  $K$  sera défini, parmi d'autres facteurs empiriques, par la distance qui sépare les centres des capteurs optiques des deux souris. Ainsi, la position relative du prototype est donnée par  $(X_M, Y_M)$  et son orientation relative par  $\theta = K(Y_1 - Y_M)$ .

Le fait qu'uniquement l'orientation relative peut être obtenue avec un tel dispositif n'est pas clairement mentionné dans les autres travaux qui proposent des double-souris (voir la section 2.3.3). Par exemple, si on tourne la double-souris de  $90^\circ$  dans le sens horaire et ensuite on la déplace vers la droite, ses capteurs détecteront un mouvement le long de l'axe Y (rapprochant la souris de l'utilisateur) et non pas le long de l'axe X. En réalité, il s'agit d'un problème typique des dispositifs *imbriqués* (voir section 2.3.1) : le capteur d'orientation est mécaniquement relié à celui de translation et ils ne disposent pas de repères absolus.

Afin de contourner ce problème, nous définissons  $\theta_{tot}$ , une variable à laquelle est ajoutée itérativement toutes les mesures de  $\theta$  depuis l'initialisation de la double-souris. Les coordonnées corrigées  $(X_{fin}, Y_{fin})$  de la double-souris sont dérivées de la façon suivante :

$$X_{fin} = \sin(\theta_{total})X_m + \cos(\theta_{total})Y_m \quad (3.2.1)$$

<sup>4</sup>Le choix d'utiliser les Y plutôt que le X est empirique : Nous avons observé, en visualisant les valeurs fournies au long d'une rotation, que le Y varie de façon plus marquée et régulière que le X. Cela est peut-être dû aux caractéristiques des capteurs optiques des souris utilisées.

$$Y_{fin} = \cos(\theta_{total})X_m - \sin(\theta_{total})Y_m \quad (3.2.2)$$

La précision des mesures captées par ce dispositif est toutefois limitée. Au fil du temps, les erreurs de captation dans  $Y_1$  et  $Y_2$  font  $\theta_{total}$  accumuler une erreur trop importante, ce qui rend les coordonnées  $(X_{fin}, Y_{fin})$  inexploitable. Le prototype est néanmoins assez robuste pour servir à l'exploration des potentialités d'un tel dispositif et pour nous permettre de comparer une technique à trois DDL avec celles basées sur une souris normale.

**Initialisation du prototype.** Comme le prototype construit ne dispose pas de repères absolus, ses capteurs doivent être alignés par rapport au référentiel de travail de l'utilisateur lorsqu'on l'initialise. A ce moment, la souris prendra  $0^\circ$  comme valeur pour son orientation absolue de départ. En pratique, le devant de la souris doit être perpendiculaire à l'écran de l'ordinateur au moment où on lance l'application.<sup>5</sup> Cet alignement fait partie du protocole de démarrage de la tâche dans le cadre de l'évaluation réalisée.

### 3.3 Une étude comparative

Nous avons conduit une expérience contrôlée dont l'objectif était de comparer l'efficacité et le confort d'une technique basée sur la souris rotative avec des techniques traditionnelles basées sur une souris normale. Un autre aspect étudié était l'impact que pourrait avoir la différence dans la forme de la coque de la souris - un dispositif rotatif doté d'une coque asymétrique par rapport au même dispositif doté d'une coque symétrique. Voici les caractéristiques des quatre *techniques* qui ont été comparées (la figure 3.3 illustre les quatre techniques) :

**La souris rotative asymétrique.** Il s'agit de la double-souris telle que décrite dans la dernière section, c'est-à-dire qu'on la manipule en saisissant directement les deux mini-souris qui la composent. Ce dispositif à trois DDL permet de contrôler la position et l'orientation d'un objet dans une scène 3D via une correspondance naturelle. Sa forme rectangulaire (6,5 cm par 8,5 cm de côté et 3 cm de hauteur) avec un côté plus long que l'autre, présente des angles qui favorisent certaines prises et défavorisent peut-être d'autres. La coque asymétrique fournit une indication claire d'orientation, ce qui aide, peut-être, l'utilisateur (en lui fournissant des repères tactiles) lorsque l'orientation de l'objet contrôlé coïncide avec celle du dispositif. Cette asymétrie est peut-être dérangement lorsque son orientation et celle du dispositif contrôlé sont en désaccord. Une correspondance 1 : 1 a

---

<sup>5</sup>Pour le prototype de double-souris couvert par une coque ronde (décrit dans la prochaine section), nous avons inscrit une flèche en haut de sa coque. On doit ainsi orienter le prototype afin que la flèche soit toujours perpendiculaire à l'écran au moment de lancer l'application.

été établie entre la rotation du dispositif et celle de la fenêtre manipulée. Ainsi, si l'utilisateur reconnaît une coïncidence entre l'orientation de la souris et celle de la fenêtre, il peut imprimer la rotation voulue se guidant par le feed-back haptique fourni par la souris plutôt que par le feed-back graphique de la fenêtre. Puisque la double-souris est sans borne pour la rotation (puisque sans fil), on n'a pas besoin d'effectuer des opérations de débrayage pour le remettre à une orientation neutre. Cela réduit le nombre de mouvements inutiles lorsqu'on imprime la rotation.

**La souris rotative symétrique.** Nous avons décidé de tester quel serait l'apport d'une coque symétrique pour la souris rotative. Pour cela, nous avons couvert la souris par un hémisphère en polystyrène creux. Malgré la taille plus importante de ce prototype (12 cm de diamètre et 8 cm de hauteur), il était facilement maniable et la coque cachait toute les pistes tactiles d'orientation que la double souris naturellement fournit. Ce prototype est ainsi peut-être plus approprié pour contrôler la rotation d'objets graphiques dotés d'une forme et d'une orientation différentes des siennes. Par ailleurs, puisque cette coque ne dispose pas de coins, il est moins facile de la saisir latéralement qu'il ne l'est pour la double-souris ordinaire asymétrique. La correspondance appliquée entre ce dispositif et la fenêtre est aussi 1:1. Puisque ce prototype est symétrique, l'utilisateur ne peut pas associer son orientation à celle de la fenêtre. Cette correspondance 1:1 permet néanmoins à l'utilisateur d'appliquer des gestes de rotation qui auront la même amplitude sur la fenêtre. Par exemple, si l'on souhaite imprimer une rotation de 30° à la fenêtre, on peut esquisser un geste de rotation de 30° sur la souris.

**Rotation de la molette de la souris.** La molette est un dispositif couplé à la souris ordinaire qui est très employé pour défiler des documents qui sont plus grands que la fenêtre d'affichage. Ce mécanisme permet de naviguer dans le document sans avoir recours à la barre de défilement, ce qui minimise les mouvements de la souris. Dans notre étude, nous l'avons associée à la rotation de la fenêtre. En tournant ce contrôle vers l'arrière, on tourne l'objet en sens horaire ; vers l'avant, en sens anti-horaire. La molette de la souris que nous avons utilisée ne dispose pas d'un mouvement continu mais à cliquet. Nous avons ainsi établi une correspondance entre les degrés de l'orientation du document et le cliquet de la molette. Un ratio plus élevé (par exemple, 10:1 degré/cliquet) permet une rotation assez rapide et exige moins de mouvements des doigts. Ce ratio ne nous a pas toutefois semblé suffisamment fin pour l'opération de rotation de la tâche étudiée. Une configuration 4:1 degré/cliquet s'est montrée comme étant le meilleur compromis lors de nos études pilotes. Cette prise en charge de la rotation par le doigt indicateur tire profit des mouvements courts et rapides dont ce membre est capable (Zhai et al., 1996).

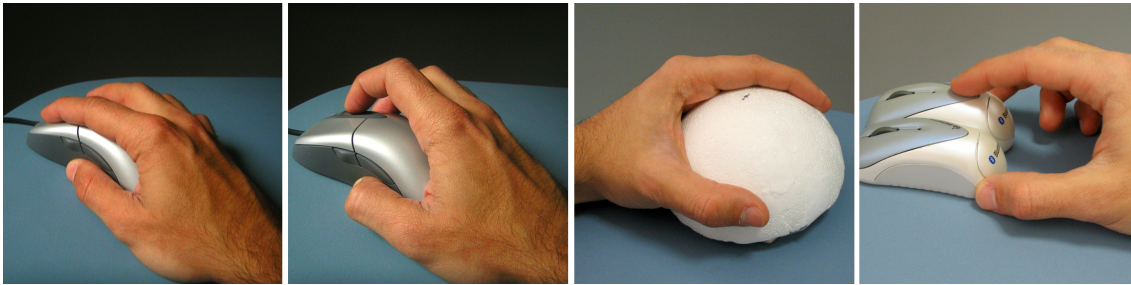


FIG. 3.3: Les quatre techniques d'interaction utilisées dans l'étude comparative.  
De gauche à droite : 2DM, 2DW, 3DR et 3DM.

**Rotation par réglage linéaire.** Le changement de mode est une façon de contrôler un paramètre d'une interface qui ne peut pas être pris en compte par défaut. Puisque les deux DDL de la souris sont déjà associés au mouvement de la fenêtre, le changement de « mode » permet d'associer un des deux transducteurs à la rotation de la fenêtre. Dans les logiciels de mixage sonore, par exemple, la rotation des potentiomètres graphiques est contrôlée de cette façon. Dans l'expérience décrite dans Jacob et Sibert (1992), cette technique a été aussi utilisée pour contrôler la translation le long d'un troisième axe. Pour notre étude, la rotation du document est contrôlée via une action composée : clic bouton suivi de déplacement gauche-droite de la souris avec le bouton toujours appuyé. Le déplacement vers la gauche tourne la boîte dans le sens horaire tandis que vers la droite la tourne dans le sens anti-horaire. Le déplacement le long de l'axe avant-arrière du bureau ne produit aucun effet. Pendant les études pilotes, nous avons observé que les participants pouvaient « oublier » que la fenêtre se trouvait en mode rotation (en ne relâchant pas le bouton). Nous avons ainsi introduit une sphère rouge sur la fenêtre lorsqu'elle est en mode rotation. Cela s'est avéré un feedback à la fois efficace et « non-envahissant ». La correspondance entre le déplacement gauche-droite et la rotation horaire ainsi que le ratio 1:2 degré/pixel ont également été choisis lors des études pilotes. Cette technique est la seule qui n'offre pas la possibilité (ni mécanique, ni logique) pour que la translation et la rotation de la fenêtre soient réalisées simultanément.

### 3.3.1 Hypothèses

Les quatre techniques citées ci-dessus étaient les *conditions expérimentales* de l'évaluation que nous avons conduite. Nous avons deux hypothèses principales :

**H1 : Contrôler trois DDL permet de réaliser la tâche plus vite qu'en contrôlant deux DDL.** Les manipulations basées sur la souris rotative devraient s'avérer plus performantes que celles des techniques basées sur la souris ordinaire (par molette et par réglage linéaire). Puisque le changement de la position et de l'orientation d'un objet sur un plan horizontal peut être perçu comme un attribut inté-

gral, la manipulation de ces trois DDL devrait tirer profit du fait qu'elle procure un contrôle simultané de ces trois variables. Les techniques à deux DDL procurent, quant à elles, un contrôle séparable de ces trois variables. Elles seraient, ainsi, moins bien adaptées à la représentation que l'utilisateur construit de la tâche. Un deuxième avantage des techniques à trois DDL est la correspondance directe qui existe entre la rotation de la fenêtre et celle du dispositif : le dispositif se déplace et tourne par rapport aux mêmes axes que la fenêtre. Les techniques basées sur une souris ordinaire offrent un contrôle de la rotation qui n'est pas intuitif, c'est-à-dire que l'on est obligé de suivre visuellement la rotation que son action imprime sur la fenêtre.

**H2 : La symétrie de la coque a un apport positif.** Le dispositif à trois DDL rectangulaire serait, à cause de son asymétrie, moins bien évalué par les participants que le dispositif rond. Le fait que la coque du dispositif rond ne « communique » aucune orientation, évite que l'orientation de ce dispositif se trouve en désaccord avec celle de la fenêtre. Je pense que cette incompatibilité entre le toucher et la vue pourrait gêner les participants pendant ces manipulations. Cette gêne, même si elle n'affecte pas la performance des opérations, pourrait être exprimée lors de l'évaluation subjective (comme réalisée par Hinckley et al., 1997). En ce qui concerne les différences de performance, en matière de durée de la tâche, entre les souris rotatives ronde et rectangulaire, nous n'avions pas d'hypothèse.

### 3.3.2 Méthode

Un grand nombre de choix de notre *design* expérimental se sont basés sur l'expérience rapportées par Hinckley et al. (1997) et par Balakrishnan et al. (1997).

#### Tâche

La tâche réalisée est une version simplifiée d'un bureau 3D avec une seule fenêtre. La fenêtre est contrainte au sol, déplaçable le long des axes de la profondeur (Z) et droite-gauche (X) et orientable autour de l'axe vertical (Y).

Les participants devraient réaliser une « tâche d'appariement<sup>6</sup> » 3D qui comprenait les actions de placer et d'orienter une boîte rectangulaire opaque (la « fenêtre ») à l'intérieur d'une boîte translucide (voir figure 3.4). La boîte opaque est jaune et un fac-similé en noire et blanc est plaqué sur l'une de ses faces larges. Toutes les deux boîtes sont contraintes au sol et ont la même hauteur. La boîte cible est 10% plus grande en largeur et 90% en profondeur que la boîte contrôlée. Le fait que la boîte cible soit plus grande que la boîte contrôlée fournit un seuil de tolérance à la tâche. Cette tolérance linéaire engendre par conséquent une tolérance angulaire de 6°.

---

<sup>6</sup>En anglais, « docking task ».

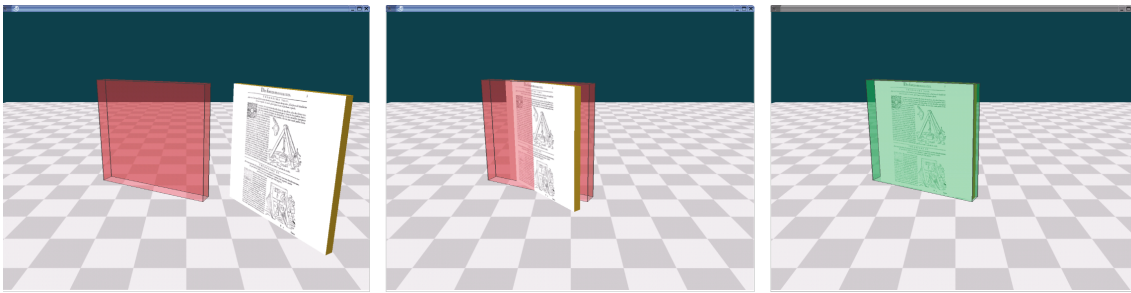


FIG. 3.4: Copies d'écran du logiciel d'évaluation de la tâche de manipulation trois DDL d'une fenêtre.

La scène dispose de plusieurs pistes visuelles en vue de faciliter la perception de la profondeur : le sol est revêtu par une texture en échiquier, les surfaces sont éclairées et la boîte cible est translucide. La boîte contrôlée doit être orientée de telle sorte que la face sur laquelle la texture est appliquée soit visible, c'est-à-dire *il n'y a qu'une orientation correcte pour la fenêtre*. La boîte cible translucide, initialement rouge, devient verte une fois que la fenêtre est correctement orientée et entièrement à l'intérieur de la boîte translucide.

La position et l'orientation de toutes les deux boîtes sont préalablement générées aléatoirement et enregistrées dans une liste statique. Ainsi, la configuration de chaque essai était identique pour tous les participants et sous toutes les quatre conditions testées. En moyenne, la rotation que les participants devraient imprimer sur la fenêtre a été de  $62^\circ$ , dont la plus courte était de  $4^\circ$  et la plus longue de  $176^\circ$ .

### *Design expérimental*

Nous avons utilisé un plan d'expériences en carré latin à mesures répétées (avec quatre séries de conditions expérimentales) afin de contrôler les effets d'apprentissage entre les participants. La variable dépendante étaient le durée pour achever la tâche. Les seize participants ont été tous soumis aux quatre conditions. Ils ont ainsi exécuté chacun 20 essais sous chaque condition. Les *stimuli* des 20 essais étaient toujours les mêmes pour les seize participants et dans les quatre conditions. Un *essai* consistait à *placer la boîte contrôlée (solide) à l'intérieur de la boîte cible (translucide)*. Une session, comportant les 4 blocs de 20 essais relatifs à chaque condition, durait en moyenne 50 minutes. Toutes les seize séries de session ont eu lieu durant une période de trois jours consécutifs. Afin d'éviter des effets d'apprentissage initial, nous avons décidé de suivre la méthode présentée par Hinckley et al. (1997) et de ne pas pas prendre en compte les cinq premiers essais de chaque condition (ils ont été quand même enregistrés).

En bref, il y a eu seize participants, quatre conditions expérimentales, quinze essais valides par condition :  $16 \times 4 \times 15 = 960$  *essais valides*.

### Les conditions expérimentales

Il y avait quatre conditions expérimentales d'évaluation. Elles correspondent aux quatre techniques d'interaction décrites plus haut. Elles sont surnommées 2DM, 2DW, 3DM et 3DR (voir figure 3.3). La manière de *déplacer* la boîte contrôlée était la même pour ces quatre techniques : une correspondance naturelle entre le déplacement de la boîte contrôlée sur le plan de l'échiquier et le déplacement de la souris sur la table. C'est-à-dire en déplaçant la souris vers la gauche, la boîte se déplace vers la gauche. La boîte contrôlée est toujours « attachée » à la souris et l'utilisateur n'a pas besoin de la sélectionner. Comme le curseur de la souris n'est pas utilisé (puisque'il n'y a pas de sélection à faire), il n'est pas affiché.

Le *clutching* du dispositif pendant la translation (lorsque le participant souhaite retourner la souris au centre du tapis, par exemple) est fait par la procédure standard. C'est-à-dire, on soulève la souris et on la remet sur sa nouvelle place. Cela est identique pour les quatre techniques. C'est en effet la façon dont la rotation est prise en charge qui varie d'une condition à l'autre (voir section 3.3).

Voici un récapitulatif des quatre conditions expérimentales et de la méthode de rotation associée (la figure 3.3 présente les dispositifs correspondants à chaque technique) :

- 3DM : la double-souris « asymétrique » sans coque ;
- 3DR : la double-souris « symétrique » habillée par une coque ronde ;
- 2DW : la souris normale ; la rotation de la boîte contrôlée est assurée par la manipulation de la molette ;
- 2DM : la souris normale ; la rotation de la boîte contrôlée est assurée par le déplacement de la souris le long de l'axe vertical pendant que l'on garde le bouton gauche appuyé.

### Participants

Seize volontaires non-payés ont accepté de participer à cette évaluation. Ils étaient onze hommes et cinq femmes, tous droitiers avec un âge moyen de 27,6 ans (dont le plus jeune avait 22 ans et le plus âgé 46 ans), la plupart étaient étudiants en informatique du CNAM Paris. Tous les participants étaient des utilisateurs de la souris et se servaient habituellement de la molette de la souris. Treize participants se sont déclarés comme des « utilisateurs avancés d'ordinateur » dont quatre comme des « joueurs avancés ».

### Apparatus

Pour les conditions 2DM et 2DW, le dispositif utilisé a été une souris optique filaire USB DELL dotée d'une molette. Les doubles-souris étaient construites à partir de deux mini-souris Bluetooth optiques (fabriquées par IOGear, modèle GME225B).

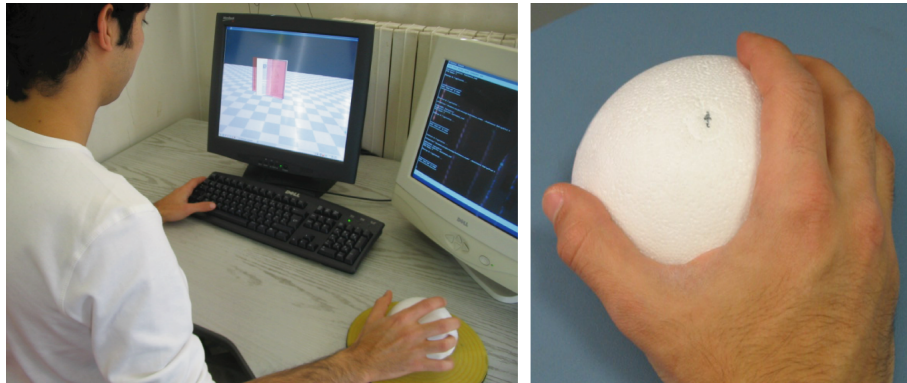


FIG. 3.5: A gauche, *Setup* expérimental et, à droite, la flèche indiquant l'orientation du capteur de la souris rotative ronde.

La double-souris utilisée dans la condition 3DR est recouverte par un hémisphère en polystyrène faisant 12 cm de diamètre. Le logiciel d'évaluation était programmé en C, *OpenGL*, *XGL* et *XInput*. L'application était affichée sur un écran de 19 pouces, avec une résolution de 1280x1024 pixels.

### Protocole

Les participants étaient assis à environ 60 cm de l'écran et ils ont manipulé le dispositif avec leur main droite. Ils ont tous reçu les mêmes consignes et, avant chaque condition, nous leur faisons une démonstration d'une minute sur le fonctionnement de la technique. Afin de donner toujours les mêmes informations, les instructions et les démonstrations ont été faites suivant un texte imprimé. De fait, nous avons présenté la technique associée à chaque condition comme si elle était complètement différente des techniques « concurrentes » pour ne pas expliciter les similarités qui existaient entre elles (notamment entre 2DM et 2DW et entre 3DM et 3DR). Ensuite, il y a eu une minute d'« échauffement » de la technique en question. Pendant ce temps, le participant pouvait manipuler librement la boîte contrôlée (la boîte cible n'était cependant pas encore visible).

Après cette introduction, les participants recevaient des instructions sur la manière de démarrer et finir un essai. Avant le début de chaque essai, le dispositif était positionné au centre du tapis de la souris. L'axe profondeur (qui va du tronc de l'utilisateur à l'écran) de la souris devrait être perpendiculaire à l'écran. Cet axe était indiqué sur la souris rotative ronde par une flèche (dessinée en haut de la coque comme le montre la photo droite de la figure 3.5). Les participants pouvaient laisser leur main sur le dispositif avant de démarrer l'essai. La commande pour déclencher le début de l'essai et pour l'arrêter était la touche `ESPACE` du clavier, sur laquelle les participants pouvaient appuyer en utilisant leur main gauche. Pour finaliser la tâche, la boîte contrôlée devrait se trouver dans la boîte cible transparente (condition indiquée par le changement de couleur de cette dernière) et, dans ce cas, le participant pouvait appuyer sur la touche `ESPACE`. S'il le



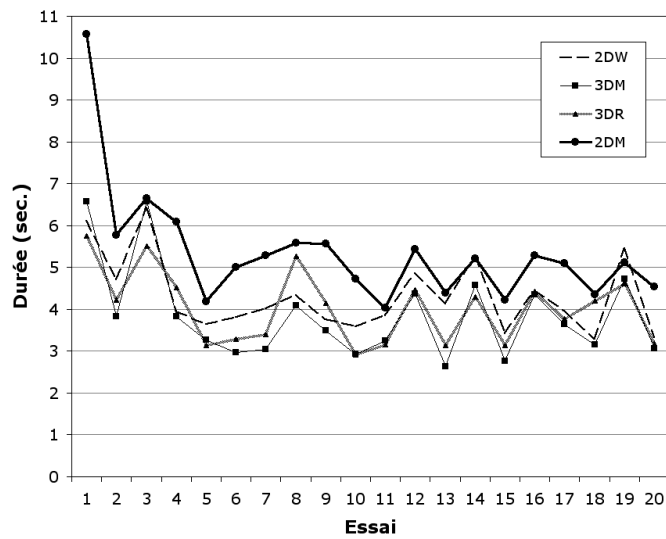


FIG. 3.6: Variation de la durée moyenne de chaque essai au fil des 20 essais.

faisait lorsque la boîte était encore rouge, cette action ne produisait aucun effet.

Suite à chaque série de 20 essais sous une condition, le participant faisait une pause de cinq minutes avant de recevoir les instructions relatives à la condition suivante. Après avoir fini les 80 essais, nous leur demandions de remplir un questionnaire.

**Exclusion du 18<sup>e</sup> essai.** Toutes les données relatives au 18<sup>e</sup> essai (toutes conditions et tous participants) ont été enlevées (suite aux tests mais avant l'analyse des données) car nous avons observé que, pendant son exécution, plusieurs participants ont eu des difficultés pour finaliser l'essai : ils positionnaient la fenêtre à l'intérieur de la boîte cible mais celle-ci ne devenait pas verte (feed-back indiquant que la tâche pourrait être terminée). Il nous semble que, dans cet essai, la configuration spatiale de la boîte cible ne permettait pas aux participants de voir sans ambiguïté si la boîte contrôlée était bien placée. Malgré le fait que la fenêtre semblait être entièrement à l'intérieur de la boîte cible, il est probable qu'une partie de sa face arrière dépassait la boîte cible.

### 3.3.3 Résultats

Comme prévu, uniquement les quinze derniers essais (sur vingt) de chaque condition ont été pris en compte dans notre analyse. Les données recueillies ont confirmé nos attentes : les durées des tâches à partir du sixième essai variaient de façon stable par rapport aux durées des essais précédents (voir figure 3.6).

Les durées moyennes pour achever la tâche dans chaque condition sont transcrites dans le tableau 3.1 et présentées dans la figure 3.7. En moyenne, **la technique**

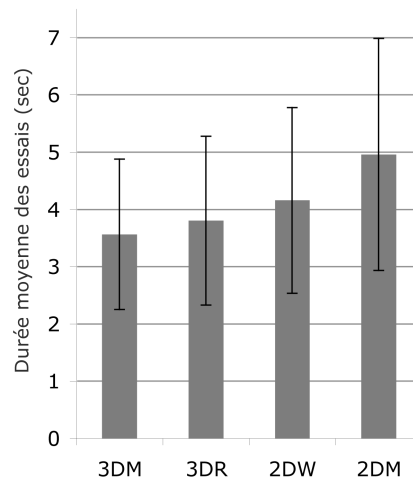


FIG. 3.7: Les durées moyennes des essais pour chaque technique d'interaction.

3DM était la plus rapide ; elle a été 6% plus rapide que 3DR, 14% plus rapide que 2DW et 28% plus rapide que 2DM. La technique 2DW a été 16% plus rapide que 2DM.

Les figures 3.8 et 3.9 présentent les histogrammes des durées de chaque essai par participant sous chaque technique comparée. Les distributions étaient toutes unimodales avec une asymétrie positive et une dispersion importante des valeurs. Parmi les essais les plus brefs (inférieurs à 5 secondes), les techniques rotatives (surtout 3DM) prédominent. Parmi les essais longs (5 secondes ou plus), les techniques basées sur la souris ordinaire prédominent (surtout 2DM).

Ces courbes sont complétées par les résultats du tableau 3.1. Il y a eu des différences individuelles importantes concernant la durée d'exécution. Le participant le plus rapide a pris 2,5 secondes en moyenne dans la condition 3DR. Le participant le moins rapide a pris 7,5 secondes en moyenne dans la condition 2DM. Les techniques basées sur la souris-rotative (3DM et 3DR) ont présenté des résultats supérieurs aux techniques basées sur la souris normale (2DW et 2DM) pour 70% des essais.

**Analyse statistique** Une analyse de variance a été effectuée, à l'aide du logiciel SAS 9.1, avec des mesures répétées ayant la durée de la tâche comme variable dépendante.

	2DM	2DW	3DM	3DR
Durée moyenne (sec.)	4,96	4,16	3,56	3,8
Ecart type (sec.)	2,02	1,62	1,31	1,47
Technique la plus rapide (% des essais)	8,4	21,3	37,7	32,4

TAB. 3.1: Résultats pour chaque condition : durée moyenne, écart type et le nombre de fois où chaque technique a été la plus rapide.

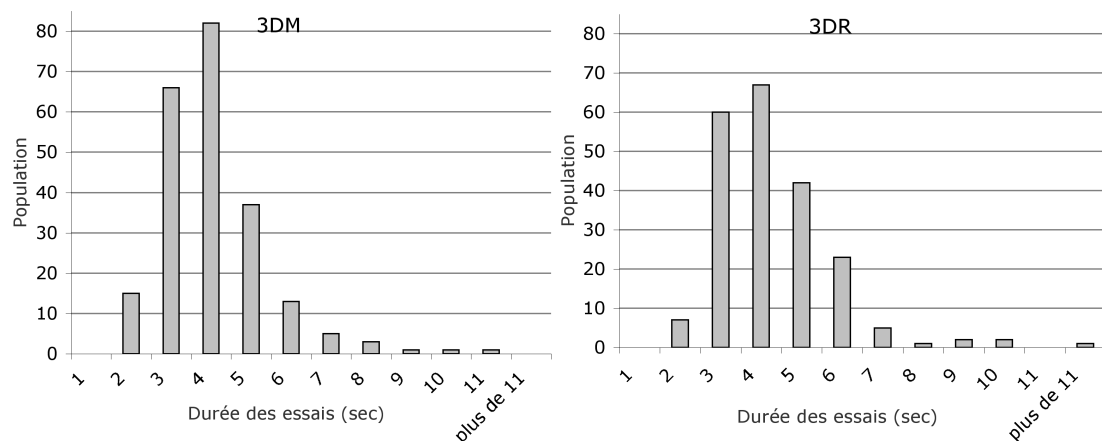


FIG. 3.8: Histogrammes des durées des essais pour les techniques 3DM (gauche) et 3DR (droite).

Globalement, les différences parmi les quatre techniques d'interaction se sont avérées hautement significatives pour la durée de la tâche ( $F_{(3, 878)} = 31,22 ; p < .0001$ ). Des comparaisons par paire ont été aussi réalisées et ajustées avec la méthode Tukey-Kramer. La différence entre les durées de 3DM et de 3DR s'est avérée non-significative ( $p = 0.4930$ ), les durées de 2DW et de 3DR présentent une différence significative ( $p = 0.0241$ ). Dans tous les autres quatre paires comparées, les différences ont été toutes hautement significatives ( $p < .0001$ ).

Ces résultats suggèrent que notre hypothèse H1 est correcte : dans la tâche en question, les techniques basées sur une souris rotative sont plus performantes, en matière de durée d'exécution, que les deux techniques basées sur une souris normale.

### Evaluation subjective

Suite aux 80 essais, les participants ont répondu à un questionnaire. Nous leur avons demandé de classer les quatre techniques d'interaction selon leur préférence dans une échelle d'intervalles allant de 1 à 5 (1 pour « la moins bonne » et 5 pour « la meilleure »).

La figure 3.10 présente une distribution des préférences des participants. Ces préférences sont bien distribuées parmi les quatre techniques avec un léger avantage pour les techniques basées sur la souris rotative. La technique 3DR a eu la meilleure moyenne (3,18) tandis que 2DW a eu la pire (2,68) ; les techniques 3DM et 2DM ont eu des moyennes similaires (2,93 e 3,0 respectivement). Ces résultats sont positifs pour les souris rotatives puisque les techniques basées sur ces dispositifs, inédits pour les participants, ont reçu une évaluation équivalente aux autres deux techniques basées sur la souris ordinaire (associées à des techniques avec lesquelles les participants étaient plus familiarisés).

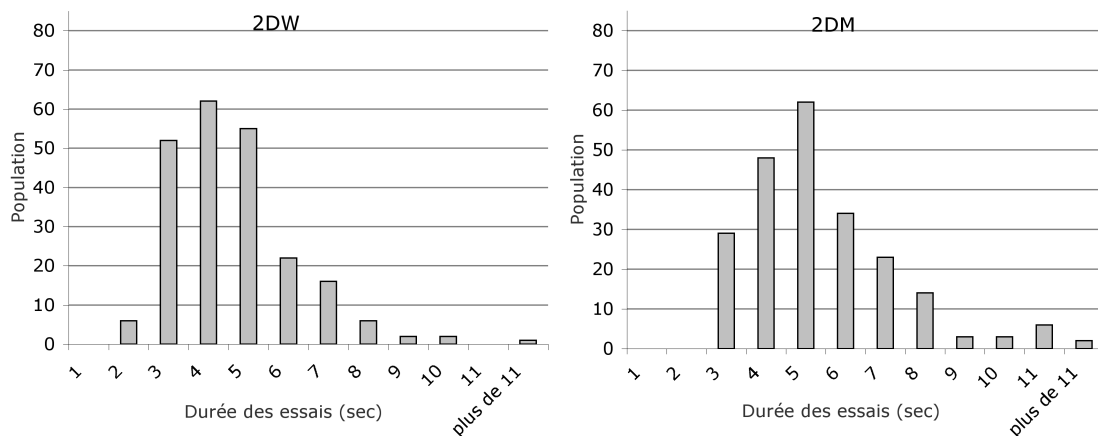


FIG. 3.9: Histogrammes des durées des essais pour les techniques 2DW (gauche) et 2DM (droite).

Concernant l'hypothèse **H2**, bien que la technique  $3DR$  ait été légèrement mieux évaluée que  $3DM$ , les résultats ne nous semblent pas assez conclusifs. Nous nous attendions à une réponse plus prononcée en faveur de la technique  $3DR$ . Cette question sera reprise dans la section suivante.

La figure 3.11 montre les résultats relatifs aux questions « Quelle technique a été la moins fatigante ? » et « Quelle technique a offert le plus grand contrôle sur l'objet virtuel ? ». Les réponses à la première question suggèrent « indirectement » que la technique  $2DW$  était la plus fatigante. Cela est peut-être dû au fait que la technique  $2DW$  nécessite beaucoup d'opérations courtes et répétitives (déroulement de la molette) lorsque la rotation est importante alors que les autres favorisent des gestes plus longs et continus. Les réponses à la deuxième question suggèrent qu'une majorité relative de participants (40%) a identifié la technique  $3DR$  comme celle qui offre la plus grande maîtrise sur la tâche en question. Cela peut être dû au fait que la forme du dispositif rotatif rond (utilisé dans la technique  $3DR$ ) ne présente pas une orientation identifiable. Ce résultat suggère que l'hypothèse **H2** ne devrait pas être complètement invalidée et pourrait être mieux explorée dans une prochaine étude.

### 3.3.4 Discussion

Dans cette section, les résultats obtenus sont discutés ; ils sont mis en relation avec les observations que nous avons faites pendant les sessions, avec les impressions que les participants ont écrites sur le questionnaire et avec la bibliographie du domaine.

**Manipulation naturelle.** Une partie des participants a considéré les techniques  $3DR$  et  $3DM$  comme étant des techniques plus naturelles. Je pense que ce « naturel » est due à la correspondance directe entre les trois DDL captés par le périphérique

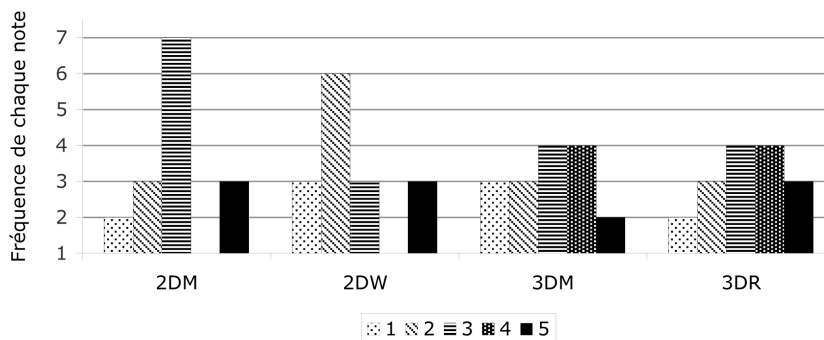


FIG. 3.10: La fréquence des notes (de 1 à 5 : 1 pour « la moins bonne » et 5 pour « la meilleure ») données par les participants à chaque technique d'interaction.

et l'objet virtuel. C'est-à-dire que la souris rotative se présente comme une instance physique de la fenêtre manipulée. Les mouvements qu'on lui imprime sont reproduits à l'identique par la fenêtre.

Lorsque les participants utilisaient les techniques 3DM et 3DR, ils ne se trompaient jamais sur le sens de la rotation de la souris qui correspondait au sens de la rotation qu'ils voulaient imprimer à la fenêtre. Des erreurs de ce genre se sont produites lorsque la technique utilisée était 2DM ou 2DW (souvent lors des premiers essais) : le participant démarrait son action et quelques millisecondes après il se rendait compte qu'il tournait la fenêtre dans le mauvais sens.

**Manipulation intégrale.** Nous avons observé qu'une grande partie des participants 3DM ou 3DR réalisaient des mouvements de rotation et de déplacement simultanés lorsqu'ils se servaient de la souris rotative. Cette simultanéité était observée au niveau de la manipulation gestuelle. Ces mouvements simultanés semblaient avoir lieu à deux moments de la tâche : (1) lors d'une opération de rotation principale (avec un seul ou avec plusieurs *pas angulaires*) le long du mouvement principal de déplacement et (2) au moment de faire le réglage angulaire final.<sup>7</sup> Ces derniers moments de manipulation simultanée étaient donc inévitables pour les techniques 3DR et 3DM car il était difficile de tourner le dispositif sans le bouger un peu (ce qui entraînait un déplacement de la fenêtre). Le participant devrait alors trouver le bon angle tout en corrigeant à la fois le déplacement accidentel que la rotation engendrerait mais aussi l'écart qui séparait la position de la boîte contrôlée de celle de la boîte cible. Cette interférence entre les trois DDL peut être vue soit comme un inconvénient soit comme un point fort.

Il nous a semblé que les mouvements réalisés dans les techniques 3DM et 3DR étaient, comparés aux techniques 2DM et 2DW, très efficaces. Ces dernières - basées sur la souris normale - exigeaient beaucoup d'opérations alternées de rotation et

<sup>7</sup>Les participants tournaient la souris en même temps qu'ils centraient (ou ré-centraient) la boîte contrôlée dans la boîte cible.

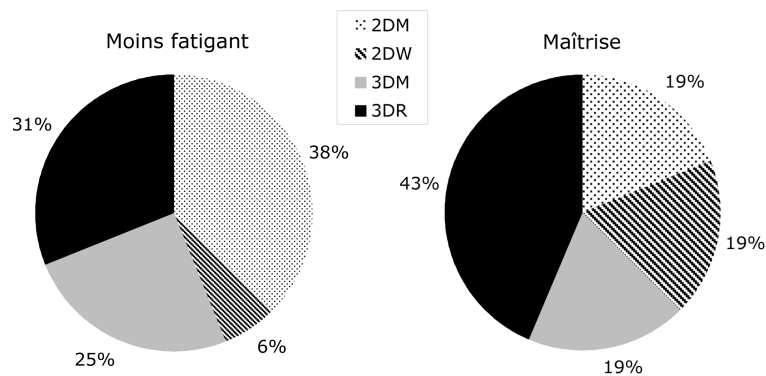


FIG. 3.11: A gauche, l'avis de participants sur la technique qu'ils ont considérée « la moins fatigante ». A droite, l'avis des participants sur la technique qui a procuré « la plus grande maîtrise » sur l'objet virtuel.

de translation. Malgré le fait que, dans la technique 2DW, la molette peut être utilisée en même temps que l'on déplace la souris, nous n'avons observé qu'un utilisateur réaliser ce geste.

Le fait que les réglages doivent être réalisés en deux étapes (déplacement et rotation), lorsqu'on manipule la souris traditionnelle, a été critiqué par certains participants. Le changement entre le mode rotation et le mode translation<sup>8</sup> semblait être fatigant dans la phase finale de la tâche, évoquée plus haut. A ce moment, la boîte contrôlée était près de la boîte cible et les participants réalisaient une série alternée de micro-déplacements et des micro-rotations jusqu'à ce que la boîte se trouve dans la bonne configuration. Cette alternance entre les modes semblaient particulièrement coûteuse pour la technique 2DM. De fait, dans cette technique, suite à un mouvement de GLISSER<sup>9</sup>, nous avons observé que, assez souvent, les participants remettaient la souris à une position plus confortable pour reprendre ensuite la translation de la fenêtre.

**Réglages fins.** Les techniques 3DM et 3DR ont présenté un compromis efficace entre vitesse et précision (dans le positionnement de la fenêtre). Les participants exécutaient des rotations rapides et, approchant l'orientation cible, ralentissaient naturellement le mouvement vers une rotation de haute précision.

La technique 2DM était probablement la plus rapide pour imprimer une rotation d'une grande amplitude. Cette technique a présenté néanmoins la durée moyenne la plus élevée. Trois aspects de cette technique sont peut-être à l'origine de ce problème : (1) elle n'est pas intuitive pour le réglage des petits angles ; (2) beaucoup de temps est pris pour le basculement entre le mode rotation et le mode translation (on peut dire que le le temps d'acquisition de chaque mode est élevé, c'est-à-dire

<sup>8</sup>Pour la technique 2DW, ce « changement » correspondait au changement entre la posture rotation et la posture déplacement.

<sup>9</sup>En l'anglais « drag ». Déplacer la souris en gardant le bouton appuyé.

le changement de posture de la main n'est pas une opération rapide) et (3) elle était la seule technique qui, pendant la rotation, provoquait un déplacement important de la souris sur le bureau. Cela engendrait de nombreux *clutchings* et, par conséquent, du temps dépensé pour le recentrage de la souris. Par ailleurs, il est arrivé à certains participants d'oublier dans quel mode ils étaient lorsqu'ils testaient la condition 2DM, c'est-à-dire ils essayaient de déplacer la fenêtre sans lâcher le bouton et ne comprenaient pas pourquoi cela ne marchait pas.

Concernant la technique 2DW, certains participants l'ont appréciée pour les rotations courtes car la manipulation « séparable » qu'elle procure leur permettait de régler l'orientation de la boîte gardant inaltérée la position de celle-ci. Nous ne sommes pas sûrs si cette technique étaient considérée comme intéressante lorsque les rotations étaient plus longues.

**Prise de la souris rotative asymétrique.** Nous avons questionné les participants à propos des différences entre la technique 3DM et la technique 3DR. Pour plusieurs participants elles étaient similaires (certains ont dit qu'elles étaient la même), mais que le dispositif de la technique 3DM était « moins adapté à la rotation ».

Pendant, le dispositif 3DM étant plus petit, les participants le manipulaient avec le bout des doigts. Les participants saisissaient le dispositif 3DM par sa latérale en pinçant soit son côté plus court, soit son côté plus long ou encore, moins souvent, les deux coins de la diagonale du rectangle.

Toutefois, à cause de sa forme asymétrique, cette « prise fine » ne suffisait pas pour procurer une rotation facile comme celle décrite par Hinckley et al. (1997). En revanche, l'asymétrie a peut-être joué un rôle positif grâce au fait qu'elle fournissait des indications tactiles qui pouvaient aider dans le réglage d'une rotation plus large.

Nous avons observé que l'asymétrie du dispositif incitait certains participants à réaliser un geste maladroit impliquant le poignet, le bras et l'épaule. Avant de saisir le dispositif, les participants semblaient tourner tout le bras dans les sens inverse de la rotation qui serait ensuite imprimée. Cela était peut-être fait en vue de pouvoir saisir la souris de telle sorte que tout le mouvement de rotation puisse être effectué par un seul geste.

Nous pensons que le dispositif 3DM, avec sa forme plus longue d'un côté et quatre angles droits, transmet l'idée que l'on ne peut pas le saisir dans n'importe quelle orientation - ce qui est vrai en partie, puisque le pincer du côté plus long est moins confortable que le faire du côté plus court. Cela encourageait peut-être les participants à minimiser le nombre de fois qu'ils devaient lâcher et reprendre ce dispositif. Quatre participants nous ont signalé à ce propos qu'ils ont considéré ce dispositif comme étant « difficile à tourner ».

Par ailleurs, nous avons aussi remarqué que certains participants réalisaient des manipulations plus maladroites lorsque l'orientation du dispositif 3DM présen-

tait une différence importante par rapport à celle de la fenêtre (c'est-à-dire lorsque l'orientation de départ de la fenêtre était différente de 0°, 90°, 180° ou 270°). Nous n'avons cependant pas eu de retour des participants sur cette question précise.

**Taille et prise de la souris rotative ronde.** Concernant la technique  $3DR$ , les participants manipulaient la souris ronde en disposant leur main sur la coque, créant ainsi une grande surface de contact entre la main et la souris. Toutefois, cette prise était abandonnée durant les moments de « rotation pure » ; ils se servaient alors des doigts pour essayer de pincer soit la base latérale de la souris, soit le haut. Contrairement aux mouvements larges réalisés avec la souris rotative asymétrique, les participants exécutaient plusieurs mouvements courts, rapides et itératifs. « Comme dévisser un bouchon d'une bouteille » - nous a dit un participant.

La taille du dispositif  $3DR$  s'est avérée un vrai problème : plusieurs participants l'ont trouvé trop large et leur doigts atteignaient à peine la latérale du dispositif lorsqu'ils essayaient de le prendre par le haut. Cette saisie par le haut ne procurait pas, dans le cas de certains participants, une prise suffisamment ferme pour tourner le dispositif. De plus la surface en polystyrène est peut-être un peu glissante lorsqu'on ne dispose pas d'une prise ferme. Par ailleurs, la hauteur du dispositif  $3DR$  (presque 5 cm plus haut qu'une souris) semblait gêner certains participants. Un participant a commenté que ce dispositif lui déplaisait car il devait « soulever trop la main » pour le manipuler tandis qu'un autre participant s'est plaint que la manipulation de la souris  $3DR$  lui avait fait mal aux épaules.

Nous avons constaté qu'il peut être problématique pour une souris rotative d'avoir une forme et des dimensions qui encouragent les utilisateurs à la saisir par une « prise forte » ou une « prise élargie »<sup>10</sup>, comme c'est le cas des prises dont on se sert pour saisir une souris ordinaire (sur les types de prise, voir le chapitre 2.1.4). Cette prise immobilise le dispositif contre la surface de la main et le mouvement de rotation sera alors assuré principalement par le poignet et par le bras. Cela engendre des gestes nuisibles aux articulations et dotés d'une amplitude plus réduite. Il nous semble qu'un dispositif rond mais plus petit, favorisant des prises par le bout des doigts, aurait contribué à des manipulations plus rapides et plus confortables.

### 3.4 Conclusion et travaux futurs

Les bureaux 3D ont besoin d'un dispositif d'interaction qui les rende faciles et intuitifs à utiliser comme la souris l'a fait pour les interface 2D *WIMP*. J'ai contribué à cette question via une approche précise : la conception et l'évaluation d'une technique d'interaction, basée sur une souris rotative, pour la manipulation na-

---

<sup>10</sup>En anglais, « *whole hand grasp* »



turelle de fenêtres dans un espace à trois DDL. Les résultats de cette évaluation ont été globalement très positifs et encourageants pour l'approfondissement des investigations.

Comparée, toutefois, à la performance des techniques basées sur une souris ordinaire, la performance des techniques à trois DDL ont présenté un gain de performance qui est un peu en-deçà de nos attentes. Nous pensons que des tâches plus longues et complexes (comme, par exemple, manipuler plusieurs fenêtres) ou avec une exigence de précision plus élevée pourraient nous fournir des résultats plus clairs. De même, une expérience où la difficulté des tâches varierait de façon croissante et plus régulière pourrait également fournir des pistes sur la manière dont chaque technique répond à ces variations. Par exemple, d'après nos observations et d'après quelques éléments des fichiers de *log*, nous avons l'impression que la souris ronde pourrait être mieux adaptée pour des contextes de micro-réglages précises alors que les réglages plus grossiers pourraient être bien réalisés par la souris à molette. En bref, plusieurs aspects pourront être approfondis dans les travaux à venir : corrélation entre performance de la technique et amplitude de l'angle, la durée des phases de simultanéité des opérations de translation et rotation et des explorations détaillées sur les facteurs de forme de la coque du dispositif rotatif. Le prototype devra aussi être amélioré même si, du point de vue de la captation, il semble avoir été à la hauteur des besoins de l'expérience.

Les aspects de forme évoqués dans la section précédente sont des questions ouvertes qui méritent d'être examinées et approfondies par des expériences dédiées uniquement à ces questions. Une compréhension élargie du niveau pragmatique (Buxton, 1983) et gestuel de la manipulation rotative peut contribuer à la conception de techniques plus efficaces. Par exemple, il serait intéressant d'enregistrer en vidéo les prises et les gestes employés par les mains lors de la manipulation de la souris rotative afin d'identifier les patrons les plus importants et d'essayer de concevoir des coques qui y soient compatibles. Par ailleurs, dans une prochaine expérience, une des variables observées (ou du moins contrôlées) devrait être la taille de la main des participants. Le ratio entre la taille de la main et le diamètre du dispositif semble être un facteur important lors de la manipulation de dispositifs rotatifs. Froehlich et al. (2006) et Øvergård et al. (2007) mentionnent, dans les comptes-rendus de leurs expériences, que la taille des mains des participants a été mesurée et notée, bien que ces mesures n'aient pas été utilisées ultérieurement lors de l'analyse des résultats.



# Manipulation quasi-intégrale de *widgets* rotatifs

*Ce chapitre traite de la manipulation de potentiomètres rotatifs dans les logiciels de son. Une technique, basée sur une souris rotative, est présentée. Elle sert à faciliter la manipulation de potentiomètres graphiques et à rendre plus fluide le passage d'un potentiomètre à l'autre. Des questions liées à l'utilisation de différents mappings pour la sélection d'une valeur dans des ensembles de dimensions importantes sont aussi discutées.*

Les activités de synthèse, édition et mixage sonore sont riches en paramètres qui exigent de nombreux réglages. Les professionnels et les amateurs qui travaillent dans ce domaine disposent souvent du matériel propre pour ce métier. En tant qu'utilisateurs de l'ordinateur, on peut les considérer comme étant des utilisateurs avancés puisqu'ils se servent de logiciels dotés d'interfaces et de fonctionnalités bien plus complexes que celles d'autres domaines. Dans le contexte de cette thèse, c'est la manipulation des potentiomètres rotatifs graphiques qui nous intéresse. L'objectif est de proposer une technique permettant d'utiliser ces *widgets* de façon plus efficace.

Or, les potentiomètres rotatifs et linéaires sont les composantes basiques du matériel d'un ingénieur du son. Nombreuses applications aujourd'hui sont consacrées à la création et à l'édition multimédia ; les boutons rotatifs sont souvent représentés graphiquement dans les interfaces de ces applications. C'est pourquoi cette étude commence par la présentation du contexte dans lequel les boutons rotatifs sont utilisés dans les métiers du son. J'ai consulté un ingénieur du son et, au cours d'un entretien, il a décrit et illustré les tâches qu'il réalise, les logiciels dont il se sert et les dispositifs qu'il utilise.

Il s'agit de la transformation d'un cadre de travail riche en dispositifs d'entrée,



FIG. 4.1: Cadre de travail de mixage et édition sonore. A gauche, un *setup* professionnel. A droite, une configuration plus « bureautique ». (A gauche, *operator* ! par Reinis Traidas et, à droite, *Traktor DJ setup* par Harald Walker.)

en pistes haptiques et en actions physiques dans un nouveau cadre plus restreint à la configuration souris-écran (figure 4.1, gauche). En réalité, le professionnel du son se dote souvent d'une configuration plus diversifiée en termes de dispositifs (surfaces de contrôle, molettes haut de gamme) et d'écrans que celle d'un utilisateur ordinaire (voir figure 4.1, droite).

Andersen (2003) distingue quatre types d'interface permettant de faire *DJ* et de mixer en temps réel : les systèmes analogues, les interfaces *WIMP*, les interfaces qui vont au-delà des métaphores *WIMP* et les interfaces graphiques qui se servent des métaphores des systèmes analogues. Comme mon point d'accroche dans la problématique du son est la manipulation des boutons rotatifs, c'est surtout ce dernier type d'interface auquel ce chapitre s'intéresse.

Il ne semble pas avoir eu d'autres études visant l'amélioration de l'interaction avec des boutons rotatifs graphiques. Des écrans tactiles qui simulent une surface de contrôle et permettent de régler les boutons directement avec les doigts sont une voie d'exploration intéressante. Dans la conception sonore, on retrouve des projets de réalité augmentée comme la *Reactable* ou comme *Audiopad* (Patten et al., 2002). Dans *Audiopad*, des morceaux sonores sont représentés par des pastilles disposées sur une table ; les paramètres de chaque morceau sont modifiés en tournant et en déplaçant les pastilles correspondantes.

Dans ce chapitre, l'étude d'une technique pour améliorer la manipulation de *widgets* rotatifs sera présentée. Ce chapitre établit d'abord un parallèle entre la façon dont on contrôle des potentiomètres réels dans des équipements sonores et celle dont on manipule des potentiomètres virtuels dans les applications de son. La technique proposée, basée sur une souris rotative, a trois objectifs. Premièrement, elle vise à procurer une correspondance naturelle entre le mouvement de manipulation de la souris et le retour graphique de cette action. Deuxièmement, elle vise à réduire le *temps d'activation* de chaque *widget* et, troisièmement, à permettre que les gestes entre chaque réglage soient plus fluides. Enfin, une exploration en cours est présentée : la manipulation rotative de *widgets* pour permettre

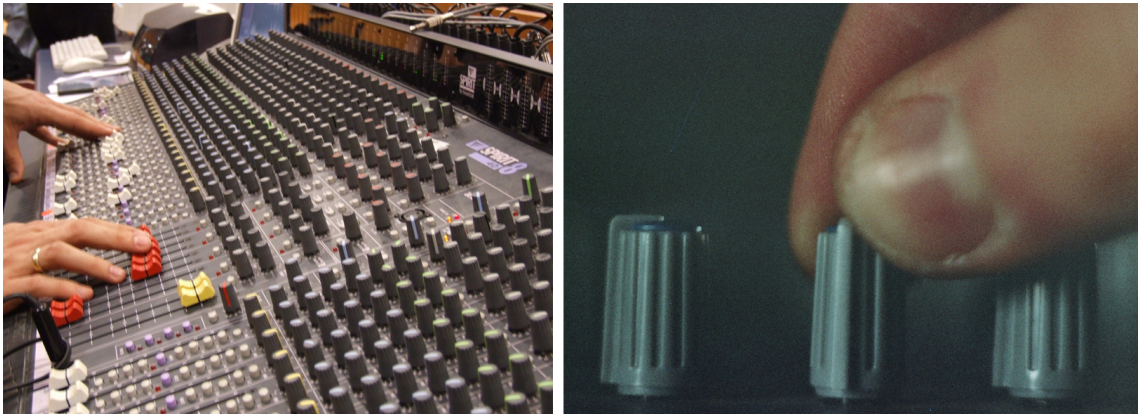


FIG. 4.2: Manipulation précise basée sur les doigts. (A gauche, *Mixing Desk* par Chris Simmons et, à droite, *Mixer* par Wiktor Gajewski)

une sélection de grande précision, c'est-à-dire le choix d'une valeur dans un ensemble de dimension importante.

## 4.1 Interaction réelle et avancée

Dans nombreux contextes, l'ingénieur du son effectue une succession d'opérations par minute, souvent se servant des deux mains. Ces caractéristiques émergent de façon plus marquée lors d'un mixage en temps réel. Dans ces circonstances, il doit passer rapidement d'un bouton à l'autre ; l'utilisation des deux mains devient indispensable pour réaliser deux, voire plusieurs, ajustements en même temps, ainsi que pour accélérer la transition d'un paramètre à un autre lorsque les réglages doivent être effectués l'un après l'autre.

Cependant, même dans un travail de post-production, où la vitesse des réglages n'est pas le facteur déterminant, le matériel d'interaction rend la tâche plus facile (par exemple, par la précision du réglage qu'il procure) et moins pénible (par le fait que les multiples paramètres de la tâche soient physiquement présents devant lui). En effet, le travail avec le son, que ce soit la synthèse, le mixage ou l'édition, comprend des centaines d'ajustements de paramètres continus. Ceci est une particularité qui différencie ce type de tâche de beaucoup d'autres tâches réalisées dans des contextes professionnels ou de loisir.

L'espace de représentation de la tâche (Norman, 1998) est beaucoup plus lisible dans le cas des interfaces physiques. De fait, le nombre de boutons rend visible et tangible la complexité de la tâche. Etant physiques, ces contrôles vont à la fois rappeler l'existence des paramètres et procurer un accès direct et instantané à ceux-ci. Le nombre de paramètres contrôlables est en effet très important mais, pour chaque paramètre, un contrôle physique correspondant est disponible.

Aspect	Bouton linéaire	Bouton rotatif
Paramètres par main	trois	un
Membre responsable	bras	doigts
Espace du mouvement	long	ponctuel
Position distinguable	oui	possible
Mouvement infini	impossible	possible
<i>Nulling problem</i>	oui	ça dépend

**TAB. 4.1:** Résumé des différences entre les caractéristiques de la manipulation d'un bouton rotatif et d'un bouton linéaire.

### 4.1.1 Boutons linéaires et boutons rotatifs

Parmi les dispositifs manipulés par les ingénieurs du son, c'est le bouton rotatif qui intéresse cette étude. En effet, il peut y avoir des dizaines de potentiomètres rotatifs et linéaires sur une table professionnelle. Quelques différences entre ces deux dispositifs sont évoquées car elles se manifestent aussi lorsque ces boutons sont « traduits graphiquement » comme des *widgets*.

Les boutons linéaires peuvent être déplacés simultanément par de différents doigts de la main (pourvu qu'on souhaite les ajuster de façon similaire). De plus, l'*identification de la position* d'un bouton linéaire est peut-être plus immédiate que l'*identification de l'orientation* d'un boutons rotatif. De fait, les boutons rotatifs peuvent être *distinguables* ou *non-distinguables* alors que les boutons linéaires sont toujours *distinguables* (voir chapitre 2.3.1).

Cependant, contrairement à un bouton rotatif, un bouton linéaire ne peut pas être glissé au-delà de la longueur de sa piste. De même, lorsqu'un bouton linéaire est associé à un paramètre logique, puis à un autre, il devra être re-initialisé physiquement. Ce besoin de re-initialisation est appelée « *nulling problem* »<sup>1</sup> (Buxton, 1983). Enfin, les boutons linéaires sont saisis par les doigts mais le membre responsable pour la manipulation est le bras. Le tableau 4.1 résume ces différences.

## 4.2 Réglages graphiques appauvris

Les logiciels d'édition et de mixage, comme *Cubase*<sup>2</sup>, ou de synthèse, comme *Reason*<sup>3</sup>, créent de nouvelles façons d'exécuter ces tâches ou de réaliser la création sonore. Ces suites de production numérique ont élargi le nombre de pratiquants de ces activités et remplacent en partie ou complètement le matériel analogique.

Des dispositifs entièrement numériques simulant les tables de mixages sont

<sup>1</sup>Concernant le *nulling problem*, voir le chapitre 2.3.1, paragraphe « Le périphérique d'entrée n'est pas qu'une entité logique ».

<sup>2</sup>S.M.T. *Cubase* : [http://www.steinberg.net/en/products/musicproduction/cubase4\\_product.html](http://www.steinberg.net/en/products/musicproduction/cubase4_product.html)

<sup>3</sup>*Propellerhead Reason* : <http://www.propellerheads.se/products/reason/>

maintenant d'usage courant et un large spectre de produits est disponible : de la surface de contrôle (où les comparateurs sont motorisés pour être repositionnables) à l'interface reposant exclusivement sur du matériel bureautique conventionnel. Néanmoins, de telles solutions sont, en général, chères et ciblées sur certaines applications. De plus, elles occupent un important espace sur le bureau et sont incapables de prendre en compte tous les paramètres de l'application. C'est l'interaction dans la configuration la plus simple (c'est-à-dire avec la souris) mais aussi la plus généralisable à d'autres contextes d'interaction, qui fait l'objet de l'étude de ce chapitre.

Une grande partie des interfaces graphiques de ces logiciels s'inspirent dans les équipements analogiques. Ceci peut aller de l'utilisation ponctuelle de certains potentiomètres graphiques jusqu'au mimétisme complet de l'interface et de l'apparence du matériel (comme c'est le cas de *Reason*). Ce type d'interface graphique basée sur une analogie de l'interface physique présente des avantages importants :

- l'espace fonctionnel minimal requis pour chaque bouton, spécialement les boutons rotatifs ;
- le fait que chaque bouton soit à la fois le contrôle du paramètre et l'indication de la valeur de ce paramètre ;
- la capacité de l'interface de rendre visible la configuration globale des paramètres en cours ;
- une distribution spatiale des contrôles qui a évolué et a été perfectionnée par l'industrie d'équipements sonores.

### 4.2.1 Multiplexage temporel des contrôles physiques

D'un point de vue visuel, la multitude de paramètres sonores accessibles a été conservée et transposée à l'écran (voir fig. 4.3), même s'il n'est pas sûr que tous les contrôles puissent être simultanément affichés sur l'espace de la fenêtre. Cependant, d'un point de vue des dispositifs physiques, les boutons rotatifs et les *sliders* ont été « réduits » à un seul périphérique physique : la souris. Il s'agit d'un appauvrissement de la dimension gestuelle : la manipulation des *widgets* est sérialisée et une partie importante du temps de la tâche est prise par la transition d'un contrôle à l'autre. La manipulation de potentiomètres simulés sur un ordinateur à l'aide d'une souris est une opération coûteuse en termes de temps et de ressources cognitives.

Les boutons rotatifs ne se trouvent pas uniquement dans les applications de son mais aussi dans des API de programmation visuelle (comme *Pure Data*), des logiciels de VJ, d'édition vidéo (comme *Apple Final Cut*) et de traitement graphique (comme le *widget* pour sélectionner l'orientation de l'ombre dans *Adobe Photoshop*). Des techniques courantes pour manipuler les potentiomètres graphiques avec une souris traditionnelle sont décrites dans la section suivante.

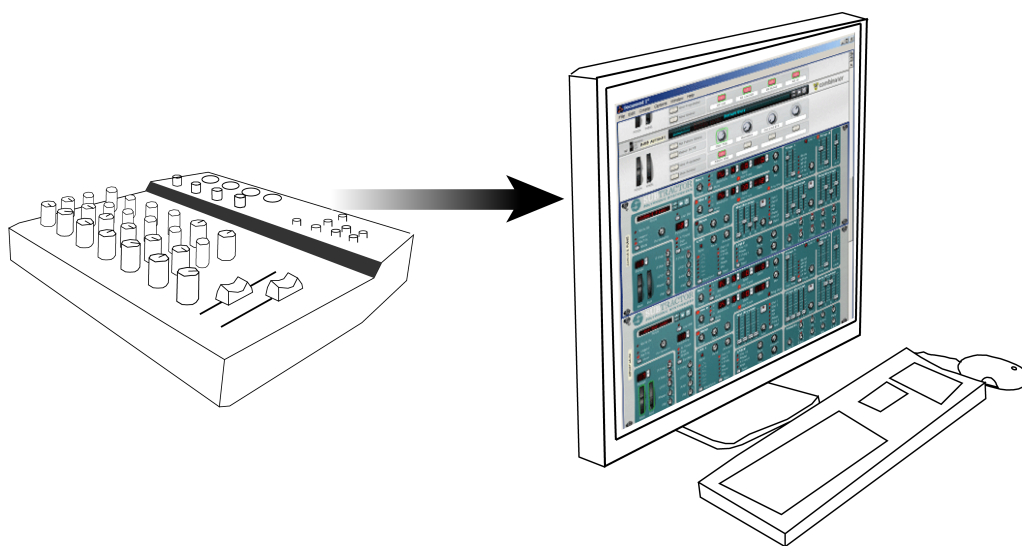


FIG. 4.3: Un exemple d'interface physique (société *Phonic*) et d'interface graphique (copie d'écran de *Propellerhead Reason*) pour le contrôle des paramètres sonores.

#### 4.2.2 Réglage linéaire

Le réglage de *widgets* rotatifs est souvent réalisé par des actions de clic sur le bouton ciblé suivi d'un mouvement linéaire de la souris. Cette opération peut être décomposée en quatre étapes :

1. on amène le curseur jusqu'au *widget* visé ;
2. on actionne le *widget* en cliquant sur le bouton de la souris ;
3. on attribue une nouvelle valeur au potentiomètre ;
4. on désactive le *widget* en relâchant le bouton de la souris.

La *première phase* est longuement étudiée dans les travaux concernant la *target acquisition* et ne sera pas traitée ici (voir, par exemple, Balakrishnan, 2004).

L'activation constitue la *deuxième phase*. Elle est décisive car elle fait le pont entre la sélection d'un *widget* et le réglage de celui-ci. Pendant que le *widget* est activé, le déplacement de la souris le long de l'axe Y le fait tourner.

Puis, dans la *troisième phase*, d'autres techniques (moins communes que le déplacement le long de Y) peuvent être utilisées pour le réglage de la valeur : tourner la molette (voir sous-section suivante), glisser le curseur dans un mouvement circulaire jusqu'à la nouvelle orientation ou cliquer sur la nouvelle orientation du curseur du bouton rotatif. Bien que ces deux dernières techniques soient plus cohérentes avec le principe de la manipulation directe, elles permettent toutefois une sélection moins précise que ne le permettent le réglage linéaire ou le réglage par molette. Cette technique principale, le réglage linéaire, se sert d'un *slider* vertical masqué. Elle offre ainsi un retour visuel incompatible avec les mouvements



de la souris. Un bouton rotatif graphique offre une « fausse *affordance*<sup>4</sup> » puisqu'il indique qu'il doit être tourné alors que cela est faux (Gaver, 1991). Il cache l'*affordance* « effective » ; celle-ci devrait indiquer qu'il peut être glissé verticalement comme un *slider*.

Ce réglage par déplacement vertical peut demander des opérations de *clutching* si l'utilisateur ne dispose pas de beaucoup d'espace sur son bureau. La souris peut aussi devoir être repositionnée à cause d'une contrainte de *pawing*<sup>5</sup>. Si le curseur est arrivé au haut ou au bas de l'écran et la valeur souhaitée n'a pas encore été atteinte, on va devoir lâcher le bouton de la souris et reconduire le curseur sur le *widget* rotatif pour reprendre le réglage.

Enfin, dans la *quatrième phase*, on désactive le *widget* en lâchant le bouton de la souris. A ce point, le curseur peut se trouver (verticalement) éloigné du *widget* et l'utilisateur doit le reconduire à la zone d'intérêt dans la fenêtre. Cette technique impose donc une *transition*<sup>6</sup> importante.

**Réglage par la molette.** Dans certains logiciels, il est possible de régler le potentiomètre en positionnant le curseur sur le *widget* et en tournant ensuite la molette de la souris. Cette méthode permet de sauter la *deuxième phase* décrite en haut et de ne pas éloigner le curseur de la zone d'intérêt dans la fenêtre. Cependant, on se sert de mouvements itératifs pour manipuler la molette, en incrémentant ou en décrémentant peu d'unités du paramètre associé au *widget* à chaque nouveau coup. Cette caractéristique fait d'elle un dispositif efficace pour sélectionner de valeurs voisins mais moins satisfaisant pour sélectionner une valeur dans l'autre extrême de la plage de valeurs. De plus, l'axe de rotation de la molette est arbitraire par rapport à celui du *widget* rotatif ; on ne dispose pas de repères indiquant la correspondance correcte entre le sens de la rotation de l'un et de l'autre.

---

<sup>4</sup>L'*affordance* est un concept de la psychologie perceptive introduit par Gibson (1979), puis appliqué au design par Norman (2002) et à l'IHM par Gaver (1991). Dans ce dernier contexte, l'*affordance* d'une interface concerne les actions dont on peut se servir pour utiliser cette interface. En réalité, lorsqu'on est en face aux éléments interactifs d'une application, ils nous invitent à les manipuler d'une façon donnée en fonction de leurs couleurs, de leurs tailles, de leurs textures, etc. Cette perception des actions qui sont possibles *précède la compréhension*. C'est la perception qui voit un dispositif comme étant « cliquable », « tournable », « déplaçable », etc. en fonction de l'*affordance* de cet objet.

<sup>5</sup>Le *pawing* est l'équivalent logique du *clutching* (Fitzmaurice et al., 2003). Tandis que dans le *clutching* on repositionne la souris car on est arrivé au bord de la table, dans le *pawing*, on doit repositionner le curseur lorsqu'on glisse quelque chose et le curseur arrive au bord de la fenêtre avant qu'on ait fini sa manipulation.

<sup>6</sup>La transition fait référence à la distance à parcourir par le curseur entre l'objet d'intérêt et le menu (Hascoët et al., 2006). J'emprunte ce terme pour indiquer la distance à parcourir par le curseur entre l'endroit où il se trouve à la fin du réglage d'un *widget* rotatif donné et la zone de la fenêtre - où sont les autres *widgets* associés à la tâche en question - à laquelle il doit retourner.

### 4.3 Une technique à trois degrés de liberté

Cette section introduit une technique pour le réglage des boutons rotatifs basée sur une souris rotative. L'interaction basée sur le réglage linéaire se sert de l'*activation temporelle* (Beaudouin-Lafon, 2000). La solution proposée ici se sert de l'*activation spatiale*. Dans ce modèle, le *widget* est activé et désactivé lorsque, respectivement, le curseur entre dans la zone d'activation et lorsqu'il la quitte. Cependant, le principe de ce type d'activation est que le curseur doit rester sur le *widget* tant que ce dernier est manipulé. Or, avec une souris standard, cela n'est pas faisable (sauf si on se sert de la molette) car elle ne dispose que des deux transducteurs utilisés pour déplacer le curseur.

Un prototype de souris rotative a été utilisé. Le transducteur qui capte la rotation est *dédié* au réglage des potentiomètres. Il est ainsi possible d'établir une correspondance naturelle entre la rotation de la souris et la rotation du potentiomètre graphique.

#### 4.3.1 Transition quasi-intégrale

La sélection d'un *widget* et la rotation de celui-ci sont des opérations séquentielles. Cependant, la transition entre la première et la deuxième opération peut devenir plus fluide. Si le geste de l'utilisateur guidant le curseur jusqu'au *widget* rotatif est le même geste qui commence à le tourner, le passage d'une opération à l'autre peut alors devenir plus naturel.

Comme nous l'avons vu dans la section 2.1.2, certaines tâches peuvent être exécutées plus naturellement à l'aide d'un dispositif qui permet de contrôler simultanément plusieurs degrés de liberté (DDL) (Jacob et al., 1994). C'est le cas des actions de déplacer et de changer la taille d'un objet graphique. D'autres tâches sont exécutées plus facilement lorsqu'on manipule les transducteurs les un après les autres de façon isolée (Buxton, 1986). Un exemple de cette deuxième situation : lors d'une édition vidéo, on doit trouver un cadre dans un morceau vidéo A et puis un point d'entrée pour ce morceau dans un autre morceau B. Un troisième type de tâche peut être celui dont la manipulation idéale conjugue la manipulation isolée des transducteurs lors d'une étape et la manipulation « groupée » lors d'une autre étape.

Les exigences de ce troisième type de tâche pourraient être posées de la façon suivante. L'utilisateur doit pouvoir utiliser tantôt un seul DDL tantôt les autres deux DDL sans qu'il soit obligé de faire ni un changement de mode explicite pour passer de l'un à l'autre ni relayer le réglage à un autre périphérique. Par ailleurs, dans les situations où la vitesse est un facteur plus important que la précision, on peut vouloir tourner la souris lorsque le curseur est en train de s'approcher ou de quitter la zone graphique du potentiomètre. Ces millisecondes de simultanéité pourraient rendre plus fluide la transition entre la sélection et le réglage d'un

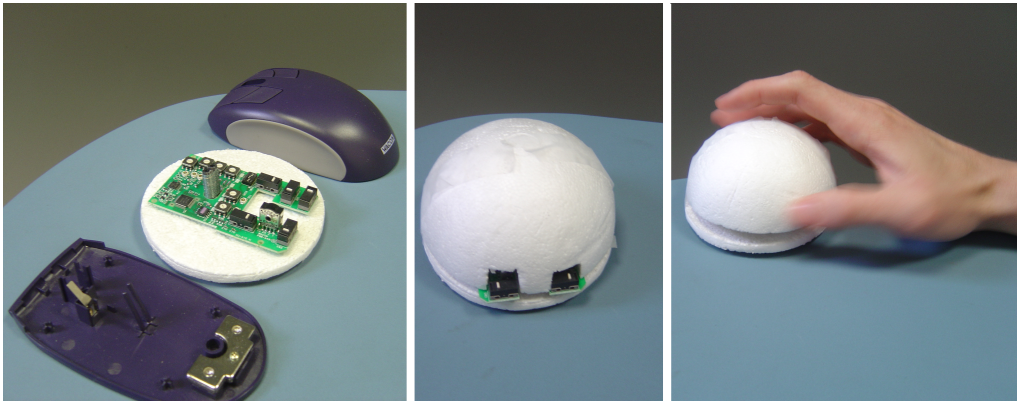


FIG. 4.4: Le dispositif *Wacom 4D Mouse* dans une coque ronde de polystyrène.

bouton.

Enfin, le fait que ni le curseur ni la souris ne bougent pendant le réglage réduit les *transitions*, minimisant les opérations de *pawing* et de *clutching* (expliquées dans la section 4.2.2). Cette étude propose ainsi d'améliorer les techniques actuelles en apportant une contribution sur trois aspects :

- la mise en place d'une correspondance plus naturelle entre le mouvement du dispositif et le retour visuel ;
- la réduction du temps d'activation pour la manipulation d'un *widget* rotatif ;
- et l'augmentation de la fluidité dans la transition entre les gestes qui composent l'activité.

### 4.3.2 Le prototype de la souris rotative

Comme nous l'avons vu dans la section 2.3.3, une souris rotative peut capter la rotation selon différentes techniques. J'ai utilisé le dispositif *4D Mouse*, qui fait partie du kit de la tablette *Wacom Intuos 4*. Il s'agit d'une souris sans fil à trois DDL (captation par technologie magnétique de la tablette).

Les points forts de ce dispositif sont : captation absolue, précision (0.2 degré) et stabilité. Ses principaux inconvénients sont ses dimensions (12.5cm par 7cm), le bruit dans le signal capté (ce qui impose soit un délai pour le filtrage soit une perte de précision) et l'asymétrie de sa forme. En effet, le fait que cette souris ait la forme d'une souris traditionnelle limite les orientations dans lesquelles on peut la saisir confortablement. Afin d'avoir un dispositif symétrique et plus petit, j'ai enlevé la carte de la souris de son conteneur et le recouvert d'un hémisphère de polystyrène de 10cm de diamètre (voir fig. 4.4). Un prototype d'interface avec des potentiomètres graphiques a été développé sous *Linux*, *OpenGL* et utilisant le pilote *Linux Wacom*.

### 4.3.3 Premiers essais

Une première configuration a été testée : le mouvement de la souris déplace le curseur ; lorsque ce dernier se trouve sur un *widget* rotatif, celui-ci sera tourné par la rotation de la souris. Cette rotation est relative et basée dans un *mapping* un-vers-un : lorsqu'on tourne la souris d'un degré, le *widget* rotatif est tourné d'un degré.

Sans imposer aucune contrainte au déplacement du curseur pendant la rotation, ce dernier quittera facilement la zone d'activation du potentiomètre et celui-ci sera désactivé. De fait, lorsqu'on tourne la souris, le curseur bouge - selon la dextérité de l'utilisateur et les facteurs de forme de la souris - et le *widget* finit par être désactivé<sup>7</sup>. Une autre variable qui entraîne cette dérive du curseur est la taille du dispositif manipulé.<sup>8</sup>

Cependant, même si on utilisait des prototypes très petits avec le capteur de rotation placé correctement dans son centre, le curseur pourrait continuer à s'écarter de la zone du potentiomètre. Cette tendance à s'écarter du *widget* est même accrue par la fonction d'accélération du curseur car les mouvements de rotation, bien qu'ils soient courts, sont assez accélérés. Par ailleurs, j'ai pu remarquer lors de premiers essais que l'utilisateur semble se concentrer afin de ne pas réaliser des mouvements brusques qui pourraient faire bouger le curseur de façon involontaire. Sa concentration est donc en partie consacrée à stabiliser le curseur graphique plutôt que dans le but de faire le réglage qui l'intéresse. Ainsi, il est nécessaire une technique qui minimise les sorties involontaires du curseur de la zone du potentiomètre.

La solution la plus directe pour ce problème serait d'essayer d'identifier si un geste de rotation est en cours et, à ce moment, d'immobiliser le curseur sur le *widget*. L'identification du mouvement de rotation doit cependant pouvoir prendre en compte les facteurs suivants :

- (1) Le curseur passe sur un *widget* qui n'est pas la cible finale et, en dépit de sa trajectoire apparemment droite, les paquets de données indiquent une rotation ;
- (2) l'utilisateur essaie de conduire le curseur hors de la zone du *widget* en même temps qu'il finit la rotation ;
- (3) le dernier paquet de données reçu n'indique pas correctement la tendance plus globale du mouvement.

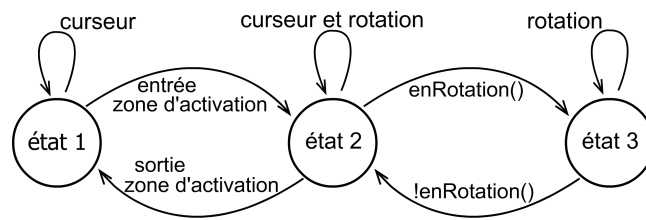


FIG. 4.5: Le modèle à trois états utilisé par la technique pour faciliter l'activation spatiale.

### 4.3.4 Un modèle à trois états

La figure 4.5 présente un modèle à trois états (Buxton, 1990) décrivant la solution adoptée pour gérer les cas cités ci-dessus.

Dans l'ÉTAT 1, la rotation du périphérique n'a aucun effet sur le système. L'ÉTAT 2 est ambigu ; la manipulation du dispositif contrôle le *widget* mais aussi le curseur.

Dans l'ÉTAT 3, la rotation est considérée comme étant « intentionnelle » et le curseur reste bloqué au centre du *widget* activé. Il y a deux conditions importantes pour accéder à l'ÉTAT 3 et pour y rester :

- les deux coordonnées de position doivent varier en-dessous d'un seuil  $a$  ;
- les valeurs du LACET doivent varier de façon monotone et au-dessus d'un seuil  $b$ .

Cette variation n'est pas indiquée par la différence entre les deux derniers échantillons mais issue d'un filtre passe bas appliqué aux  $n$  derniers échantillons.

L'ÉTAT 2 resout les points (1) et (2). C'est-à-dire que le curseur ne sera pas bloqué ni lorsqu'il passe sur un *widget*, ni lorsqu'on essaie de le faire sortir du *widget* actuel tout en finissant le geste de la rotation (c'est-à-dire lorsque LACET n'est pas encore nul).

Le filtre passe bas, utilisé par Oshita (2005) pour les filtrage des données d'inclinaison d'un stylo *Wacom*, permet, en lissant les variations entre les paquets, d'obtenir la tendance de la variation des données de LACET.

$$y_i = y_{i-1} + (1 - \alpha)(y_{i-1} - y_{i-2}) + \alpha(y_i - y_{i-1}) \quad (4.3.1)$$

Afin d'accroître le poids des données issues des paquets les plus récents et de rendre ainsi les transitions de l'ÉTAT 3 à l'ÉTAT 2 suffisamment réactives, la constante  $\alpha$  du filtre a été définie à une valeur élevée (autour de 0.8).

*Réduire la vitesse de rotation* (ce qui décroît aussi le risque de quitter involon-

<sup>7</sup>Plus le *widget* est petit, plus grande sera la probabilité que le curseur quitte la zone. En effet, les *widgets* rotatifs de logiciels sonores professionnels peuvent avoir une taille très réduite.

<sup>8</sup>Ayant manipulé d'autres prototypes rotatifs auparavant, j'ai pu constater que plus le rayon du périphérique est petit, plus la rotation sera précise et moins le dispositif oscillera en dehors de sa position de départ.

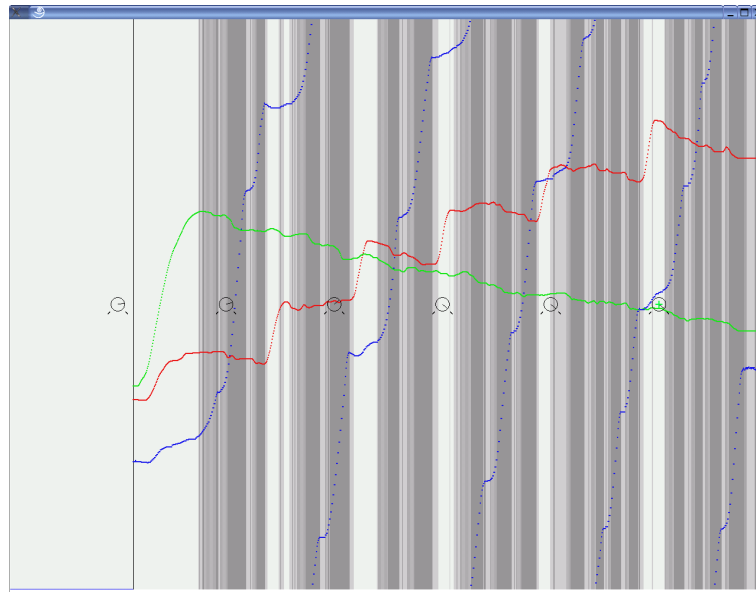


FIG. 4.6: La fenêtre de visualisation des données de chaque transducteur. Chaque tonalité du gris de fond indique un état différent : le plus clair correspond à l'ETAT 1, le moyen à l'ETAT 2 et le plus foncé à l'ETAT 3.

tairement la zone), *inverser le sens* de la rotation ou *déplacer la souris de façon monotone* sont les trois événements qui déclenchent la sortie de l'ETAT 3. Effectivement, à la fin de chaque *pas angulaire* (lorsque les doigts lâchent le périphérique et le re-saisissent avec une prise neutre afin de recommencer la rotation), l'ETAT 3 est abandonné car le mouvement ralentit jusqu'à ce qu'il s'arrête complètement (voir fig. 4.6).

### 4.3.5 Visualisation des entrées et mise au point

L'application du modèle à trois états décrit ci-dessus et sa mise au point (la définition des seuils) ont été faites grâce à une interface de visualisation des données d'entrée (voir fig. 4.6). Cette visualisation était utile pour comprendre la façon dont varient les données d'entrée pendant les moments de simultanéité entre les opérations de rotation et de déplacement.

Les  $x$ ,  $y$  et  $LACET$  sont représentés par des pixels rouge, vert et bleu. Chaque colonne de pixels représente un pas dans la boucle d'interaction et, à chaque pas, cette colonne est déplacé d'un pixel vers la gauche. La hauteur du pixel dans la colonne correspond à sa valeur (normalisée selon la dimension verticale de l'écran). Lorsque le système est à l'ETAT 2, la couleur de fond de la colonne de pixels devient gris clair ; à l'ETAT 3, cette couleur devient gris foncé.

Une interface, similaire à celle que nous venons de voir, destinée à la visualisation des données d'entrée et des états du dispositif d'interaction, a été proposée par Guimbretière et al. (2007). Elle a été conçue dans le but de comprendre de

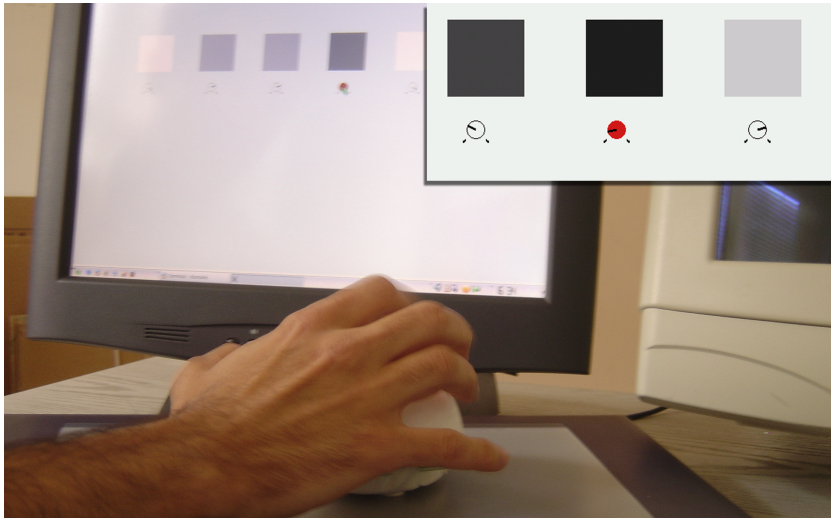


FIG. 4.7: *Widgets* rotatifs associés à des paramètres couleur d'un objet de la fenêtre

manière plus globale les caractéristiques de différentes techniques d'interaction utilisées pour la réalisation d'une même tâche.

#### 4.3.6 Retours de l'utilisation

J'ai développé une interface de test avec cinq boutons rotatifs (30 pixels de diamètre chacun). Chaque bouton est associé à un carré ; le bouton règle la transparence de la couleur affichée dans le carré (voir fig. 4.7). Cinq personnes (tous utilisateurs avancés de la souris mais n'ayant pas l'habitude avec d'autres périphériques d'entrée) ont été invitées à essayer l'interface. On leur a demandé de tourner chaque bouton jusqu'à ce que la couleur du carré devienne identique à la couleur de fond de la fenêtre. Les utilisateurs ont apprécié l'idée du périphérique d'entrée rotatif et ont trouvé que la technique fonctionnait conformément à leurs attentes. Un utilisateur a remarqué que le curseur semblait « coller » au bouton à un moment où il a essayé de quitter rapidement ce bouton. En effet, la souris *4D Mouse* présente un délai dans l'arrivée des données d'orientation de moins d'une seconde dû au filtrage qu'elles subissent. Le problème signalé par l'utilisateur semble être causé par le délai qui fait l'application estimer parfois que la souris est en train de tourner lorsqu'elle ne l'est plus.

### 4.4 Travaux en perspective : sélectionner avec une grande précision

Un deuxième type d'interface de sélection, basée sur la manipulation rotative et destinée à l'exécution de sélections précises, a été esquissée. Cette section

présente quelques principes de son fonctionnement. Dans la technique décrite ci-dessus, un *mapping* un-vers-un a été utilisé. Le but de ce *mapping* est de permettre à l'utilisateur d'associer facilement la distance angulaire qu'il voit sur l'écran à celle qu'il doit effectuer avec la souris. Lorsqu'il doit tourner la souris d'une amplitude importante (par exemple, 120°), sa mémoire musculaire pourra l'aider à réaliser plus rapidement une partie du mouvement. Toutefois, il est également important d'explorer un *mapping* « dissocié » du curseur angulaire car il peut permettre à l'utilisateur d'effectuer une sélection dans une plage de valeurs beaucoup plus grande.

#### 4.4.1 Espace moteur et espace graphique

Comme nous le montrent les travaux de Masui et al. (1995) et Appert et Fekete (2006), il est pénible et peu intuitif de sélectionner un item lorsque l'ensemble dispose de plus d'éléments qu'il y en a des pixels sur l'écran. Par exemple, dans la palette d'attributs du crayon de *Adobe Photoshop CS 8.0*, l'utilisateur contrôle le diamètre de l'épaisseur du crayon par un *slider* (d'environ 200 pixels de large). Ce diamètre peut varier entre 1 et 2 500. Le *slider* devrait être long de 2 500 pixels pour que toutes les valeurs d'épaisseur du crayon soient accessibles. La solution adoptée est d'utiliser une échelle non-linéaire : au début du *slider*, on avance une unité par pixel, ensuite on avance 10 unités par pixel puis, à la fin, 100 unités par pixel. Il en est ainsi car, pour chaque valeur de l'attribut, il faut au moins un pixel pour le « représenter » sur le *slider*. En pratique, sur le plan moteur, la sélection d'une valeur précise est encore plus difficile ; faire déplacer le curseur d'un pixel à l'autre est une opération qui demande beaucoup d'effort et de précision.

Masui et al. proposent une technique qui crée un espace secondaire, dans lequel le curseur de sélection avance de façon plus granulaire. Cette technique peut s'appliquer aux *sliders*, aux ascenseurs, ainsi qu'aux points de contrôle des figures géométriques dans les logiciels de CAO. Appert et Fekete introduisent une technique multi-échelle qui permet à l'utilisateur de contrôler le niveau de zoom de la sélection. Pour faire une sélection fine, l'utilisateur doit zoomer sur la partie de l'ensemble qui l'intéresse et, pour avancer rapidement, il doit faire un zoom en arrière, réduisant la taille graphique - et motrice - de l'interface. En résumant ces approches, ou bien on sépare l'espace moteur de l'espace graphique, ou bien on laisse l'utilisateur faire le réglage dans un espace à plusieurs échelles (Furnas et Bederson, 1995).

#### 4.4.2 Réglage rotatif de grande précision

La souris rotative, comme la molette, n'est pas associée à un paramètre graphique limité (comme, le nombre de pixels de la fenêtre). On peut ainsi réaliser des manipulations très longues sans que cela soit incompatible avec le feed-back



graphique.

Prenons comme exemple le cas de l'édition d'une vidéo. Dans une vidéo, il y a 24 cadres par seconde. Dans un film de 90 minutes, il y a donc 130 000 cadres. Il est souhaitable que l'éditeur puisse désigner de façon précise n'importe quel cadre parmi les 130 000 possibles. En utilisant cette grandeur dans la présente étude, une maquette d'une technique de sélection précise a été réalisée.

**Un-vers-un.** Premièrement, un *mapping* simple de  $1^\circ$  par cadre<sup>9</sup> a été établi. Ainsi on peut sélectionner n'importe quel cadre, mais si on souhaite aller du premier cadre au dernier, on devra tourner la souris 361 fois !

**Exponentiel.** Lors d'une deuxième configuration, un *mapping* non-linéaire s'est avéré bien plus commode pour sauter d'un bout à l'autre de la vidéo - plus la rotation est rapide, plus le nombre de cadres défilés sera grand. Toutefois, ce *mapping* variable rend la manipulation peu intuitive. Cela est peut-être dû au délai de la captation du prototype utilisé. En effet, ce délai peut induire en erreur l'utilisateur qui ne pourra pas bien comprendre la relation entre son action et le résultat affiché.

**Mapping absolu puis relatif.** Dans une troisième configuration - un *mapping* à deux échelles -, l'utilisateur réalise d'abord une *sélection haptique* et il effectue ensuite une *sélection itérative*. Dans la *sélection haptique*, l'orientation absolue de la souris est associée à l'ensemble des cadres vidéo. Ainsi, si la souris est orientée à  $0^\circ$ , le premier cadre est sélectionné, si elle est orientée à  $359,9^\circ$ , le dernier cadre est sélectionné et ainsi de suite. Dans la *sélection itérative*, c'est la rotation de la souris (la variation de l'orientation) qui sera associée à la désignation d'un cadre. Il s'agit donc d'un *mapping* relatif dont la rotation de  $1^\circ$  fait avancer ou reculer de 1 cadre.

L'utilisateur pourra ainsi faire une sélection grossière de la zone qui l'intéresse avec la *sélection haptique*. Il raffinerait ensuite son choix par la *sélection itérative*. Dans une situation idéale, deux tours de rotation suffisent pour sélectionner n'importe quel cadre. Cela est dû au fait que la *sélection haptique* réduit l'ensemble de départ en 360 fois ; la *sélection itérative* est donc réalisée dans un sous-ensemble de 361 cadres.

La *sélection haptique* est faite avec le bouton appuyé.<sup>10</sup> Lorsque le bouton de la souris est appuyé, la rotation de la souris est captée en « mode absolu » et, si le curseur se trouve sur le *widget* de sélection, l'orientation de ce *widget* sera immédiatement associée à l'orientation de la souris. Le mode absolu permet que l'utilisateur positionne la vidéo grossièrement dans la partie de la vidéo qui l'intéresse en regardant (ou en sentant par le toucher) uniquement la souris. Cela sera encore plus efficace si la souris dispose de pistes tactiles par lesquelles l'utilisateur pourrait reconnaître son orientation sans aucun effort. La *sélection itérative* est active

---

<sup>9</sup>Je me sers des termes « cadre » et « vidéo » de l'exemple ci-dessus juste à titre d'illustration. Dans ce prototype, ils est affiché simplement la valeur courante de la sélection dans un champ de texte.

<sup>10</sup>Je l'ai pu testée grâce à la modification de la souris (l'ajout d'un bouton sur la coque), réalisée lors de l'étude décrit dans le chapitre 5.

lorsque le bouton n'est pas enfoncé. En positionnant le curseur sur le *widget*, ce dernier avance la sélection cadre par cadre lorsqu'on tourne la souris. Ces deux manipulations pourraient être réalisées l'une après l'autre comme s'il s'agissait d'un geste composé.

## 4.5 Conclusion

Dans ce chapitre, quelques obstacles à la manipulation fluide de *widgets* rotatifs ont été identifiés. Une technique, visant à faciliter ce type d'interaction et basée sur une souris rotative, a été proposée. La tâche analysée est un cas particulier de manipulation séquentielle. Dans ce type de tâche, l'une des opérations se sert de DDL distincts de ceux nécessaires à l'autre opération. La performance de cette technique doit être encore évaluée par rapport aux techniques traditionnelles (« clic-glisser » ou molette). Idéalement, cela doit être réalisé avec un périphérique plus petit et capable de fournir des données avec une latence plus courte. Il serait également important de faire tester cette interface par des professionnels avancés en matière d'utilisation de boutons rotatifs (physiques et virtuels). Enfin, j'envisage de réaliser des explorations analogues dans d'autres contextes quasi-intégraux afin de tester la validité et l'extensibilité de cette approche.

Par ailleurs, cette technique pourrait être améliorée si elle détectait avec quelques millisecondes d'avance le *widget* de destination du curseur<sup>11</sup>. Ce *widget* pourrait ainsi être activé et associé à l'orientation de la souris avant même que le curseur ne l'atteigne. Enfin, une première maquette d'interface pour sélectionner une valeur avec une grande précision a été présentée. Cette exploration doit encore être développée dans plusieurs directions bien qu'un prototype disposant d'une captation plus fiable et précise soit également nécessaire pour cette étude.

---

<sup>11</sup>En se servant, par exemple, d'approches similaires à celle utilisée par Blanch et al. (2004) pour identifier des « distracteurs ».

# Une technique pour le changement d'outils à l'aide d'une palette rotative locale

*Ce chapitre introduit la problématique du changement d'outils dans les applications graphiques. La technique « palette satellitaire » est proposée afin de faciliter ces opérations : cette technique permet de passer d'un outil à l'autre en tournant la souris, laissant le curseur immobile pendant les opérations de changement. Un prototype d'une application de dessin utilisant cette palette est décrit et quelques limitations et variations de cette technique sont présentées.*

On peut facilement constater que, dans de nombreuses applications, les utilisateurs font des allers-retours fréquents entre les outils de la palette graphique et l'objet d'intérêt. Par exemple, en retouchant une image, on doit alterner en permanence entre le crayon, le pinceau et l'aérographe. Ce chapitre décrit la conception et le fonctionnement d'une technique d'interaction qui permet que les transitions entre outils soient réalisées grâce à la rotation d'une souris enrichie d'un capteur d'orientation.

Il s'agit d'une palette qui suit le curseur. Les outils se trouvent en « orbite » autour du curseur. Ainsi, par la rotation de la souris, on fait tourner toutes les icônes jusqu'à ce que celui que l'on souhaite activer se trouve sous le curseur. Pour faciliter la compréhension de la technique, je restreins les exemples au cadre des logiciels d'édition graphique tels qu'*Adobe Illustrator* ou *GIMP*.

Les bénéfices que nous envisageons d'obtenir par cette approche sont :

1. réduire les allers-retours et les clics consacrés à la sélection d'outils ;
2. conserver le curseur où il se trouvait avant le changement d'outil ;

3. amoindrir l'attention visuelle qui doit être consacrée à ces opérations de transition.

Dans un premier temps, la problématique de cette étude est présentée, à savoir la difficulté à réaliser les changements d'outils de façon fréquente dans nombreuses applications actuelles. Dans un deuxième temps, ce chapitre présente quelques techniques d'interaction « avancées », basées sur la souris traditionnelle ou sur d'autres dispositifs, qui visent à réduire le temps d'activation des outils. Ensuite, une technique d'interaction basée sur une souris rotative est proposée : la « palette satellitaire ». Le construction des prototypes du dispositif et de la palette, ainsi que les limites de la technique, sont rapportées. Enfin, ce chapitre évoque une exploration en cours : un « mode avancé » de cette technique, encourageant l'utilisation de raccourcis gestuels, ce qui permettrait à des utilisateurs expérimentés d'activer un outil sans regarder la palette graphique.

## 5.1 Problématique

Les palettes graphiques sont nombreuses dans la plupart des applications bureautiques destinées à la conception ou à l'édition de documents, qu'ils soient textuels (par exemple, dans *Word* ou *Excel*) ou graphiques (comme dans *Microsoft Powerpoint*, *Adobe Photoshop* ou *Flash*).

Les palettes sont des *meta-instruments* (Beaudouin-Lafon, 2000) qui peuvent contenir soit des instruments (crayon, gomme, champ de texte), soit d'autres meta-instruments (familles de police de caractères, épaisseur du trait, couleur du trait). Lorsque les palettes présentent des propriétés d'un instrument, elles sont appelées aussi « boîte de propriétés » ou « inspecteurs ».

Comme les menus, les palettes permettent aux utilisateurs de choisir un instrument de travail ou de configurer celui-ci en explorant visuellement les options disponibles (Bailly et al., 2007). De même, elles ont l'avantage de servir à la fois comme interface de sélection et de visualisation des propriétés en cours. Ainsi, en regardant les inspecteurs, l'utilisateur peut voir quelles sont les propriétés de l'outil actif ou de l'objet sélectionné dans le document. Les propriétés sont appliquées sur l'objet en cliquant directement dessus. De ce fait, les inspecteurs ne nécessitent pas une action de validation explicite et ne sont pas modaux (Beaudouin-Lafon, 2006). Puisqu'elles permettent d'agir presque directement et immédiatement sur le document, les palettes sont plus en accord avec les principes de la *manipulation directe* que d'autres interface, comme les boîtes de dialogue.

Grâce à la sélection directe et visuelle de l'outil, l'effort cognitif nécessaire pour que l'utilisateur se rappelle toutes les options possibles est minimisé et les erreurs de syntaxes peuvent aussi être évitées. Enfin, contrairement aux menus, les palettes graphiques sont rémanentes, c'est-à-dire qu'elles ne disparaissent pas après qu'on active un de ses outils.

Une fois sélectionné, l'outil reste activé jusqu'au moment où on sélectionne un autre outil de la même catégorie (en général, dans la même palette). Par exemple, si on sélectionne la couleur rouge, elle restera active si on change d'instrument de dessin mais elle sera désactivée si l'on sélectionne la couleur bleu.

### 5.1.1 Contextes d'utilisation des palettes graphiques

La distance à parcourir entre l'objet d'intérêt (ou la zone d'attention) et la palette graphique est appelée *transition* (Hascoët et al., 2006; Bailly et al., 2007). On distingue trois types principaux de transition :

- **objet-outil** : la distance entre l'objet sur lequel on travaille et l'outil de la palette ;
- **outil-objet** : une fois l'outil sélectionné, on retourne le curseur à la zone d'intérêt afin d'utiliser l'outil ;
- **outil-outil** : lorsqu'un objet est sélectionné et on a déjà appliqué un nouvel attribut à cet objet, on peut passer à un autre attribut sans retourner à l'objet.

Dans les éditeurs de figures vectorielles, comme *Adobe Illustrator* ou *CorelDraw*, lorsqu'on active un attribut, ce dernier est directement appliqué à l'objet graphique qui est sélectionné à ce moment. L'utilisateur peut alors se servir des stratégies qui lui permettent de gagner du temps, comme sélectionner plusieurs objets et changer tous les attributs en même temps (par exemple, grossir l'épaisseur de toutes les lignes sélectionnées). Dans ce cas, les *transitions outil-objet* ne surviennent que lorsqu'on souhaite sélectionner un nouvel instrument (par exemple, activer l'outil « insertion de texte »).

Toutefois, même dans les applications qui n'autorisent pas ces changements d'attribut en groupe, l'ordre et la fréquence des opérations que réalise l'utilisateur peuvent varier en fonction de la façon dont l'utilisateur « voit » la tâche. Mackay (2002) distingue deux « stratégies de tâche » dont peut se servir l'utilisateur :

- une stratégie « orientée-outil » : l'utilisateur sélectionne un outil et exécute toutes les opérations associées à cet outil avant de passer à un autre (par exemple, peindre en bleu toutes les parties d'un dessin qui sont prévues être de cette couleur) ;
- une stratégie « orientée-objet » : l'utilisateur se concentre sur un aspect donné de la tâche et il fait le tour de tous les outils nécessaires pour accomplir cet aspect-là (par exemple, dessiner le visage d'un bonhomme avec tous les outils et toutes les couleurs nécessaires).

Ces stratégies résultent de la combinaison de plusieurs facteurs, à savoir le type de tâche, le périphérique utilisé, l'expérience du participant et le modèle mental dont il se sert pour la tâche. Certaines techniques favoriseraient plutôt une stratégie que l'autre. Par exemple, le *Toolglass* (présenté plus loin) est une technique plus *orientée-objet* tandis que la palette graphique traditionnelle est plus *orientée-outil*.

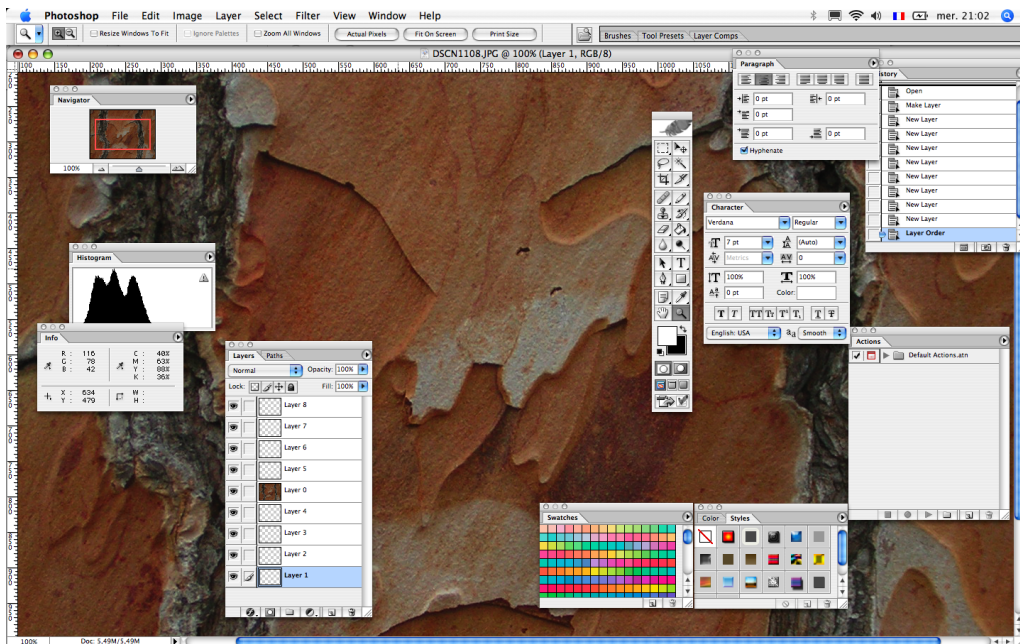


FIG. 5.1: Exemple de plusieurs palettes ouvertes dans *Photoshop* (sur un écran 1400 par 900 pixels).

### 5.1.2 Inconvénients des palettes graphiques

Quels sont les facteurs qui nuisent aux opérations de changement d'outils ? On exécute ces opérations par des mouvements qui requièrent de la concentration visuelle et motrice ; cette concentration sera soustraite de la tâche principale. De plus, le déplacement de la souris peut être large, ce qui demande de l'espace libre sur le bureau. Si cet espace pour le déplacement de la souris est réduit, les opérations de *clutching* seront très fréquentes, ce qui rendra la tâche pénible et peut ralentir son exécution. D'autre part, quelques changements dans le contexte matériel et logiciel qui ont eu lieu au cours des dernières années peuvent être cités :

1. La définition des périphériques d'affichage est de plus en plus élevée. La taille croissante des écrans et l'augmentation du nombre de pixels permettent l'augmentation du nombre de fenêtres simultanément affichées, de la taille des fenêtres et de l'espacement entre les objets, les menus et les palettes graphiques. On peut déplacer le curseur d'un extrême de la fenêtre à l'autre plusieurs fois par minute. Cet espace de pixels, devenu plus grand, rend les *transitions* plus difficiles.
2. Les applications graphiques sont dotées d'un nombre croissant d'outils. Par ailleurs, la complexification des applications joue un rôle dans la multiplication du nombre de fonctionnalités qui prennent la forme de commandes disposées dans des menus et palettes (Beaudouin-Lafon, 2000). Des versions récentes de logiciels tels qu'*Adobe Illustrator* peuvent présenter plus d'une dizaine de palettes. Lorsqu'elles sont toutes affichées en même temps, elles

couvrent une partie importante de la fenêtre (espace dont le but principal est d'afficher l'objet d'intérêt ; voir, comme exemple, la figure 5.1). Chez les professionnels de la création qui se servent des applications graphiques il est commun aujourd'hui de travailler sur plus d'un écran, l'un étant consacré uniquement à l'affichage du document tandis que l'autre affiche toutes les palettes en cours d'utilisation. Le fait de consacrer tout un écran au document et d'isoler les outils à une périphérie lointaine et discontinue augmente aussi le « coût » des *transitions*.

3. L'utilisation d'écrans tactile avec stylet devient plus commune. Ces périphériques rendent les opérations de sélection de cibles distantes plus pénibles car l'utilisateur doit déplacer sa main jusqu'à la cible visée. En effet, les périphériques indirects (tel que la souris ou le *trackball*) se servent de fonctions de transfert permettant au curseur de couvrir des grandes zones de pixels en déplaçant le périphérique d'un mouvement court mais accéléré. Ce n'est pas le cas des périphériques directs (tel que le stylo en question). Nulle fonction de transfert optimisante n'est possible pour ce type de périphérique car on sélectionne une cible en la touchant directement avec la pointe du stylo.

Le cadre de travail traité ici peut également être comparé à son activité analogue réelle, c'est-à-dire l'action de dessiner avec des outils réels sur une feuille de papier. Lorsqu'on travaille sur le papier avec des outils d'écriture et de dessin (crayon, gomme, stylo, règle), on place certes le papier sur une position centrale privilégiée tandis qu'on relègue les outils aux bords. Si on veut utiliser la gomme, on peut la saisir avec la main libre ou occupée (après avoir lâché le crayon) sans troubler sa concentration visuelle sur la zone principale du travail. Une certaine autonomie des mains par rapport au canal visuel (grâce aux informations kinaesthésiques) procure un soulagement de ce canal. D'autre part, la coopération entre les deux mains (lorsque la main gauche apporte l'outil jusqu'à la main droite) minimise les déplacements de la main dominante (Guiard, 1987).

## 5.2 Approches connexes

Plusieurs techniques d'interaction ont été proposées en vue de minimiser les transitions entre la zone de travail et les outils dans une palette mais aussi entre ces objets et les commandes dans un menu. Bien que cette étude porte sur le premier cas de figure, quelques techniques de sélection de menu proposent des solutions qui pourraient être appliquées aux cas des palettes graphiques.

### 5.2.1 Survol des concepts

Une distinction entre les deux types d'espace où on exécute ces mouvements sera utile avant d'analyser les techniques de sélection avancées :

**espace du document** - il s'agit de l'espace comprenant les actions qui interviennent directement dans la tâche principale. Ainsi, dans une application WYSIWYG, comme *Microsoft Word*, cet espace comprend toutes les opérations qui modifient le « papier » du document : la sélection d'un mot pour changer son style est une action dans l'espace du document.

**espace de paramétrage** - il englobe les mouvements exécutés en vue de modifier les paramètres des outils utilisés. Lorsque le curseur se trouve sur le texte et l'on déplace vers l'inspecteur de styles, il ne s'agit pas d'une *action directe* sur le document mais de la modification des propriétés du cadre de travail. Comme nous allons voir, il n'est pas obligatoire que cet espace obéisse aux mêmes lois ou qu'il se superpose à l'espace du document.

Lorsqu'on finit un trait et on dirige ensuite le curseur vers la palette de styles de trait, il s'agit d'une transition entre l'espace du document et l'espace de paramétrage. Cette distinction nous permet d'identifier, dans la vaste littérature du domaine, quelques stratégies principales :

**espace de paramétrage tangible** : cette approche est issue de l'interaction tangible et chaque outil logique correspond à un, et à un seul, périphérique d'entrée (Fitzmaurice, 1996). Lorsqu'il souhaite changer d'outil, l'utilisateur dépose le périphérique qu'il avait dans les mains, et il saisit celui qui est associé au nouvel outil. Dans ce cadre, l'espace de paramétrage est tangible et, pour y intervenir, on peut se passer de tout retour graphique et se reposer sur les retours haptique et kinesthésique (comme lorsqu'on change d'outil de dessin dans une situation réelle).

**espace de paramétrage doté de coordonnées relatives** : une deuxième approche consiste à repositionner dynamiquement la palette en fonction de la position du curseur. Comme l'espace de paramétrage demande habituellement beaucoup moins de pixels que celui du document, ce premier peut être conservé proche du curseur sans engendrer des problèmes d'occlusion trop importants. Dans la technique *Pie Cursor*, un disque contenant un certain nombre d'outils est déplacé avec le curseur (Fitzmaurice et al., 2008). Il s'agit d'une mini-palette circulaire « sur place » dont les options sont disposées en secteurs. Le curseur standard est en réalité invisible et se déplace à l'intérieur du *Pie* (un petit cercle de 32 pixels de diamètre). On active un outil en déplaçant ce curseur sur le secteur qui lui est attribué. Le curseur invisible ne sort jamais du cercle ; lorsqu'il arrive au bord du cercle, il déplace ce dernier avec lui. Les transitions d'un outil à l'autre sont réalisées dans ce micro-espace.

**capteurs supplémentaires** : on peut utiliser des capteurs supplémentaires complètement dédiés aux opérations associées à l'espace de paramétrage. Plusieurs configurations sont envisageables : **(1)** Il est possible d'effectuer un multiplexage temporel sur la captation ordinaire (en appuyant sur une touche qui change le mode) et d'utiliser le même périphérique pour naviguer uniquement dans l'espace de paramétrage. Dans la technique proposée par Has-



coët et al. (2006), l'utilisateur appuie sur la touche `SHIFT` et, lorsqu'il déplace la souris, ce mouvement lui permet de naviguer parmi les options d'un *Pie Menu* qui s'affiche. A la fin de la commande, le curseur se trouvera au même endroit où il était avant qu'il ne commence la sélection. (2) Deuxièmement, le capteur supplémentaire peut se trouver sur un périphérique auxiliaire. Dans ce cas, la manipulation est assurée par la main non-dominante. (3) La captation supplémentaire est placée sur le périphérique traditionnel et celui-ci peut être contrôlé, par rapport à la captation principale, tantôt de façon « séparable », tantôt de façon « intégrale ». Toutes ces diverses approches ont en commun le fait qu'elles laissent le curseur immobile pendant qu'on sélectionne un autre outil, ce qui réduit le coût des *transitions*. La section suivante présente plus en détail quelques techniques qui se servent des approches (2) et (3).

### 5.2.2 Sélection via un capteur supplémentaire

Je présente ici quelques techniques qui, via l'utilisation de capteurs supplémentaires, séparent l'espace moteur de sélection d'un outil de l'espace moteur du document : Nous pouvons noter trois avantages de cette séparation :

1. les outils peuvent être ramenés jusqu'au curseur, permettant à ce dernier de rester à l'endroit précis du document où l'on intervenait alors ;
2. les interventions sur ces deux espaces peuvent être réalisées en parallèle ;
3. l'utilisateur peut spécialiser ses mouvements sur chaque espace.

Il y a plus d'une manière d'activer un outil ou une propriété. Dans l'*activation temporelle*, l'outil devient actif par une opération explicite (un clic sur l'icône de l'outil) et il reste ainsi jusqu'au moment où on sélectionne un autre outil (Beaudouin-Lafon, 2000). C'est la méthode la plus courante. Par ailleurs, dans l'*activation spatiale*, il y a une zone de la fenêtre qui est l'espace d'activation de la fonctionnalité ; le curseur prend le rôle de cet outil tant qu'il y reste (par exemple, dans une page Web, le curseur se transforme de flèche en barre de sélection de texte lorsqu'il entre dans un champ de saisie). D'autre part, dans l'*activation spatiale relative*, la zone d'activation peut soit être déplacée, soit suivre le curseur (en restant toujours près de celui-ci).

#### Des techniques d'interaction avancées

Ci-dessous, quelques techniques *post-wimp* illustrent les propriétés que nous venons de voir :

**Manipulation bi-manuelle.** Dans le *Toolglass* (une palette flottante translucide), les outils sont sélectionnés par l'activation spatiale relative (Bier et al., 1993). On

déplace la palette sur la zone du document où l'on souhaite utiliser un des outils et on clique avec le curseur directement sur son icône. L'utilisateur peut déplacer ces palettes avec la main gauche par un périphérique à deux degrés de liberté (DDL), ce qui procure un parallélisme et une spécialisation des mouvements de la part de l'utilisateur.<sup>1</sup> La réduction des allers-retours est favorisée au détriment de la persistance du dernier outil utilisé.

**Captation de la pression du stylet.** Le *Pressure Widget* est une technique qui exploite des données relatives à la pression que certains stylets peuvent capter (Ramos et al., 2004). Le menu n'apparaît qu'au moment où l'on touche la tablette avec la pointe du stylet. On navigue de façon linéaire parmi les options grâce à la modulation de la pression appliquée. Un inconvénient de ce menu est qu'il se sert d'une méthode arbitraire pour valider la sélection (la méthode standard de clic n'est pas disponible puisqu'elle aussi est basée sur la pression). D'autres inconvénients sont le nombre réduit d'options qu'il peut supporter et le fait que le curseur saute entre ces options. L'espace de paramétrage n'a que sept niveaux et ne possède qu'un DDL. Cet espace est perpendiculaire au plan de l'espace du document.

**Captation de l'inclinaison.** Le *Tilt Menu* est une technique qui se sert de la captation de l'inclinaison du stylet pour permettre la sélection d'une commande dans un menu circulaire (Tian et al., 2008). Ce menu porte la forme d'un anneau avec plusieurs secteurs où chaque secteur représente une commande. L'inclinaison du stylet est représentée par un « curseur d'inclinaison » - un trait partant du centre du menu correspondant à la projection du stylet sur le plan de la tablette (Tian et al., 2007). Ainsi, en changeant l'inclinaison du stylet et la position de son autre extrémité, on fait varier la longueur du trait et sa direction. Lorsque le trait entre sur un des secteurs du menu, la commande correspondante à ce secteur est sélectionnée.

### 5.3 La palette satellitaire

La palette « satellitaire » est une technique d'interaction formée par deux composantes : l'une physique - la souris rotative - et l'autre logique - l'anneau tournant qui s'affiche sur la fenêtre.

L'activation est *angulaire* : on sélectionne l'outil en tournant le périphérique d'entrée. L'*activation angulaire* possède à la fois les avantages de l'activation temporelle - comme la persistance temporelle de l'outil sélectionné - et de l'activation spatiale - comme l'association d'une région de l'espace de travail à un outil.

---

<sup>1</sup>Par exemple, la main droite réalise les actions principales de dessin tandis que la main gauche facilite le travail de dessin en rapprochant les outils nécessaires à ces actions.

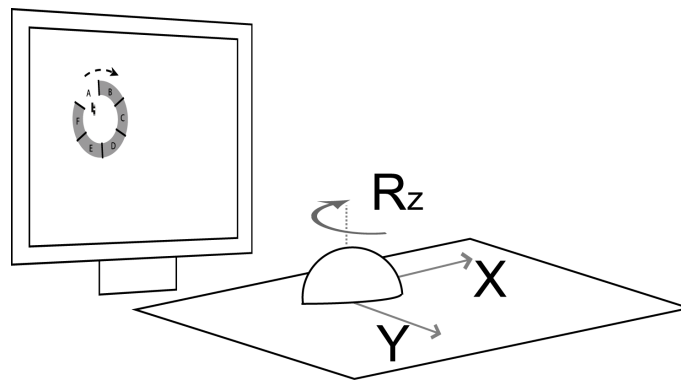


FIG. 5.2: Les trois DDL captés par le dispositif et la correspondance avec les mouvements de la « palette satellitaire » sur l'écran.

Chaque option de la palette correspond à un secteur de l'anneau et tous les secteurs ont la même dimension. Ainsi, si on veut disposer de six options, on aura six secteurs de  $60^\circ$  chacun (voir figure 5.3, gauche).

Le centre de rotation de l'anneau est décalé, et non centré par rapport au curseur, afin que ce dernier se trouve toujours sur un secteur de l'anneau. Au moment où le premier pixel d'un secteur de la palette entre sous la pointe du curseur, l'option de ce secteur est activée et sa couleur est changée. Puisque le curseur est toujours sur un des secteurs de l'anneau, il indique sans incertitude quelle est l'option activée. De plus, ce design permet d'utiliser un seul curseur (le curseur standard du système de fenêtrage) au lieu d'en utiliser un deuxième pour indiquer l'option active.

La rotation de la palette est restituée de façon continue et ses secteurs sont activés de manière prévisible et sans sauts abrupts. Lorsque le pointeur est sur le secteur, ce dernier est activé et un seul clic suffit pour utiliser l'outil associé (par exemple, en commençant un trait ou en changeant la couleur de l'objet sur lequel on clique).

### 5.3.1 Propriétés de la technique

Contrairement aux allers-retours que nécessitent les palettes classiques, un changement d'outil avec la palette satellitaire n'entraîne pas un déplacement du curseur (voir fig. 5.3, droite). On peut donc appliquer plusieurs opérations à un même objet ou à une même zone de travail sans bouger le curseur. La *transition* entre l'objet d'intérêt et l'outil continue à exister mais cette opération est effectuée par un mouvement rotatoire et non pas linéaire. En revanche, la *transition* outil-objet devient nulle puisque le curseur continue au même endroit où il se trouvait avant l'activation de l'outil.

La palette satellitaire fusionne les actions de sélection d'un outil et de son utilisation. Selon Guimbretière et al. (2005), cela contribue de façon importante pour

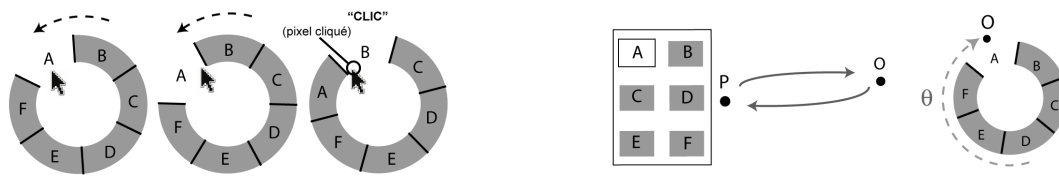


FIG. 5.3: A gauche, changement d'outil : passage de l'outil A à l'outil B puis clic sur B. A droite, les opérations nécessaires pour activer l'outil D alors qu'on travaille sur le point précis O. Dans la palette classique, le curseur doit être déplacé du point O au point P et revenir au point O. Dans la palette satellitaire, le curseur reste sur O et l'option D est « ramenée » jusqu'au curseur par un mouvement de rotation de la souris.

une sélection rapide et directe. Il me semble que la palette satellitaire dispose de cette propriété car l'utilisateur peut réaliser la rotation, cliquer et commencer à glisser dans un même geste (sans devoir repositionner le curseur après avoir choisi la couleur).

Enfin, la palette satellitaire procure une *navigaion circulaire* à l'intérieur de l'ensemble d'outils qu'elle présente. L'utilisateur peut ainsi accéder aux outils en la tournant dans un sens ou dans l'autre. Ainsi on peut toujours choisir le trajet le plus court (par exemple, dans la fig. 5.3, gauche, on va de F à B en passant par A, au lieu de passer par  $\overline{EDC}$ ). Cela plafonne à  $180^\circ$  la rotation que l'utilisateur doit imprimer à la souris afin d'atteindre n'importe quelle option de la palette satellitaire.

La palette satellitaire partage quelques propriétés avec les techniques présentées dans la section précédente. Elle se sert, comme le *Toolglass*, de l'activation spatiale relative et fusionne la sélection d'une commande avec la manipulation directe. De plus, elle présente ses options dans une zone graphique qui, comme le *Tracking Menu*, suit le curseur et reste toujours visible. Par ailleurs, elle procure la sélection d'un de ses items d'une façon similaire à celle du *Pressure Widget*. De fait, dans ces deux techniques, l'utilisateur se sert d'un périphérique d'entrée à deux DDL doté d'un transducteur supplémentaire. Les valeurs continues captées par ce transducteurs sont utilisées pour que l'utilisateur puisse faire son choix dans un ensemble réduit d'options. Dans la *palette saltelliraire* et dans le *Pressure Widget*, la manipulation continue est « discrétisée » afin de faciliter la sélection .

### 5.3.2 Un prototype rotatif et « cliquable »

Pour cette étude, j'ai utilisé un prototype construit à partir d'une souris *4D Mouse* de la société *Wacom*. Ce dispositif capte l'orientation absolue grâce à la tablette magnétique sur laquelle on le déplace. Il s'agit en effet du même dispositif déjà décrit dans le chapitre 4. Cependant par rapport à la configuration précédente du prototype, je l'ai davantage réduit afin que sa coque soit plus petite ; j'ai également installé un bouton sur sa coque (voir fig. 5.4). Il a été placé sur le sommet de



FIG. 5.4: Le prototype de la souris rotative fait à partir du 4D Mouse.

la coque afin de rendre la souris cliquable dans toutes les orientations. La coque ronde et le bouton en haut permettent en effet que le périphérique puisse être saisi indépendamment de son orientation.

### 5.3.3 Manipulation rotative

Grâce à la possibilité de manipuler simultanément les trois DDL de la souris rotative, la palette satellitaire peut être tournée en même temps qu'on la déplace. Si l'utilisateur connaît l'orientation finale qu'il souhaite attribuer à la palette, il peut la tourner jusqu'à l'atteindre en même temps qu'il déplace la palette vers une autre zone du document. Dans la pratique, il me semble que ce parallélisme est réduit. La relation temporelle entre les actions de déplacement et de rotation me paraît être plutôt du type *concomitant* (voir chap. 2.1.1), récurrente dans les gestes bi-manuels (Bailly et al., 2005).

L'inconvénient d'avoir les trois DDL sous cette configuration de capteurs est de ne pas pouvoir facilement changer une des dimensions sans que les autres dimensions ne soient changées. De fait, on tourne un peu le menu lorsqu'on ne veut que translater le curseur et on bouge un peu le curseur lorsqu'on ne veut que tourner le menu.<sup>2</sup> Afin de pallier à ce problème, la palette est immobilisée dans l'orientation qu'elle avait au moment du clic et elle reste ainsi durant l'action de dessin.

### 5.3.4 Suivi visuel de la sélection

J'ai évoqué plus haut le potentiel de cette technique pour réduire la dépendance au retour visuel lors du changement d'outil. Il y a en effet trois aspects de cette technique qui favorisent une sélection moins gourmande de concentration visuelle. Premièrement, la palette se trouve toujours à côté du curseur et, par

---

<sup>2</sup>J'ai me suis penché un peu sur une question similaire à celle-ci dans l'étude décrite dans le chapitre 4 et Almeida et Cubaud (2007).

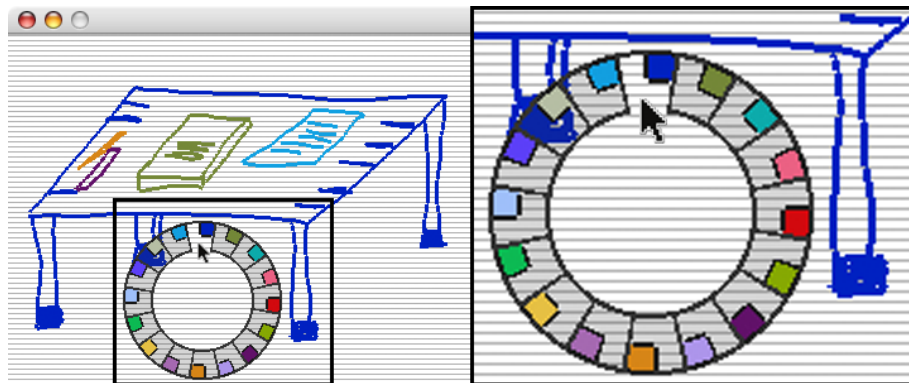


FIG. 5.5: La palette satellitaire avec seize couleurs au milieu d'un dessin.

conséquent, dans le voisinage de l'objet d'intérêt. Si l'utilisateur travaille sur les détails d'une ligne ou sur le coloriage de certains pixels, sa vision est bien focalisée sur cette zone précise de la fenêtre et il est souhaitable qu'elle ne s'en écarte pas trop.

**Mémoire musculaire.** Deuxièmement, si l'utilisateur alterne souvent entre les deux mêmes outils, il peut tirer profit de la mémoire musculaire pour ces transitions. Il exécutera ainsi ces transitions les plus fréquentes sans regarder la palette. Un retour sonore ou visuel « non-envahissant » (par exemple, un clignotement rapide de l'écran lorsqu'on passe d'un secteur à l'autre) est peut-être une piste intéressante dans cette situation afin d'offrir une confirmation qu'on ait atteint la cible visée. Pour favoriser l'utilisation de la mémoire musculaire dans d'autres situations, le nombre d'options ne peut pas être trop important et l'activation doit pouvoir être faite de façon plus « grossière ». Nous reviendrons sur cette question dans la section suivante.

**Repères pour le toucher.** Troisièmement, des repères pour le toucher peuvent aider l'utilisateur à reconnaître plus intuitivement l'orientation du prototype. Par exemple, dans Hinckley et al. (1994), l'utilisateur peut manipuler un modèle 3D d'un cerveau grâce au capteur six DDL qui se trouve à l'intérieur d'une tête de poupée. En tâtonnant le nez, le menton et le haut de la tête de la poupée, l'utilisateur comprend rapidement comment il doit orienter la tête pour que la vue (du modèle 3D) qui l'intéresse soit affichée. De ce fait, le feed-back de la rotation du modèle 3D vers une nouvelle orientation est fourni par les éléments tactiles de tête de la poupée plutôt que de l'image du modèle 3D. Cette manipulation beaucoup plus « physique » est plus naturelle que si elle était guidée par un feed-back graphique ; elle permet à l'utilisateur de se concentrer sur les aspects vraiment importants de la tâche en question.

En ce qui concerne la souris rotative, jusqu'ici, j'ai installé une bande de texture sur une portion de la coque de la souris rotative. Cette texture peut être sentie facilement et, en principe, elle n'est pas dérangement lors de l'utilisation de la souris

Boutons		
	Diamètre	Hauteur
maniable par 2 ou 3 doigts	10 à 30 mm	15 à 25 mm
maniable par toute la main	35 à 75 mm	30 à 50 mm
Nombre de degrés entre les options d'un <i>switch</i> rotatoire		
sélection avec suivi visuel	> 15° (24 options)	
sélection sans suivi visuel	> 30° (12 options)	

**TAB. 5.1:** Résumé des recommandations données dans Kroemer et Grandjean (1997) concernant la taille des boutons et l'écart en degrés entre les options d'un *switch* rotatoire.

dans d'autres contextes de travail. Si l'orientation de cette bande correspond à celle d'un outil « préféré » dans la palette, l'utilisateur peut rapidement retourner la souris à cette orientation (jusqu'à ce qu'on sent la texture sous son doigt index) pour que l'outil en question soit activé. Il est possible de communiquer davantage d'informations tactiles mais cela devra être fait dans une étude consacrée uniquement à cette question.

### 5.3.5 Nombre d'outils

Dans sa conception actuelle, plusieurs facteurs déterminent le nombre maximal de secteurs dont dispose la palette satellitaire. Si on applique une correspondance absolue entre le périphérique et la palette, c'est la précision de la captation angulaire qui détermine ce nombre maximal. Le *4D Mouse* peut discriminer des variations d'un peu moins d'un degré, ce qui nous permettrait d'exploiter 360 secteurs. En pratique, on atteindrait jamais ce nombre de secteurs pour des raisons graphiques et motrices.

**Limites graphiques.** D'une part, chaque secteur requiert une étiquette pour l'identifier et l'on peut prendre comme étalon de ses dimensions le carré de 11 pixels utilisé dans *Adobe Photoshop* pour représenter chaque outil de sa palette. Ainsi, si on souhaite présenter plus de secteurs, il faut augmenter le diamètre de l'anneau (voir fig. 5.6). Deux grandeurs pour illustrer : 26 secteurs requièrent un diamètre de 120 pixels ; 360 secteurs requièrent à peu près 1 500 pixels.

**Limites motrices.** D'autre part, sur le plan moteur, les incertitudes sont plus grandes. Dans des essais informelles, des secteurs inférieurs à 15° sont accidentellement activés et désactivés trop souvent. Par exemple, lorsqu'on va activer une option, on dépasse la cible souhaitée et on doit y retourner. Il reste ainsi à considérer la précision minimale dont disposent les doigts pour avancer ou reculer confortablement un *pas angulaire*. C'est-à-dire : combien de degrés d'écart

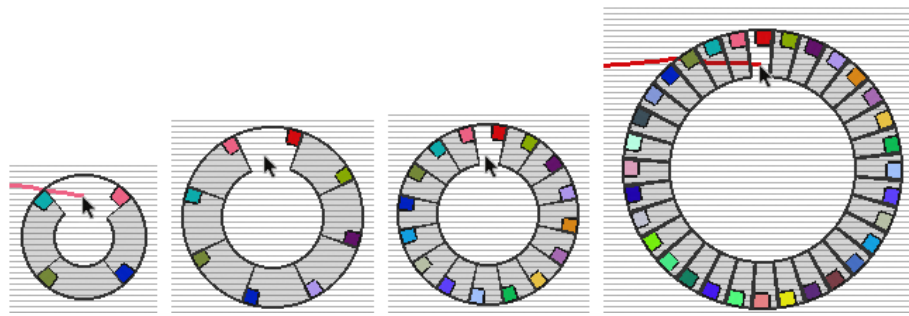


FIG. 5.6: La palette satellitaire avec quatre, huit, seize et trente-deux zones.

entre chaque option pour que la main puisse passer d'une option à l'option voisine par un geste décontracté et rapide? La réponse à cette question dépend de nombreuses variables dont (1) le ratio entre la taille de la main et le diamètre de la souris, (2) le poids de la souris, (3) le coefficient de friction entre la surface de déplacement et la base de la souris et, même, (4) le sens de la rotation.

Pour donner une idée approximative : avec notre prototype rond (sept cm de diamètre), une main longue de 20 cm peut avancer confortablement par des petits *pas angulaires* qui varient entre  $20^\circ$  et  $25^\circ$ . A peu près *seize secteurs* seraient donc envisageables.

Kroemer et Grandjean (1997) recommandent que les options d'un *switchs* rotatoires soient séparées d'au moins  $30^\circ$  les unes des autres dans une condition de sélection sans suivi visuel et d'au moins  $15^\circ$  si on se sert du contrôle en le regardant<sup>3</sup>. Les seize zones que j'ai proposées se trouvent entre ces deux limites. Un espacement supérieur à  $25^\circ$  se justifiera peut-être lorsqu'on essaiera de sélectionner un secteur sans suivi visuel.

## 5.4 Etude exploratoire : une palette de couleurs

J'ai développé une petite application de dessin (utilisant les *API Carbon* et *OpenGL*) dans laquelle le périphérique est associé à une palette satellitaire composée de seize couleurs (voir fig. 5.5). Quatre personnes habituées aux applications graphiques ont été invitées à réaliser un dessin simple utilisant la palette satellitaire.

Le retour global de la part de ces utilisateurs était très positif : ce *widget* leur semblait utile et le concept amusant. Ils ont toutefois manipulé la souris rotative avec un peu d'étonnement et de maladresse (surtout au moment de faire les traits du dessin).

**Facteurs de forme et confort du clic.** En effet, j'ai rencontré des obstacles pour

---

<sup>3</sup>Je me sers de ces recommandations avec une certaine distance car, contrairement aux souris rotatives, un *switch* rotatoire dispose d'indications tactiles à chaque option angulaire.





FIG. 5.7: La souris ronde d'Apple. (*Ye olde mouse* par Niclas Lindh)

adapter le prototype à l'ergonomie qu'exige une tâche de dessin. Pour ce type d'opération, il est nécessaire d'avoir à la fois de la précision et du confort lorsqu'on déplace la souris tout en gardant le bouton enfoncé. La tension nécessaire pour que le bouton (du prototype en question) reste appuyé et la posture que cela impose à la main ne semblent pas permettre qu'on se serve de mouvements fins et légers. Plusieurs aspects du prototype de souris rotative sont peut-être dérangeants : sa hauteur, le fait que le bouton soit un peu dur à cliquer, le fait d'appuyer sur le centre de la souris et non pas sur son devant.

En revanche, lorsqu'on saisit ce prototype sans appuyer sur le bouton, on peut le déplacer grâce à des mouvements du poignet ou même à des mouvements avec le bout des doigts. Dans ces conditions, les mouvements sont courts mais bien maîtrisés - sans la contrainte du bouton appuyé, le prototype est plutôt facile à manier. D'ailleurs, le diamètre et la forme ronde ne sont pas complètement nouveaux ; par exemple, on retrouve ces caractéristiques dans les anciennes souris Apple (voir figure 5.7, droite).

J'ai testé un autre emplacement pour le bouton : sous la base du périphérique. Cet emplacement a l'avantage de ne pas exiger une posture de clic pour l'activer. Toutefois, j'ai constaté que le bouton était parfois activé par accident et qu'il était difficile aussi de le garder enfoncé. Le retour des utilisateurs nous indique que le mieux serait de faire une bande « pressable », ou tactile, autour de la base de la coque. Une autre solution serait de voir des emplacements où les doigts se posent naturellement au moment d'un déplacement précis et d'y installer des capteurs capables de détecter le contact (Hinckley et Sinclair, 1999). Evidemment ces emplacements devraient être « indiqués » par une texture différente pour que l'utilisateur sache qu'il s'agit de zones qui déclenchent le clic. Cette détection pourrait ainsi remplacer le bouton sans faire appel à la tension musculaire que l'action de clic traditionnel nécessite.

**Occlusion et lisibilité.** Par ailleurs, j'ai reçu aussi des critiques en ce qui concerne l'occlusion générée par la palette. Pour cela, j'ai réalisé un mécanisme pour que les

secteurs désactivés disparaissent lorsqu'on conserve le bouton enfoncé. D'autres deux solutions sont également envisagées : (1) faire disparaître progressivement la palette suite à un délai de quelques millisecondes sans variation significative de l'orientation ; (2) séparer davantage le secteur activé des secteurs désactivés pour pouvoir ainsi dégager la région du curseur.

La visibilité de la palette et la lisibilité de l'outil activé semblent poser des problèmes aussi. De fait, lorsque la fenêtre de dessin se trouve emplie de traits, les options de la palette sont difficilement identifiables. Enfin, on m'a suggéré également de remplacer le curseur système par quelque chose qui représente directement le mode activé. Dans, ce cas, au lieu de la flèche, ou du curseur cruciforme il y aurait l'icône de l'outil (par exemple, un pot de peinture).

## 5.5 Travaux en perspective : activation gestuelle des outils

J'ai commencé l'exploration d'une variante pour la palette satellitaire qui est destinée aux utilisateurs experts. Bien que la maquette fonctionnelle ne soit pas encore terminée, les principes de la technique sont définis et je les rapporte ici. L'idée est de favoriser davantage la sélection sans le regard grâce à la mémoire musculaire. Le principe s'inspire dans la technique des *Marking Menus* (Kurtenbach et Buxton, 1994), dont la sélection d'une commande est faite par un geste qui peut se passer de tout feed-back visuel. Cette sélection sans regarder est possible car un geste donné est toujours associé à une même commande, quelle que soit la position du curseur sur l'écran. Dans les *Marking Menus*, la sélection sans regarder est encore plus robuste car c'est la direction du geste qui compte pour l'identifier et non pas sa longueur.

Dans le *mode expert* de la palette satellitaire, chaque option est associée à une amplitude de *pas angulaire* (voir figure 5.8). L'activation se fait à partir d'un point de départ neutre et c'est d'ailleurs pourquoi le geste pour sélectionner une option donnée est toujours le même. Ainsi, si l'on tourne la souris de 30°, quelle que soit son orientation de départ, c'est l'option associée à cette longueur qui sera activée. De ce fait, l'utilisateur peut ignorer quelle est l'option activée au moment où il décide d'en sélectionner une autre.

L'avantage de cette méthode d'activation par rapport à la méthode de base est le fait que la mémoire musculaire peut être utilisée pour toutes les options. Dans la palette satellitaire de base, les seuls gestes qui peuvent être mémorisés sont ceux dont on se sert pour alterner entre deux options connues.

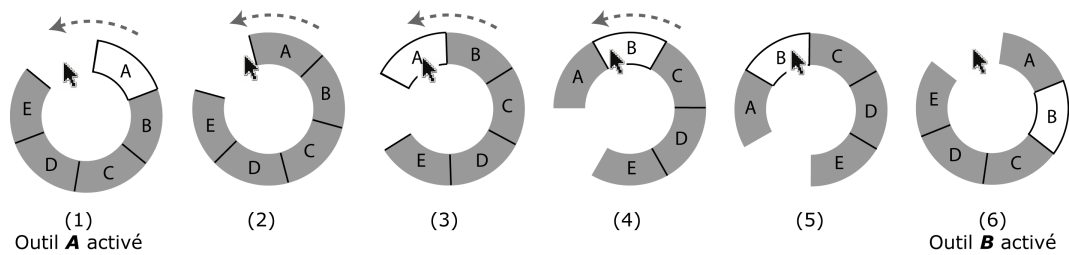


FIG. 5.8: Exemple de sélection d'un outil par la technique d'« activation gestuelle » : activation de l'outil **B** lorsque l'outil **A** est activé. La souris est en mouvement de rotation de (2) à (5); elle est statique sur (1) et sur (6). Dans cet exemple, l'outil activé au départ était le **A** mais cela n'a aucun impact sur l'activation de **B**. Si cet outil de départ était **C**, **D** ou **E**, le geste d'activation de **B** aurait été strictement le même. Le mouvement qui commence dans la figure (2), (3), (4) et (5) est réalisé dans un même geste, c'est-à-dire dans un même *pas angulaire*. Dans (1), l'outil **A** est sélectionné et la palette se trouve à une orientation neutre; dans (5) le geste finit, l'outil **B** est activé et la palette revient à une orientation neutre dans (6).

### 5.5.1 Nombre et configuration des secteurs

L'« activation gestuelle » impose un nombre de zones encore plus limité que la méthode d'activation de base. Cela est dû à deux facteurs : on doit pouvoir faire la sélection sans regarder et on doit la faire par un seul geste.

**Précision d'un « geste mémorisé ».** Lorsqu'on fait une sélection sans regarder, on dispose d'une précision plus réduite que lorsqu'on dispose d'un feedback visuel. Cela est montré dans les résultats de Øvergård et al. (2007) et dans les recommandations de Kroemer et Grandjean (1997)

**Longueur du *pas angulaire*.** Le fait de devoir faire la sélection au bout d'un seul *pas angulaire* donne de la vitesse mais aussi des contraintes à la technique. De combien de degrés peut on tourner la souris par un *pas angulaire*? Comme nous l'avons vu dans la section 5.3.5, cela dépend de plusieurs facteurs.

D'après Kroemer et Grandjean (1997), un contrôle rotatif (dont les dimensions respectent les limites rapportées dans le tableau 5.1) peut être tourné d'au moins 120° sans réengagement. Cette grandeur semble correcte pour des boutons, petits et grands, installés sur une surface verticale ou pour les petits boutons installés sur une surface horizontale. On constate même que souvent on peut les tourner de 180° par un geste continu. Cependant, on peut difficilement imprimer une rotation à la souris rotative qui dépasse 100°.

Mon interprétation de la disparité entre ces deux grandeurs est la suivante : lorsqu'on est assis et on manipule un bouton rotatif « grand » sur une table, on ne peut pas tourner tout l'avant-bras<sup>4</sup> comme on fait lorsque ce bouton est sur une

<sup>4</sup>Il s'agit d'un mouvement de supination/pronation.

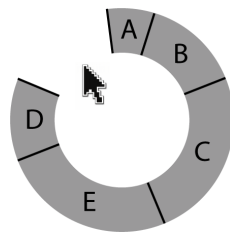


FIG. 5.9: Configuration des zones pour la palette satellitaire basée sur une sélection angulaire. Les options *A*, *B* et *C* sont activées par un geste de rotation horaire, tandis que les options *D* et *E* sont activées par un geste de rotation anti-horaire. Les options *C* et *E* disposent de longueurs infinies. Par exemple, si j'exécute un geste de rotation horaire, une fois la zone de *B* dépassée, mon geste finira pour activer *C*, quelle que soit la longueur du geste à partir de ce point.

surface verticale. Les mouvements des doigts ne suffisent pas pour imprimer une rotation supérieure à  $100^\circ$ .

**L'index de difficulté des secteurs.** Une configuration qui me semble intéressante est d'offrir trois zones d'activation situées d'un côté et de l'autre du point de départ. Cela permettrait, pour chaque sens de la rotation, que l'utilisateur puisse atteindre et qu'il puisse facilement se souvenir de trois zones. Il s'agit (1) d'une zone à laquelle on accède par un geste « *le plus court possible* », (2) une deuxième par un geste *moyen* et, enfin, (3) une troisième zone accessible par un geste « *le plus long possible* ». Cependant, il me paraît que la rotation de la souris dans le sens anti-horaire est moins précise et dotée d'une portée plus limitée. Deux zones « anti-horaire » pour la palette satellitaire gestuelle (voir figure 5.9) sont donc envisagées : une proche et une éloignée.

Le rapport *précision : vitesse* des mouvements indique que les cibles les plus éloignées sont plus difficiles à atteindre que les plus proches. Afin de proposer des options ayant un *index de difficulté* similaire (Fitts, 1954), les secteurs sont dotés de longueurs différentes ; elles sont proportionnelles à la valeur de l'angle qui les sépare du point de départ.

## 5.6 Conclusion

Ce chapitre a décrit la problématique des changements d'outils et d'attributs dans les applications graphiques, soulevé les contraintes qui interviennent dans ces opérations et proposé une technique basée sur une souris rotative - la « palette satellitaire » - comme un moyen de faciliter la transition entre les outils.

Dans cette étude exploratoire, la palette satellitaire a été envisagée surtout comme un meta-outil permettant de sélectionner rapidement et sans effort des outils de travail. Elle serait donc utilisée pour « porter » les outils principaux ou les

outils/attributs les plus fréquemment activés. L'utilisation de la palette pour naviguer dans un grand ensemble d'options (comme, par exemple, la liste de toutes les polices d'écriture) ou dans une liste hiérarchique n'a pas été évoquée. Je compte étudier ces cas ainsi que d'autres possibilités de design (comme celles proposées dans Bier et al. (1993) : « couper-coller visuel », loupes sémantiques, etc). D'autres scénarios d'usage peuvent aussi contribuer à améliorer les propriétés de cette technique.

Les premiers retours obtenus avec cette exploration suggèrent que les palettes satellitaires sont une technique d'interaction originale et intuitive. J'ai exploré plus en détail une configuration pour les utilisateurs « novices » et j'ai proposé les principes d'une différente version de cette technique (un « mode expert ») qui serait destinée aux utilisateurs avancés. Ce mode expert propose des « raccourcis moteurs » qui pourraient accélérer davantage la sélection d'un outil et permettre la sélection sans regarder. Toutefois, il ne me semble pas que le mode expert puisse être appris à partir du mode novice (Bailly et al., 2007) ; chaque mode a son propre principe d'activation.

Cette étude doit être approfondie et des évaluations sont nécessaires pour répondre à certaines questions : la technique procure-t-elle un soulagement du canal visuel ? Est-elle plus rapide que des techniques similaires (palette graphique traditionnelle et autres techniques *post-wimp*) ? Les utilisateurs pourraient se servir de repères tactiles installés sur la coque de la souris si l'orientation de celle-ci correspondait de façon absolue à celle de la palette ?

Enfin, je cherche également à mieux comprendre le rapport qui existe entre la précision du mouvement et les facteurs de forme du dispositif. Par exemple, le poids du dispositif et la friction engendrée par sa base semblent être des aspects importants à considérer afin de permettre des réglages courts et précis. Par ailleurs, la taille du dispositif soulève aussi des questions importantes. Une souris rotative dotée d'un diamètre plus petit (inférieur à six centimètres) permettrait des pas angulaires plus longs mais, en revanche, elle réduirait la précision du mouvement (Øvergård et al., 2007).



# Navigation et immersion dans une collection de documents numériques

*L'exploration d'un catalogue de documents numériques n'est pas une activité intuitive et favorise peu à la découverte de nouveaux documents. Une des pistes explorées par certains travaux du domaine a été de fournir davantage d'indices visuels aux documents présents dans les catalogues. Cela permettrait à l'utilisateur d'employer les heuristiques de recherche visuelle dont il se sert pour ses activités réelles. L'affichage de documents dans un espace peut également permettre à l'utilisateur de se servir de sa mémoire spatiale afin qu'il se rappelle l'emplacement d'un document déjà consulté. Ce chapitre présente un travail sur l'adaptation d'un dispositif d'affichage immersif pour enrichir la visualisation et la navigation à l'intérieur d'un catalogue 3D de documents numériques.*

Les bibliothèques numériques constituent des entrepôts de stockage et d'accès à une masse de documents en ligne : à la fois des documents numérisés et des documents « nativement » numériques. Ces bibliothèques en ligne présentent une foule de données textuelles associées aux documents archivés. Les documents écrits sont indexés par leurs métadonnées (nom de l'auteur, mots-clés, titre du volume, date et maison d'édition, etc.) et, de plus en plus souvent, par le texte extrait de l'image de chaque page numérisée. Grâce à ces données, il est désormais possible de trouver, à l'aide d'un moteur de recherche, des passages et des résultats qui étaient inaccessibles lorsque la seule information extérieure dont on disposait était la notice bibliographique du document écrit.

Néanmoins, bien que plus accessibles, les documents numériques peuvent passer inaperçus bien plus facilement que les documents en papier. Ces derniers oc-

cupent un espace, sont sous l'effet de la pesanteur, peuvent être trouvés par hasard lorsqu'on ne les cherche pas, etc. En revanche, les documents numériques, par défaut, ne se « manifestent » que lorsqu'ils sont interrogés ; le reste du temps ils sont « invisibles » aux yeux de l'utilisateur. C'est d'ailleurs ce que l'on constate lorsqu'on visite des sites de librairies en ligne<sup>1</sup> ou des bibliothèques numériques. Cette situation est comparable à celle des livres qui se trouvent en réserve (en opposition à ceux qui se trouvent en accès libre) dans une bibliothèque réelle. Il est probable qu'un nombre important de ces livres n'ait été que rarement consulté par un lecteur.

Ce chapitre décrit l'utilisation d'un périphérique d'affichage immersif comme un moyen d'enrichir la consultation d'un catalogue de documents numériques. Quelques éléments de contexte sont d'abord évoqués : la problématique de la consultation de catalogues numériques, quelques approches existantes dont le but est de rendre cette consultation plus visuelle et certaines techniques pour enrichir l'espace visuel de l'utilisateur. Cette étude s'est basée sur l'interface de consultation 3D d'une collection de documents proposée par Cubaud et al. (2002). Son objectif principal a été d'étudier la faisabilité technique d'utiliser un périphérique d'affichage immersif pour faciliter la visualisation et la navigation dans cette interface 3D. Le dispositif immersif, la *VisionStation*, et les développements logiciels nécessaires à son utilisation sont décrits. Enfin, les pistes et les obstacles rencontrés lors de son utilisation sont expliqués ainsi que le travail réalisé pour les contourner.

## 6.1 Au-delà du catalogue de titres

**Architecture de navigation des grands sites.** Dans un site disposant d'un nombre important de documents numériques, l'utilisateur accède à un livre soit via le résultat d'une interrogation dans le moteur de recherche, soit via la navigation à travers les rubriques définies par les concepteurs du site. D'une part, pour le moteur de recherche, l'utilisateur devra savoir formuler une requête de la façon la plus précise possible (en fournissant le nom de l'auteur ou les mots présents dans le titre). Cela n'est pas toujours possible (Rao et al., 1995). D'autre part, les rubriques peuvent être organisées de deux façons : « exacte » ou « ambiguës » (Rosenfeld et Morville, 1998). Dans un schéma *exact*, les rubriques regroupent les documents de manière exclusive : par l'année d'édition, par la première lettre du nom de l'auteur, par la langue dans laquelle le livre est écrit. Dans une organisation *ambiguë*, un même livre peut être « rangé » dans plus d'une rubrique ; elles présentent des frontières floues et sont plus expressives du point de vue de l'utilisateur. C'est le cas des rubriques thématiques : « littérature », « livres de voyage », « culinaire »,

---

<sup>1</sup>Il est vrai que les librairies en ligne explorent de plus en plus des techniques permettant au visiteur de découvrir de façon un peu aléatoire quelques-uns de ses produits. Quelques-unes de ces techniques sont mentionnées dans ce chapitre.



etc.

**Quatre démarches de l'utilisateur.** Rosenfeld et Morville distinguent quatre types de démarche qu'un utilisateur de site Web peut entreprendre : (1) recherche *d'un item connu d'avance*, (2) recherche *sans question*, (3) recherche *sans attente à un type de réponse précise* et (4) recherche *compréhensive*, lorsqu'on essaie de voir tous les livres qui traitent d'une thématique donnée. C'est précisément dans les cas (2) et (3) que les interfaces traditionnelles - listes et moteur de recherche - échouent. Une organisation ambiguë des rubriques des documents peut procurer des pistes pour l'utilisateur mais il se retrouvera devant des listes difficiles à « balayer ».

Par ailleurs, le problème n'est pas résolu non plus lorsqu'on interroge directement le moteur de recherche. Les résultats d'une recherche sont souvent nombreux et ce sous-ensemble de documents constitue une autre collection dans laquelle on doit également naviguer pour trouver ce que l'on cherche. La plupart du contenu reste donc « invisible ». Dans de telles conditions, comment de nouvelles interfaces peuvent-elles faire en sorte que les documents numériques « remontent plus facilement à la surface » ?

### 6.1.1 Exploration d'un catalogue et découverte fortuite

Lorsque l'utilisateur n'a pas une question précise à poser, l'interface devrait lui permettre de découvrir « fortuitement » ce qu'il cherche. Comment une interface peut-elle permettre à l'utilisateur d'explorer un corpus de documents comme lorsqu'il visite les rayons d'une bibliothèque ou d'une librairie ? Dans ce type d'exploration intuitive, l'utilisateur se dirige vers certains documents, attiré par leurs attributs visuels ; attributs qu'il n'aurait pas su formuler dans une requête.

Les librairies en ligne, telles que *Amazon.com*, ont développé des mécanismes pour que les livres deviennent visibles autrement que par une recherche explicite. Les systèmes de recommandation en sont un exemple. Il s'agit du croisement des informations sur les intérêts des utilisateurs (livres achetés et livres visités) ayant des profils similaires. Cela permet à ces sites de proposer à l'utilisateur des livres sur un critère avec un faible degré de hasard.

Par ailleurs, des sites de bibliothèques numériques comme *Gallica 2* et des entrepôts de photo numérique, tels que *Flickr*, se servent aussi d'affichages de mots-clés par « nuage de *tags* » (voir figure 6.1, gauche). Ces mots peuvent être ou bien issus des métadonnées des documents (saisis au moment du catalogage) ou bien générés par les utilisateurs (les mots utilisés pour étiqueter un document donné). La taille du mot est généralement *mappée* à la fréquence avec laquelle il est employé. Le nuage de *tags* constitue ainsi un moyen quasi-aléatoire et quasi-graphique pour que les utilisateurs aient davantage des pistes sur le contenu d'un entrepôt.

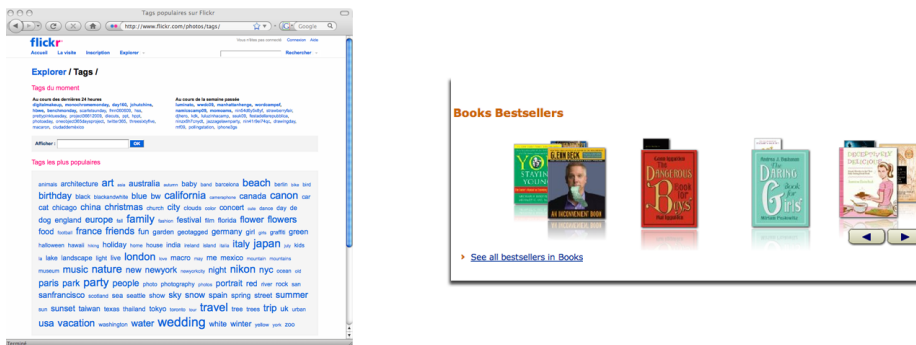


FIG. 6.1: A gauche, un « nuage de mots-clés » - employé par les utilisateurs du site Flickr pour décrire les photos qu'ils mettent en ligne. A droite, une partie de la page d'accueil d'Amazon.com montrant les images des couvertures des livres dans une interface « carrousel 3D ».

## 6.1.2 Représentation visuelle des documents

La représentation graphique des livres est un moyen de favoriser une navigation plus intuitive dans une collection. Des sites comme Fnac ou AbeBooks se servent de l'image de la couverture du livre pour le représenter ; ou bien accompagnées de la notice du livre, ou bien *per se* dans une vitrine (par exemple dans un « carrousel 3D », comme montre la fig. 6.1, droite). La même pratique commence à exister dans les bibliothèques numériques. Cependant, dans celles destinées aux livres anciens, la couverture originale peut se trouver très dégradée ou avoir été remplacée par une couverture qui n'apporte pas d'information ; cela rend l'utilisation de cette ressource moins intéressante. Le site de recherche de livres Google Books a développé un générateur automatique de couvertures pour les livres dont les couvertures qui n'existaient pas ou qui n'avaient pas été numérisées (voir figure 6.2, gauche).

Un autre exemple intéressant de l'exploitation de l'image de la couverture du livre est donné par le logiciel Delicious<sup>2</sup> (voir figure 6.2, droite). Cette application, destinée aux particuliers, permet à l'utilisateur d'avoir une représentation numérique des livres qu'il possède. Avec l'exemplaire physique en mains, il passe le code-barres du livre devant son webcam<sup>3</sup> qui va alors chercher les méta-données du livre (titre, auteur, résumé) ainsi que l'image de sa couverture sur le site Amazon.com. L'utilisateur peut ainsi « gérer » ses documents physiques (par exemple, regrouper les préférés, regrouper ceux qui sont au bureau et ceux qui sont à la maison, faire une liste de ceux qu'il a prêtés, etc.) à l'aide de cette représentation graphique.

Dans les exemples que je viens de donner, la taille du volume n'est pas prise en compte : les livres apparaissent comme s'ils avaient tous la même dimension. Toutefois, cette information, lorsqu'elle est traduite graphiquement, fournit un in-

<sup>2</sup>Delicious Library : <http://www.delicious-monster.com/>

<sup>3</sup>Si l'utilisateur ne dispose pas d'une webcam, il peut saisir le code-barres du livre.

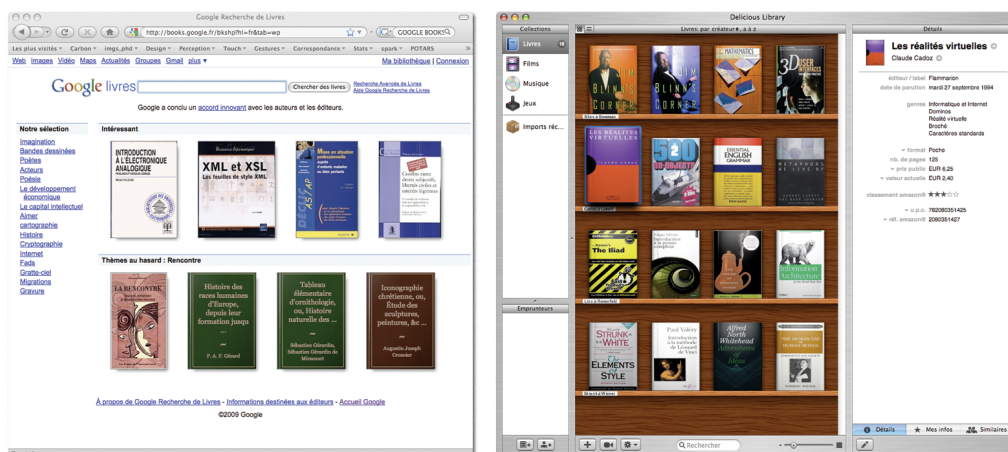


FIG. 6.2: A gauche, une copie d'écran de la page d'accueil du site *Google Books* (les couvertures des trois livres à droite dans la ligne du bas ont été générées par le site). A droite, une copie d'écran du logiciel de gestion personnelle de documents (livres, DVDs, CDs), *Delicious*.

dice visuel qui permet à l'utilisateur de discriminer rapidement les livres qui l'intéressent. Par exemple, la figure 6.3 montre deux livres qui apparaissent avec la même taille dans l'interface de *Delicious* (fig. 6.2, droite). Cependant, ces livres proviennent de collections connues (l'un est un essai de Folio chez Gallimard, l'autre un livre d'informatique de chez *O'Reilly*) et les utilisateurs qui connaissent vaguement ces deux collections savent facilement que les volumes de l'une sont petits (11 x 18 cm) tandis que ceux de l'autre sont plus grands et carrés (18 x 23 cm). Quelques-uns des projets mentionnés dans les prochains paragraphes affichent les « tailles relatives » des volumes.

**La tranche du livre est un icône efficace.** La tranche du livre est l'un des repères dont se servent les lecteurs pour trouver les livres qu'ils cherchent dans une librairie et dans une bibliothèque. Les tranches fonctionnent comme un « icône » du document et les concepteurs graphiques des livres essaient de faire en sorte que cette partie de la reliure puisse jouer ce rôle avec un maximum d'efficacité. De fait, la tranche permet d'appréhender et d'identifier le nombre de pages, le type de reliure, la hauteur, la collection et la maison d'édition. L'auteur et le titre y sont souvent indiqués mais ces informations sont moins directement accessibles ; un effort de lecture est nécessaire si on souhaite les appréhender. Cubaud et al. (1998) proposent d'utiliser directement des images des reliures des livres pour les représenter. Dumas et al. (2002) se servent des images des couvertures des livres pour représenter les volumes dans une bibliothèque numérique. Ces images sont rétrécies dans le sens de la largeur pour qu'elles aient la géométrie du « dos » d'un volume. La déformation engendrée par cette transformation a l'inconvénient de rendre certaines figures et textes illisibles.

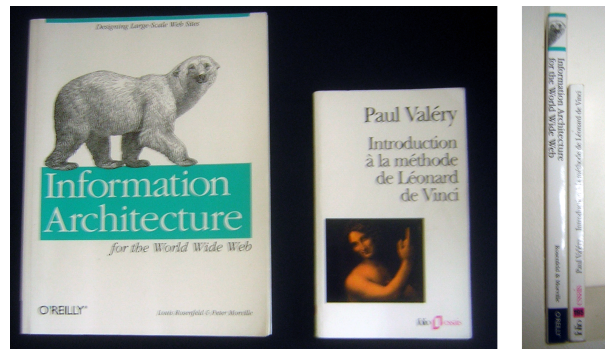


FIG. 6.3: Deux livres de collections connues placés côte à côte. La différence de taille entre eux est l'un des attributs qui permet à l'observateur de les reconnaître à distance avant de devoir lire le titre de chacun.

### 6.1.3 Organisation spatiale de la collection

En poursuivant l'analogie avec la façon dont on se sert des livres réels, il est important de s'interroger sur l'organisation spatiale des documents. Comment l'interface peut-elle favoriser les « heuristiques visuelles » pour que l'utilisateur sache aller naturellement vers les documents les plus susceptibles de l'intéresser ? Pourra-t-il facilement retrouver un document déjà consulté ?

L'affichage des icônes des livres dans une configuration 2D sur une position fixe basée dans la métaphore d'un étagère, comme dans *Delicious* ou dans *LibViever* (Rauber et Bina, 2000), permet de retrouver facilement un document connu puisqu'il reste toujours au même endroit. Ces heuristiques sont des stratégies banales auxquelles on est habitué mais qui ont été peu intégrées dans les interfaces WIMP. Par exemple, grâce aux tranches (comme nous venons de voir), les utilisateurs peuvent repérer les volumes épais ou de grandes dimensions. De même, l'utilisateur peut espérer trouver les volumes très lourds dans des étagères du bas, ou bien les collections et les thématiques peuvent être rassemblées dans chaque étage des étagères, etc.

### Bibliothèques 3D

La métaphore de l'étagère pour organiser les documents est poussée encore plus loin par les interfaces 3D. Ces interfaces offrent un espace « infini » pour la disposition des documents. L'utilisateur peut zoomer sur un document éloigné, l'agrandir sans effort et de façon fluide. Dans le *Web Forager*, des étagères sont disposées dans un espace 3D et les documents (des pages Web et non pas des livres numérisés) peuvent être rangés dans des livres (Card et al., 1996).

D'autres projets proposent une interface qui reproduit l'architecture d'une vraie bibliothèque. Par exemple, Dumas et al. (2002) ont proposé une interface de bibliothèque virtuelle pour que les utilisateurs accèdent aux données des livres d'une bi-

bibliothèque réelle. Cette interface 3D reproduit l'architecture (étages, murs, placement des étagères) de la bibliothèque réelle (il s'agit ici de la bibliothèque universitaire de Lille). Leur hypothèse : les utilisateurs habitués à la bibliothèque réelle pourront se bénéficier de cet acquis et retrouver facilement, dans son double virtuel, les rayons et les livres qu'ils cherchent. Toutefois, cette interface est peut-être uniquement avantageuse lorsque l'utilisateur connaît la bibliothèque réelle. Cela est rarement le cas le cas lorsqu'il s'agit d'une bibliothèque numérique consultée par des utilisateurs de plusieurs régions du monde. Par ailleurs, une interface reproduisant l'architecture de la bibliothèque physique peut créer des obstacles qui ne font pas partie de la tâche principale : l'utilisateur devra concentrer son attention sur le déplacement de la caméra à travers les couloirs et les escaliers de la bibliothèque (Almeida et al., 2006). De même, dans une bibliothèque numérique patrimoniale, comme c'est le cas du CNUM par exemple, les ouvrages en ligne ne se trouvent pas en « libre accès ». De ce fait, la localisation réelle des documents qui ont été numérisés n'est pas représentable ou ne présente pas d'intérêt d'être représentée telle quelle (Cubaud, 2006).

## 6.2 Expérience visuelle enrichie

Lorsque l'utilisateur se trouve face à un grand catalogue de documents à parcourir (que ce soit en texte, ou représenté en 2D ou en 3D), il ne pourra pas visualiser l'ensemble en une seule fois. De fait, plus l'écran est petit, plus il devra interagir via le dispositif d'entrée pour que les documents deviennent visibles.

Si on prend comme analogie le travail avec les documents papier, les études de Malone (1982), puis de O'Hara et Sellen (1997), montrent l'importance d'un espace de travail large et facilement accessible. Malone observe que la disposition spatiale, la création de piles classées, mais aussi de piles « floues » permet à une personne de se repérer au milieu de plusieurs dizaines de documents, de se rappeler les rapports à lire, etc. O'Hara et Sellen observent comment le fait de consulter simultanément plus d'un document, de les mettre côte à côte, de les plier, etc. facilite la tâche de lecture et de prise de notes. Par ailleurs, Hutchins (1995) suggère que, en distribuant l'information dans l'espace, leur complexité (la relation entre les éléments, la priorité d'un sur l'autre, etc.) est « transférée » à cette organisation. Ainsi, un espace virtuel *visuellement plus large* pourrait favoriser à cette utilisation intelligente de l'espace.

Lorsqu'on travaille dans un cadre réel, l'espace visuel est beaucoup plus large et facilement re-configurable. On réalise, presque sans efforts, des mouvements des yeux, de la tête et, voire même du tronc afin de chercher et de consulter des informations dans un grand espace de travail. On contrôle plusieurs variables visuelles simultanément, à savoir l'orientation que la main gauche donne aux papiers qui sont sur la table, la distance à laquelle la main droite tient une feuille de papier, etc. En revanche, interagir avec des documents numériques est frustrant,

l'utilisateur regarde le bureau comme à travers une petite fenêtre et les variables visuelles sont contrôlées de façon « hachée ».

C'est dans la perspective d'une interface 3D que je traiterai maintenant des possibilités d'enrichir la modalité visuelle de l'application interactive. Outre l'élargissement du champ visuel, d'autres techniques peuvent être exploitées pour que l'expérience visuelle de l'utilisateur soit enrichie lors de la consultation de documents.

**Parallaxes.** La *parallaxe oculaire* fournit une piste statique sur la profondeur relative de chaque objet observé. Plusieurs techniques peuvent en tirer profit : des casques avec des vues séparées pour chaque œil ; des lunettes anaglyphes qui filtrent l'image destinée à chaque œil ; des lunettes dotées d'obturateurs synchronisés à un écran qui alterne l'affichage des vues gauche et droite ; ou encore des écrans réticulés qui font en sorte que chaque œil ne voit que l'image qui lui est destinée. Dans une scène où les documents sont représentés à des distances différentes, la parallaxe oculaire permet à l'utilisateur de savoir intuitivement différencier si un document apparaît petit parce qu'il est petit ou parce qu'il est éloigné.

Par ailleurs, la variation de la position des éléments regardés lorsqu'on déplace la tête fournit aussi des pistes importantes sur la distance à laquelle se trouvent ces objets. Il s'agit du phénomène de *parallaxe motrice*. A l'aide de la tête, mais aussi de tout le corps, on se sert quotidiennement de cet effet. Outre la perception de la profondeur, cela permet également que l'on voie partiellement des objets cachés derrière d'autres objets (Gibson, 1979). Avec la prolifération des *webcam*, des périphériques pour la suivi du mouvement<sup>4</sup> et des algorithmes pour le suivi de la tête et des yeux, il est désormais possible d'intégrer cet effet comme un paramètre de l'interface que l'utilisateur contrôle. Il est donc possible que cette captation soit intégrée, de façon transparente, au cadre de l'informatique bureautique de demain. Dans une interface 3D, similaire à celles décrites dans la section précédente, cela donnerait un moyen rapide et intuitif à l'utilisateur d'« inspecter » les documents couverts par d'autres.

### Immersion visuelle.

L'immersion visuelle constitue un autre domaine qui, à part par les investigations de la Réalité virtuelle, a été peu exploré dans l'Interaction homme-machine. Les dispositifs immersifs permettent que l'utilisateur se serve davantage de son champ de vision pour travailler avec des documents. L'appareil oculaire procure un champ de vision d'environ 200° en horizontal et 150° en vertical (Arthur, 1996). Les écrans traditionnels n'utilisent qu'une petite zone de ce champ (entre 60° et 90°). Une exploitation plus importante de ce champ pourrait augmenter le confort

---

<sup>4</sup>Voir, à ce sujet, le prototype bas coût proposé par J. Lee. A l'aide de la caméra infra-rouge d'une *Wii* la position relative de la tête est captée : <http://johnnylee.net/projects/wii/>

de l'utilisateur dans la réalisation de tâches cognitives.

Bowman et McMahan (2007) distinguent le concept d'*immersion* de celui de *présence*. Le degré d'immersion désigne les aspects objectifs quant à la fidélité de restitution de l'environnement virtuel (champ de vision, stéréoscopie, définition de l'image, etc.). La présence est subjective et indique si, sur le plan psychologique, l'utilisateur « se sent » dans l'environnement restitué par l'application. Une application très immersive peut fournir un faible sentiment de présence et vice-versa. La présence est importante surtout dans les applications destinées au loisir, à l'apprentissage et aux traitements psychologiques (comme, par exemple, des thérapies contre les phobies).

**Dispositifs immersifs.** L'aspect de l'immersion qui intéresse cette présente étude est celui de la disposition spatiale de l'information visuelle. A ce propos, Bowman et al. (2004) précisent la différence entre deux caractéristiques des dispositifs d'affichage : le *champ de vision* et le *champ de regard*<sup>5</sup>. Le champ de vision désigne l'étendue angulaire, verticale et horizontale, que la surface d'affichage occupe dans le champ de vision de l'utilisateur lorsque celui-ci est statique. Par exemple, un écran occupe entre 60° et 90° du champ de vision de l'utilisateur (variant en fonction de sa taille et de la distance qui le sépare de l'utilisateur). A ce sujet, Patrick et al. (2000) remarque que visualiser un environnement via un grand champ de vision permet à l'utilisateur de mieux comprendre et de mieux se repérer dans cet espace virtuel.

Le champ de regard désigne l'étendue de la surface d'affichage autour de l'utilisateur. Une cave présente un champ de regard de 360° (en horizontal du moins). En revanche, un écran immersif sphérique peut occuper la quasi-totalité du champ de vision de l'utilisateur mais, lorsque celui-ci tourne sa tête, cette surface d'affichage n'est que partiellement visible. A l'inverse, un casque de réalité virtuelle peut disposer d'un champ de vision réduit (par exemple, 40°) mais avoir un grand champ de vision. Ainsi, lorsque l'utilisateur tourne la tête, l'image restituée correspond à la nouvelle orientation de la caméra dans la scène virtuelle.

Enfin, des dispositifs comme le *Cave* et les casques stéréoscopiques sont l'objet de plusieurs expériences par les chercheurs de Réalité virtuelle. Ils sont cependant envahissants pour le cadre d'interaction bureautique : les casques coupent l'utilisateur de son environnement et sont peu confortables pour une utilisation longue tandis que le *Cave* demande un espace physique trop important. Dans un cadre bureautique, on peut se servir de la projection sur de grandes surfaces (Patrick et al., 2000) ou d'utiliser plusieurs écrans, disposés les uns à côté des autres. Ce type d'assemblage peut fournir une surface d'affichage large et de haute résolution (Mackinlay et al., 2003; Shupp et al., 2009). Notre choix s'est porté vers le système *VisionStation* - présenté dans la section suivante - un dispositif immersif individuel doté d'une surface de projection hémisphérique

---

<sup>5</sup>En anglais, « *field of regard* ».

## 6.3 Un poste de consultation et lecture immersif

Dans des travaux précédents, Cubaud et al. (1998, 2002) ont étudié de nouvelles métaphores pour la représentation de la collection d'une bibliothèque numérique. Ces travaux ont été le point de départ pour l'exploration décrite dans cette section. L'espace virtuel, dans lequel le catalogue de documents est présenté, est cylindrique et correspond à l'une des géométries de scène étudiée par Topol (2002). Les documents sont représentés par l'image du dos de la reliure du volume original. Ces images préservent les rapports de taille des tranches réelles, ce qui permet à l'utilisateur d'estimer à quels types de livre (et collection) ces images correspondent. Les tranches sont organisées dans des « étagères ».

### 6.3.1 Fonctionnement du périphérique d'affichage immersif

Le dispositif *VisionStation*, modèle *VS1024*, a été utilisé dans cette étude. Il s'agit un dispositif individuel d'immersion visuelle produit par la société *Elumens*. Il est composé d'un écran hémisphérique de 1,5 m de diamètre, qui est tenu debout par un trépied à roulettes et d'une tablette centrale dans laquelle est logé un projecteur *Epson*. Ce dernier, doté d'une résolution de 1024 x 768 pixels, est équipé d'une lentille de type fish-eye capable de projeter l'image sur la totalité<sup>6</sup> de la partie intérieure de la surface hémisphérique. L'utilisateur s'assoit devant la surface projetée et, dans une configuration idéale, ses yeux se trouvent à peu près à la hauteur du centre de la sphère. Il peut se servir de la tablette comme une table de travail, sur laquelle il peut placer un clavier et une souris.

Livrée avec le dispositif matériel, la bibliothèque *Spherical Projection for Images* (SPI) est destinée à traiter les images générées par l'application en vue de leur projection.<sup>7</sup> Cet API repose sur une bibliothèque de fonctions graphiques bas niveau (*OpenGL* ou *Direct3D*). Les fonctions de la bibliothèque SPI interviennent dans deux étapes de la génération de l'image finale : lors du rendu d'une image grand angle et lors de la compensation de la déformation sphérique.

### 6.3.2 Rendu d'une image grand angle

Lorsque l'utilisateur est assis face à la *VisionStation*, sa surface concave occupe environ 160° de son champ de vision horizontal et environ 130° du champ de vision vertical. Ainsi, il faudrait que la caméra de la scène virtuelle projetée ait un champ de vision d'une même amplitude. Cependant, les cartes graphiques privilégient les calculs des rendu par projection perspective. Ce type de projection ne produit pas des résultats acceptables lorsque le champ de vision de la camera dépasse 90°. En effet, au-delà de cette valeur, les informations de la périphérie du

---

<sup>6</sup>Avec peu de déformation entre les pixels du centre et ceux de la périphérie.

<sup>7</sup>Voir la brochure de l'API livrée avec le dispositif (Elumens, 2001).



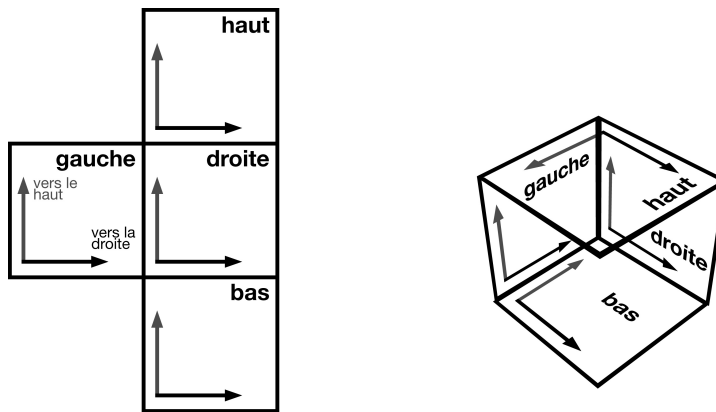


FIG. 6.4: L'image finale formée à partir des quatre images intermédiaires (d'après Bourke, 2004).

champ seront progressivement moins représentées que celles du centre. Le rendu d'un champ de vision de  $180^\circ$  est donc infaisable par la projection perspective standard.

**Rendu d'images intermédiaires.** La solution mise en oeuvre par le SPI consiste à produire une image finale à partir de la fusion d'images intermédiaires. Tout d'abord, quatre rendus sont générés - appelés « canaux » - à partir d'un même point de vue de la scène 3D. Ces rendus sont générés en orientant une caméra ( $45^\circ$  à chaque fois), qui dispose d'un champ de vision de  $90^\circ$  (horizontal et vertical), vers le haut, vers le bas, vers la droite et vers la gauche (voir fig. 6.4). Les *viewports* de ces quatre caméras devront être carrés et les dimensions des images résultantes doivent être des puissances de deux (en pixels).

Les quatre canaux ne sont pas affichés ; ils sont dessinés sur une *surface temporaire*, telle que la mémoire de texture ou le *Pbuffer*. La disponibilité de ces zones de dessin intermédiaires varient beaucoup d'un matériel à l'autre. Si l'on n'arrive pas à se servir de ces deux surfaces mentionnées, on peut, en dernier recours, dessiner les canaux sur l'arrière-tampon. L'inconvénient d'utiliser l'arrière-tampon comme surface intermédiaire vient du fait que les dimensions des canaux seront limitées à celles de la fenêtre de l'application. La définition du rendu final sera donc dépendante de la configuration de l'affichage du matériel utilisé.

Dans la configuration utilisée pour cette étude, la définition était fixée à  $1024 \times 768$  pixels, c'est-à-dire à la définition du projecteur de la *VisionStation*. Lorsqu'on est obligé d'utiliser l'arrière-tampon, les dimensions des quatre canaux seraient limitées à  $512 \times 512$  pixels. Le résultat final d'un rendu formé par des images intermédiaires de  $512 \times 512$  pixels est moins acceptable que lorsque ces images font  $1024 \times 1024$  pixels. Une fois les quatre images prêtes, les images sont fusionnées dans une grande image.

**Déformation compensatoire.** Cette étape consiste à assembler les canaux dans une seule image et ensuite à la transformer pour qu'elle soit compatible avec la dé-

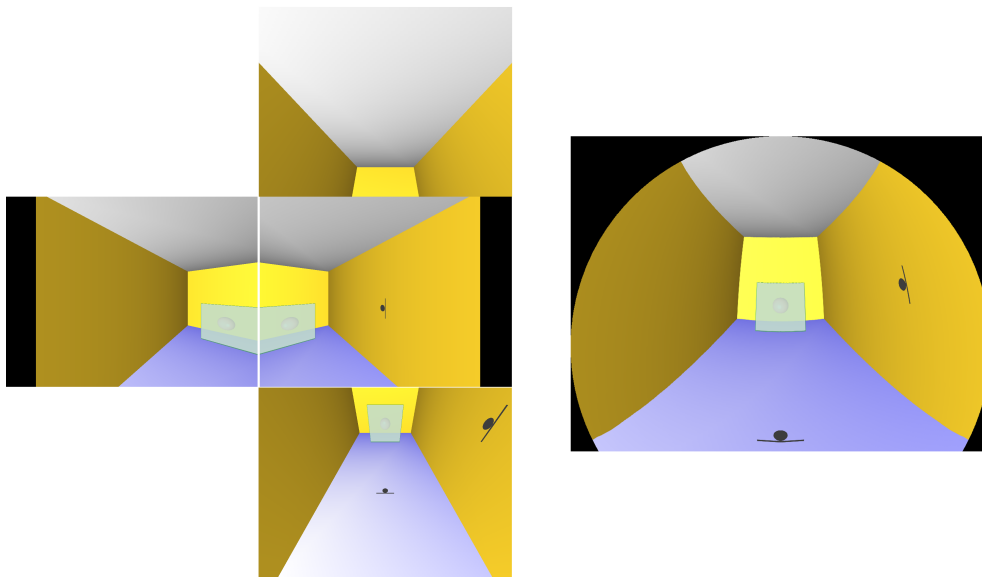


FIG. 6.5: A gauche, les quatre « canaux » utilisés pour créer l'image grand angle et le rendu final. A droite, l'image finale après le plaquage des quatre rendus intermédiaires sur le maillage hémisphérique. Ici, ce procédé est illustré dans le jeu Pong.

formation de la surface de projection. Le rendu de chaque canal est utilisé comme texture. Ces quatre textures sont plaquées sur un maillage convexe afin de compenser la géométrie concave de l'écran (fig. 6.5). Enfin, un cinquième rendu est produit à partir du maillage « texturisé » et cette image finale, dessinée sur l'arrière tampon, est finalement projetée sur l'écran de la *VisionStation*.

Sans cette étape de transformation, l'image initialement destinée à un écran plat apparaîtrait déformée. La distribution des pixels n'est pas homogène entre les quatre canaux. Les textures des canaux gauche et droit sont responsables pour bien plus de la moitié de l'image finale.

Cet assemblage en quatre canaux n'est qu'une des configurations proposées par le SPI. Elle a été utilisée dans le présent prototype. Il s'agit de la configuration produisant le meilleur résultat visuel mais aussi la plus lourde du point de vue des calculs et la mémoire utilisée. D'autres possibilités se servent de deux ou trois redus intermédiaires mais génèrent un champs de vision de caméra moins large.<sup>8</sup>

### 6.3.3 Immersion visuelle dans la collection

L'application est composée par une scène hémicylindrique dotée de cinq étages de quatre étages chacune, ce qui permet d'afficher environ 400 tranches de livres. Ces 400 icônes peuvent être tous vus en même temps lorsqu'on recule la caméra complètement ; la plupart des tranches conservent, même lorsqu'elles apparaissent très réduites, des indices visuels permettant de les identifier.

<sup>8</sup>Voir à ce sujet (Elumens, 2001).



FIG. 6.6: Copies d'écran du catalogue immersif avec l'affichage traditionnel et avec le rendu à deux étapes qui crée un champ de vision de 180°.

### Navigation dans la collection

Afin de rendre plus efficiente et naturelle la navigation dans l'espace virtuel de la collection, une « souris 3D » est utilisée pour contrôler le point de vue de l'utilisateur dans la collection. Il s'agit du *SpaceMouse Plus*, un dispositif isométrique à six degrés de liberté (DDL) produit par *3D Connexion*.

Dans le prototype, la force appliquée à ce dispositif permet de contrôler la vitesse à laquelle se déplace la caméra virtuelle dans la scène, c'est-à-dire le point de vue de l'utilisateur. Ce choix s'appuie sur l'étude de Zhai et al. (1997), selon laquelle les dispositifs isométriques seraient les plus appropriés pour contrôler la vitesse d'un paramètre visuel. Afin de faciliter la navigation, les mouvements ont été limités : l'utilisateur dispose des translations le long des trois axes mais il contrôle uniquement la rotation autour de l'axe vertical. Ces limitations sont compatibles avec la topologie de la scène et elles se sont montrées nécessaires pour minimiser les mouvements inutiles lors des essais réalisés. Par exemple, la rotation autour de l'axe Z n'apporte rien et, en plus, rendrait moins lisibles les images des reliures. De plus, le déplacement de la caméra est restreint au volume interne du cylindre dans lequel se trouve la collection. Le mouvement de rotation autour de l'axe vertical permet à l'utilisateur, lorsqu'il est proche et ne voit qu'une partie du catalogue, de percevoir très rapidement tous les documents qui l'entourent (voir fig. 6.7).

### Implémentation du premier prototype

L'application a été développée et essayée sur un PC (*Pentium 4*, 2,4 GHz, 512 RAM) avec *Microsoft Windows XP* et doté d'une carte graphique *NVIDIA Quadro FX3000*. Elle a été écrite en C, *OpenGL*. Outre la bibliothèque *SPI*, la bibliothèque *GLUT* (*OpenGL Utility Toolkit*)<sup>9</sup> a été utilisée pour la gestion du fenêtrage et du *Space-*

<sup>9</sup>J'ai utilisé une version de *GLUT* légèrement modifiée qui prend en compte les entrées du *Space-Mouse*. Cette version est disponible sur : <http://www.doc.ic.ac.uk/~ajchung/CodeHacks/>

*Mouse.*

### 6.3.4 Affichage final et qualité visuelle

Le principal inconvénient rencontré lors de l'utilisation de ce premier prototype a été la résolution des textures des livres. La restitution finale des images des textures était très dégradée. Cela est dû aux filtrages successifs subis par les images originales.

Une image de tranche plaquée au départ sur un objet géométrique dans la scène est premièrement transformée lorsqu'elle est projetée sur le *viewport* correspondant à un des quatre canaux. Ce premier rendu, utilisé en tant que texture, est plaqué sur le maillage déformé. Les pixels de la texture sont à nouveau filtrés vers le *viewport* final. Au bout de ces deux transformations, des effets de « crénelage » s'accumulent. De plus, la définition du projecteur (1024 x 768 pixels) est assez limitée pour l'affichage de certains éléments graphiques plus fins.

On peut facilement discerner les pixels projetés (et même un léger décalage entre les composantes RGB des pixels qui se trouvent dans la périphérie de la zone focale du projecteur). Cette qualité d'affichage est surtout satisfaisante pour des applications affichant des images animées en mouvement constant, comme c'est le cas des simulateurs de vol, des jeux vidéo, etc. Néanmoins, lors d'un travail de recherche, appréciation ou lecture de documents textuels ou graphiques, la qualité de l'image perçue devient un critère essentiel pour le confort et la bonne compréhension des informations. La texture n'est plus utilisée comme un élément de décor pour la géométrie de la scène. Elle est, au contraire, l'élément principal de l'environnement.<sup>10</sup> L'importance de la qualité visuelle de l'image restituée pour une « tâche cognitive » est montrée par l'étude de Muter et Maurutto (1991). Les résultats de ce travail suggèrent que la vitesse de lecture sur un écran d'ordinateur est proportionnelle à la résolution de l'image affichée. C'est donc dans ce contexte de lecture et consultation de documents que la *VisionStation*, et une grande partie des dispositifs immersifs, atteint ses limites.

Deux techniques ont été adoptées afin de contourner ces limitations : pré-traitement des images et mise en place une configuration bi-écran.

#### *MIP mapping*

Concernant le « pré-traitement des textures », j'ai utilisé la technique de *MIP mapping* (Williams, 1983) afin de réduire le filtrage brutal réalisé au moment des transformations de la texture au *viewport*. Le *MIP mapping* est un pré-filtrage qui fournit pour chaque texture un jeu de versions réduites (qui s'appellent des *MIP maps*) de cette image. Au moment du filtrage, le rendu effectue l'échantillonnage à partir des *MIP maps* dont les dimensions sont les plus proches de celles de la taille

---

<sup>10</sup>De fait, c'est la géométrie de la scène qui devient un élément secondaire.



FIG. 6.7: Un utilisateur navigue dans le catalogue immersif projeté sur la *VisionS-tation*.

que la texture apparaît sur le *viewport*. Cela allège les calculs (en rendant le mouvement plus fluide) et réduit le bruit graphique des textures lorsque la caméra est en mouvement mais aussi lorsqu'elle est statique. Bien que cette technique n'améliore pas la résolution des images, elle a pu tout de même accroître sensiblement le confort visuel procuré par les textures restituées.

### 6.3.5 Une stratégie « focus plus contexte »

La deuxième stratégie a consisté à augmenter la finesse de l'espace d'affichage de l'application via le couplage d'un deuxième dispositif d'affichage. Cet écran auxiliaire est consacré à l'affichage d'une vue « zoomée » et en haute résolution des documents. Le travail de Baudisch et Good (2002), explorant une approche « focus plus contexte » pour une problématique similaire, a servi d'inspiration pour cette solution.

Pour l'affichage de cartes urbaines ou des images satellitaires sur une grande surface, la résolution du projecteur atteint ses limites (par exemple, les tracés des rues et ses noms deviennent illisibles). Afin de résoudre ce problème, Baudisch et Good proposent l'« augmentation » d'une zone de l'espace de projection à l'aide d'un deuxième écran. Dans ce travail, un écran plat normal est installé au centre de la surface projetée. Cet écran affiche une zone réduite de la même image qui est envoyée au projecteur par l'application. Enfin, un gestionnaire du tampon d'image fait en sorte que la zone de la surface où l'écran se trouve ne soit pas projetée. Le résultat est une surface d'affichage très large avec une zone de haute

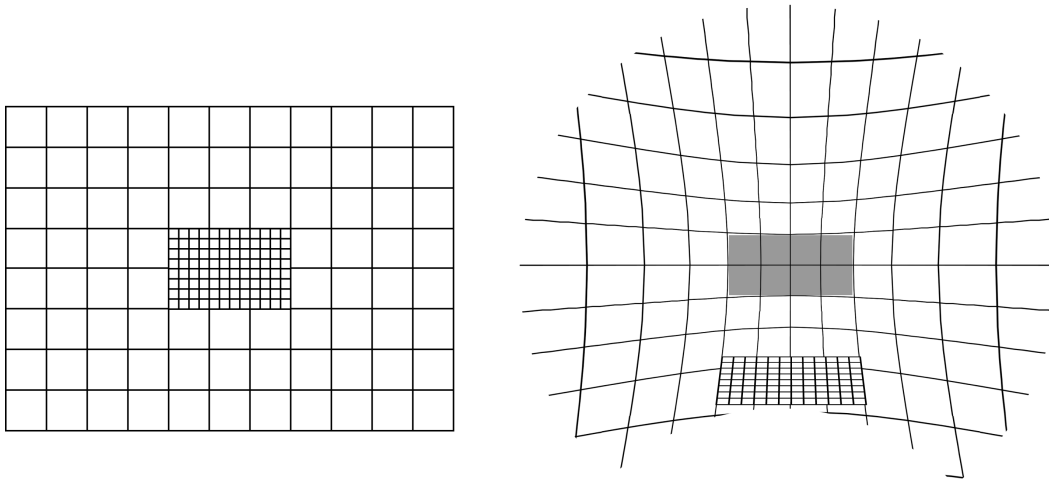


FIG. 6.8: A gauche, un schéma de la solution « focus plus contexte » présentée par Baudisch et Good (2002). A droite, le schéma d’une solution analogue utilisée pour l’affichage immersif. Les carrés en gris sur la surface parabolique indiquent la zone qui est affichée (en haute résolution) sur l’écran secondaire.

résolution au centre (voir fig. 6.8). L’utilisateur peut ainsi consulter un « focus » haute résolution dans l’image sans que ce focus soit coupé de son contexte « topologique ».

Fort de ces constats, un écran plat (TFT 17 pouces, 1280x1024) a été ajouté à la configuration immersive (voir fig. 6.7). Cet écran secondaire, procurant une résolution spatiale bien plus grande que celle de la *VisionStation*, est mis à plat sur la tablette du dispositif immersif, à côté du *SpaceMouse*. L’avantage d’une telle configuration est de conserver à la fois l’immersion de l’utilisateur dans la scène 3D et la possibilité de travailler sur une image en haute résolution.

L’écran auxiliaire affiche l’image que l’on voit au centre de la surface hémisphérique (fig. 6.8). Pour que la transition du regard entre la vue immersive et la vue détaillée soit la plus fluide possible, la zone de la scène 3D affichée sur l’écran plat est ajustée pour que les documents affichés sur cet écran paraissent avoir les mêmes dimensions de ceux présents dans la zone équivalente qui s’affiche sur la surface parabolique. Ainsi, une tranche de livre a une même hauteur et une même largeur sur l’une et sur l’autre, quoiqu’elle apparaisse fine sur la première et grossière sur la seconde (fig. 6.9).

### 6.3.6 Implémentation du deuxième prototype

Pour travailler dans une configuration bi-écran (les deux écrans sont pris en compte par la carte *NVIDIA Quadro FX3000*), deux contextes *OpenGL* ont été créés et associés à des fenêtres distinctes. Pour chaque contexte, des fonctions de projection différentes ont été spécifiées. Ces deux contextes se servent de la même

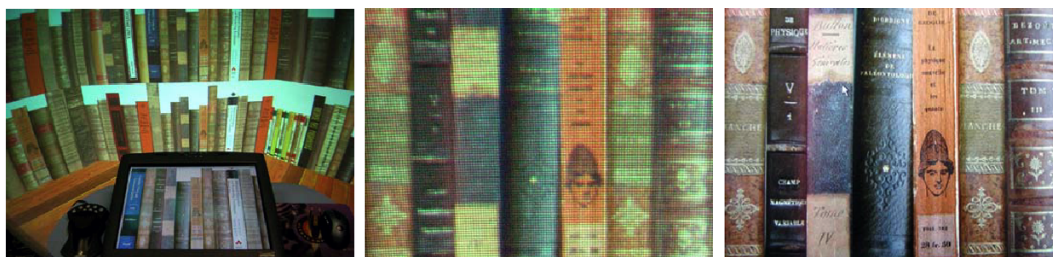


FIG. 6.9: De gauche à droite : photo de la configuration bi-écran, photo d'un détail central du catalogue de documents affiché sur la *VisionStation*, photo du même détail du catalogue affiché à ce même moment sur l'écran auxiliaire.

description de scène mais ils disposent chacun de sa caméra dotée d'un champ de vision à elle. La fenêtre destinée à la *VisionStation* utilise les fonctions d'affichage et de projection spécifiques au *SPI* alors que le contexte associé à l'écran auxiliaire utilise les fonctions de projection de la bibliothèque *GLU* (OpenGL Utility Library). L'équivalence de taille relative entre l'image affichée sur l'écran plat et la zone correspondante affichée sur la *VisionStation* est faite en ajustant le champ de vision de la caméra du contexte « détail ». Le résultat a été satisfaisant lorsque le champ de vision de la deuxième caméra a été configuré à 30°.

## 6.4 Conclusion et travaux futurs

Cette première étude sur l'utilisation de périphériques d'affichage immersifs indique de nouvelles opportunités pour la visualisation et pour l'interaction dans les espaces de documents. Parfaitement adapté au rendu des scènes 3D en mouvement, ce dispositif doit cependant être complété sur différents aspects. Tout d'abord, il me semble nécessaire d'étudier les autres bénéfices dans l'interaction que la configuration bi-écran peut apporter à l'utilisateur.

Dans la continuation de ce travail, il me semble intéressant que l'écran auxiliaire soit testée comme un moyen pour permettre à l'utilisateur de réaliser d'autres tâches. Par exemple, sélectionner un document ou le consulter. De fait, s'agissant d'un écran tactile, l'utilisateur pourrait contrôler la caméra avec la main gauche sur le *SpaceMouse* et cliquer directement sur la tranche qui l'intéresse avec sa main droite.

Par ailleurs, outre l'augmentation de la résolution visuelle, la « résolution sémantique » de cette vue dupliquée peut être également augmentée. Cet affichage sémantique pourrait fournir des informations supplémentaires à celles disponibles sur les reliures du livres, comme le nom de l'auteur et l'année d'édition (issues des métadonnées des documents). Le fait que la vue haute résolution soit spatialement dissociée de la vue globale constitue ici un avantage puisque des couches d'informations complémentaires (ou voire même, d'autres vues) peuvent être attribuées

à cet affichage auxiliaire. Dans la technique de Baudisch et Good (2002), comme les deux vues sont superposées, il ne serait pas possible d'augmenter uniquement l'une d'elles.

Enfin, cet écran secondaire pourra évidemment être utilisé pour la lecture du document sélectionné. Il afficherait ainsi les pages du document et on pourrait le feuilleter directement en touchant l'écran. Des métaphores d'interaction faisant appel à un vocabulaire gestuel plus naturel pourraient faciliter la découverte et la lecture de ces documents numérisés (O'Hara et Sellen, 1997). On pourrait ainsi passer d'un contexte à l'autre, du contexte de parcours d'une collection à celui de lecture d'un document, d'une façon très naturelle et intuitive (en levant ou en baissant sa tête).

A l'instar de *Google Books*, qui génère automatiquement des couvertures pour les livres dont la couverture n'a pas été numérisée, il serait envisageable de développer ou d'adapter un moteur pour la génération *stylisée* d'images de reliures de livres. Etant donné que nombreux documents numériques ne disposent pas de reliures, ces « fausses » reliures pourraient néanmoins communiquer des aspects pertinents de ces documents, à savoir le nombre de pages, le format physique du document, la date d'édition et le type de document.



# Techniques de visualisation pour le contrôle qualité de lots d'images numérisées

*Plusieurs milliers d'images issues de programmes de numérisation massive sont intégrées tous les mois dans les fonds des bibliothèques numériques. Ce chapitre traite de la problématique de la visualisation pour le contrôle qualité de ces grands lots d'images. Quelques caractéristiques de cette activité sont présentées, en soulignant les difficultés relatives à l'utilisation d'applications traditionnelles de visualisation dans ce contexte. Par ailleurs, ce chapitre présente quatre explorations réalisées durant cette thèse qui permettent de consulter les grands ensembles de fac-similés de pages : une expérience graphique de visualisation globale à partir de la réduction extrême de toutes les pages d'une collection de volumes, une interface Web de vérification visuelle utilisée dans une vraie bibliothèque numérique et deux techniques de visualisation zoomables permettant de naviguer de manière fluide entre les vues globale et détaillée d'un ensemble de pages numérisées.*

Les coûts des services de numérisation de documents papier et du matériel associé ont considérablement baissé ces dernières années. Ce contexte, qui comprend aussi la maturation des technologies utilisées dans le processus de numérisation, favorise l'émergence d'un bon nombre de projets de numérisation de documents papier. Dans ces bibliothèques numériques, les volumes sont souvent numérisés par des prestataires extérieurs. La numérisation est entreprise loin du personnel qui connaît les particularités de chaque fonds (bibliothécaires, conservateurs et spécialistes). Chez les sous-traitants qui effectuent la numérisation, un opérateur peut atteindre la vitesse de 700 pages par heure. La productivité de ce processus varie, entre autres facteurs, en fonction de l'homogénéité du lot. Par ailleurs,

les volumes anciens, comme c'est le cas des documents numérisés par les bibliothèques numériques patrimoniales, requièrent un traitement plus individuel en raison de leurs particularités (planches dépliantes, pages abîmées, reliure ne permettant qu'une ouverture partielle du volume). De fait, la dissimilitude entre les volumes et les particularités de chacun imposent un plus grand soin et un plus grand nombre d'ajustements de la configuration du scanneur.

Dans ces conditions, la productivité est inversement associée au nombre d'interventions humaines nécessaires à la numérisation d'un lot. Cela fait que les volumes de tailles régulières, peu déformés, sans planches et dont les pages ne sont pas collées les unes aux autres soient préférés à d'autres volumes (possédant moins de particularités) dans les programmes de numérisation massive, comme *Google Books*<sup>1</sup> ou *Gallica*<sup>2</sup>.

Le débit important des pipe-lines de numérisation et le fait que ces programmes soient peu préparés pour gérer les particularités de certains volumes imposent la nécessité d'un contrôle robuste de la qualité des images numérisées. La partie du contrôle qualité qui concerne cette thèse est celle de la vérification visuelle. Une des pistes explorées consiste à évaluer l'apport de techniques récentes de visualisation d'information pour les interfaces de contrôle qualité. Les logiciels actuellement en usage chez les sous-traitants sont basés sur les techniques usuelles d'association de vignettes et de vues de page en gros plan (comme le logiciel *ACDSee*<sup>3</sup>). Ces applications, conçues selon la logique de la visualisation de photos personnelles et n'étant pas destinées à la visualisation de milliers d'images, n'offrent que peu de modalités d'affichage prenant en compte les particularités graphiques des pages des documents.

Ce chapitre présente d'abord la problématique du contrôle qualité visuel de pages numérisées. Ensuite, quelques techniques de visualisation seront décrites. Nous verrons quels aspects de ces interfaces sont peu convenables à la vérification d'un grand nombre de fac-similés de pages. Enfin, ce chapitre rapportera quatre techniques exploratoires de visualisation conçues pour faciliter la visualisation exhaustive de toutes les pages d'un lot de documents numérisés. Ces techniques permettent à la fois d'avoir une vue de l'ensemble du lot et de rapidement inspecter les détails de certaines de ses pages.

## 7.1 Contrôle qualité des pages numérisées

Lorsque les images sont livrées par les prestataires de numérisation, on peut identifier différents types d'erreur ; quelques-unes sont assez imprévues. La numérisation de certains ouvrages revêt une importance parfois grande. De fait, lorsqu'on numérise des ouvrages rares et difficilement accessibles ailleurs, le contrôle

---

<sup>1</sup>*Google Books* : <http://books.google.com>

<sup>2</sup>Site Web de la bibliothèque numérique Gallica : <http://gallica.bnf.fr/>

<sup>3</sup>*ACDSee* : <http://fr.acdsee.com/>

Phase	Type de problème
avant	pages manquantes pages abîmées (pliés, déchirées, tachées) mauvais volume (mauvaises informations sur la reliure)
durant	volume peu ouvert volume mal orienté pages mal tournées planches non dépliées doigts ou d'autres objets couvrant la page images floues bordures du <i>setup</i> trop larges artefacts numériques variation de la couleur parmi les pages d'un même volume illisibilité des caractères / dessins les plus petits
après	images « binarisées » trop foncées ou trop claires mélange d'images issues de volumes différents images répétées images manquantes incohérence entre le n° de la page imprimé et celui déclaré dans le fichier incohérence entre le n° du vol. imprimé et celui déclaré dans le répertoire

TAB. 7.1: Type de problème de numérisation en fonction du moment où ils apparaissent.

qualité devrait mériter encore plus d'attention afin que les documents numérisés soient suffisamment fiables pour être consultés à la place des originaux. Certains lecteurs trouveraient peu utilisables des documents numérisés dont quelques pages ou planches seraient absentes puisqu'il se peut que ces éléments soient essentiels pour la compréhension du document. La phase de contrôle qualité est donc indispensable pour repérer des images manquantes, des pages absentes du volume original (et non détectées avant l'envoi du volume), des prises de vue inexploitables, etc. Cette phase de contrôle occupe de fait une place importante dans les chaînes de traitement des projets de numérisation (University of Michigan Library, 2009).

Certains aspects du contrôle qualité peuvent être validés de façon automatique ou quasi-automatique. Par exemple, les métadonnées de chaque volume, les noms donnés aux fichiers image, l'intégrité de ces fichiers, la résolution des images, etc. (Riley et Whitsel, 2005). Il est cependant recommandable de procéder à un contrôle visuel des images livrées (Riley et Whitsel, 2005). Lors du contrôle visuel, on cherche des erreurs qui ne sont pas identifiables par les procédures automatiques. Ces erreurs peuvent être classées selon le moment où elles sont survenues : avant, durant ou après la prise de vue. Le tableau 7.1 présente une liste non-exhaustive des problèmes que l'on peut trouver lors d'un contrôle visuel des pages numérisées.

Quelques-unes des erreurs décrites dans ce tableau - pages répétées, incohérence entre le numéro imprimé sur la page et le numéro dans le nom du fichier, page mal orientée - pourraient éventuellement être détectées avec l'aide de techniques de vision par ordinateur. Cette approche automatique est plutôt complé-

mentaire de l'approche visuelle. Tandis que la première approche permet de détecter des erreurs « modélisables », un avantage clair de la seconde est de permettre de déceler des erreurs qui n'étaient pas « prévues » ou qui sont plus facilement détectées par l'œil humain.

Dans un contrôle qualité traditionnel, un échantillon de pages numérisées est examiné et on cherche des problèmes récurrents. Les vignettes sont rapidement balayées et quelques-unes sont chargées en taille normale. Par ailleurs, dans un pipe-line de numérisation non-massive, le document original peut être facilement accessible par le personnel de la bibliothèque numérique et peut être confronté à son équivalent numérique. En revanche, dans un programme de masse, des connaissances sur le document numérisé, ou en cours de numérisation, sont réduites, voire inexistantes, à l'exception de celles explicitement déclarées dans le cahier des charges. De même la comparaison des images numérisées avec l'original physique est infaisable.

Ainsi, pour compenser le traitement accéléré dispensé aux documents originaux, les programmes de numérisation massive devraient adopter une stratégie de *visualisation exhaustive*, et non pas *par échantillon*. L'opérateur du contrôle qualité, ou le « bibliothécaire numérique » responsable pour la validation du lot, devrait pouvoir rapidement balayer *toutes les images* et pouvoir passer sans effort à une vue détaillée des pages qui attirent son attention.

## 7.2 Vue globale et vue détaillée

Les applications consacrées à la visualisation d'images sont principalement conçues pour la gestion et la visualisation de photos personnelles, plutôt que pour la consultation de pages numérisées. L'utilisation des pixels de la fenêtre n'est pas optimale et passer d'une image à l'autre prend du temps lorsqu'on souhaite en voir un grand nombre. On accorde plus d'importance à la notion d'élément individuel qu'à la notion d'ensemble d'éléments disposant de traits en commun (comme c'est le cas des pages numérisées issues d'un même volume). Or, dans la visualisation de pages, l'important est de voir chaque image dans son « contexte » - c'est-à-dire entourée par les pages « voisines » et à côté des pages issues des volumes d'un même lot - et de pouvoir en visualiser un grand nombre avec un minimum d'encombrement.

Pour mieux comprendre les principales caractéristiques de ce type d'exploration visuelle, il est intéressant de voir la taxonomie des tâches de navigation et de visualisation d'images de grandes dimensions proposée par Plaisant et al. (1995). Dans cette classification, le contrôle qualité des images numérisées serait une tâche de « diagnostic », similaire à l'exploration réalisée par un pathologiste qui inspecte en détail les photos en haute définition d'un tissu. La vitesse et la facilité avec lesquelles le zoom et le panorama peuvent être réalisés sont essentielles

pour que l'utilisateur repère des indices visuels pertinents.

Dans le contexte de l'étude de ce chapitre, l'élément graphique de base peut être considéré comme étant soit la page, soit *toutes les pages qui font partie d'un ensemble cohérent* (par exemple, les pages d'un même volume). Ces pages, lorsqu'elles sont affichées les unes à côté des autres, constituent un espace graphique de grandes dimensions. Chaque page peut être vue comme étant une zone de focus, c'est-à-dire que la page est le détail d'une image plus grande - qui serait formée par l'assemblage de toutes les pages.

Nombreux travaux se sont consacrés à la problématique de la configuration des vues du contexte et du détail et de la transition entre elles lorsque l'espace graphique est doté de dimensions très importantes. Cette section n'a pas la prétention d'offrir un état de l'art de cette problématique mais uniquement d'évoquer quelques concepts et travaux du domaine qui aident à situer la problématique de ce chapitre.

Parmi les techniques de visualisation, Leung et Apperley (1994) distinguent celles destinées à visualiser des données nativement graphiques de celles destinées à visualiser d'autres types de données (sonores, quantitatives, etc.). C'est la première classe de techniques qui nous intéresse ici. Dans ce premier groupe, les techniques peuvent offrir une vue *avec distorsion* (pour que le contexte graphique du focus soit représenté) ou une vue *sans distorsion* (dans ce cas, une zone de l'image est affichée sans que les zones qui l'entourent soient simultanément représentées).

**Vues avec distorsion.** Furnas (2006) explique que les approches par distorsion conservent la continuité topologique de l'espace visuel mais introduisent des déformations qui nuisent à la compréhension visuelle. Cette nuisance est encore plus importante dans le cas du contrôle qualité puisque la distorsion de l'image risque d'être confondue avec les anomalies issues de la prise de vue de la page. D'autre part, les approches à plusieurs *viewports* n'introduisent pas de distorsion dans l'espace visuel ; en revanche, elles ne préservent pas sa continuité topologique. Il existe aussi les approches « mono-focales » zoomables ; ceux-ci distribuent l'information dans le temps : on doit se rappeler que le détail que l'on voit à un moment donné vient de telle zone de l'image qu'on voyait à un moment précédent.

D'une manière générale, plusieurs techniques de visualisation par distorsion s'inspirent des « vues fish-eye » (Furnas, 1986). Dans sa forme abstraite, il s'agit d'un filtre qui supprime des données en fonction de leurs valeurs *a priori* et de leurs *distances* par rapport à un *point de focus*. Appliquée à un espace graphique, cette vue distord, par un rétrécissement, toute la zone de l'espace entourant le focus. Cette distorsion peut se présenter graphiquement de façon très variée en fonction des paramètres appliqués au filtre. Par exemple, le *Perspective Wall* est une technique qui distord l'axe horizontal de l'espace graphique d'un document (MacKinlay et al., 1991). Uniquement la partie centrale de la vue n'est pas affichée en

perspective ; dans les latérales gauche et droite de l'écran, le document rétrécit progressivement.

Dans la suite du *Perspective Wall*, Robertson et Mackinlay (1993) ont proposé la technique *Document Lens*, destinée à la visualisation de documents présentés sur plusieurs pages. Les pages sont affichées en échelle réduite, distribuées dans une matrice et l'utilisateur dispose ainsi d'une vue globale du document. Cela lui permet de comprendre la structure et d'identifier des éventuelles liaisons entre les différentes pages. L'utilisateur peut voir de près une zone du document grâce à une « lentille » rectangulaire qui peut être déplacée sur la fenêtre. La zone périphérique à la lentille est affichée en perspective, comme le *Perspective Wall*, afin que toutes les pages du document restent visibles. Toutefois, contrairement au *Perspective Wall*, le *Document Lens* distord la zone de contexte sur les deux dimensions de l'espace de la fenêtre.

Une technique plus récente qui se sert de la distorsion par perspective est *Mélange* (Elmqvist et al., 2008). Il s'agit d'une visualisation « multi-focale » d'un grand espace d'informations en rétrécissant progressivement la périphérie de chaque focus. C'est l'utilisateur qui choisit les *foci* et le niveau de zoom associé à chacune de ces zones. On peut ainsi visualiser plusieurs détails de l'espace graphique ; le contexte de chaque zone de focus est préservé.

**Vues sans distorsion.** Les approches *split-screen* (voir Elmqvist et al., 2008) peuvent aussi être considérées comme étant « multi-focales » puisqu'elles affichent plus d'un *viewport* par fenêtre. Chaque *viewport* se comporte comme une fenêtre pouvant être indépendamment défilé ou zoomé.

Lorsque l'interface ne présente qu'un *viewport* principal et ne distord pas l'image affichée, l'utilisateur devra se servir d'opérations de zoom et de panorama (c'est-à-dire défilement de l'image pour que la partie qui intéresse l'utilisateur soit dans le *viewport*). A ce propos, Furnas et Bederson (1995) suggèrent qu'utiliser le zoom et le panorama peut être plus efficace pour la *navigation* dans un grand espace graphique que d'utiliser seulement le panorama. Cet avantage vient du fait que le panorama associé au zoom permet à l'utilisateur de réaliser des trajectoires plus « courtes »<sup>4</sup> que s'il se servait uniquement de mouvements panoramiques restant toujours au même niveau de zoom.

**Visualisation de vignettes.** La consultation d'un ensemble de pages ou de photos à l'aide des vignettes de ces images existe dans un bon nombre d'applications bien qu'elle soit souvent présentée comme un mode de consultation auxiliaire.<sup>5</sup> Dans ce type d'interface, la priorité est donnée à la présentation de l'image dans

---

<sup>4</sup>Furnas et Bederson proposent un moyen pour évaluer la longueur des trajectoires dans un espace multi-échelle.

<sup>5</sup>Par exemple, l'interface du *Microsoft Windows File Explorer*, le « chemin de fer » du logiciel *Book Restorer* ou le site de partage de photos *Flickr*.

sa totalité.

L'interface *Space-filling Thumbnails* (SFT) essaie d'exploiter davantage l'affichage de vignettes (miniatures des pages) afin de procurer une vue d'ensemble d'un document (Cockburn et al., 2006). Cette interface présente toutes les pages d'un document distribuées de façon matricielle où chaque page a une position fixe par rapport aux pages voisines. La matrice a toujours la même configuration (la réduction de la fenêtre réduit la taille de toutes les cellules sans changer le nombre de lignes et de colonnes), ce qui permet à l'utilisateur de mémoriser la position relative d'une vignette et de la revisiter facilement. Par ailleurs, cette interface ne dispose d'aucun mécanisme de défilement. De ce fait, toutes les pages (d'un même document) sont toujours simultanément visibles sur la fenêtre. Pour cela, la taille de chaque vignette décroît à mesure que le nombre total de vignettes s'accroît. Le désavantage de ce changement d'échelle automatique est le fait que les vignettes deviennent microscopiques lorsque le volume est doté d'un grand nombre de pages.

Outre la visualisation globale via un affichage « miniaturisé » des pages, la possibilité de consulter une des page en haute résolution est une fonctionnalité également importante. Certains problèmes (comme la lisibilité des caractères du texte ou la netteté des illustrations) ne sont identifiables que lorsqu'on visualise l'image en « taille réelle ». Plusieurs mécanismes de zoom existent (Plaisant et al., 1995) : le zoom peut agrandir toute la page qui occupera toute la fenêtre (comme dans *Adobe Photoshop*), il peut agrandir toutes les photos présentées dans la fenêtre en même temps (comme dans *Picasa*) ou il peut agrandir temporairement une seule photo en conservant en fond les autres images (comme le zoom contextuel de SFT). La transition entre les différents niveaux d'échelle peut aussi être, soit discrète (comme dans *Adobe Acrobat Reader*), soit continue (comme dans *Photomesa* ou *Picasa*).

### 7.3 Visualisation massive statique d'une série de volumes

Une première exploration concernant les apports de la visualisation par vignettes a été la fabrication de miniatures des pages des volumes de « La Nature »<sup>6</sup>. Cette revue est l'un des documents parmi les plus visités de la bibliothèque numérique du CNAM (CNUM) et l'un des plus volumineux. Il compte 32 500 pages, avec environ une gravure toutes les deux pages ; actuellement 32 années (soit 1 696 fascicules reliés dans 65 volumes) de cette publication sont disponibles en ligne. À l'aide de la bibliothèque de programmes NetPBM<sup>7</sup>, les miniatures ont été produites, pour chaque page de tous les volumes de cette collection et un script

---

<sup>6</sup><http://cnum.cnam.fr/redirect?4KY28>

<sup>7</sup><http://netpbm.sourceforge.net/>

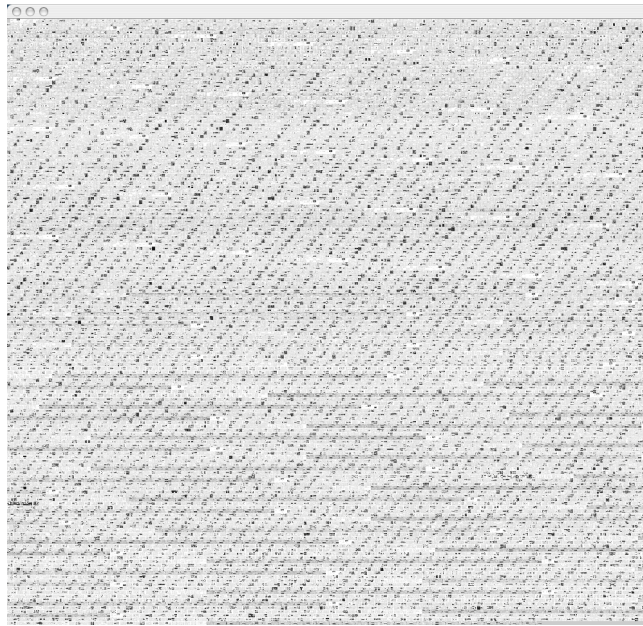


FIG. 7.1: Les 32 500 pages (5 x 8 pix. chacune) de La Nature affichées sur une même image (1000 x 965 pix.).

*Processing*<sup>8</sup> les a toutes affichées sur une seule fenêtre.

Dans une première version - présentée dans la fig. 7.1 - les vignettes sont affichées côte à côte sur plusieurs lignes. Cette première configuration permet d'avoir un aperçu sur le lot : la proportion entre le nombre de pages de texte et d'illustration, le nombre de pages blanches. Une deuxième configuration graphique est présentée dans la fig. 7.2 où les pages sont disposées autrement : les pages d'un même volume apparaissent toutes sur une même ligne. Cette configuration fournit des pistes visuelles qui passent inaperçues dans la première configuration. On peut, par exemple, observer : la longueur relative des volumes les uns par rapport aux autres, les pages typiques qui apparaissent en fin de volume et celles qui apparaissent en début de volume, la régularité des pages illustrées, si un volume est plus illustré que les autres.<sup>9</sup> Toutes ces indications sont utiles pour repérer des lacunes dans les volumes, qu'elles soient dues à la numérisation ou aux volumes physiques fournis à la numérisation.

Un des points forts de cette visualisation statique est d'offrir une vue globale sur une collection numérisée, principalement pour vérifier son intégrité et comprendre quels sont les types de page qui la composent. La visualisation dynamique d'un ensemble d'une telle grandeur, proche du giga-pixel, est actuellement hors de portée de la puissance de calcul des ordinateurs standard. Les techniques présentées dans les prochaines sections ont exploré des pistes interactives bien

---

<sup>8</sup><http://processing.org/>

<sup>9</sup>Grâce à cet affichage, nous avons pu découvrir que, dans le volume 34, les suppléments qui apparaissent en fin du volume sont visiblement manquants (voir détail de la fig. 7.2).



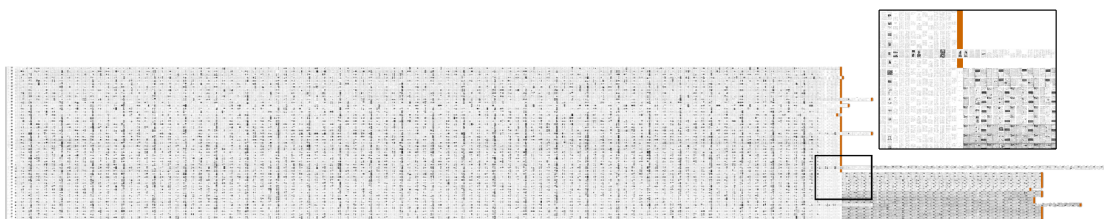


FIG. 7.2: Les 32.500 pages (10 x 16 pix. chacune) de La Nature affichées sur une même image (8000 x 600 pix.). Chaque ligne correspond à un volume de fascicules reliés. Détail sur la fin des volumes dans les années 1890.

que sur des ensembles de pages plus réduits.

## 7.4 Interface Grille de détails

La « Grille de détails » est un outil développé pour la bibliothèque numérique du CNAM dont le but est de faciliter la *vérification exhaustive* de certaines caractéristiques des pages des documents numérisés. Il s'agit d'une interface Web qui affiche les vignettes des pages dans une disposition matricielle.

Dans cette interface, hormis l'espace d'un pixel entre les vignettes, tous les pixels de la fenêtre sont utilisés pour afficher les pages numérisées. Des aspects visuels de toutes les pages d'un volume de mille pages peuvent ainsi être vérifiés en quelques minutes. En outre, les vignettes sont partiellement recoupées afin de prioriser la zone de la page contenant du texte ou de l'illustration. L'affichage résultant est donc *dense* en informations visuelles. Enfin, le fait que les pages soient serrées les unes contre les autres, sans beaucoup d'espace entre elles, facilite la détection d'anomalités. Par exemple, une page avec du texte plus foncé que le texte des pages voisines saute aux yeux. Les paragraphes suivants décrivent la façon dont les images numérisées sont prises en compte.

La numérisation au CNUM n'est pas faite en interne mais par des prestataires de numérisation. Ces prestataires livrent deux types d'images master, tous les deux en 400 DPI : un master brut et un master « binarisé ». Le premier est livré sous le format PNG sans compression, 256 niveaux de gris (fig. 7.3). Il présente l'image sans recadrage, correction géométrique ou séparation des pages gauche et droite (dans le cas où elles aient été photographiées ensemble). Le second est livré sous le format TIFF compressé groupe 4, noir et blanc, avec les pages séparées, recadrée et après une correction de courbure. Les illustrations des pages ou des planches hors texte (souvent des gravures à traits fins) subissent un processus de tramage pour minimiser la perte d'informations au moment de la « binarisation ». Les PNG, bien plus lourds que les TIFF, sont stockés en tant que copies de préservation tandis que les TIFF sont utilisés pour générer les images finales diffusées au CNUM.

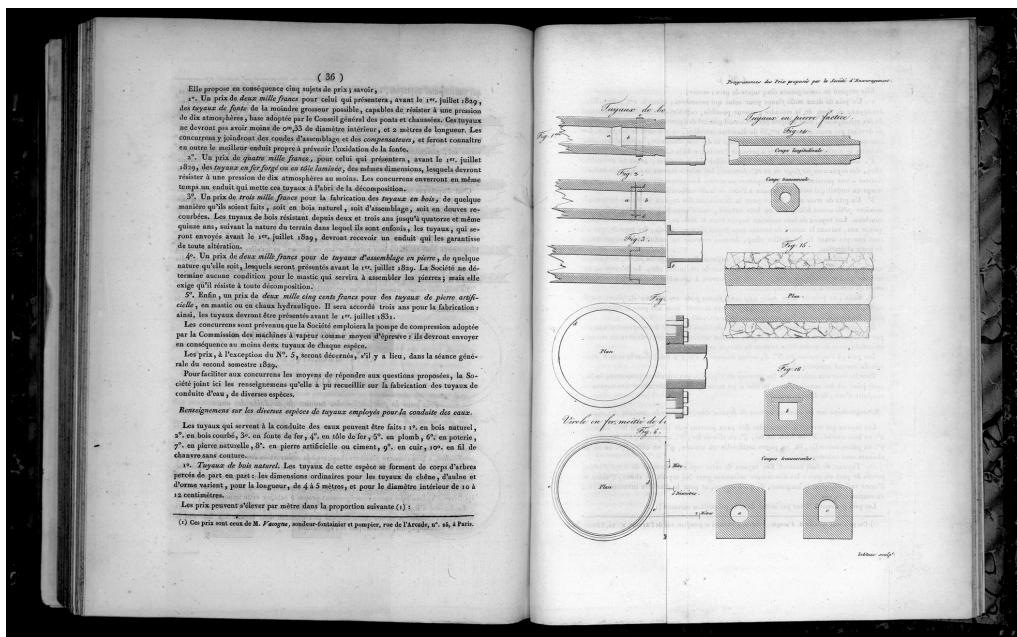


FIG. 7.3: Exemple d'une image master issue d'un volume numérisé par un pres-tataire du CNUM. A gauche, une page de texte, à droite une planche hors-texte pliée. Cette image (PNG, 400 DPI, 256 niveaux de gris) fait 7 800 x 5 000 pixels et pèse 20 Mo. (Droit d'utilisation de l'image gentiment cédé par la Bibliothèque numérique du CNAM (<http://cnum.cnam.fr>)).

**Fabrication des vignettes** Dans la *Grille de détails*, les vignettes des pages sont générées au moment où l'on finit de charger les images dans le serveur du CNUM. Deux types de vignette sont fabriqués à partir des images master : une miniature de la page et un détail du centre de la page. Ce détail a la même résolution que l'image qui sera diffusée sur le site. Pour la miniature, on recoupe un carré dont le côté fait la dimension la plus courte de la page et on l'échantillonne ensuite à 128 x 128 pixels. Pour le détail, on échantillonne d'abord l'image à la résolution du format de diffusion ; ensuite on recoupe le centre de cette image échantillonnée, un carré de 128 x 128 pixels. Pour la miniature des images brutes, s'agissant des prises de vue des deux pages, les vignettes font 256 x 128 pixels (ces images sont recadrées le moins possible). Ces vignettes sont fabriquées par lot à l'aide des programmes de la bibliothèque *NetPBM*.

### 7.4.1 Interface Web

La *Grille de détails* affiche dans une page générée par du code PHP dont des scripts JavaScript offrent quelques micro interactions. Trois modes de visualisation ont déjà été testés et sont couramment utilisés :

**Inspection des images de diffusion.** Ce mode de visualisation est actuellement utilisé dans la vérification des images master TIFF. Des vignettes de 128 x 128 pixels

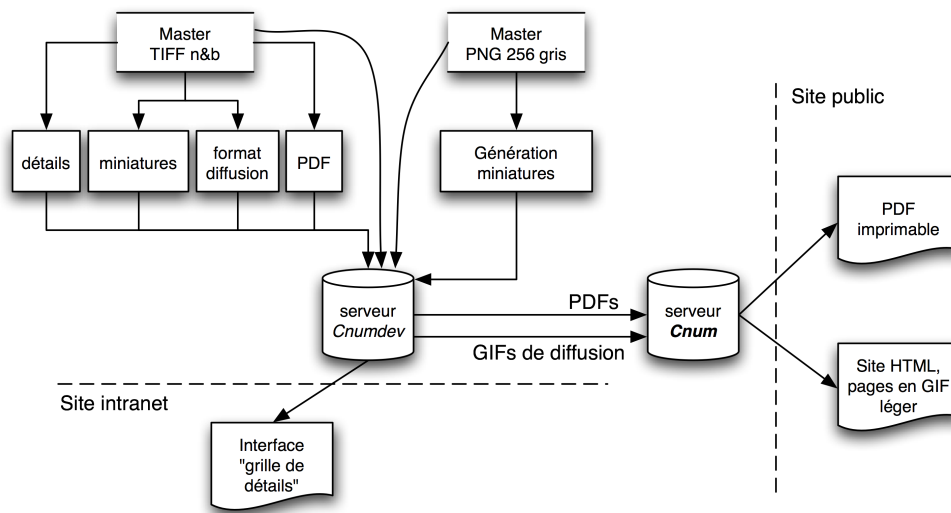


FIG. 7.4: Formats des images et étapes du traitement dans le CNUM.

sont affichées dans une matrice qui se sert de tout l'espace de la fenêtre. L'utilisateur décide s'il veut que, par défaut, toutes les vignettes affichent la vue « détaillée » ou la vue « miniature » de la page. Lorsqu'on passe le curseur sur une vignette, celle-ci bascule vers le mode de présentation qui n'est pas celui par défaut. En cliquant sur la vignette, on charge la page dans le format GIF de diffusion. La vue de détail permet de vérifier, par exemple, si le texte est lisible (si les caractères ne sont ni trop rongés ni trop petits) ou si le résultat du tramage des gravures est satisfaisant. Ce changement rapide et sans clic entre la vue miniature et la vue détail fonctionne comme un zoom contextuel et discret (d'un seul pas). Il permet de vérifier simultanément des indices visuels de tout un volume avant de charger les pages en taille réelle.

**Inspection des images de préservation.** Cette deuxième interface présente les miniatures des images brutes. Lorsqu'il s'agit de prises de vue des pages opposées, les vignettes font 256 x 128 pixels. Les images sont recadrées le moins possible (juste assez pour entrer dans un format 2 :1). La visualisation des images brutes fournit des pistes importantes. Par exemple, le fait d'afficher les pages impaires à côté des pages paires permet de voir si une page est manquante dans le volume original ou si on l'a « perdue » lors d'un premier traitement. Par ailleurs, les pages apparaissent dans l'ordre dans lequel elles ont été numérisées, ce qui facilite la vérification de l'emplacement des planches ou d'autres éléments sans pagination. Enfin, les tonalités de gris et d'autres éléments visuels, comme des traits ou des taches, communiquent une certaine homogénéité des pages et peuvent éclairer l'utilisateur sur des pages « étrangères » ayant été placées par erreur dans le répertoire du volume.

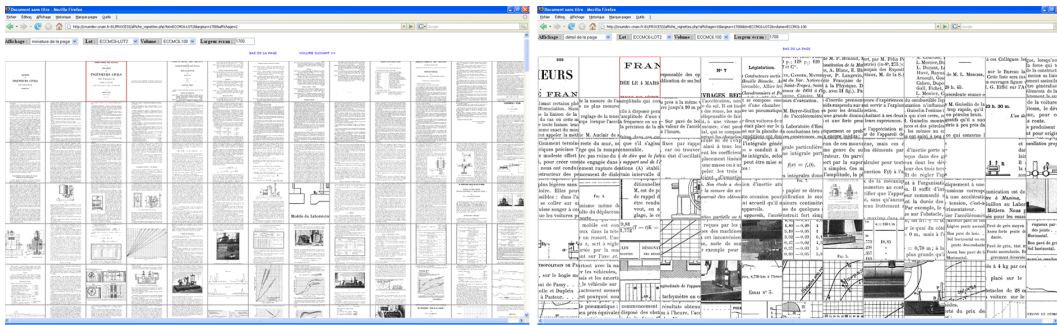


FIG. 7.5: Copies d'écran (fenêtres 1600 x 900 pix.) de l'interface de contrôle qualité « Grille de détails ». A gauche, les vignettes affichent les pages en miniature ; à droite on voit le détail du centre des pages.

**Comparaison « intra-lot »** Cette troisième méthode n'est pas faite pour vérifier les images elles-mêmes mais pour vérifier la cohérence entre les images des volumes appartenant à une même collection.<sup>10</sup> On se sert de mêmes vignettes (miniature et détail) utilisées dans la première interface et les images de chaque volume sont affichées sur une même ligne.<sup>11</sup> Ainsi, en balayant les lignes de haut en bas, on peut voir si des différences importantes entre les volumes (ou des similarités, indiquant des volumes répétés) sautent aux yeux. Cette présentation est spécialement intéressante dans la visualisation de revues (comme « La Nature ») qui utilisent une même structure au fil de plusieurs volumes.

Comme le montre le tableau 7.2, on peut considérer ces différentes options d'affichage comme des modalités d'échantillonnage auquel l'ensemble d'un volume est soumis. Etant donnée la difficulté pratique de visualiser tous les pixels de toutes les pages du volume, on échantillonne soit le *nombre de pages*, soit la *zone visualisée de chaque page*, soit la *résolution de la page*.

## 7.5 Mur de pages

La *Grille de détails* augmente le débit de la visualisation de pages mais elle ne fournit qu'un *niveau de zoom* pour chaque page. On sacrifie le paramétrage de la visualisation au profit de la simplicité d'interaction (ainsi qu'au profit d'une technologie Web facile à déployer). Lorsque l'utilisateur souhaite visualiser plus de détails, il va devoir charger l'image entière. Il s'agit donc d'une *interaction discrète* à trois pas : (1) miniature, (2) zoom serré et (3) chargement de l'image en taille réelle.

Pensant à ces limitations, j'ai développé un prototype qui pourrait restituer les détails de l'image de façon fluide et progressive. Comme dans la *Grille de détails*, les pages sont disposées de façon serrée dans une matrice. Grâce à cette disposition,

<sup>10</sup>On peut l'utiliser également pour visualiser les volumes d'un même lot.

<sup>11</sup>Comme il serait trop lourd d'afficher toutes les images de chaque volume (certains lots font 24 000 pages), juste les 15 premières vignettes de chaque volume sont affichées.

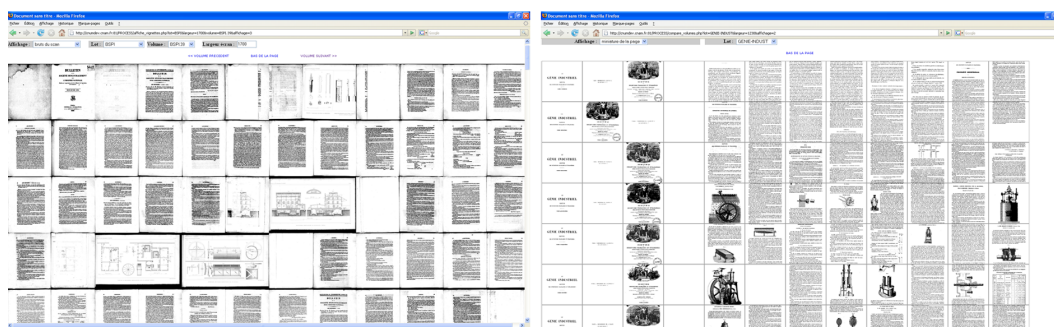


FIG. 7.6: Copies d'écran (fenêtres 1600 x 900 pix.) de l'interface de contrôle qualité « Grille de détails ». A gauche, les vignettes affichent les images master de chaque couple de pages ; à droite chaque ligne montre les premières pages d'un volume issu d'une même série.

l'ensemble de pages est perçu comme une grande image. Utilisant la technologie 3D, les pages sont disposées dans un environnement où le zoom et la navigation deviennent des opérations très naturelles. Le zoom en arrière réduit la grille progressivement jusqu'à ce que l'intégralité des pages soit visible en même temps. Par le zoom en avant, l'utilisateur peut rapidement trouver le niveau de détail qui lui convient. Il peut alors défiler les pages à une très haute vitesse grâce au dispositif isométrique utilisé pour contrôler la caméra.

### 7.5.1 Stratégies de navigation

Deux stratégies de navigation semblent être importantes dans le contexte d'une tâche de contrôle qualité : la « navigation global-détail » et le « balayage séquentiel ».

**Navigation global-détail.** Dans la *navigation global-détail*, l'utilisateur doit pouvoir visualiser la totalité des pages d'un coup d'œil. Cette vue globale est importante pour qu'il puisse estimer la proportion entre des pages de texte, d'illustrations et des planches. Cet affichage peut aussi permettre de voir s'il y a des pages dont les aspects visuels (niveaux de gris, taille des caractères, bordure) semblent différents de la majorité des pages. Puis, une fois identifiée des « zones du mur » qui méritent d'être observées avec plus d'attention, l'utilisateur doit pouvoir rapidement s'en approcher allant, si nécessaire, jusqu'à la trame du papier.

Type de de présentation	nombre de pages	zone observée	résolution
page entière	2%	100%	100%
miniature	100%	90%	10%
détail du centre	100%	10%	100%

TAB. 7.2: Résumé des caractéristiques de l'« échantillonnage visuel » procuré par chaque méthode de la Grille de détails.

**Balayage séquentiel.** Pour la stratégie *balayage séquentiel*, l'utilisateur doit pouvoir « balayer » toutes les pages à une résolution et à une vitesse qui lui conviennent. En voyant les pages dans l'ordre dans laquelle elles ont été numérisées, il pourra plus aisément avoir des pistes sur des pages manquantes (par exemple, un chapitre qui finit dans une page et la page suivante commence par du texte). De même, si on souhaite vérifier la qualité des illustrations dans un volume qui en possède beaucoup il sera plus facile de faire défiler toutes les pages plutôt que de faire des allers-retour de zoom sur chaque illustration. C'est pour cette raison aussi que ce défilement doit pouvoir se faire à une grande vitesse : pour que l'utilisateur puisse « sauter » les zones visuelles méritant peu d'attention.

### 7.5.2 Scène et contrôle de la caméra

Les pages sont plaquées sur la face interne d'un hémicylindre. Lorsque la caméra se trouve au centre de la scène, toutes les pages sont visibles : le rayon du cylindre est calculé en fonction du numéro de pages à afficher et de l'amplitude du champ de vision de la caméra virtuelle. Les limites du mouvement de la caméra : en zoomant, elle peut aller jusqu'au détail de la page ; en reculant, elle doit s'arrêter au centre du cylindre.

Cette disposition permet que les trajectoires rectilignes du centre du cylindre vers n'importe quelle page aient la même longueur. En revanche, si le mur était plat, il serait plus coûteux de déplacer la caméra jusqu'aux pages qui se trouvent dans un point extrême de cette surface.

**Navigation cylindrique.** La navigation d'un premier prototype du *Mur de pages* s'est inspirée de la caméra décrite dans le chapitre 6. Il était alors possible de contrôler la caméra dans quatre DDL, à savoir, les déplacements le long des trois axes et la rotation autour de l'axe vertical. Cependant, j'ai constaté que les utilisateurs n'arrivaient pas à maîtriser confortablement ces quatre DDL et cela les retardait d'atteindre les vues qui leur convenaient. Or, la rotation de la caméra autour de l'axe Y et son déplacement le long de l'axe X jouent, dans ce type de tâche, des rôles similaires. Dans cette configuration, la navigation horizontale a été restreinte à la rotation autour de l'axe vertical. Alors, bien que le déplacement le long de X et de Z soit interdit, toutes les zones du mur restent accessibles et visibles (dans une topologie convexe comme celle décrite).

Du fait que le déplacement de la caméra suit des coordonnées cylindriques, elle reste toujours parallèle au mur (voir figure 7.7). L'utilisateur voit ainsi les images sans aucune distorsion de perspective. De ce fait, contrairement à d'autres interfaces qui se servent des lignes de fuite pour montrer la « périphérie » d'un document (voir section 7.2 ou Guiard et al., 2006), la non-distorsion dans le *Mur de pages* réduit les chances que des erreurs de numérisation soient confondues avec des effets de perspective.

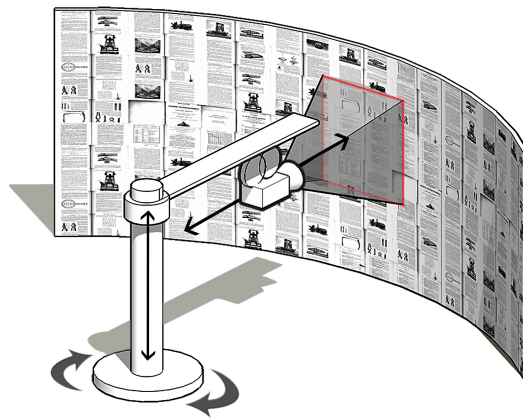


FIG. 7.7: Une métaphore décrivant les trois DDL de la caméra du *Mur de pages*.

(Illustration apparue dans Almeida et al. (2009), réalisée par Pedro Alessio.)

**Navigation sur l'axe vertical.** Dans le *Mur de pages*, les pages sont disposées sur deux dimensions pour tirer un maximum de profit de l'espace de la fenêtre. Puisque les pages sont ordonnées le long des rangées, parcourir l'axe vertical procure une « visualisation aléatoire » des pages, ce qui est intéressant dans le contexte du contrôle qualité. Par exemple, si l'utilisateur zoome sur la page 7 et monte la caméra, il verra passer la page 36, puis 65, 104 et ainsi de suite. Cette navigation lui permet de visualiser rapidement un échantillon de pages du volume.

**Balayage séquentiel sans rupture.** La longueur est l'axe privilégié par la disposition et elle est consacrée au *balayage séquentiel*. Son effet est similaire à l'action de défiler rapidement un document dans une application comme *Adobe Acrobat Reader*. Cependant, l'utilisateur doit passer à la ligne d'au dessus lorsqu'il atteint la fin de la rangée de pages où il se trouve. Ce mouvement ponctuel de monter la caméra d'une ligne introduit une interruption dans le balayage. C'est pourquoi j'ai considéré d'autres topologies de scène qui éviteraient cette opération.

L'idéal serait que l'utilisateur puisse parcourir toutes les pages avec un même geste pour que son action porte uniquement sur la vitesse du balayage et son attention se concentre sur l'identification des problèmes. Dans cette perspective, un cylindre complet, où les pages sont disposées de façon hélicoïdale, produirait une bande d'images continue de la première à la dernière page (Topol, 2002). Cependant, dans cette configuration il n'y a aucune point où la caméra puisse voir toute le scène (il y aurait toujours une partie du mur derrière la caméra). Par ailleurs, l'environnement serait trop homogène, sans points de repère, ce qui pourrait rendre plus difficile à l'utilisateur de retourner à des pages déjà visitées.

**Comportements pour passer d'une rangée à l'autre.** Deux comportements associés à la navigation pourraient combler cette lacune dans la scène. Un test d'ergonomie pourra montrer s'ils facilitent la navigation et lequel des deux offre une utilisation plus naturelle.

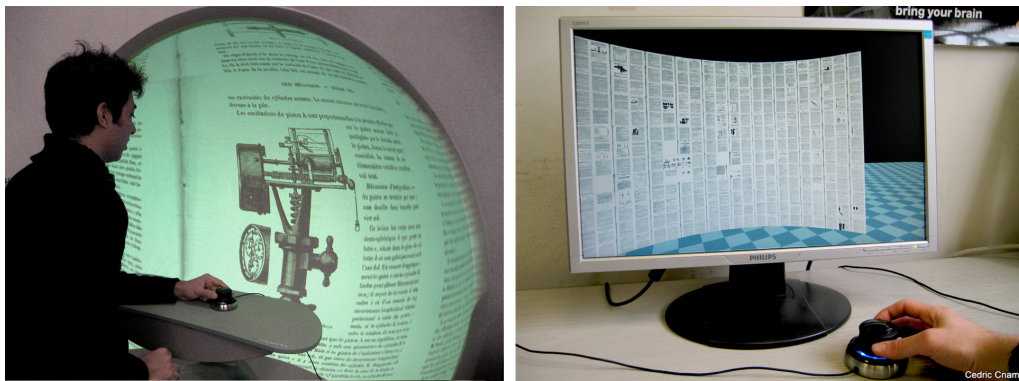


FIG. 7.8: Le Mur de pages dans deux configurations différentes. A gauche, l'interface affichée sur l'écran immersif *VisionStation* ; à droite, sur un écran 22 pouces.

Le comportement *monte-page* serait déclenché dans des conditions précises : en réalisant un mouvement panoramique à la proximité du *Mur de pages* (lorsqu'une rangée de pages occupe plus de la moitié de la hauteur du *viewport*), on atteint la fin de cette rangée de pages. Le *monte-page* transporterait alors la caméra à la rangée de pages du dessus. Bien que cette solution impose un changement dans le sens du mouvement de contrôle de la caméra, elle a l'avantage de permettre que les pages soient disposées dans un ordre facile à suivre visuellement. C'est-à-dire qu'on ne doit pas retourner le regard vers le début de la ligne lorsqu'on atteint la fin d'une rangée.

Le deuxième comportement, le *vide-de-pages*, fait que la caméra est transportée d'un côté à l'autre de l'hémicylindre lorsqu'elle atteint la fin d'une ligne. L'avantage de cette technique est que les pages peuvent être disposées toutes dans le même ordre (croissant de gauche à droite, quelle que soit la ligne).

### 7.5.3 Implémentation

Le *Mur de pages* a été présenté dans deux configurations de dispositif d'affichage (fig. 7.8) : sur un écran 22 pouces ou sur le dispositif d'immersion visuelle *VisionStation* (voir chap. 6.3.1 pour une présentation de ce dispositif).

**Dispositif pour le contrôle de la caméra.** Un *SpaceNavigator* a été utilisé comme périphérique d'entrée pour le contrôle de la caméra. Il s'agit d'une sorte de joystick isométrique à six DDL. Ce dispositif offre un feed-back élastique à peine perceptible. Il est formé d'une base lourde sur laquelle un petit cylindre (la manette) est monté ; on la saisit par le bout des doigts. Les dispositifs isométriques procurent un contrôle plus efficace pour une tâche de navigation car, par une simple modulation de la pression appliquée, on peut faire défiler toutes les pages (Zhai et al., 1997). En relâchant la manette, la caméra s'arrête.



**Préparation des textures.** L'application prototype a été développée en C et *OpenGL*. Les images master en niveaux de gris ont été légèrement recadrées pour que leurs dimensions aient un ratio de 2 :1. Elles ont été ensuite six fois échantillonnées dans des versions plus petites. Toutes ces versions avaient des dimensions qui étaient des puissances de deux (de 2048x1024 jusqu'à 32x16 pixels). Des images ayant des dimensions de ce type tirent profit de la mémoire de la carte graphique lorsqu'elles sont chargées en tant que textures. Le jeu de versions réduites alimente le filtrage par *MIP mapping*, ce qui procure un zoom plus fluide et plus net (voir 6.3.4 à ce propos).

Les GIF sont chargées dans la mémoire de la carte graphique utilisant une fonctionnalité de compression de textures lors de l'initialisation de l'application. Comme les textures sont toutes résidentes, il n'y a pas de délai dû au *paging* et l'animation n'est pas saccadée. En effet, le prototype peut charger 500 pages et l'utilisateur parvient à zoomer sur n'importe quelle page en moins de deux secondes.

#### 7.5.4 Retour des utilisateurs

Vingt personnes, parmi des bibliothécaires, des professionnels de la numérisation, des utilisateurs débutants et avancés, ont déjà testé le prototype. Il a été présenté dans un Salon des techniques de la numérisation (FAN'2008, Paris) où il a été très apprécié par les personnes qui l'ont essayé. Ceux qui travaillent dans la numérisation (ou dans la visualisation d'images en général) ont trouvé que cette interface apporte une facilité qu'ils n'ont pas trouvée dans d'autres applications déjà essayées.

Plusieurs utilisateurs, surtout ceux qui ne sont pas habitués aux dispositifs 3D, ont eu des difficultés pour contrôler les mouvements de zoom de la caméra. La forme du dispositif et son pauvre feed-back élastique paraissent troubler les utilisateurs qui finissent par réaliser des mouvements imprécis. Le mouvement vertical, par exemple, est assez difficile puisque la prise pour ce type de mouvement n'est pas confortable ainsi que très pauvre de feed-back élastique.

#### 7.5.5 Mieux contrôler la vitesse du panorama.

Igarashi et Hinckley (2000) évoquent le problème du « débordement visuel » généré par le défilement accéléré d'un document (par exemple, une longue page Web). Ce problème est bien proche de celui expérimenté par les utilisateurs qui ont essayé le *Mur de pages* lorsque la caméra se trouve proche des pages. En effet, ce prototype a adopté une « navigation cylindrique ». De ce fait, la vitesse angulaire parcourue par la caméra, qu'elle soit proche du centre ou collée au mur, est toujours la même. En revanche, la longueur parcourue sera toutefois beaucoup plus grande lorsque la caméra se trouve proche du mur, c'est-à-dire loin du centre

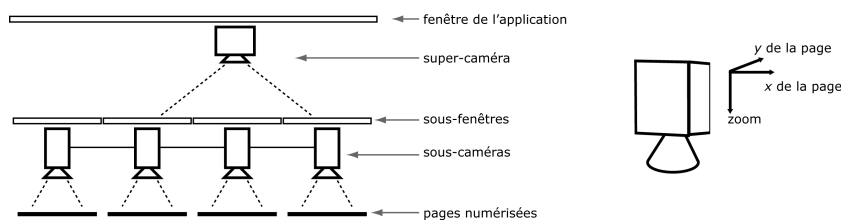


FIG. 7.9: A gauche, schéma avec les caméras virtuelles de la maquette « Panorama de détails ». Les sous-caméras sont reliées les unes aux autres et exécutent toujours les mêmes mouvements. A droite, les DDL le long desquels les caméras virtuelles se déplacent. Les sous-caméras restent toujours parallèles aux images des pages numérisées. La super-caméra reste toujours parallèle au plan des sous-fenêtres.

du cylindre.

Deux *mappings* pour la caméra sont envisagés pour améliorer le confort visuel des opérations de ballayage. Un *mapping* variable pourrait réduire la vitesse de rotation de la camera en fonction de sa distance par rapport au mur. Cela ferait que visuellement la rotation produit toujours le même effet. Un deuxième *mapping* étudié est une adaptation directe de la technique de Igarashi et Hinckley. Il s'agit de reculer la caméra vers le centre du cylindre lorsqu'on la tourne rapidement et de rapprocher du mur dès que le mouvement ralentit ou dès qu'il s'arrête. L'avantage de cette deuxième configuration vient du fait que la vitesse de la rotation n'est pas ralentie et, en même temps, le confort visuel de l'utilisateur est pris en compte.

## 7.6 Panorama de détails

Comme nous l'avons vu dans la *Grille de détails*, il peut être avantageux de visualiser les détails de plusieurs pages en même temps. On se sert des indices disponibles dans le détail affiché pour vérifier la normalité et la qualité de la page ou l'intérêt qu'elle peut porter. Dans l'interface utilisée actuellement, ces détails sont pré-fabriqués en recoupant le centre de la page. Il s'agit d'un « cache graphique ». Ce cache graphique est nécessaire pour que l'interface soit utilisable pour les lots de toutes tailles et aussi pour qu'on la consulte via des navigateurs différents et sans plugin. On compte ajouter une étape de reconnaissance du contenu de la page pour que le détail recoupé encadre forcément du texte ou de l'illustration et non pas du vide (comme il arrive dans certaines pages).

Cependant, la *Grille de détails* ne permet pas de changer dynamiquement la position de la fenêtre de détail. On peut vouloir balayer les détails du haut de la page, pour vérifier si des notes ont été prises. Ou bien, on peut également vouloir voir une zone de détail un peu plus grande de chaque page. Ces changements en temps réel ne pourraient pas être pris en compte par la plate-forme de travail actuelle.

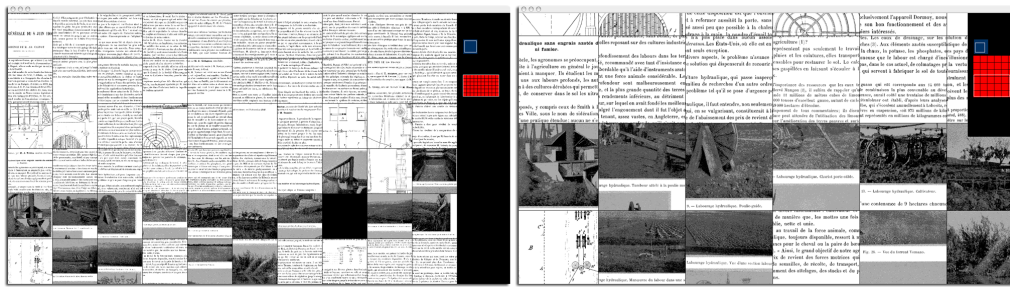


FIG. 7.10: Copies d'écran de la maquette interactive du *Panorama de détails* réalisé sur *Processing*. A gauche, on voit un grand morceau de toutes les pages disponibles. A droite, on voit un morceau plus petit (mais avec plus de résolution) d'un nombre plus réduit de pages. Dans la maquette, à côté des pages, le carré à l'intérieur du carré bleu représente la position et le zoom de la caméra abstraite sur chaque page et le carré à l'intérieur de la grille rouge représente la position et le zoom de la super-caméra sur toutes les pages.

J'ai donc exploré une première maquette d'une grille de détail paramétrable, appelée « Panorama de détails ». Il s'agit d'une fenêtre composée de plusieurs sous-fenêtres. On peut imaginer que l'image-détail de chaque page est fournie par une caméra qui plane au-dessus et qui est parallèle à la page correspondante. Dans cette maquette, l'utilisateur contrôle la position et le zoom d'une « camera-abstraite ». La camera-abstraite est associée, en réalité, à toutes les caméras qui planent sur chaque page de la grille (fig. 7.9). Il est ainsi possible de faire un panorama multiple qui s'affiche dans chaque sous-fenêtre de l'interface.

### 7.6.1 Paramètres de navigation

Outre le mouvement de panorama sur les pages, notre maquette permet aussi de changer la taille de la fenêtre de visualisation de chaque page. L'utilisateur peut ainsi, par exemple, voir une partie plus grande des pages à l'échelle 100%. Cependant, l'augmentation de la taille des sous-fenêtres réduit le nombre de sous-fenêtres qui peuvent être simultanément affichées sur la fenêtre principale. La solution est de considérer la fenêtre principale elle aussi comme une « super-caméra » qui plane sur une grille. On peut approcher ou éloigner la super-caméra des sous-fenêtres. Comme le montre la figure 7.9, on peut contrôler les mouvements de la camera-abstraite et les mouvements de la super-camera.

### 7.6.2 Interface de navigation

L'obstacle rencontré dans cette première version du *Panorama de détails* est le grand nombre de paramètres de visualisation à contrôler, à savoir le zoom et le déplacement panoramique à la fois de la caméra-abstraite et de la super-caméra (trois DDL pour chaque caméra). Il s'agit ainsi d'une interface à six DDL. Dans cette

maquette développé en *Processing* (voir fig. 7.10), les *viewports* des deux caméras sont représentés par des icônes qui fonctionnent comme une interface de navigation. Ces représentations, à l'instar du widget *Navigator* dans *Photoshop*, peuvent être glissées et redimensionnées directement avec la souris ou par des touches du clavier. Cette interface, similaire à une barre de défilement 2D, a l'avantage d'indiquer quels sont la position et le niveau de zoom de la caméra par rapport à la page.

Le contrôle de six variables est naturel s'il s'agit des six variables qui définissent la position et l'orientation d'un objet dans l'espace, c'est-à-dire  $(X, Y, Z)$  et  $(R_X, R_Y, R_Z)$ . Un joystick isométrique pourrait offrir une solution bureautique pour cette manipulation. Cependant, une captation six DDL associée à la métaphore des deux caméras décrite en haut n'offrirait pas une manipulation naturelle. Une voie d'exploration serait de se servir d'une manipulation bi-manuelle : la main gauche pourrait contrôler la *super-caméra* tandis que la main droite manipule la *caméra-abstraite*.

Toutefois, il se peut aussi que la métaphore de deux caméras explorée ne soit pas la plus appropriée. Une deuxième voie est déjà en cours d'étude ; il s'agit d'une interface basée sur une « métaphore optique ». A l'instar de certains types de lentille ou de miroir, l'utilisateur pourrait éprouver différentes distorsions visuelles en déplaçant une caméra le long d'un axe principal. Ainsi, à l'aide de plusieurs seuils d'agrandissement et de réduction, l'utilisateur pourrait trouver des configurations satisfaisantes à la fois pour chaque page et pour toutes les pages via le déplacement d'une seule caméra.

## 7.7 Conclusions

Les quatre techniques de visualisation décrites dans ce chapitre se trouvent à différents stades de développement. Toutes les quatre ont néanmoins le même but : de favoriser la compréhension d'un lot de pages numérisées. La *Grille de détails* est couramment utilisée au jour le jour dans la bibliothèque numérique du CNAM (CNUM). Il sera important d'étudier comment d'autres bibliothèques numériques gèrent, dans la pratique, le problème du contrôle qualité.

Le *Mur de pages*, bien qu'il soit techniquement bien avancé, ne procure pas encore de fonctions permettant de l'utiliser dans un cadre réel de contrôle qualité. Par exemple, on ne peut pas encore récupérer des informations additionnelles sur une page donnée (nom du fichier, date de numérisation, dimensions en pixel, colorimétrie) et on ne dispose pas non plus d'une interface d'annotation intégrée.

Outre le développement de ces fonctions, j'envisage conduire une étude ergonomique dont les résultats aideraient à définir un *mapping* plus efficient et confortable pour la caméra. Deux idées concernant le *mapping* ont déjà été proposées et sont en cours d'implémentation : le passage semi-automatique d'une ligne à

l'autre lors d'un *balayage séquentiel* et la variation « écologique » de la vitesse de rotation (en fonction de la distance qui sépare la caméra du *Mur de pages*).

D'autre part, le prototype du *Mur de pages* doit devenir plus robuste pour supporter la visualisation d'un ensemble de dimensions très importantes (plus de 2 000 pages). Une architecture qui prenne en compte le chargement et le déchargement dynamique des textures doit être étudiée (Cline et Egbert, 1998). Les travaux liés à la visualisation des images Gigapixel pourraient offrir des pistes (Kopf et al., 2007) intéressantes à ce sujet. D'autant plus que ces techniques évoluent rapidement et se démocratisent grâce aux outils Web de consultation de cartes et photos satellitaires. Un gestionnaire de chargement adapté aux parcours caractéristiques de la tâche de contrôle qualité pourra ainsi être développé.



## Conclusions et perspectives

*Ce chapitre résume les aspects principaux de l'étude accomplie par cette thèse et récapitule ses principales contributions. Y sont indiquées également les directions les plus fertiles pour poursuivre les investigations des sujets abordés.*

Dans cette thèse, de nouvelles techniques pour gérer la manque d'« espace » dans l'interaction bureautique ont été explorées. Deux types d'espace ont été abordés : *l'espace moteur* et *l'espace visuel*.

Concernant l'espace moteur, les apports d'une souris dotée d'un capteur de rotation ont été explorés. Nous avons vu que, dans certains contextes de travail, l'utilisation de la souris traditionnelle impose une interaction « hachée » et marquée par des mouvements répétitifs. Cette thèse a proposé des techniques, basées sur la rotation de la souris, qui permettent à l'utilisateur de réaliser des mouvements plus continus, plus fins et plus expressifs. La réalisation d'une évaluation contrôlée a permis de comparer quatre techniques : deux basées sur une souris rotative et deux sur une souris traditionnelle. Cette évaluation a montré que la souris rotative était plus performante pour une manipulation à trois degrés de liberté (DDL).

Par rapport à l'espace visuel, le contexte des bibliothèques numériques a fourni un bon cadre d'analyse. Deux points de vue ont été choisis pour cette étude. Premièrement, depuis la perspective du « lecteur numérique », c'est-à-dire de l'utilisateur qui cherche à « comprendre » une collection ou à trouver un document. Deuxièmement, depuis la perspective du « bibliothécaire numérique », c'est-à-dire de l'utilisateur chargé de s'assurer de la validité logique et graphique des documents numérisés qui seront intégrés à l'entrepôt numérique. A ce deux cas, marqués par la difficulté d'appréhender toute l'information (essentiellement visuelle), de nouvelles interfaces de visualisation, proposées dans cette thèse, tentent d'y remédier. Ces interfaces, en traitant les données comme de la texture, les distribuent dans un espace virtuel. L'utilisateur peut ainsi, en contrôlant la position d'une ca-

méra dans cet espace, voir *toute l'information* et s'approcher des zones qui offrent les indices visuels qui l'intéressent.

## 8.1 Contributions

Les contributions à la problématique des limitations des espaces visuel et moteur sont à la fois en termes de discussion de concepts, analyse des besoins, résultats quantitatifs et prototypes matériels et logiciels construits et adaptés.

**Aspects humains de la manipulation rotative.** Dans le chapitre 2, nous avons passé en revue une série d'études - issues de l'ergonomie, de la psychologie perceptive et de la psychologie motrice - qui se sont intéressées à l'interaction « multi-dimensionnelle », aux patrons de mouvements manuels et à la façon dont on oriente un objet. Ces concepts ont été interprétés, dans le contexte de cette thèse, dans l'objectif de déterminer quels sont les aspects humains qui interviennent lors de la manipulation d'une souris rotative.

**Etat de l'art et caractérisation des souris rotatives.** Le chapitre 2 apporte également une contribution notable en présentant un état de l'art des souris rotatives proposées à ce jour. Elles ont été regroupées selon la technique dont elles se servent pour capter la rotation et j'ai proposé qu'on les représente comme étant un sous-ensemble des « souris augmentées ». Les analyses empruntées des taxonomies générales - créées pour décrire les périphériques d'entrée manuels - ont également permis d'identifier les caractéristiques fondamentales des souris rotatives.

**Pistes pour la conception des souris rotatives.** Deux prototypes de souris rotative ont été construits, ils ont évolué et connu quelques variations. L'un a été constitué à partir de l'assemblage de deux souris optiques sans fil et l'autre a été adapté à partir d'une souris *Wacom 4D Mouse* (qui capte nativement la rotation verticale). A partir de ces deux développements, certaines contraintes associées à ce type de dispositif ont été identifiées et pourront être prises en compte dans de futurs dispositifs ou expériences. La taille de la main par rapport au dispositif, l'emplacement du bouton lié à la difficulté de réaliser certains mouvements en le gardant appuyé, l'asymétrie de la coque et les avantages d'une correspondance absolue sont, entre autres, des questions rencontrées au fil des explorations (et auxquelles j'espère avoir apporté quelques pistes).

**Techniques d'interaction rotatives.** Cette thèse propose deux nouvelles techniques d'interaction : une pour le réglage de potentiomètres graphiques rotatifs et l'autre pour la sélection d'un outil dans une palette graphique. La première



technique fournit un moyen de manipuler un outil logique sans changement explicite de mode, ce qui favorise un mouvement plus fluide et continu. Dans la deuxième technique, l'espace de sélection et l'espace d'édition d'une application sont associés à des sous-ensembles de transducteurs distincts.

**Visualisation immersive et « bi-focale ».** Une interface de navigation immersive, développée dans le cadre de ces travaux, autorise la consultation d'une collection de documents numériques. Les avantages de l'utilisation d'un dispositif qui élargit le champ de vision de l'utilisateur ont été indiqués ainsi que les inconvénients, en matière de qualité graphique, dérivés de la solution technique que nous avons adoptée. Une configuration « bi-focale » permet à l'utilisateur de se servir simultanément de deux affichages de la collection (une vue zoomée en haute définition et une vue globale en faible résolution), pour ce type de tâche « cognitive » et très visuelle.

**Navigation expérimentale d'un grand espace graphique.** Enfin, dans le chapitre 7, la problématique de la visualisation d'un grand nombre d'images numérisées a été introduite et approfondie. Cette problématique, bien qu'elle ne soit pas très présente dans la recherche en IHM (en comparaison avec l'intérêt porté à la visualisation de collections de photos personnelles), présente des enjeux de plus en plus importants (du fait de la numérisation massive des documents papiers). Par ailleurs, les caractéristiques de la fouille visuelle demandée par ce type de tâche pourront peut-être fournir des pistes à d'autres tâches comme, par exemple, la visualisation d'images giga-pixel. Outre la description des challenges de ce contexte, une série d'interfaces expérimentales ont été développées. Une de ces interfaces est opérationnelle et elle est utilisée au sein de la bibliothèque numérique CNUM. Dans ces interfaces, j'essaie de montrer que, afin de visualiser des pages qui appartiennent à un même ensemble (que cet ensemble soit un volume, une collection ou un lot de documents similaires), on doit faire appel à des stratégies différentes que lorsqu'on visualise des photos. Deux stratégies imaginées dans ces travaux sont les suivantes : (1) traiter toutes les pages d'un même ensemble comme étant une grande image et (2) visualiser simultanément les détails de plusieurs photos avec un même encadrement. Concernant, la deuxième stratégie, elle se base sur l'idée que les pages ont des « topologies » connues. Contrairement à une photo, l'utilisateur peut « espérer » trouver certains éléments graphiques dans certaines zones plutôt que dans d'autres. Par exemple, les bords de la page sont généralement des zones a priori moins intéressantes, sauf s'il cherche des notes prises par un lecteur.

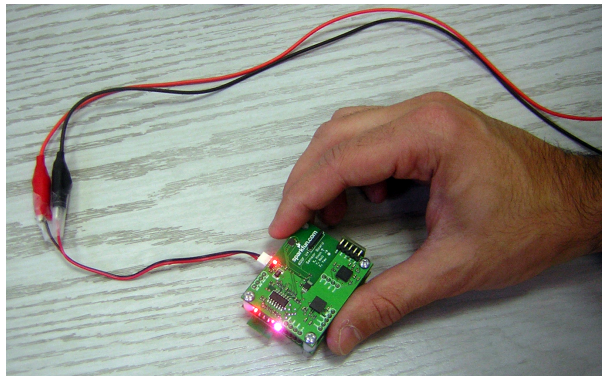


FIG. 8.1: La centrale inertielle à six DDL *Sparkfun v4*.

## 8.2 Perspectives

Outre les perspectives indiquées à la fin de chaque chapitre, quelques questions d'intérêt qui pourraient être approfondies dans le court et moyen terme sont rappelées ici.

**Amélioration du prototype rotatif.** Les prototypes employés au cours de cette thèse présentaient des déficiences concernant la précision des mesures captées, à savoir l'absence de captation absolue dans le premier prototype (la double-souris) et le délai des données d'orientation dans le prototype magnétique (*Wacom 4D Mouse*). Certaines investigations n'ont pas pu être approfondies à cause de ces limitations. La double-souris est peu fiable pour les manipulations plus longues (puisque les erreurs d'estimation de l'orientation absolue s'accumulent et deviennent gênantes) ainsi que pour des manipulations demandant de repositionner souvent le dispositif (car ce type d'action tend aussi à introduire un écart dans la correspondance entre l'orientation physique et l'orientation logique). D'autre part, la souris magnétique *Wacom* est peu fiable lorsqu'on souhaite évaluer des techniques d'interaction où les durées des opérations font moins de dix secondes. Ce problème pourrait être contourné en concevant des scénarios d'évaluation avec des tâches plus longues ; cela pourrait amenuiser l'impact négatif de ce délai.

Par ailleurs, j'envisage d'utiliser un prototype plus robuste pour les travaux à venir. Une possibilité étudiée en ce moment consiste à réaliser le prototype de la souris rotative avec une centrale inertielle dotée d'un magnétomètre (comme par exemple, la centrale inertielle à six DDL *Sparkfun v4 Bluetooth* montrée dans la figure 8.1). La rotation pourrait être rapidement calculée grâce à l'accéléromètre tandis que le magnétomètre permettrait de constamment corriger l'orientation absolue. Cette technologie présente néanmoins des inconvénients, à savoir la sensibilité du magnétomètre aux objets métalliques et aux champs magnétiques environnants.

**La forme et la texture comme canal de sortie.** Il s'agit à l'avenir d'explorer les bénéfices que des repères tactiles présents sur la coque de la souris pourraient apporter à l'utilisateur. Ces informations peuvent-elles indiquer intuitivement à l'utilisateur quel est l'outil sélectionné? Cela a été une des questions en suspens durant la conception de la technique « palette satellitaire ». Ces indications permettraient de mélanger un espace absolu (l'orientation de la souris) avec un espace relatif (la position de la souris). Par ailleurs, à moyen terme, il faudrait étudier comment un retour tactile « actif » (Snibbe et al., 2001) - par exemple, la vibration - pourrait rendre certaines manipulations davantage indépendantes d'un guidage visuel intensif - l'activation d'un outil pourrait être indiquée par une petite vibration. Enfin, dans une perspective plus ambitieuse, la forme du dispositif elle-même pourrait varier dynamiquement en fonction des options logiques qu'il contrôle (Coelho et al., 2009). C'est-à-dire que, dans le cas de la technique « palette satellitaire », la coque de la souris aurait autant de côtés que d'outils qu'elle peut sélectionner. Par exemple, si l'on lance une application dans laquelle l'orientation de la souris peut désigner huit outils différents, la coque de celle-ci prendrait une forme octogonale.

**Design et mécanique de la souris.** D'autres aspects concernant la forme de la souris sont étroitement liés aux techniques d'interaction qui peuvent être associées à ce dispositif. Comment la taille, le poids et la hauteur de la souris peuvent favoriser des mouvements rotatifs très fins, tels que ceux procurés par des boutons sonores professionnels? La « volatilité » dans la rotation de la souris, telle qu'explorée par Fitzmaurice (1996), offre aussi un espace de conception pour ces dispositifs sans équivalent dans les dispositifs linéaires. A nouveau, des questions de forme, poids, mécanique de rotation et surface de déplacement interviennent. Enfin, ces souris créent un « problème » quant à l'emplacement et le mécanisme d'activation du bouton. Ce sujet doit être traité à partir d'une réflexion qui passe par le design et l'ergonomie des coques mais aussi par la conception de *techniques d'activation implicites*.

**Navigation dans un grand espace graphique.** Concernant la visualisation de grands ensembles de données graphiques, plusieurs directions ont été indiquées pour la poursuite de cette étude.

Tout d'abord, comme relaté dans le dernier chapitre, il faut mentionner la mise au point et l'évaluation de *mappings* pour le contrôle de la caméra. Comme le montre Igarashi et Hinckley (2000), il y a un espace d'expérimentation peu exploré qui concerne l'interaction entre les variables de navigation dans une tâche de visualisation de document. La vitesse de la caméra peut varier en fonction de la distance entre celle-ci et le document, entre la caméra et le sol et entre la caméra et le bord du document. La « friction » que subit la caméra peut s'atténuer lorsque celle-ci traverse des zones avec moins de pages et augmenter lorsqu'elle

passer devant des zones du document possédant une densité de pixels noirs plus grande que la moyenne du document (on suppose qu'il s'agit d'une illustration). Il s'agit donc d'expérimenter une navigation plus « écologique » vis-à-vis des besoins de la tâche de navigation en question et vis-à-vis de l'importance graphique des différentes zones du document.

Une autre voie d'exploration vers une navigation écologique est celle de la conception de nouvelles métaphores pour le contrôle de la caméra. Par exemple, dans les interfaces développées, la rotation de la caméra autour de l'axe transversale (le roulis) a été désactivée car il me semblait que ce mouvement était dispensable et qu'il rendait le contrôle de la caméra plus pénible pour les utilisateurs débutants. Il apparaît toutefois que ce type de mouvement pourrait enrichir la navigation si on l'exploitait comme un mouvement temporaire ou « élastique » (à l'instar de la façon dont on soulève la tête en regardant des objets réels). Lorsqu'on souhaite regarder un objet qui se trouve au dessus de notre champ de vision (avec la tête en position neutre), on incline la tête, mais on la rabaisse dès qu'il finit son inspection. De même, dans les contextes de visualisation décrits, en inclinant la caméra vers le haut, l'utilisateur pourrait savoir s'il y a des images qui l'intéressent. En lâchant le *joystick*, la caméra reviendrait à une position neutre. Cela éviterait à l'utilisateur de naviguer dans l'axe vertical s'il n'est pas sûr de trouver le document qu'il cherche.

Un début d'exploration concerne le *Panorama de détails*, une technique multifocale pour la navigation jumelée et simultanée sur plusieurs pages. L'affichage varie selon six paramètres et il me semble que la métaphore d'une caméra à six DDL dans une scène ne fonctionne pas dans ce cas. Ainsi une expérimentation de métaphores optiques (par exemple, des distorsions fish-eye) et « topologiques » (Elmqvist et al., 2008) pourrait apporter de nouvelles pistes.

Par ailleurs, le prototype du *Mur de pages* devra être amélioré pour qu'il soit capable d'« avaler » des lots d'images sans limitation du nombre de pages. Actuellement, le prototype ne dispose pas de gestion dynamique de la mémoire ; le nombre de pages est donc limité à la mémoire de la carte graphique. Concernant les aspects techniques, il serait intéressant de chercher des pistes dans les interfaces de visualisation de photos giga-pixel et de cartes. D'autre part, étudier la possibilité de créer un « chargement intelligent » (Cline et Egbert, 1998) est une voie d'enrichissement du dispositif. Dans le chapitre précédent, ont été identifiées deux stratégies de navigation principales dont se servent les utilisateurs pour le contrôle d'un lot de pages. En approfondissant la connaissance sur ces patrons de navigation, il peut être possible de déterminer des zones du document plus passibles d'être visitées à partir d'une position donnée. Ces informations permettraient donc que l'application priorise le chargement d'images en haute définition associées à ces zones.

Enfin, l'utilisation des diagrammes « espace-échelle » (Furnas et Bederson, 1995) permettrait de mieux illustrer et analyser, comme il a été fait par Jul et Furnas

(1998), les caractéristiques de la navigation dans ces grands espaces graphiques. De même, une description plus précise de la *Grille de détails*, comme en utilisant les outils de conception proposés par Plaisant et al. (1995), pourrait permettre d'identifier d'autres interfaces ayant exploré des problématiques similaires.

Voilà quelques-unes des principales réalisations de cette thèse et des perspectives ouvertes pour la continuation de ses travaux.



# Bibliographie

- Agarawala, A. et Balakrishnan, R. (2006). Keepin'it real : pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1283–1292. ACM New York, NY, USA. (pages 4, 57 et 59)
- Almeida, R. et Cubaud, P. (2006). Supporting 3d-window manipulation with a yawing mouse. In *Proc. of NordiCHI'06*, pages 477–480, New York, NY, USA. ACM Press. (page 47)
- Almeida, R. et Cubaud, P. (2007). Nearly-integral manipulation of rotary widgets. In *11th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'07)*, volume 1, pages 489–492, Rio de Janeiro, Brazil. Springer-Verlag. (page 107)
- Almeida, R., Cubaud, P., Dupire, J. et Topol, A. (2006). Interactions et métadonnées riches pour les bibliothèques numérisées. *Document numérique*, (2006/2) :83–109. (pages 11 et 123)
- Almeida, R. A. B., Alessio, P., Topol, A. et Cubaud, P. (2009). A Visualization Technique for Quality Control of Massive Digitization Programs. In *Proceedings of the 13th European Conf. on Res. and Adv. Tech. for Digital Libraries (ECDL'09)*, LNCS5714, pages 150–155. Springer Verlag. (page 149)
- Andersen, T. H. (2003). Mixxx : towards novel dj interfaces. In *NIME '03 : Proceedings of the 2003 conference on New interfaces for musical expression*, pages 30–35, Singapore, Singapore. National University of Singapore. (page 82)
- Appert, C. et Fekete, J.-D. (2006). Orthozoom scroller : 1d multi-scale navigation. In *CHI '06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 21–30, New York, NY, USA. ACM. (pages 7 et 94)
- Arthur, K. (1996). Effects of field of view on task performance with head-mounted displays. In *CHI '96 : Conference companion on Human factors in computing systems*, pages 29–30, New York, NY, USA. ACM. (page 124)
- Ayatsuka, Y., Rekimoto, J. et Matsuoka, S. (1998). Popup vernier : a tool for sub-pixel-pitch dragging with smooth mode transition. In *UIST '98 : Proceedings of*

- the 11th annual ACM symposium on User interface software and technology*, pages 39–48, New York, NY, USA. ACM. (page 8)
- Baecker, R. et Buxton, W. (1987). *Readings in human-computer interaction : A multi-disciplinary approach*, pages 357–365. Morgan Kaufmann Publishers, Los Altos, CA. (page 35)
- Bailly, G., Lecolinet, E. et Nigay, L. (2007). Quinze ans de recherche sur les menus : critères et propriétés des techniques de menus. In *Actes de la 19ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'07)*, pages 119–126, Paris, France. ACM Press. (pages 98, 99 et 115)
- Bailly, G., Nigay, L. et Auber, D. (2005). 2m : un espace de conception pour l'interaction bi-manuelle. In *Conférence UBIMOB 2005, Deuxièmes Journées Francophones : Mobilité et Ubiquité 2005*, pages 166–173. (pages 19 et 107)
- Balakrishnan, R. (2004). "Beating" Fitts' law : virtual enhancements for pointing facilitation. *Int. Journal Human-Computer Studies*, 61(6) :857–874. (page 86)
- Balakrishnan, R., Baudel, T., Kurtenbach, G. et Fitzmaurice, G. (1997). The rockin'mouse : integral 3d manipulation on a plane. In *CHI '97 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 311–318, New York, NY, USA. ACM. (pages 6, 44, 52 et 67)
- Balakrishnan, R. et Patel, P. (1998). The padmouse : facilitating selection and spatial positioning for the non-dominant hand. In *CHI '98 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 9–16, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. (pages 42)
- Baudisch, P. et Good, N. (2002). Focus plus context screens : displays for users working with large visual documents. In *CHI '02 : CHI '02 extended abstracts on Human factors in computing systems*, pages 492–493, New York, NY, USA. ACM. (pages 5, 131, 132 et 134)
- Beaudouin-Lafon, M. (2000). Instrumental interaction : an interaction model for designing post-wimp user interfaces. In *Interact., A. T. C.-H., editor, Proc. of CHI '00*, pages 446–453, New York, NY, USA. ACM Press. (pages 88, 98, 100 et 103)
- Beaudouin-Lafon, M. (2001). Novel interaction techniques for overlapping windows. In *UIST '01 : Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 153–154, New York, NY, USA. ACM. (pages 4, 57 et 60)
- Beaudouin-Lafon, M. (2006). Interaction graphique. In Akoka, J. et Comyn-Wattiau, I., editors, *Encyclopédie de l'Informatique et des Systèmes d'Information*. Vuibert. (page 98)
- Bederson, B. B. et Hollan, J. D. (1994). Pad++ : a zooming graphical interface for exploring alternate interface physics. In *UIST '94 : Proceedings of the 7th annual*



- ACM symposium on User interface software and technology*, pages 17–26, New York, NY, USA. ACM Press. (pages 3 et 60)
- Bier, E. A., Stone, M. C., Pier, K., Buxton, W. et DeRose, T. D. (1993). Toolglass and magic lenses : the see-through interface. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, New York, NY, USA. ACM. (pages 7, 42, 103 et 115)
- Bjorklund, E. (2001). Toward a general theory of control knobs. *ECONF C*, 011127 :THAP071. (page 33)
- Blanch, R., Guiard, Y. et Beaudouin-Lafon, M. (2004). Semantic pointing : improving target acquisition with control-display ratio adaptation. In *CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 519–526, New York, NY, USA. ACM. (page 96)
- Bourgeois, F. et Guiard, Y. (2002). Multiscale pointing : facilitating pan-zoom coordination. In *CHI '02 : CHI '02 extended abstracts on Human factors in computing systems*, pages 758–759, New York, NY, USA. ACM. (page 8)
- Bourke, P. (2004). Direct Rendering of Warped Fisheye Views for Spherical Mirror Projection. In (*unpublished*) <http://local.wasp.uwa.edu.au/pbourke/>. (page 127)
- Bowman, D. A., Kruijff, E., LaViola, J. J. et Poupyrev, I. (2004). *3D User Interfaces : Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA. (pages 24 et 125)
- Bowman, D. A. et McMahan, R. P. (2007). Virtual Reality : How Much Immersion Is Enough ? *Computer*, 40(7) :36–43. (page 125)
- Buxton, W. (1983). Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.*, 17(1) :31–37. (pages 37, 38, 79 et 84)
- Buxton, W. (1986). There's more to interaction than meets the eye : some issues in manual input. In Norman, D. A. et Draper, S. W., editors, *User Centered System Design : New Perspectives on Human-Computer Interaction*, pages 319–337. Lawrence Erlbaum Associates, Hillsdale, NJ, USA. (pages 41 et 88)
- Buxton, W. (1990). A three-state model of graphical input. In *INTERACT '90 : Proc. of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*, pages 449–456. North-Holland. (pages 37, 43 et 91)
- Buxton, W. (2007). Interfaces homme-machine et production audiovisuelle. Présentation orale lors de la Journée professionnelle Cap Digital. Présentation orale (en anglais) lors de la Journée professionnelle Cap Digital au Centre Pompidou, Paris, France. (page 7)
- Buxton, W. (2008). The Long Nose of Innovation. *BusinessWeek* (02/01/2008). (page 6)

- Buxton, W. et Myers, B. (1986). A study in two-handed input. In *CHI '86 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–326, New York, NY, USA. ACM Press. (page 19)
- Card, S. K., Mackinlay, J. D. et Robertson, G. G. (1990). The design space of input devices. In *CHI '90 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 117–124, New York, NY, USA. ACM Press. (pages 36, 37, 38 et 41)
- Card, S. K., Mackinlay, J. D. et Robertson, G. G. (1991). A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.*, 9(2) :99–122. (page 28)
- Card, S. K., Robertson, G. G. et York, W. (1996). The webbook and the web forager : an information workspace for the world-wide web. In *CHI '96 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 111–ff., New York, NY, USA. ACM. (pages 59 et 122)
- Cechanowicz, J., Irani, P. et Subramanian, S. (2007). Augmenting the mouse with pressure sensitive input. In *CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1385–1394, New York, NY, USA. ACM. (page 43)
- Chapuis, O. et Roussel, N. (2005). Metisse is not a 3d desktop ! In *UIST '05 : Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 13–22, New York, NY, USA. ACM. (pages 59 et 61)
- Cline, D. et Egbert, P. K. (1998). Interactive display of very large textures. In *VIS '98 : Proceedings of the conference on Visualization '98*, pages 343–350, Los Alamitos, CA, USA. IEEE Computer Society Press. (pages 155 et 162)
- Cockburn, A., Gutwin, C. et Alexander, J. (2006). Faster document navigation with space-filling thumbnails. In *CHI '06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1–10, New York, NY, USA. ACM. (page 141)
- Coelho, M., Poupyrev, I., Sadi, S., Vertegaal, R., Berzowska, J., Buechley, L., Maes, P. et Oxman, N. (2009). Programming reality : from transitive materials to organic user interfaces. In *CHI EA '09 : Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4759–4762, New York, NY, USA. ACM. (page 161)
- Crew, L. (2002). 3 DOF mouse Final Report. Technical report, Hi-Fi Design. (pages 54)
- Cubaud, P. (2006). Du texte au volume : contributions aux bibliothèques numériques. Habilitation à diriger des recherches, Université Paris 8. (pages 10 et 123)

- Cubaud, P. et Deblock, G. (2003). Le conservatoire numérique des arts et métiers : Historique du projet et organisation du site. *Actes du colloque La numérisation des textes et des images : techniques et réalisations*, pages 13–22. (page 12)
- Cubaud, P., Dupire, J. et Topol, A. (2005). Digitization and 3d modeling of movable books. In *JCDL '05 : Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 244–245, New York, NY, USA. ACM. (page 10)
- Cubaud, P. et Girard, D. (1998). ABU : une bibliothèque numérique et son public. *Document numérique*, 2(3-4) :13–30. (page 12)
- Cubaud, P., Haas, J. et Topol, A. (2004). Numérisation 3d de documents par photogrammétrie. In *Colloque International Francophone sur l'Écrit et le Document*. (page 11)
- Cubaud, P., Stokowski, P. et Topol, A. (2002). Binding browsing and reading activities in a 3d digital library. In *JCDL '02 : Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 281–282, New York, NY, USA. ACM. (pages 11, 118 et 126)
- Cubaud, P., Thiria, C. et Topol, A. (1998). Experimenting a 3d interface for the access to a digital library. In *DL '98 : Proceedings of the third ACM conference on Digital libraries*, pages 281–382, New York, NY, USA. ACM. (pages 11, 13, 121 et 126)
- Cubaud, P. et Topol, A. (2001). A VRML-based user interface for an online digitized antiquarian collection. In *Proceedings of the sixth international conference on 3D Web technology*, pages 51–59. ACM New York, NY, USA. (page 11)
- Cubaud, P., Topol, A. et Vodislav, D. (2000). Les limites de VRML pour les comportements interactifs : étude de cas. In *Proceedings of ERGO-IHM'2000*. (page 11)
- Darken, R. P. et Cevik, H. (1999). Map usage in virtual environments : Orientation issues. In *VR '99 : Proceedings of the IEEE Virtual Reality*, page 133, Washington, DC, USA. IEEE Computer Society. (page 58)
- Dourish, P. (2001). *Where the Action Is : The Foundations of Embodied Interaction*. The MIT Press. (page 3)
- Dragicevic, P. (1998). Concrétiser les dispositifs d'entrée dans les outils de développement. In *Actes des dixièmes journées francophones sur l'Interaction Homme Machine (IHM'98)*. (page 36)
- Dumas, C., et Jourdan, F. Plenacoste, P. (2002). Libraries : comparisons between the real and the virtual in 3D, 2D zoomable and 2D arborescent. In *Proc. JCDL Workshop on Usability*, pages 11–14. (pages 121 et 122)
- Dupire, J., Topol, A. et Cubaud, P. (2005). Using Game Engines For Non 3D Gaming Applications. In *Proceedings of CGAMES '05*. (page 11)

- Elliott, J. et Connolly, K. (1984). A classification of manipulative hand movements. *Developmental Medicine and Child Neurology*, 26 :238—296. (pages 7, 8, 25, 26, 28, 29 et 30)
- Elmqvist, N. et Fekete, J.-D. (2008). Semantic pointing for object picking in complex 3d environments. In *GI '08 : Proceedings of graphics interface 2008*, pages 243–250, Toronto, Ont., Canada, Canada. Canadian Information Processing Society. (page 8)
- Elmqvist, N., Henry, N., Riche, Y. et Fekete, J.-D. (2008). Melange : space folding for multi-focus interaction. In *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1333–1342, New York, NY, USA. ACM. (pages 140 et 162)
- Elumens, C. (2001). *The SPIClops API*. (pages 126 et 128)
- Fallman, D., Mikaelsson, A. et Yttergren, B. (2007). The Design of a Computer Mouse Providing Three Degrees of Freedom. In *Proceedings of HCI International*, volume Part II of LNCS4551, pages 53–62. Springer-Verlag. (pages 48)
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol*, 47(6) :381–391. (page 114)
- Fitzmaurice, G. et Buxton, W. (1999). Temporal data control system. US Patent 5,973,669. (page 45)
- Fitzmaurice, G., Khan, A., Pieké, R., Buxton, B. et Kurtenbach, G. (2003). Tracking menus. In *UIST '03 : Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 71–79, New York, NY, USA. ACM. (page 87)
- Fitzmaurice, G., Matejka, J., Khan, A., Glueck, M. et Kurtenbach, G. (2008). Piecursor : merging pointing and command selection for rapid in-place tool switching. In *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1361–1370, New York, NY, USA. ACM. (page 102)
- Fitzmaurice, G. W. (1996). *Graspable user interfaces*. PhD thesis, University of Toronto, Ontario, Canada. Adviser-William Buxton. (pages 35, 45, 46, 102 et 161)
- Fitzmaurice, G. W. et Buxton, W. (1997). An empirical evaluation of graspable user interfaces : towards specialized, space-multiplexed input. In *CHI '97 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 43–50, New York, NY, USA. ACM. (page 7)
- Foley, J., Dam, A., Feiner, J. et Hughes, J. (1990). *Computer graphics : principles and practice1990*. Reading : Addison Wesley Publishing Company. (page 36)
- Forlines, C., Shen, C. et Buxton, B. (2005). Glimpse : a novel input model for multi-level devices. In *CHI '05 : CHI '05 extended abstracts on Human factors in computing systems*, pages 1375–1378, New York, NY, USA. ACM. (page 43)

- Forlines, C., Vogel, D. et Balakrishnan, R. (2006). Hybridpointing : fluid switching between absolute and relative pointing with a direct input device. In *UIST '06 : Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 211–220, New York, NY, USA. ACM. (page 37)
- Froehlich, B., Hochstrate, J., Skuk, V. et Huckauf, A. (2006). The globefish and the globemouse : two new six degree of freedom input devices for graphics applications. In *CHI '06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 191–199, New York, NY, USA. ACM Press. (pages 40 et 79)
- Furnas, G. W. (1986). Generalized fisheye views. In *CHI '86 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA. ACM Press. (page 139)
- Furnas, G. W. (2006). A fisheye follow-up : further reflections on focus + context. In *CHI '06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 999–1008, New York, NY, USA. ACM Press. (page 139)
- Furnas, G. W. et Bederson, B. B. (1995). Space-scale diagrams : understanding multiscale interfaces. In *CHI '95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. (pages 94, 140 et 162)
- Garner, W. R. (1974). *The Processing of Information and Structure*. Lawrence Erlbaum, Potomac, MD, United States. (page 21)
- Gaver, W. W. (1989). The sonic finder : An interface that uses auditory icons. *Human-Computer Interaction*, 4 :67–94. (page 3)
- Gaver, W. W. (1991). Technology affordances. In *Proceedings of CHI-91*, pages 79–84, New Orleans, LA, USA. (pages 87)
- GEC, G. (1998). *Comparison of Postures from Pen and Mouse Use (rapport développé pour la société Wacom)*. Guerneville, CA 95446 - U.S.A. (page 7)
- Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. Houghton Mifflin, Boston, MA, USA. (pages 23 et 24)
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates. (pages 5, 24, 87 et 124)
- Grudin, J. (2001). Partitioning digital worlds : focal and peripheral awareness in multiple monitor use. In *CHI '01 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 458–465, New York, NY, USA. ACM. (page 5)
- Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action : The kinematic chain as a model. *Journal of Motor Behavior*, 19 :486–517. (pages 3, 19, 29 et 101)

- Guiard, Y., Chapuis, O., Du, Y. et Beaudouin-Lafon, M. (2006). Allowing camera tilts for document navigation in the standard GUI : A discussion and an experiment. In *Proceedings of the working conference on Advanced visual interfaces*, pages 241–244. ACM New York, NY, USA. (page 148)
- Guimbretière, F., Dixon, M. et Hinckley, K. (2007). Experiscope : an analysis tool for interaction data. In *CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1333–1342, New York, NY, USA. ACM. (page 92)
- Guimbretière, F., Martin, A. et Winograd, T. (2005). Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.*, 12(3) :460–476. (page 105)
- Hannagan, J. (2007). TwistMouse for Simultaneous Translation and Rotation. Master's thesis, Department of Information Science, University of Otago. (pages 49 et 50)
- Hascoët, M., Collomb, M. et Cance, J. (2006). Accelerating object-command transitions with pie menus. In *Proceedings of Enactive'06 : Conference on Enaction and Complexity*, pages 109–111. (pages 87, 99 et 102)
- Hinckley, K., Pausch, R., Goble, J. C. et Kassell, N. F. (1994). Passive real-world interface props for neurosurgical visualization. In *CHI '94 : roceedings of the SIGCHI conference on Human factors in computing systems*, pages 452–458, New York, NY, USA. ACM. (pages 23, 24 et 108)
- Hinckley, K. et Sinclair, M. (1999). Touch-sensing input devices. In *CHI '99 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 223–230, New York, NY, USA. ACM Press. (pages 42 et 111)
- Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R. et Conway, M. (1999). The videomouse : a camera-based multi-degree-of-freedom input device. In *UIST '99 : Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 103–112, New York, NY, USA. ACM. (pages 52)
- Hinckley, K., Tullio, J., Pausch, R., Proffitt, D. et Kassell, N. (1997). Usability analysis of 3d rotation techniques. In *UIST '97 : Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 1–10, New York, NY, USA. ACM. (pages 7, 13, 27, 67, 68 et 77)
- Hsu, S. H., Jumpertz, S. et Cubaud, P. (2008). A tangible interface for browsing digital photo collections. In *TEI '08 : Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 31–32, New York, NY, USA. ACM. (page 32)
- Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science : A Multidisciplinary Journal*, 19(3) :265–288. (page 123)

- Igarashi, T. et Hinckley, K. (2000). Speed-dependent automatic zooming for browsing large documents. In *UIST '00 : Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 139–148, New York, NY, USA. ACM. (pages 151, 152 et 161)
- Isokoski, P., Raisamo, R., Martin, B. et Evreinov, G. (2007). User performance with trackball-mice. *Interact. Comput.*, 19(3) :407–427. (page 41)
- Jacob, R. J. K. et Sibert, L. E. (1992). The perceptual structure of multidimensional input device selection. In *CHI '92 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 211–218, New York, NY, USA. ACM Press. (pages 8, 20, 21, 22 et 66)
- Jacob, R. J. K., Sibert, L. E., McFarlane, D. C. et M. Preston Mullen, J. (1994). Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.*, 1(1) :3–26. (pages 20, 61 et 88)
- Jacomy, B. (2002). *L'âge du flip*. Editions Seuil. (page 31)
- Jul, S. et Furnas, G. W. (1998). Critical zones in desert fog : aids to multiscale navigation. In *UIST '98 : Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 97–106, New York, NY, USA. ACM. (page 162)
- Kandogan, E. et Shneiderman, B. (1996). Elastic windows : improved spatial layout and rapid multiple window operations. In *AVI '96 : Proceedings of the workshop on Advanced visual interfaces*, pages 29–38, New York, NY, USA. ACM. (page 60)
- Kim, S., Kim, H., Lee, B., Nam, T.-J. et Lee, W. (2008). Inflatable mouse : volume-adjustable mouse with air-pressure-sensitive input and haptic feedback. In *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 211–224, New York, NY, USA. ACM. (page 43)
- Kirsh, D. et Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18 :513–549. (pages 5 et 58)
- Kopf, J., Uyttendaele, M., Deussen, O. et Cohen, M. F. (2007). Capturing and viewing gigapixel images. *ACM Trans. Graph.*, 26(3) :93. (page 155)
- Kroemer, K. et Grandjean, E. (1997). *Fitting the Task to the Human*. Taylor & Francis Philadelphia, PA. (pages xiii, 109, 110 et 113)
- Kruger, R., Carpendale, S., Scott, S. D. et Greenberg, S. (2004). Roles of orientation in tabletop collaboration : Comprehension, coordination and communication. *Comput. Supported Coop. Work*, 13(5-6) :501–537. (pages 57 et 58)
- Kurtenbach, G. et Buxton, W. (1994). User learning and performance with marking menus. In *CHI '94 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 258–264, New York, NY, USA. ACM. (pages 42 et 112)

- Leung, Y. K. et Apperley, M. D. (1994). A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2) :126–160. (page 139)
- Lingrand, D., de Morais, W. O. et Tigli, J. Y. (2005). Ordinateur porté : dispositifs d'entrée sortie. In *IHM 2005 : Proceedings of the 17th international conference on Francophone sur l'Interaction Homme-Machine*, pages 219–222, New York, NY, USA. ACM. (pages 36 et 40)
- Lipscomb, J. S. et Pique, M. E. (1993). Analog input device physical characteristics. *SIGCHI Bull.*, 25(3) :40–45. (pages 38 et 39)
- Mackay, W. E. (2002). Which interaction technique works when ? floating palettes, marking menus and toolglasses support different task strategies. In Levialdi, S., editor, *Proceedings of the Conference on Advanced Visual Interface (AVI2002)*, pages 203–209. ACM Press. (page 99)
- Mackenzie, C. L. et Iberall, T. (1994). *The Grasping Hand (Advances in Psychology)*. North Holland, 1 edition. (pages 8 et 25)
- MacKenzie, I. S., Soukoreff, R. W. et Pal, C. (1997). A two-ball mouse affords three degrees of freedom. In *CHI '97 : CHI '97 extended abstracts on Human factors in computing systems*, pages 303–304, New York, NY, USA. ACM. (pages 47)
- Mackinlay, J. D., Heer, J. et Royer, C. (2003). Wideband visual interfaces : Sense-making on multiple monitors. Technical report, Xerox Parc. (pages 5 et 125)
- Mackinlay, J. D., Robertson, G. G. et Card, S. K. (1991). The perspective wall : Detail and context smoothly integrated. In *CHI '91 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 173–176, New York, NY, USA. ACM. (page 139)
- Malone, T. W. (1982). How do people organize their desks ? (extended abstract) : Implications for the design of office information systems. In *Proceedings of the SIGOA conference on Office information systems*, pages 47–49, New York, NY, USA. ACM. (pages 4 et 123)
- Martin, B. et Raisamo, R. (2004). Trackmouse : a new solution for 2+2d interactions. In *NordiCHI '04 : Proceedings of the third Nordic conference on Human-computer interaction*, pages 89–92, New York, NY, USA. ACM. (page 41)
- Masui, T., Kashiwagi, K. et George R. Borden, I. (1995). Elastic graphical interfaces to precise data manipulation. In *CHI '95 : Conference companion on Human factors in computing systems*, pages 143–144, New York, NY, USA. ACM. (pages 7 et 94)
- McCormick, E. (1970). *Human Factors Engineering*. McGraw-Hill, New York, NY, USA. (pages 33)
- Moggridge, B. (2006). *Designing Interactions*, chapter 1. The Mouse and the Desktop. The MIT Press. (pages 6, 28 et 39)



- Muter, P. et Maurutto, P. (1991). Reading and skimming from computer screens and books : The paperless office revisited ? *Behaviour & Information Technology*, 10(4) :257–266. (page 130)
- Napier, J. (1956). The prehensile movements of the human hand. *Journal of Bone & Joint Surgery, British Volume*, 38(4) :902–913. (pages 25, 26 et 27)
- Norman, D. (2002). *The Design of Everyday Things*. Basic Books, New York. (pages 52 et 87)
- Norman, D. A. (1993). *Things that make us smart : defending human attributes in the age of the machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. (page 18)
- Norman, D. A. (1998). *The invisible computer*. MIT Press, Cambridge, MA, USA. (page 83)
- O'Hara, K. et Sellen, A. (1997). A comparison of reading paper and on-line documents. In *CHI '97 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 335–342, New York, NY, USA. ACM. (pages 1, 3, 123 et 134)
- Olwal, A. et Feiner, S. (2004). Unit : modular development of distributed interaction techniques for highly interactive user interfaces. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 131–138. ACM New York, NY, USA. (page 48)
- Omata, M., Matsumura, K. et Imamiya, A. (2007). A pressure-sensing mouse button for multilevel click and drag. In *INTERACT (1)*, volume 1, pages 434–446. (page 43)
- Oshita, M. (2005). Pen-to-mime : Pen-based interactive control of a human figure. *Computers & Graphics*, 29(6) :931–945. (page 91)
- Øvergård, K. I., Fostervold, K. I., Bjelland, H. V. et Hoff, T. (2007). Knobology in use : an experimental evaluation of ergonomics recommendations. *Ergonomics*, 50(5) :694–705. (pages 79, 113 et 115)
- Patrick, E., Cosgrove, D., Slavkovic, A., Rode, J. A., Verratti, T. et Chiselko, G. (2000). Using a large projection screen as an alternative to head-mounted displays for virtual environments. In *CHI '00 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 478–485, New York, NY, USA. ACM. (pages 125)
- Patten, J., Recht, B. et Ishii, H. (2002). Audiopad : a tag-based interface for musical performance. In *NIME '02 : Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore. National University of Singapore. (page 82)

- Perlin, K. et Fox, D. (1993). Pad : an alternative approach to the computer interface. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64, New York, NY, USA. ACM Press. (page 3)
- Pheasant, S. et Haslegrave, C. (2006). *Bodyspace : anthropometry, ergonomics, and the design of work*. Taylor & Francis : London, UK, London, UK, 3rd ed. edition. (page 8)
- Plaisant, C., Carr, D. et Shneiderman, B. (1995). Image-browser taxonomy and guidelines for designers. *IEEE Softw.*, 12(2) :21–32. (pages 138, 141 et 163)
- Poincaré, H. (1902). *La Science et l'hypothèse*, page 81. Champs Flammarion (1968). (page 19)
- Poston, T., Srikanth, M. et Vaidya, P. (2006). Sensing Linear and Angular Change Through a Small Optical Window. Technical report, Indian Institute of Science. (page 55)
- Ramos, G., Boulos, M. et Balakrishnan, R. (2004). Pressure widgets. In *CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494, New York, NY, USA. ACM. (page 104)
- Rao, R., Pedersen, J. O., Hearst, M. A., Mackinlay, J. D., Card, S. K., Masinter, L., Halvorsen, P.-K. et Robertson, G. C. (1995). Rich interaction in the digital library. *Commun. ACM*, 38(4) :29–39. (pages 10 et 118)
- Rauber, A. et Bina, H. (2000). Visualizing electronic document repositories : Drawing books and papers in a digital library. In *VDB 5 : Proceedings of the Fifth Working Conference on Visual Database Systems*, page 95, Deventer, The Netherlands, The Netherlands. Kluwer, B.V. (page 122)
- Rekimoto, J. et Sciammarella, E. (2000). Toolstone : effective use of the physical manipulation vocabularies of input devices. In *UIST '00 : Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 109–117, New York, NY, USA. ACM. (pages 24, 50 et 51)
- Riley, J. et Whitsel, K. (2005). Practical quality control procedures for digital imaging projects. *OCLC Systems & Services : International Digital Library Perspectives*, 21(1). (pages 137)
- Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risdén, K., Thiel, D. et Gorokhovskiy, V. (2000). The task gallery : a 3d window manager. In *CHI '00 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 494–501, New York, NY, USA. ACM. (page 59)
- Robertson, G. G. et Mackinlay, J. D. (1993). The document lens. In *UIST '93 : Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 101–108, New York, NY, USA. ACM. (page 140)

- Rosenfeld, L. et Morville, P. (1998). *Information architecture for the World Wide Web*. O'Reilly & Associates, Inc. Sebastopol, CA, USA. (pages 118 et 119)
- Sellen, A., Guiard, Y. et Buxton, W. (2002). Two-handed input (chapter 11), in *Human Input to Computer Systems : Theories, Techniques and Technology*. (unfinished book). (pages 3 et 19)
- Shepard, R. et Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171(3972) :701. (page 58)
- Shneiderman, B. (1987). Direct manipulation : A step beyond programming languages. In *Human-computer interaction : a multidisciplinary approach*, pages 461–467. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (page 2)
- Shupp, L., Yost, B. et North, C. (2009). Shaping the Display of the Future : The Effects of Display Size and Curvature on User Performance and Insights. (to appear in) *Human-Computer Interaction*. (page 125)
- Simon, H. A. (1969). *Les sciences de l'artificiel*. Gallimard (1996). (page 18)
- Snibbe, S. S., MacLean, K. E., Shaw, R., Roderick, J., Verplank, W. L. et Scheeff, M. (2001). Haptic techniques for media control. In *UIST '01 : Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 199–208, New York, NY, USA. ACM Press. (pages 3 et 161)
- Sugiyama, K. et al. (2000). Angle information input apparatus. US Patent 6,075,468. (page 45)
- Sutherland, I. E. (1964). Sketch pad a man-machine graphical communication system. In *DAC '64 : Proceedings of the SHARE design automation workshop*, pages 6.329–6.346, New York, NY, USA. ACM. (page 31)
- Tian, F., Ao, X., Wang, H., Setlur, V. et Dai, G. (2007). The tilt cursor : enhancing stimulus-response compatibility by providing 3d orientation cue of pen. In *CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 303–306, New York, NY, USA. ACM. (page 104)
- Tian, F., Xu, L., Wang, H., Zhang, X., Liu, Y., Setlur, V. et Dai, G. (2008). Tilt menu : using the 3d orientation information of pen devices to extend the selection capability of pen-based user interfaces. In *CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1371–1380, New York, NY, USA. ACM. (page 104)
- Topol, A. (2000). Immersion of xwindow applications into a 3d workbench. In *CHI '00 : CHI '00 extended abstracts on Human factors in computing systems*, pages 355–356, New York, NY, USA. ACM. (pages 4, 59 et 61)
- Topol, A. (2002). *Interaction 3D pour les paysages informationnels*. Thèse de doctorat en informatique. PhD thesis, Conservatoire national des arts et métiers. (pages 11, 126 et 149)

- University of Michigan Library (2009). One Million Digitized Books (<http://www.lib.umich.edu/news/millionth.html>). (page 137)
- Viviani, P. (1994). Les habiletés motrices. In M. Richelle and J. Requin and M. Robert (Eds.), *Traité de psychologie expérimentale*, volume 1, pages 777–857. PUF (Presses Universitaires de France). (page 23)
- Wanderley, M. M., Viollet, J.-P., Isart, F. et Rodet, X. (2000). On the choice of transducer technologies for specific musical functions. In *ICMC : International Computer Music Conference*, Berlin, Allemagne. (page 20)
- Wang, J. et Mankoff, J. (2003). Theoretical and architectural support for input device adaptation. In *CUU '03 : Proceedings of the 2003 conference on Universal usability*, pages 85–92, New York, NY, USA. ACM. (page 36)
- Wang, S., Poturalski, M. et Vronay, D. (2005). Designing a generalized 3d carousel view. In *CHI '05 : CHI '05 extended abstracts on Human factors in computing systems*, pages 2017–2020, New York, NY, USA. ACM. (page 58)
- Wang, Y., MacKenzie, C. L. et Summers, V. A. (1997). Object manipulation in virtual environments : human bias, consistency and individual differences. In *CHI '97 : CHI '97 extended abstracts on Human factors in computing systems*, pages 349–350, New York, NY, USA. ACM. (page 8)
- Wang, Y., MacKenzie, C. L., Summers, V. A. et Booth, K. S. (1998). The structure of object transportation and orientation in human-computer interaction. In *CHI '98 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 312–319, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. (pages 22, 23 et 62)
- Ware, C. et Arsenault, R. (2004). Frames of reference in virtual object rotation. In *APGV '04 : Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 135–141, New York, NY, USA. ACM. (page 24)
- Ware, C. et Rose, J. (1999). Rotating virtual objects with real handles. *ACM Trans. Comput.-Hum. Interact.*, 6(2) :162–180. (pages 18, 23 et 25)
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3) :94–104. (page 2)
- Wellner, P. (1991). The digitaldesk calculator : tangible manipulation on a desk top display. In *UIST '91 : Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 27–33, New York, NY, USA. ACM. (page 4)
- Williams, L. (1983). Pyramidal parametrics. In *SIGGRAPH '83 : Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA. ACM. (page 130)

- Zeleznik, R., Miller, T. et Forsberg, A. (2001). Pop through mouse button interactions. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 195–196. ACM New York, NY, USA. (page 43)
- Zhai, S. et MacKenzie, I. S. (1998). Teaching Old Mice New Tricks : Innovations in Computer Mouse Design. In *Proceedings of Ergon-Axia'98*, pages 80–83. (page 47)
- Zhai, S., Milgram, P. et Buxton, W. (1996). The influence of muscle groups on performance of multiple degree-of-freedom input. In *CHI '96 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 308–315, New York, NY, USA. ACM. (pages 27, 28 et 65)
- Zhai, S., Smith, B. A. et Selker, T. (1997). Improving browsing performance : A study of four input devices for scrolling and pointing tasks. In *INTERACT '97 : Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction*, pages 286–293, London, UK, UK. Chapman & Hall, Ltd. (pages 7, 32, 51, 129 et 150)



**Auteur :** Rodrigo ANDRADE BOTELHO DE ALMEIDA

---

### **Contribution aux techniques pour enrichir l'espace moteur et l'espace visuel des dispositifs d'interaction bureautique**

De nombreux travaux ont montré que, à l'origine des limitations de l'interaction bureautique, il y a un manque à la fois d'espace moteur et d'espace visuel. Cette thèse explore des moyens pour optimiser l'usage de ces espaces.

D'une part, à partir du constat que l'on contrôle la position et l'orientation d'un objet par un geste naturel, cette thèse étudie les bénéfices que peut offrir une souris dotée d'un capteur de rotation. Cette « souris rotative » permet à l'utilisateur de maîtriser avec aisance trois variables continues d'une tâche informatique. Un état de l'art présente des aspects perceptifs et moteurs des actions en question et les particularités ergonomiques et techniques d'un tel dispositif. Deux techniques d'interaction - visant à faciliter des tâches métier récurrentes - sont proposées : le réglage « quasi-intégral » et la « palette satellitaire ». Par ailleurs, une évaluation expérimentale compare la performance d'une souris rotative avec celle d'une souris traditionnelle.

D'autre part, ce travail se penche sur les questions de la visualisation de documents dans le contexte des bibliothèques numériques. D'abord, il examine l'apport et la faisabilité technique de l'utilisation d'un dispositif d'affichage immersif pour la navigation dans un catalogue de titres virtuel. Puis, afin de faciliter l'inspection massive d'un lot de pages numérisées, il avance des techniques de visualisation zoomables et multi-focales. Ces dernières permettent, dans une recherche d'anomalies, de saisir vite les caractéristiques visuelles de quelques centaines de pages. Et cela grâce à un va-et-vient entre la vue d'ensemble et la navigation panoramique des détails.

**Mots clés :** interaction 3D, dispositifs d'entrée, techniques d'interaction, visualisation de documents, bibliothèques numériques, dispositifs d'affichage immersif

---

### **Contribution to techniques that enhance the motor space and the visual space of desktop interaction devices**

Past research has suggested that among the reasons for the limitations of present desktop interaction style is the lack of both motor and visual space. The goal of this thesis is to optimize the use of such spaces.

Based on the fact that one can control an object's position and orientation through a natural movement, the first main contribution of this thesis is to explore the advantages of enhancing the sensing of the standard mouse through a rotation sensor. This « rotary mouse » allows one to easily control three continuous variables of a computer task. A survey presents the perceptual and motor issues of some rotary manipulations and also the technical and ergonomic requirements of such device. Two interaction techniques, aimed to simplify repetitive tasks, are proposed : the « nearly-integral selection » and the « satellite palette ». Furthermore, an experimental evaluation compares the performance of the rotary mouse with that of a standard one.

The other main contribution of this work is to investigate document visualization issues in the context of digital libraries. First, it analyses the advantages and the technical feasibility of integrating an immersive display to an interface aimed to support navigation in a virtual catalog. Second, in order to inspect the quality of a batch of digitized pages, it explores some zoomable and multi-focal visualization techniques. The overview and the panoramic detail browsing enabled by such techniques try to help users, which have to identify the flaws resulted from the digitization process, to quickly grasp the visual characteristics of a large set of pages.

**Keywords :** 3D interaction, input devices, interaction techniques, document visualization, digital libraries, immersive display devices

---

Discipline : **Informatique**, spécialité : **Interaction Homme-Machine**

Thèse préparée au **Cedric** - Laboratoire d'étude et recherche en informatique du CNAM, EA 1395 - dans le cadre de l'EDITE de Paris. 292, rue Saint Martin, FR-75141 Paris Cedex 03, France.